

A PRACTICAL FACTORIZATION OF A SCHUR COMPLEMENT FOR PDE-CONSTRAINED DISTRIBUTED OPTIMAL CONTROL*

YOUNGSOO CHOI[†], CHARBEL FARHAT[‡], WALTER MURRAY[§], AND MICHAEL SAUNDERS[¶]

Abstract. A distributed optimal control problem with the constraint of a linear elliptic partial differential equation is considered. A necessary optimality condition for this problem forms a saddle point system, the efficient and accurate solution of which is crucial. A new factorization of the Schur complement for such a system is proposed and its characteristics discussed. The factorization introduces two complex factors that are complex conjugate to each other. The proposed solution methodology involves the application of a parallel linear domain decomposition solver—FETI-DPH—for the solution of the subproblems with the complex factors. Numerical properties of FETI-DPH in this context are demonstrated, including numerical and parallel scalability and regularization dependence. The new factorization can be used to solve Schur complement systems arising in both range-space and full-space formulations. In both cases, numerical results indicate that the complex factorization is promising.

Key words. PDE-constrained optimization, Schur complement, Poisson operator, FETI, range-space method, full-space method, distributed optimal control

AMS subject classifications. 65N22, 65N55, 65F10, 65F50

1. Introduction. Numerical methods for solving partial differential equations (PDEs) have broad applications in the simulation of complicated physical models, the prediction of their response, and design. An important and practical subset of these applications—particularly for design—involves the use of a mathematical optimization technique in which the PDE takes the role of a constraint equation. Two methodologies for PDE-constrained optimization problems are SAND (simultaneous analysis and design) and NAND (nested analysis and design) [3]. NAND uses PDEs to express decision variables as an implicit function of state variables and does not include state variables as optimization variables. Thus, the size of the optimization problem is not typically large. On the other hand, the SAND approach takes both decision variables and state variables as optimization variables and considers PDEs to be equality constraints. Consequently, the size of the system in the SAND approach is generally much larger. The NAND approach has traditionally been the method of choice for physics-based applications not only because SAND requires the solution of a large-scale system of equations but also because NAND conveniently permits the direct use of existing solvers for both optimization and PDE simulation as a black box. However, NAND suffers from the fact that many PDE simulations are required for function evaluations—typically the most expensive part of this approach. Due to the continuous increase in computational power (e.g., speed of processing, capacity of memory, high performance computing) accompanied by the development of robust and versatile numerical algorithms (e.g., parallel algorithms), the SAND approach

*The first and second authors acknowledge partial support by the Army Research Laboratory through the Army High Performance Computing Research Center under Cooperative Agreement W911NF-07-2-0027. The third and fourth authors acknowledge partial support by the ONR grant N000141110067.

[†]Aeronautics and Astronautics, Stanford University, Stanford, CA, USA

[‡]Aeronautics and Astronautics, Mechanical Engineering, Stanford University, Stanford, CA, USA

[§]Management Science and Engineering, Stanford University, Stanford, CA, USA

[¶]Management Science and Engineering, Stanford University, Stanford, CA, USA

has received increasing attention from researchers in recent years [4, 5, 7, 31], and the present study continues this line of research.

The SAND approach to PDE-constrained optimization takes the form

$$\begin{aligned} & \underset{y,u}{\text{minimize}} && F(y,u) \\ & \text{subject to} && C(y,u) = 0, \end{aligned} \tag{1.1}$$

where $C(y,u) = 0$ is the time-independent PDE constraint, y is the vector of state variables, and u is the vector of decision variables. State variables by definition are the unknown variables in the forward PDE problem. For example, state variables comprise temperature in heat conduction problems and displacements in elastostatics. For the class of PDE-constrained optimization known as *optimal control* the decision variables u are referred to as control variables, whereas for *optimal design* or *shape optimization* problems the decision variables u are called design variables. The decision variables may also be a set of parameters describing the material properties or the system of dynamics for some inverse problems. All three types of PDE-constrained optimization problems share a similar structure of their linear or linearized system of equations known as a KKT system, after the Karush-Kuhn-Tucker optimality conditions, or a saddle point system.

In what follows, a robust and versatile numerical method for solving a distributed optimal control problem is considered. In particular, heat conduction and elastostatic PDE-constrained problems are studied, although the method developed here can also be applied or extended easily to other types of PDE-constrained optimization problems. From various possible objective functions that may be used to formulate a distributed PDE-constrained optimal control problem, the one considered is that in which a target state is assumed to be given. Thus, the aim is to find a state that is close to the prescribed target and a control that realizes that particular state. For example, the static thermal conduction optimal control problem with a target temperature distribution \bar{y} is formulated as

$$\begin{aligned} & \underset{y,u}{\text{minimize}} && F(y,u) := \frac{1}{2} \int_{\Omega} (y - \bar{y})^2 dx + \frac{\phi}{2} \int_{\Omega} u^2 dx \\ & \text{subject to} && -\nabla^2 y = u \text{ on } \Omega \\ & && y = y_c \text{ on } \Gamma_g. \end{aligned} \tag{1.2}$$

The solution of this problem is considered in Section 6.1. Variables y and u are the temperature state and heat source control, respectively, while y_c is the prescribed boundary conditions and ϕ is a regularization parameter. The domain of interest is denoted as Ω and a Dirichlet boundary condition is imposed on the boundary Γ_g . Note that a unit conductivity is assumed.

Two alternative approaches to PDE-constrained optimization problems such as (1.2) are *optimize-and-discretize* and *discretize-and-optimize*. The latter approach, in which one first discretizes both the objective function and constraints and then obtains the discretized optimality condition, is typically used for PDE-constrained optimal control [5, 9, 30, 29, 32, 35, 11] and is followed here.

Solving the saddle point system representing the discretized optimality condition efficiently is crucial to the competitiveness of the SAND approach, in comparison with NAND. There are two methodologies for solving a saddle point system.

- In reduced-space methods, one attempts to reduce the size of a saddle point system by eliminating some variables and solving a smaller system [37, 38,

35, 4]. The range-space method and the null-space method are two popular reduced-space methods. In the range-space method, one solves for the dual variables first using the corresponding Schur complement and subsequently updates the primal variables, whereas the null-space method subdivides the variables algebraically into null-space variables and range-space variables using null-space and range-space bases of the constraint Jacobian.

- In full-space methods, one solves for the primal and dual variables simultaneously. The resultant system of equations becomes a sparse saddle point system and iterative methods are the only practical choice for large problems. Various efficient preconditioners for saddle point systems have been developed [27, 2, 22, 16, 32, 30, 10, 9, 23, 5, 36]. However, there is still motivation for further research in this area. For example, most preconditioners are not robust when ϕ is small. Recently, Pearson and Wathen have developed a new approximation of the Schur complement and used it to facilitate a regularization-robust preconditioner for a particular distributed optimal control problem [30]. Pearson et al. have further extended the usage of the aforementioned approximation to a broader range of optimal control problems [29].

The factorization of the Schur complement presented in this paper has a similar form to Pearson and Wathen's approximation, and furthermore is applied to the same particular distributed optimal control problem [30]. However the approach taken here differs from that of Pearson and Wathen in two ways. First and foremost, because the factorization is exact by its nature, the range-space method can be adopted and one efficient solve with the Schur complement results in a solution to the distributed optimal control problem. Second, a scalable domain decomposition based parallel linear solver FETI-DPH [13] is used to solve each of the subproblems arising in the application of the proposed factorization. In contrast, Pearson and Wathen suggest a multigrid method.

The outline of the paper is as follows: Section 2 describes discretization of the distributed optimal control problem (1.2) and the corresponding optimality condition, which is a saddle point system. Section 3 outlines two methods for solving the saddle point system: the range-space method and the full-space method. Additionally, this section reviews existing preconditioners for the full-space method related to the Schur complement in the range-space method. Section 4 introduces a practical factorization of the Schur complement and explains how it can be used in both the range-space method and the full-space method. Section 5 describes the Finite Element Tearing and Interconnecting (FETI) method and its so-called *dual-primal* variants, FETI-DP and FETI-DPH. Section 6 presents numerical results that illustrate the scalability and efficiency of the method applied to a selection of distributed optimal control problems.

2. Distributed control problems. In this section, the finite element discretization of a distributed optimal control problem is introduced and the corresponding optimality condition is presented. For brevity, we choose to show the discretization of a distributed optimal control problem with a constraint of the Poisson equation (i.e., Eq. (1.2)). However, distributed optimal control problems with other types of PDE constraints (e.g., linear elasticity) will have the same discretized formulation. The

finite element discretization of (1.2) gives

$$\begin{aligned} & \underset{\mathbf{y}, \mathbf{u}}{\text{minimize}} && F(\mathbf{y}, \mathbf{u}) := \frac{1}{2} \|\mathbf{y} - \bar{\mathbf{y}}\|_{\mathbf{V}}^2 + \frac{\phi}{2} \|\mathbf{u}\|_{\mathbf{V}}^2 \\ & \text{subject to} && \mathbf{K}\mathbf{y} + \mathbf{K}_c \mathbf{y}_c = \mathbf{V}\mathbf{u}, \end{aligned} \quad (2.1)$$

where $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times n}$, and $\mathbf{K}_c \in \mathbb{R}^{n \times m}$ are the stiffness matrix, volume matrix, and constrained stiffness matrix, respectively. Vector valued quantities $\bar{\mathbf{y}}, \mathbf{y}, \mathbf{u} \in \mathbb{R}^n$, and $\mathbf{y}_c \in \mathbb{R}^m$ are the discretized versions of the target state, state, control, and prescribed boundary conditions, respectively. All the discrete variables are denoted with bold fonts for the remainder of the paper. The dimensions n and m are the number of unconstrained degrees of freedom (i.e., the size of \mathbf{y}) and the number of constrained degrees of freedom, respectively. It is assumed that both \mathbf{K} and \mathbf{V} are symmetric positive definite (SPD) matrices, which is the case for the thermal problem above. The energy norm $\|\cdot\|_{\mathbf{V}}$ is defined as $\|\mathbf{q}\|_{\mathbf{V}} = \sqrt{\mathbf{q}^T \mathbf{V} \mathbf{q}}$. The details of the finite element discretization procedure can be found in [7]. Problem (2.1) is a convex quadratic program and its solution is a saddle point of the Lagrangian,

$$L(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{y} - \bar{\mathbf{y}}\|_{\mathbf{V}}^2 + \frac{\phi}{2} \|\mathbf{u}\|_{\mathbf{V}}^2 + \boldsymbol{\lambda}^T (\mathbf{K}\mathbf{y} + \mathbf{K}_c \mathbf{y}_c - \mathbf{V}\mathbf{u}). \quad (2.2)$$

A saddle point $(\mathbf{y}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$ must satisfy the following optimality condition:

$$\begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{K} \\ \mathbf{0} & \phi \mathbf{V} & -\mathbf{V} \\ \mathbf{K} & -\mathbf{V} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{y}^* \\ \mathbf{u}^* \\ \boldsymbol{\lambda}^* \end{pmatrix} = \begin{pmatrix} \mathbf{V}\bar{\mathbf{y}} \\ \mathbf{0} \\ -\mathbf{K}_c \mathbf{y}_c \end{pmatrix}. \quad (2.3)$$

This equation is of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, where \mathbf{A} is a symmetric indefinite matrix. Section 3 explains two ways of solving the saddle point system in (2.3): the range-space method and full-space method.

3. Range-space and full-space methods.

3.1. Range-space method. The range-space method was introduced for large-scale inequality-constrained convex quadratic programming problems with small active sets in order to overcome the disadvantage of the null-space method: the increasingly high dimension of the null space basis as a solution is approached [19, 20]. For equality-constrained convex quadratic programming, the range-space method is equivalent to the Schur complement method [37] where the corresponding dual problem is solved first. The range-space method is a reduced-space method in a sense that the size of the saddle point system in (2.3) is reduced by eliminating \mathbf{y}^* and \mathbf{u}^* and solving for $\boldsymbol{\lambda}^*$ first. The dual variable $\boldsymbol{\lambda}^*$ is obtained by solving

$$\mathbf{S}\boldsymbol{\lambda}^* = \mathbf{K}_c \mathbf{y}_c + \mathbf{K}\bar{\mathbf{y}}, \quad (3.1)$$

where $\mathbf{S} = \mathbf{K}\mathbf{V}^{-1}\mathbf{K} + \frac{1}{\phi}\mathbf{V}$ is known as the negative Schur complement on the dual variables. Then, \mathbf{y}^* and \mathbf{u}^* are computed from

$$\mathbf{y}^* = \bar{\mathbf{y}} - \mathbf{V}^{-1}\mathbf{K}\boldsymbol{\lambda}^* \quad \text{and} \quad \mathbf{u}^* = \frac{1}{\phi}\boldsymbol{\lambda}^*. \quad (3.2)$$

The most expensive and crucial step in the range-space method is to solve with \mathbf{S} in (3.1). The eigenvalues (and consequently, the condition number) of \mathbf{S} depend on those

of \mathbf{K} and \mathbf{V} and the value of ϕ [37, 7]. For small values of ϕ , the condition number of \mathbf{S} is affected most by the eigenvalues of \mathbf{V} , whose condition number is bounded above by a constant, C , for P_q or Q_q (the q^{th} order triangular or quadrilateral finite elements in 2D and tetrahedral or brick elements in 3D) if a set of grids is quasi-uniform (see Eq. (1.116) and Eq. (1.117) in [12]). The constant C depends on the order of approximation q but not on the mesh size h . Thus for small values of ϕ , even if the grid is refined or the problem size is increased, the condition number of \mathbf{S} is bounded. For large values of ϕ , the condition number of \mathbf{S} (denoted by $\kappa(\mathbf{S})$) depends on $\kappa(\mathbf{KV}^{-1}\mathbf{K})$, which is proportional to $\kappa(\mathbf{K})^2$. For a second-order elliptic operator, one can prove that $\kappa(\mathbf{K}) < ch^{-2}$ if P_q or Q_q is used on a quasi-uniform discretization of a domain (see Eq. (1.119) and Eq. (1.121) in [12]). For a higher-order elliptic operator, the dependence on h of $\kappa(\mathbf{K})$ increases (e.g., the shell or beam elements). This is unfortunate because $\kappa(\mathbf{K})$ is likely to increase as the mesh size h decreases or the problem size increases. One remedy is to apply iterative refinement with a high precision data type. If the solution is inaccurate even after iterative refinement due to the ill-conditioning of \mathbf{S} , an alternative way of solving (2.3) that avoids an exact solve with \mathbf{S} is to use the full-space method, which is explained in the next section.

An additional challenge in applying the range-space method for solving (2.3) resides in solving with \mathbf{S} itself. Because \mathbf{S} is a sum of two matrices ($\mathbf{KV}^{-1}\mathbf{K}$ and $\frac{1}{\phi}\mathbf{V}$), the first of which is a product requiring two matrix-matrix multiplications to evaluate, it is not straightforward to come up with an efficient way of solving with \mathbf{S} . In Section 4, a practical factorization of \mathbf{S} is introduced and the factorization does not require matrix-matrix products. The factorization introduced in Section 4 introduces complex symmetric matrices (not Hermitian matrices) in its factors. A list of efficient solvers for complex symmetric matrices includes CG-type methods [17], CS-MINRES-QLP [6], GMRES [34, 8], FETI-DPH [13], and multi-grid methods [24, 33]. FETI-DPH is explained in Section 5 and used in numerical experiments in Section 6.

3.2. Full-space Method. The full-space method attempts to solve for all the variables of the saddle point system in (2.3) simultaneously. As the discretization is refined, the size of the system becomes large and an iterative method is often the only available method. Because \mathbf{A} is a symmetric indefinite matrix, any Krylov iterative method suitable for this class of matrices, such as MINRES [28], SYMMLQ [28], SQMR [18], and GMRES, can be used. For successful convergence of the iterative method, one is often required to apply a good preconditioner. Even without any preconditioner, the saddle point system itself tends to be better conditioned for moderately large values of ϕ than the Schur complement in the range-space method [37]. If a good preconditioner is used, the full-space method is likely to have a better scalability than the range-space method, meaning that computational cost does not grow at an exponential rate as the problem size increases. Thus, many preconditioners have been developed recently. Here we focus on a Schur complement-based preconditioner because the primary interest is to discuss the usage of a Schur complement factorization that will be introduced in Section 4. Murphy, et al. [27] have shown that if the following block diagonal preconditioner is used, then the preconditioned system has at most three nonzero distinct eigenvalues:

$$\mathbf{P}_{mgw} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \phi\mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S} \end{bmatrix}, \quad (3.3)$$

where \mathbf{S} is the negative Schur complement defined in (3.1). This implies that the maximum number of iterations required for convergence in a Krylov iterative method is three if \mathbf{P}_{mgw} is used as a preconditioner. However, applying this preconditioner has previously been considered impractical because it requires solving with \mathbf{S} in order to apply \mathbf{P}_{mgw} . Even if it were practical to solve with \mathbf{S} efficiently and accurately, then the preferred approach would typically be to use the range-space method rather than to apply \mathbf{P}_{mgw} in the full-space method because the range-space method requires only one solve with \mathbf{S} , while the full space method with \mathbf{P}_{mgw} as a preconditioner is likely to require more than one solve with \mathbf{S} . There has been some research done on finding a good approximation of the Schur complement for use in a Schur complement-based preconditioner (e.g., \mathbf{P}_{mgw}) [32, 30, 29]. Particularly, Pearson and Wathen [30] have developed the following approximation, which is regularization-robust:

$$\begin{aligned}\mathbf{S}_p &= (\mathbf{K} + \frac{1}{\sqrt{\phi}}\mathbf{V})\mathbf{V}^{-1}(\mathbf{K} + \frac{1}{\sqrt{\phi}}\mathbf{V}) \\ &= \mathbf{S} + \frac{2}{\sqrt{\phi}}\mathbf{K},\end{aligned}\tag{3.4}$$

and proved that the eigenvalues of $\mathbf{S}_p^{-1}\mathbf{S}$ are between $\frac{1}{2}$ and 1 regardless of the mesh size h and ϕ . This implies theoretically that a Krylov iterative method must converge in $O(1)$ iterations regardless of h and ϕ . They have used an algebraic multi-grid method to solve with the factor $(\mathbf{K} + \frac{1}{\sqrt{\phi}}\mathbf{V})$ in their numerical examples and demonstrated that the number of iterations required for convergence is indeed $O(1)$. However, in the problems they solve, the condition number of the Schur complement is small enough so that the approximation to the Schur complement behaves well. The factorization introduced in Section 4 has a similar form to (3.4). However, it is an exact representation of \mathbf{S} and therefore enables the application of the range-space method.

4. A “practical” factorization of the Schur complement. The Schur complement \mathbf{S} in (3.1) can be factored into the following form:

$$\mathbf{S}_c = (\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V})\mathbf{V}^{-1}(\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}),\tag{4.1}$$

where $i = \sqrt{-1}$. The form of this factorization is similar to \mathbf{S}_p in (3.4) in the sense that the first and third factors are some linear combinations of \mathbf{K} and \mathbf{V} , and the middle factor is \mathbf{V}^{-1} . This suggests that the same methods for solving with the factors in \mathbf{S}_p may also be used to solve with factors in \mathbf{S}_c . However, there are some important differences between \mathbf{S}_p and \mathbf{S}_c . The first and third factors of \mathbf{S}_p are the same, but the corresponding factors of \mathbf{S}_c are not. Consequently, if direct solvers were to be used for each factor, then only one factorization would be required for \mathbf{S}_p , while two factorizations of two different systems would be required for \mathbf{S}_c . Additionally, \mathbf{S}_c introduces complex numbers, but all the elements in \mathbf{S}_p are real. Thus, one solve with \mathbf{S}_c requires four times more storage and floating point operations than one solve with \mathbf{S}_p . In spite of the disadvantages of applying \mathbf{S}_c , it is an exact representation of \mathbf{S} , and this permits use of the range-space method where only one solve with \mathbf{S}_c is sufficient to obtain a solution to (2.1). On the other hand, many solves with \mathbf{S}_p are required because \mathbf{S}_p is an approximation to \mathbf{S} , so it can only be used as a preconditioner. This discussion leads to the following two extreme cases when dealing with the Schur complement:

- If a direct method is the preferred choice for solving the Schur complement (i.e., a factorization of a given system is required), it would be advantageous to apply the direct method to \mathbf{S}_p assuming that the dominating computational cost occurs in factorization of the system.
- If an iterative method is the only option and the range-space method is applicable, then it would be preferable to apply the iterative method to \mathbf{S}_c .

Between these two extreme cases, a choice must be made depending on the characteristics of the problem and a numerical solver. For example, if the domain of a problem is complex, then the computational overhead incurred by applying \mathbf{S}_c rather than \mathbf{S}_p will be substantially diminished since in this case the factors of both \mathbf{S}_c and \mathbf{S}_p will be complex. Such problems include any frequency domain analyses with damping in structural or acoustic problems. On the other hand, if a problem requires many solves with a Schur complement and with multiple right-hand sides and is small enough to permit a direct solver to be used for the factors of \mathbf{S} , applying \mathbf{S}_p is favorable because direct methods are more efficient than iterative methods in general for multiple right-hand sides. If a domain decomposition method—in which both direct and iterative methods are used—is chosen, then one needs to examine the costs of each component of the computation (e.g., data structure, building and storing the operator, and solving with the operator). In Section 6, numerical results for solving with \mathbf{S}_c are shown using the domain decomposition solver FETI-DPH [13].

5. FETI. In order to be self-contained, the FETI method [15] and two of its variants (FETI-DP [14] and FETI-DPH [13]) are explained in this section. For a more detailed description, see [15, 14, 13, 1]. The FETI method was developed in order to solve, in parallel, the following linear system of equations arising from the finite element discretization of a linear elasticity PDE:

$$\mathbf{K}\mathbf{y} = \mathbf{f}, \quad (5.1)$$

where \mathbf{K} is the stiffness matrix (or a linear combination of the stiffness matrix and mass matrix), \mathbf{y} is the vector of unknown displacements, and \mathbf{f} is an external force term. Solving (5.1) is equivalent to minimizing a quadratic function:

$$\underset{\mathbf{y}}{\text{minimize}} \quad \frac{1}{2}\mathbf{y}^T\mathbf{K}\mathbf{y} - \mathbf{y}^T\mathbf{f}. \quad (5.2)$$

The objective function in (5.2) often represents a physical quantity (e.g., energy in linear elasticity). In domain decomposition methods, the spatial domain Ω is divided into N_s either overlapping or non-overlapping subdomains. The FETI method divides Ω into N_s non-overlapping subdomains as illustrated in Figure 5.1 and minimizes a local function in each subdomain (i.e., minimize $\mathbf{y}^{sT}\mathbf{K}^s\mathbf{y}^s - \mathbf{y}^{sT}\mathbf{f}^s$ in Ω^s where the superscript “s” designate the restrictions to the specific subdomain). In order to be equivalent to (5.2), the following additional continuity condition on \mathbf{y}^s between interfaces must be imposed:

$$\sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{y}^s = \mathbf{0}, \quad (5.3)$$

where \mathbf{B}^s is a signed Boolean matrix. Hence one can write the following constrained quadratic programming equivalent to (5.2):

$$\begin{aligned} & \underset{\mathbf{y}^s; s=1, \dots, N_s}{\text{minimize}} && \sum_{s=1}^{N_s} \left(\frac{1}{2}\mathbf{y}^{sT}\mathbf{K}^s\mathbf{y}^s - \mathbf{y}^{sT}\mathbf{f}^s \right) \\ & \text{subject to} && \sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{y}^s = \mathbf{0}. \end{aligned} \quad (5.4)$$

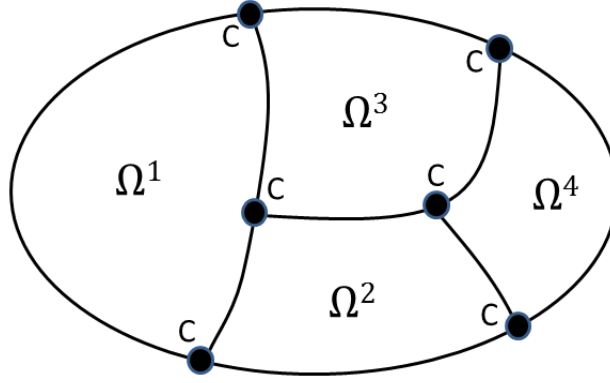


FIG. 5.1. Illustration of 4 non-overlapping subdomains. The sample corner nodes are denoted by C . FETI-DP enforces continuity on these corner nodes.

The Lagrangian for (5.4) is

$$L(\mathbf{y}^s, \boldsymbol{\lambda}; s = 1, \dots, N_s) = \sum_{s=1}^{N_s} \left(\frac{1}{2} \mathbf{y}^{sT} \mathbf{K}^s \mathbf{y}^s - \mathbf{y}^{sT} \mathbf{f}^s \right) + \boldsymbol{\lambda}^T \sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{y}^s. \quad (5.5)$$

For simplicity of notation, we will omit the writing of $s = 1, \dots, N_s$. The variable \mathbf{y}^s either means an individual subvector on Ω^s or a set of subvectors over all the subdomains. Because (5.4) is convex and only linear equality constraints are present, Slater's condition holds. Therefore strong duality also holds. Thus, an optimal solution \mathbf{y}^{s*} to (5.4) can be obtained from a saddle point $(\mathbf{y}^{s*}, \boldsymbol{\lambda}^*)$ to (5.5), that is,

$$\sup_{\boldsymbol{\lambda}} \inf_{\mathbf{y}^s} L(\mathbf{y}^s, \boldsymbol{\lambda}) = L(\mathbf{y}^{s*}, \boldsymbol{\lambda}^*) = \inf_{\mathbf{y}^s} \sup_{\boldsymbol{\lambda}} L(\mathbf{y}^s, \boldsymbol{\lambda}). \quad (5.6)$$

Statement (5.6) says that one can find an optimal solution \mathbf{y}^{s*} in two ways. The first way is to minimize $L(\mathbf{y}^s, \boldsymbol{\lambda})$ with respect to \mathbf{y}^s , then maximize the resultant with respect to $\boldsymbol{\lambda}$. The second way is to maximize $L(\mathbf{y}^s, \boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$, then minimize the resultant with respect to \mathbf{y}^s . The FETI method takes the first approach where the Lagrange dual function $g(\boldsymbol{\lambda}) = \inf_{\mathbf{y}^s} L(\mathbf{y}^s, \boldsymbol{\lambda})$ is obtained and then maximized in order to obtain the Lagrange multipliers first. This approach is attractive if the number of constraints (i.e., interface continuity condition) is small, which is expected to be the case in general.

For a fixed $\boldsymbol{\lambda}$, $L(\mathbf{y}^s, \boldsymbol{\lambda})$ is separable in \mathbf{y}^s (i.e., $\left(\frac{1}{2} \mathbf{y}^{sT} \mathbf{K}^s \mathbf{y}^s - \mathbf{y}^{sT} \mathbf{f}^s + \boldsymbol{\lambda}^T \mathbf{B}^s \mathbf{y}^s \right)$ on Ω^s). Thus, the Lagrange dual function can be obtained by solving $\mathbf{K}^s \mathbf{y}^s = \mathbf{f}^s - \mathbf{B}^s \boldsymbol{\lambda}$ on each subdomain Ω^s . In the case of \mathbf{K} being a stiffness matrix as in (5.1), \mathbf{K}^s is symmetric positive definite or semidefinite. If \mathbf{K}^s is positive semidefinite, then it is necessary to explicitly ensure that $\mathbf{K}^s \mathbf{y}^s = \mathbf{f}^s - \mathbf{B}^s \boldsymbol{\lambda}$ is compatible (i.e., $\mathbf{R}^{sT}(\mathbf{f}^s - \mathbf{B}^s \boldsymbol{\lambda}) = \mathbf{0}$ where the columns of \mathbf{R}^s span the left null space of \mathbf{K}^s). Otherwise, the minimum of $\left(\frac{1}{2} \mathbf{y}^{sT} \mathbf{K}^s \mathbf{y}^s - \mathbf{y}^{sT} \mathbf{f}^s + \boldsymbol{\lambda}^T \mathbf{B}^s \mathbf{y}^s \right)$ in terms of \mathbf{y}^s becomes $-\infty$ and the objective value of primal problem (5.4) is also $-\infty$ by strong duality. However, this is not a physical solution. Thus, we restrict ourselves to the case when $g(\boldsymbol{\lambda}) > -\infty$.

Finally, the Lagrange dual function $g(\boldsymbol{\lambda}) = \inf_{\mathbf{y}^s} L(\mathbf{y}^s, \boldsymbol{\lambda})$ is defined as

$$g(\boldsymbol{\lambda}) = \begin{cases} -\frac{1}{2}\boldsymbol{\lambda}^T \mathbf{F}\boldsymbol{\lambda} + \mathbf{d}^T \boldsymbol{\lambda} - \mathbf{c} & \text{if } \mathbf{R}^{sT}(\mathbf{f}^s - \mathbf{B}^{sT}\boldsymbol{\lambda}) = \mathbf{0} \text{ for } \forall s, \\ -\infty & \text{otherwise,} \end{cases} \quad (5.7)$$

where

$$\mathbf{F} = \sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{K}^{s+} \mathbf{B}^{sT}, \quad \mathbf{d} = \sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{K}^{s+} \mathbf{f}^s, \quad \mathbf{c} = \frac{1}{2} \sum_{s=1}^{N_s} \mathbf{f}^{sT} \mathbf{K}^{s+} \mathbf{f}^s, \quad (5.8)$$

and \mathbf{K}^{s+} is a pseudo-inverse of \mathbf{K}^s . This defines the following dual problem to (5.4):

$$\begin{aligned} & \underset{\boldsymbol{\lambda}}{\text{maximize}} && -\frac{1}{2}\boldsymbol{\lambda}^T \mathbf{F}\boldsymbol{\lambda} + \mathbf{d}^T \boldsymbol{\lambda} \\ & \text{subject to} && \mathbf{R}^{sT}(\mathbf{f}^s - \mathbf{B}^{sT}\boldsymbol{\lambda}) = \mathbf{0} \quad \text{for } \forall s. \end{aligned} \quad (5.9)$$

The FETI method solves (5.9) with a Preconditioned Conjugate Projected Gradient (PCPG) algorithm. The FETI formulation above is suitable for parallel processing. Any subdomain level computations (e.g., factorization or computation with \mathbf{K}^s and \mathbf{f}^s) can be assigned to an individual process. The size of $\boldsymbol{\lambda}$ is the total number of degrees of freedom restricted to the interfaces between subdomains. Within PCPG, $\boldsymbol{\lambda}$ is projected onto the domain of feasibility, which in turn reduces the effective dimension of the problem. Indeed the FETI method equipped with the Dirichlet preconditioner is proven numerically scalable with respect to both problem size and number of subdomains. The Dirichlet preconditioner is defined as

$$\mathbf{P}^{-1} = \mathbf{W} \sum_{s=1}^{N_s} \mathbf{B}^s \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{bb}^s \end{bmatrix} \mathbf{B}^{sT} \mathbf{W}, \quad (5.10)$$

where

$$\mathbf{W} = \left(\sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{B}^{sT} \right)^+, \quad \mathbf{S}_{bb}^s = \mathbf{K}_{bb}^s - \mathbf{K}_{ib}^{sT} \mathbf{K}_{ii}^{s-1} \mathbf{K}_{ib}^s, \quad (5.11)$$

and the subscript i and b denote subdomain internal degrees of freedom and interface degrees of freedom, respectively. If it is applied in the conjugate gradient algorithm with the projected gradient [21] to second-order elliptic problems, the condition number κ of the interface problem (5.9) is approximately [26]

$$\kappa = O(1 + \log^m(H/h)), \quad m \leq 3. \quad (5.12)$$

However, the first-generation FETI method is not numerically scalable for fourth-order plate and shell problems. This leads to FETI-DP. FETI-DP is one variant considered to be the third-generation FETI method. FETI-DP enforces continuity at some interface corner nodes at each iteration (see Figure 5.1). An extra coarse problem needs to be solved [14] and in each subdomain the remaining subdomain stiffness matrix \mathbf{K}_{rr}^s (i.e., that excluding the degrees of freedom in the corner nodes at which the continuity is enforced) becomes positive definite. Consequently, the corresponding dual interface problem becomes an unconstrained QP. This treatment helps to achieve numerical scalability for fourth-order elliptic problems. The continuity constraints can be augmented by additional constraints that are enforced exactly throughout the

iterations in FETI-DP in order to accelerate the convergence. This augmentation procedure results in the augmented coarse problem in FETI-DP [25]. A standard augmentation procedure uses the edge-based rigid body modes (rotational and/or translational) [13]. The positive definiteness of \mathbf{K}_{rr}^s is not guaranteed when \mathbf{K} in (5.1) is different from the stiffness matrix. For example, for a Helmholtz problem or frequency response elastodynamic problem, \mathbf{K} in (5.1) becomes

$$\mathbf{Z} \equiv \mathbf{K} - k^2 \mathbf{M} + i\mathbf{C}, \quad (5.13)$$

where \mathbf{C} is a symmetric matrix that arises from the discretization of an absorbing boundary condition, and $k > 0$ is a frequency (or a wave number for acoustic scattering problems). In this case, depending on the value of k , \mathbf{Z} may become indefinite. The same difficulty is encountered when solving with complex factors in (4.1). In order to resolve this difficulty, two special treatments were required.

- The first treatment is required to deal with indefinite matrices. A solver that is suitable for indefinite matrices must be applied. Such an iterative solver includes GMRES [34] for a general square matrix and MINRES [28] for a symmetric indefinite matrix. FETI-DPH ('H' stands for Helmholtz) [13] uses GMRES to deal with indefiniteness.
- The second treatment is required for improved convergence when the wave number of frequency is large. A particular augmentation used in FETI-DPH that accelerates convergence for the Helmholtz or frequency response elastodynamic problem is the free-space solutions of the corresponding equations, which are plane waves.

Note that the complex factors in Section 4.1 are similar to (5.13), but not identical, and not surprisingly the plane wave augmentation was found to not provide the same beneficial effects as in the Helmholtz problem. Thus, the edge-based rigid body modes (both rotational and translational) augmentation is applied in the next section for two numerical examples. Strictly speaking, although FETI-DPH normally refers to the variant of FETI-DP that uses both GMRES and plane wave augmentation, in what follows, FETI-DPH refers to a FETI-DP solver incorporating GMRES and edge-based rigid body modes.

6. Numerical Results. Two pedagogical examples are considered: linear static heat control of a 2D square plate and structure control of a 3D solid cantilever. The heat control problem in Section 6.1 is taken identical to the thermal problem solved in the paper by Rees et al. [32]. The Schur complement that arises in this thermal problem is sufficiently well-conditioned for the range of ϕ values and mesh size h considered here for the range-space method to be applicable and therefore it is the only method used. The cantilever control problem in Section 6.2 is very flexible with material properties of rubber. For a relatively large ϕ value, the range-space method fails to converge to an accurate solution. Thus, the range-space method is used for small values of ϕ only, and the full-space method is used for large values of ϕ . In the case of the full-space method, both the approximate Schur complement proposed by Pearson and Wathen [30] and the complex factorization introduced in this paper are used in a Schur-complement based preconditioner and compared in terms of computational time and iteration counts.

It is well established that augmentation is a beneficial feature that improves the performance of FETI. However, the optimality of the augmentation's performance depends on the nature of the problem. Because neither the FETI solver nor any of its variants has ever been applied to a system of the form $\mathbf{K} \pm \frac{1}{i\sqrt{\phi}} \mathbf{V}$, the numerical

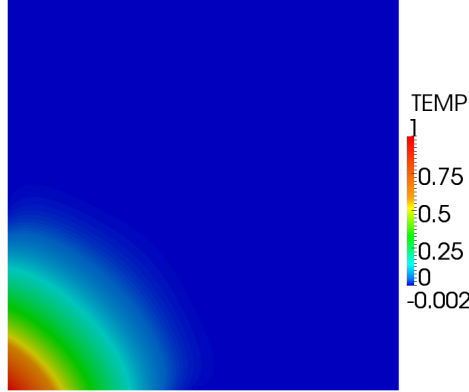


FIG. 6.1. Target temperature defined as in (6.2).

performance of FETI-DPH with and without augmentation is compared in this section. All the simulations were run on a heterogeneous Linux cluster with 2.6 GHz hexa-core Westmere processors (32 blades, 2 processors per blade, QDR Infiniband interconnect, 24GB/blade) and 2.6 GHz octa-core Sandybridge processors (4 blades, 2 processors per blade, FDR Infiniband interconnect, 256 GB/node).

6.1. Thermal problem. A linear static heat control problem is solved. This example is identical to example 5.1 in [32] by Rees et al. The same objective function and PDE constraints are used as in (1.2), whose continuous formulation is reproduced here for better access:

$$\begin{aligned} & \underset{y,u}{\text{minimize}} && F(y,u) := \frac{1}{2} \int_{\Omega} (y - \bar{y})^2 dx + \frac{\phi}{2} \int_{\Omega} u^2 dx \\ & \text{subject to} && -\nabla^2 y = u \text{ on } \Omega \\ & && y = y_c \text{ on } \Gamma_g. \end{aligned} \tag{6.1}$$

The domain Ω is $[0, 1]^2 \subset \mathbb{R}^2$, which is a unit square plate, whose heat conductivity is unity. The target temperature \bar{y} is defined as

$$\bar{y} = \begin{cases} (2x_1 - 1)^2(2x_2 - 1)^2 & \text{if } (x_1, x_2) \in [0, \frac{1}{2}]^2, \\ 0 & \text{otherwise,} \end{cases} \tag{6.2}$$

which is illustrated in Figure 6.1. The boundary condition y_c is defined as

$$y_c = \bar{y} \quad \text{on } \Gamma_g = \partial\Omega = \{(x_1, x_2) \mid x_1 \in \{0, 1\}, x_2 \in \{0, 1\}\}. \tag{6.3}$$

The optimal control problem (6.1) tries to find a temperature y that is close to the target temperature \bar{y} by controlling heat u . How close y can be to \bar{y} is determined by the parameter ϕ . As ϕ decreases, y is expected to approach \bar{y} , but $\|u\|$ is also expected to increase. This makes sense because the objective function value is not sensitive to the second term if the parameter ϕ is small and the first term dominates the objective function value. The results shown in Figure 6.2 confirm the validity of these expectations. For this particular example, according to Figure 6.2(a), one needs ϕ less than 2×10^{-6} in order to match the target temperature to within 1%.

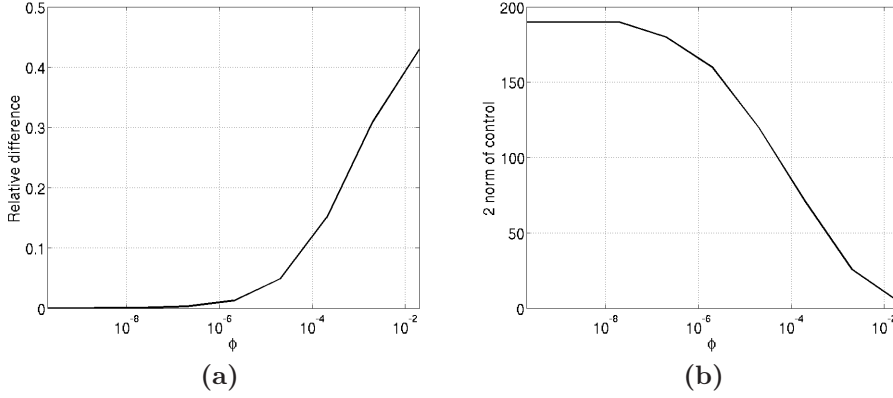


FIG. 6.2. (a) Graph of ϕ vs relative difference between temperature solution and target temperature. (b) Graph of ϕ vs norm of control solution.

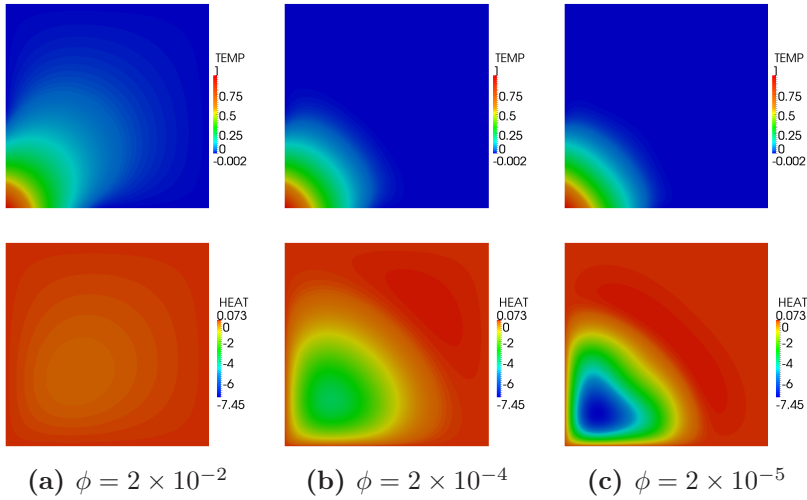


FIG. 6.3. Temperature (the upper three figures) and heat distribution solutions (the lower three figures) for various ϕ values.

According to Figure 6.2(b), however, one needs to set ϕ greater than 2×10^{-4} in order for the norm of control solution not to exceed 100.

The same effects can be demonstrated visually in Figure 6.3. Temperature and heat distributions for various ϕ values are shown in Figures 6.3(a)-(c). Note that for $\phi = 0.02$, the heat is almost zero everywhere and the corresponding temperature distribution (induced mainly by the boundary condition y_c) is slightly different from the target temperature (i.e., Figure 6.1). The target temperature can be matched more precisely if a smaller ϕ is used. Figures 6.3(b) and (c) show temperature distributions that are closer to the target temperature. They are produced by using smaller ϕ values (i.e., $\phi = 2 \times 10^{-4}$ and 2×10^{-5}). Note that the corresponding heat distributions show noticeable non-zero heat at the left bottom of the domain. As ϕ decreases, a sharper heat gradient is visible near the boundary.

Table 6.1 shows FETI-DPH's dependence on ϕ and the effects of augmentation

TABLE 6.1
FETI-DPH's ϕ dependency for the heat control problem.

ϕ	-	+	CPU	RAUG-	RAUG+	CPU
2E-2	33	32	9.4	13	13	6.2
2E-3	36	35	10.2	13	13	6.3
2E-4	39	38	10.4	13	13	6.2
2E-5	38	37	10.4	13	13	6.3
2E-6	34	34	10.5	13	13	6.3
2E-7	34	34	9.7	13	13	6.2
2E-8	38	38	10.6	13	13	6.1
2E-9	42	41	11.2	14	14	6.5
2E-10	47	44	12.1	18	17	7.0
2E-11	56	52	14.3	22	21	7.7
2E-12	69	62	15.8	28	26	8.8

in the FETI-DPH solver. The regularization parameter ϕ varies from 2×10^{-12} to 2×10^{-2} . The mesh size is 2^{-11} and the convergence threshold is 10^{-10} . The size of each subdomain is 2^{-5} and the number of processes is 48. The second and third columns show the number of FETI-DPH iterations without any augmentation, and the fourth column shows the corresponding FETI-DPH CPU time in seconds. The signs “-” and “+” indicate FETI-DPH solves on $\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$ and $\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$, respectively. The fifth and sixth columns show the number of FETI-DPH iterations with edge-based rigid body modes augmentation, and the seventh column shows the corresponding FETI-DPH CPU time in seconds. Table 6.1 shows that, in the absence of augmentation, the number of iterations tends to increase, but not strictly, as ϕ decreases. This dependence on ϕ is alleviated by the introduction of rigid body modes augmentation and some CPU time is saved. Table 6.1 demonstrates that rigid body modes augmentation works well for the thermal optimal control problem. The effects of rigid body modes augmentation is even more dramatic in the case of solid elements, as described in the following section. Further research on the existence of an optimal augmentation for the thermal optimal control problem is an interesting future topic (e.g., a set of modes that decreases the number of FETI-DPH iterations to an order of a constant regardless of the values of ϕ). Table 6.1 also shows that almost the same number of iterations of FETI-DPH is required on $\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$ and $\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$ for each value of ϕ . This result is not surprising, given that the two operators are complex conjugate to each other, and consequently have complex conjugate eigenvalues, the same pattern of clusterings, and an identical condition number.

There are two kinds of scalability tests for domain decomposition methods: numerical scalability and parallel scalability [13]. Numerical scalability is studied by varying the problem size, subdomain size, and the number of subdomains. Table 6.2 shows numerical scalability of FETI-DPH on the thermal problem (6.1). The number of iterations is shown for various mesh sizes h and for various subdomain sizes H . The regularization parameter ϕ is 2×10^{-8} . The convergence threshold for FETI-DPH is 10^{-10} . The fixed ratio $H/h = 2^6$ is used. Because the ratio H/h is fixed to be 2^6 , each subdomain contains 4096 elements. The problem size is varied from around 15 thousand degrees of freedom to 67 million degrees of freedom. Based on the theoretical result of (5.12), the number of FETI-DPH iterations must be more or less constant no matter what problem size is considered, provided that the ratio H/h is fixed. Indeed, Table 6.2 shows that it is the case. As h decreases, the total number of elements increases, meaning that the problem size increases. Table 6.2 shows that the number

of FETI-DPH iterations actually decreases as the problem size increases, which is consistent with (5.12). Again, almost the same number of iterations of FETI-DPH is required on $\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$ and $\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$ for each h .

TABLE 6.2

Numerical scalability of the FETI-DPH solver for a fixed ϕ value for the heat control problem.

num. of dofs	num. of elem.	h	H	nsub	$\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$	$\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$
15,876	16,384	1/128	1/2	4	26	25
64,516	65,536	1/256	1/4	16	25	25
260,100	262,144	1/512	1/8	64	21	20
1,044,484	1,048,576	1/1024	1/16	256	15	15
4,186,116	4,194,304	1/2048	1/32	1024	13	13
16,760,836	16,777,216	1/4096	1/64	4096	12	13
67,076,100	67,108,864	1/8192	1/128	16384	12	12

Parallel scalability measures how fast a domain decomposition method converges for a fixed problem size, a fixed subdomain size, and a fixed number of subdomains with increasing number of processes. Table 6.3 presents the parallel scalability of FETI-DPH for the heat conduction control problem (6.1). A mesh size $h = 1/1024$ (around 1 million degrees of freedom) and subdomain size $H = 1/16$ (4096 elements in each subdomain) are used. The regularization parameter ϕ is 2×10^{-8} . The convergence threshold is 10^{-10} . The number of processes (N_p) varies from 1 to 144 and parallel speed-up is measured relative to the FETI-DPH CPU time of $N_p = 1$. As the number of processes increases up to 144, Table 6.3 shows that a parallel speed-up of 26.8 is gained.

TABLE 6.3

Parallel scalability of the FETI-DPH solver for the heat control problem.

N_p	CPU time (sec)	Parallel speed-up
1	101.99	1
2	52.60	1.9
4	28.13	3.6
8	15.51	6.6
16	9.02	11.3
32	6.02	16.9
64	5.52	18.5
128	4.30	23.7
144	3.81	26.8

The thermal element (i.e., finite element for Laplacian operator) is widely used when one wants to verify a new numerical method or to analyze it. In the next section, a more difficult problem is considered (in the sense that properties of a real material are used and 3D solid elements are used).

6.2. Solid Cantilever Control. In this section, the proposed optimal control techniques are applied to a linear static structural problem, a cantilever with solid elements. In this case, the constraints in (6.1) are replaced by an elastostatic PDE. The cantilever has a square cross-section of 1×1 m² and a length of 3 m. The target displacement $\bar{\mathbf{y}}$ is generated by running a forward PDE simulation with a uniform pressure load of 100 kPa applied to the bottom surface of the cantilever. The fol-

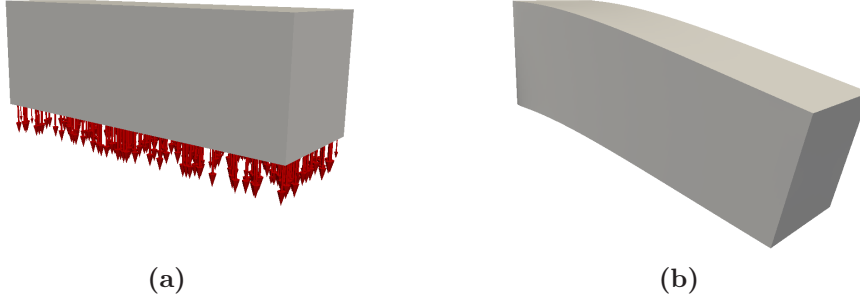


FIG. 6.4. (a) a uniform downward pressure of 100 kPa is applied to the undeformed configuration of cantilever. (b) the deformed configuration that is used as a target state in the optimal control problem.

lowing material properties are used: Young's modulus of 20.7 MPa, Poisson's ratio of 0.45, and density of 1.1 kg/m^3 . The applied force and the initial configuration of the cantilever are shown on Figure 6.4(a) and the corresponding target deformed configuration on Figure 6.4(b). Note that the left end is completely fixed. The body force is used as a control variable. Although this control variable is not practical because the body force control is not physically attainable, this problem is useful to investigate how the FETI-DPH solver performs on each factor of the Schur complement factorization for problems involving solid finite elements and linear elastostatic PDEs.

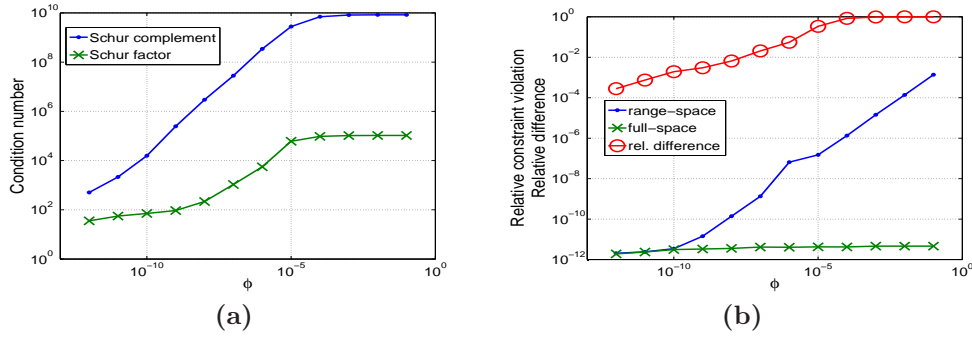


FIG. 6.5. (a) Condition number vs. ϕ (b) Relative constraint violation and relative difference between the target and solution displacements.

Figure 6.5 shows the accuracy issue of the range-space method that was explained in Section 3.1. The results were generated for a relatively small problem ($h = 1/8$ and 5184 degrees of freedom). Figure 6.5(a) shows how the condition numbers of both the negative Schur complement $\mathbf{S} = \mathbf{K}\mathbf{V}^{-1}\mathbf{K} + \frac{1}{\phi}\mathbf{V}$ and a complex factor $\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$ vary as ϕ varies. Both condition numbers increase as ϕ increases but the order of magnitude of the condition number of the factor is one half that of \mathbf{S} . Figure 6.5(b) demonstrates the accuracy dependency of both the range-space and the full-space methods on the value of ϕ by showing the relative constraint violation marked by a blue line with dots and a green line with x symbol (see the constraint in (2.1)). The

relative constraint violation is defined as

$$\frac{\|\mathbf{K}\mathbf{y} + \mathbf{K}_c\mathbf{y}_c - \mathbf{V}\mathbf{u}\|_2}{\|\mathbf{K}\mathbf{y}\|_2}. \quad (6.4)$$

As ϕ increases, the accuracy of the range-space method degenerates. For example, for values of ϕ larger than 10^{-2} , the constraint violation becomes larger than 10^{-4} , which is of questionable acceptability. On the other hand, the full-space method shows high accuracy consistently for the entire range of ϕ values considered. It is possible to achieve small improvements in the accuracy of the range-space method by using iterative refinement and higher floating-point precision. However, this approach incurs an additional cost and preliminary experiments suggest that it is not a competitive solution. One could argue that the values of interest of the parameter ϕ for this problem are the ones smaller than 10^{-5} because the relative difference between target and solution states is greater than 83% for values of ϕ outside this range. For this range of ϕ values, the accuracy of the range-space method is acceptable for this particular mesh size ($h = 1/8$).

TABLE 6.4
FETI-DPH's ϕ dependency for the structure control problem.

ϕ	-	+	CPU	RAUG-	RAUG+	CPU	A. speed-up
1E-7	327	344	909.9	30	33	187.4	4.9
1E-8	341	357	880.1	29	32	189.4	4.6
1E-9	286	300	777.4	29	31	192.4	4.0
1E-10	244	258	690.2	29	31	176.1	3.9
1E-11	228	239	673.9	29	30	185.3	3.6
1E-12	225	235	657.2	28	29	180.1	3.6
1E-13	220	229	652.3	27	28	193.6	3.4
1E-14	236	214	682.9	29	30	186.6	3.7
1E-15	262	260	773.2	34	34	173.5	4.5
1E-16	280	284	817.6	40	39	222.3	3.7
1E-17	315	316	976.3	46	46	214.1	4.6
1E-18	321	321	1009.6	52	52	236.5	4.3

Table 6.4 shows FETI-DPH's dependence on the value of ϕ for the cantilever control problem. The mesh size is $h = 1/90$ and the convergence threshold is 10^{-9} . The size of the subdomain is $H = 1/6$ and the number of processes is 64. The second and third columns show the number of FETI-DPH iterations without any augmentation, and the fourth column shows the corresponding FETI-DPH CPU time in seconds. The signs “-” and “+” indicate the FETI-DPH solves on $\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$ and $\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$, respectively. The fifth and sixth columns show the number of FETI-DPH iterations with edge-based rigid body modes augmentation, and the seventh column shows the corresponding FETI-DPH CPU time in seconds. The last column shows the speedup due to augmentation. Without augmentation, the performance of the FETI-DPH solver degrades as the value of ϕ increases from 10^{-12} to 10^{-7} . This makes sense because the condition number of the complex factors in (4.1) and the Schur complement itself increase as ϕ increases (see Figure 6.5). However, for the range of smaller ϕ values (i.e., less than 10^{-12}), the number of FETI-DPH iterations without augmentation increases as ϕ decreases. This is analogous to FETI-DPH being dependent on wave numbers in acoustic problems if no augmentation is applied (i.e., more iterations are required for a larger wave number without augmentation) although the complex factors are not the same as in the system that normally arises from a

Helmholtz problem. In order to alleviate this deterioration, edge-based rigid body mode augmentation is used. Table 6.4 shows that the augmentation decreases the number of FETI-DPH iterations substantially and reduces the CPU time by a factor of four on average. However, the rigid body modes augmentation is not optimal in the sense that as ϕ decreases the number of FETI-DPH iterations still increases.

Table 6.5 shows the numerical scalability of the FETI-DPH solver for the cantilever control problem. The number of iterations is shown for various mesh sizes h and for various subdomain sizes H . The regularization parameter ϕ is 1×10^{-16} . The convergence threshold for FETI-DPH is 10^{-9} . The problem size is varied from around 2 million degrees of freedom to 40 million degrees of freedom. As in the numerical scalability test for the thermal problem, the H/h ratio is fixed to 15 (exactly 3375 elements in each subdomain) and the number of FETI-DPH iterations is counted for both factors $\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$ and $\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$. As the problem size increases, the number of FETI iterations required for convergence slightly decreases, which is again consistent with the theoretical result (5.12).

TABLE 6.5

Numerical scalability of the FETI-DPH solver for a fixed ϕ value for the structure control problem.

num. of dofs	num. of elem.	h	H	nsub	$\mathbf{K} - \frac{1}{i\sqrt{\phi}}\mathbf{V}$	$\mathbf{K} + \frac{1}{i\sqrt{\phi}}\mathbf{V}$
1,976,400	648,000	1/60	1/4	192	43	43
3,847,500	1,265,625	1/75	1/5	375	41	40
6,633,900	2,187,000	1/90	1/6	648	39	38
10,517,850	3,472,875	1/105	1/7	1029	37	37
15,681,600	5,184,000	1/120	1/8	1536	36	36
22,307,400	7,381,125	1/135	1/9	2187	35	34
30,577,500	10,125,000	1/150	1/10	3000	33	33
40,674,150	13,476,375	1/165	1/11	3993	33	33

Table 6.6 shows the parallel scalability of FETI-DPH for the cantilever control problem. The FETI-DPH CPU time and the corresponding parallel speed-up are shown for the various numbers of processes. The regularization parameter ϕ is 10^{-12} . The convergence threshold is 10^{-9} . A mesh size $h = 1/90$ (i.e., around 6.6 million degrees of freedom) and a subdomain size $H = 1/6$ are used. The number of processes (N_p) increases from 1 to 64 and parallel speed-up is measured as the ratio with respect to the CPU time of $N_p = 1$. As the number of processes increases up to 64, Table 6.6 shows the speed up is greater than half the maximum possible.

TABLE 6.6

Parallel scalability of FETI-DPH solver for the structure control problem.

N_p	CPU time (sec)	Parallel speed-up
1	5854.9	1.0
2	2834.6	2.1
4	1547.0	3.8
8	886.6	6.6
16	586.4	10.0
32	317.4	18.4
64	163.5	35.8

Due to the introduction of complex numbers in the factorization, more storage

and more computational cost per solve are required compared to Pearson and Wathen's approximation to the Schur complement (\mathbf{S}_p in (3.4)). Thus, one must consider the effectiveness of the complex factorization carefully although only one solve with complex factorization is required in the range-space method and fewer iterations are required in the full-space method. Table 6.7 reports comparison of CPU time and number of iterations between \mathbf{S}_p and \mathbf{S}_c in the full-space method. The block diagonal preconditioner of Murphy, et al. (3.3) is used. Comparison is made for various values of ϕ and each solve is done by FETI-DPH with edge augmentation. The mesh size is $h = 1/30$ and the convergence threshold for FETI-DPH is 10^{-9} . GMRES is used for the main solver and the convergence threshold for GMRES is 10^{-10} . The size of each subdomain is $H = 1/2$ and 64 processes are used. The second column ($B_{\mathbf{S}_p}$) shows the CPU time for building an operator and the third column ($S_{\mathbf{S}_p}$) shows the CPU time for one solve with \mathbf{S}_p . The fourth column ($NI_{\mathbf{S}_p}$) shows the number of solves with \mathbf{S}_p that are required for convergence. The fifth column (TCPU) shows the total CPU time in seconds. The sixth to ninth columns show the corresponding results for \mathbf{S}_c . Note that the number of GMRES iterations required for convergence with \mathbf{S}_c is more than 3, which is not consistent with the spectral analysis done in [27]. Each solve time and building time is considerably higher for \mathbf{S}_c than \mathbf{S}_p , as expected. However, the number of GMRES iterations required for convergence is much higher for \mathbf{S}_p than \mathbf{S}_c . Additionally, the number of iterations for \mathbf{S}_p increases as ϕ decreases, while for \mathbf{S}_c it is bounded above. For this particular problem \mathbf{S}_p is a better choice for relatively large values of ϕ (i.e., $\phi = 10^{-7}$), while \mathbf{S}_c is a better choice for any value of ϕ smaller than 10^{-9} .

TABLE 6.7
Comparison between \mathbf{S}_p and \mathbf{S}_c in the full-space method.

ϕ	$B_{\mathbf{S}_p}$	$S_{\mathbf{S}_p}$	$NI_{\mathbf{S}_p}$	TCPU	$B_{\mathbf{S}_c}$	$S_{\mathbf{S}_c}$	$NI_{\mathbf{S}_c}$	TCPU
1E-7	2.1	2.0	43	88.1	9.2	3.9	25	106.5
1E-8	2.1	1.9	49	95.2	8.9	3.7	25	101.4
1E-9	2.1	2.0	55	112.1	9.6	4.1	25	112.1
1E-10	2.3	2.3	75	174.8	9.6	3.9	25	107.1
1E-11	2.4	2.4	77	187.2	9.3	4.1	25	111.8
1E-12	2.1	2.0	97	196.1	8.7	3.9	25	106.2
1E-13	2.2	2.0	97	196.2	9.2	4.6	25	124.2
1E-14	2.2	2.1	97	205.9	9.6	6.0	23	147.6
1E-15	2.2	2.1	110	233.2	8.8	5.6	23	137.6
1E-16	2.3	2.3	323	745.2	9.0	6.9	23	167.7

7. Conclusion. We have introduced a practical factorization of the Schur complement that arises from distributed optimal control of linear static systems. Due to the exact representation, if the range-space method is applicable, then one solve with the Schur complement is sufficient to obtain an optimal control solution. However, the Schur complement becomes ill-conditioned for large values of ϕ and as the mesh is refined. For example, the Schur complement that arises from a 3D cantilever problem with real material properties of rubber is prone to ill-conditioning when a relatively large regularization value is used. In such a case, an inaccurate solution is likely to be obtained if the range-space method is used, so solving a full KKT system simultaneously using a Krylov iterative method with a good preconditioner is recommended. The complex factorization of the Schur complement introduced in this paper can be used with a Schur-complement based preconditioner. The comparison between the approximate Schur complement of Pearson and Wathen and the complex factorization

as a preconditioner in the full-space method shows promising results for the complex factorization even in the context of the full-space method. The Schur complement solve is done with the parallel domain decomposition linear solver FETI-DPH. The scalability of FETI-DPH (both numerical and parallel) as well as its dependence on ϕ are studied in two academic problems: a thermal 2D problem and a structural 3D problem. Edge-based rigid body modes augmentation is able to bring the number of iterations down, but further research is necessary to find an optimal augmentation in the context of PDE-constrained distributed optimal control. The combination of exact representation of the Schur complement and good scalability of the FETI-DPH solver in addition to extensibility of the representation indicates promise for use of the complex factorization in more complicated and practical problems.

Acknowledgments. The authors thank Philip Avery in the Farhat Research Group for his valuable comments and essential help with coding the physics-based C++ PDE solver Aero-S.

REFERENCES

- [1] P. Avery and C. Farhat. The FETI family of domain decomposition methods for inequality-constrained quadratic programming: Application to contact problems with conforming and nonconforming interfaces. *Comput. Methods Appl. Mech. Engrg.*, 198:1673–1683, 2009.
- [2] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
- [3] L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders. *Large-scale PDE-Constrained Optimization: an introduction*. Springer, 2003.
- [4] L. T. Biegler and A. Wächter. SQP SAND strategies that link to existing modeling systems. In *Large-Scale PDE-Constrained Optimization*, pages 199–217. Springer, 2003.
- [5] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur Methods for PDE-Constrained Optimization. Part I: The Krylov-Schur Solver. *SIAM J. Sci. Comput.*, 27:687–713, 2005.
- [6] S. C. T. Choi. Minimal residual methods for complex symmetric, skew symmetric, and skew hermitian systems. *arXiv preprint arXiv:1304.6782*, 2013.
- [7] Y. Choi. *Simultaneous Analysis and Design in PDE-constrained Optimization*. PhD thesis, Stanford University, 2012.
- [8] D. Day and M. A. Heroux. Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, 23:480–498, 2001.
- [9] H. S. Dollar, N. I. M. Gould, M. Stoll, and A. J. Wathen. Preconditioning saddle-point systems with applications in optimization. *SIAM J. Sci. Comput.*, 32:249–270, 2010.
- [10] H. S. Dollar and A. J. Wathen. Approximate factorization constraint preconditioners for saddle-point matrices. *SIAM J. Sci. Comput.*, 27:1555–1572, 2006.
- [11] A. Draganescu and A. M. Soane. Multigrid solution of a distributed optimal control problem constrained by the Stokes equations. submitted, 2012.
- [12] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Oxford Science Publications, 2005.
- [13] C. Farhat, P. Avery, R. Tezaur, and J. Li. FETI-DPH: a dual-primal domain decomposition method for acoustic scattering. *Journal of Computational Acoustics*, 13:499–524, 2005.
- [14] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50:1523–1544, 2001.
- [15] C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.
- [16] A. Forsgren, P. E. Gill, and J. D. Griffin. Iterative solution of augmented systems arising in interior methods. *SIAM J. Optim.*, 18:666–690, 2007.
- [17] R. W. Freund. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Stat. Comput.*, 13:425–448, 1992.
- [18] R. W. Freund and N. M. Nachtigal. A new Krylov-subspace method for symmetric indefinite

- linear systems. In *Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics*, pages 1253–1256, 1994.
- [19] P. E. Gill, N. Gould, W. Murray, M. A. Saunders, and M. H. Wright. Range-space methods for convex quadratic programming. Technical report, Systems Optimization Laboratory, Stanford University, Stanford, CA, 1982.
 - [20] P. E. Gill, N. I. M. Gould, W. Murray, M. A. Saunders, and M. H. Wright. A weighted Gram-Schmidt method for convex quadratic programming. *Mathematical Programming*, 30(2):176–195, 1984.
 - [21] P. E. Gill and W. Murray. *Numerical methods for constrained optimization*, volume 1. Academic Press London, 1974.
 - [22] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM J. Matrix Anal. Appl.*, 13(1):292–311, 1992.
 - [23] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. on Matrix Anal. Appl.*, 21:1300–1317, 2000.
 - [24] D. Lahaye, H. De Gersem, S. Vandewalle, and K. Hameyer. Algebraic multigrid for complex symmetric systems. *IEEE Transactions on Magnetics*, 36:1535–1538, 2000.
 - [25] M. Lesoinne. 19. a feti-dp corner selection algorithm for three-dimensional problems. In *Domain Decomposition Methods in Science and Engineering*, Cocoyoc, Mexico, 2003. Conference Presentation.
 - [26] J. Mandel and R. Tezaur. Convergence of a substructuring method with lagrange multipliers. *Numerische Mathematik*, 73:473–487, 1996.
 - [27] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, 21(6):1969–1972, 2000.
 - [28] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–624, 1975.
 - [29] J. W. Pearson, M. Stoll, and A. J. Wathen. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, 33(4):1126–1152, 2012.
 - [30] J. W. Pearson and A. J. Wathen. A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. *Numerical Linear Algebra with Applications*, 19:816–829, 2012.
 - [31] E. Prudencio, R. Byrd, and X. C. Cai. Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems. *SIAM J. Sci. Comput.*, 27:13051328, 2006.
 - [32] T. Rees, H. S. Dollar, and A. J. Wathen. Optimal solvers for PDE-constrained optimization. *SIAM J. Sci. Comput.*, 32:271–298, 2010.
 - [33] S. Reitzinger, U. Schreiber, and U. Van Rienen. Algebraic multigrid for complex symmetric matrices and applications. *Journal of computational and applied mathematics*, 155(2):405–421, 2003.
 - [34] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7, 1986.
 - [35] V. Simoncini. Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems. *Computational Optimization and Applications*, 53(2):591–617, 2012.
 - [36] M. Stoll and A. Wathen. Combination preconditioning and the Bramble-Pasciak⁺ preconditioner. *SIAM Journal on Matrix Anal. Appl.*, 2011.
 - [37] H. S. Thorne. Properties of linear systems in PDE-constrained optimization. Part I: Distributed control. Technical report, Rutherford Appleton Laboratory, 2009.
 - [38] H. S. Thorne. Properties of linear systems in PDE-constrained optimization. Part II: Neumann boundary control. Technical report, Rutherford Appleton Laboratory, 2009.