CHAPTER 24

A Domain Decomposition Method for Eigenvalue Problems

Jenn-Ching Luo*

Abstract. With the advent of highspeed multiprocessor computer, the domain decomposition method has become perhaps the preferred technique for dealing with large scale problems and parallel computing. This work presents a domain decomposition method for numerical solutions of eigenvalue problems based upon two quotient iterations of the author. An innovative method for solving eigenvalue problems is developed, by which an eigenpair can be efficiently computed by solving a system of related subproblems. A mathematical foundation for manipulating a procedure of domain decomposition is also discussed.

Introduction. Many engineering and scientific problems are defined as eigenvalue problems. Solving an eigenproblem is one of the most time consuming procedures. The domain decomposition method provides an important idea to partition an entire problem into a system of related subproblems so that independent subproblems can be implemented in parallel. The importance of domain decomposition has been shown in many areas of applications, for example [1] which illustrates two types of decomposition, overlappped decomposition and nonoverlapped decomposition. In the overlapped decomposition, each pair of adjacent subdomains has a common region; while in nonoverlapped decomposition the intersection of adjacent subdomains is only the common edge. This study, based upon the successive projection approximations [2], falls into the scope of overlapped decomposition of which the Schwarz alternating method is a well known technique for solving elliptic equations. In the Schwarz alternating method, the governing equation in a subdomain is a restriction to the subdomain. Such a definition of governing equation limits the Schwarz alternating method to particular problems, because the restriction to a subdomain cannot generate a convergent procedure for a general type of partial differential equations. The domain decomposition method studies mathematical foundations for manipulating a procedure of domain decomposition and defining the subproblems. This work will use the successive projection approximations [2] to manipulate a procedure of domain decomposition and will use the author's quotient iterations ([3] and [4]) to define the subproblems. It should be emphasized that both

*Department of Civil Engineering and Engineering Mechanics, Columbia University, New York, NY 10027

the successive projection approximations and the author's quotient iterations can be applied to a decomposition with arbitrarily shaped subdomains.

The original version of successive projection approximations is derived from the concept of extractable projections, each of which maps a finite dimensional space onto a subspace. In the original version of successive projection approximation, a subproblem is defined in a subspace which can be termed space decomposition [6]. The successive projection approximation will be extended by the concept of subdomains so that a framework for manipulating a procedure of domain decomposition can be developed. The author's quotient iterations have been successfully formulated into four methods ([3], [4], and [5]) by the technique of space decomposition. These four methods have demonstrated to be robust techniques to compute real eigenpairs, and also have the potential to deal with parallel computations. An application of them to mode–shape analysis was reported by [7]. This work will formulate the quotient iterations into algebraic equations by the method of domain decomposition. A method for solving eigenproblem is developed, by which an eigenpair is computed by solving a system of related subproblems.

Domain Decomposition. In this section, a mathematical foundation for manipulating a procedure of domain decomposition in a finite dimensional space will be developed. Consider a domain Ω of a partial differential equation bounded by a closed boundary $\partial\Omega$. Ω is partitioned into a set of overlapped subdomains $\{\Omega^{(i)}\}_1^{N_p}$ where N_p is the number of subdomains. The subdomain $\Omega^{(i)}$ has a closed boundary $\partial\Omega^{(i)}$ which contains a new edge and probably a portion of the original boundary $\partial\Omega$. Denote by $\partial\Omega^{(i)}_{\text{new}}$ the new edge of $\Omega^{(i)}$ and denoted by $\partial\Omega^{(i)}_{\text{ori}}$ a portion of the original boundary $\partial\Omega$ along $\partial\Omega^{(i)}$. Then, $\partial\Omega^{(i)} = \partial\Omega^{(i)}_{\text{new}} + \partial\Omega^{(i)}_{\text{ori}}$. It is possible that $\partial\Omega^{(i)} = \partial\Omega^{(i)}_{\text{new}}$ if $\Omega^{(i)}$ does not attach to $\partial\Omega$. A subdomain may be a collection of the *supports* of some elements of basis if an appropriate finite dimensional space is provided. The support of an element of basis is a subdomain where non–zero interpolations are defined. For example,

$$N(X) = \begin{cases} \frac{X - X_{n-1}}{X_n - X_{n-1}}, & X_{n-1} \le X \le X_n \\ \frac{X - X_{n+1}}{X_n - X_{n+1}}, & X_n \le X \le X_{n+1} \\ 0, & \text{otherwise} \end{cases}$$

can be an element of basis (or called shape function) for one–dimensional piecewise linear approximation, whose support is the interval $[X_{n-1}, X_{n+1}]$ in which non–zero interpolations are defined. An interval is one kind of subdomain. In an N–dimensional space, grouping supports of elements of basis can generate a set of subdomains.

Generating finite elements in the domain of a partial differential equation including nodes and edges is the essential step in finite element analysis. There exist many ways to mesh a domain into finite elements. When dealing with the technique of domain decomposition, however, one condition that every subdomain is the collection of supports of some basis elements has to be satisfied. Therefore, there exists an additional requirement to generate meshes so that a set of appropriate finite elements can be obtained. For a given set of overlapped subdomains $\{\Omega^{(i)}\}_1^{N_p}$, there exists a set of new boundaries such as $\{\partial\Omega^{(i)}_{\text{new}}\}_1^{N_p}$. In order to meet the additional requirement, each new boundary has to place element nodes.

After obtaining a set of appropriate finite elements, i.e. a set of nodes and edges, a basis $\{B_j\}_1^N$ (so called shape functions) with proper order of interpolations can be written in accordance with the considered problem. This sets up an N–dimensional space Φ in which every $\phi \in \Phi$ can be written in the form

$$\phi = \sum_{j=1}^{N} \Lambda_j B_j \tag{1}$$

where Λ_j $(j = 1, 2, \ldots, N)$ are coefficients to be determined.

Once an N–dimensional space Φ spanned by basis $\{B_j\}_1^N$ has been constructed, the subdomain $\Omega^{(i)}$ can be generated as a collection of supports of some elements of basis, denoted by $b^{(i)}$ which is a subset of $\{B_j\}_1^N$. The subset $b^{(i)}$ spans a subspace of Φ. There exists an extractable projection $P^{(i)}$ which maps Φ onto the subspace spanned by $b^{(i)}$. A similar argument can be applied to define the extractable projections corresponding to the other subdomains. Denote by Υ_s an available fundamental subset [6], which is the collection of $\{P^{(i)}\}_1^{N_p}$. Υ_s has finite number of extractable projections. In order to derive an iterative procedure by using the finite set Υ_s, let n be a subscript such that P_n is the n–th employed projection of an iterative procedure. Therefore, $P_n \equiv P^{(i)}$ for some i where the superscript (i) indicates the i–th extractable projection in Υ_s. For a $\phi \in \Phi$ in the form of equation (1), with $P_n \equiv P^{(i)}$, the projection $P_n\phi$ is written as:

$$P_n\phi = \sum_{\substack{j=1 \\ B_j \in b^{(i)}}}^{N} \Lambda_j B_j. \tag{2}$$

By successive projection approximation, a $\phi \in \Phi$ can be approximated by a sequence $\{\overline{\phi}_n\}$ where $\overline{\phi}_n$ is written as [6]:

$$\overline{\phi}_n = (I - P_n)\,\overline{\phi}_{n-1} + P_n\eta_n, \tag{3}$$

in which $P_n\eta_n$ is the solution to a *guide* denoted by $\mathcal{G}(\overline{\phi}_n) = \mathcal{G}(P_n\eta_n) = 0$. Once the solution to $\mathcal{G}(P_n\eta_n) = 0$ has been obtained, a new approximation $\overline{\phi}_n$ can be updated by equation (3) where $\overline{\phi}_{n-1}$ is the previous approximation. An admissible trial $\overline{\phi}_0$ satisfying the boundary condition is necessary (please see [6]).

A coefficient with respect to a basis element is called a degree of freedom, which must be determined such that a governing equation subject to the boundary conditions can be satisfied. Each element, in an N–dimensional space Φ spanned by $\{B_j\}_1^N$, has N degrees of freedom. All the N degrees of freedom are in the domain Ω, because an admissible trial satisfies the boundary conditions. However, for the subdomain $\Omega^{(i)}$, both $\Omega^{(i)}$ and $\partial\Omega^{(i)}_{\text{new}}$ have degrees of freedom. The degrees of freedom with respect to $\Omega^{(i)}$ and $\partial\Omega^{(i)}_{\text{new}}$ have to be discussed separately, because the subspace spanned by $b^{(i)}$ contains only the degrees of freedom in $\Omega^{(i)}$ i.e. the degrees of freedom on $\partial\Omega^{(i)}_{\text{new}}$ cannot be updated by equation (3) when $P_n \equiv P^{(i)}$.

Denote by $\{A_n^{(i)}\}$ all the degrees of freedom associated with $\Omega^{(i)}$ and $\partial\Omega^{(i)}_{\text{new}}$ where the subscript n indicates the iterative step. Denote by $N_{\Omega^{(i)}}$ the number

of degrees of freedom in $\Omega^{(i)}$, and denoted by $N_{\partial\Omega^{(i)}_{\text{new}}}$ the number of degrees of freedom on $\partial\Omega^{(i)}_{\text{new}}$. $\{A^{(i)}_n\}$ has $(N_{\Omega^{(i)}}+N_{\partial\Omega^{(i)}_{\text{new}}})$ degrees of freedom. There are two separable sets of coefficient, which can be denoted by $[\widehat{Z}^{(i)}]\{A^{(i)}_n\}$ and $[Z^{(i)}]\{A^{(i)}_n\}$ with respect to $\partial\Omega^{(i)}_{\text{new}}$ and $\Omega^{(i)}$, where $[\widehat{Z}^{(i)}]$ and $[Z^{(i)}]$ are Boolean matrices of order $(N_{\partial\Omega^{(i)}_{\text{new}}} \times N)$ and $(N_{\Omega^{(i)}} \times N)$, respectively. There exist many possible ways to order the entries of $\{A^{(i)}_n\}$. For the purpose of convenience, let $\left[[\widehat{Z}^{(i)}]^T[Z^{(i)}]^T\right]$ be an identity matrix, i.e. the entries of $\{A^{(i)}_n\}$ begin with the degrees of freedom on $\partial\Omega^{(i)}_{\text{new}}$, and then are followed by the degrees of freedom in $\Omega^{(i)}$.

For every degree of freedom in $\Omega^{(i)}$, there exists an element in $b^{(i)}$, called shape functions, which defines an approximation associated with the degree of freedom. For every degree of freedom on $\partial\Omega^{(i)}_{\text{new}}$, however, there does not exist an element in $b^{(i)}$ to support the degree of freedom. The associated shape functions for the degrees of freedom on $\partial\Omega^{(i)}_{\text{new}}$ are defined by elements of basis of the subdomains adjacent to $\Omega^{(i)}$. Denote by $\{N^{(i)}\}$ the collection of shape functions associated with the degrees of freedom on $\partial\Omega^{(i)}_{\text{new}}$ and in $\Omega^{(i)}$, with the same ordering of entries as $\{A^{(i)}_n\}$ such that $\left[[\widehat{Z}^{(i)}]^T[Z^{(i)}]^T\right]$ is an identity matrix. If the subdomain $\Omega^{(i)}$ is viewed as a substructure, then the vector $\{N^{(i)}\}$ is the whole shape functions, except the ones associated with the degrees of freedom on $\partial\Omega^{(i)}_{\text{ori}}$. An approximation associated with the degrees of freedom in $\Omega^{(i)}$, i.e. in the subspace spanned by $b^{(i)}$, can be written as:

$$\overline{\phi}_n(\Omega^{(i)}) = \{N^{(i)}\}^T[Z^{(i)}]^T[Z^{(i)}]\{A^{(i)}_n\}. \tag{4}$$

Similarly, an approximation associated with the degrees of freedom on $\partial\Omega^{(i)}_{\text{new}}$ can be written as:

$$\overline{\phi}_n(\partial\Omega^{(i)}_{\text{new}}) = \{N^{(i)}\}^T[\widehat{Z}^{(i)}]^T[\widehat{Z}^{(i)}]\{A^{(i)}_n\}. \tag{5}$$

Using equation (3), with $P_nP_n = P_n$, we can obtain

$$P_n\overline{\phi}_n = P_n\eta_n. \tag{6}$$

Since $P_n \equiv P^{(i)}$ maps an N–dimensional space onto the subspace spanned by $b^{(i)}$, $P_n\overline{\phi}_n$ is defined by the degrees of freedom in $\Omega^{(i)}$ so that $P_n\overline{\phi}_n$ can be written as $P_n\overline{\phi}_n = \overline{\phi}_n(\Omega^{(i)})$. With equation (6),

$$P_n\eta_n = \overline{\phi}_n(\Omega^{(i)}). \tag{7}$$

Since the projection of P_n is defined by the degrees of freedom in $\Omega^{(i)}$, the projection of $(I - P_n)$ will be defined by the degrees of freedom in $(\Omega - \Omega^{(i)})$ and on $\partial\Omega^{(i)}_{\text{new}}$ so that

$$(I - P_n)\overline{\phi}_{n-1} = \overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}}) + \overline{\phi}_{n-1}(\Omega - \Omega^{(i)}). \tag{8}$$

Substitution of equations (7) and (8) into (3), the successive projection approximation [2] becomes

$$\overline{\phi}_n = \overline{\phi}_n(\Omega^{(i)}) + \overline{\phi}_{n-1}(\Omega - \Omega^{(i)}) + \overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}}). \tag{9}$$

Equation (9) updates the approximation in the subdomain $\Omega^{(i)}$. A similar result can be obtained from every subdomain. Equation (9) is a mathematical foundation to deal with an approximation of subdomains. The notations that $\mathcal{G}(\overline{\phi}_n) = \mathcal{G}(P_n\eta_n) = \mathcal{G}(\overline{\phi}_n(\Omega^{(i)})) = 0$ with $P_n \equiv P^{(i)}$ can be easily verified. A mathematical foundation for manipulating a procedure of domain decomposition can be summarized below.

For a given partial differential equation defined in a domain Ω partitioned into a set of overlapped subdomains $\{\Omega^{(i)}\}_1^{N_p}$, with an admissible $\overline{\phi}_o$ satisfying the boundary conditions, a domain decomposition procedure can be achieved by iterating the following equations:

For $n = 1, 2, \ldots,$ until convergence

$$\text{Solve } \mathcal{G}(\overline{\phi}_n(\Omega^{(i)})) = 0 \tag{10}$$

$$\text{Update } \overline{\phi}_n = \overline{\phi}_n(\Omega^{(i)}) + \overline{\phi}_{n-1}(\Omega - \Omega^{(i)}) + \overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}}) \tag{11}$$

where the superscript (i) indicates the subdomain considered at step n. Each iteration step has a subproblem in a subdomain solved.

Luo's Quotient Iterations. As introduced previously, equations (10) and (11) are the fundamental equations for manipulating a procedure of domain decomposition. An approximation of subdomains has been shown in the form of (11). The underlying goal is to find a guide to make a convergent decomposition. Two guides for decomposing an eigenproblem into a system of related subproblems have been studied by Luo ([3] and [4]). Let

$$\mathcal{L}\phi + \lambda f\phi = 0 \quad \text{in } \Omega \qquad \text{subject to} \qquad \mathcal{M}\phi = 0 \quad \text{on } \partial\Omega \tag{12}$$

be an eigenproblem where $\mathcal{L}$ and $\mathcal{M}$ are differential operators, f is a given function independent of ϕ. The first guide for partitioning the problem in the form of equation (12) is written as:

$$\mathcal{G}\left(\overline{\phi}_n(\Omega^{(i)})\right) = \delta(\mathcal{L}\overline{\phi}_n, \mathcal{L}\overline{\phi}_n) + C^+_{n-1} \cdot \delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n) = 0 \tag{13}$$

$$C^+_{n-1} = -\frac{(\mathcal{L}\overline{\phi}_{n-1}, \mathcal{L}\overline{\phi}_{n-1})}{(\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})} \tag{14}$$

with a sufficient condition for convergence

$$\zeta^+_n(\overline{\phi}_n) \cdot \delta^2\zeta^+_n(\overline{\phi}_n) \geq 0 \tag{15}$$

where $\zeta^+_n(\overline{\phi}_n) = (\mathcal{L}\overline{\phi}_n, \mathcal{L}\overline{\phi}_n) + C^+_{n-1} \cdot (\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n)$, $(\bullet, \bullet)$ indicates an inner product, and the variation is taken with respect to the approximation $\overline{\phi}_n$. A guide contains the degrees of freedom in the subdomain $\Omega^{(i)}$. This shows the nature of domain decomposition. The sufficient condition for convergence in the form of equation (15) becomes a criteria to define a convergent region. When $\{\overline{\phi}_n\}$ approaches ϕ, $\zeta^+_n(\phi) = 0$ and equation (15) is satisfied. This implies that a convergent region contains at least one element in a finite dimensional space. Obviously, Some solutions may have large regions of convergence; while others may have small ones. Repeatedly solving

equations (13) and (14) for $n = 1, 2, \ldots$, yields a sequence of $\{\overline{\phi}_n\}$ and a sequence of $\{C^+_{n-1}\}$. It has been proved [4] that the limit of $\overline{\phi}_n$ is an eigenfunction and the limit of C^+_{n-1} is a corresponding eigenvalue. Substituting equation (14) into (13) yields

$$\frac{\delta(\mathcal{L}\overline{\phi}_n, \mathcal{L}\overline{\phi}_n)}{\delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n)} = \frac{(\mathcal{L}\overline{\phi}_{n-1}, \mathcal{L}\overline{\phi}_{n-1})}{(\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})} \tag{16}$$

which is linear in $\overline{\phi}_n(\Omega^{(i)})$. The procedure to repeat the equation (16) leads to the author's *first quotient iteration*:

(a) Select an admissible trial $\overline{\phi}_0$.

(b) Repeat the following procedure for solving $\overline{\phi}_n$ where $n = 1, 2, \ldots$, until convergence

$$\frac{\delta(\mathcal{L}\overline{\phi}_n, \mathcal{L}\overline{\phi}_n)}{\delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n)} = \frac{(\mathcal{L}\overline{\phi}_{n-1}, \mathcal{L}\overline{\phi}_{n-1})}{(\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})}. \tag{17}$$

The negative value of the limit of quotient is an eigenvalue of equation (12) [Note if equation (12) is written into the form $\mathcal{L}\phi = \lambda f\phi$ then the limit of quotient is an eigenvalue]. An admissible trial has to satisfy the following requirements: (a) $\overline{\phi}_0 \neq 0$; (b) $\overline{\phi}_0$ satisfies the boundary condition; (c) $\overline{\phi}_0 = \sum_{j=1}^{N} \alpha_j B_j$ where $\alpha_j \neq 0$ for all j; (d) $\overline{\phi}_0$ is defined in a convergent region. This study will use such a quotient iteration to define the subproblem in a subdomain. Once an admissible trial has been selected, an eigenpair can be computed by the first quotient iteration. The second guide for decomposing an eigenproblem is written as:

$$\mathcal{G}(P_n\eta_n) = \delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n) + C^-_{n-1} \cdot \delta(f\overline{\phi}_n, f\overline{\phi}_n) = 0 \tag{18}$$

$$C^-_{n-1} = -\frac{(\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})}{(f\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})}. \tag{19}$$

Substituting equation (19) into (18) yields

$$\frac{\delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n)}{\delta(f\overline{\phi}_n, f\overline{\phi}_n)} = \frac{(\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})}{(f\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})} \tag{20}$$

which leads to the author's *second quotient iteration*:

(a) Select an admissible trial $\overline{\phi}_0$.

(b) Repeat the following procedure for solving $\overline{\phi}_n$ where $n = 1, 2, \ldots$, until convergence:

$$\frac{\delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n)}{\delta(f\overline{\phi}_n, f\overline{\phi}_n)} = \frac{(\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})}{(f\overline{\phi}_{n-1}, f\overline{\phi}_{n-1})}. \tag{21}$$

Both the first and second quotient iterations can be used to solve an eigenvalue problem in partial differential equations. For the purpose of computing, the second quotient is better than the first quotient. Because for some operator $\mathcal{L}$ and function f, the required order of elements of basis for formulating a guide can be reduced. For example, if $\mathcal{L} = \mathcal{T} \cdot \mathcal{T}$ where $\mathcal{T}$ is self-adjoint and f is a real constant, then $(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n) = f(\mathcal{T}\overline{\phi}_n, \mathcal{T}\overline{\phi}_n)$ in which the required order of elements of basis $\{B_j\}_1^N$ to define $\overline{\phi}_n$ can be reduced so as to simplify the computations. Luo [4] showed that the absolute value of the first quotient is always greater than the one of the second

quotient until convergence. The difference of the two quotients can be used as a criterion to examine accuracy and terminate a computing procedure. This paper will use the quotient iterations to define subproblems in a decomposition procedure.

Finite Element Formulations. First, let the second quotient iteration be formulated into a finite element equation. It can be easily verified that

$$\overline{\phi}_{n-1} = \overline{\phi}_{n-1}(\Omega^{(i)}) + \overline{\phi}_{n-1}(\Omega - \Omega^{(i)}) + \overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}}). \tag{22}$$

Adding $\overline{\phi}_{n-1}(\Omega^{(i)}) - \overline{\phi}_{n-1}(\Omega^{(i)})$ to the right side of equation (11), with equation (22), yields

$$\overline{\phi}_n = \overline{\phi}_{n-1} + \overline{\phi}_n(\Omega^{(i)}) - \overline{\phi}_{n-1}(\Omega^{(i)}). \tag{23}$$

Using equations (22) and (23), we can obtain that

$$\begin{aligned}(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n) = (\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1}) &+ \left(\mathcal{L}\overline{\phi}_n(\Omega^{(i)}), f\overline{\phi}_n(\Omega^{(i)})\right) - \\ &\left(\mathcal{L}\overline{\phi}_{n-1}(\Omega^{(i)}), f\overline{\phi}_{n-1}(\Omega^{(i)})\right) + \\ &\left(\mathcal{L}\overline{\phi}_n(\Omega^{(i)}) - \mathcal{L}\overline{\phi}_{n-1}(\Omega^{(i)}), f\overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}})\right) + \\ &\left(\mathcal{L}\overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}}), f\overline{\phi}_n(\Omega^{(i)}) - f\overline{\phi}_{n-1}(\Omega^{(i)})\right).\end{aligned} \tag{24}$$

Using equations (4) and (5), we can obtain

$$\left(\mathcal{L}\overline{\phi}_n(\Omega^{(i)}), f\overline{\phi}_n(\Omega^{(i)})\right) - \left(\mathcal{L}\overline{\phi}_{n-1}(\Omega^{(i)}), f\overline{\phi}_{n-1}(\Omega^{(i)})\right) = \sigma^{(i)}_n \tag{25}$$

$$\begin{aligned}&\left(\mathcal{L}\overline{\phi}_n(\Omega^{(i)}) - \mathcal{L}\overline{\phi}_{n-1}(\Omega^{(i)}), f\overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}})\right) + \\ &\quad\left(\mathcal{L}\overline{\phi}_{n-1}(\partial\Omega^{(i)}_{\text{new}}), f\overline{\phi}_n(\Omega^{(i)}) - f\overline{\phi}_{n-1}(\Omega^{(i)})\right) = \tau^{(i)}_n\end{aligned} \tag{26}$$

where

$$\sigma^{(i)}_n = \{\Delta A^{(i)}_n\}^T [Z^{(i)}]^T [K^{(i)}][Z^{(i)}]\{2A^{(i)}_n - \Delta A^{(i)}_n\} \tag{27}$$

$$\tau^{(i)}_n = 2\{\Delta A^{(i)}_n\}^T [Z^{(i)}]^T [\widehat{K}^{(i)}][\widehat{Z}^{(i)}]\{A^{(i)}_{n-1}\} \tag{28}$$

$$\{\Delta A^{(i)}_n\} = \{A^{(i)}_n\} - \{A^{(i)}_{n-1}\} \tag{29}$$

$$[K^{(i)}] = [Z^{(i)}][S^{(i)}][Z^{(i)}]^T \equiv \text{the i–th subdomain stiffness matrix} \tag{30}$$

$$[\widehat{K}^{(i)}] = [Z^{(i)}][S^{(i)}][\widehat{Z}^{(i)}]^T \equiv \text{the i–th carryover stiffness matrix} \tag{31}$$

$$[S^{(i)}] = \frac{1}{2}\left(\mathcal{L}\{N^{(i)}\}, f\{N^{(i)}\}^T\right) + \frac{1}{2}\left(f\{N^{(i)}\}, \mathcal{L}\{N^{(i)}\}^T\right). \tag{32}$$

Substituting equations (25) and (26) into equation (24) yields

$$(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n) = (\mathcal{L}\overline{\phi}_{n-1}, f\overline{\phi}_{n-1}) + \sigma^{(i)}_n + \tau^{(i)}_n. \tag{33}$$

Using a similar procedure to derive equation (33), we can obtain

$$(f\overline{\phi}_n, f\overline{\phi}_n) = (f\overline{\phi}_{n-1}, f\overline{\phi}_{n-1}) + \zeta_n^{(i)} + \eta_n^{(i)} \tag{34}$$

where

$$\zeta_n^{(i)} = \{\Delta A_n^{(i)}\}^T [Z^{(i)}]^T [M^{(i)}][Z^{(i)}]\{2A_n^{(i)} - \Delta A_n^{(i)}\} \tag{35}$$

$$\eta_n^{(i)} = 2\{\Delta A_n^{(i)}\}^T [Z^{(i)}]^T [\widehat{M}^{(i)}][\widehat{Z}^{(i)}]\{A_{n-1}^{(i)}\} \tag{36}$$

$$[M^{(i)}] = [Z^{(i)}][W^{(i)}][Z^{(i)}]^T \equiv \text{ the i–th subdomain mass matrix} \tag{37}$$

$$[\widehat{M}^{(i)}] = [Z^{(i)}][W^{(i)}][\widehat{Z}^{(i)}]^T \equiv \text{ the i–th carryover mass matrix} \tag{38}$$

$$[W^{(i)}] = (f\{N^{(i)}\}, f\{N^{(i)}\}^T). \tag{39}$$

Taking variation of equations (33) and (34) with respect to $\overline{\phi}_n(\Omega^{(i)})$, with equations (27), (28), (29), (35), and (36), yields

$$\begin{aligned}\delta(\mathcal{L}\overline{\phi}_n, f\overline{\phi}_n) =& \delta\sigma_n^{(i)} + \delta\tau_n^{(i)} \\ =& 2\cdot\delta\left(\{A_n^{(i)}\}^T [Z^{(i)}]^T\right)[K^{(i)}][Z^{(i)}]\{A_n^{(i)}\} + \\ & 2\cdot\delta\left(\{A_n^{(i)}\}^T [Z^{(i)}]^T\right)[\widehat{K}^{(i)}][\widehat{Z}^{(i)}]\{A_{n-1}^{(i)}\}\end{aligned} \tag{40}$$

$$\begin{aligned}\delta(f\overline{\phi}_n, f\overline{\phi}_n) =& \delta\zeta_n^{(i)} + \delta\eta_n^{(i)} \\ =& 2\cdot\delta\left(\{A_n^{(i)}\}^T [Z^{(i)}]^T\right)[M^{(i)}][Z^{(i)}]\{A_n^{(i)}\} + \\ & 2\cdot\delta\left(\{A_n^{(i)}\}^T [Z^{(i)}]^T\right)[\widehat{M}^{(i)}][\widehat{Z}^{(i)}]\{A_{n-1}^{(i)}\}.\end{aligned} \tag{41}$$

Substituting equations (19), (40), and (41) into (21), since $\delta\left(\{A_n^{(i)}\}^T [Z^{(i)}]^T\right)$ is arbitrary, yields

$$\begin{aligned}&[[K^{(i)}] + C_{n-1}^{-}[M^{(i)}]][Z^{(i)}]\{A_n^{(i)}\} + \\ &\quad [[\widehat{K}^{(i)}] + C_{n-1}^{-}[\widehat{M}^{(i)}]][\widehat{Z}^{(i)}]\{A_{n-1}^{(i)}\} = 0.\end{aligned} \tag{42}$$

If $[[K^{(i)}] + C_{n-1}^{-}[M^{(i)}]]^{-1}$ exists, equation (42) becomes

$$\begin{aligned}[Z^{(i)}]\{A_n^{(i)}\} = &-[[K^{(i)}] + C_{n-1}^{-}[M^{(i)}]]^{-1} \\ &[[\widehat{K}^{(i)}] + C_{n-1}^{-}[\widehat{M}^{(i)}]][\widehat{Z}^{(i)}]\{A_{n-1}^{(i)}\}\end{aligned} \tag{43}$$

which defines the subproblem in subdomain $\Omega^{(i)}$. Each subdomain has a governing equation in the form of equation (43) which allows it to be a ***homogenous decomposition.*** Once $[Z^{(i)}]\{A_n^{(i)}\}$ has been obtained from equation (43), $\overline{\phi}_n(\Omega^{(i)})$ can be defined by equation (4) and $\overline{\phi}_n$ can be updated by equation (11). This procedure will be repeated for any n until convergence. For the situation where $[[K^{(i)}] + C_{n-1}^{-}[M^{(i)}]]^{-1}$ does not exist, the iterative procedure just skips to solve another subproblem because there is no particular order for selecting an extractable projection P_n employed in equation (21), i.e. if an extractable projection in Υ_s

causes a singular subproblem, we can use another projection in Υ_s to define another subproblem.

The first quotient iteration in the form of (17) also can be formulated into a finite element equation for defining a subproblem which will lead to a result similar to the second quotient iteration except for the definitions of C_{n-1}^{+}, stiffness and mass matrices which may be written as:

$$[S^{(i)}] = (\mathcal{L}\{N^{(i)}\}, \mathcal{L}\{N^{(i)}\}^T) \tag{44}$$

$$[W^{(i)}] = \frac{1}{2}(\mathcal{L}\{N^{(i)}\}, f\{N^{(i)}\}^T) + \frac{1}{2}(f\{N^{(i)}\}, \mathcal{L}\{N^{(i)}\}^T). \tag{45}$$

The following discussions will focus on the applications of the second quotient iteration.

Computational Considerations. A domain decomposition procedure solves a group of subproblems, each of which has an individual subdomain with a set of degrees of freedom and data. It may be a burden for an analyst to arrange a procedure to solve a group of subproblems. In order to avoid such burden and for the purpose of convenience, an appropriate arrangement to a computational procedure is necessary so that it would be better to label each computational step physically, i.e. give each computational step a physical name. Physical labels can help analysts in arranging computational procedures. This section will label the computational steps physically and simplify the mathematical notations.

Let us rearrange equation (43) into the system

$$[Z^{(i)}]\{A_n^{(i)}\} = -\left[[K^{(i)}] + C_{n-1}^{-}[M^{(i)}]\right]^{-1}\{r_{n-1}^{(i)}\} \tag{46}$$

$$\{r_{n-1}^{(i)}\} = \left[[\widehat{K}^{(i)}] + C_{n-1}^{-}[\widehat{M}^{(i)}]\right][\widehat{Z}^{(i)}]\{A_{n-1}^{(i)}\} \tag{47}$$

where the right sides of equations (46) and (47) are all known values at step n. Equation (46) can be rewritten in the form

$$[Z^{(i)}]\{\Delta A_n^{(i)}\} = -\left[[K^{(i)}] + C_{n-1}^{-}[M^{(i)}]\right]^{-1}\{r_{n-1}^{(i)}\} - [Z^{(i)}]\{A_{n-1}^{(i)}\}. \tag{48}$$

A notation convention is made so that all the equations are computed from their right sides and result in the left side vectors. The subscript n and $n-1$ can be neglected if an equation is rewritten so as to meet the notation convention. Furthermore, let

$$\begin{aligned}
\{\widehat{\Lambda}^{(i)}\} &= [\widehat{Z}^{(i)}]\{A^{(i)}\} \equiv \text{ a carryover} \\
\{\Delta\Lambda^{(i)}\} &= [Z^{(i)}]\{\Delta A^{(i)}\} \equiv \text{ an increment} \\
\{\Lambda^{(i)}\} &= [Z^{(i)}]\{A^{(i)}\} \equiv \text{ an approximation to eigenfunction} \\
\lambda &= C^{-} \equiv \text{ an approximation to eigenvalue.}
\end{aligned} \tag{49}$$

Using the notation convention, with equation (49), equations (47) and (48) become

$$\{r^{(i)}\} \leftarrow \left[[\widehat{K}^{(i)}] + \lambda[\widehat{M}^{(i)}]\right]\{\widehat{\Lambda}^{(i)}\} \tag{50}$$

$$\{\Delta\Lambda^{(i)}\} \leftarrow -\left[[K^{(i)}] + \lambda[M^{(i)}]\right]^{-1}\{r^{(i)}\} - \{\Lambda^{(i)}\} \tag{51}$$

respectively. Equation (50) is termed *receiving procedure* which receives a carryover $\{\widehat{\Lambda}^{(i)}\}$ from its cooperating subproblems. Equation (51) is termed *solving procedure* which computes an increment $\{\Delta\Lambda^{(i)}\}$ if $\{r^{(i)}\}$ provided. Once a $\{\Delta\Lambda^{(i)}\}$ has been obtained, an approximation $\{\Lambda^{(i)}\}$ can be updated by equation (29) which can be simplified into the form

$$\{\Lambda^{(i)}\} \leftarrow \{\Lambda^{(i)}\} + \{\Delta\Lambda^{(i)}\} \tag{52}$$

Equation (52) is a step of *updating procedure*. An updating procedure also contains the step to update λ defined by equation (19). The procedure to update an eigenvalue is more complex than the one to update an eigenfunction. Using equations (27), (28), (33), (34), (35), and (36), a procedure to update an eigenvalue contains the following seven equations:

$$\begin{aligned}
\sigma^{(i)} &\leftarrow \{\Delta\Lambda^{(i)}\}^T[K^{(i)}]\{2\Lambda^{(i)} - \Delta\Lambda^{(i)}\} \\
\zeta^{(i)} &\leftarrow \{\Delta\Lambda^{(i)}\}^T[M^{(i)}]\{2\Lambda^{(i)} - \Delta\Lambda^{(i)}\} \\
\tau^{(i)} &\leftarrow 2 \cdot \{\Delta\Lambda^{(i)}\}^T[\widehat{K}^{(i)}]\{\widehat{\Lambda}^{(i)}\} \\
\eta^{(i)} &\leftarrow 2 \cdot \{\Delta\Lambda^{(i)}\}^T[\widehat{M}^{(i)}]\{\widehat{\Lambda}^{(i)}\} \\
(\mathcal{L}\overline{\phi}, f\overline{\phi}) &\leftarrow (\mathcal{L}\overline{\phi}, f\overline{\phi}) + \sigma^{(i)} + \tau^{(i)} \\
(f\overline{\phi}, f\overline{\phi}) &\leftarrow (f\overline{\phi}, f\overline{\phi}) + \zeta^{(i)} + \eta^{(i)} \\
\lambda &\leftarrow -\frac{(\mathcal{L}\overline{\phi}, f\overline{\phi})}{(f\overline{\phi}, f\overline{\phi})}
\end{aligned} \tag{53}$$

with a set of previous $(\mathcal{L}\overline{\phi}, f\overline{\phi})$ and $(f\overline{\phi}, f\overline{\phi})$. The updating procedure begins with equation (52) and is followed by equation (53). The receiving–solving–updating procedure is considered under the situation a $\{\widehat{\Lambda}^{(i)}\}$ has been received, i.e. the procedure is activated by its cooperating subproblems. Similarly, the subproblem in $\Omega^{(i)}$ also can activate its cooperating subproblems by sending them corresponding carryovers. Denote by $[C_{ki}]$ a Boolean matrix which transfers an approximation $\{\Lambda^{(i)}\}$ to a carryover of the subproblem in subdomain $\Omega^{(k)}$. Then a carryover to the cooperating subproblem in $\Omega^{(k)}$ can be written as

$$\{\widehat{\Lambda}^{(k)}\} \leftarrow [C_{ki}]\{\Lambda^{(i)}\} \tag{54}$$

which is termed *sending procedure*. In an asynchronous system, only the new carryover $\{\widehat{\Lambda}^{(k)}\}$ is valid, i.e. an old $\{\widehat{\Lambda}^{(k)}\}$ has to be discarded no matter it has been proceeded or not when a new $\{\widehat{\Lambda}^{(k)}\}$ has arrived. Subdomain $\Omega^{(i)}$ may have more than one adjacent subdomain. For every $\{\Lambda^{(i)}\}$, each cooperating subproblem should be sent one set of corresponding carryovers. Since $[C_{ki}]$ is a Boolean matrix, some advantage for computing the right side of equation (54) can be taken of. A realistic computing procedure does not really generate a Boolean matrix. An alternative to (54) can be made for example in a Fortran code

```
DO S=1,U
    T=MAP(S)
    Λ̂_S^(k) = Λ_T^(i)
```

END DO

where U is the number of degrees of freedom transferred from the i–th subdomain to the k–th subdomain, $\widehat{\Lambda}_S^{(k)}$ is the S–th entry in $\{\widehat{\Lambda}^{(k)}\}$, $\Lambda_T^{(i)}$ is the T–th entry in $\{\Lambda^{(i)}\}$, and MAP is a map for transferring degrees of freedom. For example, a Boolean matrix $[C_{ki}]$ is written as:

$$[C_{ki}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

then U= 3, MAP(1) = 1, MAP(2) = 3, and MAP(3) = 4.

With these physical labels, a computing procedure in a subdomain is just a repeated application of receiving–solving–updating–sending procedure until an eigenpair converges. The initial procedure needs to select an admissible trial and compute $(\mathcal{L}\overline{\phi}, f\overline{\phi})$, $(f\overline{\phi}, f\overline{\phi})$ and λ. Using equation (22), we can obtain the result with respect to subdomain $\Omega^{(i)}$

$$\begin{aligned}(\mathcal{L}\overline{\phi}, f\overline{\phi}) = &\left(\mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi}(\Omega^{(i)})\right) \\ &+ \left(\mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi}(\partial\Omega^{(i)}_{\text{new}})\right) + \left(f\overline{\phi}(\Omega^{(i)}), \mathcal{L}\overline{\phi}(\partial\Omega^{(i)}_{\text{new}})\right) \\ &+ \left(\mathcal{L}\overline{\phi} - \mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi} - f\overline{\phi}(\Omega^{(i)})\right). \end{aligned} \tag{55}$$

If each overlapped region contains a layer of finite elements, equation (55) can be written into the form

$$\begin{aligned}(\mathcal{L}\overline{\phi}, f\overline{\phi}) = &[(\mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi}(\Omega^{(i)})) \\ &+ \frac{1}{2}(\mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi}(\partial\Omega^{(i)}_{\text{new}})) + \frac{1}{2}(f\overline{\phi}(\Omega^{(i)}), \mathcal{L}\overline{\phi}(\partial\Omega^{(i)}_{\text{new}}))] \\ &+[(\mathcal{L}\overline{\phi} - \mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi} - f\overline{\phi}(\Omega^{(i)})) \\ &+ \frac{1}{2}(\mathcal{L}\overline{\phi}(\Omega^{(i)}), f\overline{\phi}(\partial\Omega^{(i)}_{\text{new}})) + \frac{1}{2}(f\overline{\phi}(\Omega^{(i)}), \mathcal{L}\overline{\phi}(\partial\Omega^{(i)}_{\text{new}}))]. \end{aligned} \tag{56}$$

Denote by $\mu^{(i)}$ the quantity in the first brace of the right side of equation (56). The quantity in the second brace of the right side of equation (56) can be further rewritten in terms of the remaining subdomains. Therefore, equation (56) can be written as:

$$(\mathcal{L}\overline{\phi}, f\overline{\phi}) = \sum_{i=1}^{N_p} \mu^{(i)} \tag{57}$$

where

$$\mu^{(i)} = \{\Lambda^{(i)}\}^T[K^{(i)}]\{\Lambda^{(i)}\} + \{\Lambda^{(i)}\}^T[\widehat{K}]\{\widehat{\Lambda}^{(i)}\}. \tag{58}$$

Similarly, we can obtain

$$(f\overline{\phi}, f\overline{\phi}) = \sum_{i=1}^{N_p} \nu^{(i)} \tag{59}$$

where

$$\nu^{(i)} = \{\Lambda^{(i)}\}^T[M^{(i)}]\{\Lambda^{(i)}\} + \{\Lambda^{(i)}\}^T[\widehat{M}^{(i)}]\{\widehat{\Lambda}^{(i)}\}.$$

The initial procedure is to select $\{\Lambda^{(i)}\}$ where $(i = 1, 2, \ldots, N_p)$, then by equation (54) to transfer each initial $\{\Lambda^{(i)}\}$ to its cooperating subproblems, and finally to compute $(\mathcal{L}\overline{\phi}, f\overline{\phi}) = \sum_{i=1}^{N_p} \mu^{(i)}$ and $(f\overline{\phi}, f\overline{\phi}) = \sum_{i=1}^{N_p} \nu^{(i)}$ and λ. This new approach not only has the capability of dealing with large scale problems but also provides a high potential to parallel implementations. A domain decomposition procedure for eigenvalue problems is shown in Table I.

Examples and Discussions. Two examples will be used to demonstrated the procedures shown in Table I. The first example is defined as

$$y'' + \lambda y = 0 \qquad \text{where} \qquad x \in [0, \pi] \text{ and } \lambda = w^2 \tag{60}$$

subject to $y(0) = y(\pi) = 0$. It can be easily carried out that an eigenpair of equation (60) can be written in the form ω and $\sin \omega x$ where ω is an integer. In this example, the interval $[0, \pi]$ is divided into 3 equal length subintervals. Each subinterval is an element. Since $(y, y'') = -(y', y')$, equations (32) and (39) require a linear interpolation, i.e. piecewise linear interpolations are sufficient for approximating a solution to equation (60). Each internal node has an interpolation so that there are two unknowns y_1 and y_2 to be determined. For subinterval $[x_{m-1}, x_m]$ where $x_m = m\frac{\pi}{3}$ is the coordinate of the m–th node, the interpolation $N_m(x)$ and $N_{m-1}(x)$ can be written as

$$N_{m-1}(x) = \frac{3}{\pi}(x_m - x) \qquad \text{and} \qquad N_m(x) = \frac{3}{\pi}(x - x_{m-1}), \tag{61}$$

respectively. Then

$$\begin{Bmatrix} N_{m-1} \\ N_m \end{Bmatrix} = \frac{3}{\pi} \begin{Bmatrix} x_m - x \\ x - x_{m-1} \end{Bmatrix} \qquad \text{and} \qquad \frac{d}{dx} \begin{Bmatrix} N_{m-1} \\ N_m \end{Bmatrix} = \frac{3}{\pi} \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}.$$

Using equation (32), an element stiffness matrix can be written as:

$$\begin{aligned} \text{element stiffness matrix} &= -\int_{x_{m-1}}^{x_m} (\frac{3}{\pi})^2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} dx \\ &= -\frac{3}{\pi} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \end{aligned} \tag{62}$$

Using equation (39), an element mass matrix can be written as:

$$\begin{aligned} \text{element mass matrix} &= \int_{x_{m-1}}^{x_m} (\frac{3}{\pi})^2 \begin{bmatrix} (x_m - x)^2 & (x_m - x)(x - x_{m-1}) \\ (x_m - x)(x - x_{m-1}) & (x - x_{m-1})^2 \end{bmatrix} dx \\ &= \frac{\pi}{18} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \end{aligned} \tag{63}$$

where the property $x_m - x_{m-1} = \frac{\pi}{3}$ has been used. The generated 3 finite elements are then grouped into 2 subdomains for defining subproblems, which are ranged within $[x_0, x_2]$ and $[x_1, x_3]$, respectively.

Table I. A Domain Decomposition Procedure for Eigenproblems

Initial Procedure	(a) Select $\{\Lambda^{(i)}\}$ in each subdomain. (b) For every i, send $\{\widehat{\Lambda}^{(k)}\} \leftarrow [C_{ki}]\{\Lambda^{(i)}\}$ to its neighboring subdomains. (c) Compute $(\mathcal{L}\overline{\phi}, f\overline{\phi}) = \sum_{i=1}^{N_p} \mu^{(i)}$. (d) Compute $(f\overline{\phi}, f\overline{\phi}) = \sum_{i=1}^{N_p} \nu^{(i)}$. (e) $\lambda \leftarrow -\dfrac{(\mathcal{L}\overline{\phi}, f\overline{\phi})}{(f\overline{\phi}, f\overline{\phi})}$.
In each subdomain, repeat the following procedures until convergence.	
Receiving Procedure	$\{r^{(i)}\} \leftarrow \left[[\widehat{K}^{(i)}] + \lambda[\widehat{M}^{(i)}]\right]\{\widehat{\Lambda}^{(i)}\}$.
Solving Procedure	$\{\Delta\Lambda^{(i)}\} \leftarrow -[[K^{(i)}] + \lambda[M^{(i)}]]^{-1}\{r^{(i)}\} - \{\Lambda^{(i)}\}$.
Updating Procedure	(a) $\{\Lambda^{(i)}\} \leftarrow \{\Lambda^{(i)}\} + \{\Delta\Lambda^{(i)}\}$. (b) $\sigma^{(i)} \leftarrow \{\Delta\Lambda^{(i)}\}^T[K^{(i)}]\{2\Lambda^{(i)} - \Delta\Lambda^{(i)}\}$. (c) $\zeta^{(i)} \leftarrow \{\Delta\Lambda^{(i)}\}^T[M^{(i)}]\{2\Lambda^{(i)} - \Delta\Lambda^{(i)}\}$. (d) $\tau^{(i)} \leftarrow 2 \cdot \{\Delta\Lambda^{(i)}\}^T[\widehat{K}^{(i)}]\{\widehat{\Lambda}^{(i)}\}$. (e) $\eta^{(i)} \leftarrow 2 \cdot \{\Delta\Lambda^{(i)}\}^T[\widehat{M}^{(i)}]\{\widehat{\Lambda}^{(i)}\}$. (f) $(\mathcal{L}\overline{\phi}, f\overline{\phi}) \leftarrow (\mathcal{L}\overline{\phi}, f\overline{\phi}) + \sigma^{(i)} + \tau^{(i)}$. (g) $(f\overline{\phi}, f\overline{\phi}) \leftarrow (f\overline{\phi}, f\overline{\phi}) + \zeta^{(i)} + \eta^{(i)}$. (h) $\lambda \leftarrow -\dfrac{(\mathcal{L}\overline{\phi}, f\overline{\phi})}{(f\overline{\phi}, f\overline{\phi})}$.
Sending Procedure	Send $\{\widehat{\Lambda}^{(k)}\} \leftarrow [C_{ki}]\{\Lambda^{(i)}\}$ to subdomain $\Omega^{(k)}$ adjacent to $\Omega^{(i)}$.
The superscript (i) indicates a local variable in subdomain $\Omega^{(i)}$.	

The first subdomain $[x_0, x_2]$ has two degrees of freedom y_1 and y_2 to be determined where y_2 is assigned to the first degree of freedom and y_1 is assigned to the second degree of freedom so that

$$
\begin{aligned}
&[\widehat{Z}^{(1)}] = [1 \;\; 0], && [Z^{(1)}] = [0 \;\; 1], && [S^{(1)}] = -\tfrac{3}{\pi}\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}, \\
&[W^{(1)}] = \tfrac{\pi}{18}\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}, && [K^{(1)}] = -\tfrac{6}{\pi}, && [\widehat{K}^{(1)}] = \tfrac{3}{\pi}, \\
&[M^{(1)}] = \tfrac{2}{9}\pi, && [\widehat{M}^{(1)}] = \tfrac{\pi}{18}.
\end{aligned}
\tag{64}
$$

The second subdomain $[x_1, x_3]$ contains two finite elements $[x_1, x_2]$ and $[x_2, x_3]$ with two degrees of freedom y_1 and y_2 where y_1 is the first degree of freedom and y_2 is

the second degree of freedom so that

$$
\begin{aligned}
&[\widehat{Z}^{(2)}] = [1 \;\; 0], && [Z^{(2)}] = [0 \;\; 1], && [S^{(2)}] = -\tfrac{3}{\pi}\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}, \\
&[W^{(2)}] = \tfrac{\pi}{18}\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}, && [K^{(2)}] = -\tfrac{6}{\pi}, && [\widehat{K}^{(2)}] = \tfrac{3}{\pi}, \\
&[M^{(2)}] = \tfrac{2}{9}\pi, && [\widehat{M}^{(2)}] = \tfrac{\pi}{18}.
\end{aligned} \tag{65}
$$

Since $\{\widehat{\Lambda}^{(1)}\} = y_2$ and $\{\widehat{\Lambda}^{(2)}\} = y_1$ in this example, with the procedures shown in Table I and equations (64) and (65), the governing equations for these 2 subdomains are shown in Table II. By the trial that $y_1 = y_2 = 1$, we can obtain an eigenpair that $y_1 = 1$ and $y_2 = 1$ and $\lambda = \omega^2 = 1.094269$, which is the least eigenvalue with the exact value $\omega = 1$. A good approximation to the least eigenvalue can be obtained by refining mesh, for example $\lambda = \omega^2 = 1.003663$ when the interval $[0, \pi]$ is divided into 15 finite elements with initial $y_1 = y_2 = \ldots = y_{14} = 1$. Similarly, using the trial $y_1 = 1$ and $y_2 = -2$ obtains the eigenpair with $\lambda = 5.471346$.

The framework shown in Table I can be used to solve an eigenvalue problem in the form of (12) where Ω can be a one–dimensional, two–dimensional, or higher–dimensional domain. For example, let us consider the second example defined as:

$$
\begin{cases} \nabla^2\phi + \lambda\phi = 0 & \text{in } \Omega \text{ the square } (-1,1) \times (-1,1) \\ \phi\,|_{\partial\Omega} = 0. & \end{cases}
$$

Ω is meshed into 20449 square elements which are then divided into 28 overlapped subdomains. Since $(B, \nabla^2 B) = -(\nabla B, \nabla B)$, piecewise bilinear interpolations are sufficient to formulate the stiffness and mass matrices defined by equations (32) and (39), respectively.

A computer code based upon the procedures in Table I has been developed for numerical solutions of eigenvalue problems. Using an admissible trial with unit coefficients, an eigenpair with $\lambda = 4.935176$, which is the least eigenvalue with the exact value $\lambda = 4.9348022\ldots$, can be obtained in 305 iterations. An admissible trial is required by the author's quotient iterations, which has to satisfy the requirements introduced previously. It is easy to realize that different admissible trials can lead to the same eigenpair, for example using the trial $y_1 = 2$ and $y_2 = 5$ to solve the first example leads to the pair with $\lambda = \omega^2 = 1.094269$ which also can be obtained by the trial $y_1 = 1$ and $y_2 = 1$. This means that when computing multi–eigenpairs, the presented method would waste some computing time because different admissible trials can converge to a similar eigenpair. Selecting an admissible trial for a particular eigenpair is theoretical difficult for nonlinear iterative schemes for example the Newton method, and is a common problem to nonlinear analyses. The presented method also suffers from this disadvantage. An efficient numerical procedure to find an admissible trial for a particular eigenvalue is under study. It is still at early stage for dealing with eigenvalue problems by the technique of domain decomposition. Four methods of Luo ([3], [4], and [5]) were derived by using a different technique, space decomposition [6], to formulate the two author's quotient iterations. Among them, the third method ([3] and [4]) can efficiently compute multi–eigenpairs.

This work uses the author's quotient iterations to define subproblems, which leads to an innovative procedure to deal with eigenvalue problems as shown in Table I. The procedures shown in Table I are well suited in a parallel environment, because

most computing procedures require only local variables the ones with superscripts (i) and (k). Only the steps (f), (g), and (h) of the Updating Procedure need global variables $(\mathcal{L}\overline{\phi}, f\overline{\phi})$, $(f\overline{\phi}, f\overline{\phi})$ and λ which can be modified locally. The presented method not only provides a procedure to deal with large scale problems but also shows a procedure with high potential to deal with parallel computations. Parallel implementations of this approach report by [8].

Table II. Governing Equations of the First Example

	Subdomain I: $[x_0, x_2]$	Subdomain II: $[x_1, x_3]$
Initial Procedure	$\mu^{(1)} = -\frac{6}{\pi}y_1^2 + \frac{3}{\pi}y_1y_2$ $\nu^{(1)} = \frac{2\pi}{9}y_1^2 + \frac{\pi}{18}y_1y_2$	$\mu^{(2)} = -\frac{6}{\pi}y_2^2 + \frac{3}{\pi}y_1y_2$ $\nu^{(2)} = \frac{2\pi}{9}y_2^2 + \frac{\pi}{18}y_1y_2$
Receiving + Solving Procesure	$\Delta y_1 \leftarrow -\frac{\frac{3}{\pi}+\lambda\frac{\pi}{18}}{\frac{-6}{\pi}+\lambda\frac{2\pi}{9}}y_2 - y_1$	$\Delta y_2 \leftarrow -\frac{\frac{3}{\pi}+\lambda\frac{\pi}{18}}{\frac{-6}{\pi}+\lambda\frac{2\pi}{9}}y_1 - y_2$
Updating Procedure	$(a)\ y_1 \leftarrow y_1 + \Delta y_1$ $(b)\ (\mathcal{L}\overline{\phi}, f\overline{\phi}) \leftarrow (\mathcal{L}\overline{\phi}, f\overline{\phi}) + \Delta y_1 \cdot \frac{6}{\pi} \cdot (-2y_1 + \Delta y_1 + y_2)$ $(c)\ (f\overline{\phi}, f\overline{\phi}) \leftarrow (f\overline{\phi}, f\overline{\phi}) + \frac{\pi}{9} \cdot \Delta y_1 \cdot (4y_1 - 2 \cdot \Delta y_1 + y_2)$	$(a)\ y_2 \leftarrow y_2 + \Delta y_2$ $(b)\ (\mathcal{L}\overline{\phi}, f\overline{\phi}) \leftarrow (\mathcal{L}\overline{\phi}, f\overline{\phi}) + \Delta y_2 \cdot \frac{6}{\pi} \cdot (-2y_2 + \Delta y_2 + y_1)$ $(c)\ (f\overline{\phi}, f\overline{\phi}) \leftarrow (f\overline{\phi}, f\overline{\phi}) + \frac{\pi}{9} \cdot \Delta y_2 \cdot (4y_2 - 2 \cdot \Delta y_2 + y_1)$
For a set of trial y_1 and y_2, compute $(\mathcal{L}\overline{\phi}, f\overline{\phi})$, $(f\overline{\phi}, f\overline{\phi})$ and λ by initial procedure. Then, alternatively compute subproblems until convergence.		

References.

[1] Glowinski R., Golub G. H., Meurant G. A., and Périaux J., First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988.

[2] J.-C. Luo, Parallel Algorithms for the Finite Element Method, Ph.D. Thesis, Columbia University, 1988.

[3] J.-C. Luo, Computing Eigenvalues/Eigenfunctions by Using Implicit Decomposition Method, 1990 SIAM Annual Meeting, July 16–20, 1990, Hyatt Regency Hotel, Chicago, Illinois.

[4] J.-C. Luo, Solving Eigenvalue Problems by Implicit Decomposition, *Numerical Methods for Partial Differential Equations*, Vol. 7, No. 2 (1991), pp. 113–145.

[5] J.-C. Luo, An Extension of Luo Methods for Computing Eigenpairs, Computer Methods in Applied Mechanics and Engineering, (in review).

[6] J.-C. Luo and M. B. Friedman, A Study on Decomposition Methods, Computers & Mathematics with Applications, Vol. 21, No. 8 (1991), pp. 79–84.

[7] J.-C. Luo and M. B. Friedman, Analysis of Mode Shapes Using Implicit Decomposition, Computers and Structures, (to appear).

[8] J.-C. Luo and M. B. Friedman, A Parallel Algorithm for Solving Eigenvalue Problems, Second International Conference on Industrial and Applied Mathematics, July 8–12, 1991, Sheraton Washington Hotel, Washington, D.C.