Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 192 (2003) 3799–3830
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

The mosaic of high performance domain Decomposition Methods for Structural Mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods

Yannis Fragakis, Manolis Papadrakakis *

Institute of Structural Analysis and Seismic Research, National Technical University of Athens, 9, Iroon Polytechniou, Zografou Campus, GR-15780 Athens, Greece

Received 20 June 2002; received in revised form 28 December 2002; accepted 19 May 2003

Abstract

A multitude of domain decomposition methods (DDM) for structural mechanics is available in the literature today. A unified framework for formulating primal and dual DDM is thus presented in this paper, aiming at providing a mathematical platform for a uniform treatment of high performance DDM in structural mechanics. A novel approach for developing new DDM from existing methods is also proposed and is applied to dual and primal methods. In the field of the FETI methods, this approach leads to a new category of methods derived from existing FETI variants. Furthermore, two alternative formulations of the balancing domain decomposition method are described, while interrelations between the introduced and existing methods are established. Finally, comparative numerical tests demonstrate the differences in the computational performance of the methods in question.

1. Introduction

In the last two decades, extensive research efforts have been devoted to the development of efficient solution methods for finite element simulations and have led to the development of several high performance solvers. Domain decomposition methods (DDM) constitute today an important category of methods for the solution of a variety of problems in computational mechanics. Their performance in both serial and parallel computer environments is demonstrated in a number of papers and conferences specialized in DDM over the last decade. In fact, DDM have gradually become the main field of interest of a large number of researchers and have started making their way into high performance commercial software applications.

* Corresponding author. Tel.: +30-1-7721694; fax: +30-1-7721693.
E-mail addresses: fragayan@central.ntua.gr (Y. Fragakis), mpapadra@central.ntua.gr (M. Papadrakakis).

0045-7825/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0045-7825(03)00374-8

In structural mechanics, the most popular DDM can be divided in two major categories, the primal and dual substructuring methods. In the early 90s, an important dual DDM, the finite element tearing and interconnecting (FETI) method was introduced by Farhat and Roux [1]. Since their introduction, FETI and several variant methods have gained a high importance and today are considered as highly efficient DDM. Furthermore, an important family of primal DDM are considered to be the balancing domain decomposition (BDD) methods, introduced by Mandel [2]. Research in BDD and FETI methods has been progressing in parallel since their introduction and both methods are currently capable of successfully addressing a large number of complex problems in computational mechanics.

Today, a number of DDM versions exist in the literature capable of handling homogeneous or heterogeneous problems, that is problems with similar or different stiffness coefficients across the subdomain interfaces. Furthermore, dual DDM are often implemented with different definitions of the Lagrange multipliers which represent the interaction forces between the subdomains, thus leading to different formulations. Due to this variety of DDM versions proposed in the past, a large number of theoretical findings and implementation techniques that are available in the literature for both primal and dual methods, are not correlated to each other through a consistent formulation. For this reason, a unified framework for formulating both primal and dual DDM is presented in this paper, aiming at providing a mathematical tool for uniform treatment of the high performance DDM in structural mechanics. The formulation of standard DDM (like the primal substructuring method (PSM) and the FETI methods) is then revisited in the context of the proposed unified framework.

Furthermore, primal preconditioners are proposed in this paper by implementing an idea that allows the construction of preconditioners for PSM, based on the theory of DDM that incorporate a coarse problem, like for example the FETI methods. A primal category of methods is thus created from the FETI methods. The relations between the new methods and the FETI methods are then investigated.

Following the development of PSM preconditioners based on existing FETI methods, two alternative formulations of the BDD are described. The relations between the introduced methods and existing DDM are then used to prove a connection between the BDD method and an instance of the FETI method. In addition, the newly introduced and previously known methods are compared in numerical tests. Many issues treated in this paper have also been included in [3,4].

The remainder of this paper is organized as follows. Section 2 presents a theoretical framework for mapping the displacements and forces of the subdomains on the global domain. Relations between mapping matrices that are extensively used in subsequent sections are also outlined. The treatment of local subdomain and interface problems is then described in Section 3. Subsequently, Sections 4 and 5 outline the basic theory of the PSM and FETI methods. The primal methods derived from FETI variants are then introduced in Section 6, followed by an analysis of their relations to existing FETI methods. Furthermore, Section 7 refers to the BDD method, while the relation between the BDD and FETI methods is treated in Section 8. Section 9 presents numerical tests performed in order to compare the discussed methods and concluding remarks are provided in Section 10.

2. Mapping subdomains on the global domain

This section presents a theoretical framework for mapping the displacements and forces of the subdomains on the global domain. The introduced framework treats uniformly the scaling of DDM for homogeneous or heterogeneous problems and for different Lagrange multiplier definitions. It has to be clarified that this section does not treat the mapping of subdomains from the point of view of computer implementation, but it presents the theory of a unified theoretical framework for subdomain mapping from the point of view of mathematical formulation of the DDM. In particular, different choices that have been used in the past in the formulation of DDM in structural mechanics are integrated in generalized equations.

We then state previously known as well as new relations, which connect the mapping matrices that are used in this section. The presented relations are used extensively in subsequent sections of this paper.

2.1. Mapping displacements and applied loads of the subdomains

If u and f are the displacement and applied loads vectors of the global domain and $u^{(s)}$ and $f^{(s)}$ are the corresponding vectors of the subdomains $s = 1, \ldots, N_s$, then these global and local subdomain variables are connected with the relations

$$u^s = \left[u^{(1)^{\mathrm{T}}} \quad \cdots \quad u^{(N_s)^{\mathrm{T}}} \right]^{\mathrm{T}} = Lu, \tag{1}$$

$$f = L^{\mathrm{T}} f^s = L^{\mathrm{T}} \left[f^{(1)^{\mathrm{T}}} \quad \cdots \quad f^{(N_s)^{\mathrm{T}}} \right]^{\mathrm{T}}, \tag{2}$$

where L is a Boolean mapping matrix that is usually referred to as a global to local map. Restricting Eqs. (1) and (2) to the d.o.f. that belong to the interface between subdomains, we obtain the equations

$$u_{\mathrm{b}}^s = \left[u_{\mathrm{b}}^{(1)^{\mathrm{T}}} \quad \cdots \quad u_{\mathrm{b}}^{(N_s)^{\mathrm{T}}} \right]^{\mathrm{T}} = L_{\mathrm{b}} u_{\mathrm{b}}, \tag{3}$$

$$f_{\mathrm{b}} = L_{\mathrm{b}}^{\mathrm{T}} f_{\mathrm{b}}^s = L_{\mathrm{b}}^{\mathrm{T}} \left[f_{\mathrm{b}}^{(1)^{\mathrm{T}}} \quad \cdots \quad f_{\mathrm{b}}^{(N_s)^{\mathrm{T}}} \right]^{\mathrm{T}}, \tag{4}$$

where subscript b denotes the restriction of a matrix or vector to d.o.f. that belong to the boundaries between subdomains.

In DDM it is often needed to split an interface force vector f_{b} between the subdomains, or to estimate an interface displacement vector u_{b} from different subdomain values u_{b}^s. In these operations it is favorable to take into account the possible heterogeneity between the subdomains. Such operations are often performed within the preconditioning step of DDM and may be expressed by the formulas

$$u_{\mathrm{b}_{\mathrm{estimate}}} = L_{\mathrm{p_b}}^{\mathrm{T}} u_{\mathrm{b}}^s, \tag{5}$$

$$f_{\mathrm{b}_{\mathrm{estimate}}}^s = L_{\mathrm{p_b}} f_{\mathrm{b}}. \tag{6}$$

In Eqs. (5) and (6), if the underlying problem is homogeneous, $L_{\mathrm{p_b}}$ is set equal to

$$L_{\mathrm{p_b}} = L_{\mathrm{b}} M_{\mathrm{b}}^{-1}, \tag{7}$$

where M_{b} denotes a diagonal scaling matrix with each diagonal entry being equal to the multiplicity of the corresponding d.o.f., which is equivalent to the number of its neighboring subdomains. For heterogeneous problems, however, it is more appropriate to scale matrix $L_{\mathrm{p_b}}$ with respect to the subdomain stiffness coefficients. In this case, matrix $L_{\mathrm{p_b}}$ takes the form

$$L_{\mathrm{p_b}} = D_{\mathrm{b}}^s L_{\mathrm{b}} (L_{\mathrm{b}}^{\mathrm{T}} D_{\mathrm{b}}^s L_{\mathrm{b}})^{-1} \tag{8}$$

with D_{b}^s denoting a diagonal matrix where each diagonal entry is equal to the diagonal element of the subdomain stiffness matrix of the corresponding d.o.f. It should be noted that any matrix $L_{\mathrm{p_b}}$ of Eqs. (7) and (8) may in fact be used for both homogeneous and heterogeneous problems. The distinction between homogeneous and heterogeneous is made here, because $L_{\mathrm{p_b}}$ of Eq. (7) is best suited for homogeneous problems, while $L_{\mathrm{p_b}}$ of Eq. (8) is best suited for heterogeneous problems. Therefore, in the rest of this paper whenever homogeneous or heterogeneous cases are examined, the most suitable scaling operator is implicitly assumed.

Similarly to Eqs. (1)–(4), Eqs. (5)–(8) may be expanded to include internal d.o.f. of the subdomains

$$u_{\mathrm{estimate}} = L_{\mathrm{p}}^{\mathrm{T}} u^s, \tag{9}$$

$$f^s_{\text{estimate}} = L_{\text{p}} f, \tag{10}$$

where for homogeneous cases L_{p} is given by

$$L_{\text{p}} = LM^{-1} \tag{11}$$

and for heterogeneous cases

$$L_{\text{p}} = D^s L (L^{\text{T}} D^s L)^{-1}. \tag{12}$$

In Eqs. (11) and (12), matrices M and D^s are formed similar to matrices M_{b} and D^s_{b}, but they are now expanded to include both the interface and the internal d.o.f. of the subdomains.

2.2. Mapping Lagrange multipliers on the subdomains

In a number of DDM, as for example in FETI methods, it is required to define the nodal interaction forces between the subdomains. These interface tractions are also used to enforce the interface displacement compatibility between subdomains and are commonly referred to as Lagrange multipliers in DDM literature. Among infinite possible ways to define the Lagrange multipliers, we note the non-redundant, redundant and orthonormal choices which have been used in the FETI methods [5–7]. These three different choices are depicted in Fig. 1 for a node belonging to three subdomains. For each d.o.f. of the node of Fig. 1, the subdomain forces $f^s_{\lambda_{\text{b}}}$ which are due to the Lagrange multipliers λ, are given by:

$$\text{For non-redundant multipliers:} \quad f^s_{\lambda_{\text{b}}} = \begin{bmatrix} f^{(1)}_{\lambda_{\text{b}}} \\ f^{(2)}_{\lambda_{\text{b}}} \\ f^{(3)}_{\lambda_{\text{b}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = B^{\text{T}}_{\text{b}} \lambda. \tag{13}$$

$$\text{For redundant multipliers:} \quad f^s_{\lambda_{\text{b}}} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = B^{\text{T}}_{\text{b}} \lambda. \tag{14}$$

$$\text{For orthonormal multipliers:} \quad f^s_{\lambda_{\text{b}}} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{6} \\ -1/\sqrt{2} & -1/\sqrt{6} \\ 0 & 2/\sqrt{6} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = B^{\text{T}}_{\text{b}} \lambda. \tag{15}$$

Note that in the orthonormal case, B_{b} satisfies $B_{\text{b}} B^{\text{T}}_{\text{b}} = I$. It should be further noted that the subdomain interaction forces can also be mapped using a three-field formulation, which has been proven equivalent to the orthonormal choice of Lagrange multipliers [6,7] and to the redundant choice [8].

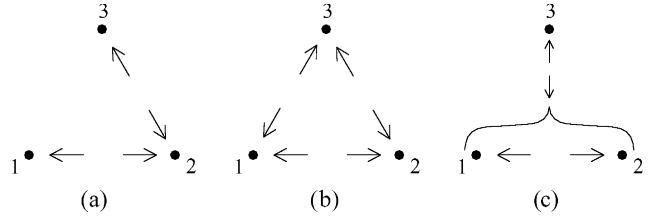

Fig. 1. Different definitions of Lagrange multipliers for a node belonging to three subdomains: (a) non-redundant, (b) redundant and (c) orthonormal.

Given a Lagrange multiplier vector λ, the applied forces on the interface nodes of the disconnected subdomains may be expressed as $f_{\mathrm{b}}^{s} - B_{\mathrm{b}}^{\mathrm{T}}\lambda$. If we also include the internal d.o.f. of the subdomains, the applied forces take the form $f^{s} - B^{\mathrm{T}}\lambda$, where matrix B_{b} has been expanded to B. The B and B_{b} mapping matrices may also be used to express the displacement compatibility condition at subdomain interfaces as

$$B_{\mathrm{b}}u_{\mathrm{b}}^{s} = \begin{bmatrix} B_{\mathrm{b}}^{(1)} & \cdots & B_{\mathrm{b}}^{(N_s)} \end{bmatrix} \begin{bmatrix} u_{\mathrm{b}}^{(1)} \\ \vdots \\ u_{\mathrm{b}}^{(N_s)} \end{bmatrix} = 0 \tag{16}$$

or

$$Bu^{s} = \begin{bmatrix} B^{(1)} & \cdots & B^{(N_s)} \end{bmatrix} \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(N_s)} \end{bmatrix} = 0. \tag{17}$$

In the past, a number of modified versions of B_{b} that incorporate scaling effects for homogeneous or heterogeneous problems, have been used in preconditioning steps of DDM. Thus, for homogeneous problems, the following $B_{\mathrm{p_b}}$ matrices can be written, as modified variations of B_{b}:

(a) For non-redundant Lagrange multipliers: $B_{\mathrm{p_b}} = (B_{\mathrm{b}}B_{\mathrm{b}}^{\mathrm{T}})^{-1}B_{\mathrm{b}}$. (18)

(b) For redundant Lagrange multipliers: $B_{\mathrm{p_b}} = B_{\mathrm{b}}M_{\mathrm{b}}^{s^{-1}}$. (19)

(c) For orthonormal Lagrange multipliers: $B_{\mathrm{p_b}} = B_{\mathrm{b}}$. (20)

For heterogeneous problems the corresponding matrices are the following:

(a) For non-redundant or orthonormal Lagrange multipliers: $B_{\mathrm{p_b}} = (B_{\mathrm{b}}D_{\mathrm{b}}^{s^{-1}}B_{\mathrm{b}}^{\mathrm{T}})^{-1}B_{\mathrm{b}}D_{\mathrm{b}}^{s^{-1}}$. (21)

(b) For redundant Lagrange multipliers: $B_{\mathrm{p_b}} = \hat{D}_{\lambda}B_{\mathrm{b}}D_{\mathrm{b}}^{s^{-1}}$. (22)

In Eq. (19), M_{b}^{s} is a diagonal matrix that stores in each diagonal entry the multiplicity of the corresponding d.o.f. Matrix $\hat{D}_{\lambda}$ of Eq. (22) is also diagonal with each diagonal element being equal to the product of the two respective diagonal elements of the subdomain stiffness matrices of the d.o.f. that are attached to the corresponding Lagrange multiplier, divided by the diagonal element of the corresponding d.o.f. of matrix $L_{\mathrm{b}}^{\mathrm{T}}D_{\mathrm{b}}^{s}L_{\mathrm{b}}$. For example, the diagonal entry of $\hat{D}_{\lambda}$ related to the first Lagrange multiplier defined in Eq. (14) is equal to $k_{11}k_{22}/(k_{11}+k_{22}+k_{33})$, where k_{ii} denotes the diagonal element of the subdomain stiffness matrix of the corresponding d.o.f. Matrices $B_{\mathrm{p_b}}$ of Eqs. (19) and (22) are used in the FETI methods with redundant Lagrange multipliers [8–10], while matrix $B_{\mathrm{p_b}}$ of Eq. (21) was first used in a FETI method for non-redundant Lagrange multipliers in [11]. Finally, we note that the subscript p is used in $B_{\mathrm{p_b}}$ and in other matrices of this section, in order to denote that these matrices are usually used in preconditioning steps.

Furthermore, Eqs. (18)–(22) can be expanded to include both internal and interface d.o.f. of the subdomains. The corresponding expressions for matrix B_{p} are obtained by simply removing subscript b from all matrices in these equations.

2.3. Connections between subdomain mapping matrices

In this subsection, we present a number of relations connecting the subdomain mapping matrices introduced in the previous sections. The displacement and Lagrange multiplier mapping matrices of the previous sections satisfy the relations

$$\text{range}(L_{\text{b}}) = \text{null}(B_{\text{b}}) \quad \text{and} \quad \text{range}(L_{\text{p}_{\text{b}}}) = \text{null}(B_{\text{p}_{\text{b}}}), \tag{23}$$

$$B_{\text{b}}L_{\text{b}} = 0 \quad \text{and} \quad B_{\text{p}_{\text{b}}}L_{\text{p}_{\text{b}}} = 0, \tag{24}$$

$$L_{\text{b}}^{\text{T}}L_{\text{p}_{\text{b}}} = I, \tag{25}$$

$$(L_{\text{p}_{\text{b}}}L_{\text{b}}^{\text{T}})^2 = L_{\text{p}_{\text{b}}}L_{\text{b}}^{\text{T}} \quad \text{and} \quad (B_{\text{b}}^{\text{T}}B_{\text{p}_{\text{b}}})^2 = B_{\text{b}}^{\text{T}}B_{\text{p}_{\text{b}}}, \tag{26}$$

$$L_{\text{p}_{\text{b}}}L_{\text{b}}^{\text{T}} + B_{\text{b}}^{\text{T}}B_{\text{p}_{\text{b}}} = I. \tag{27}$$

Eqs. (23)–(27) hold for any of the discussed definitions of Lagrange multipliers and for homogeneous or heterogeneous problems. To be more precise, Eqs. (23)–(27) hold in the following cases:

(a) Case of homogeneous scaling: $L_{\text{p}_{\text{b}}}$ is set equal to Eq. (7) and $B_{\text{p}_{\text{b}}}$ is selected equal to any of Eqs. (18)–(20), and
(b) Case of heterogeneous scaling: $L_{\text{p}_{\text{b}}}$ is set equal to Eq. (8) any $B_{\text{p}_{\text{b}}}$ is selected equal to any of Eqs. (21) and (22).

The proofs of the above equations are omitted for the sake of brevity. These proofs are in fact similar to those provided in [8] for redundant Lagrange multipliers and homogeneous problems. Proofs for Eq. (27) can also be found in [11].

In contrast to Eqs. (23)–(27), the subsequent equations of this section hold for specific choices of Lagrange multipliers and homogeneous or heterogeneous scaling. For instance, relation

$$B_{\text{b}}B_{\text{p}_{\text{b}}}^{\text{T}} = I \tag{28}$$

holds for non-redundant or orthonormal Lagrange multipliers, while it does not hold in general for redundant multipliers.

Furthermore, in the case of homogeneous problems, it follows from Eqs. (7) and (27) that

$$B_{\text{b}}^{\text{T}}B_{\text{p}_{\text{b}}} = I - L_{\text{p}_{\text{b}}}L_{\text{b}}^{\text{T}} = I - L_{\text{b}}M_{\text{b}}^{-1}L_{\text{b}}^{\text{T}}. \tag{29}$$

In Eq. (29), matrix $I - L_{\text{b}}M_{\text{b}}^{-1}L_{\text{b}}^{\text{T}}$ is symmetric. Hence, in homogeneous problems both matrix $B_{\text{b}}^{\text{T}}B_{\text{p}_{\text{b}}}$ and matrix $L_{\text{p}_{\text{b}}}L_{\text{b}}^{\text{T}} = L_{\text{b}}M_{\text{b}}^{-1}L_{\text{b}}^{\text{T}}$ are symmetric, while in general this is not the case for heterogeneous problems. For homogeneous scaling, we thus have

$$L_{\text{p}_{\text{b}}}L_{\text{b}}^{\text{T}} = L_{\text{b}}L_{\text{p}_{\text{b}}}^{\text{T}} \quad \text{and} \quad B_{\text{b}}^{\text{T}}B_{\text{p}_{\text{b}}} = B_{\text{p}_{\text{b}}}^{\text{T}}B_{\text{b}}. \tag{30}$$

In further sections of this paper, we generally make use of Eqs. (23)–(27), which hold for all three definitions of Lagrange multipliers and homogeneous or heterogeneous problems. When however Eqs. (28) or (30) are used, we specify the particular version—homogeneous or heterogeneous scaling and Lagrange multiplier definition—for which the related mathematical expressions and numerical algorithms hold. It can easily be deduced that all equations of this subsection also hold after removal of subscript b.

3. Solving the local subdomain and interface problems

3.1. Solving the local subdomain problems

Several DDM, like the FETI methods, require the solution of local subdomain problems of the form

$$K^{(s)}u^{(s)} = f^{(s)} - B^{(s)^{\mathrm{T}}}\lambda, \quad s = 1, \ldots, N_s, \tag{31}$$

where $K^{(s)}$ is the stiffness matrix of subdomain s. Eq. (31) may be gathered to form one matrix equation using block diagonal notation

$$K^s u^s = f^s - B^{\mathrm{T}}\lambda \Longleftrightarrow \begin{bmatrix} K^{(1)} & & \\ & \ddots & \\ & & K^{(N_s)} \end{bmatrix} \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(N_s)} \end{bmatrix} = \begin{bmatrix} f^{(1)} \\ \vdots \\ f^{(N_s)} \end{bmatrix} - \begin{bmatrix} B^{(1)^{\mathrm{T}}} \\ \vdots \\ B^{(N_s)^{\mathrm{T}}} \end{bmatrix} \lambda. \tag{32}$$

The solution of Eq. (31) takes two different forms, depending on the presence or not of zero energy modes in the subdomains. In the case of a non-floating subdomain, that is a subdomain whose external constraints prevent all possible zero energy motion, the solution of Eq. (31) is

$$u^{(s)} = K^{(s)^{-1}}\left(f^{(s)} - B^{(s)^{\mathrm{T}}}\lambda\right). \tag{33}$$

On the other hand, in the more general case of a floating subdomain, Eq. (31) is solvable if the loads $f^{(s)} - B^{(s)^{\mathrm{T}}}\lambda$ are self-equilibrated, that is if

$$R^{(s)^{\mathrm{T}}}\left(f^{(s)} - B^{(s)^{\mathrm{T}}}\lambda\right) = 0, \tag{34}$$

where matrix $R^{(s)}$ stores in its columns the zero energy modes of subdomain s. If condition (34) holds, then the general solution of Eq. (31) is given by

$$u^{(s)} = K^{(s)^{+}}\left(f^{(s)} - B^{(s)^{\mathrm{T}}}\lambda\right) + R^{(s)}a^{(s)}, \tag{35}$$

where $K^{(s)^{+}}$ is a generalized inverse of $K^{(s)}$ and $\alpha^{(s)}$ is a vector of arbitrary entries that represent the amplitudes of the zero energy modes. Similarly to Eq. (31), Eqs. (33)–(35) may be written in block diagonal notation, as

$$R^{s^{\mathrm{T}}}\left(f^s - B^{\mathrm{T}}\lambda\right) = 0, \tag{36}$$

$$u^s = K^{s^{+}}\left(f^s - B^{\mathrm{T}}\lambda\right) + R^s a, \tag{37}$$

where

$$K^{s^{+}} = \begin{bmatrix} K^{(1)^{+}} & & \\ & \ddots & \\ & & K^{(N_s)^{+}} \end{bmatrix}, \quad R^s = \begin{bmatrix} R^{(1)} & & \\ & \ddots & \\ & & R^{(N_s)} \end{bmatrix} \quad \text{and} \quad a = \begin{bmatrix} a^{(1)} \\ \vdots \\ a^{(N_s)} \end{bmatrix}. \tag{38}$$

For non-floating subdomains, submatrices $K^{(s)^{+}}$ of $K^{s^{+}}$ are substituted by $K^{(s)^{-1}}$ and R^s and a are modified accordingly, to satisfy Eqs. (33) and (35).

In DDM, the equations which are related to internal d.o.f. of the subdomains are often eliminated first. For that purpose, Eq. (32) is transformed to

$$S^s u^s_{\mathrm{b}} = \hat{f}^s_{\mathrm{b}} - B^{\mathrm{T}}_{\mathrm{b}}\lambda, \tag{39}$$

where

$$S^s = \begin{bmatrix} S^{(1)} & & \\ & \ddots & \\ & & S^{(N_s)} \end{bmatrix}, \quad S^{(s)} = K_{\mathrm{bb}}^{(s)} - K_{\mathrm{bi}}^{(s)} K_{\mathrm{ii}}^{(s)^{-1}} K_{\mathrm{ib}}^{(s)}, \qquad s = 1, \ldots, N_s \tag{40}$$

and

$$\hat{f}_{\mathrm{b}}^s = f_{\mathrm{b}}^s - K_{\mathrm{bi}}^s K_{\mathrm{ii}}^{s^{-1}} f_{\mathrm{i}}^s. \tag{41}$$

Subscripts b and i denote the restriction of the matrices to interface and internal d.o.f., respectively. $S^{(s)}$ is a condensed stiffness matrix, also called Schur complement.

Following the reasoning of Eqs. (36) and (37), a solution of Eq. (39) exists under the condition

$$R_{\mathrm{b}}^{s^{\mathrm{T}}} (\hat{f}_{\mathrm{b}}^s - B_{\mathrm{b}}^{\mathrm{T}} \lambda) = 0 \tag{42}$$

and is given by

$$u_{\mathrm{b}}^s = S^{s^+} (\hat{f}_{\mathrm{b}}^s - B_{\mathrm{b}}^{\mathrm{T}} \lambda) + R_{\mathrm{b}}^s a, \tag{43}$$

where R_{b}^s denotes the restriction of R^s to the interface d.o.f.

The generalized inverse matrices used in this section are related to the stiffness matrices of the subdomains with the following property: For a generalized inverse K^{s^+} (or S^{s^+}) of K^s (or S^s) there is a matrix Y such that

$$K^{s^+} K^s = I + R^s Y \quad (\text{or } S^{s^+} S^s = I + R_{\mathrm{b}}^s Y). \tag{44}$$

Furthermore, we have

$$\mathrm{range}(R^s) = \mathrm{null}(K^s) \Rightarrow K^s R^s = 0 \quad \text{and} (R_{\mathrm{b}}^s) = \mathrm{null}(S^s) \Rightarrow S^s R_{\mathrm{b}}^s = 0. \tag{45}$$

Eqs. (44) and (45) imply that

$$K^s K^{s^+} K^s = K^s \quad \text{and} \quad S^s S^{s^+} S^s = S^s. \tag{46}$$

3.2. Solving the interface problem

In structural mechanics, a number of DDM employ the preconditioned conjugate gradient (PCG) method to solve iteratively the interface problem. In this section, we outline the standard PCG algorithm, which is used in this paper as a basis for the different proofs provided in subsequent sections. Thus, given a linear system of equations $Ax = b$, where A is symmetric positive semi-definite and a preconditioning matrix $\tilde{A}^{-1}$, the PCG algorithm proceeds as follows:

- Initialize

$$r^0 = b - Ax^0, \quad z^0 = \tilde{A}^{-1} r^0, \quad p^0 = z^0, \quad q^0 = Ap^0, \quad \eta^0 = \frac{p^{0^{\mathrm{T}}} r^0}{p^{0^{\mathrm{T}}} q^0}.$$

- Iterate $k = 1, 2, \ldots$ until convergence.

Estimate of the solution	$x^k = x^{k-1} + \eta^{k-1} p^{k-1}$
Residual	$r^k = r^{k-1} - \eta^{k-1} q^{k-1}$
Preconditioned residual	$z^k = \tilde{A}^{-1} r^k$
Search vector	
(a) No reorthogonalization case	$p^k = z^k + \dfrac{z^{k\mathrm{T}} r^k}{z^{k-1\mathrm{T}} r^{k-1}} p^{k-1}$
(b) Reorthogonalization case	$p^k = z^k - \sum_{i=0}^{k-1} \dfrac{z^{k\mathrm{T}} q^i}{p^{i\mathrm{T}} q^i} p^i$
Product of search vector by A	$q^k = A p^k$
Step length	
(a) No reorthogonalization case	$\eta^k = \dfrac{z^{k\mathrm{T}} r^k}{p^{k\mathrm{T}} q^k}$
(b) Reorthogonalization case	$\eta^k = \dfrac{p^{k\mathrm{T}} r^k}{p^{k\mathrm{T}} q^k}$

4. The primal substructuring method

A basic DDM is the primal substructuring method, abbreviated in the following as PSM. In the context of this DDM, the internal d.o.f. of the subdomains are eliminated first. The PSM interface displacement problem is thus obtained by combining Eqs. (3), (4) and (39) in order to form the equation

$$\hat{S} u_{\mathrm{b}} = \hat{f}_{\mathrm{b}}, \tag{47}$$

where

$$\hat{S} = L_{\mathrm{b}}^{\mathrm{T}} S^s L_{\mathrm{b}} \quad \text{and} \quad \hat{f}_{\mathrm{b}} = L_{\mathrm{b}}^{\mathrm{T}} \hat{f}_{\mathrm{b}}^s. \tag{48}$$

The solution of the linear system of Eq. (47) is usually performed with the PCG method.

In the past, several strategies have been proven efficient for preconditioning the PCG for these types of problems (see for example [2,12,13]). A quite common choice is the preconditioner

$$\tilde{A}^{-1} = L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} L_{\mathrm{p_b}}, \tag{49}$$

which is used in the so-called Neumann–Neumann PSM. More precisely, the preconditioner (49) is implemented as follows:

$$\tilde{A}^{-1} = L_{\mathrm{p_b}}^{\mathrm{T}} N_{\mathrm{b}}^s K^{s^+} N_{\mathrm{b}}^{s\mathrm{T}} L_{\mathrm{p_b}}, \tag{50}$$

where N_{b}^s is a Boolean matrix which extracts the interface d.o.f. from subdomain d.o.f. vectors, as in equations

$$u_{\mathrm{b}}^s = N_{\mathrm{b}}^s u^s \quad \text{and} \quad f_{\mathrm{b}}^s = N_{\mathrm{b}}^s f^s. \tag{51}$$

Further on in this paper, S^{s^+} is often used in place of $N_{\mathrm{b}}^s K^{s^+} N_{\mathrm{b}}^{s\mathrm{T}}$ for brevity. Matrix $N_{\mathrm{b}}^s K^{s^+} N_{\mathrm{b}}^{s\mathrm{T}}$ is thus the particular generalized inverse of S^s that is implicitly assumed, in the rest of the paper. Another important class of efficient preconditioners for the PCG method implemented within the PSM are those used in the BDD methods, which were introduced in [2] and are discussed in further sections.

5. The FETI method

The FETI method, is a dual DDM that has been implemented for a number of problems in computational mechanics. Since its introduction [1], it has attracted a lot of attention and is today considered as a fast domain decomposition algorithm suitable for both serial and parallel computing environments.

While its predecessor, the PSM, performs iterations in order to compute the interface displacement vector u_b of the structure, the FETI method iterates on the Lagrange multiplier vector λ. The Lagrange multipliers, that is the interaction forces between the subdomains are dual with respect to the interface displacements and this explains the name dual substructuring method, in comparison to PSM.

In the context of the FETI method, the nodal force vector f of the structure is first split to the subdomains:

$$f^s = L_p f. \tag{52}$$

Combining Eqs. (17), (36) and (37), the following system is obtained:

$$\begin{bmatrix} F_I & -G \\ -G^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} d \\ -e \end{bmatrix}, \tag{53}$$

where

$$F_I = BK^{s^+}B^T, \quad G = BR^s, \quad d = BK^{s^+}f^s \quad \text{and} \quad e = R^{s^T}f^s. \tag{54}$$

In order to decouple the linear system (53), the projection operator

$$P = I - QG(G^TQG)^{-1}G^T \tag{55}$$

is used. Eq. (55) defines the operator P for a square matrix Q. P satisfies the equations

$$G^TP = 0, \tag{56}$$

$$PQG = 0, \tag{57}$$

$$P^2 = P. \tag{58}$$

Premultiplying the first of the two matrix equations in (53) with $(G^TQG)^{-1}G^TQ$, it follows that for a given Lagrange multiplier vector λ, the vector of the zero energy mode amplitudes a is equal to

$$a = -(G^TQG)^{-1}G^TQ(d - F_I\lambda). \tag{59}$$

Furthermore, using Eqs. (37) and (59), it follows that the jump $\Delta u_b = Bu^s$ of the displacement field at subdomain interfaces is equal to

$$\Delta u_b = Bu^s = d - F_I\lambda + Ga = P^T(d - F_I\lambda). \tag{60}$$

Note that Eq. (60) implies that the interface displacement jump Δu_b satisfies the condition

$$G^TQ\Delta u_b = 0. \tag{61}$$

Based on Eq. (60), the linear system (53) may easily be proven equivalent to the following system of equations, in which the unknown vectors λ and a are decoupled:

$$a = -(G^TQG)^{-1}G^TQ(d - F_I\lambda), \tag{62a}$$

$$P^TF_I\lambda = P^Td, \tag{62b}$$

$$G^T\lambda = e. \tag{62c}$$

In order to solve Eqs. (62b) and (62c) for the Lagrange multiplier vector λ, the following splitting of the Lagrange multipliers is performed:

$$\lambda = \lambda_0 + P\overline{\lambda}, \tag{63}$$

where λ_0 is a vector satisfying (62c). The component λ_0 is chosen as

$$\lambda_0 = QG(G^{\mathrm{T}}QG)^{-1}e, \tag{64}$$

which satisfies (62c). Based on Eqs. (63) and (64), the system of the Eqs. (62b) and (62c) can thus be written as

$$(P^{\mathrm{T}}F_IP)\overline{\lambda} = P^{\mathrm{T}}(d - F_I\lambda_0). \tag{65}$$

The one-level FETI (FETI-1) method consists in applying the PCG algorithm to Eq. (65) in order to compute the Lagrange multiplier vector $\overline{\lambda}$. This method was named one-level, in order to be distinguished from the so-called two-level FETI (FETI-2) method, which was later proposed for faster solution of plate and shell problems [14,15]. Another more recently proposed variant of the FETI method which should be noted is the dual–primal FETI (FETI-DP) method [16,17].

The FETI-1 algorithm is completed as follows: After computing the Lagrange multiplier vector $\overline{\lambda}$ by applying the PCG algorithm to Eq. (65), the amplitudes a of the subdomain rigid body modes are computed from Eq. (62a). Subdomain displacement fields u^s are then computed from Eq. (37) and the total displacement field u of the structure is finally given by

$$u = L_{\mathrm{p}}^{\mathrm{T}}u^s. \tag{66}$$

It should further be noted that if $\overline{\lambda}^k$, z^k and p^k are the kth solution estimate, preconditioned residual and search vector, respectively, of the PCG applied for the solution of Eq. (65), then the PCG algorithm can be written in an alternative form in order to operate directly on the vectors

$$\lambda^k = \lambda_0 + P\overline{\lambda}^k, \quad \tilde{z}^k = Pz^k \quad \text{and} \quad \tilde{p}^k = Pp^k. \tag{67}$$

The corresponding algorithm, which is given in [5,18] and other publications, is usually used instead of the standard PCG algorithm in order to solve Eq. (65).

5.1. Preconditioners for the FETI method

Two preconditioners are more often used in the FETI method:

$$\text{(a) } \tilde{F}_I^{D^{-1}} = B_{\mathrm{p_b}}S^sB_{\mathrm{p_b}}^{\mathrm{T}} \quad \text{(Dirichlet preconditioner)}, \tag{68}$$

$$\text{(b) } \tilde{F}_I^{L^{-1}} = B_{\mathrm{p_b}}K_{\mathrm{bb}}^sB_{\mathrm{p_b}}^{\mathrm{T}} \quad \text{(lumped preconditioner)}. \tag{69}$$

The Dirichlet preconditioner is typically used in fourth-order problems. Moreover, in second-order problems the lumped preconditioner is usually more efficient in terms of the total solution time. In some second-order problems however, namely in highly heterogeneous structures and in problems where subdomains of bad aspect ratio are generated, the Dirichlet preconditioner may outperform the lumped one [10]. Variant forms of the Dirichlet preconditioner using approximate expressions for K_{ii}^s of the Schur complement $S^s = K_{\mathrm{bb}}^s - K_{\mathrm{bi}}^sK_{\mathrm{ii}}^{s^{-1}}K_{\mathrm{ib}}^s$, have also been tested in [19].

5.2. Projection operators for the FETI method

For homogeneous problems, operator P defined in Eq. (55) is usually implemented with $Q = I$. However, for heterogeneous problems matrix Q is set equal to one of the following three choices:

$$\text{(a) } Q^{\mathrm{D}} = \tilde{F}_I^{D^{-1}} \quad \text{(Dirichlet preconditioner)}, \tag{70}$$

$$\text{(b) } Q^{\mathrm{L}} = \tilde{F}_I^{L^{-1}} \quad \text{(lumped preconditioner)}, \tag{71}$$

$$\text{(c) } Q^{\mathrm{SL}} = B_{\mathrm{p_b}} K^s_{\mathrm{bb_d}} B^{\mathrm{T}}_{\mathrm{p_b}} \quad \text{(superlumped matrix)}, \tag{72}$$

where $K^s_{\mathrm{bb_d}}$ denotes a diagonal matrix whose main diagonal is equal to the main diagonal of K^s_{bb}. The superlumped choice of Eq. (72) has been recently proposed in a discussion of highly heterogeneous problems in [10].

In the FETI operator P of Eq. (55), it is required to solve linear problems of the form

$$(G^{\mathrm{T}} Q G)x = b. \tag{73}$$

The linear system (73) constitutes the so-called "coarse-grid" problem of the FETI method. This name is explained by the fact that $G^{\mathrm{T}}QG$ is a sparse matrix, with the typical sparsity pattern of a finite element stiffness matrix, if one considers each subdomain as a finite element node having the same number of d.o.f. as the number of its zero energy modes. The coarse problem of FETI ensures the exchange of information between remote subdomains of the structure at each FETI iteration. It thus guarantees fast convergence of the PCG algorithm.

6. A family of FETI derived preconditioners for the primal substructuring method

In this section, we introduce a new category of preconditioners for the PSM. The PCG is applied for the solution of Eq. (47) in order to compute the interface displacement vector u_{b}, given an interface force vector $\hat{f}_{\mathrm{b}}$. A good preconditioner for the PSM must treat the kth PCG residual $r^k = \hat{f}_{\mathrm{b}} - \hat{S}u^k_{\mathrm{b}}$ as applied forces on the interface nodes of the structure and return in $z^k = \tilde{A}^{-1} r^k$ a good estimate of the interface displacements of the structure for the applied forces r^k. For instance, if the PSM preconditioning step is performed with any solver, like for example the FETI method, the PCG will immediately converge in the first iteration. We therefore propose to use as preconditioner of the PSM, a crude approximation of the FETI solution and as such we choose the first estimate for the interface displacements of the structure obtained from the first iteration of the FETI method.

For example, consider the FETI-1 algorithm with an applied forces vector f equal to

$$f = N^{\mathrm{T}}_{\mathrm{b}} r^k, \tag{74}$$

where r^k is the kth residual of the PSM and N_{b} is a Boolean matrix which extracts the interface d.o.f. from a vector with the size of all d.o.f. of the structure, as in equations $u_{\mathrm{b}} = N_{\mathrm{b}} u$ and $f_{\mathrm{b}} = N_{\mathrm{b}} f$. Since all forces in the load vector of Eq. (74) are applied on the interface nodes of the structure, we have $f_{\mathrm{b}} = r^k$ and $f^s_{\mathrm{i}} = 0$. Furthermore, the interface forces f_{b} may be split to the subdomains with the equation

$$f^s_{\mathrm{b}} = L_{\mathrm{p_b}} f_{\mathrm{b}} = L_{\mathrm{p_b}} r^k. \tag{75}$$

From Eqs. (41) and (75), it follows that $\hat{f}^s_{\mathrm{b}} = f^s_{\mathrm{b}} = L_{\mathrm{p_b}} r^k$. The components e and d of Eq. (54) thus become $e = R^{s^{\mathrm{T}}} f^s = R^{s^{\mathrm{T}}}_{\mathrm{b}} r^k$ and $d = BK^{s^+} f^s = B_{\mathrm{b}} S^{s^+} L_{\mathrm{p_b}} r^k$. We also note that with respect to interface values, matrix F_I of Eq. (54) may be written as $F_I = BK^{s^+} B^{\mathrm{T}} = B_{\mathrm{b}} S^{s^+} B^{\mathrm{T}}_{\mathrm{b}}$. Furthermore, the initial Lagrange multiplier vector λ_0 of Eq. (64) is equal to

$$\lambda_0 = QG(G^{\mathrm{T}}QG)^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}}r^k. \tag{76}$$

From Eq. (62a) it follows that the zero energy mode amplitudes a_0, which correspond to the initial Lagrange multiplier vector λ_0, may be expressed by the relation

$$\begin{aligned} a_0 &= -(G^{\mathrm{T}}QG)^{-1}G^{\mathrm{T}}Q(d - F_I\lambda_0) \\ &= -(G^{\mathrm{T}}QG)^{-1}G^{\mathrm{T}}QB_{\mathrm{b}}S^{s^+}(L_{\mathrm{p_b}}r^k - B_{\mathrm{b}}^{\mathrm{T}}\lambda_0). \end{aligned} \tag{77}$$

The above analysis combined with Eqs. (5) and (43) leads to the following interface displacements u_{b_0} estimated from the initialization of the FETI-1 algorithm:

$$\begin{aligned} u_{\mathrm{b}_0} &= L_{\mathrm{p_b}}^{\mathrm{T}}u_{\mathrm{b}}^s \\ &= L_{\mathrm{p_b}}^{\mathrm{T}}(S^{s^+}(\hat{f}_{\mathrm{b}}^s - B_{\mathrm{b}}^{\mathrm{T}}\lambda_0) + R_{\mathrm{b}}^s a_0) \\ &= L_{\mathrm{p_b}}^{\mathrm{T}}(S^{s^+}(r^k - B_{\mathrm{b}}^{\mathrm{T}}\lambda_0) - R_{\mathrm{b}}^s(G^{\mathrm{T}}QG)^{-1}G^{\mathrm{T}}QB_{\mathrm{b}}S^{s^+}(L_{\mathrm{p_b}}r^k - B_{\mathrm{b}}^{\mathrm{T}}\lambda_0)) \\ &= L_{\mathrm{p_b}}^{\mathrm{T}}(I - R_{\mathrm{b}}^s(G^{\mathrm{T}}QG)^{-1}G^{\mathrm{T}}QB_{\mathrm{b}})S^{s^+}(L_{\mathrm{p_b}}r^k - B_{\mathrm{b}}^{\mathrm{T}}\lambda_0) \\ &= L_{\mathrm{p_b}}^{\mathrm{T}}(I - R_{\mathrm{b}}^s(G^{\mathrm{T}}QG)^{-1}G^{\mathrm{T}}QB_{\mathrm{b}})S^{s^+}(I - B_{\mathrm{b}}^{\mathrm{T}}QG(G^{\mathrm{T}}QG)^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}})L_{\mathrm{p_b}}r^k. \end{aligned} \tag{78}$$

Hence, from Eq. (78) we deduce the PSM preconditioner

$$\tilde{A}^{-1} = L_{\mathrm{p_b}}^{\mathrm{T}}H_{\mathrm{b}}^{\mathrm{T}}S^{s^+}H_{\mathrm{b}}L_{\mathrm{p_b}}, \tag{79}$$

where

$$H_{\mathrm{b}} = I - B_{\mathrm{b}}^{\mathrm{T}}QG(G^{\mathrm{T}}QG)^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}}. \tag{80}$$

Note that operator H_{b} satisfies the equations

$$H_{\mathrm{b}}B_{\mathrm{b}}^{\mathrm{T}} = B_{\mathrm{b}}^{\mathrm{T}}P, \tag{81}$$

$$R_{\mathrm{b}}^{s^{\mathrm{T}}}H_{\mathrm{b}} = 0, \tag{82}$$

$$H_{\mathrm{b}}B_{\mathrm{b}}^{\mathrm{T}}QG = 0, \tag{83}$$

$$H_{\mathrm{b}}^2 = H_{\mathrm{b}}. \tag{84}$$

The PSM preconditioner (79) has been constructed from the initialization of the FETI-1 algorithm. Applying the same reasoning, it is simple to construct PSM preconditioners from other FETI variants, like for instance the FETI-2 method. In the FETI-2 algorithm, additional Lagrange multipliers are introduced to a small set of interface nodes. This set of nodes is usually composed of the cross-points, that is the interface nodes belonging to three or more subdomains. The additional Lagrange multipliers are then used to enforce at each FETI iteration the displacement compatibility condition for the selected set of interface nodes:

$$C_\lambda^{\mathrm{T}}\Delta u_{\mathrm{b}} = C_\lambda^{\mathrm{T}}P^{\mathrm{T}}(d - F_I\lambda) = 0, \tag{85}$$

where C_λ is a Boolean matrix which maps the additional Lagrange multipliers, on the full Lagrange multiplier vector of the structure. Like in FETI-1 and the respective preconditioner of Eq. (79), the first estimate for the interface displacements of the structure, obtained during the initialization of the FETI-2 algorithm, leads to the preconditioner

$$\tilde{A}^{-1} = L_{\mathrm{p_b}}^{\mathrm{T}}H_{\mathrm{b}}^{\mathrm{T}}(S^{s^+} - S^{s^+}B_{\mathrm{b}}^{\mathrm{T}}PC_\lambda(C_\lambda^{\mathrm{T}}P^{\mathrm{T}}F_IPC_\lambda)^{-1}C_\lambda^{\mathrm{T}}P^{\mathrm{T}}B_{\mathrm{b}}S^{s^+})H_{\mathrm{b}}L_{\mathrm{p_b}}. \tag{86}$$

The PSM with the preconditioner of Eq. (79) or (86) requires the solution of the same local subdomain problems and coarse problems as the Dirichlet preconditioned FETI-1 or FETI-2 methods, respectively. Hence, a PSM iteration with the preconditioner of Eq. (79) or (86) has a similar computational cost as one iteration of the FETI-1 or FETI-2 method, respectively. Furthermore, the reasoning behind the development of preconditioners (79) and (86) can easily be applied to FETI-DP and other FETI variants, leading to the corresponding PSM preconditioners.

In the case of the standard FETI-DP formulation [16], a small set of interface d.o.f. is selected for the coarse space. These coarse space interface d.o.f. contain the d.o.f. of the cross-points plus the ends of interface edges, or simply the d.o.f. of the cross-points. Let $u_{\mathrm{b_c}}$ denote the vector of the coarse space interface d.o.f. and

$$u_{\mathrm{r}}^{s} = \begin{bmatrix} u_{\mathrm{r}}^{(1)^{\mathrm{T}}} & \cdots & u_{\mathrm{r}}^{(N_s)^{\mathrm{T}}} \end{bmatrix}^{\mathrm{T}} \tag{87}$$

denote the vector of the remaining subdomain d.o.f. If

$$u_{\mathrm{b_c}}^{s} = \begin{bmatrix} u_{\mathrm{b_c}}^{(1)^{\mathrm{T}}} & \cdots & u_{\mathrm{b_c}}^{(N_s)^{\mathrm{T}}} \end{bmatrix}^{\mathrm{T}} \tag{88}$$

is the vector of all subdomain d.o.f. that belong to the coarse space, then vectors $u_{\mathrm{b_c}}$ and $u_{\mathrm{b_c}}^{s}$ are related with the equation

$$u_{\mathrm{b_c}}^{s} = L_{\mathrm{c}} u_{\mathrm{b_c}}, \tag{89}$$

where L_{c} is a Boolean mapping matrix. Eq. (89) expresses the interface displacement compatibility condition for the coarse space interface d.o.f., similar to Eq. (3). Hence, using the notation of Eqs. (87)–(89), the PSM preconditioner that can be derived from the first iteration of the FETI-DP method may easily be shown equal to

$$\tilde{A}^{-1} = \tilde{L}_{\mathrm{p_r}}^{\mathrm{T}} K_{\mathrm{rr}}^{s^{-1}} \tilde{L}_{\mathrm{p_r}} + (N_{\mathrm{b_c}}^{\mathrm{T}} - \tilde{L}_{\mathrm{p_r}}^{\mathrm{T}} K_{\mathrm{rr}}^{s^{-1}} K_{\mathrm{rc}}^{s} L_{\mathrm{c}}) K_{\mathrm{cc}}^{*^{-1}} (-L_{\mathrm{c}}^{\mathrm{T}} K_{\mathrm{cr}}^{s} K_{\mathrm{rr}}^{s^{-1}} \tilde{L}_{\mathrm{p_r}} + N_{\mathrm{b_c}}), \tag{90}$$

where

$$K_{\mathrm{cc}}^{*} = L_{\mathrm{c}}^{\mathrm{T}} (K_{\mathrm{cc}}^{s} - K_{\mathrm{cr}}^{s} K_{\mathrm{rr}}^{s^{-1}} K_{\mathrm{rc}}^{s}) L_{\mathrm{c}} \tag{91}$$

and

$$\tilde{L}_{\mathrm{p_r}} = N_{\mathrm{r}}^{s} N_{\mathrm{b}}^{s^{\mathrm{T}}} L_{\mathrm{p_b}}. \tag{92}$$

In Eq. (90), subscripts c and r denote the restriction of the matrices to the coarse problem d.o.f. and the remaining d.o.f., respectively. Matrix N_{r}^{s} is a Boolean matrix which extracts the subdomain d.o.f. that do not belong to the coarse problem, from subdomain d.o.f. vectors, like in equation $u_{\mathrm{r}}^{s} = N_{\mathrm{r}}^{s} u^{s}$, similar to the way matrix N_{b}^{s} is used in Eq. (51). Furthermore, matrix $N_{\mathrm{b_c}}$ is used in Eq. (90), in order to extract the coarse problem d.o.f. from global interface d.o.f. vectors, like in equation $u_{\mathrm{b_c}} = N_{\mathrm{b_c}} u_{\mathrm{b}}$.

6.1. Connections of the FETI derived primal preconditioners with the FETI methods

The proposed preconditioned PSM are closely related to the FETI methods from which they were constructed. It can be deduced that at each PCG iteration of the PSM with the preconditioners of Eq. (79), (86) or (90), the solution of the same local subdomain and coarse-grid problems is required, as in the corresponding Dirichlet preconditioned FETI methods. It thus becomes obvious that the PSM with these preconditioners, have comparable computational cost as the respective FETI methods with the Dirichlet preconditioner, in both the initialization and at each iteration of the PCG.

Apart from these remarks, theoretical connections between the introduced primal methods and the respective FETI methods can be established. If for example, all methods discussed in this paper are considered as PCG methods applied to an equation of the form $Ax = b$ with a preconditioner $\tilde{A}^{-1}$, then for the Dirichlet preconditioned FETI-1 method (Eq. (65)), due to Eq. (81), we have

$$A = P^{\mathrm{T}} F_I P = P^{\mathrm{T}} B_{\mathrm{b}} S^{s^+} B_{\mathrm{b}}^{\mathrm{T}} P = B_{\mathrm{b}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}}) B_{\mathrm{b}}^{\mathrm{T}} \quad \text{and} \quad \tilde{A}^{-1} = B_{\mathrm{p_b}} S^s B_{\mathrm{p_b}}^{\mathrm{T}}, \tag{93}$$

while for the respective primal method with the preconditioner of Eq. (79), we have

$$A = L_{\mathrm{b}}^{\mathrm{T}} S^s L_{\mathrm{b}} \quad \text{and} \quad \tilde{A}^{-1} = L_{\mathrm{p_b}}^{\mathrm{T}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}}) L_{\mathrm{p_b}}. \tag{94}$$

Thus, for FETI-1

$$\tilde{A}^{-1} A = B_{\mathrm{p_b}} S^s B_{\mathrm{p_b}}^{\mathrm{T}} B_{\mathrm{b}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}}) B_{\mathrm{b}}^{\mathrm{T}}, \tag{95}$$

while for the corresponding primal method

$$\tilde{A}^{-1} A = L_{\mathrm{p_b}}^{\mathrm{T}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}}) L_{\mathrm{p_b}} L_{\mathrm{b}}^{\mathrm{T}} S^s L_{\mathrm{b}}. \tag{96}$$

From Eqs. (24), (44) and (82), it follows that there is a matrix Y such that the following relations hold:

$$\begin{aligned} H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} S^s L_{\mathrm{b}} &= H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} S^s L_{\mathrm{b}} \\ &= H_{\mathrm{b}}^{\mathrm{T}} (I + R_{\mathrm{b}}^s Y) L_{\mathrm{b}} \\ &= H_{\mathrm{b}}^{\mathrm{T}} L_{\mathrm{b}} = L_{\mathrm{b}}. \end{aligned} \tag{97}$$

In view of Eqs. (27) and (97), the matrix product $\tilde{A}^{-1} A$ of Eq. (96) thus takes the form

$$\begin{aligned} \tilde{A}^{-1} A &= L_{\mathrm{p_b}}^{\mathrm{T}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}}) L_{\mathrm{p_b}} L_{\mathrm{b}}^{\mathrm{T}} S^s L_{\mathrm{b}} \\ &= L_{\mathrm{p_b}}^{\mathrm{T}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}}) (I - B_{\mathrm{b}}^{\mathrm{T}} B_{\mathrm{p_b}}) S^s L_{\mathrm{b}} \\ &= L_{\mathrm{p_b}}^{\mathrm{T}} (H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} S^s L_{\mathrm{b}} - H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} B_{\mathrm{b}}^{\mathrm{T}} B_{\mathrm{p_b}} S^s L_{\mathrm{b}}) \\ &= L_{\mathrm{p_b}}^{\mathrm{T}} (L_{\mathrm{b}} - H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} B_{\mathrm{b}}^{\mathrm{T}} B_{\mathrm{p_b}} S^s L_{\mathrm{b}}) \\ &= L_{\mathrm{p_b}}^{\mathrm{T}} (I - H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} B_{\mathrm{b}}^{\mathrm{T}} B_{\mathrm{p_b}} S^s) L_{\mathrm{b}}. \end{aligned} \tag{98}$$

Finally, from Eq. (95), for FETI-1 we have

$$\tilde{A}^{-1} A = B_{\mathrm{p_b}} A_{\mathrm{F}}^{\mathrm{T}} B_{\mathrm{b}}^{\mathrm{T}}, \tag{99}$$

where

$$A_{\mathrm{F}} = H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} B_{\mathrm{b}}^{\mathrm{T}} B_{\mathrm{p_b}} S^s, \tag{100}$$

while for the respective primal method Eq. (98) implies that

$$\tilde{A}^{-1} A = L_{\mathrm{p_b}}^{\mathrm{T}} (I - A_{\mathrm{F}}) L_{\mathrm{b}}. \tag{101}$$

Note that matrix A_{F} appears in the corresponding expressions for $\tilde{A}^{-1} A$ in both FETI-1 and the respective primal method. It is well known from the theory of the PCG method that the condition number and the eigenspectrum of matrix $\tilde{A}^{-1} A$ have an important impact on the rate of convergence of the PCG. In order to investigate the structure of the eigenspectrum of the two methods, we computed the eigenvalues of matrices

$\tilde{A}^{-1}A$ of Eqs. (99) and (101) in a number of small characteristic test examples in Matlab. The most important conclusions that can be drawn from these tests on the Dirichlet preconditioned FETI-1 method and the respective primal method in different structural models may be summarized in the following:

1. Depending on the structural problem, the choice for the definition of non-redundant, redundant or orthonormal Lagrange multipliers and the choice of matrix Q, matrix $\tilde{A}^{-1}A$ of FETI-1 may have a number of eigenvalues equal to zero or one. Similarly, the corresponding matrix of its primal alternative method may have a number of eigenvalues equal to one. The interesting observation lies in the fact that the remaining part of the eigenspectrum which includes eigenvalues not equal to zero or one, is the same in both methods. Hence, the two methods have the same non-zero and non-unit eigenvalues, provided that for both methods the same choice for matrix Q, definition of Lagrange multipliers and homogeneous or heterogeneous scaling is implemented.
2. Furthermore, if we choose any of the Q matrices which are suggested for heterogeneous problems (see Section 5.2), or if we choose $Q = B_{p_b} B_{p_b}^T$ for homogeneous problems, all versions of FETI-1 and the respective primal method with any choice of Lagrange multipliers have again the same non-zero and non-unit eigenvalues. Therefore, excluding zero or unit eigenvalues, the eigenspectrum remains unaffected by the way the Lagrange multipliers are defined. We also note that using the relations of Section 2.3, it can be shown that for orthonormal Lagrange multipliers, the choice of $Q = B_{p_b} B_{p_b}^T$ for homogeneous problems is equivalent to $Q = I$, while for redundant Lagrange multipliers, it is equivalent to the choice of Q which has been tested in [8] and was found to produce similar iteration counts with the choice of $Q = I$.

The second of the above remarks leads to the conclusion that any of the Lagrange multiplier definitions discussed in Section 2.2 generates methods with similar eigenspectrum and therefore similar iteration count. These comparable iteration counts have been verified in all tests that we have performed in different structural problems and with several FETI methods or their primal alternatives. Therefore, since all three definitions of Lagrange multipliers appear to be equivalent in terms of the computational cost of the corresponding solution method, one can freely implement any of them.

Furthermore, from the first of the above remarks, one can deduce that the Dirichlet preconditioned FETI-1 method should require iteration counts comparable to those obtained by the respective primal method. It also follows that the primal alternative of FETI-1 should inherit the well-known optimal convergence properties of FETI-1 [20]. In fact in all tests that we have conducted so far with the Dirichlet preconditioned FETI-1, FETI-2, FETI-DP methods and their primal alternatives, we have obtained comparable iteration counts for the same configuration of primal and dual methods. These findings are further explained in subsequent sections.

From the above discussion it can be concluded that the PSM preconditioners derived from the first estimate of the displacement field obtained by the first iteration of a FETI method, owe their small iteration count to the optimal coarse problem which is solved at each preconditioning step. As this coarse problem is formulated using FETI methods theory, in the following we refer to this primal category of methods as a primal class of FETI methods or P-FETI class methods, the letter P designating that the outside method is a primal method. We thus refer to the PSM with the preconditioner of Eq. (79), (86) or (90) as the P-FETI1, P-FETI2 or P-FETIDP method, respectively.

A general remark drawn from this section is that each dual method can be transformed into a primal method by means of a suitable preconditioning of the PSM. Therefore, for each coarse problem used in the FETI family of methods, there are two possible formulations, one primal and one dual. For example, for the one-level FETI coarse problem, we have a dual formulation (Eq. (65) of the FETI-1 method) and a primal formulation (Eq. (79) of the P-FETI1 method). The two different formulations are further compared in subsequent sections of this paper.

7. The balancing domain decomposition method

This section discusses the balancing domain decomposition (BDD) method [2], in connection to concepts used in the derivation of the P-FETI methods from the original FETI methods. More precisely, in Section 6, it was shown that taking the well-known dual FETI methods as a basis, one is led to a family of closely related methods by means of preconditioning the PSM with the first iteration of the dual methods. The present section describes two formulations of the BDD that are interrelated in a similar way. In fact, in the field of multigrid methods and of multilevel Schwarz DDM it is well known that many iterative methods can also be written as preconditioners and vice versa. When applied to the BDD, this concept leads to two variant iterative methods, described in the following two subsections.

7.1. A two-level primal substructuring method

The formulation that is described in this subsection consists in applying a standard two-level solution technique to the PSM. This technique, which has been used under a different framework in multigrid methods, in the FETI methods (see for example [14,18,21]) and other iterative methods, is here briefly presented in the context of the PSM.

Let C denote a matrix of arbitrary entries, in which each column expresses a particular linear combination of the equations of the linear system (47). By setting

$$C^{\mathrm{T}}(\hat{f}_{\mathrm{b}} - \hat{S}u_{\mathrm{b}}) = 0, \tag{102}$$

one enforces a weak satisfaction of Eq. (47), with respect to the defined in C linear combinations of Eq. (47). One way to satisfy Eq. (102) while preserving the symmetry of Eq. (47) is by introducing the following splitting of the interface displacement vector u_{b}:

$$u_{\mathrm{b}} = \hat{u}_{\mathrm{b}} + Cu_{\mathrm{c}}. \tag{103}$$

Using the splitting (103), condition (102) and Eq. (47), the following redundant system of equations is formed:

$$\begin{bmatrix} \hat{S} & \hat{S}C \\ C^{\mathrm{T}}\hat{S} & C^{\mathrm{T}}\hat{S}C \end{bmatrix} \begin{bmatrix} \hat{u}_{\mathrm{b}} \\ u_{\mathrm{c}} \end{bmatrix} = \begin{bmatrix} \hat{f}_{\mathrm{b}} \\ C^{\mathrm{T}}\hat{f}_{\mathrm{b}} \end{bmatrix}. \tag{104}$$

It is worth noting that the augmented system (104) retains the symmetry of Eq. (47).

Solving for u_{c} from the second equation of the system (104) gives

$$u_{\mathrm{c}} = (C^{\mathrm{T}}\hat{S}C)^{-1}C^{\mathrm{T}}(\hat{f}_{\mathrm{b}} - \hat{S}\hat{u}_{\mathrm{b}}). \tag{105}$$

If Eq. (105) is substituted to the first equation of system (104), the following equation is obtained:

$$(\hat{S} - \hat{S}C(C^{\mathrm{T}}\hat{S}C)^{-1}C^{\mathrm{T}}\hat{S})\hat{u}_{\mathrm{b}} = (I - \hat{S}C(C^{\mathrm{T}}\hat{S}C)^{-1}C^{\mathrm{T}})\hat{f}_{\mathrm{b}}, \tag{106}$$

which may also be written as

$$P_{\mathrm{c}}^{\mathrm{T}}\hat{S}\hat{u}_{\mathrm{b}} = P_{\mathrm{c}}^{\mathrm{T}}\hat{f}_{\mathrm{b}}, \tag{107}$$

where

$$P_{\mathrm{c}} = I - C(C^{\mathrm{T}}\hat{S}C)^{-1}C^{\mathrm{T}}\hat{S}. \tag{108}$$

Note that Eq. (106) is symmetric and that operator P_{c} has the properties

$$P_{\mathrm{c}}C = 0, \quad C^{\mathrm{T}}\hat{S}P_{\mathrm{c}} = 0, \quad P_{\mathrm{c}}^{\mathrm{T}}\hat{S}P_{\mathrm{c}} = \hat{S}P_{\mathrm{c}} = P_{\mathrm{c}}^{\mathrm{T}}\hat{S} \quad \text{and} \quad P_{\mathrm{c}}^2 = P_{\mathrm{c}}. \tag{109}$$

A DDM can now be defined by a simple application of the PCG method in order to solve Eq. (107) with the preconditioner of Eq. (49). In the following subsections, it is explained that this DDM in fact constitutes an alternative formulation of the BDD method. At each PCG iteration, this method requires the solution of a linear problem of the form

$$(C^{\mathrm{T}}\hat{S}C)x = b, \tag{110}$$

which constitutes its coarse problem. The coarse space is spanned by the columns of matrix C, which represent linear combinations of the equations of the linear system (47). The defined in C linear combinations of Eq. (47) are implicitly satisfied at each iteration of the PCG, since due to Eq. (109) the kth residual of the PCG satisfies

$$C^{\mathrm{T}}r^k = C^{\mathrm{T}}P_{\mathrm{c}}^{\mathrm{T}}(\hat{f}_{\mathrm{b}} - \hat{S}\hat{u}_{\mathrm{b}}^k) = 0. \tag{111}$$

In any variation of this method, subdomain-, face-, edge- or vertex-wise linear combinations can be defined for the coarse space. (In the following subsection particular choices for the columns of matrix C that are used in the context of the BDD are considered.)

Alternatively, in Eq. (103) the columns of C can be considered as interface displacement modes. Following this reasoning, condition (111) becomes an equation of virtual work, since r^k denotes the interface residual forces of the kth iteration. As a general remark, it should be noted that it is preferable to choose displacement modes in C that do not relate remote interface d.o.f. Then, the coarse problem matrix $C^{\mathrm{T}}\hat{S}C$ will be banded and, thus allowing the use of solvers that exploit the sparsity of the problem matrix, like the skyline or sparse direct solvers.

7.2. A two-level primal preconditioner

The PSM preconditioner that can be obtained from the first estimate of interface displacements obtained from the first iteration of the algorithm of Section 7.1 can easily be shown equal to (if the coefficient $\eta^0 = 1$ is used instead of the value $\eta^0 = (p^{0\mathrm{T}}r^0)/(p^{0\mathrm{T}}q^0)$ in the PCG algorithm)

$$\tilde{A}^{-1} = P_{\mathrm{c}}L_{\mathrm{p_b}}^{\mathrm{T}}S^{s^+}L_{\mathrm{p_b}}P_{\mathrm{c}}^{\mathrm{T}} + A_{\mathrm{c}}, \tag{112}$$

where

$$A_{\mathrm{c}} = C(C^{\mathrm{T}}\hat{S}C)^{-1}C^{\mathrm{T}}. \tag{113}$$

Note that A_{c} satisfies the equations

$$P_{\mathrm{c}} = I - A_{\mathrm{c}}\hat{S} \Rightarrow A_{\mathrm{c}} = (I - P_{\mathrm{c}})\hat{S}^{-1}. \tag{114}$$

The standard formulation of the BDD [2] consists in applying the PCG in order to solve Eq. (47) with the preconditioner of Eq. (112). In addition, the columns of matrix C are set equal to the traces of the subdomain zero energy modes on the interface [2], that is

$$C = L_{\mathrm{p_b}}^{\mathrm{T}}R_{\mathrm{b}}^s. \tag{115}$$

Furthermore, using other subdomain-wise interface displacement modes C, one also obtains other BDD variants as the BDD method for plate and shell problems [12,22].

In the context of the BDD, if the initial vector of the PCG applied to Eq. (47), is not taken equal to 0, but instead is set equal to

$$u_{\mathrm{b}}^0 = A_{\mathrm{c}}\hat{f}_{\mathrm{b}}, \tag{116}$$

then the BDD residuals satisfy condition

$$C^{\mathrm{T}} r^k = 0, \quad k = 0, 1, \ldots \tag{117}$$

Condition (117), implies that

$$A_{\mathrm{c}} r^k = 0 \quad \text{and} \quad P_{\mathrm{c}}^{\mathrm{T}} r^k = r^k, \qquad k = 0, 1, \ldots \tag{118}$$

From Eq. (118) it thus follows that, if the initial vector u_{b}^0 is computed as in Eq. (116), then the BDD preconditioner (112) may be implemented as

$$\tilde{A}^{-1} = P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} L_{\mathrm{p_b}}, \tag{119}$$

which requires only one solution of the coarse problem per iteration.

Therefore, the standard formulation of the BDD preconditioner can be derived from the algorithm of the previous subsection in almost the same way in which the P-FETI methods are derived from the original FETI methods. In addition, in Section 6.1, the P-FETI1 method was found to be closely related to the FETI-1 methods. In particular, numerical experiments have shown that the two methods have practically the same eigenspectrum. Section 7.3 that follows immediately afterwards shows that the two algorithms described in Sections 7.1 and 7.2 have a stronger relation: they are in fact equivalent.

A number of existing as well as newly introduced in this paper methods are extensively discussed in further sections of this work. Table 1 shows a reference list of the methods, with their acronyms, initial literature sources and basic equations. Furthermore, Fig. 2 classifies the discussed methods, according to their coarse problems. Note that each row of Fig. 2 refers to methods that share the same coarse problem. Additionally, the first column of Fig. 2 contains methods in which the coarse problem is incorporated in the linear equation solved by PCG, while in the methods of the second column the coarse problem is included in the preconditioner.

7.3. Connection between the two-level primal formulations

With the preconditioner of Eq. (119), the standard BDD algorithm (Section 7.2) has practically the same computational cost per PCG iteration as the algorithm of Section 7.1. It can further be shown that these two algorithms have the same iteration count as well. This can easily be proven if one goes through the steps of the PCG method, while monitoring the relations of the PCG variables that pertain to each of the two algorithms. One is thus led to the conclusion that the the standard BDD algorithm with the initial vector of Eq. (116) and its alternative that has been described in Section 7.1 produce the same estimate for

Table 1
List of the discussed in this paper DDM

Name	Acronym	Proposed in	
One-level FETI method (dual formulation)	FETI-1	[1]	
Two-level FETI method (dual formulation)	FETI-2	[14]	
Dual–primal FETI method (dual formulation)	FETI-DP	[16]	
Balancing domain decomposition method	BDD	[2]	
		Linear equation to solve	Preconditioner
Primal class of FETI methods or P-FETI methods			
One-level FETI method in primal formulation or one-level P-FETI method	P-FETI1	(47)	(79)
Two-level FETI method in primal formulation or two-level P-FETI method	P-FETI2	(47)	(86)
Dual–primal FETI method in primal formulation or dual–primal P-FETI method	P-FETIDP	(47)	(90)

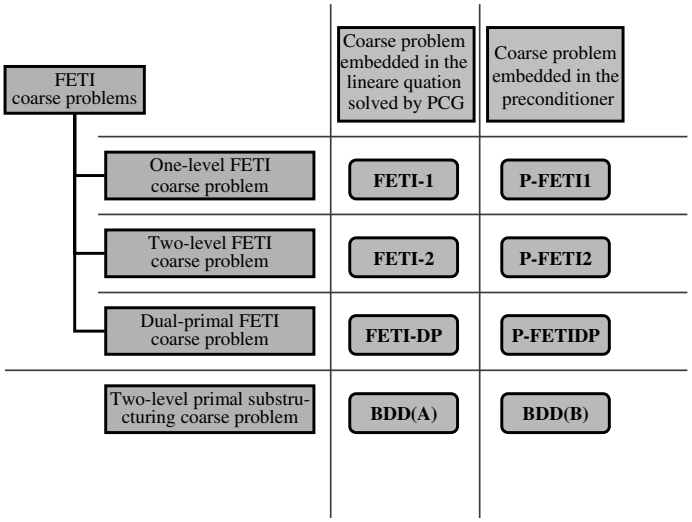

Fig. 2. A diagram of the discussed in this paper DDM. BDD(A): the BDD method formulated as the PCG applied for the solution of Eq. (107) with the preconditioner to Eq. (49) (Section 7.1). BDD(B): the BDD method formulated as the PCG applied for the solution of Eq. (47) with the preconditioner to Eq. (112) (Section 7.2).

the interface displacement field u_b of the structure at each iteration. This proof is here described for the case where no reorthogonalization is used, while it can easily be deduced that all relations that are proven between the variables of the two methods also hold in the case where reorthogonalization of the direction vectors is performed. In the course of this proof, subscript P is used in order to distinguish the variables of the standard BDD formulation from those of the formulation of Section 7.1.

The variables of the PCG initialization stage can be written as:

- For the algorithm of Section 7.1:

$$\hat{u}_b^0 = 0, \quad u_c^0 = (C^T \hat{S} C)^{-1} C^T \hat{f}_b, \quad u_b^0 = C u_c^0 = A_c \hat{f}_b, \tag{120}$$

$$r^0 = P_c^T \hat{f}_b, \quad z^0 = L_{p_b}^T S^{s^+} L_{p_b} r^0 \quad \text{and} \quad p^0 = z^0. \tag{121}$$

- For the standard BDD algorithm:

$$u_{b_P}^0 = A_c \hat{f}_b = u_b^0, \tag{122}$$

$$r_P^0 = \hat{f}_b - \hat{S} u_{b_P}^0 = \hat{f}_b - \hat{S} A_c \hat{f}_b = (I - \hat{S} A_c) \hat{f}_b = P_c^T \hat{f}_b = r^0, \tag{123}$$

$$z_P^0 = P_c L_{p_b}^T S^{s^+} L_{p_b} r^0 = P_c z^0 \quad \text{and} \quad p_P^0 = z_P^0 = P_c z^0 = P_c p^0. \tag{124}$$

In order to conclude the proof, we assume that the following relations between the variables of the two algorithms hold for the kth iteration of the PCG:

$$u_{b_P}^k = u_b^k = P_c \hat{u}_b^k + A_c \hat{f}_b, \quad r_P^k = r^k, \quad z_P^k = P_c z^k \quad \text{and} \quad p_P^k = P_c p^k. \tag{125}$$

The proof can thus be concluded by showing that the relations (125) hold for the next iteration $k+1$:

From Eq. (125), it follows that:

$$\eta_{\mathrm{P}}^{k}=\frac{z_{\mathrm{P}}^{k^{\mathrm{T}}}r_{\mathrm{P}}^{k}}{p_{\mathrm{P}}^{k^{\mathrm{T}}}q_{\mathrm{P}}^{k}}=\frac{z_{\mathrm{P}}^{k^{\mathrm{T}}}r_{\mathrm{P}}^{k}}{p_{\mathrm{P}}^{k^{\mathrm{T}}}\hat{S}p_{\mathrm{P}}^{k}}=\frac{z^{k^{\mathrm{T}}}P_{\mathrm{c}}^{\mathrm{T}}r^{k}}{p^{k^{\mathrm{T}}}P_{\mathrm{c}}^{\mathrm{T}}\hat{S}P_{\mathrm{c}}p^{k}}=\frac{z^{k^{\mathrm{T}}}r^{k}}{p^{k^{\mathrm{T}}}P_{\mathrm{c}}^{\mathrm{T}}\hat{S}p^{k}}=\eta^{k}, \tag{126}$$

$$u_{\mathrm{b_P}}^{k+1}=u_{\mathrm{b_P}}^{k}+\eta_{\mathrm{P}}^{k}p_{\mathrm{P}}^{k}=u_{\mathrm{b}}^{k}+\eta^{k}P_{\mathrm{c}}p^{k}=P_{\mathrm{c}}\hat{u}_{\mathrm{b}}^{k}+A_{\mathrm{c}}\hat{f}_{\mathrm{b}}+\eta^{k}P_{\mathrm{c}}p^{k}=P_{\mathrm{c}}(\hat{u}_{\mathrm{b}}^{k}+\eta^{k}p^{k})+A_{\mathrm{c}}\hat{f}_{\mathrm{b}}=P_{\mathrm{c}}\hat{u}_{\mathrm{b}}^{k+1}+A_{\mathrm{c}}\hat{f}_{\mathrm{b}}=u_{\mathrm{b}}^{k+1}, \tag{127}$$

$$r_{\mathrm{P}}^{k+1}=r_{\mathrm{P}}^{k}-\eta_{\mathrm{P}}^{k}q_{\mathrm{P}}^{k}=r_{\mathrm{P}}^{k}-\eta_{\mathrm{P}}^{k}\hat{S}p_{\mathrm{P}}^{k}=r^{k}-\eta^{k}\hat{S}P_{\mathrm{c}}p^{k}=r^{k}-\eta^{k}P_{\mathrm{c}}^{\mathrm{T}}\hat{S}p^{k}=r^{k}-\eta^{k}q^{k}=r^{k+1}, \tag{128}$$

$$z_{\mathrm{P}}^{k+1}=P_{\mathrm{c}}L_{\mathrm{p_b}}^{\mathrm{T}}S^{s^{+}}L_{\mathrm{p_b}}r_{\mathrm{P}}^{k+1}=P_{\mathrm{c}}L_{\mathrm{p_b}}^{\mathrm{T}}S^{s^{+}}L_{\mathrm{p_b}}r^{k+1}=P_{\mathrm{c}}z^{k+1}, \tag{129}$$

$$p_{\mathrm{P}}^{k+1}=z_{\mathrm{P}}^{k+1}+\frac{z_{\mathrm{P}}^{k+1^{\mathrm{T}}}r_{\mathrm{P}}^{k+1}}{z_{\mathrm{P}}^{k^{\mathrm{T}}}r_{\mathrm{P}}^{k^{\mathrm{T}}}}p_{\mathrm{P}}^{k}=P_{\mathrm{c}}z^{k+1}+\frac{z^{k+1^{\mathrm{T}}}P_{\mathrm{c}}^{\mathrm{T}}r^{k+1}}{z^{k^{\mathrm{T}}}P_{\mathrm{c}}^{\mathrm{T}}r^{k^{\mathrm{T}}}}P_{\mathrm{c}}p^{k}=P_{\mathrm{c}}\left(z^{k+1}+\frac{z^{k+1^{\mathrm{T}}}r^{k+1}}{z^{k^{\mathrm{T}}}r^{k^{\mathrm{T}}}}p^{k}\right)=P_{\mathrm{c}}p^{k+1}. \tag{130}$$

Eq. (130) concludes recursively this proof. It is therefore clear that the two formulations are equivalent.

8. Connection between the balancing domain decomposition method and the FETI method

The research in both the BDD and FETI families of methods has been progressing in parallel over the last decade and both methods have been applied in several areas of structural mechanics as well as in other fields of computational mechanics. Some efforts have been made in the past to interconnect the two methods, i.e. [11]. In this section, we first prove that the basic formulation of the BDD method [2] is equivalent to P-FETI1, with matrix Q set equal to the Dirichlet preconditioner. We then discuss the impact of this equivalence to the general comparison of BDD and FETI.

In order to prove that BDD and P-FETI1 (with matrix Q set equal to the Dirichlet preconditioner) are equivalent, it suffices to show that their preconditioning matrices are equal. When Q is set equal to the Dirichlet preconditioner (68), operator H_{b}, which is used in the formulation of the P-FETI1 preconditioner of Eq. (79), becomes

$$\begin{aligned}H_{\mathrm{b}}&=I-B_{\mathrm{b}}^{\mathrm{T}}QG(G^{\mathrm{T}}QG)^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}}\\&=I-(B_{\mathrm{b}}^{\mathrm{T}}B_{\mathrm{p_b}})S^{s}(B_{\mathrm{p_b}}^{\mathrm{T}}B_{\mathrm{b}})R_{\mathrm{b}}^{s}(R_{\mathrm{b}}^{s^{\mathrm{T}}}(B_{\mathrm{b}}^{\mathrm{T}}B_{\mathrm{p_b}})S^{s}(B_{\mathrm{p_b}}^{\mathrm{T}}B_{\mathrm{b}})R_{\mathrm{b}}^{s})^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}}\\&=I-A_{\mathrm{R}}(R_{\mathrm{b}}^{s^{\mathrm{T}}}A_{\mathrm{R}})^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}},\end{aligned} \tag{131}$$

where

$$A_{\mathrm{R}}=(B_{\mathrm{b}}^{\mathrm{T}}B_{\mathrm{p_b}})S^{s}(B_{\mathrm{p_b}}^{\mathrm{T}}B_{\mathrm{b}})R_{\mathrm{b}}^{s}. \tag{132}$$

Using Eqs. (27), (45) and setting the interface modes matrix C equal to Eq. (115), we obtain

$$\begin{aligned}A_{\mathrm{R}}&=(I-L_{\mathrm{p_b}}L_{\mathrm{b}}^{\mathrm{T}})S^{s}(I-L_{\mathrm{b}}L_{\mathrm{p_b}}^{\mathrm{T}})R_{\mathrm{b}}^{s}\\&=(L_{\mathrm{p_b}}L_{\mathrm{b}}^{\mathrm{T}}-I)S^{s}(L_{\mathrm{b}}L_{\mathrm{p_b}}^{\mathrm{T}})R_{\mathrm{b}}^{s}\\&=(L_{\mathrm{p_b}}L_{\mathrm{b}}^{\mathrm{T}}-I)S^{s}L_{\mathrm{b}}C\\&=(L_{\mathrm{p_b}}(L_{\mathrm{b}}^{\mathrm{T}}S^{s}L_{\mathrm{b}})-S^{s}L_{\mathrm{b}})C\\&=(L_{\mathrm{p_b}}\hat{S}-S^{s}L_{\mathrm{b}})C\end{aligned} \tag{133}$$

and similarly

$$\begin{aligned} R_{\mathrm{b}}^{s^{\mathrm{T}}} A_{\mathrm{R}} &= R_{\mathrm{b}}^{s^{\mathrm{T}}} (L_{\mathrm{p_b}} \hat{S} - S^s L_{\mathrm{b}}) C \\ &= (R_{\mathrm{b}}^{s^{\mathrm{T}}} L_{\mathrm{p_b}} \hat{S} - R_{\mathrm{b}}^{s^{\mathrm{T}}} S^s L_{\mathrm{b}}) C \\ &= C^{\mathrm{T}} \hat{S} C. \end{aligned} \tag{134}$$

From Eqs. (133) and (134), we have

$$\begin{aligned} H_{\mathrm{b}} L_{\mathrm{p_b}} &= (I - A_{\mathrm{R}} (R_{\mathrm{b}}^{s^{\mathrm{T}}} A_{\mathrm{R}})^{-1} R_{\mathrm{b}}^{s^{\mathrm{T}}}) L_{\mathrm{p_b}} \\ &= L_{\mathrm{p_b}} - A_{\mathrm{R}} (R_{\mathrm{b}}^{s^{\mathrm{T}}} A_{\mathrm{R}})^{-1} C^{\mathrm{T}} \\ &= L_{\mathrm{p_b}} - (L_{\mathrm{p_b}} \hat{S} - S^s L_{\mathrm{b}}) C (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} \\ &= L_{\mathrm{p_b}} - (L_{\mathrm{p_b}} \hat{S} - S^s L_{\mathrm{b}}) A_{\mathrm{c}} \\ &= L_{\mathrm{p_b}} (I - \hat{S} A_{\mathrm{c}}) + S^s L_{\mathrm{b}} A_{\mathrm{c}} \\ &= L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}} + S^s L_{\mathrm{b}} A_{\mathrm{c}}. \end{aligned} \tag{135}$$

The P-FETI1 preconditioner of Eq. (79) with Q set equal to the Dirichlet preconditioner thus becomes

$$\begin{aligned} \tilde{A}^{-1} &= L_{\mathrm{p_b}}^{\mathrm{T}} H_{\mathrm{b}}^{\mathrm{T}} S^{s^+} H_{\mathrm{b}} L_{\mathrm{p_b}} \\ &= (A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s + P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}}) S^{s^+} (L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}} + S^s L_{\mathrm{b}} A_{\mathrm{c}}) \\ &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}} + A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s S^{s^+} S^s L_{\mathrm{b}} A_{\mathrm{c}} + P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} S^s L_{\mathrm{b}} A_{\mathrm{c}} + A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s S^{s^+} L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}}. \end{aligned} \tag{136}$$

From Eq. (46) it follows that matrix A_{c} satisfies the relation:

$$\begin{aligned} A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s S^{s^+} S^s L_{\mathrm{b}} A_{\mathrm{c}} &= A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s L_{\mathrm{b}} A_{\mathrm{c}} = A_{\mathrm{c}} \hat{S} A_{\mathrm{c}} \\ &= C (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} \hat{S} C (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} \\ &= C (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} = A_{\mathrm{c}}. \end{aligned} \tag{137}$$

Furthermore, from Eqs. (27), (44) and (109) it follows that there is a matrix Y such that the following relations hold:

$$\begin{aligned} P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} S^s L_{\mathrm{b}} A_{\mathrm{c}} &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} (I + R_{\mathrm{b}} Y) L_{\mathrm{b}} A_{\mathrm{c}} = P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} L_{\mathrm{b}} A_{\mathrm{c}} + P_{\mathrm{c}} C Y = P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} L_{\mathrm{b}} A_{\mathrm{c}} \\ &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} L_{\mathrm{b}} C (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} = P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} L_{\mathrm{b}} L_{\mathrm{p_b}}^{\mathrm{T}} R_{\mathrm{b}}^s (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} \\ &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} (I - B_{\mathrm{p_b}}^{\mathrm{T}} B_{\mathrm{b}}) R_{\mathrm{b}}^s (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} \\ &= P_{\mathrm{c}} (L_{\mathrm{p_b}}^{\mathrm{T}} - L_{\mathrm{p_b}}^{\mathrm{T}} B_{\mathrm{p_b}}^{\mathrm{T}} B_{\mathrm{b}}) R_{\mathrm{b}}^s (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} \\ &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} R_{\mathrm{b}}^s (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} = P_{\mathrm{c}} C (C^{\mathrm{T}} \hat{S} C)^{-1} C^{\mathrm{T}} = 0. \end{aligned} \tag{138}$$

Finally, using Eqs. (137) and (138), Eq. (136) takes the form

$$\begin{aligned} \tilde{A}^{-1} &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}} + A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s S^{s^+} S^s L_{\mathrm{b}} A_{\mathrm{c}} + P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} S^s L_{\mathrm{b}} A_{\mathrm{c}} + A_{\mathrm{c}} L_{\mathrm{b}}^{\mathrm{T}} S^s S^{s^+} L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}} \\ &= P_{\mathrm{c}} L_{\mathrm{p_b}}^{\mathrm{T}} S^{s^+} L_{\mathrm{p_b}} P_{\mathrm{c}}^{\mathrm{T}} + A_{\mathrm{c}}. \end{aligned} \tag{139}$$

It therefore follows that the P-FETI1 preconditioner (79) with Q set equal to the Dirichlet preconditioner is equivalent to the BDD preconditioner (Eq. (112) with C equal to Eq. (115)).

Furthermore, the similarity between the eigenspectrum of FETI-1 and P-FETI1 (which has been discussed in Section 6.1), combined with the equivalence between P-FETI1 (with Q equal to the Dirichlet preconditioner) and BDD, lead to the conclusion that the BDD method and the Dirichlet preconditioned FETI-1 method (with Q equal to the Dirichlet preconditioner) should have comparable CPU time counts.

Indeed, the above configuration of FETI-1 and BDD appear to have comparable performance, as is later shown in Section 9 where numerical tests are performed. The connection established in this section thus allows to predict the relative performance of the one-level FETI method and the basic formulation of the BDD method [2]. For example, in the case of highly ill-conditioned problems (where the optimal configuration of FETI-1 uses the Dirichlet preconditioner and Q equal to the Dirichlet preconditioner [10]), the BDD method and the optimal FETI-1 configuration for this problem should result in comparable solution times. For less ill-conditioned problems, a similar reasoning leads to conclusions regarding the relative performance of BDD and FETI.

Remark 1. This remark makes use of the equivalence between the BDD and the P-FETI1 method (with Q equal to the Dirichlet preconditioner), in order to show that like the BDD, in this particular instance of P-FETI1 it is also possible to avoid one of the two coarse problem solutions at each iteration. More precisely, in the BDD method, it is possible to avoid one of the two coarse problem solutions at each iteration by setting the initial vector of interface displacements equal to Eq. (116). Similarly, the choice of a u_{b}^0 vector equal to Eq. (116) has the same positive effect in the P-FETI1 method with Q equal to the Dirichlet preconditioner. Note that due to Eqs. (132) and (134), in the case of P-FETI1 the initial vector u_{b}^0 of Eq. (116) takes the form

$$\begin{aligned} u_{\mathrm{b}}^0 &= A_{\mathrm{c}}\hat{f}_{\mathrm{b}} = C(C^{\mathrm{T}}\hat{S}C)^{-1}C^{\mathrm{T}}\hat{f}_{\mathrm{b}} = C(R_{\mathrm{b}}^{s^{\mathrm{T}}}A_{\mathrm{R}})^{-1}C^{\mathrm{T}}\hat{f}_{\mathrm{b}} \\ &= L_{\mathrm{p_b}}^{\mathrm{T}}R_{\mathrm{b}}^{s}(R_{\mathrm{b}}^{s^{\mathrm{T}}}(B_{\mathrm{b}}^{\mathrm{T}}B_{\mathrm{p_b}})S^{s}(B_{\mathrm{p_b}}^{\mathrm{T}}B_{\mathrm{b}})R_{\mathrm{b}}^{s})^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}}L_{\mathrm{p_b}}\hat{f}_{\mathrm{b}} \\ &= L_{\mathrm{p_b}}^{\mathrm{T}}R_{\mathrm{b}}^{s}(G^{\mathrm{T}}QG)^{-1}R_{\mathrm{b}}^{s^{\mathrm{T}}}L_{\mathrm{p_b}}\hat{f}_{\mathrm{b}}. \end{aligned} \tag{140}$$

Hence, if the initial vector u_{b}^0 of P-FETI1 is set equal to Eq. (140), the residuals of P-FETI1 with Q equal to the Dirichlet preconditioner, satisfy the condition (117)

$$C^{\mathrm{T}}r^k = 0 \Rightarrow R_{\mathrm{b}}^{s^{\mathrm{T}}}L_{\mathrm{p_b}}r^k = 0, \quad k = 0, 1, \ldots \tag{141}$$

Due to Eq. (141), the following holds:

$$H_{\mathrm{b}}L_{\mathrm{p_b}}r^k = L_{\mathrm{p_b}}r^k, \quad k = 0, 1, \ldots \tag{142}$$

From Eq. (142), it can be deduced that with the initial vector u_{b}^0 of Eq. (140), the P-FETI1 preconditioner (79) may be implemented as follows:

$$\tilde{A}^{-1} = L_{\mathrm{p_b}}^{\mathrm{T}}H_{\mathrm{b}}^{\mathrm{T}}S^{s^{+}}L_{\mathrm{p_b}}, \tag{143}$$

where only one solution of the coarse problem is required within the preconditioning step. It is also worth noting the analogy between the P-FETI1 and FETI-1 method in this issue. In particular, as in P-FETI1, in FETI-1 it is also possible to remove the one of the two coarse problem solutions per iteration if matrix Q is set equal to the Dirichlet preconditioner [5].

9. Numerical performance of the introduced methods

In this section, numerical tests are performed in order to compare the performance of the DDM that have been discussed in this paper. In particular, the number of iterations required to reach different levels of solution accuracy, as well as the minimum achieved error are compared for the different methods. It is worth noting that each original Dirichlet preconditioned FETI variant discussed in this paper and its corresponding P-FETI variant have a similar computational cost in both the initialization phase and at each iteration. Indeed, they require the solution of the same local subdomain and coarse space problems at

each iteration. Only small differences in the CPU time of each iteration appear, like for example in the computational costs related to the reorthogonalization of the direction vectors. In original FETI methods the dimension of the direction vectors is equal to the number of Lagrange multipliers, while in P-FETI methods their dimension is equal to the number of interface d.o.f., which is typically smaller than the number of Lagrange multipliers in the presence of cross-points, particularly in 3-D elasticity problems and in the case of redundant Lagrange multipliers. However, such differences are relatively small. Therefore, in order to estimate which method is actually faster, it suffices to compare the iteration counts required for convergence.

Furthermore, it is worth investigating the solution precision that can be achieved by the discussed methods. In fact, DDM have repeatedly been used in the past to solve highly large-scale and ill-conditioned problems. It is thus important to investigate the convergence capabilities of the methods, in highly ill-conditioned problems. Therefore, the discussed DDM are also compared in this section in terms of the level of solution accuracy that can be achieved.

In order to compare the iteration counts and the minimum attainable error of the methods, we have run a number of test cases using a research code that has been developed for that purpose on the Matlab computing environment. In order to compute accurately the iteration counts and minimum residual, this code simulates the state of the art techniques used to implement the discussed DDM in FORTRAN, C or C++ programming languages for high performance applications (see for example the FETI implementation discussion in [18]). Hence, in the subsequently presented results, unless otherwise specified, the DDM are implemented as follows:

- All methods are implemented with reorthogonalization of the direction vectors.
- In FETI-1 and FETI-2 methods, we employ the algorithms that are derived after using the transformations of Eq. (67). Furthermore, the FETI-2 coarse problem matrix that is related to the cross-points is formed using the formulation presented in [18].
- It was found in a number of tests performed by the authors, that the choice of any of the three different definitions of Lagrange multipliers practically does not affect the convergence history of original FETI and P-FETI variants. Therefore, in the results presented in this section, redundant Lagrange multipliers are used for FETI-1, while non-redundant multipliers are used for the FETI-2 runs, in order to minimize the dimension of the FETI-2 coarse problem which is related to the cross-point Lagrange multipliers.
- The local subdomain problems of the implemented DDM are usually solved using the direct skyline solver. The Choleski factorization method is thus used in the performed tests with optimal node renumbering for skyline solvers. In addition, the zero energy modes and artificial constraints of semi-definite subdomain problems are obtained with the geometric–algebraic method presented in [23].
- In order to use the same convergence criterion for the different methods, convergence is monitored with the criterion:

$$\|f - Ku\|/\|f\| < \varepsilon, \tag{144}$$

  where ε is the convergence threshold parameter. Therefore, even though the discussed DDM require computing a full displacement field estimate only after convergence, it is computed here at each iteration in order to obtain the residual quotient of Eq. (144).
- A Pentium IV processor at 1.7 GHz has been used in the test runs.

9.1. A 3-D elasticity test problem

The 3-D elasticity problem of Fig. 3 is used as the first test problem. The cubic structure of Fig. 3 is composed of five layers of two different materials and is discretized with $28\times28\times28$ 8-node brick elements. Additionally, it is pinned at the four corners of its left surface and is subjected to gravity load. An optimal

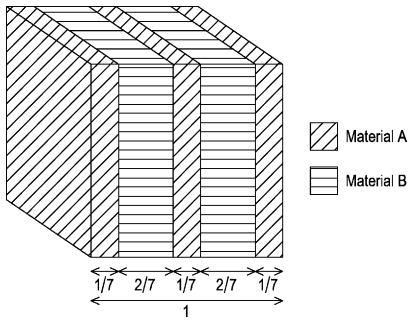

Fig. 3. A cubic structure composed of two materials.

decomposition of this heterogeneous problem must generate subdomains with good aspect ratios, while preserving the material interfaces when partitioning the model [10]. Hence, two decompositions of this heterogeneous model of 73,155 d.o.f. in 100 subdomains, obtained with TOPDOMDEC [24], are considered: In the first decomposition (Fig. 4a), the model has been partitioned in subdomains with good aspect ratios without taking into account the material interfaces (decomposition P1). In the second decomposition (Fig. 4b), the five layers of different materials have been partitioned independently, thus generating a decomposition which preserves the material interfaces but produces subdomains of suboptimal aspect ratio in the thin layers (decomposition P2).

The number of iterations required to reach different levels of solution accuracy and the maximum accuracy achieved by FETI-1 and P-FETI1 methods in this 3-D elasticity problem are presented in Table 2. Different ratios E_A/E_B of the Young moduli of the two materials are considered, while their Poisson ratio is set equal to $v_A = v_B = 0.30$. Table 2 thus presents the results obtained for homogeneous material properties using partitioning P1 and for heterogeneous material properties using either partitioning P1 or P2. In the homogeneous case, matrix Q is set equal to the unit matrix and homogeneous scaling is used, while in the heterogeneous case, the methods are implemented with heterogeneous scaling and Q is set equal to the superlumped matrix Q^{SL}.

The choice of the superlumped matrix is optimal for this problem, since the choices of $Q = Q^D$ or $Q = Q^L$ have been found not to improve significantly the iteration count of the methods, while they entail more computational cost in the initialization and in each iteration of the methods.

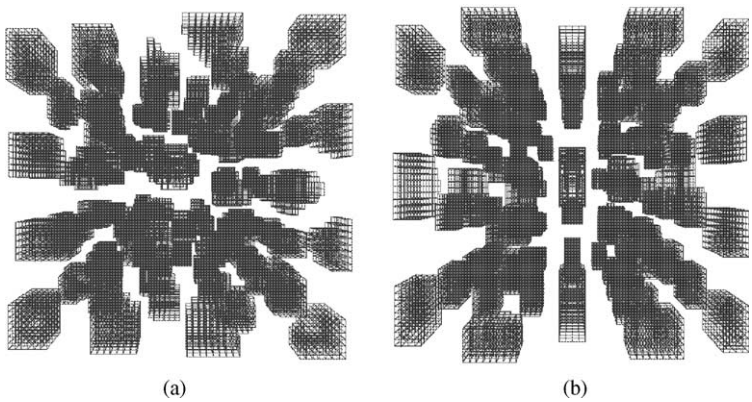

Fig. 4. Two decompositions of the cubic test problem in 100 subdomains: (a) optimal aspect ratio partitioning (decomposition P1), (b) layered partitioning (decomposition P2).

Table 2
Number of iterations (for error tolerance: 10^{-5}, 10^{-7} and 10^{-9}) and maximum attained precision achieved by FETI-1 and P-FETI1 methods for the 3-D elasticity test problem

E_A/E_B	Decomposition	FETI-1 lumped preconditioner				FETI-1 Dirichlet preconditioner				P-FETI1			
		10^{-5}	10^{-7}	10^{-9}	Minimum error	10^{-5}	10^{-7}	10^{-9}	Minimum error	10^{-5}	10^{-7}	10^{-9}	Minimum error
10^0	P1	32	39	47	2.1×10^{-10}	27	33	39	2.6×10^{-10}	27	33	39	5.2×10^{-11}
10^3	P1	56	67	79	8.4×10^{-10}	41	49	59	9.1×10^{-10}	39	48	56	5.1×10^{-10}
	P2	33	40	–	1.1×10^{-9}	27	32	–	1.2×10^{-9}	26	31	37	4.8×10^{-10}
10^6	P1	63	–	–	4.0×10^{-6}	48	–	–	3.6×10^{-6}	44	–	–	4.7×10^{-7}
	P2	37	–	–	5.0×10^{-6}	31	–	–	5.4×10^{-6}	29	–	–	4.3×10^{-7}

From the results of Table 2, it follows that in the homogeneous case P-FETI1 has the same iteration count as the FETI-1 method with the Dirichlet preconditioner. In the heterogeneous cases however, the P-FETI1 method performs 2–8% less iterations than the Dirichlet preconditioned FETI-1 method. It should be noted that decomposition P2 leads to less iterations than decomposition P1. Moreover, in the heterogeneous cases P-FETI1 achieves higher solution accuracy than FETI-1. In particular, the minimum errors of P-FETI1 and FETI-1 methods differ by one-half to one order of magnitude.

Similar results have also been observed in other 2-D and 3-D elasticity tests performed by the authors. More precisely, in most of the tests performed in homogeneous problems, P-FETI and original FETI methods with the Dirichlet preconditioner have the same iteration counts. However, in the heterogeneous test cases and in some homogeneous ones, P-FETI variants give smaller number of iterations than the Dirichlet preconditioned FETI variants with dual formulation. Additionally, even though P-FETI and original FETI methods reach approximately the same minimum error in some of the tests performed on heterogeneous problems, in most tests P-FETI methods achieve higher solution accuracy than the original FETI methods.

9.2. A semi-cylindrical panel problem

The semi-cylindrical panel of Fig. 5 is used in this section as a fourth-order test problem for comparing the DDM discussed in this paper. The radius of the panel is equal to 0.5 and its length is 1.6. Two configurations of this example are considered: In the first configuration, the thickness of the panel is set equal to $t = 10^{-3}$, while in the second it is set equal to $t = 10^{-4}$. Moreover, the Young modulus is 1×10^6 and the Poisson ratio 0.30. The panel is modeled with a structured mesh of 131×131 nodes and is discretized with triangular TRIC shell elements [25]. Furthermore, it is fixed on 16 nodes along its two linear edges as shown in Fig. 5 and a gravity load is applied. This model of 102,870 d.o.f. is decomposed in 130 subdomains (Fig. 6) using TOPDOMDEC [24].

For the two configurations of this test example, the number of iterations to reach several levels of solution accuracy and the maximum achieved accuracy are presented in Table 3 for different variants of original FETI and P-FETI methods. The Dirichlet preconditioner and homogeneous scaling are used in the presented results. Furthermore, in the two-level FETI method the displacement compatibility condition (85) is satisfied at each iteration for all three translational components of the displacement vector of the interface nodes that are selected for the coarse space. Moreover, in dual–primal FETI, displacement

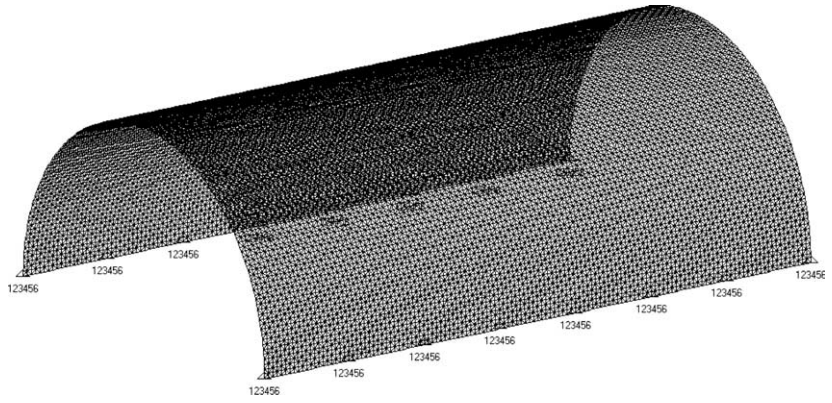

Fig. 5. A semi-cylindrical panel, discretized with triangular shell elements.

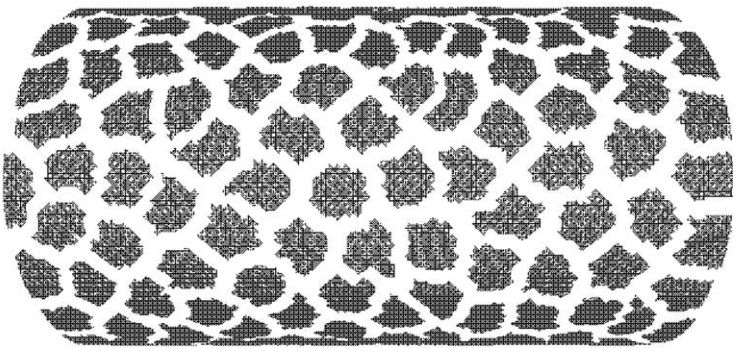

Fig. 6. A decomposition of the semi-cylindrical panel in 130 subdomains—top view.

compatibility is preserved for both translational and rotational components of the displacements of the coarse space interface nodes.

From the results of Table 3, it follows that FETI methods with dual formulation perform more iterations than their P-FETI alternatives. In particular, in all cases original FETI formulations result in 5–14% more iterations than the corresponding primal formulations, in order to reach the same solution accuracy. Furthermore, the P-FETI variants reach lower errors than the corresponding original dual variants. The minimum errors of original FETI and P-FETI methods in fact differ by approximately one-half to four orders of magnitude. In addition, comparing the minimum errors attained by the original one-level, two-level and dual–primal FETI methods in Table 3, it follows that FETI-DP achieves higher solution accuracy, than both FETI-1 and FETI-2 methods, while FETI-2 is inferior in terms of solution accuracy than the three variant FETI methods. However, the respective P-FETI variants reach approximately the same solution precision.

It should be further noted that in a number of plate and shell problems tested by the authors, a similar reduction in iteration counts and minimum errors has been observed for the P-FETI formulations. More precisely, in all tests performed in fourth-order problems, P-FETI methods give less iterations than the corresponding FETI methods with dual formulation. Furthermore, even though in a few of the performed tests the two formulations achieve almost the same maximum solution accuracy, in most of the tests performed so far, P-FETI configurations attain higher solution precision. Thus, the P-FETI methods appear capable of providing more accurate solutions in a number of ill-conditioned plate and shell problems.

The differences that typically appear in the convergence history of original FETI and P-FETI methods in fourth-order problems are depicted in Fig. 7, which illustrates the solution accuracy versus the number of iterations of FETI-1 and P-FETI1 methods, for the semi-cylindrical panel problem. In Fig. 7 it can be observed that the value of the FETI-1 initial error is high, compared to the value of the P-FETI1 initial error which lies in the vicinity of 1. After a number of iterations, the convergence curves of the two methods become almost parallel, while the P-FETI1 error remains always lower than the FETI-1 error. Finally, FETI-1 stagnates at a higher level than P-FETI1. The above mentioned differences are typically observed when comparing the dual and the corresponding primal methods in fourth-order or heterogeneous second-order problems, while in homogeneous second-order problems, the convergence curves of original FETI methods and their P-FETI alternatives typically coincide after a number of initial iterations and finally stagnate at approximately the same solution error.

Moreover, it can be seen from Table 3 that in FETI-1 and P-FETI1 methods setting matrix Q equal to the superlumped choice ($Q = Q^{SL}$) or equal to the Dirichlet preconditioner, results in a substantial reduction of the iteration counts, compared to the choice of $Q = I$. In particular, both P-FETI1 and FETI-1

Table 3
Number of iterations and maximum precision of original FETI and P-FETI methods for the two configurations of the semi-cylindrical panel

	$t = 10^{-3}$[a]								$t = 10^{-4}$[a]							
	Original FETI				P-FETI				Original FETI				P-FETI			
	10^{-2}	10^{-4}	10^{-6}	Minimum error	10^{-2}	10^{-4}	10^{-6}	Minimum error	10^{-2}	10^{-3}	10^{-4}	Minimum error	10^{-2}	10^{-3}	10^{-4}	Minimum error
One-level FETI																
$Q = I$	341	372	–	6.9×10^{-6}	318	354	385	7.9×10^{-8}	>500				>500			
$Q = Q^{SL}$	197	224	–	6.0×10^{-6}	181	212	237	7.0×10^{-8}	424	–	–	1.7×10^{-3}	395	408	425	7.1×10^{-5}
$Q = Q^{D}$ – two coarse problem solutions per iteration	132	156	–	6.7×10^{-6}	118	144	169	4.6×10^{-8}	330	–	–	1.9×10^{-3}	297	319	340	6.1×10^{-5}
$Q = Q^{D}$ – one coarse problem solutions per iteration	132	159	–	3.6×10^{-5}	118	144	169	1.7×10^{-7}	–	–	–	9.5×10^{-1}	297	319	–	2.4×10^{-4}
Two-level FETI																
Coarse problem related only to cross-points	43	54	–	1.1×10^{-5}	38	50	61	1.8×10^{-8}	–	–	–	2.1×10^{-1}	99	111	123	2.7×10^{-5}
Coarse problem related to cross-points and ends of interface edges	34	44	–	1.1×10^{-5}	30	40	51	1.5×10^{-8}	–	–	–	9.7×10^{-2}	80	91	102	1.7×10^{-5}
Dual–primal FETI																
Coarse problem related only to cross-points	72	86	101	2.7×10^{-7}	64	80	94	4.2×10^{-8}	186	202	–	4.4×10^{-4}	165	180	196	2.8×10^{-5}
Coarse problem related to cross-points and ends of interface edges	46	59	71	1.1×10^{-7}	41	54	66	6.3×10^{-8}	142	157	–	1.5×10^{-4}	125	139	153	4.2×10^{-5}

[a] Panel thickness.

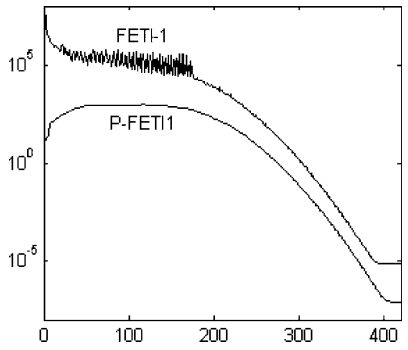

Fig. 7. Error quotient of Eq. (144) versus number of iterations for FETI-1 and P-FETI1 methods with $Q = I$ with for the semi-cylindrical panel with $t = 10^{-3}$.

with Q set equal to the Dirichlet preconditioner perform 58–63% less iterations than when choosing $Q = I$. A similar reduction in FETI-1 iteration counts has also been observed in fourth-order problems by Rixen [26]. Based on these remarks, it follows that setting $Q \neq I$ can substantially improve the computational time required by FETI-1 and P-FETI1 methods, for the solution of plate and shell problems. It is also worth noting that the results reported in Table 3 for P-FETI1 with $Q = Q^{D}$, are practically the same as those that we have obtained for the standard BDD method [2], due to the equivalence between this configuration of the P-FETI1 method and the BDD method. Moreover, it should be noted that in FETI-1 and P-FETI1 methods with Q set equal to the Dirichlet preconditioner, when omitting the one of the two coarse problem solutions per iteration, a decrease in the maximum achieved solution accuracy is observed. In particular, Table 3 shows that a decrease of one-half to almost three orders of magnitude to the maximum accuracy is caused by skipping the one coarse solution at each iteration.

10. Summary and concluding remarks

In this paper novel DDM are introduced, while previously known DDM are classified using a unified framework for primal and dual DDM. The developed framework is independent from the choice of homogeneous or heterogeneous scaling and the definition of the Lagrange multipliers.

Moreover, a new class of methods (P-FETI class of methods) is proposed based on the respective existing FETI variants. Furthermore, the introduced P-FETI methods are shown to be closely related to the original FETI variants. In particular, the one-level FETI method (FETI-1) has a similar eigenspectrum as the corresponding P-FETI method (P-FETI1). This finding and the similarity in the way different P-FETI methods are formulated from the original FETI methods, lead to the conclusion that there probably exist theoretical connections, similar to those between P-FETI1 and FETI-1, for the other FETI variants. Furthermore, two alternative formulations of the BDD method are described, while a connection between the BDD and FETI methods is established.

Additionally, the computational efficiency of the P-FETI methods is compared with the corresponding original FETI methods in a number of numerical tests. The obtained results show that in homogeneous second-order problems the previously known Dirichlet preconditioned FETI methods and the respective P-FETI variants have approximately the same computational cost. In fourth-order and heterogeneous second-order problems, however, the original FETI methods with the Dirichlet preconditioner are typically slower than the corresponding P-FETI configurations. Furthermore, the P-FETI methods typically attain higher solution accuracy in these problems, as well. They thus provide a more robust category of methods

for solving ill-conditioned fourth-order and heterogeneous second-order problems. In general, the introduced P-FETI methods retain the thoroughly investigated in the past years optimal convergence properties of the FETI method, while they appear capable of providing faster and more accurate solutions in a number of problems.

In addition, general remarks drawn from the present study may be summarized to the following. It is well known that each DDM is basically characterized by the quality of its coarse problem. The importance of the coarse space leads to a classification of the existing and introduced methods according to the type of coarse problem employed and its position in the method's formulation. The thus obtained classification is schematically depicted in Fig. 2. This figure shows that for each coarse problem, two formulations can be obtained, placing the coarse problem either in the linear problem solved by the PCG or in the preconditioner. The two formulations lead to methods with either similar or different theoretical and numerical properties, depending on the type of problem at hand, as is shown in the sections of this paper.

Acknowledgements

The authors would like to thank Dimos Charmpis for developing a part of the Matlab DDM code employed in this paper. The first author would further like to thank his collaborators Sotiris Bitzarakis, Constantina Apostolopoulou and George Stavroulakis, for their advice and help in issues regarding the FETI method. Finally, this work is partially supported by the research project "Archimedes" of the National Technical University of Athens.

References

[1] C. Farhat, F.-X. Roux, A method of finite element and interconnecting and its parallel solution algorithm, Int. J. Numer. Meth. Engrg. 32 (1991) 1205–1227.
[2] J. Mandel, Balancing domain decomposition, Commun. Appl. Numer. Meth. 9 (1993) 233–241.
[3] Y. Fragakis, M. Papadrakakis, A unified framework for formulating domain decomposition methods in structural mechanics, Technical Report, National Technical University of Athens, Greece, March 2002. Available from <http://users.ntua.gr/fragayan>.
[4] Y. Fragakis, M. Papadrakakis, A family of new FETI methods for structural mechanics, in: Proceedings of the Fifth World Congress on Computational Mechanics (WCCMV), Vienna, Austria, July 7–12, 2002. Available from <http://wccm.tuwien.ac.at>.
[5] C. Farhat, F.-X. Roux, Implicit parallel processing in structural mechanics, Comput. Mech. Adv. 2 (1994) 1–124.
[6] K.C. Park, M.R. Justino Jr., C.A. Felippa, An algebraically partitioned FETI method for parallel structural analysis: algorithm description, Int. J. Numer. Meth. Engrg. 40 (1997) 2717–2737.
[7] M.R. Justino Jr., K.C. Park, C.A. Felippa, An algebraically partitioned FETI method for parallel structural analysis: implementation and numerical performance evaluation, Int. J. Numer. Meth. Engrg. 40 (1997) 2739–2758.
[8] D. Rixen, C. Farhat, R. Tezaur, J. Mandel, Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver, Int. J. Numer. Meth. Engrg. 46 (1999) 501–534.
[9] D. Rixen, C. Farhat, A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems, Int. J. Numer. Meth. Engrg. 44 (4) (1999) 489–516.
[10] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, D. Rixen, Application of the FETI method to ASCI problems: scalability results on one-thousand processors and discussion of highly heterogeneous problems, Int. J. Numer. Meth. Engrg. 47 (2000) 513–536.
[11] A. Klawonn, O.B. Widlund, FETI and Neumann–Neumann iterative substructuring methods: connection and new results, Commun. Pure Appl. Math. 54 (2001) 57–90.
[12] P. LeTallec, M. Vidrascu, Solving large scale structural problems on parallel computers using domain decomposition techniques, in: M. Papadrakakis (Ed.), Parallel Solution Methods in Computational Mechanics, John Wiley & Sons, New York, 1997.
[13] S. Bitzarakis, M. Papadrakakis, A. Kotsopoulos, Parallel solution techniques in computational structural mechanics, Comput. Methods Appl. Mech. Engrg. 148 (1997) 75–104.
[14] C. Farhat, J. Mandel, The two-level FETI method for static and dynamic plate problems—part I: an optimal iterative solver for biharmonic systems, Comput. Methods Appl. Mech. Engrg. 155 (1998) 129–152.

[15] C. Farhat, P.S. Chen, J. Mandel, F.X. Roux, The two-level FETI method—part II: extension to shell problems, parallel implementation and performance results, Comput. Methods Appl. Mech. Engrg. 155 (1998) 153–180.
[16] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, D. Rixen, FETI-DP: A dual–primal unified FETI method—part I: a faster alternative to the two-level FETI method, Int. J. Numer. Meth. Engrg. 50 (2001) 1523–1544.
[17] C. Farhat, M. Lesoinne, K. Pierson, A scalable dual–primal domain decomposition method, Numer. Linear Alg. Appl. 7 (7–8) (2000) 687–714.
[18] C. Farhat, K. Pierson, M. Lesoinne, The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems, Comput. Methods Appl. Mech. Engrg. 184 (2000) 333–374.
[19] D.C. Charmpis, M. Papadrakakis, Enhancing the performance of the FETI method with preconditioning techniques implemented on clusters of networked computers, Comput. Mech. 30 (2002) 12–28.
[20] C. Farhat, J. Mandel, F.-X. Roux, Optimal convergence properties of the FETI domain decomposition method, Comput. Methods Appl. Mech. Engrg. 115 (1994) 365–385.
[21] C. Farhat, P.-S. Chen, F. Risler, F.X. Roux, A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers, Int. J. Numer. Meth. Engrg. 42 (1998) 257–288.
[22] P. LeTallec, J. Mandel, M. Vidrascu, A Neumann–Neumann domain decomposition algorithm for solving plate and shell problems, SIAM J. Numer. Anal. 35 (1998) 836–867.
[23] M. Papadrakakis, Y. Fragakis, An integrated geometric–algebraic method for solving semi-definite problems in structural mechanics, Comput. Methods Appl. Mech. Engrg. 190 (49–50) (2001) 6513–6532.
[24] M. Sharp, C. Farhat, TOPDOMDEC—A totally object oriented program for visualization, domain decomposition and parallel processing, User's manual, PGSoft and University of Colorado, Boulder, USA, 1994.
[25] J. Argyris, L. Tenek, M. Papadrakakis, C. Apostolopoulou, Postbuckling performance of the TRIC natural mode triangular shell element for isotropic and laminated composite shells, Comput. Methods Appl. Mech. Engrg. 166 (1998) 211–231.
[26] D. Rixen, Choosing the coarse grid when applying FETI to heterogeneous structures, Presentation at the 13th International Conference on Domain Decomposition Methods, Lyon, France, October 9–12, 2000.