

# Nonoverlapping Discretization Methods for Partial Differential Equations

Ismael Herrera, Luis M. de la Cruz, Alberto Rosas-Medina

*Instituto de Geofísica, Universidad Nacional Autónoma de México (UNAM),*

*Apdo. Postal 22-220, D.F. 14000, México*

*Received 28 May 2013; accepted 29 November 2013*

*Published online in Wiley Online Library (wileyonlinelibrary.com).*

*DOI 10.1002/num.21852*

Ideally, domain decomposition methods (DDMs) seek what we call the DDM-paradigm: “constructing the ‘global’ solution by solving ‘local’ problems, exclusively”. To achieve it, it is essential to disconnect the subdomain problems. This explains in part the success of nonoverlapping DDMs. However, in this kind of methods, different subdomains are linked by interface nodes that are shared by several subdomains. Discretization procedures for partial differential equations of a new kind, in which each node belongs to one and only one coarse-mesh subdomain, are here introduced and analyzed. A discretization method of this type was very successfully used to develop the derived vector-space-framework. Using it, it is possible to develop algorithms that satisfy the DDM-paradigm. Other enhanced numerical and computational properties of them are also discussed. © 2014 Wiley Periodicals, Inc. *Numer Methods Partial Differential Eq* 000: 000–000, 2014

*Keywords:* 100%-parallel algorithms; DDM with constraints; nonoverlapping DDM; parallel-computation; PDE-solution-methods

## I. INTRODUCTION

Mathematical models of many systems of interest, including very important continuous systems of engineering and science, are constituted by a great variety of boundary-value problems (BVP) of partial differential equations [1], or systems of such equations, whose solution methods are based on the computational processing of large-scale algebraic systems. Furthermore, the incredible expansion experienced by the existing computational hardware and software has made amenable to effective treatment problems of an ever increasing diversity and complexity, posed by engineering and scientific applications [2].

Parallel computing is outstanding among the new computational tools, especially at present when further increases in hardware speed apparently have reached insurmountable barriers. As it is well-known, the main difficulties of parallel computing are associated with the coordination of the many processors that carry out the different tasks and the information-transmission between them. Ideally, given a task, these difficulties disappear when such “a task is carried out with the

*Correspondence to:* Ismael Herrera, Instituto de Geofísica, Universidad Nacional Autónoma de México (UNAM), Apdo. Postal 22-220, D.F. 14000, México (e-mail: [iherrerarevilla@gmail.com](mailto:iherrerarevilla@gmail.com))

© 2014 Wiley Periodicals, Inc.

processors working independently of each other.” We refer to this latter condition as the “paradigm of parallel-computing software.”

The emergence of parallel computing prompted on the part of the computational-modeling community a continued and systematic effort with the purpose of harnessing it for the endeavor of solving the mathematical models of scientific and engineering systems [3]. Very early after such an effort began, it was recognized that domain decomposition methods (DDMs) were the most effective technique for applying parallel computing to the solution of partial differential equations, because such an approach drastically simplifies the coordination of the many processors that carry out the different tasks and also reduces very much the requirements of information-transmission between them. When a DDM is applied, first a discretization of the mathematical model is carried out in a fine-mesh and, afterwards, a coarse-mesh is introduced, which properly constitutes the domain-decomposition. The “DDM-paradigm,” a paradigm for domain decomposition methods concomitant with the paradigm of parallel-computing software, consists in “obtaining the ‘global’ solution by solving ‘local’ problems exclusively” (a “local” problem is one defined separately in a subdomain of the coarse-mesh). Stated in a simplistic manner, the basic idea is that, when the DDM-paradigm is satisfied, full parallelization can be achieved by assigning each subdomain to a different processor.

When intensive DDM research began much attention was given to overlapping DDMs, but soon after attention shifted to nonoverlapping DDMs. When the DDM-paradigm is taken into account, this evolution seems natural because it is easier to uncouple the “local” problems when the subdomains do not overlap. However, as it is further discussed in the next section, in this kind of methods, different subdomains are linked by interface nodes that are shared by several subdomains and, therefore, even nonoverlapping DDMs are actually overlapping when seen from the perspective of the nodes used in the discretization. So, one would expect that a more thorough uncoupling of the “local” problems could be achieved if it were possible to carry out the discretization of the BVP to be solved using a “non-overlapping system of nodes”; that is, a set of nodes with the property that each one of them belongs to one and only one subdomain of the coarse mesh. Therefore, in what follows a discretization procedure is said to be a “non-overlapping discretization method” when the system of nodes applied in it, is nonoverlapping. In this article, one such method, the derived vector-space (DVS)-discretization method, is presented and discussed. To our knowledge, this is the first nonoverlapping discretization method reported in the literature. Furthermore, this discretization method has a very general character and is equally applicable to symmetric, nonsymmetric, and indefinite (neither positive nor negative definite) matrices.

Actually, the DVS-algorithms introduced by Herrera et al. in [4, 5] apply a similar kind of discretization but its use remained unnoticed in those papers, in spite of the fact that the novelty of the DVS-approach is to a large extent due its use. Therefore, this article is devoted to present and discuss the new discretization methodology in its own merits. In Section II, the generic boundary-value problem (BVP) considered is introduced and discretized by means of any “standard” method, with “overlapping” nodes. A great diversity of BVPs can be incorporated in this scheme as very little is assumed about the nature of such a problem; the BVP may be associated with a single differential equation or a system of such equations, and the corresponding differential operator may be formally symmetric or nonsymmetric, and indefinite. Also, the “standard” discretization method used may be any, albeit the matrix of the discretized system so obtained is required to satisfy Eq. (4.3). A procedure for constructing a nonoverlapping system of nodes is also explained in Section II. Using such nodes, an enlarged vector-space containing “discontinuous-vectors” is introduced in Section III, and the nonoverlapping discretized problem is obtained in Section IV. Schur-complement formulations are given in Sections VII and VIII; they are based on generalized

Schur-complement formulas of wide applicability that are derived in Section VI. Achieving this generality was possible due to the use of very convenient matrix notations, defined in Section V. Section IX is devoted to apply the new discretization methods to develop DDM-algorithms; in particular, the DVS-algorithms of Herrera et al. mentioned before are there derived, and in Section X a geometrical summary for them is supplied. Altogether, there are four DVS-algorithms and two of them are new versions of the well-known balancing domain decomposition with constraints (BDDC) [6–8] and dual-primal finite-element tearing and interconnecting (FETI-DP) [9–12]. As for the other two, nothing similar had been reported in the literature prior to the publication of [4, 5]. An important advantage of using nonoverlapping discretizations of BVPs is that such procedures permit achieving the DDM-paradigm, as it is shown in Section XI. Section XII is devoted to numerical and computational experiments, while Section XIII to conclusions.

## II. THE BVP AND THE NONOVERLAPPING NODES

Consider a well-posed BVP (“the BVP”), defined by the partial differential equation (or system of such equations)

$$\mathcal{L}u = f_{\Omega} \quad (2.1)$$

and suitable boundary conditions. To treat it, we first introduce a mesh (“the fine-mesh”) and apply a standard (“overlapping”) method of discretization to obtain the following discrete version of it (“the standard discretization” of the BVP):

$$\underline{M} \underline{U} = \underline{F} \quad (2.2)$$

The set of nodes of the fine-mesh will be denoted by  $\widehat{X}$  and will be referred to as the “original-nodes.” In Eq. (2.2),  $\underline{U}$  and  $\underline{F}$  are “original-vectors”; such vectors are functions defined in the set of original-nodes. To be able to treat BVPs defined by a single differential equation and also by systems of such equations in a unified manner, it is assumed that the values of original-vectors at each node of the fine-mesh are  $n - D$  vectors. When  $n = 1$ , original-vectors such as  $\underline{U}$  and  $\underline{F}$ , are real-valued functions defined in the set of original-nodes, as it is usually the case when treating a single differential equation.

Next, as it is usually done in nonoverlapping domain decomposition methods [4], we introduce another mesh (the coarse-mesh); properly, this latter mesh constitutes the domain-decomposition from which this kind of methods receives its name. However, when the coarse-mesh is introduced generally, some of the nodes of the fine-mesh belong to the closure of more than one subdomain of the coarse-mesh (this situation is illustrated in Figs. 1 and 2) and, due to this fact, we say that the system of nodes of the fine-mesh is overlapping (in spite of the fact that the method is nonoverlapping); therefore, we proceed to introduce a nonoverlapping system of nodes. To this end, we divide each node into a number of pieces equal to the number of subdomains it belongs to (Fig. 3) and then one, and only one, of such pieces is allocated in each one of such subdomains. Clearly, in this manner, a nonoverlapping set of nodes is obtained, in the sense that each node of such a set belongs to one and only one subdomain of the coarse-mesh (Fig. 4); the nodes so obtained will be referred to as derived-nodes.

In our developments, the following notation is adopted: the labels  $p, q$ , etc. will be used to denote original-nodes. Therefore, under the assumption that the total number of nodes of the fine mesh is  $N$ , the labels  $p, q$ , etc. may be any number of the set  $\{1, \dots, N\}$ . On the other hand, the

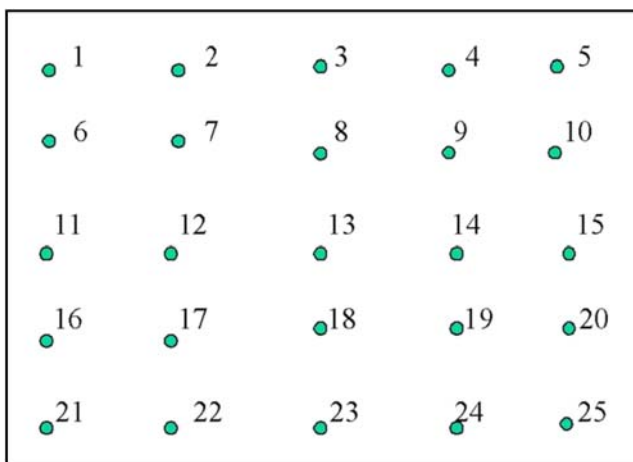


FIG. 1. The nodes of the fine-mesh. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

labels  $\alpha$ ,  $\beta$ , etc. will be used to denote the subdomains of the coarse-mesh and therefore, if the total number of subdomains of the coarse-mesh is  $E$ , the labels  $\alpha$ ,  $\beta$ , etc. may be any number of the set  $\{1, \dots, E\}$ . As for the nomenclature and notation to be used for derived-nodes, they will be labeled by pairs:  $(p, \alpha)$ ,  $p$  being the original-node it derives from and  $\alpha$  the subdomain to which it belongs. In domain decomposition methods, it is customary to classify the original-nodes into internal and interface-nodes; a node is internal if it belongs to only one subdomain closure and it is interface otherwise. The same classification applies to any derived-node,  $(p, \alpha)$ , its class being determined by that of  $p$ . Furthermore, two complementary classes of interface nodes are frequently considered: primal and dual. The subsets corresponding to internal, interface, primal,

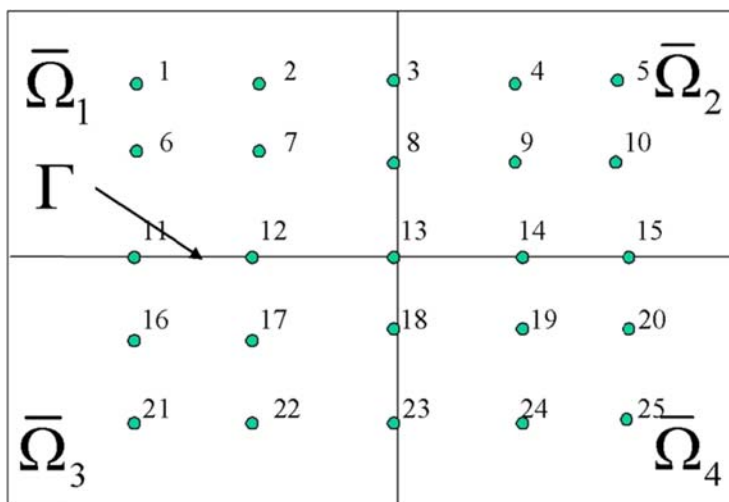


FIG. 2. The original nodes in the coarse-mesh. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

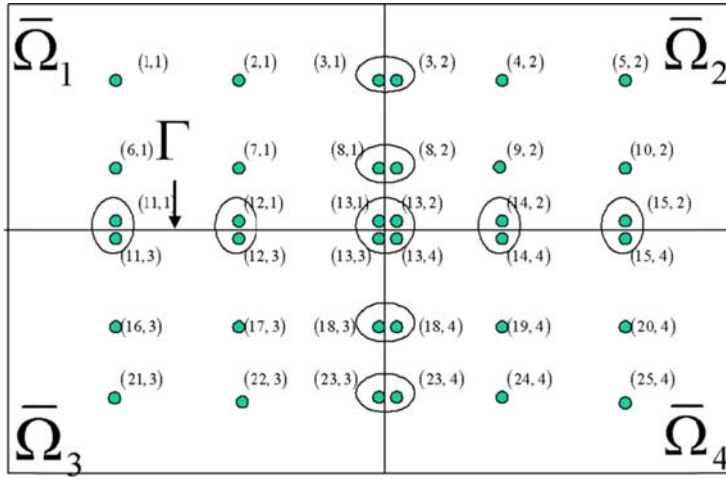


FIG. 3. The mitosis. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

and dual-derived-nodes, are denoted by  $I$ ,  $\Gamma$ ,  $\pi$ , and  $\Delta$ , respectively. Furthermore, the set  $\Pi$  is defined by

$$\Pi \equiv I \cup \pi \quad (2.3)$$

We also use the notation  $X$  for the total set of derived-nodes and introduce, for each fixed  $\alpha = 1, \dots, E$ , the subset  $X^\alpha \equiv \{(p, \alpha)\}$  of  $X$ , constituted by the derived-nodes that belong to  $\bar{\Omega}_\alpha$ . Clearly:

$$X = \bigcup_{\alpha=1}^E X^\alpha \quad \text{and} \quad X^\alpha \cap X^\beta = \emptyset, \quad \text{when} \quad \alpha \neq \beta \quad (2.4)$$

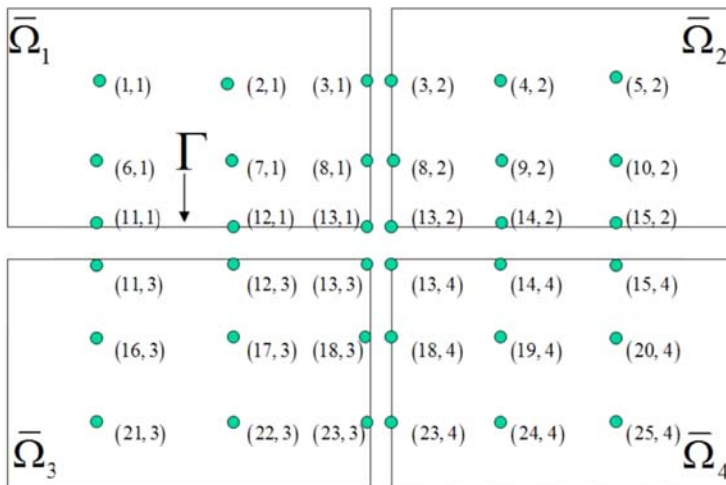


FIG. 4. The derived-nodes distributed in the coarse-mesh. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

Therefore, the family  $X^\alpha$ ,  $\alpha = 1, \dots, E$ , constitutes a nonoverlapping decomposition of  $X$ . Given a node  $p$  of the fine-mesh, the derived-nodes that originate from it constitute the set:

$$Z(p) \equiv \{(p, \alpha) \mid (p, \alpha) \in X\} \quad (2.5)$$

The multiplicity  $m(p)$  of  $p \in \widehat{X}$ , is the “cardinality” (i.e., the number of elements) of the set  $Z(p)$ . Some relevant properties of the derived-node sets introduced so far are:

$$X = I \cup \Gamma = I \cup \pi \cup \Delta = \Pi \cup \Delta \quad (2.6)$$

and

$$\emptyset = I \cap \Gamma = I \cap \pi = \pi \cap \Delta = \Pi \cap \Delta \quad (2.7)$$

### III. THE “DERIVED VECTOR-SPACE (DVS)”

The original-vectors constitute a linear space that is denoted by  $\widehat{W}$ . On the other hand, by a “derived-vector” we mean a function defined in the set of derived-nodes,  $X$ . As in the case of original-vectors, the values of such functions at each derived-node will be vectors of  $\mathbb{R}^n$ . The whole set of derived-vectors constitute a linear space,  $W$ : the “derived-vector space (DVS).” When  $\underline{u} \in W$ , we use  $\underline{u}(p, \alpha)$  for the value of  $\underline{u}$  at the derived-node  $(p, \alpha)$ . Then,  $\underline{u}(p, \alpha, i)$  -  $i = 1, \dots, n$  - stands for the  $i$ -th component of the  $n$ -D-vector  $\underline{u}(p, \alpha)$ . When  $n = 1$ , the index  $i$  is unnecessary and will be deleted.

Let  $W' \subset W$  be a linear subspace and assume  $M \subset X$  is a subset of derived-nodes. Then, the notation  $W'(M)$  represents the vector subspace of  $W'$ , whose elements vanish at every derived-node that does not belong to  $M$ . Corresponding to each local subset of derived-nodes  $X^\alpha$  introduced in Section II, there is a “local subspace of derived-vectors,”  $W^\alpha \equiv W(X^\alpha) \subset W$ . The space  $W$  is the direct sum of the family of subspaces  $\{W^1, \dots, W^E\}$ ;

$$W = W^1 \oplus \dots \oplus W^E \quad (3.1)$$

This is an important property. In particular, it implies that any  $\underline{u} \in W$  can be written uniquely as

$$\underline{u} = \underline{u}^1 + \dots + \underline{u}^E, \quad \text{with} \quad \underline{u}^\alpha \in W^\alpha, \alpha = 1, \dots, E \quad (3.2)$$

For every pair of vectors,  $\underline{u} \in W$  and  $\underline{w} \in W$ , the “Euclidean inner product” is defined to be

$$\underline{u} \cdot \underline{w} \equiv \sum_{(p, \alpha) \in X} \underline{u}(p, \alpha) \odot \underline{w}(p, \alpha) = \sum_{\alpha=1}^E \sum_{p \in \overline{\Omega}_\alpha} \underline{u}(p, \alpha) \odot \underline{w}(p, \alpha) \quad (3.3)$$

Here, the symbol  $\odot$  stands for the standard inner-product of  $\mathbb{R}^n$  - vectors ; that is,

$$\underline{u}(p, \alpha) \odot \underline{w}(p, \alpha) \equiv \sum_{i=1}^n \underline{u}(p, \alpha, i) \underline{w}(p, \alpha, i) \quad (3.4)$$

When  $n = 1$ , this reduces to

$$\underline{u} \cdot \underline{w} = \sum_{(p,\alpha) \in X} \underline{u}(p,\alpha) \underline{w}(p,\alpha) = \sum_{\alpha=1}^E \sum_{p \in \bar{\Omega}_\alpha} u(p,\alpha) w(p,\alpha) \quad (3.5)$$

The derived-vector space,  $W$ , constitutes a finite dimensional Hilbert-space with respect to the Euclidean inner product. We observe that Euclidean inner product, depends on the fine-mesh that is used, but it is independent of the BVP considered.

A derived-vector,  $\underline{u}$ , is said to be “continuous” when  $u(p,\alpha)$  is independent of  $\alpha$ , for every  $(p,\alpha) \in X$ . The subset of continuous vectors,  $W_{12} \subset W$ , constitutes a linear subspace of  $W$ . The natural injection,  $R : \widehat{W} \rightarrow W$ , of  $\widehat{W}$  into  $W$ , is defined by the condition that, for every  $\widehat{u} \in \widehat{W}$ , one has

$$(R\widehat{u})(p,\alpha) = \widehat{u}(p), \quad \forall (p,\alpha) \in X \quad (3.6)$$

We observe that  $R : \widehat{W} \rightarrow W_{12}$ . Furthermore, it can be seen that the mapping  $R : \widehat{W} \rightarrow W_{12}$  is one-to-one. Thus, we write  $R^{-1} : W_{12} \rightarrow \widehat{W}$  for the inverse of  $R$ , when restricted to  $W_{12}$ . The whole set,  $\widehat{W}$ , of vectors associated with the fine-mesh also constitutes a finite dimensional Hilbert-space, in this case with respect to the inner product:

$$\underline{U} \cdot \underline{V} \equiv \sum_{p=1}^N U(p) \odot V(p), \quad \underline{U}, \underline{V} \in \widehat{W} \quad (3.7)$$

Linear transformations of the space  $W$  into itself, and also of  $\widehat{W}$  into itself, will be considered. To each one of them corresponds a unique matrix. For matrices such as  $\underline{\underline{M}}$ , occurring in Eq. (2.2), we use the notation:

$$\underline{\underline{M}} \equiv (M_{pq}), \quad \text{where } p, q \in \widehat{X} \quad (3.8)$$

and for matrices such as  $\underline{\underline{A}}$ , to be introduced later, which transforms  $W$  into itself, we use the notation:

$$\underline{\underline{A}} \equiv (A_{(p,\alpha)(q,\beta)}), \quad \text{where } (p,\alpha), (q,\beta) \in X \quad (3.9)$$

Using this notation, we define the auxiliary matrices  $\underline{\underline{m}}$  and  $\widehat{\underline{\underline{m}}}$  by:

$$\underline{\underline{m}} \equiv (m(p) \delta_{(p,\alpha)(q,\beta)}) \quad \text{and} \quad \widehat{\underline{\underline{m}}} \equiv (m(p) \delta_{pq}) \quad (3.10)$$

They correspond to mappings  $\underline{\underline{m}} : W_{12} \rightarrow W_{12}$  and  $\widehat{\underline{\underline{m}}} : \widehat{W} \rightarrow \widehat{W}$ . We notice also that, when  $\underline{U} \in \widehat{W}$  and  $\underline{u} \in W_{12}$ ,

$$R(\widehat{\underline{\underline{m}}} \underline{U}) = \underline{\underline{m}} R \underline{U} \quad \text{and} \quad R^{-1}(\underline{\underline{m}} \underline{u}) = \widehat{\underline{\underline{m}}} R^{-1} \underline{u} \quad (3.11)$$

Using the fact that the mapping  $R : \widehat{W} \rightarrow W_{12}$  is one-to-one, in the following developments we write  $\underline{u} \equiv RU$  and  $\underline{v} \equiv RV$ , when  $\underline{u}, \underline{v} \in W_{12}$  or  $\underline{U}, \underline{V} \in \widehat{W}$ . Using this notation, we observe that

$$\underline{u} \cdot \underline{v} = \sum_{p \in \widehat{\Omega}_\alpha} \sum_{\alpha=1}^E \underline{u}(p, \alpha) \odot \underline{v}(p, \alpha) = \sum_p m(p) \underline{U}(p, \alpha) \odot \underline{V}(p, \alpha) = (\widehat{\underline{m}} \underline{U}) \cdot \underline{V} \quad (3.12)$$

Furthermore,

$$\underline{u} \cdot \underline{v} = \widehat{\underline{m}} \underline{U} \cdot \underline{V} = \underline{U} \cdot (\widehat{\underline{m}} \underline{V}), \quad \forall \underline{u}, \underline{v} \in W_{12} \quad (3.13)$$

and

$$\underline{U} \cdot \underline{V} = \underline{\underline{m}}^{-1} \underline{u} \cdot \underline{v} = \underline{u} \cdot (\underline{\underline{m}}^{-1} \underline{v}), \quad \forall \underline{U}, \underline{V} \in \widehat{W} \quad (3.14)$$

More explicitly, these relations are

$$\underline{u} \cdot \underline{v} = \underline{\underline{m}} R^{-1} \underline{u} \cdot R^{-1} \underline{v} = R^{-1} \underline{u} \cdot (\widehat{\underline{m}} R^{-1} \underline{v}), \quad \forall \underline{u}, \underline{v} \in W_{12} \quad (3.15)$$

and

$$\underline{U} \cdot \underline{V} = \underline{\underline{m}}^{-1} RU \cdot RV = RU \cdot (\underline{\underline{m}}^{-1} RV), \quad \forall \underline{U}, \underline{V} \in \widehat{W} \quad (3.16)$$

Next, we introduce the subspace  $W_{11} \subset W$ , which is defined to be the orthogonal complement of  $W_{12}$ , with respect to the Euclidean inner product. In this manner, the space  $W$  is decomposed into two orthogonal complementary subspaces:  $W_{11}$  and  $W_{12}$ , which fulfill

$$W = W_{11} \oplus W_{12} \quad (3.17)$$

Two matrices  $\underline{\underline{a}} : W \rightarrow W$  and  $\underline{\underline{j}} : W \rightarrow W$  are now introduced; they are the orthogonal-projection operators, with respect to the Euclidean inner product, on  $W_{12}$  and  $W_{11}$ , respectively. The first one will be referred to as the “average operator” and the second one will be the “jump operator.” If  $\underline{u} \in W_{11}$ , then  $\underline{\underline{a}} \underline{u} = 0$ ; that is, vectors of  $W_{11} \subset W$  are “zero-average vectors.” Similarly, if  $\underline{u} \in W_{12}$ , then  $\underline{\underline{j}} \underline{u} = 0$ ; that is, vectors of  $W_{12} \subset W$  are “zero-jump vectors.” We observe that in view of Eq. (3.17), every derived-vector,  $\underline{u} \in W$ , can be written in a unique manner as the sum of a zero-average vector plus a continuous vector; indeed:

$$\underline{u} = \underline{u}_{11} + \underline{u}_{12} \quad \text{with} \quad \begin{cases} \underline{u}_{11} \equiv \underline{\underline{j}} \underline{u} \in W_{11} \\ \underline{u}_{12} \equiv \underline{\underline{a}} \underline{u} \in W_{12} \end{cases} \quad (3.18)$$

It can be seen that  $\underline{\underline{I}} = \underline{\underline{j}} + \underline{\underline{a}}$ , from which it follows that

$$\underline{\underline{j}} = \underline{\underline{I}} - \underline{\underline{a}} \quad (3.19)$$

An explicit expression for  $\underline{\underline{a}}$  is:

$$\underline{\underline{a}} = (a_{(p,\alpha)(q,\beta)}) \quad \text{with} \quad a_{(p,\alpha)(q,\beta)} = \frac{\delta_{pq} \delta_{\alpha\beta}}{m(q)} \quad (3.20)$$



Next, we define several subspaces of  $W$  that will be used in the sequel. They are:  $W_I \equiv W(\Gamma)$  and  $W_\Gamma \equiv W(\Gamma)$ . In view of Eqs. (2.6) and (2.7), we have

$$W = W_I \oplus W_\Gamma \quad (3.21)$$

#### IV. THE NONOVERLAPPING DISCRETIZATION

In this section, we address the main subject of the present article and to this end, we present a nonoverlapping discretization method; namely, the DVS-discretization method.

To start with, we define

$$\delta_{pq}^\alpha \equiv \begin{cases} 1, & \text{if } p, q \in \overline{\Omega}_\alpha, \\ 0, & \text{otherwise} \end{cases}, \quad \alpha = 1, \dots, E; \text{ and} \quad (4.1)$$

together with

$$m(p, q) \equiv \sum_{\alpha=1}^E \delta_{pq}^\alpha \quad (4.2)$$

The function  $m(p, q)$  is the “multiplicity” of the pair  $(p, q)$ , which can be zero, when the pair  $p$  and  $q$  do not occur simultaneously in any subdomain-closure. The DVS-discretization method, to which the present paper is devoted, has a wide range of applicability; it can be applied whenever the following basic assumption (or, axiom) is fulfilled:

$$m(p, q) = 0 \Rightarrow M_{pq} = 0 \quad (4.3)$$

The fact that  $m(p, q)$  may take the value zero is inconvenient for the following developments and, therefore, we replace it by a function  $s(p, q)$ , which is essentially the same except that it never is zero. It is defined by

$$s(p, q) \equiv \begin{cases} 1, & \text{when } m(p, q) = 0 \\ m(p, q), & \text{otherwise} \end{cases} \quad (4.4)$$

For  $\gamma = 1, \dots, E$ , we define the matrices

$$\underline{\underline{M}}^\gamma \equiv (M_{pq}^\gamma) \quad \text{with} \quad M_{pq}^\gamma \equiv \frac{M_{pq}}{s(p, q)} \delta_{pq}^\gamma \quad (4.5)$$

It can be verified that

$$\sum_{\gamma=1}^E \underline{\underline{M}}^\gamma = \underline{\underline{M}} \quad \text{because} \quad \sum_{\gamma=1}^E M_{pq}^\gamma = M_{pq} \quad (4.6)$$

Eq. (4.6) implies that

$$\sum_{\gamma=1}^E \underline{V} \cdot \underline{\underline{M}}^\gamma \underline{U} = \underline{V} \cdot \underline{\underline{M}} \underline{U}, \quad \forall \underline{U}, \underline{V} \in \widehat{W} \quad (4.7)$$

Next, we define the matrices:

$$\underline{\underline{A}}^\gamma \equiv (A_{(p,\alpha)(q,\beta)}^\gamma) \quad \text{with} \quad A_{(p,\alpha)(q,\beta)}^\gamma \equiv M_{pq}^\gamma \delta_{(\alpha,\gamma)} \delta_{(\beta,\gamma)} \quad (4.8)$$

and

$$\underline{\underline{A}}' \equiv \sum_{\gamma=1}^E \underline{\underline{A}}^\gamma \quad (4.9)$$

We observe that  $\underline{\underline{A}}^\gamma$  is a mapping of  $W^\gamma$  into itself; that is,  $\underline{\underline{A}}^\gamma : W^\gamma \rightarrow W^\gamma$  and the matrix  $\underline{\underline{A}}'$  is block-diagonal. Furthermore, for every  $\underline{U}, \underline{V} \in \widehat{W}$  define  $\underline{u} \equiv R\underline{U}$  and  $\underline{v} \equiv R\underline{V}$ , then with this notation for every  $\gamma = 1, \dots, E$ , one has:

$$\underline{v} \cdot \underline{\underline{A}}^\gamma \underline{u} = \underline{V} \cdot \underline{\underline{M}}^\gamma \underline{U} \quad (4.10)$$

Applying Eq. (4.7) it is seen that

$$\underline{v} \cdot \underline{\underline{A}}' \underline{u} = \underline{V} \cdot \underline{\underline{M}} \underline{U}, \quad \forall \underline{u}, \underline{v} \in W_{12} \quad (4.11)$$

**Theorem 4.1.** Let  $\underline{U} \in \widehat{W}$  and  $\underline{u} \in W$  be related by  $\underline{u} = R\underline{U}$ , while  $\underline{f} \in W_{12}$  is defined by

$$\underline{f} \equiv R \left( \widehat{\underline{\underline{m}}}^{-1} \underline{F} \right) \quad (4.12)$$

Then, the equation

$$\underline{\underline{M}} \underline{U} = \underline{F} \quad (4.13)$$

is fulfilled, if and only if

$$\underline{\underline{a}} \underline{\underline{A}}' \underline{u} = \underline{f} \quad \text{and} \quad \underline{\underline{j}} \underline{u} = 0 \quad (4.14)$$

**Proof.** Assume Eq. (4.14) holds, then:

$$\underline{v} \cdot \underline{\underline{A}}' \underline{u} = \underline{v} \cdot \underline{\underline{a}} \underline{\underline{A}}' \underline{u} = \underline{v} \cdot \underline{f}, \quad \forall \underline{v} \in W_{12} \quad (4.15)$$

Hence, using Eq. (4.11), it is seen that

$$\underline{V} \cdot \underline{\underline{M}} \underline{U} = \underline{v} \cdot \underline{\underline{A}}' \underline{u} = \underline{v} \cdot \underline{\underline{a}} \underline{\underline{A}}' \underline{u} = \underline{v} \cdot \underline{f} = \underline{V} \cdot R^{-1} \left( \widehat{\underline{\underline{m}}} \underline{f} \right) = \underline{V} \cdot \underline{F}, \quad \forall \underline{V} \in \widehat{W} \quad (4.16)$$

This implies Eq. (4.13). Conversely, assume  $\underline{U} \in \widehat{W}$  and  $\underline{u} = R\underline{U}$ , then  $\underline{\underline{j}} \underline{u} = 0$ . Furthermore, when  $\underline{U}$  satisfies Eq. (4.13) we have

$$\underline{v} \cdot \underline{\underline{a}} \underline{\underline{A}}' \underline{u} = \underline{v} \cdot \underline{\underline{A}}' \underline{u} = \underline{V} \cdot \underline{\underline{M}} \underline{U} = \underline{V} \cdot \underline{F} = \underline{v} \cdot \widehat{\underline{\underline{m}}}^{-1} R \left( \underline{F} \right) = \underline{v} \cdot \underline{f}, \quad \forall \underline{v} \in W_{12} \quad (4.17)$$

This implies the first equality of Eq. (4.14), as both  $\underline{\underline{a}} \underline{\underline{A}}' \underline{u}$  and  $\underline{f}$  are continuous derived-vectors. This completes the proof of Theorem 4.1. ■

**Definition 4.1.** When Eq. (2.2) is a standard discretization of the BVP of Eq. (2.1), then Eq. (4.14) is a “DVS-discretization” of the same BVP.

In connection with Definition 4.1, and taking into account Theorem 4.1, we observe that any DVS-discretization of the BVP of Eq. (2.1) is a nonoverlapping discretization of the same BVP. Furthermore, in turn, Eq. (4.14) can be replaced by

$$\underline{\underline{a}} \underline{\underline{A}}' \underline{\underline{a}} \underline{\underline{u}} = \underline{\underline{f}} \quad \text{and} \quad \underline{\underline{j}} \underline{\underline{u}} = 0 \quad (4.18)$$

## V. MATRIX NOTATIONS

In what follows, all matrices considered are defined on the whole derived-vector-space  $W$  and, when applied to any vector of  $W$ , they yield derived-vectors. In this section, the symbol  $\underline{\underline{B}}$  will be used generically for any such a matrix.

When  $M \subset X$  and  $N \subset X$  are two sets of derived-nodes, the matrix the  $\underline{\underline{B}}_{\underline{\underline{MN}}}$  that also maps  $W$  into  $W$  is defined in the Appendix. In particular, when  $M = N$ , the matrix  $\underline{\underline{B}}_{\underline{\underline{MM}}}$  is obtained. When

$$X = M \cup N \quad \text{and} \quad \emptyset = M \cap N \quad (5.1)$$

Then,

$$W = W(M) \oplus W(N) \quad \text{and} \quad W(M) \perp W(N) \quad (5.2)$$

When Eq. (5.2) holds, each  $\underline{\underline{u}} \in W$  can be written uniquely as

$$\underline{\underline{u}} = \underline{\underline{u}}_M + \underline{\underline{u}}_N, \underline{\underline{u}}_M \in W(M) \quad \text{and} \quad \underline{\underline{u}}_N \in W(N) \quad (5.3)$$

Furthermore (see the Appendix for details),

$$\begin{aligned} \underline{\underline{B}} &= \underline{\underline{B}}_{\underline{\underline{MM}}} + \underline{\underline{B}}_{\underline{\underline{NM}}} + \underline{\underline{B}}_{\underline{\underline{MN}}} + \underline{\underline{B}}_{\underline{\underline{NN}}} \quad \text{and} \\ \underline{\underline{B}} \underline{\underline{u}} &= \underline{\underline{B}}_{\underline{\underline{MM}}} \underline{\underline{u}}_M + \underline{\underline{B}}_{\underline{\underline{NM}}} \underline{\underline{u}}_M + \underline{\underline{B}}_{\underline{\underline{MN}}} \underline{\underline{u}}_N + \underline{\underline{B}}_{\underline{\underline{NN}}} \underline{\underline{u}}_N \end{aligned} \quad (5.4)$$

We will use the notation:

$$\underline{\underline{B}} = \begin{pmatrix} \underline{\underline{B}}_{\underline{\underline{MM}}} & \underline{\underline{B}}_{\underline{\underline{MN}}} \\ \underline{\underline{B}}_{\underline{\underline{NM}}} & \underline{\underline{B}}_{\underline{\underline{NN}}} \end{pmatrix} \quad \text{and} \quad \underline{\underline{u}} = \begin{pmatrix} \underline{\underline{u}}_M \\ \underline{\underline{u}}_N \end{pmatrix} \quad (5.5)$$

Also, the standard rules of multiplication applies:

$$\begin{pmatrix} \underline{\underline{B}}_{\underline{\underline{MM}}} & \underline{\underline{B}}_{\underline{\underline{MN}}} \\ \underline{\underline{B}}_{\underline{\underline{NM}}} & \underline{\underline{B}}_{\underline{\underline{NN}}} \end{pmatrix} \begin{pmatrix} \underline{\underline{u}}_M \\ \underline{\underline{u}}_N \end{pmatrix} = \begin{pmatrix} \underline{\underline{B}}_{\underline{\underline{MM}}} \underline{\underline{u}}_M + \underline{\underline{B}}_{\underline{\underline{MN}}} \underline{\underline{u}}_N \\ \underline{\underline{B}}_{\underline{\underline{NM}}} \underline{\underline{u}}_M + \underline{\underline{B}}_{\underline{\underline{NN}}} \underline{\underline{u}}_N \end{pmatrix} \quad (5.6)$$

The definitions of the matrices used above are given in the Appendix. They fulfill

$$\begin{aligned} \underline{\underline{B}}_{\underline{\underline{MM}}} : W(M) &\rightarrow W(M), \underline{\underline{B}}_{\underline{\underline{MN}}} : W(N) \rightarrow W(M) \\ \underline{\underline{B}}_{\underline{\underline{NM}}} : W(M) &\rightarrow W(N), \underline{\underline{B}}_{\underline{\underline{NN}}} : W(N) \rightarrow W(N) \end{aligned} \quad (5.7)$$

## VI. GENERAL SCHUR-COMPLEMENT DECOMPOSITIONS

To start with, we introduce a concept that will be used in the sequel: a matrix  $\underline{\underline{B}} : W \rightarrow W$  is said to be “invertible everywhere” when, for every  $Y \subset X$ ,  $N(\underline{\underline{B}}_{YY})^\perp \supset W(Y)$ . Here, the symbols  $N(\underline{\underline{B}}_{YY})$  and  $N(\underline{\underline{B}}_{YY})^\perp$  stand for the null-subspace of  $\underline{\underline{B}}_{YY}$  and its orthogonal complement, respectively. It can be verified that every positive-definite matrix is invertible everywhere. Thus, using this latter concept some results well-known for positive-definite matrices will be extended to more general classes of matrices.

Next, we consider a matrix  $\underline{\underline{B}} : W \rightarrow W$  that is invertible everywhere, together with two subsets,  $M \subset X$  and  $N \subset X$ , of derived-nodes such that

$$W(M) + W(N) \supset N(\underline{\underline{B}})^\perp \quad \text{and} \quad W(M) \perp W(N) \quad (6.1)$$

We introduce the following notation: define the (nonlinear) operators

$$\sigma_{NN}(\underline{\underline{B}}) \equiv \left\{ \underline{\underline{B}}_{NN} - \underline{\underline{B}}_{NM}(\underline{\underline{B}}_{MM})^{\sim 1} \underline{\underline{B}}_{MN} \right\} \quad (6.2)$$

Let

$$\underline{v} = \underline{\underline{B}}^{\sim 1} \underline{w}, \quad \underline{j}_{\underline{\underline{B}}} \underline{v} = 0 \quad (6.3)$$

For the definition of  $\underline{j}_{\underline{\underline{B}}}$ , see the Appendix. Then

$$\sigma_{NN}(\underline{\underline{B}}) \underline{v}_N = \underline{w}_N - \underline{\underline{B}}_{NM}(\underline{\underline{B}}_{MM})^{\sim 1} \underline{w}_M, \quad \underline{j}_{\underline{\underline{B}}} \underline{v} = 0 \quad (6.4)$$

and

$$\underline{v}_M = (\underline{\underline{B}}_{MM})^{\sim 1} (\underline{w}_M - \underline{\underline{B}}_{MN} \underline{v}_N) \quad (6.5)$$

If we write,

$$\underline{\underline{B}}^{\sim 1} \equiv \left\{ \begin{array}{cc} (\underline{\underline{B}}^{\sim 1})_{MM} & (\underline{\underline{B}}^{\sim 1})_{MN} \\ (\underline{\underline{B}}^{\sim 1})_{NM} & (\underline{\underline{B}}^{\sim 1})_{NN} \end{array} \right\} \quad (6.6)$$

Then, it is seen that

$$\begin{aligned} (\underline{\underline{B}}^{\sim 1})_{MN} &= -(\underline{\underline{B}}_{MM})^{\sim 1} \underline{\underline{B}}_{MN} \sigma_{NN}(\underline{\underline{B}})^{\sim 1} \\ (\underline{\underline{B}}^{\sim 1})_{NM} &= -\sigma_{NN}(\underline{\underline{B}})^{\sim 1} \underline{\underline{B}}_{NM} (\underline{\underline{B}}_{MM})^{\sim 1}, \quad (\underline{\underline{B}}^{\sim 1})_{NN} = \sigma_{NN}(\underline{\underline{B}})^{\sim 1} \end{aligned} \quad (6.7)$$

## VII. SCHUR COMPLEMENT FORMULATION WITHOUT CONSTRAINTS

To deal with Eq. (4.18), we apply the general Schur formulas of Section VI setting  $\underline{\underline{B}} = \underline{\underline{a}} \underline{\underline{A}}^t$ ,  $M=I$  and  $N=\Gamma$ . Eqs. (6.4) and (6.5), respectively, are:

$$\underline{\underline{a}} \underline{\underline{S}}^t \underline{\underline{u}}_\Gamma = \underline{\underline{f}}_\Gamma - \underline{\underline{a}} \underline{\underline{A}}^t_{\Gamma I} \left( \underline{\underline{A}}_{\Pi I} \right)^{\sim 1} \underline{\underline{f}}_I, \underline{\underline{j}} \underline{\underline{u}}_\Gamma = 0 \quad (7.1)$$

and

$$\underline{\underline{u}}_I = \left( \underline{\underline{A}}_{\Pi I} \right)^{\sim 1} \left( \underline{\underline{f}}_I - \underline{\underline{A}}^t_{\Gamma I} \underline{\underline{a}} \underline{\underline{u}}_\Gamma \right) \quad (7.2)$$

Here, we have used the fact that

$$\sigma_{\Gamma\Gamma} \left( \underline{\underline{a}} \underline{\underline{A}}^t \right) = \underline{\underline{a}} \left( \underline{\underline{A}}^t_{\Gamma\Gamma} - \underline{\underline{A}}^t_{\Gamma I} \left( \underline{\underline{A}}_{\Pi I} \right)^{\sim 1} \underline{\underline{A}}^t_{\Pi\Gamma} \right) \quad (7.3)$$

and have defined

$$\underline{\underline{S}}^t \equiv \underline{\underline{A}}^t_{\Gamma\Gamma} - \underline{\underline{A}}^t_{\Gamma I} \left( \underline{\underline{A}}_{\Pi I} \right)^{\sim 1} \underline{\underline{A}}^t_{\Pi\Gamma} \quad (7.4)$$

## VIII. SCHUR COMPLEMENT FORMULATION WITH CONSTRAINTS

The constrained space,  $W' \subset W$ , to be considered is defined by

$$W' \equiv W(I) + W(\Delta) + \underline{\underline{a}} W(\pi) \quad (8.1)$$

Let  $\underline{\underline{a}}'$  be the projection in  $W'$ , then for any  $\underline{\underline{u}} \in W$  we have

$$\underline{\underline{a}}' \underline{\underline{u}} = \underline{\underline{u}}_I + \underline{\underline{u}}_\Delta + \underline{\underline{a}} \underline{\underline{u}}_\pi \quad (8.2)$$

Then, we define

$$\underline{\underline{A}} \equiv \underline{\underline{a}}' \underline{\underline{A}}^t \underline{\underline{a}}' \quad (8.3)$$

and proceed to apply the general results of Section VI, taking  $M = I \cup \pi \equiv \Pi$ ,  $N = \Delta$  and

$$\underline{\underline{B}} = \underline{\underline{a}}' \underline{\underline{A}}^t \underline{\underline{a}}' \quad (8.4)$$

In passing, we observe that Eq. (4.14) is equivalent to

$$\underline{\underline{a}} \underline{\underline{A}} \underline{\underline{u}} = \underline{\underline{f}} \quad \text{and} \quad \underline{\underline{j}} \underline{\underline{u}} = 0 \quad (8.5)$$

since  $\underline{\underline{f}} \in W'$ . Then, applying Eqs. (6.4) and (6.5), respectively, are:

$$\underline{\underline{a}} \underline{\underline{S}} \underline{\underline{u}}_\Delta = \underline{\underline{f}}_\Delta - \underline{\underline{a}} \underline{\underline{A}}_{\Delta\Pi} \left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} \underline{\underline{f}}_\Pi, \quad \underline{\underline{j}} \underline{\underline{u}}_\Delta = 0 \quad (8.6)$$

and

$$\underline{u}_\Pi = \left( \underline{A}_{\Pi\Pi} \right)^{\sim 1} \left( \underline{f}_\Pi - \underline{A}_{\Pi\Delta} \underline{a} \underline{u}_\Delta \right) \quad (8.7)$$

Here, we have used the fact that

$$\sigma_{\Delta\Delta} \left( \underline{a} \underline{A} \right) = \underline{a} \left( \underline{A}_{\Delta\Delta} - \underline{A}_{\Delta\Pi} \left( \underline{A}_{\Pi\Pi} \right)^{\sim 1} \underline{A}_{\Pi\Delta} \right) \quad (8.8)$$

and have defined

$$\underline{S} \equiv \underline{A}_{\Delta\Delta} - \underline{A}_{\Delta\Pi} \left( \underline{A}_{\Pi\Pi} \right)^{\sim 1} \underline{A}_{\Pi\Delta} \quad (8.9)$$

Following the notation indicated in Eq. (5.5), we have written

$$\underline{A} = \begin{pmatrix} \underline{A}_{\Pi\Pi} & \underline{A}_{\Pi\Delta} \\ \underline{A}_{\Delta\Pi} & \underline{A}_{\Delta\Delta} \end{pmatrix} \quad (8.10)$$

## IX. THE PRECONDITIONED DVS-ALGORITHMS WITH CONSTRAINTS

Different DVS-algorithms are derived by seeking for different pieces of information such that  $\underline{u}_\Delta \in W(\Delta)$  can be derived from it in a noncomputational costly. In particular, four algorithms will be derived by successively seeking for:  $\underline{u}_\Delta$ ,  $\underline{j} \underline{S} \underline{u}_\Delta$ ,  $\underline{S}^{\sim 1} \underline{j} \underline{S} \underline{u}_\Delta$ , and  $\underline{S} \underline{u}_\Delta$ . In all these algorithms, once  $\underline{u}_\Delta \in W_\Delta$  has been obtained, Eq. (8.7) is applied to get

$$\underline{u}_\Pi = \left( \underline{A}_{\Pi\Pi} \right)^{\sim 1} \left( \underline{f}_\Pi - \underline{A}_{\Pi\Delta} \underline{a} \underline{u}_\Delta \right) \quad (9.1)$$

### A. The DVS-BDDC Algorithm

This algorithm is obtained when the sought information is  $\underline{u}_\Delta$ . It is [4, 5]:

$$\underline{a} \underline{S}^{-1} \underline{a} \underline{S} \underline{u}_\Delta = \underline{a} \underline{S}^{-1} \underline{f}_\Delta \quad \text{and} \quad \underline{j} \underline{u}_\Delta = 0 \quad (9.2)$$

### B. The DVS-Primal-Algorithm

We set  $\underline{v}_\Delta \equiv \underline{S}^{\sim 1} \underline{j} \underline{S} \underline{u}_\Delta$  and the algorithm consists in searching for a function  $\underline{v}_\Delta \in W_\Delta$ , which fulfills

$$\underline{S}^{-1} \underline{j} \underline{S} \underline{j} \underline{v}_\Delta = \underline{S}^{-1} \underline{j} \underline{S} \underline{j} \underline{S}^{-1} \underline{f}_\Delta \quad \text{and} \quad \underline{a} \underline{S} \underline{v}_\Delta = 0 \quad (9.3)$$

Once  $\underline{v}_\Delta \in W(\Delta)$  has been obtained, then

$$\underline{u}_\Delta = \underline{a} \left( \underline{S}^{-1} \underline{f}_\Delta + \underline{v}_\Delta \right) \quad (9.4)$$

### C. The DVS-FETI-DP Algorithm

In this case, the sought information is  $\underline{j} \underline{S} \underline{u}_\Delta$ , which is denoted by  $\underline{\lambda}$ . Thus, the algorithm is: “Given  $\underline{f}_\Delta \in \underline{a} W_\Delta$ , find  $\underline{\lambda} \in W_\Delta$  such that

$$\underline{j} \underline{S} \underline{j} \underline{S}^{-1} \underline{\lambda} = -\underline{j} \underline{S} \underline{j} \underline{S}^{-1} \underline{f}_\Delta \quad \text{and} \quad \underline{a} \underline{\lambda} = 0 \quad (9.5)$$

Once  $\underline{\lambda} \in W_\Delta$  has been obtained,  $\underline{u}_\Delta \in \underline{a} W_\Delta$  is given by:

$$\underline{u}_\Delta = \underline{a} \underline{S}^{-1} (\underline{f}_\Delta + \underline{\lambda}) \quad (9.6)$$

### D. The DVS-Dual-Algorithm

The sought information is  $\underline{S} \underline{u}_\Delta$ , which is denoted by  $\underline{\mu}$ . Then, we seek for  $\underline{\mu} \in W(\Delta)$  such that

$$\underline{S} \underline{a} \underline{S}^{-1} \underline{a} \underline{\mu} = \underline{S} \underline{a} \underline{S}^{-1} \underline{f}_\Delta \quad \text{and} \quad \underline{j} \underline{S}^{-1} \underline{\mu} = 0 \quad (9.7)$$

Once  $\underline{\mu} \in W(\Delta)$  has been obtained,  $\underline{u}_\Delta \in W(\Delta)$  is given by:

$$\underline{u}_\Delta = \underline{S}^{-1} \underline{\mu} \quad (9.8)$$

## X. THE GEOMETRIC SUMMARY OF DVS-ALGORITHMS

We observe that the operator  $\underline{S}^{-1} \underline{a} \underline{S}$  occurring in the DVS-BDDC algorithm is an idempotent projection operator. Its complementary projection

$$\underline{I} - \underline{S}^{-1} \underline{a} \underline{S} = \underline{S}^{-1} \underline{j} \underline{S} \quad (10.1)$$

occurs in the DVS-PRIMAL algorithm. Similarly, the complementary projection operators  $\underline{S} \underline{j} \underline{S}^{-1}$  and

$$\underline{S} \underline{a} \underline{S}^{-1} = \underline{I} - \underline{S} \underline{j} \underline{S}^{-1} \quad (10.2)$$

occur in the DVS-FETI-DP and in the DVS-DUAL algorithms, respectively. In addition, we recall the pair of complementary projections,  $\underline{j}$  and  $\underline{a}$ . Corresponding to the three pairs of complementary projections just mentioned, we have three pairs of complementary subspaces (see, Figs. 5 and 6):

$$W_{11}(\Delta) \equiv \underline{j} W(\Delta), \quad W_{12}(\Delta) \equiv \underline{a} W(\Delta) \quad (10.3)$$

$$W_{21}(\Delta) = \underline{S}^{-1} \underline{a} \underline{S} W(\Delta), \quad W_{22}(\Delta) = \underline{S}^{-1} \underline{j} \underline{S} W(\Delta) \quad (10.4)$$

and

$$W_{31}(\Delta) = \underline{S} \underline{a} \underline{S}^{-1} W(\Delta), \quad W_{32}(\Delta) = \underline{S} \underline{j} \underline{S}^{-1} W(\Delta) \quad (10.5)$$

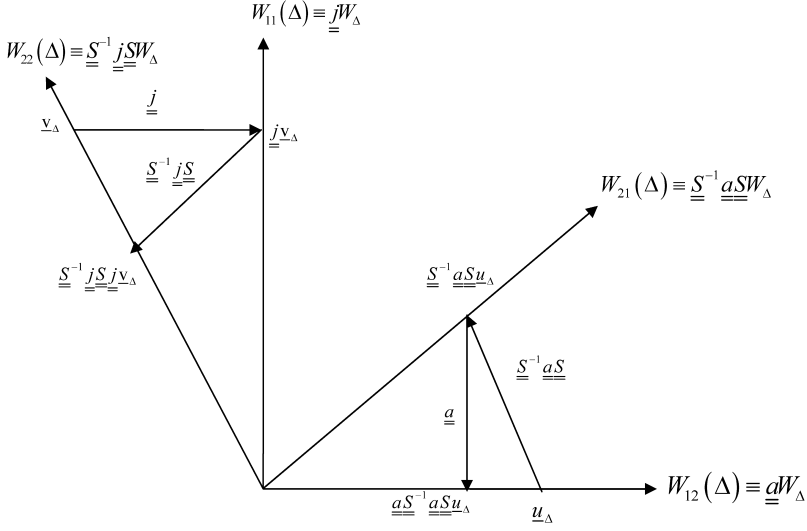


FIG. 5. DVS-algorithms: DVS-BDDC and DVS-PRIMAL.

They satisfy:

$$W(\Delta) = W_{11}(\Delta) \oplus W_{12}(\Delta) = W_{21}(\Delta) \oplus W_{22}(\Delta) = W_{31}(\Delta) \oplus W_{32}(\Delta) \quad (10.6)$$

In view of the above, a very simple geometrical summary of the DVS-algorithms can be given, which is presented in Figs. 5 and 6.

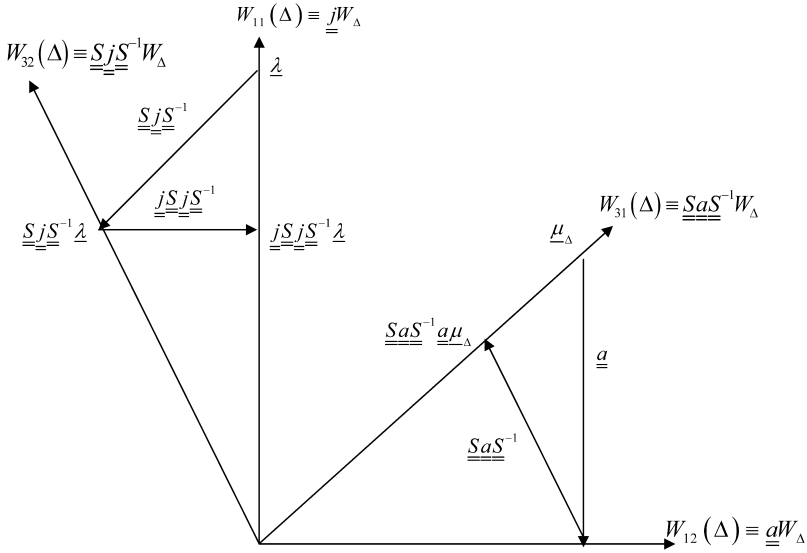


FIG. 6. DVS-algorithms: DVS-FETI-DP and DVS-DUAL.



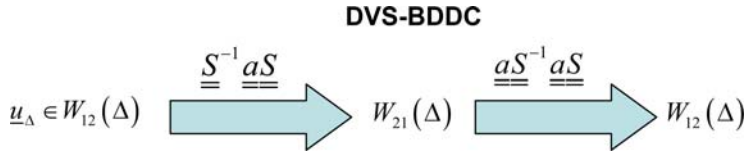


FIG. 7. DVS-BDDC. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

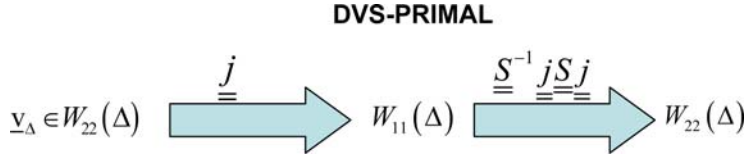


FIG. 8. DVS-PRIMAL. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

For each one of them, every iteration consists of the succession of two projections; the first one sends a trial vector of the space where the sought information is known to be, to a different space, whereas the second one returns it to the original space.

It can be seen, that according to Eqs. (9.2), (9.3), (9.5), and (9.7) for the DVS-BDDC, DVS-Primal, DVS-FETI-DP, and DVS-dual, the trial vectors are taken from  $W_{12}(\Delta)$ ,  $W_{22}(\Delta)$ ,  $W_{11}(\Delta)$ , and  $W_{31}(\Delta)$ . The processes occurring in each iteration for every DVS-algorithms are illustrated in Figs. 7–10.

## XI. HOW TO ACHIEVE THE DDM-PARADIGM

The algorithms presented in Section IX permit to develop codes that achieve the DDM-paradigm. How that can be done was explained in [4] and we draw from it. All the algorithms of Section IX are iterative algorithms and can be implemented with recourse to conjugate gradient method

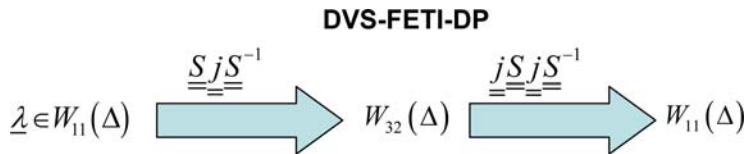


FIG. 9. DVS-FETI-DP.[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

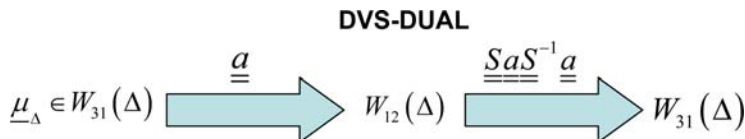


FIG. 10. DVS-DUAL.[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

(CGM), when the matrix is definite and symmetric, or some other iterative procedure such as GMRES, when that is not the case. At each iteration step, depending on the DVS-algorithm that is applied, one has to compute the action on a derived-vector of one of the following matrices:  $\underline{\underline{a}} \underline{\underline{S}}^{-1} \underline{\underline{a}} \underline{\underline{S}}$ ,  $\underline{\underline{j}} \underline{\underline{S}} \underline{\underline{j}} \underline{\underline{S}}^{-1}$ ,  $\underline{\underline{S}}^{-1} \underline{\underline{j}} \underline{\underline{S}} \underline{\underline{j}}$ , or  $\underline{\underline{S}} \underline{\underline{a}} \underline{\underline{S}}^{-1} \underline{\underline{a}}$ . Such matrices in turn are different permutations of the matrices  $\underline{\underline{S}}$ ,  $\underline{\underline{S}}^{-1}$ ,  $\underline{\underline{a}}$ , and  $\underline{\underline{j}}$ . Thus, to develop codes that achieve the DDM-paradigm one only needs to separately develop codes with that property which compute the action of each one of the matrices  $\underline{\underline{S}}$ ,  $\underline{\underline{S}}^{-1}$ ,  $\underline{\underline{a}}$ , or  $\underline{\underline{j}}$  on an arbitrary derived-vector, as was explained in [4].

### A. Application of $\underline{\underline{S}}$

We recall the definition of the matrix  $\underline{\underline{S}}$ , Eq. (8.9):

$$\underline{\underline{S}} \equiv \underline{\underline{A}}_{\Delta\Delta} - \underline{\underline{A}}_{\Delta\Pi} \left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} \underline{\underline{A}}_{\Pi\Delta} \quad (11.1)$$

To evaluate the action of  $\underline{\underline{S}}$  on any derived-vector, we need to successively evaluate the action of the following matrices  $\underline{\underline{A}}_{\Pi\Delta}$ ,  $\left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1}$ ,  $\underline{\underline{A}}_{\Delta\Pi}$ , and  $\underline{\underline{A}}_{\Delta\Delta}$ . Nothing special is required except for  $\left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1}$ , which is explained next. We have

$$\underline{\underline{A}}_{\Pi\Pi} \equiv \begin{pmatrix} \underline{\underline{A}}_{\Pi\Pi} & \underline{\underline{A}}_{\Pi\pi} \\ \underline{\underline{A}}_{\pi\Pi} & \underline{\underline{A}}_{\pi\pi} \end{pmatrix} = \begin{pmatrix} \underline{\underline{A}}_{\Pi\Pi}^t & \underline{\underline{A}}_{\Pi\pi}^t \underline{\underline{a}} \\ \underline{\underline{a}} \underline{\underline{A}}_{\pi\Pi}^t & \underline{\underline{a}} \underline{\underline{A}}_{\pi\pi}^t \underline{\underline{a}} \end{pmatrix} \quad (11.2)$$

Let  $\underline{\underline{w}} \in W$ , be an arbitrary derived-vector, and write

$$\underline{\underline{v}} \equiv \left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} \underline{\underline{w}} \quad (11.3)$$

Then,  $\underline{\underline{v}}_{\pi} \in W(\pi)$  is characterized by

$$\sigma_{\pi\pi} \left( \underline{\underline{A}}_{\Pi\Pi} \right) \underline{\underline{v}}_{\pi} = \underline{\underline{w}}_{\pi} - \underline{\underline{A}}_{\pi\Pi} \left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} \underline{\underline{w}}_1, \quad \text{subjected to} \quad \underline{\underline{j}} \underline{\underline{v}}_{\pi} = 0 \quad (11.4)$$

and can be obtained iteratively. Here,

$$\sigma_{\pi\pi} \left( \underline{\underline{A}}_{\Pi\Pi} \right) \equiv \left\{ \underline{\underline{A}}_{\pi\pi} - \underline{\underline{A}}_{\pi\Pi} \left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} \underline{\underline{A}}_{\Pi\pi} \right\} \quad (11.5)$$

We observe that the computation in parallel of the action of  $\left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1}$  is straightforward because

$$\left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} = \sum_{\alpha=1}^E \left( \underline{\underline{A}}_{\Pi\Pi}^{\alpha} \right)^{-1} \quad (11.6)$$

Once  $\underline{\underline{v}}_{\pi} \in W_r(\pi)$  has been obtained, to derive  $\underline{\underline{v}}_1$  one applies:

$$\underline{\underline{v}}_1 = \left( \underline{\underline{A}}_{\Pi\Pi} \right)^{\sim 1} \left( \underline{\underline{w}}_1 - \underline{\underline{A}}_{\Pi\pi} \underline{\underline{v}}_{\pi} \right) \quad (11.7)$$

This completes the evaluation of  $\underline{\underline{S}}$ .

## B. Application of $\underline{\underline{S}}$

We define

$$\Sigma \equiv \mathbf{I} \cup \Delta \quad (11.8)$$

Properties that are relevant for the following discussions are:

$$W_r(\Sigma) = \underline{\underline{a}} W(\Sigma) = W(\Sigma) \quad (11.9)$$

and

$$\underline{\underline{A}} = \begin{pmatrix} \underline{\underline{A}}_{\Sigma\Sigma} & \underline{\underline{A}}_{\Sigma\pi} \\ \underline{\underline{A}}_{\pi\Sigma} & \underline{\underline{A}}_{\pi\pi} \end{pmatrix} = \begin{pmatrix} \underline{\underline{A}}'_{\Sigma\Sigma} & \underline{\underline{A}}'_{\Sigma\pi} \underline{\underline{a}} \\ \underline{\underline{a}} \underline{\underline{A}}'_{\pi\Sigma} & \underline{\underline{a}} \underline{\underline{A}}'_{\pi\pi} \underline{\underline{a}} \end{pmatrix} \quad (11.10)$$

On the other hand, the matrix  $\underline{\underline{A}}^{-1}$  can be written as:

$$\underline{\underline{A}}^{-1} = \begin{pmatrix} \left( \underline{\underline{A}}^{-1} \right)_{\Pi\Pi} & \left( \underline{\underline{A}}^{-1} \right)_{\Pi\Delta} \\ \left( \underline{\underline{A}}^{-1} \right)_{\Delta\Pi} & \left( \underline{\underline{A}}^{-1} \right)_{\Delta\Delta} \end{pmatrix} = \begin{pmatrix} \left( \underline{\underline{A}}^{-1} \right)_{\Sigma\Sigma} & \left( \underline{\underline{A}}^{-1} \right)_{\Sigma\pi} \\ \left( \underline{\underline{A}}^{-1} \right)_{\pi\Sigma} & \left( \underline{\underline{A}}^{-1} \right)_{\pi\pi} \end{pmatrix} \quad (11.11)$$

Then,  $\underline{\underline{S}}^{-1} : W_\Delta \rightarrow W_\Delta$  fulfills

$$\underline{\underline{S}}^{-1} = \left( \underline{\underline{A}}^{-1} \right)_{\Delta\Delta} \quad (11.12)$$

For any  $\underline{w} \in W_\Delta$  (hence,  $\underline{w}_\Pi = \underline{w}_\pi = 0$ ), let us write

$$\underline{v} \equiv \underline{\underline{A}}^{-1} \underline{w} \quad (11.13)$$

Then,

$$\underline{v}_\Delta = \left( \underline{\underline{A}}^{-1} \underline{w} \right)_\Delta = \underline{\underline{S}}^{-1} \underline{w} \quad (11.14)$$

and  $\underline{v}_\pi$  fulfills

$$\sigma_{\pi\pi} \left( \underline{\underline{A}} \right) \underline{v}_\pi = \underline{w}_\pi - \underline{\underline{A}}_{\pi\Sigma} \left( \underline{\underline{A}}'_{\Sigma\Sigma} \right)^{-1} \underline{w}_\Sigma = -\underline{\underline{a}} \underline{\underline{A}}'_{\pi\Sigma} \left( \underline{\underline{A}}'_{\Sigma\Sigma} \right)^{-1} \underline{w}_\Delta, \quad \text{subjected to} \quad \underline{j} \underline{v}_\pi = 0 \quad (11.15)$$

Where:

$$\sigma_{\pi\pi} \left( \underline{\underline{A}} \right) \equiv \underline{\underline{a}} \left( \underline{\underline{A}}'_{\pi\pi} - \underline{\underline{A}}'_{\pi\Sigma} \left( \underline{\underline{A}}'_{\Sigma\Sigma} \right)^{-1} \underline{\underline{A}}'_{\Sigma\pi} \right) \underline{\underline{a}} \quad (11.16)$$

Furthermore, we observe that

$$\left( \underline{\underline{A}}'_{\Sigma\Sigma} \right)^{-1} = \sum_{\alpha=1}^E \left( \underline{\underline{A}}^\alpha_{\Sigma\Sigma} \right)^{-1} \quad (11.17)$$

Once  $\underline{v}_\pi$  has been obtained iteratively, we apply:

$$\underline{v}_\Sigma = \left( \underline{A}'_{\Sigma\Sigma} \right)^{-1} \left( \underline{w}_\Sigma - \underline{A}'_{\Sigma\pi} \underline{a} \underline{v}_\pi \right) \quad (11.18)$$

The operations here involved can be fully parallelized by virtue of Eq. (11.17). Then, the vector  $\underline{S}^{-1} \underline{w}$  that we are seeking for is supplied by Eq. (11.18) because  $\underline{v}_\Sigma = \underline{v}_\Gamma + \underline{v}_\Delta$  and

$$\underline{S}^{-1} \underline{w} = \underline{v}_\Delta \quad (11.19)$$

### C. Application of $\underline{a}$ and $\underline{j}$

Writing:  $\underline{v} \equiv \underline{a} \underline{w}$ , then

$$\underline{v}(p, \alpha) = \frac{1}{m(p)} \sum_{\beta \in Z(p)} \underline{w}(p, \beta) \quad (11.20)$$

Once  $\underline{a} \underline{w}$  has been obtained, the relation  $\underline{j} \underline{w} = \underline{w} - \underline{a} \underline{w}$  is used. We observe that the application of Eq. (11.20) requires exchange between a number of processors equal to the multiplicity  $m(p)$ . However, this is the only place where exchange of information between different processors is required and, furthermore, generally  $m(p)$  is small; for example, when an orthogonal mesh is used its maximum value is 4 in two-dimensional (2D) problems, and 8 in 3D problems.

## XII. NUMERICAL RESULTS

In previous publications it was verified, through certain number of numerical examples treated, that the DVS-algorithms possess up to date numerical efficiencies [5, 13, 14]. Some additional numerical work is presented in this article and some of their parallelization properties are also exhibited.

All the BVPs treated have the form:

$$\begin{aligned} -a \nabla^2 u + \underline{b} \cdot \nabla u + cu &= f(x) \quad x \in \Omega \\ u &= g(x) \quad x \in \partial\Omega \\ \Omega &= \prod_{i=1}^d (\alpha_i, \beta_i) \end{aligned} \quad (12.1)$$

where  $a > 0$  and  $c \geq 0$  are real-valued constants, whereas  $\underline{b} = (b_1, \dots, b_n)$  is a constant-vector. Details of the discretizations used are given in previous publications [5, 13]. Both the fine and coarse-meshes are constituted by orthogonal parallelepipeds. When  $\underline{b} = 0$ , these differential equations are symmetric, otherwise they are nonsymmetric. In [14], 2D and 3D examples were treated, and Tables I and II reproduce such results for 3D problems only. Table III shows the results obtained here for the case when the differential operator is Laplace's (Poisson equation). Helmholtz equation was also treated here, which corresponds to  $a = 1$ ,  $c = -1$ , and  $\underline{b} = 0$ , above. In spite that this is an indefinite equation, which generally is considered a complicated problem, the algorithms works as well as for the other equations and the results are shown in Table IV. All the DVS-algorithms are preconditioned and constrained; the constraints used are continuity at primal nodes. Such nodes have been chosen according to Algorithm "D" of Toselli and Widlund (p. 173 of [15]). The CGM algorithm [16] was used for the iterative solution of symmetric

TABLE I. Symmetric and positive definite example 3-D Eps  $1e^{-6}$ .

Partition	Subdomains	Dof	Primals	DVS-BDDC	DVS-Primal	DVS-FETI-DP	DVS-Dual
$(2 \times 2 \times 2) \times (2 \times 2 \times 2)$	8	27	7	2	2	2	2
$(3 \times 3 \times 3) \times (3 \times 3 \times 3)$	27	512	80	4	4	3	3
$(4 \times 4 \times 4) \times (4 \times 4 \times 4)$	64	3375	351	5	5	4	3
$(5 \times 5 \times 5) \times (5 \times 5 \times 5)$	125	13824	1024	6	5	4	3
$(6 \times 6 \times 6) \times (6 \times 6 \times 6)$	216	42875	2375	6	6	4	4
$(7 \times 7 \times 7) \times (7 \times 7 \times 7)$	363	110592	4752	7	6	4	4
$(8 \times 8 \times 8) \times (8 \times 8 \times 8)$	512	250047	8575	8	7	5	6
$(9 \times 9 \times 9) \times (9 \times 9 \times 9)$	729	512000	14336	8	8	7	7
$(10 \times 10 \times 10) \times (10 \times 10 \times 10)$	1000	970299	22599	8	8	8	8

TABLE II. Non-symmetric example 3-D Eps  $1e^{-6}$ .

Partition	Subdomains	Dof	Primals	DVS-BDDC	DVSPrimal	DVS-FETI-DP	DVS-Dual
$(2 \times 2 \times 2) \times (2 \times 2 \times 2)$	8	27	7	3	2	2	2
$(3 \times 3 \times 3) \times (3 \times 3 \times 3)$	27	512	80	6	4	4	4
$(4 \times 4 \times 4) \times (4 \times 4 \times 4)$	64	3375	351	7	6	5	5
$(5 \times 5 \times 5) \times (5 \times 5 \times 5)$	125	13824	1024	8	7	5	5
$(6 \times 6 \times 6) \times (6 \times 6 \times 6)$	216	42875	2375	10	7	6	6
$(7 \times 7 \times 7) \times (7 \times 7 \times 7)$	363	110592	4752	11	8	6	6
$(8 \times 8 \times 8) \times (8 \times 8 \times 8)$	512	250047	8575	11	9	7	7
$(9 \times 9 \times 9) \times (9 \times 9 \times 9)$	729	512000	14336	12	10	8	8
$(10 \times 10 \times 10) \times (10 \times 10 \times 10)$	1000	970299	22599	13	11	9	9

TABLE III. Poisson equation example 3-D Eps  $1e^{-6}$ .

Partition	Subdomains	Dof	Primals	DVS-BDDC	DVS-Primal	DVS-FETI-DP	DVS-Dual
$(2 \times 2 \times 2) \times (2 \times 2 \times 2)$	8	27	7	1	1	1	1
$(3 \times 3 \times 3) \times (3 \times 3 \times 3)$	27	512	80	3	2	2	2
$(4 \times 4 \times 4) \times (4 \times 4 \times 4)$	64	3375	351	1	1	2	2
$(5 \times 5 \times 5) \times (5 \times 5 \times 5)$	125	13824	1024	5	4	5	4
$(6 \times 6 \times 6) \times (6 \times 6 \times 6)$	216	42875	2375	6	6	5	6
$(7 \times 7 \times 7) \times (7 \times 7 \times 7)$	363	110592	4752	6	6	6	6
$(8 \times 8 \times 8) \times (8 \times 8 \times 8)$	512	250047	8575	7	7	8	8
$(9 \times 9 \times 9) \times (9 \times 9 \times 9)$	729	512000	14336	8	9	9	9
$(10 \times 10 \times 10) \times (10 \times 10 \times 10)$	1000	970299	22599	10	10	10	10

TABLE IV. Indefinite example 3-D (Helmholtz equation)  $\text{Eps } 1e^{-6}$ .

Partition	Subdomains	Dof	Primals	DVS-BDDC	DVS-Primal	DVS-FETI-DP	DVS-Dual
$(4 \times 4 \times 4) \times (4 \times 4 \times 4)$	64	3375	351	5	4	4	3
$(5 \times 5 \times 5) \times (5 \times 5 \times 5)$	125	13824	1024	6	6	5	5
$(6 \times 6 \times 6) \times (6 \times 6 \times 6)$	216	42875	2375	7	7	6	5
$(7 \times 7 \times 7) \times (7 \times 7 \times 7)$	363	110592	4752	7	7	6	5
$(8 \times 8 \times 8) \times (8 \times 8 \times 8)$	512	250047	8575	8	8	6	5
$(9 \times 9 \times 9) \times (9 \times 9 \times 9)$	729	512000	14336	8	8	6	6
$(10 \times 10 \times 10) \times (10 \times 10 \times 10)$	1000	970299	22599	9	6	6	6



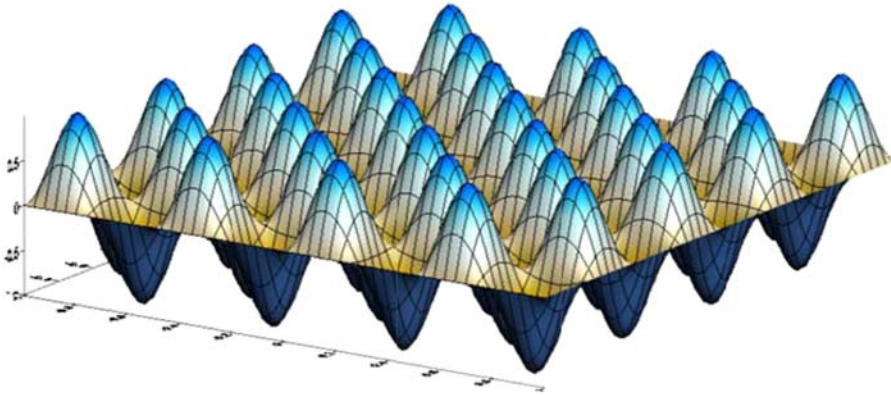


FIG. 11. The numerical solution for the 2D case, when  $n=4$ . [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

positive-definite cases; in all other cases, the DQGMRES algorithm was used. All the codes that were developed to treat the examples were written in C++ language, using the MPI library for the communications on the Master-Slave scheme that was used.

In the tables that follow, each line corresponds to a different run of the software we developed. The first column indicates the fine-mesh that was used in each one of them. The second column indicates the corresponding coarse-mesh. Third column to total number of degrees freedom. The fourth one indicates the total number of primal nodes. The fifth column the number of iterations required for convergence, when the DVS-BDDC was applied. The following columns give, successively, such a number for the cases when DVS-PRIMAL, DVS-FETI-DP, and DVS-DUAL were used.

To exhibit the parallelization efficiency of the DVS-algorithms Eq. (12.1), with  $\underline{b} = 0$ ,  $c=0$ , and  $a=1$  (Poisson equation) was treated. The analytical solution of this example is  $u(x, y) = \sin(n\pi x) \sin(n\pi y)$  can be seen in Fig. 11.

The parallelization properties of the DVS-PRIMAL algorithm are illustrated in Table V. This example was run in the Kan-Balam cluster facility of the National University of Mexico (UNAM) using up to 512 cores. To measure the parallelization efficiency, the metric used was based on the relative speed up, defined as  $S_p^{p'} = \frac{T_p}{T_p^{p'}}$  and then the relative efficiency is given by  $E_p^{p'} = \frac{p'}{p} S_p^{p'} = \frac{p'}{p} \frac{T_p}{T_p^{p'}}$ . We can see from Table V that the DVS-PRIMAL algorithm is more efficient as the number of subdomains and the number of degrees of freedom (dof) increases from 23,017,500 to 63,937,500. For this latter number of degrees of freedom, the Relative Efficiency attains the value one, which is characteristic of 100% parallelization. This is due to the increased

TABLE V. Speed up and efficiency made by the DVS-PRIMAL algorithm. The primal nodes were located at the vertices of subdomains.

Decomposition	dof	Time $T_p/T_p^{p'}$	Speed Up	Efficiency
$31 \times 33$ and $150 \times 150$	23,017,500	7315/1541	$S_{512}^{32} = 4.7$	$E_{512}^{32} = 0.29$
$31 \times 33$ and $200 \times 200$	40,920,000	16,037/2688	$S_{512}^{64} = 5.9$	$E_{512}^{64} = 0.74$
$31 \times 33$ and $250 \times 250$	63,937,500	26,587/6388	$S_{512}^{128} = 4$	$E_{512}^{128} = 1$

load on the cores, while the communication-time remains small because the DVS-algorithms achieve the DDM-paradigm; essentially, the ratio of the communication-time to processing-time is negligible.

### XIII. CONCLUSIONS

Discretization procedures for partial differential equations of a new kind, the nonoverlapping discretization methods, have been introduced, whose distinguishing feature is that they use a system of nodes with the property that each one of them belongs to one and only one subdomain of the domain decomposition. Due to this latter fact, they are highly parallelizable. Four algorithms, whose numerical and computational properties similar, have been introduced as products of the use of nonoverlapping discretizations; two of them can be derived mimicking the BDDC and FETI-DP algorithms in this setting, whereas the other two are unrelated.

The main conclusions are:

1. Using the nonoverlapping discretization methods, algorithms that achieve the DDM-paradigm can be developed; and
2. Algorithms that work equally well for symmetric, nonsymmetric, and indefinite (i.e., neither positive nor negative definite) matrices have been developed

These conclusions have been documented through numerical and computational examples. The procedures that achieve the DDM-paradigm have been shown and explained in Section XI. Other properties that should be highlighted are:

- A. The DVS-algorithms equally applicable to a single differential equation or to a system of such equations;
- B. They are directly applied to the system of equations that is obtained after the BVP has been discretized;
- C. There is considerable freedom in the choice of the local solvers to be used;
- D. Robust codes have been developed that with slight modifications have been applied to 2D and 3D problems, originating in a single or on a system of equations;

### APPENDIX

Here, as in Section V, we consider matrices, denoted generically with the symbol  $\underline{\underline{B}}$ , which correspond to transformations of the whole  $W$  into itself. To define a pseudoinverse for this class of matrices, temporarily we adopt the notation  $E \equiv N(\underline{\underline{B}})^\perp$  and  $F \equiv \underline{\underline{B}} \left\{ N(\underline{\underline{B}})^\perp \right\}$ . Furthermore, we define the function  $f : E \rightarrow F$  by

$$f(\underline{w}) \equiv \underline{\underline{B}} \underline{w}, \quad \forall \underline{w} \in E \equiv N(\underline{\underline{B}})^\perp \quad (\text{A1})$$

We observe that the function  $f : E \rightarrow F$ , so defined, is one-to-one and therefore possesses an inverse:  $f^{-1} : F \rightarrow E$ .

**Definition A1** (Pseudoinverse). For every  $\underline{u} \in W$ , we define the “pseudo-inverse,”  $\underline{B}^{\sim 1}$  of  $\underline{B}$ , as follows:

$$\underline{B}^{\sim 1} \underline{u} \equiv f^{-1}(\text{proj}_F \underline{u}), \quad \forall \underline{u} \in W \quad (\text{A2})$$

In this manner,  $\underline{B}^{\sim 1} \underline{u}$  is well-defined, because  $\text{proj}_F \underline{u} \in F \equiv \underline{B} \left\{ \mathcal{N}(\underline{B})^\perp \right\}$ . The following two properties should be noticed:

$$\underline{B} = \text{proj}_F \underline{B} \text{proj}_E \quad (\text{A3})$$

and

$$\underline{B}^{\sim 1} = \text{proj}_E \underline{B}^{\sim 1} \text{proj}_F \quad (\text{A4})$$

**Theorem A.1.** Given a matrix  $\underline{B}$ , for every  $\underline{f} \in \underline{B}(W)$ , the problem: “Find a  $\underline{u} \in W$  such that

$$\underline{B} \underline{u} = \underline{f}, \quad \text{subjected to} \quad \underline{j}_B \underline{u} = 0 \quad (\text{A5})$$

possesses a unique solution.” Here,

$$\underline{j}_B \equiv \underline{I} - \underline{a}_B, \quad \text{with} \quad \underline{a}_B = \text{proj}_E \quad (\text{A6})$$

We recall that  $E \equiv \mathcal{N}(\underline{B})^\perp$ .

**Definition A2.** Given any two sets of derived-nodes,  $M \subset X$  and  $N \subset X$ , we define

$$\underline{B}_{NM} \equiv \text{proj}_H \underline{B} \text{proj}_G \quad (\text{A7})$$

where,

$$G \equiv W(M) \quad \text{and} \quad H \equiv W(N) \quad (\text{A8})$$

When

$$X = M \cup N \quad \text{and} \quad \emptyset = M \cap N \quad (\text{A9})$$

Then:

$$W = W(M) \oplus W(N) \quad \text{and} \quad W(M) \perp W(N) \quad (\text{A10})$$

**Theorem A.2.** When Eq. (A10) holds, then:

1. Each  $\underline{u} \in W$  can be written uniquely as

$$\underline{u} = \underline{u}_M + \underline{u}_N, \quad \underline{u}_M \in W(M) \quad \text{and} \quad \underline{u}_N \in W(N) \quad (\text{A11})$$

2. Furthermore

$$\underline{B} \underline{u} = \underline{B}_{MM} \underline{u}_M + \underline{B}_{NM} \underline{u}_M + \underline{B}_{MN} \underline{u}_N + \underline{B}_{NN} \underline{u}_N \quad (\text{A12})$$

and

$$\underline{\underline{B}} = \underline{\underline{B}}_{\text{MM}} + \underline{\underline{B}}_{\text{NM}} + \underline{\underline{B}}_{\text{MN}} + \underline{\underline{B}}_{\text{NN}} \quad (\text{A13})$$

**Proof.** Equation (A11) is clear, by virtue of Eq. (A10). Using it, Eq. (A12) follows. Then, Eq. (A13) is straightforward. ■

## References

1. I. Herrera and G. F. Pinder, *Mathematical modelling in science and engineering: an axiomatic approach*, Wiley, Hoboken, N.J., 2012, p. 243.
2. President's Information Technology Advisory Committee: PITAC, *Computational Science: Ensuring America's Competitiveness*, Report to the President, June 2005, P. 104. Available at: [www.nitrd.gov/pitac](http://www.nitrd.gov/pitac).
3. DDM Organization, *Proceedings of 21 International Conferences on Domain Decomposition Methods*, Available at: [www.ddm.org](http://www.ddm.org), accessed on 2012.
4. I. Herrera and A. Rosas-Medina The derived-vector space framework and four general purposes massively parallel DDM algorithms, *Eng Anal Bound Elem* 37 (2013), 646–657.
5. I. Herrera, A. Carrillo-Ledesma and A. Rosas-Medina A brief overview of non-overlapping domain decomposition methods, *Geofis Int* 50 (2011), 445–463.
6. C. R. Dohrmann, A preconditioner for substructuring based on constrained energy minimization, *SIAM J Sci Comput* 25 (2003), 246–258.
7. J. Mandel and C. R. Dohrmann, Convergence of a balancing domain decomposition by constraints and energy minimization, *Numer Linear Algebra Appl* 10 (2003), 639–659.
8. J. Mandel, C. R. Dohrmann, and R. Tezaur, An algebraic theory for primal and dual substructuring methods by constraints, *Appl Numer Math* 54 (2005), 167–193.
9. C. Farhat, and F. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, *Int J Numer Methods Eng* 32 (1991), 1205–1227.
10. J. Mandel and R. Tezaur, Convergence of a substructuring method with Lagrange multipliers, *Numer Math* 73 (1996), 473–487.
11. C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen, FETI-DP a dual-primal unified FETI method, Part I: a faster alternative to the two-level FETI method. *Int J Numer Methods Eng* 50 (2001), 1523–1544.
12. C. Farhat, M. Lesoinne, and K. Pierson, A scalable dual-primal domain decomposition method, *Numer Linear Algebra Appl* 7 (2000), 687–714.
13. A. Carrillo-Ledesma, I. Herrera, and L. M. de la Cruz, Parallel algorithms for computational models of geophysical systems, *Geofis Int*, to appear.
14. I. Herrera, and R. A. Yates, The multipliers-free dual primal domain decomposition methods for non-symmetric matrices, *Numer Meth Partial Differential Equation* 27 (2011), 1262–1289, 2011. (Published on line April 28, 2010) DOI 10.1002/Num. 20581.
15. A. Toselli and O. Widlund, *Domain decomposition methods: algorithms and theory*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005, p. 450.
16. Y. Saad, *Iterative methods for sparse linear systems*, SIAM, Philadelphia, 2003, p. 447.