Non-overlapping domain decomposition methods in structural mechanics

Pierre Gosselet

LMT Cachan
ENS Cachan / Université Pierre et Marie Curie / CNRS UMR 8535
61 Av. Président Wilson
94235 Cachan FRANCE
Email: gosselet@lmt.ens-cachan.fr

Christian Rey

LMT Cachan
ENS Cachan / Université Pierre et Marie Curie / CNRS UMR 8535
61 Av. Président Wilson
94235 Cachan FRANCE
Email: rey@lmt.ens-cachan.fr

Summary

The modern design of industrial structures leads to very complex simulations characterized by nonlinearities, high heterogeneities, tortuous geometries... Whatever the modelization may be, such an analysis leads to the solution to a family of large ill-conditioned linear systems. In this paper we study strategies to efficiently solve to linear system based on non-overlapping domain decomposition methods. We present a review of most employed approaches and their strong connections. We outline their mechanical interpretations as well as the practical issues when willing to implement and use them. Numerical properties are illustrated by various assessments from academic to industrial problems. An hybrid approach, mainly designed for multifield problems, is also introduced as it provides a general framework of such approaches.

1 INTRODUCTION

Hermann Schwarz (1843-1921) is often referred to as the father of domain decomposition methods. In a 1869-paper he proposed an alternating method to solve a PDE equation set on a complex domain composed the overlapping union of a disk and a square (fig. 1), giving the mathematical basis of what is nowadays one of the most natural ways to benefit the modern hardware architecture of scientific computers.

In fact the growing importance of domain decomposition methods in scientific computation is deeply linked to the growth of parallel processing capabilities (in terms of number of processors, data exchange bandwidth between processors, parallel library efficiency, and of course performance of each processor). Because of the exponential increase of computational resource requirement for numerical simulation of more and more complex physical phenomena (non-linearities, couplings between physical mechanisms or between physical scales, random variables...) and more and more complex problems (optimization, inverse problems...), parallel processing appears to be an essential tool to handle the resulting numerical models.

Parallel processing is supposed to take care of two key-points of modern computations, the amount of operations and the required memory. Let us consider the simulation of a physical phenomenon, classically modeled by a PDE $\mathcal{L}(x) = f$, $x \in H(\Omega)$. To take advantage of the parallel architecture of a calculator, a reflexion has to be carried out on how the original problem could be decomposed into collaborating subprocesses. The criteria for this decomposition will be: first, the ability to solve independent problems (on independent processors); second, how often processes have to be synchronized; and last what quantity of data has to be exchanged when synchronizing. When tracing back the idle time of resolution processes and analyzing hardware and software solutions, it is often
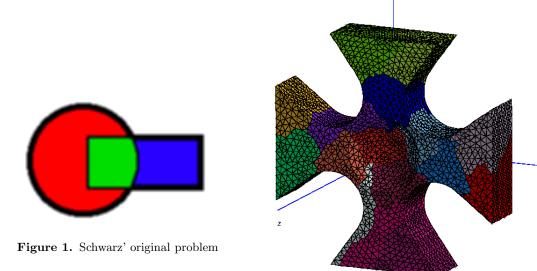
Figure 1. Schwarz' original problem

Figure 2. 16 subdomains bitraction test specimen
(courtesy of ONERA – Pascale Kanouté)

observed that inter-processor communications are the most penalizing steps.

If we now consider the three great classes of mathematical decomposition of our reference problem which are operator splitting (for instance $\mathcal{L} = \sum_i \mathcal{L}_i$ [47]), function-space decomposition (for instance $H(\Omega) = \operatorname{span}(v_i)$, an example of which is modal decomposition) and domain decomposition ($\Omega = \bigcup \Omega_i$), though the first two can lead to very elegant formulations, only domain decompositions ensure (once one subdomain or more have been associated to one processor) that independent computations are limited to small quantities and that the data to exchange is limited to the interface (or small overlap) between subdomains which is always one-to-one exchange of small amount of data.

So domain decomposition methods perfectly fit the criteria for building efficient algorithms running on parallel computers. Their use is very natural in engineering (and more precisely design) context : domain decomposition methods offer a framework where different design services can provide the virtual models of their own parts of a structure, each assessed independently, domain decomposition can then evaluate the behavior of the complete structure just setting specific behavior on the interface (perfect joint, unilateral contact, friction). Of course domain decomposition methods also work with one-piece structure (for instance fig. 2), then decomposition can be automated according to criteria which will be discussed later.

From an implementation point of view, programming domain decomposition methods is not an overwhelming task. Most often it can be added to existing solvers as a upper level of current code using the existing code as a black-box. The only requirement to implement domain decomposition is to be able to detect the interface between subdomains and use a protocol to share data on this common part. In this paper we will mostly focus on domain decomposition methods applied to finite element method [15, 109], anyhow they can be applied to any discretization method (among others meshless methods [5, 9, 73] and discrete element methods [20, 8, 22]).

Though domain decomposition methods were more than one century old, they had not

been extensively studied. Recent interest arose as they were understood to be well-suited to modern engineering and modern computational hardware. An important date in reinterest in domain decomposition methods is 1987 as first international congress dedicated to these methods occurred and DDM association was created (see `http://www.ddm.org`).

Yet the studies were first mathematical analysis oriented and emphasized on Schwarz overlapping family of algorithms. As interest in engineering problems grew, non-overlapping Schwarz and Schur methods, and coupling with discretization methods (mainly finite element) were more and more studied. Indeed, these methods are very natural to interpret mechanically, and moreover mechanical considerations often resulted in improvement to the methods. Basically the notion of interface between neighboring subdomains is a strong physical concept, to which is linked a set of conservation principles and phenomenological laws: for instance the conservation of fluxes (action-reaction principle) imposes the point-wise mechanical equilibrium of the interface and the equality of incoming mass (heat...) from one subdomain to the outgoing mass (heat...) of its neighbors; the "perfect interface" law consists in supposing that displacement field (pressure, temperature) is continuous at the interface, contact laws enable disjunction of subdomains but prohibit interpenetration.

In this context two methods arose in the beginning of the 90's : so-called Finite Element Tearing and Interconnecting (FETI) method [41] and Balanced Domain Decomposition (BDD) [104]. From a mechanical point of view BDD consists in choosing the interface displacement field as main unknown while FETI consists in privileging the interface effort field. BDD is usually referred to as a primal approach while FETI is a dual approach. One of the interests of these methods, beyond their simple mechanical interpretation, is that they can easily be explained from a purely algebraic point of view (*ie* directly from the matrix form of the problem). In order to fit parallelization criteria, it clearly appeared that the interface problem should be solved using an iterative solver, each iteration requiring local (*ie* independent on each subdomain) resolution of finite element problem, which could be done with a direct solver. Then these methods combined direct and iterative solver trying to mix robustness of the first and cheapness of the second. Moreover the use of an iterative solver was made more efficient by the existence of relevant preconditioners (based on the resolution of a local dual problem for the primal approach and a primal local problem for the dual approach).

When first released, FETI could not handle floating substructures (*ie* substructures without enough Dirichlet conditions), thus limiting the choice of decomposition, while the primal approach could handle such substructures but with loss of scalability (convergence decayed as the number of floating substructures increased). A key point then was the introduction of rigid body motions as constraints and the use of generalized inverses. Because of its strong connections with multigrid methods [108], the rigid body motions constraint took the name of "coarse problem", it made the primal and dual methods able to handle most decompositions without loss of scalability [43, 74, 102]. From a mechanical point of view, the coarse problem enables non-neighboring subdomains to interact without requiring the transmission of data through intermediate subdomains, it then enables to spread global information on the whole structure scale.

Once equipped with their best preconditioners and coarse problems, mathematical results [39, 65, 10] provide theoretical scalability of the methods. For instance for 3D elasticity problems, if h is the diameter of finite elements and H the diameter of subdomains, condition number κ of the interface problem reads (C is a real constant):

$$\kappa \simeq C \left(1 + \log \left(\frac{H}{h} \right) \right)^2 \tag{1}$$

which proves that the condition number only depends logarithmatically on the number of

elements per subdomain. Many numerical assessment campaigns confirmed the good properties of the methods, their robustness compared to iterative solvers applied to the complete structure and their low cost (in terms of memory and CPU requirements) compared to direct solvers. Thus because they are well-suited to modern hardware (like PC clusters) they enable to achieve computations which could not be realized on classical computers because of too high memory requirement or too long computational time: these methods can handle problems with several millions of degrees of freedom.

Primal and dual methods were extended to heterogeneous problems by a cheap intervention on the preconditioners [91] and on the initialization [60], and to forth order elasticity (plates and shells) problems by the adjunction of so-called "second level problem" in order to regularize the displacement field around the corners of subdomains [38, 103, 40]. As it became clear that the regularization of the displacement field was sufficient to suppress rigid body motions, specific algorithms which regularized *a priori* the subdomain problems were proposed: FETIDP [34] and its primal counterpart BDDC [19], first in the plates and shells context, then in the second order elasticity context [63]. Now FETIDP and BDDC are considered as efficient as original FETI and BDD.

Methods were employed in many other contexts: transient dynamics [50], multifield problems (multiphysic problems such as porous media [56] and constrained problems such as incompressible flows [70, 55]), Helmotz equations [21, 37, 36] and contact [27, 26]. The use of domain decomposition methods in structural dynamic analysis is a rather old idea though it can now be confronted to well established methods in static analysis; the Craig-Bampton algorithm [17] is somehow the application of the primal strategy to such problems, the dual version of which was proposed in [90], moreover ideas like the adjunction of coarse problems enabled to improve these methods.

Because of the strong connection between primal and dual approaches, some methods try to propose frameworks which generalize the two methods. The hybrid approach [61, 57] enables to select specific treatment (primal or dual) for each interface degree of freedom; if all degrees of freedom have the same treatment, the hybrid approach is exactly a classical approach. For certain multifield problems the hybrid approach enables to define physic-friendly solvers. The hybrid approach can also be obtained from specific optimality considerations [24]. Mixed approaches [66, 80, 99] consist in searching a linear combination of interface displacement and effort field, depending on the artificial stiffness introduced on the interface one can recover the classical approaches (null stiffness for the dual approach, infinite stiffness for the primal approach). Moreover, the mixed approaches enable to provide the interface mechanical behavior and provide a more natural framework to handle complex interfaces (contact, friction...) than classical approaches.

In this paper we aim at reviewing most of non-overlapping domain decomposition methods. To adopt a rather general point of view we introduce a set of notations strongly linked to mechanical considerations as it gives the interface the main role of the methods. We try to include all methods inside the same pattern so that we can easily highlight the connections and differences between them. We adopt a practical point of view as we describe the mechanical concepts, the algebraic formulations, the algorithms and the practical implementation of the methods. At each step we try to emphasize keypoints and not to avoid theoretical and practical difficulties.

This paper is organized as follows. In section 2 we introduce the mechanical framework of our study, the common notations and the notion of interface assembly operators and mechanical operators which will play a central role in the methods. Section 3 provides a rather extensive review of the nonoverlapping domain decomposition methods in the framework of discretized problems: basic primal and dual approaches (with their variations), three-field method for conforming grids, mixed and hybrid approaches. A keypoint of the previous methods is the adjunction of optional constraints to form a "coarse problem" which

transmits global data through the whole structure, the strategies to introduce these optional constraints are studied in section 4, which leads to the definition of "recondensed" strategies FETIDP and BDDC. Section 5 deals with practical issues which are very often common to most of the methods. Assessments are given in section 6 to illustrate the methods and outline their main properties. Section 7 concludes the paper. As Krylov iterative solvers are often coupled to domain decomposition methods, main concepts and algorithms to use them are given in appendix A.

2 Formulation of an interface problem

To present as smoothly as possible non-overlapping domain decomposition methods we first consider a reference continuous mechanics problem, decompose the domain in two subdomains in order to introduce interface fields, then in order to describe correctly the interface we study a N-subdomain decomposition. Since our aim is not to prove theoretical performance results but make the reader feel some key-points of substructuring, we do not go too far in continuous formulation and quickly introduce discretized systems.
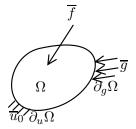
2.1 Reference problem

Figure 3. Reference problem

Let us consider domain Ω in $\mathbb{R}^n$ (n=1, 2 or 3) submitted to a classical linear elasticity problem (see figure 3) : displacement $\overline{u}_0$ is imposed on part $\partial_u\Omega$ of the boundary of the domain, effort $\overline{g}$ is imposed on complementary part $\partial_g\Omega$, volumic effort $\overline{f}$ is imposed on Ω, elasticity tensor is $\mathfrak{a}$ [53, 16]. The system is governed by the following equations :

$$\begin{cases} \overline{\mathrm{div}}(\overline{\overline{\sigma}}) + \overline{f} = \overline{0} & \text{in } \Omega \\ \overline{\overline{\sigma}} = \mathfrak{a} : \overline{\overline{\varepsilon}}(\overline{u}) & \text{in } \Omega \\ \overline{\overline{\varepsilon}}(\overline{u}) = \frac{1}{2}\left(\overline{\overline{\mathrm{grad}}}(\overline{u}) + \overline{\overline{\mathrm{grad}}}(\overline{u})^T\right) & \text{in } \Omega \\ \overline{\overline{\sigma}}.\overline{n} = \overline{g} & \text{on } \partial_g\Omega \\ \overline{u} = \overline{u}_0 & \text{on } \partial_u\Omega \end{cases} \quad (2)$$

In order to have the problem well posed, we suppose $\mathrm{mes}(\partial_u\Omega) > 0$. We also suppose that tensor $\mathfrak{a}$ defines a symmetric definite positive bilinear form on 2nd-order symmetric tensors. Under these assumptions, problem (2) has a unique solution [28].
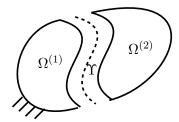
2.2 Two-subdomain decomposition

Figure 4. Two-subdomain decomposition

Let us consider a partition of domain Ω into 2 substructures $\Omega^{(1)}$ and $\Omega^{(2)}$. We define interface Υ between substructures (figure 4) :

$$\Upsilon = \partial\Omega^{(1)} \bigcap \partial\Omega^{(2)} \tag{3}$$

System (2) is posed on domain Ω, we write its restrictions to $\Omega^{(1)}$ and $\Omega^{(2)}$:

$$s = 1 \text{ or } 2, \quad \begin{cases} \overline{\mathrm{div}}(\overline{\overline{\sigma}}^{(s)}) + \overline{f}^{(s)} = \overline{0} & \text{in } \Omega^{(s)} \\ \overline{\overline{\sigma}}^{(s)} = \mathfrak{a}^{(s)} : \overline{\overline{\varepsilon}}(\overline{u}^{(s)}) & \text{in } \Omega^{(s)} \\ \overline{\overline{\varepsilon}}(\overline{u}^{(s)}) = \frac{1}{2}\left(\overline{\overline{\mathrm{grad}}}(\overline{u}^{(s)}) + \overline{\overline{\mathrm{grad}}}(\overline{u}^{(s)})^T\right) & \text{in } \Omega^{(s)} \\ \overline{\overline{\sigma}}^{(s)}.\overline{n}^{(s)} = \overline{g}^{(s)} & \text{on } \partial_g\Omega \bigcap \partial\Omega^{(s)} \\ \overline{u}^{(s)} = \overline{u}_0^{(s)} & \text{on } \partial_u\Omega \bigcap \partial\Omega^{(s)} \end{cases} \tag{4}$$

and the interface connection conditions, continuity of displacement

$$\overline{u}^{(1)} = \overline{u}^{(2)} \qquad \text{on } \Upsilon \tag{5}$$

and equilibrium of efforts (action-reaction principle)

$$\overline{\overline{\sigma}}^{(1)}\overline{n}^{(1)} + \overline{\overline{\sigma}}^{(2)}\overline{n}^{(2)} = \overline{0} \qquad \text{on } \Upsilon \tag{6}$$
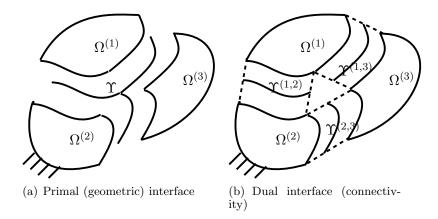
Of course system (4, 5, 6) is strictly equivalent to global problem (2).

2.3 N-subdomain decomposition

Let us consider a partition of domain Ω into N subdomains denoted $\Omega^{(s)}$. We can define the interface between two subdomains, the complete interface of one subdomain and the geometric interface at the complete structure scale:

$$\begin{cases} \Upsilon^{(i,j)} = \Upsilon^{(j,i)} = \partial\Omega^{(i)} \bigcap \partial\Omega^{(j)} \\ \Upsilon^{(s)} = \bigcup_j \Upsilon^{(s,j)} \\ \Upsilon = \bigcup_s \Upsilon^{(s)} \end{cases} \tag{7}$$

When implementing the method, one (possibly virtual) processor is commonly assigned to each subdomain, hence because we can tell "local" computations (realized independently on each processor) from "global" computations (realized by exchanging data between processors), we often refer to values as being global or local. Then $\Upsilon^{(s)}$ is the local interface

(a) Primal (geometric) interface   (b) Dual interface (connectivity)

Figure 5. Definition of the interface for a N-subdomain decomposition

and Υ the global interface. Because exchanges are most often one-to-one, $\underline{\Upsilon}^{(i,j)}$ is the $(i-j)$-communication interface.

Using more than two subdomains (except when using "band"-decomposition) leads to the appearance of "multiple-points" also called "crosspoints" (which are nodes shared by more than two subdomains). These crosspoints lead to the existence of two descriptions of the interface (figure 5): so-called geometric interface Υ and so-called connectivity interface made out of the set of one-to-one interfaces $(\underline{\Upsilon}^{(i,j)})_{1 \leqslant i < j \leqslant N}$. Each of the two most classical methods exclusively uses one of these descriptions so the geometric interface Υ is often referred to as the primal interface while the connectivity interface $\underline{\Upsilon}$ is referred to as the dual interface.

How crosspoints are handled is a fundamental key in the differentiation of domain decomposition methods. In the remainder of the paper we will always refer to data attached to the dual interface using underlined notation.

Remark 2.1 *Reader may have observed that the above presented connectivity description is redundant for crosspoints: let x be a crosspoint if x belongs to $\underline{\Upsilon}^{(1,2)}$ and $\underline{\Upsilon}^{(2,3)}$, it of course belongs to $\underline{\Upsilon}^{(1,3)}$. In the general case of a m-multiple crosspoint, there are $\binom{m}{2}$ connectivity relationships while only $(m-1)$ would be sufficient and necessary. We will present strategies to remove these redundancies in the algebraic analysis of the methods.*

Remark 2.2 *Cross-points may also introduce, at the continuous level, punctual-interfaces in 2d or edge-interfaces in 3d, which are interface with zero measure. Most often from a physical point of view these are not considered as interfaces. Anyhow after discretization all relationships are written node-to-node and the problem no longer exists.*

2.4 Discretization

We suppose that the reference problem has been discretized, leading to the resolution of $n \times n$ linear system:

$$Ku = f \tag{8}$$

Because of its key role in structural mechanics we will often refer to finite element discretization though any other technique would suit. The key points are the link between matrix K and the domain geometry and the sparse filling of matrix K (related to the fact that only narrow nodes have non-zero interaction).

We restrict to the case of element-oriented decompositions (each element belongs to one and only one substructure) which are conforming to the mesh which implies three conditions [89]:

- there is a one-to-one correspondence between degrees of freedom by the interface ;

- approximation spaces are the same by the interface ;

- models (beam, shell, 3d...) are the same by the interface.

Under these assumptions connection conditions simply write as node equalities. For non-conforming meshes, a classical solution is to use mortar-elements for which continuity and equilibrium are verified in a weak sense [1, 2, 101].

2.4.1 Boolean operators

In order to write communication relation between subdomains we have to introduce several operators.

The first one is the "local trace" operator $t^{(s)}$ which is the discrete projection from $\Omega^{(s)}$ to $\Upsilon^{(s)}$. It enables to cast data from a complete subdomain to its interface, and once transposed to extend data set on the interface to the whole subdomain (setting internal degrees of freedom to zero). In the remainder of the paper we will use subscript b for interface data and subscript i for internal data.

Then data lying on one subdomain interface has to be exchanged with its neighboring subdomains. It can be either realized on the primal interface or the dual interface, leading to two (global) "assembly" operators: the primal one $A^{(s)}$, and the dual one $\underline{A}^{(s)}$. The primal assembly operator is a strictly boolean operator while the dual assembly operator is a signed boolean operator (see figure 6): if a degree of freedom is set to 1 on one side of the interface, its corresponding degree of freedom on the other side of the interface is set to -1. Non-boolean assembly operators can be used in order to average connection conditions when using non-conforming domain decomposition methods [6].

Remark 2.3 *Our $(t^{(s)}, A^{(s)}, \underline{A}^{(s)})$ set of operators is not the most commonly used in papers related to domain decomposition. The interest of this choice is to be sufficient to explain most of the available strategies with only three operators. Other notations use "composed operators" like $(B^{(s)} = \underline{A}^{(s)} t^{(s)}$ or $L^{(s)^T} = A^{(s)} t^{(s)})$ which are not sufficient to describe all methods and which, in a way, omit the fundamental role played by the interface.*

Boolean operators have important classical properties. Please note the first one which expresses the orthogonality of the two assembly operators.
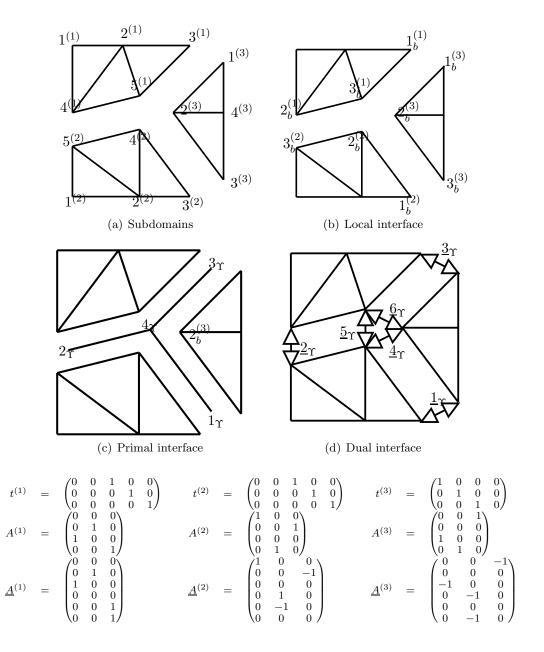
$$\sum_s \underline{A}^{(s)} A^{(s)^T} = 0 \tag{9a}$$

$$A^{(s)^T} A^{(s)} = I_{\Upsilon^{(s)}} \tag{9b}$$

$$\underline{A}^{(s)^T} \underline{A}^{(s)} = \text{diag}(\text{multiplicity} - 1)_{\Upsilon^{(s)}} \tag{9c}$$

$$A^{(s)} A^{(s)^T} = \left| \begin{array}{l} I \text{ on } \Upsilon^{(s)} \\ 0 \text{ elsewhere} \end{array} \right. \tag{9d}$$

Remark 2.4 *An interesting choice of description would have been to use redundant local interface (defining some kind of $\underline{t}^{(s)}$). This choice would stick to most classical implementations where the local interface of one subdomain is defined neighborwise. Dual assembly operator would write easily as a simple signing operator, but handling multiple points would be slightly more difficult for the primal assembly operator (see section 5.6).*

Remark 2.5 *Redundancies can easily be removed from the dual description of the interface. One has just to modify the connectivity table, so that one multiple point is connected only once to each subdomain. This can be carried out introducing two different assembly operators the "non-redundant" one and the "orthonormal" one (see figure 7) [48]. Only relationship 9c is modified (then $\underline{A}^{(s)^T}\underline{A}^{(s)} = I_{\Upsilon^{(s)}}$). The interest of the use of these assembly operators will be discussed in section 3.3.*

(a) Subdomains

(b) Local interface

(c) Primal interface

(d) Dual interface

$$
t^{(1)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad
t^{(2)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad
t^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}
$$

$$
A^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad
A^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad
A^{(3)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}
$$

$$
\underline{A}^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad
\underline{A}^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad
\underline{A}^{(3)} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}
$$

Figure 6. Local numberings, interface numberings, trace and assembly operators

(a) Local interface

(b) Redundant connectivity

(c) Non-redundant connectivity

(d) Orthonormal connectivity

$$\underline{A}^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad \underline{A}^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \underline{A}^{(3)} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

$$\underline{A}_N^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \underline{A}_N^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} \qquad \underline{A}_N^{(3)} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\underline{A}_O^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\sqrt{6}} \end{pmatrix} \qquad \underline{A}_O^{(2)} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{6}} & 0 \end{pmatrix} \qquad \underline{A}_O^{(3)} = \begin{pmatrix} 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{6}} & 0 \end{pmatrix}$$

Figure 7. Suppressing redundancies of dual interface

2.4.2 Basic equations

In order to rewrite equation (8) in a domain-decomposed context, we have to introduce the reaction unknown which is the discretization of $\overline{\overline{\sigma}}^{(1)}\overline{n}^{(1)} = -\overline{\overline{\sigma}}^{(2)}\overline{n}^{(2)}$ in equation (6). $\lambda^{(s)}$ is the reaction imposed by neighboring subdomains on subdomain (s). Commonly $\lambda^{(s)}$ is defined on the whole subdomain (s) while it is non-zero only on its interface, so

$$\lambda^{(s)} = t^{(s)T}\lambda_b^{(s)}.$$

$$\forall s, \ K^{(s)}u^{(s)} = f^{(s)} + \lambda^{(s)} \tag{10a}$$

$$\sum_s \underline{A}^{(s)}t^{(s)}u^{(s)} = 0 \tag{10b}$$

$$\sum_s A^{(s)}t^{(s)}\lambda^{(s)} = 0 \tag{10c}$$

Equation (10a) corresponds to the (local) equilibrium of each subdomain submitted to external conditions $f^{(s)}$ and reactions from neighbors $\lambda^{(s)}$. Equation (10b) corresponds to the (global) continuity of the displacement field through the interface. Equation (10c) corresponds to the (global) equilibrium of the interface (action-reaction principle).

This three-equation system (10) is the starting point from a rich zoology of methods, most of which possess strong connections we will try to emphasis on. Before going further in the exploration of these methods, we propose to introduce local condensed operators that represent a subdomain on its interface, then a set of synthetic notations.

2.4.3 Local condensed operators

Philosophically, local condensed operators are operators that represent how neighboring subdomains "see" one subdomain: a subdomain can be viewed as a black-box, the only information necessary for neighbors is how it behaves on its interface. Associated to this idea is the classical assumption that local operations are "exactly" performed. From an implementation point of view, when solving problems involving local matrices, a direct solver is employed. As we will see, the use of exact local solvers will be coupled with the use of iterative global solvers leading to a powerful combination of speed and precision of computations.

In this section we will always refer to the local equilibrium of subdomain (s) under interface loading:

$$K^{(s)}u^{(s)} = \lambda^{(s)} = t^{(s)T}\lambda_b^{(s)} \tag{11}$$

Primal Schur complement $S_p^{(s)}$**:**   If we renumber the local degrees of freedom of subdomain (s) in order to separate internal and boundary degrees of freedom, system (11) writes

$$\begin{pmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{pmatrix} \begin{pmatrix} u_i^{(s)} \\ u_b^{(s)} \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_b^{(s)} \end{pmatrix} \tag{12}$$

From the first line we draw

$$u_i^{(s)} = -K_{ii}^{(s)^{-1}}K_{ib}^{(s)}u_b^{(s)} \tag{13}$$

then the Gauss elimination of $u_i^{(s)}$ leads to

$$\left(K_{bb}^{(s)} - K_{bi}^{(s)}K_{ii}^{(s)^{-1}}K_{ib}^{(s)} \right) u_b^{(s)} = S_p^{(s)}u_b^{(s)} = \lambda_b^{(s)} \tag{14}$$

which is the condensed form of the local equilibrium of subdomains expressed in terms of interface fields. Operator $S_p^{(s)}$ is called local primal Schur complement. Its computation is realized by the inversion of matrix $K_{ii}^{(s)}$ which corresponds to Dirichlet conditions imposed on the interface of subdomain (s), so the primal Schur complement is always well defined, and commonly called the "local Dirichlet operator". Note that

the symmetry, positivity, and definition properties are inherited by matrix $S_p^{(s)}$ from matrix $K^{(s)}$.

An important result is that the kernel of matrices $K^{(s)}$ and $S_p^{(s)}$ can be deduced one from the other ($I_b^{(s)}$ is the identity matrix on the interface):

$$K^{(s)}R^{(s)} = 0 \implies S_p^{(s)}t^{(s)}R^{(s)} = S_p^{(s)}R_b^{(s)} = 0 \tag{15}$$

$$S_p^{(s)}R_b^{(s)} = 0 \implies K^{(s)} \begin{pmatrix} -K_{ii}^{(s)^{-1}}K_{ib}^{(s)} \\ I_b^{(s)} \end{pmatrix} R_b^{(s)} = K^{(s)}R^{(s)} = 0 \tag{16}$$

Primal Schur complement can also be interpreted as the discretization of the Stecklov-Poincaré operator. From a mechanical point of view, it is the linear operator that provides the reaction associated to given interface displacement field.

If we consider that the subdomain is also loaded on internal degrees of freedom, then the condensation of the equilibrium on the interface reads:

$$K^{(s)}u^{(s)} = f^{(s)} \Rightarrow S_p^{(s)}u_b^{(s)} = b_p^{(s)} \tag{17}$$

$$b_p^{(s)} = f_b^{(s)} - K_{bi}^{(s)}K_{ii}^{(s)^{-1}}f_i^{(s)} \tag{18}$$

$b_p^{(s)}$ is the condensed effort imposed on the substructure.

Dual Schur complement $S_d^{(s)}$**:** the dual Schur complement is a linear operator that computes interface displacement field from given interface effort. From equation (11) and (14) we have:

$$\left(t^{(s)}K^{(s)^+}t^{(s)^T}\right)\lambda_b^{(s)} = S_p^{(s)^+}\lambda_b^{(s)} = S_d^{(s)}\lambda_b^{(s)} = u_b^{(s)} \tag{19}$$

where $K^{(s)^+}$ is the generalized inverse of matrix $K^{(s)}$, and where it is assumed that no rigid body motion is excited. If we denote by $R^{(s)}$ the kernel of matrix $K^{(s)}$ this last condition reads:

$$R^{(s)^T}\lambda^{(s)} = 0 \quad \text{or equivalently} \quad R_b^{(s)^T}\lambda_b^{(s)} = 0 \tag{20}$$

Remark 2.6 *A generalized inverse (or pseudo-inverse) of matrix M is a matrix, denoted M^+, which verifies the following property: $\forall y \in \text{range}(M)$, $MM^+y = y$. Note that this definition leads to non-unique generalized inverse, however all results presented are independent of the choice of generalized inverse.*

Of course in order to take into account, inside (19), the possibility of the substructure to have zero energy modes, an arbitrary rigid displacement can be added leading to the next expression where vector $\alpha^{(s)}$ denotes the magnitude of rigid body motions:

$$u_b^{(s)} = S_d^{(s)}\lambda_b^{(s)} + R_b^{(s)}\alpha^{(s)} \tag{21}$$

Hybrid Schur complement: $S_{pd}^{(s)}$ this operator corresponds to an interface where degrees of freedom are partitioned into two subsets. Suppose that the first subset is submitted to given Dirichlet conditions and the second to Neumann condition, $S_{pd}^{(s)}$ is

the linear operator that associates resulting reaction on the first subset and resulting displacement on the second subset to those given conditions. We denote by subscript p data defined on the first subset and by subscript d data defined on the second subset (schematically $b = p \cup d$ and $p \cap d = \emptyset$).

$$S_{pd}^{(s)} \begin{pmatrix} u_p^{(s)} \\ \lambda_d^{(s)} \end{pmatrix} = \begin{pmatrix} \lambda_p^{(s)} \\ u_d^{(s)} \end{pmatrix} \tag{22}$$

The computation of operator $S_{pd}^{(s)}$ though no more complex than the computation of $S_p^{(s)}$ or $S_d^{(s)}$, requires more notations. A synthetic option is to denote by subscript $\bar{p}$ the sets of internal (subscript i) data and second-interface-subset (subscript d) data, schematically $\bar{p} = i \cup d$. We introduce a modified trace operator:

$$t_d^{(s)} v_{\bar{p}} = t_d^{(s)} \begin{pmatrix} v_i \\ v_d \end{pmatrix} = v_d \tag{23}$$

Then internal equilibrium (11) reads:

$$\begin{pmatrix} K_{\bar{p}\bar{p}}^{(s)} & K_{\bar{p}p}^{(s)} \\ K_{p\bar{p}}^{(s)} & K_{pp}^{(s)} \end{pmatrix} \begin{pmatrix} u_{\bar{p}}^{(s)} \\ u_p^{(s)} \end{pmatrix} = \begin{pmatrix} t_d^{(s)T} \lambda_d^{(s)} \\ \lambda_p^{(s)} \end{pmatrix} \tag{24}$$

Then hybrid Schur complement is:

$$S_{pd}^{(s)} = \begin{pmatrix} K_{pp}^{(s)} - K_{p\bar{p}}^{(s)} K_{\bar{p}\bar{p}}^{(s)+} K_{\bar{p}p}^{(s)} & K_{p\bar{p}}^{(s)} K_{\bar{p}\bar{p}}^{(s)+} t_d^{(s)T} \\ -t_d^{(s)} K_{\bar{p}\bar{p}}^{(s)+} K_{\bar{p}p}^{(s)} & t_d^{(s)} K_{\bar{p}\bar{p}}^{(s)+} t_d^{(s)T} \end{pmatrix} \tag{25}$$

As can be noticed the diagonal blocks of $S_{pd}^{(s)}$ look like fully primal and fully dual Schur complements, while extradiagonal blocks are antisymmetric (assuming $K^{(s)}$ is symmetric). Of course if all interface degrees of freedom belong to the same subset, the hybrid Schur complement equals "classical" fully primal or fully dual Schur complement. Moreover it stands out clearly that:

$$S_{pd}^{(s)+} = S_{dp}^{(s)} = \begin{pmatrix} t_p^{(s)} K_{\bar{d}\bar{d}}^{(s)+} t_p^{(s)T} & -t_p^{(s)} K_{\bar{d}\bar{d}}^{(s)+} K_{\bar{d}d} \\ K_{d\bar{d}}^{(s)} K_{\bar{d}\bar{d}}^{(s)+} t_p^{(s)T} & K_{dd}^{(s)} - K_{d\bar{d}}^{(s)} K_{\bar{d}\bar{d}}^{(s)+} K_{\bar{d}d}^{(s)} \end{pmatrix} \tag{26}$$

$S_{dp}^{(s)}$ is the operator which associates displacement on the first subset and reaction on the second subset to given effort on the first subset and given displacement on the second subset.

As both matrices $K_{\bar{p}\bar{p}}^{(s)}$ and $K_{\bar{d}\bar{d}}^{(s)}$ may not be invertible, only their pseudo-inverse has been introduced. The invertibility is strongly dependent on the choice of interface subsets.

2.4.4 Block notations

While condensed operators simplify the writing of local operations, the block notations make it easier to understand the global operations of domain decomposition. We propose to denote by superscript $\square^{\diamond}$ the row-block repetition of local vectors and the diagonal-block

repetition of matrices, block assembly operators are written in one row (column-block) and denoted by special font, for instance:

$$u^\diamond = \begin{pmatrix} u^{(1)} \\ \vdots \\ u^{(N)} \end{pmatrix} \qquad f^\diamond = \begin{pmatrix} f^{(1)} \\ \vdots \\ f^{(N)} \end{pmatrix} \qquad \lambda^\diamond = \begin{pmatrix} \lambda^{(1)} \\ \vdots \\ \lambda^{(N)} \end{pmatrix}$$

$$K^\diamond = \begin{pmatrix} K^{(1)} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & K^{(N)} \end{pmatrix} \qquad t^\diamond = \begin{pmatrix} t^{(1)} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & t^{(N)} \end{pmatrix}$$

$$\mathbb{A} = \begin{pmatrix} A^{(1)} & \dots & A^{(N)} \end{pmatrix} \qquad \underline{\mathbb{A}} = \begin{pmatrix} \underline{A}^{(1)} & \dots & \underline{A}^{(N)} \end{pmatrix}$$

Remark 2.7 *The specific notation for assembly operators aims at emphasizing at their specific role in terms of parallelism for the methods. Moreover, the only operation that requires exchange of data between subdomains is the use of non-transposed assembly operators.*

Fundamental system (10) then reads:

$$K^\diamond u^\diamond = f^\diamond + \lambda^\diamond \tag{27a}$$

$$\mathbb{A} t^\diamond \lambda^\diamond = 0 \tag{27b}$$

$$\underline{\mathbb{A}} t^\diamond u^\diamond = \underline{0} \tag{27c}$$

or in condensed form:

$$S_p^\diamond u_b^\diamond = b_p^\diamond + \lambda_b^\diamond \tag{28a}$$

$$\mathbb{A} \lambda_b^\diamond = 0 \tag{28b}$$

$$\underline{\mathbb{A}} u_b^\diamond = \underline{0} \tag{28c}$$

The orthogonal property of assembly operators (9a) simply reads:

$$\underline{\mathbb{A}} \mathbb{A}^T = 0 \tag{29}$$

Relation (9d) reads:

$$\mathbb{A} \mathbb{A}^T = \mathrm{diag}(\mathrm{multiplicity}) \tag{30}$$

Remark 2.8 *For improved readability, we will denote by bold font objects defined in a unique way on the interface (ie "assembled" quantities). Schematically, assembly operators enable to go from block notations to bold notations and transposed assembly operators realize the reciprocal operations.*

2.4.5 Brief review of classical strategies

We can define general strategies to solve system (27) or (28):

Primal approaches [104, 105, 74, 102, 75, 106, 103] a unique interface displacement unknown $\boldsymbol{u}_b$ satisfying equation (28c) is introduced, then an iterative process enables to satisfy (28b) while always verifying (28a).

Dual approaches [41, 31, 42, 43, 39, 76, 7] a unique interface effort unknown $\boldsymbol{\lambda}_b$ satisfying equation (28b) is introduced, then an iterative process enables to satisfy (28c) while always verifying (28a).

Three fields approaches [11, 81, 92] a unique interface displacement $\boldsymbol{u}_b$ is introduced, then relation (28c) is dualized so that interface efforts $\lambda_b^\diamond$ are introduced as Lagrange multipliers which yet have to verify relation (28b). Then the iterative process looks simultaneously for $(\lambda_b^\diamond, \boldsymbol{u}_b, u_b^\diamond)$ verifying exactly equation (28a). As this method is mostly designed for non-matching discretizations it will not be exposed in the remaining of this paper, anyhow a variant of the dual method which is equivalent to the three-field method with conforming grids will be described.

Mixed approaches [54, 66, 100, 99] new interface unknown is introduced which is a linear combination of interface displacement and effort, $\mu_b^\diamond = \lambda_b^\diamond + T_b^\diamond u_b^\diamond$, then the interface system is rewritten in terms of unknown $\mu_b^\diamond$, this new system is solved iteratively and then $\lambda_b^\diamond$ and $u_b^\diamond$ are postprocessed. Of course matrix $T_b^\diamond$ is an important parameter of these methods.

Hybrid approaches [64, 35, 77, 34, 57] interface is split into parts where primal, dual or mixed approaches are applied, specific recondensation methods may then be applied.

Many different methods can be deduced from these large strategies, the most common will be presented and discussed in section 3. Anyhow since iterative solvers are used to solve interface problems, we recommend the reader to refer to appendix A where most used solvers are presented, including important details about constrained resolutions.

3 Classical solution strategies to the interface problem

The aim of this section is to give extended review of classical domain decomposition methods, the principle of which has just been exposed. The association with Krylov iterative solvers is an important point of these methods, appendix A provides a summary of important results and algorithms that are used in this section.

3.1 Primal domain decomposition method

The principle of primal domain decomposition method is to write the interface problem in terms of one unique unknown interface displacement field $\boldsymbol{u}_b$. The trace of local displacement fields then writes $u_b^\diamond = \mathbb{A}^T \boldsymbol{u}_b$. Because of the orthogonality between assembly operators (29), equation (28c) is automatically verified. Using equation (28b) to eliminate unknown reaction $\lambda_b^\diamond$ inside (28a), we get the primal formulation of the interface problem:

$$\boldsymbol{S}_p \boldsymbol{u}_b = \left(\mathbb{A} S_p^\diamond \mathbb{A}^T \right) \boldsymbol{u}_b = \mathbb{A} b_p^\diamond = \boldsymbol{b}_p \tag{31}$$

Operator $\boldsymbol{S}_p$ is the global primal Schur complement of the decomposed structure, it results as the sum of local contributions (with non-block notations $\boldsymbol{S}_p = \sum_s A^{(s)} S_p^{(s)} A^{(s)^T}$). Using a direct solver to solve system (31) implies the exact computation of local contribution, the sum of these contributions (in a parallel computing context, this step correspond

to large data exchange between processors) and the inversion of the global primal Schur complement which size is the global geometric interface (the size of which is far from being neglectable) and which sparsity is very poor (each interface degree of freedom is connected to degrees of freedom belonging to the same subdomains). Using an iterative solver is much less expensive since the only required operations are matrix-vector products which can be realized in parallel because of the assembled structure of global primal Schur complement; moreover excellent and rather cheap preconditioner exists. Note that if global matrix K is symmetric positive definite then so is operator $\boldsymbol{S}_p$ and then popular conjugate gradient algorithm can be used to solve the primal interface problem, in other cases solvers like GMRes or orthodir have to be employed.

3.1.1 Preconditioner to the primal interface problem

An efficient preconditioner $\tilde{\boldsymbol{S}}_p^{-1}$ is an interface operator giving a good approximation of the inverse of $\boldsymbol{S}_p$. Various strategies are possible. For instance, a direct preconditioning method is based on the construction of an approximate Schur complement from a simplified structure defined by degrees of freedom "near" the interface. Anyhow such a method does not respect the repartition of the data through processors. A good parallel preconditioner has to minimize data exchange.

Since operator $\boldsymbol{S}_p$ is the sum of local contributions, the most classical strategy is then to define $\tilde{\boldsymbol{S}}_p^{-1}$ as a scaled sum of the inverse of local contributions:

$$\tilde{\boldsymbol{S}}_p^{-1} = \tilde{\mathbb{A}} S_p^{\diamond+} \tilde{\mathbb{A}}^T = \tilde{\mathbb{A}} S_d^{\diamond} \tilde{\mathbb{A}}^T \tag{32}$$

Since $S_p^{\diamond+} = S_d^{\diamond}$ requires the computation of local problems with given effort on the interface, this preconditioner is called the Neumann preconditioner. Scaled assembly operator $\tilde{\mathbb{A}}$ can be defined the following way [65]:

$$\tilde{\mathbb{A}} = \left(\mathbb{A} M^{\diamond} \mathbb{A}^T\right)^{-1} \mathbb{A} M^{\diamond} \tag{33}$$

where $M^{\diamond}$ is a parameter which enables to take into account the heterogeneity of the subdomains connected by the interface. It should make matrix $\left(\mathbb{A} M^{\diamond} \mathbb{A}^T\right)$ easily invertible and give a representation of the stiffness of the interface, most commonly:

- $M^{\diamond} = I^{\diamond}$ for homogeneous structures,

- $M^{\diamond} = \text{diag}(K_{bb}^{\diamond})$ for compressible heterogeneous structures,

- $M^{\diamond} = \mu^{\diamond}$ for incompressible heterogeneous structures ($\mu^{\diamond}$ is the diagonal matrix the coefficients of which are the shearing modulus of interface degrees of freedom).

The (s) notation makes it easier to understand implementation of scaled assembly operators:

$$\tilde{\boldsymbol{S}}_p^{-1} = \sum_s M^{(s)} A^{(s)} S_d^{(s)} A^{(s)^T} M^{(s)} \tag{34}$$

- $M^{(s)} = \text{diag}(\frac{1}{\text{multiplicity}})$ for homogeneous structures,

- $M^{(s)} = \text{diag}(\frac{\text{diag}(K_{bb}^{(s)})_i}{\sum_j \text{diag}(K_{bb}^{(j)})_i})$ for compressible heterogeneous structures (assuming i represents the same degree of freedom shared by the j subdomains)

- $M^{(s)} = \text{diag}(\frac{\mu_i^{(s)}}{\sum_j \mu_i^{(j)}})$ for incompressible heterogeneous structures (assuming i represents the same degree of freedom shared by the j subdomains)

The following partition of unity result clearly holds:

$$\tilde{\mathbb{A}}\mathbb{A}^T = I_\Upsilon \tag{35}$$

$$\sum_s M^{(s)} = I_\Upsilon \tag{36}$$

3.1.2 Coarse problem

The use of dual Schur complement is associated to an optimality condition, as said earlier vector being multiplied by the pseudo inverse should lie inside the image of $S_p^\diamond$. Since preconditioning is applied to residual r, the optimality condition reads:

$$R_b^{\diamond T} \tilde{\mathbb{A}}^T r = 0 \tag{37}$$

and introducing classical notation $\boldsymbol{G} = \tilde{\mathbb{A}}R_b^\diamond$, $\boldsymbol{G}^T r = 0$. Such a condition can then be interpreted as an augmented-Krylov algorithm (see section A.7). Once equipped with that augmentation problem, the primal Schur complement method is referred to as the "balanced domain decomposition" (BDD [74, 102]). Algorithm 3.1 summarizes the classical BDD approach, and figure 8 provides a schematic representation of the first iteration of the preconditioned primal approach.

Algorithm 3.1 Primal Schur complement with conjugate gradient

---
1: Set $P = I - G(G^T S_p G)^{-1} G^T S_p$
2: Compute $u_0 = G(G^T S_p G)^{-1} G^T b_p$
3: Compute $r_0 = b_p - S_p u_0 = P^T b_p$
4: $z_0 = \tilde{S}_p^{-1} r_0$ set $w_0 = z_0$
5: **for** $j = 0, \ldots, m$ **do**
6:     $p_j = S_p P w_j$ (notice $S_p P = P^T S_p = P^T S_p P$)
7:     $\alpha_j = (z_j, r_j)/(p_j, w_j)$
8:     $u_{j+1} = u_j + \alpha_j w_j$
9:     $r_{j+1} = r_j - \alpha_j p_j$
10:     $z_{j+1} = \tilde{S}_p^{-1} r_{j+1}$
11:     For $0 \leqslant i \leqslant j$, $\beta_j^i = -(z_{j+1}, p_i)/(w_i, p_i)$
12:     $w_{j+1} = z_{j+1} + \sum_{i=1}^j \beta_j^i w_i$
13: **end for**
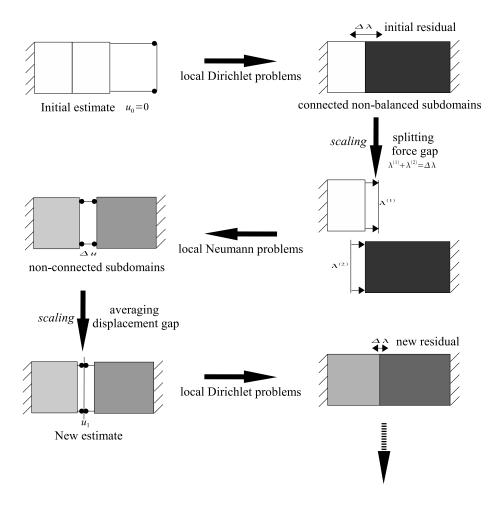
3.1.3 Error estimate

The reference error estimate is the one linked to the convergence over the complete structure: $\frac{\|Ku-f\|}{\|f\|}$. Assuming local inversions are exact, we reach the following result:

$$\frac{\|Ku - f\|}{\|f\|} = \frac{\|\boldsymbol{S}_p \boldsymbol{u}_b - \boldsymbol{b}_p\|}{\|f\|} \tag{38}$$

During the iterative process $\|\boldsymbol{S}_p \boldsymbol{u}_b - \boldsymbol{b}_p\|$ is the norm of residual r as computed line 9 of algorithm 3.1, so the global convergence can be controlled by the convergence of the interface iterative process.

Figure 8. Representation of first iteration of preconditioned primal approach

3.1.4 P-FETI method

The P-FETI method is a variation of BDD proposed by [49, 50] inspired by the dual approach (the reader should refer to the dual method before going further inside P-FETI). Its principle is to provide another assembly operator which incorporate rigid body elimination by a dual-like projector.

$$\tilde{\boldsymbol{S}}_p^{-1} \quad = \quad \mathbb{H}S_d^\diamond\mathbb{H}^T \tag{39}$$

$$\mathbb{H}^T \quad = \quad \tilde{\mathbb{A}}^T - \underline{\mathbb{A}}^T\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\left(\underline{\boldsymbol{G}}^T\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\right)^{-1}\boldsymbol{G}^T \tag{40}$$

The choice of matrix $\underline{\boldsymbol{Q}}$ is guided by the same considerations as in the dual method. It is worth noting that when $\underline{\boldsymbol{Q}}$ is chosen equal to the Dirichlet preconditioner of the dual method ($\underline{\boldsymbol{Q}} = \tilde{\underline{\mathbb{A}}}^T S_p^\diamond\tilde{\underline{\mathbb{A}}}$) then the P-FETI method is equivalent to the classical balanced domain decomposition.

3.2 Dual domain decomposition method

The principle of dual domain decomposition method is to write the interface problem in terms of one unique unknown interface effort field $\underline{\boldsymbol{\lambda}}_b$. The trace of local effort fields then

writes $\lambda_b^\diamond = \underline{\mathbb{A}}^T \boldsymbol{\lambda}_b$. Because of the orthogonality between assembly operators (29), equation (28b) is automatically verified. In order to eliminate unknown interface displacement field using (28c), we first obtain it from equation (28a) (or equivalently (27a)): as seen in (21) the inversion of local systems may require the use of generalized inverse and the introduction of rigid body motions the magnitude of which is denoted by vector $\alpha^{(s)}$, the use of generalized inverse is then submitted to compatibility condition.

$$u_b^\diamond \;=\; S_d^\diamond(b_p^\diamond + \underline{\mathbb{A}}^T \boldsymbol{\lambda}_b) + R_b^\diamond \alpha^\diamond \tag{41}$$

$$R_b^{\diamond T}(b_p^\diamond + \underline{\mathbb{A}}^T \boldsymbol{\lambda}_b) \;=\; 0 \tag{42}$$

The first line is then premultiplied by $\underline{\mathbb{A}}$ (same expressions could be obtain from non condensed notations).

$$\underline{\boldsymbol{S}}_d = \underline{\mathbb{A}} S_d^\diamond \underline{\mathbb{A}}^T$$

$$b_d^\diamond = S_d^\diamond b_p^\diamond = t^\diamond K^{\diamond +} f^\diamond$$

$$\underline{\boldsymbol{G}} = \underline{\mathbb{A}} R_b^\diamond$$

$$e^\diamond = R_b^{\diamond T} b_p^\diamond = R^{\diamond T} f^\diamond$$

we get the dual formulation of the interface problem:

$$\begin{pmatrix} \underline{\boldsymbol{S}}_d & \underline{\boldsymbol{G}} \\ \underline{\boldsymbol{G}}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_b \\ \alpha^\diamond \end{pmatrix} = \begin{pmatrix} -\boldsymbol{b}_d \\ -e^\diamond \end{pmatrix} \tag{43}$$

This is the basic dual Schur complement method, also called Finite Element Tearing and Interconnecting method (FETI [41, 43]). For similar reasons to the primal Schur complement method, this system is most often solved using an iterative solver, then we will soon discuss the preconditioning issue and how the $\underline{\boldsymbol{G}}^T \boldsymbol{\lambda}_b + e^\diamond = 0$ constraint is handled. Let us first remark that global dual Schur complement $\underline{\boldsymbol{S}}_d$ is non-definite as soon as redundancies appear in the connectivity description of the interface, anyhow it is easy to prove [41] that local contributions $\lambda_b^\diamond = \underline{\mathbb{A}}^T \boldsymbol{\lambda}_b$ are unique (non-definition only affect the "artificial" splitting of forces on multiple points), and that because the right hand side lies in range($\underline{\mathbb{A}}$) iterative process converges; other considerations on the splitting of physical efforts between subdomains will lead to improved initialization (see section 3.2.4 and [60]).

3.2.1 Preconditioner to the dual interface problem

Like it is done in the primal approach, the most interesting preconditioners are researched as assembly of local contributions, and the global dual Schur complement being a sum of local contributions, optimal preconditioner is a scaled sum of local inverses.

$$\tilde{\underline{\boldsymbol{S}}}_d^{-1} = \tilde{\underline{\mathbb{A}}} S_d^{\diamond +} \tilde{\underline{\mathbb{A}}}^T = \tilde{\underline{\mathbb{A}}} S_p^\diamond \tilde{\underline{\mathbb{A}}}^T \tag{44}$$

Because this preconditioner uses local primal Schur complement, which corresponds to the local resolution of imposed displacement problems, it is commonly called the Dirichlet preconditioner. One interesting point is the possibility to give approximation of the local Schur complement operator leading to the following preconditioners:

$$S_p^\diamond \approx K_{bb}^\diamond \qquad \text{lumped preconditioner} \tag{45}$$

$$S_p^\diamond \approx \mathrm{diag}(K_{bb}^\diamond) \qquad \text{superlumped preconditioner} \tag{46}$$

These preconditioners have very low computational cost (they do not require the computation and storage of the inverse of local internal matrices $K_{ii}^{\diamond -1}$), even if their numerical

efficiency is not as strong as the Dirichlet preconditioner, they can lead to very reduced computational time.

Scaled assembly operator $\tilde{\underline{\mathbb{A}}}$ can be defined the following way [65]:

$$\tilde{\underline{\mathbb{A}}} = \left(\underline{\mathbb{A}}M^{\diamond-1}\underline{\mathbb{A}}^T\right)^+ \underline{\mathbb{A}}M^{\diamond-1} \tag{47}$$

where $M^\diamond$ is the same parameter as for the primal approach. Such a definition is not that easy to implement, an almost equivalent strategy is then used, easily described using the (s) notation:

$$\tilde{\boldsymbol{S}}_d^{-1} = \sum_s \underline{M}^{(s)}\underline{A}^{(s)}S_p^{(s)}\underline{A}^{(s)T}\underline{M}^{(s)} \tag{48}$$

- $\underline{M}^{(s)} = \text{diag}(\frac{1}{\text{multiplicity}})$ for homogeneous structures,

- $\underline{M}^{(s)} = \text{diag}(\frac{\text{diag}(K_{bb}^{(r)})_i}{\sum_j \text{diag}(K_{bb}^{(j)})_i})$ for compressible heterogeneous structures (assuming i represents the same degree of freedom shared by the j subdomains and (r) is the subdomain connected to (s) on degree of freedom i),

- $\underline{M}^{(s)} = \text{diag}(\frac{\mu_i^{(r)}}{\sum_j \mu_i^{(j)}})$ for incompressible heterogeneous structures (assuming i represents the same degree of freedom shared by the j subdomains and (r) is the subdomain connected to (s) on degree of freedom i).

We have the following partition of unity result:

$$\tilde{\underline{\mathbb{A}}}\underline{\mathbb{A}}^T = I_{\underline{\Upsilon}} \tag{49}$$

and the following complementarity between primal and dual scalings [60]:

$$\tilde{\mathbb{A}}^T\mathbb{A} + \underline{\mathbb{A}}^T\tilde{\underline{\mathbb{A}}} = I^\diamond \tag{50}$$

$$A^{(s)T}M^{(s)}A^{(s)} + \underline{A}^{(s)T}\underline{M}^{(s)}\underline{A}^{(s)} = I_{\Upsilon^{(s)}} \tag{51}$$

3.2.2 Coarse problem

Admissibility condition $\underline{\boldsymbol{G}}^T\underline{\boldsymbol{\lambda}}_b + e^\diamond = 0$, can be handled with an initialization / projection algorithm (see section A.6): $\underline{\boldsymbol{\lambda}}_b = \underline{\boldsymbol{\lambda}}_0 + \underline{\boldsymbol{P}}\boldsymbol{\lambda}^*$ with $\underline{\boldsymbol{G}}^T\underline{\boldsymbol{\lambda}}_0 = -e^\diamond$ and $\underline{\boldsymbol{G}}^T\underline{\boldsymbol{P}} = 0$.

$$\underline{\boldsymbol{\lambda}}_0 = -\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\left(\underline{\boldsymbol{G}}^T\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\right)^{-1}e^\diamond \tag{52}$$

$$\underline{\boldsymbol{P}} = \underline{\boldsymbol{I}} - \underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\left(\underline{\boldsymbol{G}}^T\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\right)^{-1}\underline{\boldsymbol{G}}^T \tag{53}$$

The easiest choice for operator $\underline{\boldsymbol{Q}}$ is the identity matrix, projector $\underline{\boldsymbol{P}}$ is then orthogonal, this choice is well suited to homogeneous structures. For heterogeneous structures, matrix $\underline{\boldsymbol{Q}}$ has to provide information on the stiffness of subdomains, then $\underline{\boldsymbol{Q}}$ is chosen to be a version of the preconditioner leading to "superlumped projector" ($\underline{\boldsymbol{Q}} = \tilde{\underline{\mathbb{A}}}\,\text{diag}(K_{bb}^\diamond)\tilde{\underline{\mathbb{A}}}^T$), "lumped projector" ($\underline{\boldsymbol{Q}} = \tilde{\underline{\mathbb{A}}}K_{bb}^\diamond\tilde{\underline{\mathbb{A}}}^T$) and "Dirichlet projector" ($\underline{\boldsymbol{Q}} = \tilde{\underline{\mathbb{A}}}S_p^\diamond\tilde{\underline{\mathbb{A}}}^T$). Superlumped projector is often a good compromise between numerical efficiency and computational cost.

Algorithm 3.2 presents a classical implementation of FETI method, and figure 9 provides a schematic representation of the first iteration of the preconditioned dual approach.

Algorithm 3.2 Dual Schur complement with conjugate gradient

1: Set $P = I - QG(G^T QG)^{-1} G^T$
2: Compute $\lambda_0 = -QG(G^T QG)^{-1} e$
3: Compute $r_0 = P^T b_d - S_d \lambda_0)$
4: $z_0 = P \tilde{S}_d^{-1} r_0$ set $w_0 = z_0$
5: **for** $j = 0, \ldots, m$ **do**
6:     $p_j = P^T S_d w_j$
7:     $\alpha_j = (z_j, r_j)/(p_j, w_j)$
8:     $\lambda_{j+1} = \lambda_j + \alpha_j w_j$
9:     $r_{j+1} = r_j - \alpha_j p_j$
10:     $z_{j+1} = P \tilde{S}_d^{-1} r_{j+1}$
11:     For $0 \leqslant i \leqslant j$, $\beta_j^i = -(z_{j+1}, p_i)/(w_i, p_i)$
12:     $w_{j+1} = z_{j+1} + \sum_{i=1}^{j} \beta_j^i w_i$
13: **end for**
14: $\alpha^\diamond = (G^T QG)^{-1} G^t r_m$
15: $u^\diamond = K^{\diamond +} \lambda_m + R^\diamond \alpha^\diamond$

3.2.3 Error estimate

The convergence of the dual domain decomposition method is strongly linked to physical considerations. After projection, the residual can be interpreted as the jump of displacement between substructures:

$$\underline{r} = \underline{P}^T(-\underline{b}_d - \underline{S}_d \underline{\lambda}) = \mathbb{A} u^\diamond = \underline{\Delta}(u) \tag{54}$$

$$\underline{\Delta}(u)_{|\Upsilon^{(i,j)}} = u_{|\Upsilon^{(i,j)}}^{(i)} - u_{|\Upsilon^{(i,j)}}^{(j)} \tag{55}$$

Anyhow, such an interpretation cannot be connected to the global convergence of the system. In order to evaluate the global convergence, a unique interface displacement field has to be defined (most often using a scaled average of local displacement fields) and used to evaluate the global residual. When using the Dirichlet preconditioner, it is possible to cheaply evaluate that convergence criterion. Average interface displacement u_b can be defined as follow:
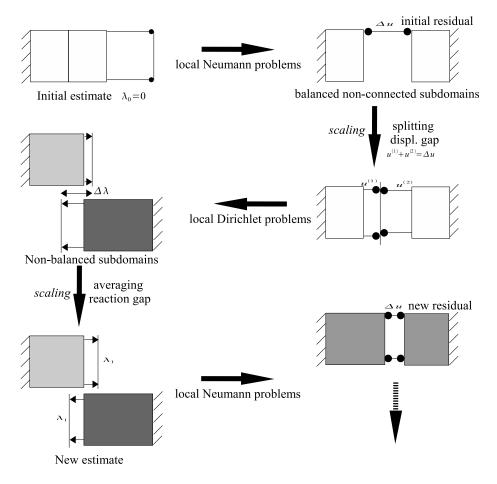
$$u_b = \mathbb{A} \tilde{\underline{\mathbb{A}}}^T \underline{\Delta} \tag{56}$$

then, from equation (38), convergence criterion reads: $\|Ku - f\| = \|\mathbb{A} S_p^\diamond \tilde{\underline{\mathbb{A}}}^T \underline{r}\|$. So when using the Dirichlet preconditioner, the evaluation of the global residual only requires the use of a geometric assembly after the local Dirichlet resolution.

3.2.4 Interpretation and improvement on the initialization

Let us come back to the original dual system (27).

$$\begin{cases} K^\diamond u^\diamond = f^\diamond + t^{\diamond T} \underline{\mathbb{A}}^T \underline{\lambda}_b \\ \underline{\mathbb{A}} t^\diamond u^\diamond = \underline{0} \end{cases} \tag{57}$$

Figure 9. Representation of first iteration of preconditioned dual approach

And suppose this system is being initialized with non zero effort $\underline{\boldsymbol{\lambda}}_{b0}$:

$$K^\diamond u^\diamond = f^\diamond + t^{\diamond T} \mathbb{A}^T \underline{\boldsymbol{\lambda}}_b \tag{58}$$

$$\text{let } \underline{\boldsymbol{\lambda}}_b = \widetilde{\underline{\boldsymbol{\lambda}}}_b + \underline{\boldsymbol{\lambda}}_{b0}$$

$$K^\diamond u^\diamond = f^\diamond + t^{\diamond T} \mathbb{A}^T \widetilde{\underline{\boldsymbol{\lambda}}}_b + t^{\diamond T} \mathbb{A}^T \underline{\boldsymbol{\lambda}}_{b0}$$

$$= \widetilde{f}^\diamond + t^{\diamond T} \mathbb{A}^T \widetilde{\underline{\boldsymbol{\lambda}}}_b \tag{59}$$

$$\text{with } \widetilde{f}^\diamond = f^\diamond + t^{\diamond T} \mathbb{A}^T \underline{\boldsymbol{\lambda}}_{b0} \tag{60}$$

So initialization $\underline{\boldsymbol{\lambda}}_{b0}$ can be interpreted as modification $t^{\diamond T} \mathbb{A}^T \underline{\boldsymbol{\lambda}}_{b0}$ of the intereffort between substructures: local problems are defined except for an equilibrated interface effort field; the only field that makes mechanical sense (and that is uniquely defined) is the assembly of interface efforts.

$$\mathbb{A} t^\diamond \widetilde{f}^\diamond = \mathbb{A} t^\diamond f^\diamond = \boldsymbol{f}_b \quad \text{global interface effort} \tag{61}$$

$$\text{because } \mathbb{A} t^\diamond t^{\diamond T} \mathbb{A}^T \underline{\boldsymbol{\lambda}}_{b0} = \mathbb{A} \underline{\mathbb{A}}^T \underline{\boldsymbol{\lambda}}_{b0} = 0 \tag{62}$$

Non-zero initialization then can be interpreted as a repartition of global interface effort $\boldsymbol{f}_b$. Two strategies can be defined in order to realize that splitting.

Classical effort splitting Though splitting is hardly ever interpreted as a specific initialization, it is commonly realized, based on the difference of stiffness between neighboring substructures (that idea is strongly connected to the definition of scaled assembly operators): the aim is to guide the stress flow inside the stiffer substructure, sticking to what mechanically occurs.

Global interface effort $\boldsymbol{f}_b$ is then split according to the stiffness scaling ($M^\diamond = \mathrm{diag}(K_{bb}^\diamond)$), which leads to modified local effort $\widetilde{f}_b^\diamond$.

$$\widetilde{f}_b^\diamond = M^\diamond \mathbb{A} \left(\mathbb{A} M^\diamond \mathbb{A}^T \right)^{-1} \boldsymbol{f}_b \tag{63}$$

Complete effort $\widetilde{f}^\diamond$ is constituted by $f^\diamond$ inside the substructure $((I - t^{\diamond T} t^\diamond) f^\diamond)$ and split effort on its interface $(t^{\diamond T} \widetilde{f}_b^\diamond)$.

$$\widetilde{f}^\diamond = (I - t^{\diamond T} t^\diamond) f^\diamond + t^{\diamond T} \widetilde{f}_b^\diamond \tag{64}$$

Because of the complementarity between scaled assembly operators (50), final effort reads

$$\widetilde{f}^\diamond = f^\diamond - t^{\diamond T} \mathbb{A}^T \left(\mathbb{A} M^{\diamond -1} \mathbb{A}^T \right)^+ \mathbb{A} M^{\diamond -1} t^\diamond f^\diamond \tag{65}$$

Interface effort splitting [60] If we start from condensed dual system (28)

$$\begin{cases} S_p^\diamond u_b^\diamond = b_p^\diamond + \mathbb{A}^T \boldsymbol{\lambda}_b \\ \mathbb{A} u_b^\diamond = 0 \end{cases} \tag{66}$$

condensed efforts can be split along the interface as long as global condensed effort remains unique. Assembled condensed interface effort reads $\boldsymbol{b}_p = \mathbb{A} b_p^\diamond$, if it is split according to the stiffness of the substructures:

$$\widetilde{b}_p^\diamond = M^\diamond \mathbb{A} \left(\mathbb{A} M^\diamond \mathbb{A}^T \right)^{-1} \boldsymbol{b}_p \tag{67}$$

We have, using the complementarity between scalings:

$$\widetilde{b}_p^\diamond = b_p^\diamond - \mathbb{A}^T \left(\mathbb{A} M^{\diamond -1} \mathbb{A}^T \right)^+ \mathbb{A} M^{\diamond -1} b_p^\diamond \tag{68}$$

Or in a non-condensed form:

$$\widetilde{f}^\diamond = f^\diamond - t^{\diamond T} \mathbb{A}^T \left(\mathbb{A} M^{\diamond -1} \mathbb{A}^T \right)^+ \mathbb{A} M^{\diamond -1} b_p^\diamond \tag{69}$$

As will be shown in assessments, the classical splitting leads to almost no improvement of the method while the condensed splitting can be very efficient for heterogeneous structures. In fact the initialization associated to this splitting can be proved to be optimal in a mechanical sense; it can also be obtained from the assumptions used for the primal approach.

Primal approach initialization is realized supposing that interface displacement field is zero on the condensed problem; from (66) we get:

$$\mathbb{A}^T \boldsymbol{\lambda}_{b0} + b_p^\diamond \simeq 0 \tag{70}$$

which could only be the solution if null interface displacement was the solution. Then local interface efforts are split into an equilibrated part and its remaining $\varrho^\diamond$:

$$\begin{cases} b_p^\diamond = \mathbb{A}^T \underline{\boldsymbol{\gamma}} + \varrho^\diamond \\ \underline{\boldsymbol{\gamma}} = \left(\mathbb{A} D^\diamond \mathbb{A}^T \right)^+ \mathbb{A} D^\diamond b_p^\diamond \end{cases} \tag{71}$$

$D^\diamond$ is a symmetric definite matrix, remaining $\varrho^\diamond$ is orthogonal to range($D^\diamond\underline{\mathbb{A}}$). If the system is initialized by:

$$\underline{\boldsymbol{\lambda}}_{b00} = -\underline{\boldsymbol{\gamma}} \tag{72}$$

then initial residual $\underline{\mathbb{A}}^T\underline{\boldsymbol{\lambda}}_{b00} + b_p^\diamond = -\varrho^\diamond$ is minimal in the sense of the norm associated to $D^\diamond$. If $D^\diamond = \text{diag}(K_{bb}^{\diamond})^{-1}$ then initialization is equivalent to the splitting of condensed efforts according to the stiffness of substructures ; $\text{diag}(K_{bb}^{\diamond})$ being an approximation of $S_p^{\diamond}$ that norm can be interpreted as an energy.

The initialization by the splitting of condensed efforts has to be made compatible with solid body motions by the computation of:

$$\underline{\boldsymbol{\lambda}}_{b0} = \underline{\boldsymbol{P}}\boldsymbol{\lambda}_{b00} - \underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\left(\underline{\boldsymbol{G}}^T\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}\right)^{-1}e^\diamond \tag{73}$$

Remark 3.1 *If $D^\diamond = S_d^\diamond$ was not computationally too expensive then improved initialization with Dirichlet preconditioner would lead to immediate convergence.*

Remark 3.2 *The recommended choice $D^\diamond = \text{diag}(K_{bb}^{\diamond})^{-1}$ is computationally very cheap, the heaviest operation is the computation of condensed efforts (one application of Dirichlet operator). Then if the Dirichlet preconditioner is used, new initialization is just as expensive as one preconditioning step but it can lead to significant reduction of iterations, so it should be employed. Of course if light preconditioner is preferred classical splitting should be used.*

3.3 Three fields method / A-FETI method

The A-FETI method [82, 81] can be explained as the application of the three-field strategy [11] to conforming grids, this method is widely studied in [92]. Back to (28), we have

$$S_p^\diamond u_b^\diamond = b_p^\diamond + \lambda_b^\diamond \tag{74}$$
$$\mathbb{A}\lambda_b^\diamond = 0 \tag{75}$$
$$\underline{\mathbb{A}}u_b^\diamond = \underline{0} \tag{76}$$

A-FETI method is based, like in the primal approach, on the introduction of unknown interface displacement field $\boldsymbol{u}_b$, the continuity of displacement then reads:

$$u_b^\diamond = \mathbb{A}^T\boldsymbol{u}_b \tag{77}$$

but local displacements are not eliminated like in the primal approach, complete system reads:

$$\begin{pmatrix} S_p^\diamond & -I & 0 \\ -I & 0 & \mathbb{A}^T \\ 0 & \mathbb{A} & 0 \end{pmatrix}\begin{pmatrix} u_b^\diamond \\ \lambda_b^\diamond \\ \boldsymbol{u}_b \end{pmatrix} = \begin{pmatrix} b_p^\diamond \\ 0 \\ 0 \end{pmatrix} \tag{78}$$

In order to eliminate interface displacement $\boldsymbol{u}_b$ a specific symmetric projector is introduced:

$$B = I - \mathbb{A}^T\left(\mathbb{A}\mathbb{A}^T\right)^{-1}\mathbb{A} \tag{79}$$

B realizes the orthogonal projection on $\ker(\mathbb{A})$ ($\mathbb{A}B = 0$). Since $\mathbb{A}\lambda_b^\diamond = 0$ then $\lambda_b^\diamond$ can be written as

$$\lambda_b^\diamond = B\mu_b^\diamond \tag{80}$$

$\mu_b^\diamond$ is a new interface effort, corresponding (recall $\mathbb{A}\mathbb{A}^T = \text{diag}(\text{multiplicity})$) to an average of original $\lambda_b^\diamond$. Introducing last result and using $B^T\mathbb{A}^T = 0$ to eliminate interface displacement we have

$$\begin{pmatrix} S_p^\diamond & -B \\ -B^T & 0 \end{pmatrix}\begin{pmatrix} u_b^\diamond \\ \mu_b^\diamond \end{pmatrix} = \begin{pmatrix} b_p^\diamond \\ 0 \end{pmatrix} \tag{81}$$

Then using classical elimination of local displacement by the inversion of the first line of the previous system, we get

$$u_b^\diamond = S_p^{\diamond+} \left(b_p^\diamond + B\mu_b^\diamond\right) + R_b^\diamond \alpha^\diamond \tag{82}$$

$$R_b^{\diamond T} \left(b_p^\diamond + B\mu_b^\diamond\right) = 0 \tag{83}$$

which leads to

$$\begin{pmatrix} B^T S_d^\diamond B & B^T R_b^\diamond \\ R_b^{\diamond T} B & 0 \end{pmatrix} \begin{pmatrix} \mu_b^\diamond \\ \alpha^\diamond \end{pmatrix} = \begin{pmatrix} -B^T S_p^{\diamond+} b_p^\diamond \\ -R_b^{\diamond T} b_p^\diamond \end{pmatrix} \tag{84}$$

This system is very similar to the classical dual approach system, and in consequence is solved in the same way (using projected algorithm). Anyhow the main difference is that Lagrange multiplier $\mu_b^\diamond$ is defined locally on each subdomain and not globally on the interface.

It was proved in [92] that A-FETI is mathematically equivalent to classical FETI with special choice of the $\underline{Q}$ matrix parameter of the rigid body motion projector. In fact if $\underline{Q} = \text{diag}(\frac{1}{\text{multiplicity}})$ then FETI leads to the same iterates as A-FETI. Moreover operator $\overline{B}$ is an orthonormal projector which realizes the interface equilibrium of local reactions $\mu_b^\diamond$, it can be analyzed as an orthonormal assembly operator as described in figure 7.

To sum up, A-FETI can be viewed as the conforming grid version of the three-field approach, a specific case of classical FETI, and a dual approach with non-redundant description of the connectivity interface with orthonormal assembly operator.

3.4 Mixed domain decomposition method

Mixed approaches offer a rich framework for domain decomposition methods. It enables to give a strong mechanical sense to the method, mostly by providing a behavior to the interface. The mixed approach is one of the bases of the LaTIn method [66, 67, 80, 68], a very interesting strategy designed for nonlinear analysis; as we have restrained our paper to linearized problems, we do not go further inside this method which would deserve extended survey. Several studies were realized on mixed approaches, these methods possess strong similarities, we here mostly refer to works on so-called "FETI-2-fields" method [100, 99].

The principle of the method is to rewrite the interface conditions:

$$\begin{cases} \mathbb{A}\lambda_b^\diamond = 0 \\ \underline{\mathbb{A}}u_b^\diamond = \underline{0} \end{cases} \tag{85}$$

in terms of a new local interface unknown, which is a linear combination of interface effort and displacement.

$$\mu_b^\diamond = \lambda_b^\diamond + T_b^\diamond u_b^\diamond \tag{86}$$

$\mu_b^\diamond$ is homogeneous to an effort and $T_b^\diamond$ can be interpreted as an interface stiffness. Mixed methods thus enable to give a mechanical behavior to the interface, in our case (perfect interfaces) it can be mechanically interpreted as the insertion of springs to connect substructures. New interface condition reads:

$$\mathbb{A}^T \mathbb{A} \lambda_b^\diamond + T_b^\diamond \underline{\mathbb{A}}^T \underline{\mathbb{A}} u_b^\diamond = 0^\diamond \tag{87}$$

$$\text{or} \qquad \mathbb{A}^T \mathbb{A} \mu_b^\diamond - \left(\mathbb{A}^T \mathbb{A} T_b^\diamond - T_b^\diamond \underline{\mathbb{A}}^T \underline{\mathbb{A}}\right) u_b^\diamond = 0^\diamond \tag{88}$$

It is of course necessary to study the condition for this system being equivalent to system (85). It is important to note that two conditions lying on the global interfaces (geometric and connectivity) were traced back to the local interfaces, so up to a zero-measure set

(multiple points) the conditions have the same dimension. The new condition is equivalent to the former if facing local interfaces do not hold the same information which is the case if matrix $\left(\mathbb{A}^T \mathbb{A} T_b^\diamond - T_b^\diamond \underline{\mathbb{A}}^T \underline{\mathbb{A}}\right)$ is invertible. An easy method to construct such matrices will be soon discussed

If unknown $\mu_b^\diamond$ is introduced inside local equilibrium equation, the local system reads:

$$\left(S_p^\diamond + T_b^\diamond\right) u_b^\diamond = b_p^\diamond + \mu_b^\diamond \tag{89}$$

If we assume that $T_b^\diamond$ is chosen so that $\left(S_p^\diamond + T_b^\diamond\right)$ is invertible then we have:

$$u_b^\diamond = \left(S_p^\diamond + T_b^\diamond\right)^{-1} \left(\mu_b^\diamond + b_p^\diamond\right) \tag{90}$$

Then substituting this expression inside interface condition (88), interface system reads:

$$\left(\mathbb{A}^T \mathbb{A} - \left(\mathbb{A}^T \mathbb{A} T_b^\diamond - T_b^\diamond \underline{\mathbb{A}}^T \underline{\mathbb{A}}\right) \left(S_p^\diamond + T_b^\diamond\right)^{-1}\right) \mu_b^\diamond$$
$$= \left(\mathbb{A}^T \mathbb{A} T_b^\diamond - T_b^\diamond \underline{\mathbb{A}}^T \underline{\mathbb{A}}\right) \left(S_p^\diamond + T_b^\diamond\right)^{-1} b_p^\diamond \quad (91)$$

so mixed approaches have the originality to rewrite global interface conditions on the local interfaces and to look for purely local unknown (which means that the size of the unknown is about twice the size of the unknown in classical primal or dual methods).

This general scheme for mixed methods has, as far as we know, never been employed. A first reason is that it leads to certain programming complexity, second the manipulation of zero-measure interfaces is not easy for methods aiming at introducing strong mechanical sense and hard to justify from a mathematical point of view. So most often a simplified method is preferred, which takes only into account non-zero-measure interfaces. Such an approach simplifies the connectivity description of the interface, every relationship on the interface only deals with couples of subdomains. In order to have the clearer expression possible, we present the algorithm in the two subdomains case. Interface equilibrium reads:

$$\begin{cases} u_b^{(1)} - u_b^{(2)} = 0 \\ \lambda_b^{(1)} + \lambda_b^{(2)} = 0 \end{cases} \tag{92}$$

which is equivalent to

$$\begin{cases} \lambda_b^{(1)} + \lambda_b^{(2)} + T^{(1)} \left(u_b^{(1)} - u_b^{(2)}\right) = 0 \\ \lambda_b^{(1)} + \lambda_b^{(2)} + T^{(2)} \left(u_b^{(2)} - u_b^{(1)}\right) = 0 \end{cases} \tag{93}$$

under the condition of invertibility of $\left(T^{(1)} + T^{(2)}\right)$. Introducing unknown $\mu_b^{(s)} = \lambda_b^{(s)} + T^{(s)} u^{(s)}$ interface system reads:

$$\begin{cases} \mu_b^{(1)} + \mu_b^{(2)} - \left(T^{(1)} + T^{(2)}\right) u^{(2)} = 0 \\ \mu_b^{(1)} + \mu_b^{(2)} - \left(T^{(1)} + T^{(2)}\right) u^{(1)} = 0 \end{cases} \tag{94}$$

Local equilibrium reads:

$$\begin{cases} \left(S_p^{(1)} + T^{(1)}\right) u_b^{(1)} = \mu_b^{(1)} + b_p^{(1)} \\ \left(S_p^{(2)} + T^{(2)}\right) u_b^{(2)} = \mu_b^{(2)} + b_p^{(2)} \end{cases} \tag{95}$$

Assuming $T^{(s)}$ is chosen so that matrix $\left(S_p^{(s)} + T^{(s)}\right)$ is invertible, we can express displacement $u_b^{(s)}$ from local equilibrium equation, and suppress it from global interface conditions, which leads to:

$$
\begin{pmatrix} I & I - \left(T^{(1)} + T^{(2)}\right)\left(S_p^{(2)} + T^{(2)}\right)^{-1} \\ \left(T^{(1)} + T^{(2)}\right)\left(S_p^{(1)} + T^{(1)}\right)^{-1} & I \end{pmatrix} \begin{pmatrix} \mu^{(1)} \\ \mu^{(2)} \end{pmatrix} =
$$
$$
\begin{pmatrix} \left(T^{(1)} + T^{(2)}\right)\left(S_p^{(2)} + T^{(2)}\right)^{-1} b_p^{(2)} \\ \left(T^{(1)} + T^{(2)}\right)\left(S_p^{(1)} + T^{(1)}\right)^{-1} b_p^{(1)} \end{pmatrix} \quad (96)
$$

This expression enables to give better interpretation of the stiffness parameters $T^{(s)}$. Suppose $T^{(1)} = S_p^{(2)}$ and $T^{(2)} = S_p^{(1)}$ then matrix (96) is equal to identity and solution is directly achieved. So the aim of matrix $T^{(s)}$ is to provide one substructure with the interface stiffness information of the other substructures.

If we generalize to N-subdomain system (91), we can deduce that the optimal choice for $T^{(s)}$ is the Schur complement of the remaining substructures on the interface of domain (s) (some kind of $S_p^{(\bar{s})}$ where $\bar{s}$ denotes all the substructures but s). Of course such a choice is not computationally feasible (mostly because it does not respect the localization of data), and approximations have to be considered. In decreasing numerical efficiency and computational cost order, we have:

- Approximate the Schur complement of the remaining of the substructure by the Schur complement of the neighbor;

- approximate the Schur complement of the neighbor by the Schur complement of the nearer nodes of the neighbor ("strip"-preconditioners which idea is developed in [83] in another context);

- approximate the Schur complement of the neighbor by the stiffness matrix of the interface of the neighbor (strategy of dual approach lumped preconditioner).

The second strategy is quite a good compromise: it respects data localization, it is not computationally too expensive and yet it enables the propagation of the information beyond the interface. Of course an important parameter is the definition of elements "near the interface", which can be realized giving an integer n representing the number of layers of elements over the interface.

3.4.1 Coarse problem

Because the interface stiffness parameter $T^\diamond$ regularizes local operators $S_p^\diamond$, local operator $\left(S_p^\diamond + T^\diamond\right)$ is always invertible. Such a property can be viewed as an advantage because it simplifies the implementation of the method introducing no kernel and generalized inverse; but it also can be considered as a disadvantage because no more coarse problem enables global transmission of data among the structure. Then the communications inducted by this method are always neighbor-to-neighbor which means that the transmission of a localized perturbation to far substructures is always a slow process. It is then necessary to add an optional coarse problem (see section A.7). Most often the optional coarse problem is constituted of would-be rigid body motions (if subdomains had not been regularized). Another possibility, which is proposed inside the LaTIn method is to use rigid body motions and extension modes of each interface as coarse problems, this leads to much larger

coarse space. The coarse matrix corresponds to the virtual works of first order of deformation of substructures; so mechanically it realizes and propagates a numerical first order homogenization of the substructures.

3.5 Hybrid approach

The hybrid approach (see [57] for a specific application) is a proposition to provide a unifying scheme for primal and dual approaches though it could easily be extended to other strategies. It relies on the choice for each interface degree of freedom of its own treatment (for now primal or dual). So let us define two subsets of interface degrees of freedom: the first is submitted to primal conditions (subscript p) and the second to Neumann conditions (subscript d). Local equilibrium then reads ($\bar{p} = i \cup d$, $b = d \cup p$, $p \cap d = \emptyset$):

$$\begin{pmatrix} K_{\bar{p}\bar{p}}^{\diamond} & K_{\bar{p}p}^{\diamond} \\ K_{p\bar{p}}^{\diamond} & K_{pp}^{\diamond} \end{pmatrix} \begin{pmatrix} u_{\bar{p}}^{\diamond} \\ u_p^{\diamond} \end{pmatrix} = \begin{pmatrix} f_{\bar{p}}^{\diamond} \\ f_p^{\diamond} \end{pmatrix} + \begin{pmatrix} t_d^{\diamond T}\lambda_d^{\diamond} \\ \lambda_p^{\diamond} \end{pmatrix} \tag{97}$$

Preferred interface unknowns are unique displacement on the first subset $\boldsymbol{u}_p$ and unique effort on the second subset $\boldsymbol{\lambda}_d$. Local contributions then reads:

$$u_p^{\diamond} = \mathbb{A}_p^T \boldsymbol{u}_p \tag{98}$$

$$\lambda_d^{\diamond} = \underline{\mathbb{A}}_d^T \boldsymbol{\lambda}_d \tag{99}$$

which ensure the continuity of displacement on the p degrees of freedom and the action-reaction principle on the d degrees of freedom, of course operators $\mathbb{A}_p$ and $\underline{\mathbb{A}}_d$ have been restricted respectfully to the p and d subsets. Remaining interface conditions read:

$$\underline{\mathbb{A}}_d u_d^{\diamond} = \underline{\mathbb{A}}_d t_d^{\diamond} u_{\bar{p}}^{\diamond} = 0 \tag{100}$$

$$\mathbb{A}_p \lambda_p^{\diamond} = 0 \tag{101}$$

To obtain the global interface system, first local unknown $u_{\bar{p}}^{\diamond}$ has to be eliminated:

$$u_{\bar{p}}^{\diamond} = K_{\bar{p}\bar{p}}^{\diamond +} \left(f_{\bar{p}}^{\diamond} + t_d^{\diamond T}\lambda_d^{\diamond} - K_{\bar{p}p}^{\diamond} u_p^{\diamond} \right) + R_{\bar{p}}^{\diamond}\alpha^{\diamond} \tag{102}$$

Applying continuity condition to previous result and equilibrium condition to the second row of (97), interface system reads:

$$\begin{pmatrix} \boldsymbol{S}_{p\underline{d}} & \begin{pmatrix} \boldsymbol{G}_p \\ \underline{\boldsymbol{G}}_d \end{pmatrix} \\ (-\boldsymbol{G}_p^T \quad \underline{\boldsymbol{G}}_d^T) & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_p \\ \boldsymbol{\lambda}_d \\ \alpha^{\diamond} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_p \\ -\underline{\boldsymbol{b}}_d \\ -e^{\diamond} \end{pmatrix} \tag{103}$$

with the following notations:

$$\boldsymbol{S}_{p\underline{d}} = \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \underline{\mathbb{A}}_d \end{pmatrix} S_{pd}^{\diamond} \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \underline{\mathbb{A}}_d \end{pmatrix}^T$$

$$\boldsymbol{G}_p = \mathbb{A}_p K_{p\bar{p}}^{\diamond} R_{\bar{p}}^{\diamond}, \ \underline{\boldsymbol{G}}_d = \underline{\mathbb{A}}_d t_d^{\diamond} R_{\bar{p}}^{\diamond}$$

$$e^{\diamond} = R_{\bar{p}}^{\diamond T} f_{\bar{p}}^{\diamond}$$

$$\boldsymbol{b}_p = \mathbb{A}_p \left(f_p^{\diamond} - K_{p\bar{p}}^{\diamond} K_{\bar{p}\bar{p}}^{\diamond +} f_{\bar{p}}^{\diamond} \right), \ \underline{\boldsymbol{b}}_d = \underline{\mathbb{A}}_d t_d^{\diamond} K_{\bar{p}\bar{p}}^{\diamond +} f_{\bar{p}}^{\diamond}$$

This interface problem corresponds to the constrained resolution of one linear system. The constraint is linked to the possible non-invertibility of matrix $K_{\bar{p}\bar{p}}^{\diamond}$ and thus to the choice

of primal subset. Notice that $\boldsymbol{G}_p$ represents the reaction of primal degrees of freedom to zero energy modes of $K_{\bar{p}\bar{p}}^{\diamond}$ and then should be zero in most cases (it may be non zero in buckling cases). Moreover this system may represent classical primal approach (if all interface degrees of freedom are in subset p) or classical dual approach (if all interface degrees of freedom are in subset d). Operator $\boldsymbol{S}_{p\underline{d}}$ is a primal/dual Schur complement, it is the sum of local contributions $S_{pd}^{\diamond}$ (25).

The above system is nonsymmetric semi-definite (because of redundancies on the dual subset) positive, it has to be solved by GMRes-like algorithm.

3.5.1 Hybrid preconditioner

Inspired by primal and dual preconditioners, we propose to approximate the inverse of the sum of local contributions by a scaled sum of local inverses.

$$\tilde{\boldsymbol{S}}_{p\underline{d}}^{-1} = \begin{pmatrix} \tilde{\mathbb{A}}_p & 0 \\ 0 & \underline{\tilde{\mathbb{A}}}_d \end{pmatrix} S_{dp}^{\diamond} \begin{pmatrix} \tilde{\mathbb{A}}_p & 0 \\ 0 & \underline{\tilde{\mathbb{A}}}_d \end{pmatrix}^{T} \tag{104}$$

Scaled assembly operator are defined in the same way as in primal and dual approaches.

3.5.2 Coarse problems

As said earlier, depending on the choice of subset p, local operator $K_{\bar{p}\bar{p}}^{\diamond}$ (involved in the computation of $S_{pd}^{\diamond}$) may not be invertible and, like in dual approach, a first coarse correction has been incorporated inside the hybrid formulation. Anyhow local operator $K_{\bar{d}\bar{d}}^{\diamond}$ involved in preconditioning step may also not be invertible and, like in primal approach, a second coarse problem has to be added to make the preconditioner optimal. Then, the optimal version of the hybrid system incorporates two coarse problems handled by specific initialization/projection algorithm. The admissibility condition reads:

$$\boldsymbol{G}^{T}\boldsymbol{x} = e \tag{105}$$

with $\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_p \\ \underline{\boldsymbol{G}}_d \end{pmatrix}$, $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{u}_p \\ \underline{\boldsymbol{\lambda}}_d \end{pmatrix}$ and $\boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}_p \\ -\underline{\boldsymbol{b}}_d \end{pmatrix}$. If $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{S}_{p\underline{d}}\boldsymbol{x}$ stands for the residual before preconditioning, the optimality condition reads:

$$\boldsymbol{H}^{T}\boldsymbol{r} = \begin{pmatrix} \boldsymbol{H}_p \\ \underline{\boldsymbol{H}}_d \end{pmatrix}^{T} \begin{pmatrix} \boldsymbol{r}_p \\ \boldsymbol{r}_d \end{pmatrix} = 0 \tag{106}$$

with $\boldsymbol{H}_p = \tilde{\mathbb{A}}_p t_p^{\diamond} R_{\bar{d}}^{\diamond}$ and $\underline{\boldsymbol{H}}_d = \underline{\tilde{\mathbb{A}}}_d K_{d\bar{d}}^{\diamond} R_{\bar{d}}^{\diamond}$ (as said before most often $\underline{\boldsymbol{H}}_d = 0$). To sum up constraints:

$$\begin{cases} \boldsymbol{G}^{T}\boldsymbol{x} = -e^{\diamond} \\ \boldsymbol{H}^{T}\boldsymbol{S}_{p\underline{d}}\boldsymbol{x} = \boldsymbol{H}^{T}\boldsymbol{b} \end{cases} \tag{107}$$

Handling such constraints is described in section A.8.

Figure 10 provides a schematic representation of the first iteration of the preconditioned hybrid approach, in the specific case of a nodal partition of the interface. Assessments will deal with partition at the degree of freedom level.

3.5.3 Error estimate

Because GMRes-like solver is used, the euclidian norm of the residual is directly available, such a norm is the sum of displacement gap on the d part of the interface and an effort gap on the p part of the interface. For now, no other estimator with better physical sense is available.
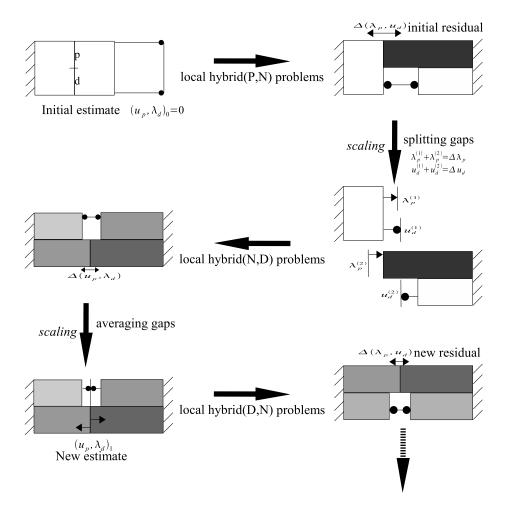
Figure 10. Representation of first iteration of preconditioned hybrid approach

4 Adding optional constraints

The aim of optional constraints is to ensure that the research space for the iterative solver possesses a certain regularity. The choice of constraints will be discussed in a later section. Going back to the interface system:

$$\begin{pmatrix} S_p^\diamond & -I^\diamond \\ \underline{\mathbb{A}} & 0 \\ 0 & \mathbb{A} \end{pmatrix} \begin{pmatrix} u_b^\diamond \\ \lambda_b^\diamond \end{pmatrix} = \begin{pmatrix} b_p^\diamond \\ \underline{0} \\ 0 \end{pmatrix} \tag{108}$$

any constraint of the form $\underline{C}^T \underline{\mathbb{A}} u_b^\diamond = 0$ or $C^T \mathbb{A} \lambda_b^\diamond = 0$ is trivially verified by the solution fields, it is just a restriction of continuity/equilibrium conditions. From an iterative process point of view, these conditions will be reached once converged ; the principle of optional constraints is to have every iteration verify that condition.

There are two classical solutions to ensure these optional constraints: either to realize a splitting of research space and ensure, using a projector, that the resolution is limited to convenient subspace, or to realize a condensation of constraints and make iterations

on a smaller space. In other words suppose there are n_c independent constraints in a n-dimension space the first strategy researches n-sized solution in a $(n - n_c)$-ranked space, while the second solution researches $(n-n_c)$-sized solution in a $(n-n_c)$-dimension space then deduce the n-sized solution. From a numerical point of view both solutions are equivalent, they are just two ways of handling the same constraints, anyhow from implementation and computational points of view they have strong differences.

We will essentially focus on application to primal and dual domain decomposition methods.

4.1 Augmentation strategy

For this strategy the constraint is reinterpreted in terms of constraint on the residual. Typically, the primal approach can be augmented by constraints on the effort field while the dual approach can have constraints on the displacement field:

$$\boldsymbol{C}^T \mathbb{A} \lambda_b^\diamond = -\boldsymbol{C}^T \left(\mathbb{A} b_p^\diamond - \mathbb{A} S_p^\diamond \mathbb{A}^T \boldsymbol{u}_b \right) \tag{109}$$

$$\underline{\boldsymbol{C}}^T \mathbb{A} u_b^\diamond = \underline{\boldsymbol{C}}^T \left(\mathbb{A} S_d^\diamond \mathbb{A}^T \underline{\boldsymbol{\lambda}}_b + \underline{b}_d^\diamond \right) \tag{110}$$

The constraint is then handled as an augmentation inside the iterative solver, its is classically realized using a projector (see sections A.7 and A.8).

4.2 Recondensation strategy

This strategy was recently introduced in the framework of the dual approach, leading to the FETIDP algorithm [34, 35, 69, 63]. Because for now only constraints on the $u_b^\diamond$ field have been considered we will restrain to this kind of constraints, the application of constraints on $\lambda_b^\diamond$ is straightforward.

4.2.1 Basic method

We first consider constraints which impose continuity of specific degrees of freedom, in other terms we suppose matrix $\underline{C}$ is identity on certain degrees of freedom and zero elsewhere; we will show how any constraint can be rewritten in such a form. Because these degrees of freedom will be submitted to a primal treatment we will denote them with subscript p while the remaining of the interface will be denoted with subscript d. Constraint reads:

$$0 = \underline{\boldsymbol{C}}^T \mathbb{A} u_b^\diamond = \begin{pmatrix} 0 \\ I_p \end{pmatrix}^T \begin{pmatrix} \mathbb{A}_d u_d^\diamond \\ \mathbb{A}_p u_p^\diamond \end{pmatrix} = \mathbb{A}_p u_p^\diamond \tag{111}$$

Like in the hybrid approach this constraint is verified using a unique displacement field on the primal part of the interface: $u_p^\diamond = \mathbb{A}_p^T \boldsymbol{u}_p$. Interface system then reads like in the hybrid approach, with the additional assumption that the constraints are so that the local problem possesses enough Dirichlet conditions to make it invertible.

$$\boldsymbol{S}_{p\underline{d}} \begin{pmatrix} \boldsymbol{u}_p \\ \underline{\boldsymbol{\lambda}}_d \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_p \\ -\underline{\boldsymbol{b}}_d \end{pmatrix} \tag{112}$$

Introducing following notations for blocks composing $S_{pd}^\diamond$:

$$S_{pd}^\diamond = \begin{pmatrix} s_{pp}^\diamond & s_{pd}^\diamond \\ -s_{dp}^\diamond & s_{dd}^\diamond \end{pmatrix} \tag{113}$$

then

$$\boldsymbol{S}_{p\underline{d}} = \begin{pmatrix} \boldsymbol{s}_{pp} & \boldsymbol{s}_{p\underline{d}} \\ -\boldsymbol{s}_{dp} & \boldsymbol{s}_{\underline{dd}} \end{pmatrix} = \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \underline{\mathbb{A}}_d \end{pmatrix} \begin{pmatrix} s_{pp}^\diamond & s_{pd}^\diamond \\ -s_{dp}^\diamond & s_{dd}^\diamond \end{pmatrix} \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \underline{\mathbb{A}}_d \end{pmatrix}^T \tag{114}$$

Unknown $\boldsymbol{u}_p$ is condensed on the remaining of the interface:

$$\boldsymbol{u}_p \;=\; \boldsymbol{s}_{pp}^{-1}\left(\boldsymbol{b}_p - \boldsymbol{s}_{p\underline{d}}\boldsymbol{\lambda}_d\right) \tag{115}$$

$$\underline{\boldsymbol{s}}_d\boldsymbol{\lambda}_d = \left(\boldsymbol{s}_{\underline{dd}} + \boldsymbol{s}_{\underline{d}p}\boldsymbol{s}_{pp}^{-1}\boldsymbol{s}_{p\underline{d}}\right)\boldsymbol{\lambda}_d \;=\; -\underline{\boldsymbol{b}}_d + \boldsymbol{s}_{\underline{d}p}\boldsymbol{s}_{pp}^{-1}\boldsymbol{b}_p \tag{116}$$

The latest equation is solved using an iterative solver, operator $\underline{\boldsymbol{s}}_d$ has the same properties as the restriction of the dual operator to the d-part of the interface (semi-definition, symmetry and positivity). Operator $\underline{\boldsymbol{s}}_d$ is the assembly of local contributions

$$\underline{\boldsymbol{s}}_d = \mathbb{A}_d s_d^{\diamond}\mathbb{A}_d^{T} = \mathbb{A}_d\left(s_{dd}^{\diamond} + s_{dp}^{\diamond}\mathbb{A}_p^{T}\boldsymbol{s}_{pp}^{-1}\mathbb{A}_p s_{pd}^{\diamond}\right)\underline{\mathbb{A}}_d^{T} \tag{117}$$

Using operator $\underline{\boldsymbol{s}}_d$ requires the computation of the inverse of matrix $\boldsymbol{s}_{pp} = \mathbb{A}_p s_{pp}^{\diamond}\mathbb{A}_p^{T}$, which is an assembled matrix. Then this formulation includes a global coarse problem set on primal variables.

The recommended preconditioner for such an approach is directly inspired by the dual approach: it consists in solving local Dirichlet problems with scaled imposed displacement on the d-part of the interface and null displacement on the p part of the interface and extracting the average reaction of the d-part of the interface. Then the preconditioner reads:

$$\underline{\tilde{\boldsymbol{s}}}_d^{-1} = \begin{pmatrix} 0_p & \tilde{\mathbb{A}} \end{pmatrix} S_p^{\diamond}\begin{pmatrix} 0_p & \tilde{\mathbb{A}} \end{pmatrix}^{T} \tag{118}$$

Figure 11 provides schematic representation of the first iteration of preconditioned FETID method.

4.2.2 More complex constraints

We now consider constraints which are not limited to one degree of freedom, for instance one can consider that we want to ensure that the average jump of displacement on one edge is equal to zero, which involves all the degrees of freedom of the edge.

The classical solution [63] is to realize a change of basis of degrees of freedom (denoted by matrix $T^{\diamond}$) so that each constraint is represented by one "modified" degree of freedom. The change of basis is the same local operation realized on every subdomain, then we can define a global change $\boldsymbol{T}$ so that $\mathbb{A}T^{\diamond} = \boldsymbol{T}\mathbb{A}$

$$\boldsymbol{C}^{T}\mathbb{A}u_b^{\diamond} = \boldsymbol{C}^{T}\mathbb{A}T^{\diamond}\hat{u}_b^{\diamond} = \boldsymbol{C}^{T}\boldsymbol{T}\mathbb{A}\hat{u}_b^{\diamond} \;=\; \hat{\boldsymbol{C}}^{T}\mathbb{A}\hat{u}_b^{\diamond} \tag{119}$$

$$\text{with } \hat{\boldsymbol{C}} \;=\; \begin{pmatrix} 0 \\ I_{\hat{p}} \end{pmatrix} \tag{120}$$

After the change of basis is realized the same algorithm can be employed. Because constraints most often respect a certain locality of data (for instance independent constraints on each edge), change of basis is not a too expensive operation, and does not make too poor the sparsity of local matrices.

4.3 Adding "constraints" to the preconditioner

This subsection deals with the dualization of the recondensation strategies [19, 23]. For instance the balanced domain decomposition with constraints (BDDC) is a primal version of FETIDP: during the preconditioning step, the continuity of displacement is ensured at specific degrees of freedom (which can be the result of a local change of basis), so that the local Neumann operator remains fully invertible. So solving classical primal approach problem

$$\boldsymbol{S}_p\boldsymbol{u}_b = \mathbb{A}S_p^{\diamond}\mathbb{A}^{T}\boldsymbol{u}_b = \boldsymbol{b}_p \tag{121}$$
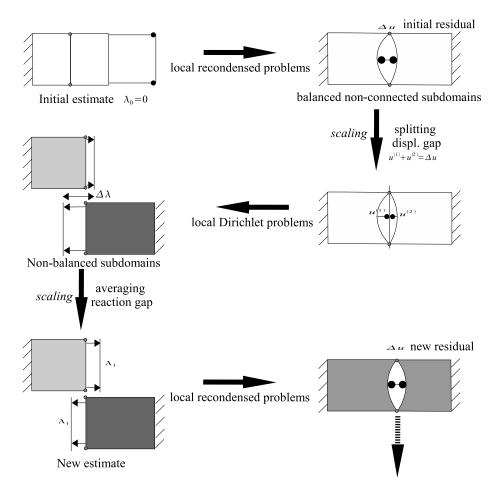
Figure 11. Representation of the first iteration of preconditioned FETIDP

the preconditioner reads

$$\tilde{\boldsymbol{S}}_p^{-1} = \begin{pmatrix} I_p & 0 \\ 0 & \tilde{\mathbb{A}}_d \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_{pp}^{-1} & -\boldsymbol{s}_{pp}^{-1}\mathbb{A}_p s_{pd}^{\diamond} \\ -s_{dp}^{\diamond}\mathbb{A}_p^T \boldsymbol{s}_{pp}^{-1} & s_d^{\diamond} \end{pmatrix} \begin{pmatrix} I_p & 0 \\ 0 & \tilde{\mathbb{A}}_d \end{pmatrix}^T \tag{122}$$

Figure 11 provides a schematic representation of the first iteration of preconditioned BDDC method.

5 Classical issues

In this section we try to provide answers to questions that commonly arise when using domain decomposition methods.

5.1 Rigid body motion detection

Handling floating substructures is definitely a very accurate issue. This difficulty is one of the reason of the success of methods which regularize the stiffness of subdomains like FETIDP or mixed approaches, leading to fully invertible matrices. Anyhow basic primal and dual methods remain competitive (mostly because zero-energy modes provide a very
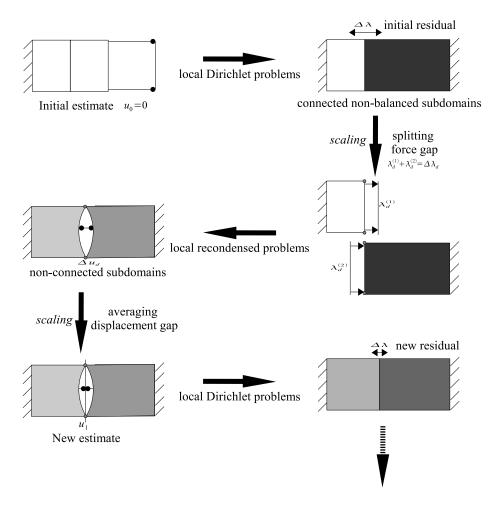
Figure 12. Representation of the first iteration of preconditioned BDDC

natural coarse problem), hence providing an efficient algorithm for the computation of rigid body motion is essential. Many strategies can be used [33, 40], and this review does not claim to be exhaustive.

First we have to discuss the exact composition of the possible kernel of substructures. What can be found is:

- rigid body motions of floating substructures,

- internal mechanisms of substructures,

- weird things due to nonlinearities (buckling, exotic behaviors, ...)

- numerical zero-energy modes.

An internal mechanism can for instance occur when a substructure is made out of two parts connected by one linear edge (pivot) or one singular point (kneecap). Methods exist to avoid such substructures either inside the decomposition algorithm or as external programs regularizing a given decomposition [48], and then should be employed.

The last two kinds of kernel are non-standard and can only be detected using fully algebraic methods (like Gauss pivoting). The problem with algebraic methods is their high sensitivity to the condition number of the stiffness matrix of substructures. The condition number can be influenced by the aspect ratio and the material composition of the substructures, even after adimensionalization the quality of the methods is hard to warranty.

Finally we only develop here two strategies to handle zero-energy modes. The first one belongs to the purely algebraic methods, it is very simple to implement and can lead to satisfying results for not-too-complex problems, it can handle more than solid body motions but it is strongly dependent on *a priori* selected degrees of freedom. The second method is purely geometric, it is very robust but only suited to detect solid body motions.

In order to simplify notations, we consider the research of the zero-energy modes of matrix K (which should be a local stiffness matrix).

5.1.1 Simple algebraic approach

This method is based on fundamental relationship between the kernel of a matrix and the kernel of Schur complement (15). The principle is to preselect a small set of degrees of freedom which we will denote by subscript N (the other degrees of freedom are denoted with subscript O). Then compute explicitly primal Schur complement associated to these degrees of freedom: $S = K_{NN} - K_{NO}K_{OO}^{-1}K_{ON}$). If N-degrees of freedom are selected so that K_{00} is invertible (if only solid body motions have to be detected then it is sufficient to take the degrees of freedom associated to three non-aligned nodes) then S is well defined and its kernel is linked to the kernel of matrix K.

Since N is a "small" set (12 degrees of freedom is often sufficient), computing the kernel of matrix S using "exact" algorithm like singular value decomposition is rather cheap and then kernel of matrix K can be computed using equation (15).

5.1.2 Geometric approach

The basic idea of this method is that kinematically admissible solid body motions can be deduced from boundary conditions imposed on one subdomain. Let R_c be a basis of candidate rigid body motions (would be solid body motions if no Dirichlet conditions were applied on the subdomain, R_c is a 6-column matrix in 3D and 3-column matrix in 2D), and let E be the matrix of Dirichlet boundary conditions: each column of E represents one (combination of) blocked degree of freedom. Kinematically admissible rigid body motions R are linear combinations of candidate rigid body motions (hence $R = R_c Q$) which do not make Dirichlet boundary conditions work (*ie* $E^T R = 0$). In order to find such linear combination, we compute singular value decomposition of matrix $E^T R_c = U D V^T$ and set $Q = V_0$ where V_0 is the submatrix of V associated to negligible singular values. Because $E^T R_c$ is a matrix only made out of geometric considerations, it is well conditioned and the criterion to detect "zero" singular values is well defined (there is a large gap between zero and non-zero singular values).

5.1.3 Generalized inverse

The two methods presented above led to r-ranked basis R of the kernel of matrix K. To compute generalized inverse K^+, the most classical way is to select r degrees of freedom, so that if they were added Dirichlet conditions, rigidity matrix would be well defined. For instance after pivoting and renumber one could get $R = \begin{pmatrix} R^* \\ I_r \end{pmatrix}$ and then the r last degrees

of freedom would suit. Then we have:

$$K = \begin{pmatrix} K_{\bar{r}\bar{r}} & K_{\bar{r}r} \\ K_{r\bar{r}} & K_{rr} \end{pmatrix} \qquad \text{and} \qquad K^+ = \begin{pmatrix} K_{\bar{r}\bar{r}}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \qquad (123)$$

This is just one instance of generalized inverse, other can be built using penalization or other modification to matrix K. Though not theoretically prohibited, choosing "blocked" degrees of freedom on the interface is often a bad idea from a practical point of view.

5.2 Choice of optional constraints

As seen in section 4, constraints can be imposed either using augmentation (using one or two projectors, sections A.7 and A.8) or using recondensation algorithms. In the case of recondensation algorithms, constraints have to be sufficient in order to suppress rigid body motions and then regularize the local stiffness matrix. In the case of augmentation algorithm, constraints have to be independent from rigid body motions which are already handled by the formulation.

Because constraints are expensive to handle, they have to be chosen with care; anyhow, except in a few cases, there are no general results on how to choose them. Mechanical comprehension of studied phenomena and anticipation of convergence difficulties may lead to efficient strategies. In the case of solving several linear systems (even with different left hand sides) interesting strategies exist [98, 30, 58, 84, 88].

The next two subsections deal with very common strategies, while the last subsection describes another framework for constraints inspired by the LaTIn method [66, 80].

5.2.1 Forth order elasticity

As plate and shell models are often used in structural mechanics, forth order problems have been carefully studied [38, 32, 94, 103, 40]. Such problems are characterized by the appearance of singularities at the corner of substructures (so-called "corner modes") which are destroying the scalability of usual methods. The classical solution consists in enforcing the continuity of the (most often only normal) displacement field at the corners in order to regularize the problem. From a practical point of view, corners are most often defined as multiple points (nodes shared by more than two substructures), that set can be enriched by extremities of edges.

The implementation, just like the singularity, is strongly linked to the chosen formulation.

Dual approach: since the projected residual corresponds to the displacement jump between substructures (54), one just has to use augmentation algorithm with one constraint for each pair of neighbor at each corner. Matrix $\underline{\underline{C}}$ is then made out of columns with one coefficient 1 on one corner degree of freedom and 0 elsewhere. Because the dual description of a m-multiple point leads to $(m-1)$ relationships, such a coarse space is rather large.

Primal approach: In order to regularize the displacement field, the constraints have to be imposed on the preconditioned residual (assuming Neumann-Neumann preconditioner is employed). The aim is then to have the local contributions of preconditioned residual equal to zero on corner points. So if $C^\diamond$ denotes the local interface matrix made out of columns with one coefficient 1 on one corner degree of freedom and 0 elsewhere, and $\tilde{C}^\diamond$ the same matrix scaled according to the scaling used inside the preconditioner, the constraints read $C = \tilde{\mathbb{A}} S_d^\diamond \tilde{C}^\diamond$. Then a m-multiple degree of freedom leads to m constraints.

Recondensed approaches: FETIDP or BDDC were first designed in this context, from the consideration that (extended) corners constraints were sufficient to suppress rigid body motions, then the first level constraints could be avoided. So the methods directly apply since they consist in constraining the displacement field. Here whatever the multiplicity of a corner may be, it always leads to just 1 constraint.

5.2.2 Second order elasticity

Because classical methods are already scalable in the frame of second order elasticity, optional constraints are not often used in such a context. Furthermore, it is hard to predict what constraint should be imposed. In some cases, efficient strategies have been proposed, such as in [58] for nonlinear problems using Newton-Raphson solver where approximations of eigen vectors are used.

The question of optional constraints arose when willing to extend recondensed algorithms (FETIDP and BDDC) to such problems, mostly because the previous definition of corners lead to significant problems in 3D (too many constraints, poor convergence...). The first solution was proposed in [69], the idea was to select 3 non-aligned nodes on each face (interface between 2 subdomains) which maximized the surface of the triangle they defined. The current solution, the scalability of which is mathematically and numerically proved, is to enforce average convergence on edges [63], which is realized by a change of basis described in (119). In order to take into account heterogeneity on the interface, the average may be scaled by a coefficient representing the stiffness of the subdomains. For more difficult problems, first order moments of edges can also be added.

5.2.3 Link with homogenization theory

This paragraph intends to present a mechanical vision of optional constraint which, though hard to implement in the framework of the presented method, may lead to better understanding of what optional constraints and associated coarse problems can provide to the methods. This analysis is inspired by the multiscale version of the LaTIn method [66].

The underlying question when choosing optional constraints (except from specific numerical questions like in the forth order elasticity) is "what global information should be transmitted to the whole structure ?" or more precisely "what should far substructures know from one substructure". A meaningful answer is provided by Saint-Venant principle and homogenization theory: at a first order development, a substructure can be represented by its rigid body motions and its constant strain states (simple traction and shearing states). Such an idea adds six (3 in 2D) more constraints per subdomains; as these constraints are somehow complex to build, they can be approximated by interface modes (but of course the number of constraints then grows quickly).

5.3 Linear Multiple Points Constrains

Multiple points constraints are relationships defined between some degrees of freedom, they are often used in order to connect nonconforming meshes, to represent boundary conditions (for instance periodicity), to model contact or apply control laws. In the case of linear(ized) constraints we can write, on the whole structure scale:

$$Ku = f \tag{124}$$
$$Cu = a \tag{125}$$

What seems most suited to the domain decomposition context is to dualize the constraint and introduce Lagrange multiplier μ in order to enforce the condition. System then reads:

$$\begin{pmatrix} K & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} u \\ \mu \end{pmatrix} = \begin{pmatrix} f \\ a \end{pmatrix} \tag{126}$$

After decomposition we have

$$
\begin{pmatrix} K^\diamond & -I^\diamond & \mathbb{C}^T \\ \underline{\mathbb{A}}t^\diamond & 0 & 0 \\ 0 & \mathbb{A}t^\diamond & 0 \\ \mathbb{C} & 0 & 0 \end{pmatrix} \begin{pmatrix} u^\diamond \\ \lambda^\diamond \\ \mu \end{pmatrix} = \begin{pmatrix} f^\diamond \\ \underline{0} \\ 0 \\ a \end{pmatrix}
\tag{127}
$$

with $\mathbb{C}$ so that $\mathbb{C}u^\diamond = Cu = a$ which implies (since $u_b^\diamond = \mathbb{A}^T u_b$):

$$
Cu = \begin{pmatrix} \mathbb{C}_i & C_b \end{pmatrix} \begin{pmatrix} u_i^\diamond \\ u_b \end{pmatrix} = \mathbb{C}u^\diamond = \begin{pmatrix} \mathbb{C}_i & \mathbb{C}_b \end{pmatrix} \begin{pmatrix} u_i^\diamond \\ u_b^\diamond \end{pmatrix}
\tag{128}
$$

$$
\text{then} \qquad \mathbb{C}_b \mathbb{A}^T = C_b
\tag{129}
$$

Or in other words, if matrix C deals with interface degrees of freedom, the associated constraints have to be correctly distributed between sharing subdomains. The constraint can be interpreted as specific (non-boolean) assembly operator which explains the chosen notation. Using MPCs with domain decomposition methods was studied in [93] in the dual method context.

In order to provide general methodology to apply MPCs, we then incorporate inside hybrid domain decomposition method (equations (97) to (100)):

$$
\begin{pmatrix} K_{\bar{p}\bar{p}}^\diamond & K_{\bar{p}p}^\diamond \mathbb{A}_p^T & \mathbb{C}_{\bar{p}}^T \\ \mathbb{A}_p K_{p\bar{p}}^\diamond & \mathbb{A}_p K_{pp}^\diamond \mathbb{A}_p^T & \boldsymbol{C}_p^T \\ \mathbb{C}_{\bar{p}} & \boldsymbol{C}_p & 0 \end{pmatrix} \begin{pmatrix} u_{\bar{p}}^\diamond \\ \boldsymbol{u}_p \\ \mu \end{pmatrix} = \begin{pmatrix} f_{\bar{p}}^\diamond + t_d^{\diamond T} \underline{\mathbb{A}}_d^T \lambda_d \\ \mathbb{A}_p f_p^\diamond \\ a \end{pmatrix}
\tag{130}
$$

$$
\underline{\mathbb{A}}_d t_d^\diamond u_{\bar{p}}^\diamond = \underline{0}
\tag{131}
$$

The elimination of $u_{\bar{p}}^\diamond$ leads to, with classical hybrid notations:

$$
\begin{pmatrix} \boldsymbol{S}_{p\underline{d}} & & \mathcal{C}_p^T & \boldsymbol{G}_p \\ & & \mathcal{C}_d^T & \underline{\boldsymbol{G}}_d \\ -\mathcal{C}_p & \mathcal{C}_d & \mathcal{C}_\mu & \boldsymbol{G}_\alpha \\ -\boldsymbol{G}_p^T & \underline{\boldsymbol{G}}_d^T & \boldsymbol{G}_\alpha^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_p \\ \underline{\boldsymbol{\lambda}}_d \\ \mu \\ \alpha^\diamond \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_p \\ -\underline{\boldsymbol{b}}_d \\ h \\ -e^\diamond \end{pmatrix}
\tag{132}
$$

with

$$
\mathcal{C}_p = \boldsymbol{C}_p - \mathbb{C}_{\bar{p}} K_{\bar{p}\bar{p}}^{\diamond+} K_{\bar{p}p}^\diamond \mathbb{A}_p^T
\tag{133}
$$

$$
\mathcal{C}_d = \mathbb{C}_{\bar{p}} K_{\bar{p}\bar{p}}^{\diamond+} t_d^{\diamond T} \underline{\mathbb{A}}_d^T
\tag{134}
$$

$$
\mathcal{C}_\mu = \mathbb{C}_{\bar{p}} K_{\bar{p}\bar{p}}^{\diamond+} \mathbb{C}_{\bar{p}}^T
\tag{135}
$$

$$
\boldsymbol{G}_\alpha = \mathbb{C}_{\bar{p}} R_{\bar{p}}
\tag{136}
$$

$$
h = a - \mathbb{C}_{\bar{p}} K_{\bar{p}\bar{p}}^{\diamond+} f_{\bar{p}}^\diamond
\tag{137}
$$

Various strategies can be used in order to solve system (132), which combine elimination of constraints (rigid body motions and/or MPCs) by projection methods (FETI-like approaches) and/or by recondensation methods (FETIDP like approaches). All these methods correspond to solving rather complex coarse problems but they are suited to traditional preconditioners. We propose, after [93], to use classical projection method to handle rigid body motions then use iterative solver to find simultaneously $(\boldsymbol{u}_p, \underline{\boldsymbol{\lambda}}_d, \mu)$ and provide efficient preconditioner to this problem.

System reads with trivial notations (for simplicity reasons we suppose that the rigid body motion constraints have been symmetrized, which is always possible and which is naturally the case if $G_p = 0$ like in many applications):

$$\begin{pmatrix} \boldsymbol{S}_{pd\mu} & \mathcal{G} \\ \mathcal{G}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \alpha^\diamond \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ -e^\diamond \end{pmatrix} \tag{138}$$

with

$$\boldsymbol{S}_{pd\mu} = \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \underline{\mathbb{A}}_d t_d^\diamond \\ 0 & \mathbb{C}_{\bar{p}} \end{pmatrix} S_{pd}^\diamond \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \underline{\mathbb{A}}_d t_d^\diamond \\ 0 & \mathbb{C}_{\bar{p}} \end{pmatrix}^T + \begin{pmatrix} 0 & 0 & \boldsymbol{C}_p \\ 0 & 0 & 0 \\ -\boldsymbol{C}_p^T & 0 & 0 \end{pmatrix} \tag{139}$$

As can be seen, MPCs have very different actions wether they are set on primal interface degrees of freedom or not: matrix C_p modifies the structure of the hybrid system while dual and internal constraints lead to classical hybrid approach with modified dual trace and assembly operator $\mathcal{A} = \begin{pmatrix} \underline{\mathbb{A}}_d t_d^\diamond \\ \mathbb{C}_{\bar{p}} \end{pmatrix}$. The definition of efficient preconditioner inspired by classical methods is then much simplified if no constraints are set on primal degrees of freedom ($C_p = 0$), which is what we suppose now:

$$\boldsymbol{S}_{pd\mu} = \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \mathcal{A} \end{pmatrix} \mathcal{S}_{pd}^\diamond \begin{pmatrix} \mathbb{A}_p & 0 \\ 0 & \mathcal{A} \end{pmatrix}^T \tag{140}$$

The $\mathcal{S}_{pd}^\diamond$ notation is due to the association of the trace operator with the assembly operator, which in fact is equivalent to defining "extended interface dual degrees of freedom" made out of dual degrees of freedom and degrees of freedom involved in MPCs. Since, in this hypothesis, system reads like classical hybrid approach, that is (modified) assembly of local contributions, the proposed preconditioner is a scaled assembly of local inverses.

$$\tilde{\boldsymbol{S}}_{pd\mu}^{-1} = \begin{pmatrix} \tilde{\mathbb{A}}_p & 0 \\ 0 & \tilde{\mathcal{A}} \end{pmatrix} \mathcal{S}_{dp}^\diamond \begin{pmatrix} \tilde{\mathbb{A}}_p & 0 \\ 0 & \tilde{\mathcal{A}} \end{pmatrix}^T \tag{141}$$

The primal scaled assembly operator can be directly imported from the primal approach. Concerning the dual approach, according to previous definitions, we have:

$$\tilde{\mathcal{A}} = \left(\mathcal{A} M_{\bar{p}}^{\diamond-1} \mathcal{A}^T \right)^+ \mathcal{A}^\diamond M_{\bar{p}}^{\diamond-1} \tag{142}$$

Where $M_{\bar{p}}^\diamond$ is a diagonal matrix chosen like in the classical methods. The matrix to inverse reads:

$$\left(\mathcal{A} M_{\bar{p}}^{\diamond-1} \mathcal{A}^T \right) = \begin{pmatrix} \underline{\mathbb{A}}_d t_d^\diamond M_{\bar{p}}^{\diamond-1} t_d^{\diamond T} \underline{\mathbb{A}}_d^T & \underline{\mathbb{A}}_d t_d^\diamond M_{\bar{p}}^{\diamond-1} \mathbb{C}_{\bar{p}}^T \\ \mathbb{C}_{\bar{p}} M_{\bar{p}}^{\diamond-1} t_d^{\diamond T} \underline{\mathbb{A}}_d^T & \mathbb{C}_{\bar{p}} M_{\bar{p}}^{\diamond-1} \mathbb{C}_{\bar{p}}^T \end{pmatrix} \tag{143}$$

The idea is then to make this system easy to inverse, having the off-diagonal blocks equal to zero. We have $\mathbb{C}_{\bar{p}} = \begin{pmatrix} \mathbb{C}_i & \mathbb{C}_d \end{pmatrix}$ with $\mathbb{C}_d \underline{\mathbb{A}}_d^T = C_d$; if we choose

$$\mathbb{C}_d = C_d \left(\underline{\mathbb{A}}_d t_d^\diamond M_{\bar{p}}^\diamond t_d^{\diamond T} \underline{\mathbb{A}}_d^T \right)^{-1} \underline{\mathbb{A}}_d t_d^\diamond M_{\bar{p}}^\diamond t_d^{\diamond T} \tag{144}$$

then $\underline{\mathbb{A}}_d t_d^\diamond M_{\bar{p}}^{\diamond-1} \mathbb{C}_{\bar{p}}^T = \underline{0}$ and the scaled assembly operator is non expensive to compute.

In others words, one simply has to split constraints on interface degrees of freedom between subdomains according to the scaling used inside the preconditioner.

5.4 Choice of decomposition

Decomposing a given structure in order to use the algorithms presented in this paper is a complex problem. Algorithms and softwares were proposed [44, 12, 62], which mostly refer to graph theory. Such approaches enable to take into account load balance between processors (supposing each processor is assigned to one subdomain, this is equivalent to making each local problem as easy to solve as others) and to minimize the dimension of the interface (so that the global condensed problem is as small as possible). They also enable to avoid internal rigid body motions (so called mechanisms).

Anyhow another important point is to have local problems as well conditioned as possible, so having subdomains with good aspect ratio (ratio between largest and smallest characteristical dimensions of the subdomain) is considered as an important point. Anyhow one has to realize that good aspect ratio is often linked to large local bandwidth and then to somehow expensive local problems to solve.

An even more difficult point to take into account high heterogeneities (see figure 13): using stiffness scaling enables to correctly handle heterogeneity when interface between subdomains matches interface between materials, anyhow when interface between subdomains "crosses" interface between materials then numerical difficulties may occur. For now scaled-average optional constraints seem to be the best solution to handle these difficulties but it leads to large coarse problems.
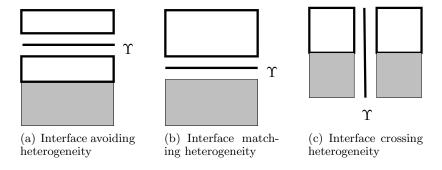

(a) Interface avoiding heterogeneity   (b) Interface matching heterogeneity   (c) Interface crossing heterogeneity

Figure 13. Different kinds of heterogeneity in domain decomposition context

Finally finding the best decomposition is still a rather open problem and mechanical sense is often a necessary complement to efficient automatic decomposing algorithm.

5.5 Extensions

5.5.1 Nonsymmetric problems

Nonsymmetry occurs in many physical modeling: plasticity, nonlocal models for fracture [52], frictional contact [3]. The use of the domain decomposition methods presented in this paper just requires more care in the implementation because some simplifications are not available (for instance coarse problem matrices are nonsymmetric), and of course the use of well suited iterative solver like GMRes, OrthoDir or BiCG because Schur complements are no longer symmetric.

Globally methods show good numerical performance results. Anyhow a real problem is the absence of theoretical results to ensure good convergence properties (this is mainly due to the fact that proofs for classical methods rely on the construction of an interface inner-product related to Schur complements which is no longer possible).

5.5.2 Nonlinear problems

For now we have considered the solution to linear systems. To adapt the method to nonlinear problems, a classical solution is to use Newton-Raphson linearization scheme: linearized stiffness matrices are computed independently on each subdomain then the linearized system is solved using domain decomposition [85, 58, 88, 40, 59]. For such approaches, domain decomposition methods can be seen as efficient black-boxed linear solvers.

One critical point in such a method is that, depending on the formulation (for instance fully Lagragian formulation of a large deformation elasticity problem) rigid body motions may vary from one system to the other. In the proposed context, what has been proved is that translations always belong to zero energy modes, what has been observed is that rotations only appeared as zero-energy-modes in the first system (which corresponds to linear elasticity problem). This might be penalizing because the size of the information transmitted inside the coarse problem is decreased after the first system; moreover, rotations are often converted to "negative-energy modes" which, if they are in small number, can be handled by fully-reorthogonalized conjugate gradient (though convergence will be slower). One classical solution is to reinject disappeared rotations as optional constraints (via augmentation algorithms).

In the case where nonlinearity is localized in few substructures, an interesting strategy can be to carry out subiterations of the nonlinear solver independently in those substructures [13, 18].

5.6 Implementation issues

Implementing domain decomposition methods from existing code is not a too complex task. We give a few details on our software architecture though practical solutions are far from being unique. Our code is a plug-in to ZeBuLoN object-oriented finite element computational software [79, 78], it takes advantage of Frederic Feyel's previous work [45, 46]. Our implementation aims at being as generic as possible, so for now hybrid domain decomposition method has been developed (and mixed approaches are under construction), and separation between formulation and solver (so that any iterative solver can be used to solve the interface problem). All classical projectors and preconditioners are implemented.

The most basic pieces of the code are:

- topological description of the interface, *ie* ability to realize trace operations;

- exchange library (PVM, MPI), *ie* ability to realize assembly operations;

- classical FE code, *ie* ability on any given subdomain to get any local field given sufficient boundary conditions.

5.6.1 Organization of the topological information

In order to implement hybrid approach, we propose to define a specific class for "interface degrees of freedom" which wraps classical degrees of freedom and provide information on:

- the number of subdomains which share this degree of freedom

- the kind of treatment (primal, dual...) which is applied

Then degrees of freedom are set together neighborwise, defining "interface" class made out of a list of pointer to "interface degrees of freedom", and the global number of the neighbor (that number enables to identify the subdomain and to realize exchanges).

A "subdomain" is defined as a collection of "interfaces" and a classical domain in the sense of usual FE software (mainly mesh), it possesses its own global identification number.
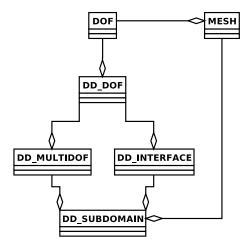
Figure 14. Topological interface information

Note that with such a description of the decomposition, the local interfaces are redundant, multiple degrees of freedom appear in several interfaces. It is then necessary to take certain care to define some operations (transposed trace for dual degrees of freedom). A specific class can then be used to ease the management of multiple degrees of freedom.

5.6.2 Defining algebraic interface objects (fig. 15)

In order to easily connect the domain decomposition formulation to an iterative solver, we propose to define "interface vectors" (displacement, intereffort...), "interface matrices" (trace of rigid body motions..., can be seen as a collection of interface vectors) and "interface operators" (square interface matrices), with all classical operations (basically sum, difference, product and transposed product).

The particularity of these objects is to be defined on the interface and then data is shared between subdomains, so all the previous operations sometimes require to assemble data (an interesting idea is to have a boolean member indicating the assembled state of data). Because of the choice of description of the interface, the assembly operation requires certain care for primal multiple degrees of freedom (these degrees of freedom shall have the same value at their different occurrences). Note that an object like "interface matrix" can highly be optimized (mainly in terms of memory storage).

Interface operators mainly know how to multiple vectors and matrices, they are used to define Schur complements, scaling operators, projectors. Composite design pattern can be used to simplify the succession of operations.

In order to let the user choose the various configurations of the domain decomposition method, we use inheritance and object factory design pattern.

5.6.3 Articulation between formulation and solver (fig. 16)

What we propose is to have client/server relationship between solver and formulation: basically an iterative solver needs to know how to initialize, how to multiply, how to precondition, how to do inner products, how to evaluate convergence. All these operations are implemented inside the "interface formulation" object which is linked to one subdomain (topology and stiffness) and creates "interface algebraic objects" in order to define required operations.
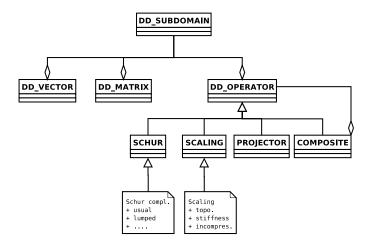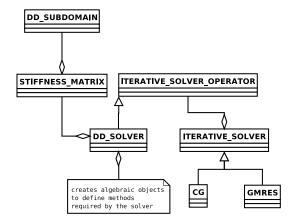
Figure 15. Algebraic interface objects

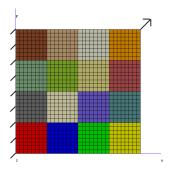**Figure 16.** Articulation between formulation and solver

Figure 17. 16 subdomains decomposed square

6 Assessments

The assessments we propose here are based on the code described in the previous section. Basically, we have implemented the hybrid approach which lets us assess the classical primal and dual approaches with most classical preconditioners and projectors.

We first present a sequence of academical tests in order to recover classical numerical performance results (scalability and relative efficiency of the different approaches): two dimensional plane stress problem, plate problem, three dimensional problem with heterogeneity and unstructured decomposition. In these problems, H denotes the characteristic size of the subdomains and h the characteristic size of the elements. We also present results on non-academic problem (bitraction test specimen).

For all those tests, in order to compare all the approaches (including the hybrid approach), GMRes solver is used and convergence is monitored by the norm of the residual as given by the solver (with ε set to 10^{-6}). In other cases (when hybrid approach is not assessed), the convergence is monitored through the evaluation of global primal residual $|Ku - f|/|f| < \varepsilon$ with ε set to 10^{-6}. Depending on the method, a different coarse problem may be introduced, we denote by CS:a+b the size of the coarse problems (a for the admissibility coarse problem and b for the optimal preconditioning coarse problem) or $number\_of\_iterations_{\text{total\_number\_of\_constraints}}$. Note that the hybrid approach deals with two independent coarse problems, so their solutions is much cheaper than the solution to a unique large coarse problem.

We test the primal approach with or without optimality coarse problem, the dual approach with lumped or Dirichlet preconditioner and identity or superlumped or Dirichlet projector (denoted by P(I), P(W) and P(D) respectively). As for such tests no physical consideration can guide the choice of hybrid treatment to the interface, in order to show the potential of the hybrid approach, we present results where all degrees of freedom of one direction (U_1, U_2 or U_3) are treated in the same way. For instance "D-P" stands for a dual treatment for degrees of freedom associated to direction U_1, and a primal treatment for degrees of freedom associated to direction U_2.

6.1 Two dimensional plane stress problem

We consider a simple second order two-dimensional problem, the structure is an homogeneous square decomposed in square substructures meshed with linear square finite element (Q1 Lagrange). The behavior is linear elastic (Young modulus $E = 200000$ MPa and Poisson coefficient $\nu = 0.3$), the loading consists in clamping on the left side and punctual effort on the top right corner (figure 17).

Method \ H/h		8	16	32	64
Primal	SC:0+0	44	45	45	45
	SC:0+36	11	12	14	15
Dual SC:36+0	Lumped - P(I)	14	25	32	42
	Dirichlet - P(I)	13	15	17	20
	Dirichlet - P(D)	12	14	15	17
Hybrid D-P	SC:12+0 - P(D)	29	30	33	35
	SC:12+12 - P(D)	12	14	16	18
Hybrid P-D	SC:12+0 - P(I)	29	32	36	38
	SC:12+12 - P(I)	14	17	20	22
	SC:12+0 - P(D)	26	29	31	33
	SC:12+12 - P(D)	12	14	16	18

Table 1. Scalability results in 2D / 16 subdomains

Method \ nb. subdomains		4	9	16	25	36	49	64
Primal	No opt. coarse	13_0	29_0	45_0	63_0	83_0	102_0	126_0
(Neumann[2])	With opt. coarse	8_6	10_{18}	12_{36}	13_{60}	14_{90}	14_{126}	15_{168}
Dual	Lumped - P(I)	18_6	24_{18}	26_{36}	27_{60}	29_{90}	29_{126}	31_{168}
	Dirichlet - P(I)	9_6	13_{18}	15_{36}	16_{60}	17_{90}	18_{126}	19_{168}
	Dirichlet - P(D)	9_6	12_{18}	14_{36}	15_{60}	16_{90}	17_{126}	18_{168}
Hybrid D-P	No opt. coarse	9_2	21_6	30_{12}	40_{20}	50_{30}	60_{42}	67_{56}
P(D)	With opt. coarse	7_4	12_{12}	14_{24}	16_{40}	17_{60}	18_{84}	18_{112}
Hybrid P-D	No opt. coarse	9_2	20_6	29_{12}	38_{20}	48_{30}	57_{42}	57_{56}
P(D)	With opt. coarse	7_4	12_{12}	14_{24}	16_{40}	17_{60}	18_{84}	17_{112}

Table 2. Performance results in 2D for given $\frac{H}{h} = 16$

Table 1 shows the number of iterations of available strategies for different number of elements per subdomain (for a 16-subdomain decomposition), and table 2 for different number of subdomains (for given ratio $H/h = 16$). Globally all approaches (primal, dual and hybrid) equipped with their best preconditioner and projector behave similarly and are scalable. Note that even in its optimal configuration the hybrid approach requires a smaller coarse space (for instance, table 1, two 12×12 coarse problems against one 36×36 coarse problem for primal or dual approaches) for equivalent efficiency. As expected if the optimality coarse problem is suppressed, performance results decay and scalability is lost. Finally for such simple problems, the simplified versions of the dual approach gives excellent results.

6.2 Bending plate

We consider a forth order plate problem, the structure is an homogeneous square decomposed in squared substructures meshed with square Mindlin plate element. The behavior is linear elastic (Young modulus $E = 200000$ MPa and Poisson coefficient $\nu = 0.3$), the loading consists in clamping on the left side and punctual normal effort on the top right corner.

Table 3 presents the number of iterations for the dual and primal approaches, with or without optional corner constraints (the subscript indicates the total size of coarse problems, *ie* rigid body motions and corner modes). As predicted, corner constraints are essential in

nb. subdomains / Method		4	9	16	25	36	49
Primal (Neumann[2])	No corner	15_{12}	24_{36}	32_{72}	40_{120}	47_{180}	55_{252}
	With corners	13_{16}	17_{64}	20_{108}	23_{184}	24_{280}	26_{396}
Dual (Dirichlet)	P(I) - No corner	17_{12}	31_{36}	43_{72}	59_{120}	75_{180}	91_{252}
	P(I) - With corners	16_{15}	24_{48}	27_{99}	29_{168}	31_{255}	33_{360}
	P(D) - No corner	15_{12}	25_{36}	34_{72}	43_{120}	51_{180}	59_{252}
	P(D) - With corners	14_{15}	21_{48}	28_{99}	31_{168}	31_{255}	36_{360}

Table 3. Bending plate: performance results for given $\frac{H}{h} = 8$

order to make the algorithms scalable. Anyhow the dimension of the coarse space associated to corners quickly explodes which makes the methods less interesting from a CPU time point of view, which justifies the FETIDP philosophy which leads to much smaller coarse spaces.

6.3 Heterogeneous 3D problem

We consider a 3D problem, the structure is an heterogeneous cube decomposed in $3 \times 3 \times 3$ cubic substructures meshed with $3 \times 3 \times 3$ Q2-Lagrange cubic elements (27 nodes per element). The heterogeneity pattern is described in figure 18a, behaviors are linear elastic (Young modulus $E_1 = 200000$ MPa, $E_2 = 2$ MPa and Poisson coefficient $\nu = 0.3$), the loading consists in clamping on the bottom side and constant pressure on top side.
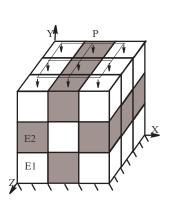
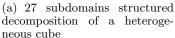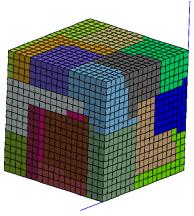| Method		Number of iterations
Primal		19
Dual P(D)	No splitting	28
	Classical splitting	28
	Condensed splitting	18
Dual P(W)	No splitting	21
	Classical splitting	21
	Condensed splitting	20
Dual P(I)	No splitting	74
	Classical splitting	74
	Condensed splitting	73

Table 4. Heterogeneous cube

Table 4 presents the number of iterations for the conjugate gradient to converge. Assessed methods are classical primal approach and dual approach with different projectors for all splittings (or equivalent initializations) presented in section 3.2.4, of course stiffness scaling is employed. What appears clearly is the good behavior of the approaches face to heterogeneity except the identity projector of the dual approach (which is definitely not suited to heterogeneous problems), and the efficiency of the condensed initialization. For such a problem the superlumped projector leads to very good results, anyhow for more complex cases Dirichlet projector is necessary and shall be improved at no extra computational cost by the condensed initialization.

6.4 Homogeneous non-structured 3D problem

We consider a 3D problem, the structure is an homogeneous cube meshed with Q1-Lagrange cubic elements (8 nodes per element). The behavior is linear elastic (Young modulus $E =$

(a) 27 subdomains structured decomposition of a heterogeneous cube

(b) 27 subdomains unstructured decomposition of a cube

Figure 18. 3D assessments

200000 MPa and Poisson coefficient $\nu = 0.3$), the loading consists in clamping on the bottom side and constant pressure on top side. We consider two kinds of decomposition: either structured decompositions ($3 \times 3 \times 3$ or $4 \times 4 \times 4$ cubic substructures) or so called "unstructured" decompositions realized by Metis software (http://www-users.cs.umn.edu/~karypis/metis/), see figure 18b.

Method ╲ Decomposition	Structured		Unstructured	
	27 sd.	64 sd.	27 sd.	64 sd.
Primal Neumann-Neumann	11_{108}	14_{288}	42	67
Dual Dirichlet P(I)	12_{108}	16_{288}	44	69
Dual Dirichlet P(D)	12_{108}	16_{288}	43	70
Hybrid D-D-P P(I)	14_{72}	17_{192}	-	-
Hybrid P-P-D P(I)	15_{72}	19_{192}	-	-
Hybrid D-P-D P(I)	13_{72}	17_{192}	-	-

Table 5. Homogeneous cube / influence of the decomposition

Table 5 enables to highlight the fundamental role played by the decomposition: scalabily result only holds for structured decomposition; moreover there might by a huge performance gap between two decompositions with the same number of subdomains (factor 3 for 27 subdomains, factor 4 for 64 subdomains).

6.5 Bitraction test specimen

In order to assess "real life" problems, we consider the bitraction test specimen presented in figure 2 (this structure, courtesy of ONERA – Pascale Kanouté –, was optimized with ZeBuLoN software in order to have stress field as homogeneous as possible in its center). It was decomposed with Metis software into 4 or 16 subdomains.

As shown in table 6 all methods give excellent performance results on non-academical problem. Note the good behavior of the hybrid approach which gives equivalent performance with much smaller coarse problem, even if no physical consideration could guide the choice

Method		4 sd.	16 sd.
Primal Neumann-Neumann		23_{0+12}	30_{0+69}
Dual	Lumped P(I)	32_{12}	41_{69}
	Dirichlet P(I)	25_{12}	32_{69}
	Dirichlet P(Q)	24_{12}	32_{69}
Hybrid P(Q)	P-P-D	31_8	44_{46}
	D-D-P	25_8	37_{46}

Table 6. Bitraction test specimen

of the treatment of interface degrees of freedom.

7 Conclusions

In this paper, we have reviewed most used non-overlapping domain decomposition methods. These methods are perfectly suited to modern computational hardware, they are based on very close concepts which we tried to outline. We introduced the hybrid framework to include as many methods as possible: the principle is to assign to each interface degree of freedom its own treatment, for now primal and dual treatments have been implemented, and mixed and recondensed should follow. Hybriding methods also enable to define physic-friendly approaches for multifield problems.

Because of the conceptual proximity of all methods, assessments showed very close numerical performance results. Once equipped with convenient preconditioner and coarse problem, all the methods proved their ability to handle second and forth order elasticity in presence of strong heterogeneities. Though from a computational point of view some combination may be more interesting: dual approach with lumped preconditioner or simplified projector (if these are sufficient to ensure fine rate of convergence), hybrid approach (which generates smaller coarse space). We also outlined the importance of the decomposition even for very simple problems. The methods have also proved their efficiency on industrial cases, some of them were implemented in computational softwares.

In this paper we limited to the solution to linearized systems, which anyhow enables to solve nonlinear problems. Another strategy is to commute the nonlinear solver and the domain decomposition method so that nonlinear problems can be solved independently on each subdomain. Another evolution of domain decomposition philosophy is the decomposition of the time interval [72] for evolution problems.

REFERENCES

1 Yves Achdou and Yuri A. Kuznetsov. Substructuring preconditioners for finite element methods on nonmatching grids. *East-West J. Numer. Math.*, 3(1):1–28, 1995.

2 Yves Achdou, Yvon Maday, and Olof B. Widlund. Méthode itérative de sous-structuration pour les éléments avec joints. *C.R. Acad. Sci. Paris*, I(322):185–190, 1996.

3 Mickael Barboteu, Pierre Alart, and Marina Vidrascu. A domain decomposition strategy for nonclassical frictional multicontact problems. *Comp. Meth. App. Mech. Eng.*, 190:4785–4803, 2001.

4 Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, 1994.

5 T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, , and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

6 C. Bernardi, Y. Maday, and T. Patera. A new non conforming approach to domain decomposition: the Mortar Element Method. In H. Brezis and J.L. Lions, editors, *Nonlinear Partial Differential Equations and their Applications.* Pitman, London, 1989.

7 M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems: Scalability results on a thousand-processor and discussion of highly heterogeneous problems. *Int. J. Num. Meth . Eng.*, 47(1-3):513–536, 2000.

8 J. E. Bolander and N. Sukumar. Irregular lattice model for quasistatic crack propagation. *Phys. Rev. B*, 71, 2005.

9 Piotr Breitkopt and Antonio Huerta, editors. *Meshfree and particle based approaches in computational Mechanics*, volume 11 of *REEF special release.* Hermes, 2002.

10 Suzanne Brenner. Lower bounds in domain decomposition. In *Proceedings of the 16th international conference on domain decomposition methods*, 2005.

11 F. Brezzi and L.D. Marini. A three-field domain decomposition method. In *Proceedings of the sixth international conference on domain decomposition methods*, pages 27–34, 1993.

12 M. Cross C. Walshaw and M. G. Everett. A parallelisable algorithm for optimising unstructured mesh partitions. Technical report, School of Mathematics, Statistics & Scientific Computing, University of Greenwich, London, 1995.

13 X-C Cai and David Keyes. Nonlinearly preconditioned inexact newton algorithms. *SIAM J. Sci. Comp.*, 24:183–200, 2002.

14 A. Chapman and Y. Saad. Deflated and augmented Krylov subspace techniques. *Numer. Linear Algebra Appl.*, 1997.

15 P.G. Ciarlet. *The finite element method for elliptic problems.* North Holland, 1979.

16 J. Salençon. *Mécanique des milieux continus.* Ecole polytechnique. Ellipses, 1988.

17 R. Craig and M. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6:1313–1319, 1968.

18 Philippe Cresta, Olivier Allix, Christian Rey, and Stéphane Guinard. Comparison of multiscale and parallel nonlinear strategies based on domain decomposition for post buckling analysis. *Comp. Meth. App. Mech. Eng.*, 196(8):1436–1446, 2005.

19 Jean-Michel Cros. A preconditioner for the schur complement domain decomposition method. In Herrera, Keyes, and Widlund, editors, *Proceedings of the 14th international conference on domain decomposition methods*, pages 373–380, 2002.

20 G.A. D'Addetta, E. Ramm, S. Diebels, and W. Ehlers. A particle center based homogenization strategy for granular assemblies. *Int J for Computer-Aided Engineering*, 21(2-4):360–383, 2004.

21 A. de La Bourdonnaye, C. Farhat, A. Macedo, F. Magoules, and F.-X. Roux. *Advances in Computational Mechanics with High Performance Computing*, chapter A method of finite element tearing and interconnecting for the Helmholtz problem, pages 41–54. Civil-Comp Press, Edinburgh, United Kingdom, 1998.

22 Arnaud Delaplace. Fine description of fracture by using discrete particle model. In *Proceedings of ICF 11 - 11th International Conference on Fracture*, 2005.

23 Clarck Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comp.*, 25(1):246–258, 2003.

24 Victorita Dolean, Frederic Nataf, and Gerd Rapin. New constructions of domain decomposition methods for systems of pdes. *C. R. Acad. Sci. Paris*, 340(1):693–696, 2005.

25 Z. Dostal. Conjugate gradient method with preconditioning by projector. *Int. J. Comput. Math.*, 23:315–323, 1988.

26 Zdenek Dostal, David Horak, and Dan Stefanica. An overview of scalable feti-dp algorithms for variational inequalities. In *Proceedings of the 16th conference on domain decomposition methods*, 2005.

27 D. Dureisseix and C. Farhat. A numerically scalable domain decomposition method for the solution of frictionless contact problems. *Internat. J. Num. Meth. Engin.*, 50(12):2643–2666, 2001.

28 Georges Duvaut. *Mécanique des milieux continus*. Masson, 1990.

29 J. Erhel, K. Burrage, and B. Pohl. Restarted GMRes preconditioned by deflation. *J. Comput. Appl. Math.*, 69:303–318, 1996.

30 Justine Erhel and F. Guyomarc'h. An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1279–1299, 2000.

31 C. Farhat. A saddle-point principle domain decomposition method for the solution of solid mechanics problems. In D. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs, and R.G. Voigt, editors, *Domain Decomposition Methods for Partial Differential Equations*, pages 271–292, 1992.

32 C. Farhat, P.-S. Chen, and F.-X. Roux. The two-level FETI method - part II: Extension to shell problems. parallel implementation and performance results. *J. Comp Meth. Appl. Mech. Eng.*, 155:153–180, 1998.

33 C. Farhat and M. Géradin. On the computation of the null space and generalized invers of large matrix, and the zero energy modes of a structure. Technical Report CU-CAS-96-15, Center for aerospace structures, may 1996.

34 C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - part i: a faster alternative to the two-level FETI method. *Int. J. Num. Meth . Eng.*, 50(7):1523–1544, 2001.

35 C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7(7-8):687–714, 2000.

36 C. Farhat, A. Macedo, and M. Lesoinne. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numerische Mathematik*, 85:283–308, 2000.

37 C. Farhat, A. Macedo, and R. Tezaur. FETI-H: a scalable domaine decomposition method for high frequency exterior Helmholtz problems. In *Domain decomposition methods in science and engineering*. Domain decomposition press, 1999.

38 C. Farhat and J. Mandel. The two-level FETI method for static and dynamic plate problems - part I: An optimal iterative solver for biharmonic systems. *J. Comp Meth. Appl. Mech. Eng.*, 155:129–152, 1998.

39 C. Farhat, J. Mandel, and F.X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comp. Meth. Appl. Mech. Eng.*, 115:365–385, 1994.

40 C. Farhat, K. Pierson, and 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. *J. Comp Meth. Appl. Mech. Eng.*, 184(2-4):333–374, 2000.

41 C. Farhat and F.-X. Roux. A method of finite tearing and interconnecting and its parallel solution algorithm. *Int. J. Num. Meth . Eng.*, 32:1205–1227, 1991.

42 C. Farhat and F.-X. Roux. The dual Schur complement method with well-posed local Neumann problems. *Contemporary Mathematics*, 157:193–201, 1994.

43 C. Farhat and F. X. Roux. Implicit parallel processing in structural mechanics. *Computational Mechanics Advances*, 2(1):1–124, 1994. North-Holland.

44 C. Farhat and H. D. Simon. Top/domdec - a software tool for mesh partitioning and parallel processing. Technical report, NASA Ames, 1993.

45 Frédéric Feyel. *Application du calcul parallèle aux modèles à grand nombre de variables internes.* Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris, 1998.

46 Frédéric Feyel. *Quelques multi-problèmes en mécanique des matériaux et structures.* Habilitation à diriger des recherches, Université Pierre et Marie Curie, 2005.

47 Michel Fortin and Roland Glowinski. *Méthodes de lagrangien augmenté - applications à la résolution numérique de problèmes aux limites.* Dunod, 1982.

48 Yannis Fragakis and Manolis Papadrakakis. A unified framework for formulating domain decomposition methods in structural mechanics. Technical report, Institute of Structural Analysis and Seismic Research, National Technical University of Athens, 2002.

49 Yannis Fragakis and Manolis Papadrakakis. The mosaic of high-performance domain decomposition methods for structural mechanics – part i: Formulation, interrelation and numerical efficiency of primal and dual methods. *Comp. Meth. Appl. Mec. Eng.*, 192(35-36):3799–3830, 2003.

50 Yannis Fragakis and Manolis Papadrakakis. The mosaic of high-performance domain decomposition methods for structural mechanics – part ii: Formulation enhancements, multiple right-hand sides and implicit dynamics. *Comp. Meth. Appl. Mec. Eng.*, 193(42-44):4611–4662, 2004.

51 V. Frayssé, L. Giraud, and H. Kharraz-Aroussi. On the influence of the orthogonalization scheme on the parallel performance of GMRes. Technical report, CERFACS, 1998.

52 Norbert Germain, Jacques Besson, and Frédéric Feyel. Méthodes de calcul non local : Application aux structures composites. In *Actes du 7ème colloque national en calcul des structures*, Giens, 2005.

53 P. Germain. *Mécanique.* Ecole polytechnique. Ellipses, 1986.

54 R. Glowinski and P. Le Tallec. Augmented lagrangian interpretation of the nonoverlapping Schwartz alternating method. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–231, 1990.

55 P. Goldfeld. Balancing Neumann-Neumann for (in)compressible linear elasticity and (generalized) Stokes – parallel implementation. In *Proceedings of the 14th international conference on domain decomposition method*, pages 209–216, 2002.

56 P. Gosselet, V. Chiaruttini, C. Rey, and F. Feyel. Une approche hybride de décomposition de domaine pour les problèmes multiphysiques : application à la poroélasticité. In *Actes du sixième colloque national en calcul de structures*, volume 2, pages 297–304, 2003.

57 P. Gosselet, Vincent Chiaruttini, C. Rey, and F. Feyel. A monolithic strategy based on an hybrid domain decomposition method for multiphysic problems. application to poroelasticity. *Revue européenne des élements finis*, 13:523–534, 2004.

58 P. Gosselet and C. Rey. On a selective reuse of krylov subspaces in newton-krylov approaches for nonlinear elasticity. In *Proceedings of the 14th conference on domain decomposition methods*, pages 419–426, 2002.

59 P. Gosselet, C. Rey, P. Dasset, and F. Léné. A domain decomposition method for quasi incompressible formulations with discontinuous pressure field. *Revue européenne des élements finis*, 11:363–377, 2002.

60 P. Gosselet, C. Rey, and D. Rixen. On the initial estimate of interface forces in FETI methods. *Comp. meth. appl. mech. engrg.*, 192:2749–2764, 2003.

61 Pierre Gosselet. *Méthodes de décomposition de domaine et méthodes d'accélération pour les problèmes multichamp en mécanique non-linéaire.* PhD thesis, Université P. et M. Curie, 2003.

62 George Karypis and Vipin Kumar. Multilevel algorithms for multi-constraint graph partitioning. Technical report, University of Minnesota, Department of Computer Science, 1998.

63 A. Klawonn, O. Rheinbach, and O.B. Widlund. Some computational results for dual-primal feti methods for three dimensional elliptic problems. *Lect. Notes Comput. Sci. Eng.*, 40:361 – 368, 2005.

64 A. Klawonn and O.B. Widlund. Dual and dual-primal FETI methods for elliptic problems with discontinuous coefficients. In *Proceedings of the 12th International Conference on Domain Decomposition Methods, Chiba, Japan*, October 1999.

65 A. Klawonn and O.B. Widlund. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. *Comm. pure and appl. math.*, LIV:0057–0090, 2001.

66 P. Ladevèze. *Nonlinear Computational Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation.* Springer Verlag, 1999.

67 P. Ladevèze, O. Loiseau, and D. Dureisseix. A micro-macro and parallel computational strategy for highly heterogeneous structures. *Int. J. Num. Meth. Engnrg.*, 52:121–138, 2001.

68 P. Ladevèze, D. Néron, and P. Gosselet. On a mixed and multiscale domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1526–1540, 2006.

69 Michel Lesoinne and Kendall Pierson. FETI-DP: An efficient, scalable, and unified Dual-Primal FETI method. In *Domain Decomposition Methods in Sciences and Engineering*, pages 421–428, 1999.

70 Jing Li. A dual-primal feti method for solving stokes/navier-stokes equations. In *Proceedings of the 14th international conference on domain decomposition method*, pages 225–233, 2002.

71 F.J. Lingen. Efficient Gram-Scmidt orthonormalisation on parallel computers. *Com. Numer. Meth. Engng.*, 16:57–66, 2000.

72 JL. Lions, Y. Maday, and G. Turinici. Résolution d'edp par un schéma en temps pararéel. *C. R. Acad. Sci. Paris*, 333(1):1–6, 2001.

73 Gui-Rong Liu and Yuan-Tong Gu. *An Introduction to Meshfree Methods and Their Programming.* Springer, 2005.

74 J. Mandel. Balancing domain decomposition. *Comm. Appl. Num. Meth. Engrg.*, 9:233–241, 1993.

75 J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65(216):1387–1401, 1996.

76 J. Mandel and R. Tezaur. Convergence of a substructuring method with Lagrange multipliers. *Numerische Mathematik*, 73:473–487, 1996.

77 J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. UCD/CCM Report 150, Center for Computational Mathematics, University of Colorado at Denver, April 2000. to appear in Numer. Math.

78 Northwest Numerics. *Z-set developper manual*, 2001.

79 Northwest Numerics. *Z-set user manual*, 2001.

80 A. Nouy. *Une stratégie de calcul multiéchelle avec homogénéisation en temps et en espace pour le calcul de structures fortement hétérogènes*. PhD thesis, ENS de Cachan, 2003.

81 K.C. Park, M.R. Justino, and C.A. Felippa. An algebraically partitioned FETI method for parallel structural analysis: algorithm description. *Int. J. Num. Meth. Eng.*, 40(15):2717–2737, 1997.

82 K.C. Park, M.R. Justino, and C.A. Felippa. An algebraically partitioned FETI method for parallel structural analysis: performance evaluation. *Int. J. Num. Meth . Eng.*, 40(15):2739–2758, 1997.

83 Rodrigo Paz and Mario Storti. An interface strip preconditioner for domain decomposition methods: application to hydrology. *Int. J. Numer. Meth. Engng.*, 62:1873–1894, 2005.

84 C. Rey and P. Gosselet. Solution to large nonlinear systems: acceleration strategies based on domain decomposition and reuse of krylov subspaces. In *Proceedings of the $6^t h$ ESAFORM conference on material forming*, 2003.

85 C. Rey and F. Léné. Reuse of krylov spaces in the solution of large-scale non linear elasticity problems. In *Domain Decomposition Methods in Sciences and Engineering*, pages 465–471, 1998.

86 C. Rey and F. Risler. A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems. *Numerical Algorithms*, 17:279–311, 1998.

87 Franck Risler and Christian Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numerical Algorithms*, 23:1–30, 2000.

88 Frank Risler and Christian Rey. On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods. In *DD10 Proceedings, Contemporary Mathematics*, volume 218, pages 334–340, 1998.

89 D. Rixen. *Substructuring and dual methods in structural analysis*. PhD thesis, University of Liège, Belgique, 1997.

90 D. Rixen. A dual craig-bampton method for dynamic substructuring. *J. Comput. Appl. Math.*, 168:383–391, 2004.

91 D. Rixen and C. Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Int. J. Num. Meth. Eng.*, 44(4):489–516, 1999.

92 D. Rixen, C. Farhat, R. Tezaur, and J. Mandel. Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver. *Int. J. Num. Meth. Eng.*, 46(4):501–534, 1999.

93 Daniel Rixen. Extended preconditioners for the feti method applied to constrained problems. *Int. Journal for Numerical methods in engineering*, 54(1):1–26, 2002.

94 F.-X. Roux. Parallel implementation of direct solution strategies for the coarse grig solvers in 2-level FETI method. Technical report, ONERA, Paris, France, 1997.

95 Y. Saad. Analysis of augmented Krylov subspace methods. *SIAM J. Matrix Anal. Appl.*, 18(2):435–449, April 1997.

96 Y. Saad. *Iterative methods for sparse linear systems*. PWS Publishing Company, 3rd edition, 2000.

97 Y. Saad and M. H. Schultz. GMRes: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

98 Youssef Saad. On the Lanczos method for solving symmetric linear systems with several right hand sides. *Math. Comp.*, 48:651–662, 1987.

99 L. Series, F. Feyel, and F.-X. Roux. Une méthode de décomposition de domaine avec deux multiplicateurs de Lagrange. In *Actes du 16eme congrès français de mécanique*, 2003.

100 L. Series, F. Feyel, and F.-X. Roux. Une méthode de décomposition de domaine avec deux multiplicateurs de Lagrange, application au calcul des structures, cas du contact. In *Actes du sixième colloque national en calcul des structures*, volume III, pages 373–380, 2003.

101 D. Stefanica and A. Klawonn. The FETI method for mortar finite elements. In *Proceedings of 11th International Conference on Domain Decomposition Methods*, pages 121–129, 1999.

102 P. Le Tallec. Domain-decomposition methods in computational mechanics. *Computational Mechanics Advances*, 1(2):121–220, 1994. North-Holland.

103 P. Le Tallec, J. Mandel, and M. Vidrascu. A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. *SIAM J. Num. Ana.*, 35(2):836–867, April 1998.

104 P. Le Tallec, Y.-H. De Roeck, and M. Vidrascu. Domain-decomposition methods for large linearly elliptic three dimensional problems. *J. Comp. Appl. Math.*, 34:93–117, 1991. Elsevier Science Publishers, Amsterdam.

105 P. Le Tallec and M. Vidrascu. Méthodes de décomposition de domaines en calcul de structures. In *Actes du premier colloque national en calcul des structures*, volume I, pages 33–49, 1993.

106 P. Le Tallec and M. Vidrascu. Generalized Neumann-Neumann preconditioners for iterative substructuring. In *Proceedings of the ninth conference on Domain Decomposition*, June Bergen 1996. to appear.

107 A. van der Sluis and H. can der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.

108 Pieter Wesseling. *An Introduction to Multigrid Methods*. R.T. Edwards, Inc, 2004.

109 O.C. Zienkiewicz and R.L. Taylor. *The finite element method*. Mc Graw-Hill Book COmpagny, 1989.

A Krylov iterative solvers

Krylov iterative solvers for the resolution of linear systems have been widely studied. The aim of this section is just to briefly present important results and algorithms, reader interested in wider documentation can refer to [96], and to [4] for shorter explanation.

Krylov methods belong to the projection class of iterative algorithms, which consist in approximating solution $S^{-1}b$ of system $Sx = b$ by vector $p(S)b$ where p is a smartly built polynomial.

In this section we consider the iterative solution to system $Sx = b$. S is a $n \times n$ matrix and b a vector in range(S). The ith iteration leads to approximation x_i of the solution, associated residual is $r_i = b - Sx_i = S(x - x_i)$. Initialization is x_0 (most often $x_0 = 0$). Canonical (orthonormal) basis of $\mathbb{R}^n$ reads $(e_1, \ldots, e_n)$.

A.1 Principle of Krylov solvers

Krylov solvers are based on the iterative construction of so-called "Krylov subspace" $\mathcal{K}_m(S, r_0)$ defined by:

$$\mathcal{K}_m(S, r_0) = \text{span}\left(r_0, \ldots, S^{m-1}r_0\right) \tag{1}$$

The solution to linear system consists in searching x_m under the following constraints:

$$\begin{cases} x_m \in x_0 + \mathcal{K}_m(S, r_0) \\ r_m \perp_? \mathcal{K}_m(S, r_0) \end{cases} \tag{2}$$

where the choice of the orthogonality relationship enables to define various approaches.

A.2 Most used solvers

We herein present two of the principal Krylov solvers. First GMRes [97] which is suited to any type of matrix, then conjugate gradient which is adapted to symmetric definite positive matrices.

Of course, the iterative solution to a linear system assumes that a convergence criterion is employed, and that a limit of precision is set so that the system is supposed to have converged once the criterion is below this precision. We note ε this limit value of the criterion.

A.3 GMRes

Algorithm GMRes (alg. A.1) consists in an oblique projection based on the construction of Krylov subspace $\mathcal{K}_m(S, v_0)$ with $v_0 = r_0/\|r_0\|_2$. The research principle is:

$$\begin{cases} x_m \in x_0 + \mathcal{K}_m(S, r_0) \\ r_m \perp S\mathcal{K}_m(S, r_0) \end{cases} \tag{3}$$

which is equivalent to finding $x_m \in x_0 + \mathcal{K}_m(S, r_0)$ minimizing $\|r_m\|_2$.

A particulary striking property of GMRes is not to compute the approximation at each iteration, a smart implementation of GMRes enables to directly access the norm of the residual $\|r_j\|_2$. Only the final approximation x_m is computed (by the inversion of a $m \times m$ upper triangular matrix). From the computation complexity point of view, each iteration consists in a full orthonormalization of vector w_j with respect to $\mathcal{K}_j$.

Algorithm GMRes(m) (or restarted GMRes) consists in stopping computation before convergence at *a priori* fixed step m and restarting computation using previous x_m as initialization. This strategy aims at minimizing orthogonalization computations by limiting the size of Krylov subspaces [29]. This method may lead to stagnation for non positive definite matrices.

Algorithm A.1 GMRes

1: Compute $r_0 = b - Sx_0$, $v_0 = r_0/\|r_0\|_2$
2: **for** $j = 0, \ldots, m-1$ **do**
3:    Compute $w_j = Sv_j$
4:    **for** $i = 0, \ldots, j$ **do**
5:       $h_{ij} = (v_i, w_j)$
6:       $w_j = w_j - h_{ij}v_i$
7:    **end for**
8:    $h_{(j+1)j} = \|w_j\|_2$
9:    **if** $\|r_j\|_2 \leqslant \varepsilon$ **then**
10:       stop
11:    **else**
12:       $v_{j+1} = w_j/h_{(j+1)j}$
13:    **end if**
14: **end for**
15: Compute y_m minimizing $\left\| \|r_0\|_2 e_1 - \bar{H}_m y \right\|_2$ and set $x_m = x_0 + V_m y_m$

A.4 Conjugate gradient

Let S be a symmetric positive definite matrix, a conjugate gradient algorithm consists in an orthogonal projection. The research principle is:

$$\left\{ \begin{array}{l} x_m \in x_0 + \mathcal{K}_m(S, r_0) \\ r_m \perp \mathcal{K}_m(S, r_0) \end{array} \right. \tag{4}$$

which is equivalent to finding $x_m \in x_0 + \mathcal{K}_m(S, r_0)$ minimizing $\|x_m - x\|_S$.

Because of the properties of S, conjugation (orthogonality) properties appear, leading to algorithm A.2. The algorithm is based on the construction of various basis of $\mathcal{K}_m(S, r_0)$:

Algorithm A.2 Conjugate gradient

1: Compute $r_0 = b - Sx_0$, set $w_0 = r_0$
2: **for** $j = 0, \ldots, m$ **do**
3:    $\alpha_j = (r_j, r_j)/(Sw_j, w_j)$
4:    $x_{j+1} = x_j + \alpha_j w_j$
5:    $r_{j+1} = r_j - \alpha_j Sw_j$
6:    $\beta_j = (r_{j+1}, r_{j+1})/(r_j, r_j)$
7:    $w_{j+1} = r_{j+1} + \beta_j w_j$
8: **end for**

(r_m) (residual basis) is orthogonal, (w_m) (research direction basis) is S-orthogonal. Step $6-7$ of algorithm A.2 is the S-orthogonalization of w_{j+1} with respect to w_j which theoretically implies the orthogonality of w_{j+1} with respect to all previous research directions. However this orthogonality property is numerically lost as the number of iterations increases, it is then more suited to use a full orthogonalization of research directions leading to algorithm A.3.

Full reorthogonalization is often compulsory for complex simulations. Various implementations are available (among others Gram-Schmidt, modified Gram-Schmidt, iterative Gram-Schmidt [71, 51]) depending on the chosen ratio between precision and computational cost. Our experience leads us to prefer modified Gram-Schmidt algorithm (the one used in algorithm A.1) to classical Gram-Schmidt (the one used in algorithm A.3). Note that once

---

Algorithm A.3 Reorthogonalized conjugate gradient

1: Compute $r_0 = b - Sx_0$, set $w_0 = r_0$
2: **for** $j = 0, \ldots, m$ **do**
3:     $\alpha_j = (r_j, r_j)/(Sw_j, w_j)$
4:     $x_{j+1} = x_j + \alpha_j w_j$
5:     $r_{j+1} = r_j - \alpha_j Sw_j$
6:     For $0 \leqslant i \leqslant j$, $\beta_j^i = -(r_{j+1}, Sw_i)/(w_i, Sw_i)$
7:     $w_{j+1} = r_{j+1} + \sum_{i=1}^{j} \beta_j^i w_i$
8: **end for**

---

fully reorthogonalized, conjugate gradient is almost as expensive as GMRes. Anyhow conjugate gradient provides the approximation at each iteration, which can be very useful (see for instance section 3.2.3, where such an information enables the computation of relevant convergence criterion).

A.5 Study of the convergence, preconditioning

Because of their error-minimization property, conjugate gradient and GMRes have convergence theorems with known minimal convergence rate, for instance for conjugate gradient:

$$\|x - x_m\|_S \leqslant 2 \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right]^m \|x - x_0\|_S \tag{5}$$

where κ is the condition number of matrix S. Condition number is the ratio between the biggest and the smallest eigenvalues.

$$\kappa = \left| \frac{\lambda_n}{\lambda_1} \right| \text{ with } |\lambda_1| \leqslant |\lambda_2| \leqslant \ldots \leqslant |\lambda_n| \text{ eigenvalues of } S \tag{6}$$

Moreover performance results of Krylov iterative solvers are strongly linked to the spectrum of matrix S. More precisely only the active spectrum (set of eigenvalues which the right hand side is not orthogonal to the associated eigenvectors) influences the convergence; condition number κ can be replaced with active condition number κ_{act} inside relation (5) leading to better convergence range. More precise study would lead to the introduction of Ritz spectrum and effective condition number [107].

These simple considerations are sufficient to explain the interest of preconditioning the system: the idea is to solve equivalent system $\widetilde{S}^{-1}Sx = \widetilde{S}^{-1}b$ where $\widetilde{S}^{-1}$ is a well-chosen matrix providing the system with better spectral properties (if $\widetilde{S}^{-1} \approx S^{-1}$ then condition number is optimal, which justifies the notation).

For conjugate gradient, the use of preconditioner may seem problematic since the symmetry is *a priori* lost. However if preconditioner $\widetilde{S}^{-1}$ is symmetric definite positive, applying conjugate gradient to nonsymmetric system is equivalent to a symmetric resolution $((L^{-T}SL^{-1})(Lx) = L^{-T}b$ with Cholesky factorization $\widetilde{S} = L^T L)$ and the method is still relevant.

So preconditioning the above two algorithms is simply realized replacing S by $\widetilde{S}^{-1}S$ and b by $\widetilde{S}^{-1}b$ in lines 1 and 3 of algorithm A.1, and in lines 1 and 5 of algorithm A.3 (anyhow the research directions are still S-orthogonal). Of course the main problem remains the definition of an efficient preconditioner.

A.6 Constrained Krylov methods, projector implementation

We may deal (for instance in the dual approach) with constrained systems such as:

$$\begin{pmatrix} S & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} x \\ \alpha \end{pmatrix} = \begin{pmatrix} b \\ e \end{pmatrix} \tag{7}$$

Because constraint $G^T x_0 = e$ is compulsory, it is often referred to as "admissibility constraint". A classical solution is to find an initialization x_0 which satisfies constraint and then ensure that the remainder of the solution is researched inside a supplemental space: $G^T(x_i - x_0) = 0$. A projected algorithm naturally arises:

$$x = x_0 + P x^* \tag{8}$$
$$G^T x_0 = e \tag{9}$$
$$G^T P = 0 \tag{10}$$

which leads to:

$$x_0 = QG \left(G^T Q G \right)^{-1} e \tag{11}$$
$$P = I - QG \left(G^T Q G \right)^{-1} G^T \tag{12}$$

where Q is a matrix so that matrix $G^T Q G$ is invertible. Iterative system then reads:

$$P^T S P x^* = P^T (b - S x_0) \tag{13}$$

α can be post-computed $\alpha = \left(G^T Q G \right)^{-1} G^T Q (b - S x)$.

A.7 Augmented-Krylov methods, projector implementation

Augmented-Krylov methods [14, 95] are employed to add optional constraints to the resolution of a system. The principle is to set subspace $\mathcal{C}$ in $\mathbb{R}^n$ of dimension n_c represented by $n \times n_c$ rectangular matrix C ($\text{range}(C) = \mathcal{C}$, for more simplicity we suppose that C is full-ranked-column), then to define augmented-Krylov subspace $\widetilde{\mathcal{K}}_m(S, r_0, C) = \mathcal{K}_m(S, r_0) + \text{range}(C)$, and to use the following research principle:

$$\begin{cases} x_m \in x_0 + \widetilde{\mathcal{K}}_m(S, r_0, C) \\ r_m \perp_? \widetilde{\mathcal{K}}_m(S, r_0, C) \end{cases} \tag{14}$$

Augmented-Krylov methods can be implemented either by reorthogonalization schemes or by projection methods which are the one we propose to present here. The research space is separated into two subspaces: $\text{range}(C)$ and a supplemental subspace. The part of the solution in $\text{range}(C)$ is detected during initialization, while the remainder is iteratively looked for, projector P ensures the research is realized inside the correct subspace.

$$x = x_0 + P x^* \tag{15}$$
$$C^T r_0 = C^T (b - S x_0) = 0 \tag{16}$$
$$C^T S P = 0 \tag{17}$$

Which leads to:

$$x_0 = C \left(C^T S C \right)^{-1} C^T b \tag{18}$$
$$P = I - C \left(C^T S C \right)^{-1} C^T S \tag{19}$$

system then reads:

$$SPx^* = b - Sx_0 \tag{20}$$

$$\text{or } P^T SPx^* = P^T (b - Sx_0) = P^T b \tag{21}$$

Though it can be proved that projected system is better conditioned than original problem [25], the efficiency of the method essentially depends on the choice of matrix C, which is most often an opened problem. Within the framework of domain decomposition methods, this choice can be guided by several considerations. In the framework of multiresolution, reuse of previous numerical information can lead to very interesting performance results [85, 58, 59, 84, 86, 87]

A.8 Constrained augmented Krylov methods

We here consider solving constrained system (7) with C-augmented algorithm. The admissibility constraint is often referred to as first level constraint and augmentation as second level constraint. Two strategies are possible, the first consists in mixing levels together while the second respects the hierarchy between constraints.

One projector strategy: set $J = \begin{pmatrix} G & S^T H \end{pmatrix}$ and $\tilde{e} = \begin{pmatrix} -e \\ H^T b \end{pmatrix}$, then system reads:

$$\begin{pmatrix} S & J \\ J^T & 0 \end{pmatrix} \begin{pmatrix} x \\ \tilde{\alpha} \end{pmatrix} = \begin{pmatrix} b \\ \tilde{e} \end{pmatrix} \tag{22}$$

the following initialization/projection are employed (Q is a parameter to tune)

$$x_0 = J \left(J^T Q J \right)^{-1} \tilde{e} \tag{23}$$

$$P = I - Q J \left(J^T Q J \right)^{-1} J^T \tag{24}$$

Because Q is not easy to interpret and choose, this method is hardly ever used.

Two-projector strategy: the two conditions are imbricated. First ensure admissibility constraint

$$x = x_0 + Px^* \tag{25}$$

$$x_0 = G \left(G^T Q G \right)^{-1} e \tag{26}$$

$$P = I - Q G \left(G^T Q G \right)^{-1} G^T \tag{27}$$

then set

$$x^* = x_0^* + P^* x^{**} \tag{28}$$

$$x_0^* = C \left(C^T P^T SPC \right)^{-1} C^T P^T (b - Sx_0) \tag{29}$$

$$P^* = I - PC \left(C^T P^T SPC \right)^{-1} C^T P^T S \tag{30}$$

so that optimality constraint is verified. As can be seen such an approach is equivalent to classical augmentation with making second level constraints consistent with the first (setting $C^* = PC$).