An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems

B. Vereecke [a] H. Bavestrello [a] D. Dureisseix [b,*]

[a] *LMT-Cachan (École Normale Supérieure de Cachan / CNRS / Univ. Paris 6)*
61 Avenue du Président Wilson, F-94235 Cachan CEDEX, FRANCE

[b] *LMGC (Univ. Montpellier 2 / CNRS)*
CC 048, Place Eugène Bataillon, F-34095 Montpellier CEDEX 5, FRANCE

Abstract

Incompressible and nearly incompressible problems are treated herein with a mixed finite element formulation in order to avoid ill-conditioning that prevents accuracy in pressure estimation and lack of convergence for iterative solution algorithms. A multilevel dual domain decomposition method is then chosen as an iterative algorithm: the original FETI and FETI-DP methods are extended to deal with such problems, when the discretization of the pressure field is discontinuous throughout the elements. A dedicated augmentation of the algorithms is proposed and the different methods are compared with several preconditioners, for bidimensional test cases. The resulting approaches are both optimal and numerically scalable, and their costs are estimated with a complexity analysis.

Key words: Stokes, Herrmann, multilevel domain decomposition, scalability, mixed formulation

1 Introduction

Incompressible or nearly incompressible problems arise, for instance, from fluid flows, polymer injection, casting, extruding, spinning, or from the simulation of elastomer structures, etc. Because low order divergence-free displacement

* corresponding author
Email addresses: vereecke@lmt.ens-cachan.fr (B. Vereecke), bavestre@lmt.ens-cachan.fr (H. Bavestrello), dureisse@lmgc.univ-montp2.fr (D. Dureisseix).

discretizations with finite elements may lead to severe locking problems (see [1] for instance), the treatment of such problems is usually performed with an approximation of the incompressibility (or divergence-free) condition. For instance, one may use the penalty method, or equivalently for the case of elasticity, a Poisson's ratio close to 0.5; but these methods lead to ill-conditioned systems of equations. When the size of the problem increases, this can prevent the use of iterative methods, though they are often less CPU expensive than direct approaches, [2,3]. Furthermore, the precision obtained for the pressure estimation is poor. Other techniques use a conforming or non-conforming mixed finite element (displacement – pressure formulation), with or without under-integration [4,5].

In this paper, we are dealing with mixed formulation and pressure – displacement (or celerity for Stokes problem) coupled problems, within the context of large-scale finite element simulations. In this situation, domain decomposition methods are efficient computational strategies well suited to parallel architecture computers, especially multilevel approaches, see [6–8] for instance. Several *multilevel* domain decomposition methods have already been applied to incompressible problems. Concerning overlapping Schwarz methods, the reader can refer to [9]. For non-overlapping Schur methods, the primal BDD method [10] (for Balancing Domain Decomposition) has been extended: see for instance [11–13], but less has been done for dual methods. Usually, the incompressible condition is treated iteratively with a Uzawa iteration (i.e. with a physical partitioning of the unknowns), or a pressure correction algorithm is used. In these cases, the inner-loop linear systems are solved with classical domain decomposition; see [14] and [15,16] for an example of each of these approaches. A *monolevel* dual Schur method has been used in [17], but the extension to a multilevel scheme has not been addressed. Concerning other monolevel Schur methods for Stokes problem, one may refer to the early work [18] or [19]. In all of these cases, a monolevel scheme does not possess the numerical scalability property.

We are using herein a dual and multilevel domain decomposition method — the FETI method [20] (for Finite Element Tearing and Interconnecting) — and we take into account the flexibility of such a method to extend the original algorithm to incompressible or nearly incompressible linear elasticity problems, while keeping the coupling between displacement and pressure inside each subdomain. We first discuss the treatment of incompressibility with a mixed formulation, for the cases where the pressure discretization is discontinuous, and propose a first extension of the method that preserve the original performances obtained on compressible cases. The test cases concern 2D plane strain problems and illustrates the influence of the size of the problem, and of the number of subdomains. The resulting algorithms are still optimal and numerically scalable. Finally, an augmentation of the method dedicated to (nearly) incompressible problems is proposed to improve the previous perfor-

mances. Both the original FETI and the dual-primal FETI-DP [21] methods are extended to (nearly) incompressible problems and lead to the FETI-I and FETI-DPI approaches.

2 Treatment of incompressibility

For compressible linear elasticity, the material behavior is modelled with the Hooke's law that relates the stress $\boldsymbol{\sigma}$ to the strain $\boldsymbol{\varepsilon}$ through a bijective linear operator $\boldsymbol{D}$. With an isotropic material, only 2 independent coefficients are needed, for instance, the Young's modulus E and the Poisson's ratio ν, or the Lamé's coefficients μ (shear modulus) and λ:

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon} = 2\mu\boldsymbol{\varepsilon} + \lambda(\operatorname{Tr}\boldsymbol{\varepsilon})\mathbf{1} = \frac{E}{1+\nu}\boldsymbol{\varepsilon} + \frac{\nu E}{(1+\nu)(1-2\nu)}(\operatorname{Tr}\boldsymbol{\varepsilon})\mathbf{1}$$

For nearly incompressible materials, when $\underline{U}$ is the displacement field, $\operatorname{div}\underline{U} = \operatorname{Tr}\boldsymbol{\varepsilon}(\underline{U}) \to 0$, $\nu \to 0.5$ and $\lambda \to \infty$, such that $\lambda(\operatorname{Tr}\boldsymbol{\varepsilon})$ remains finite. Without a particular care, this coefficient is expected to be poorly evaluated [22]; moreover, when a purely displacement-oriented finite element formulation is used, the condition number of the rigidity matrix increases as ν goes close to 0.5. This phenomenon is amplified when the size of the finite element model increases, and can prevent a classical iterative algorithm to converge, if applied to such an ill-conditioned problem.

2.1 Mixed formulation

To circumvent the previous difficulties, a mixed formulation is classically used, see [23,4,24,25] for instance. Both the displacement and the pressure (related to the term $H = \lambda \operatorname{Tr}\boldsymbol{\varepsilon}$) are discretized. The drawbacks are a larger problem with an higher fill-in, and a rigidity matrix which is no more symmetric positive definite.

In order to simplify, consider a structure Ω made with a linear isotropic material. It is subjected to a body force $\underline{f}_d$ and an external force field $\underline{F}_d$ on a part $\partial_2\Omega$ of its boundary. A classical variational formulation (also called perturbated Lagrangian) consists in finding the saddle point $(\underline{U}, H)$ of:

$$\int_\Omega \left\{ \mu \operatorname{Tr}[\boldsymbol{\varepsilon}(\underline{U})\boldsymbol{\varepsilon}(\underline{U})] - \frac{1}{2\lambda}H^2 + H \operatorname{Tr}\boldsymbol{\varepsilon}(\underline{U}) \right\} d\Omega - \int_\Omega \underline{U}\cdot\underline{f}_d d\Omega - \int_{\partial_2\Omega} \underline{U}\cdot\underline{F}_d dS$$

Once $\underline{U}$ and H are correctly discretized, the corresponding linear system of

equations is:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C}^T \\ \boldsymbol{C} & -\boldsymbol{N} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{h} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \tag{1}$$

$\boldsymbol{u}$ and $\boldsymbol{h}$ are the vectors of unknowns describing the fields $\underline{U}$ and H; $\boldsymbol{K}$ is a standard rigidity matrix associated to a material with a shear modulus μ and a null second Lamé coefficient; $\boldsymbol{C}$ is a coupling term and $\boldsymbol{N}$ becomes null when the physical material is incompressible:

$$\boldsymbol{u}^T\boldsymbol{C}^T\boldsymbol{h} = \boldsymbol{h}^T\boldsymbol{C}\boldsymbol{u} = \int_\Omega H \operatorname{Tr}\boldsymbol{\varepsilon}(\underline{U})d\Omega \qquad ; \qquad \boldsymbol{h}_1^T\boldsymbol{N}\boldsymbol{h}_2 = \int_\Omega H_1 \frac{1}{\lambda} H_2 d\Omega$$

$\boldsymbol{K}$ and $\boldsymbol{N}$ are both symmetric definite positive (SPD), but this is not the case for the matrix of the system (1).

For compressible cases, $\boldsymbol{N}$ is regular and the elimination of the unknown $\boldsymbol{h}$ in (1) can be performed with a condensation and leads to: $(\boldsymbol{K}+\boldsymbol{C}^T\boldsymbol{N}^{-1}\boldsymbol{C})\boldsymbol{u} = \boldsymbol{f}$. When $\nu \to 0.5$, $\boldsymbol{N} \to \boldsymbol{0}$ and this formulation is equivalent to a penalization procedure, intended to prescribe $\operatorname{Tr}\boldsymbol{\varepsilon}(\underline{U}) \approx 0$.

For incompressible cases, $\boldsymbol{N} = \boldsymbol{0}$ and $\boldsymbol{h}$ is interpreted as a Lagrange multiplier in (1) that enforces the discrete form of the divergence-free condition: $\boldsymbol{C}\boldsymbol{u} = \boldsymbol{0}$.

A trivial generalization of the previous formulation can be obtained with a slightly different splitting of spherical terms in the constitutive relation:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \tilde{\lambda}(\operatorname{Tr}\boldsymbol{\varepsilon})\boldsymbol{1} + p\boldsymbol{1}$$

with $p = (\lambda - \tilde{\lambda})\operatorname{Tr}\boldsymbol{\varepsilon}$; $\tilde{\lambda}$ is a parameter to be chosen ($\tilde{\lambda} = 0$ recovers the previous formulation with $p = H$). The corresponding variational formulation is to find the saddle point of:

$$\int_\Omega \left\{ \frac{1}{2}\operatorname{Tr}[\boldsymbol{\varepsilon}(\underline{U})\{2\mu\boldsymbol{\varepsilon}(\underline{U}) + \tilde{\lambda}\operatorname{Tr}\boldsymbol{\varepsilon}(\underline{U})\boldsymbol{1}\}] + p\operatorname{Tr}\boldsymbol{\varepsilon}(\underline{U}) - \frac{1}{2(\lambda-\tilde{\lambda})}p^2 \right\} d\Omega + \\ - \int_\Omega \underline{U}\cdot\underline{f}_d d\Omega - \int_{\partial_2\Omega} \underline{U}\cdot\underline{F}_d dS$$

and the discrete linear system of equations is:

$$\begin{bmatrix} \widetilde{\boldsymbol{K}} & \boldsymbol{C}^T \\ \boldsymbol{C} & -\boldsymbol{M} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \tag{2}$$

$\widetilde{\boldsymbol{K}}$ is a standard rigidity matrix associated to a material with a shear modulus μ and a second Lamé coefficient equal to $\tilde{\lambda}$; $\boldsymbol{M}$ still becomes null when the

physical material is incompressible:

$$\boldsymbol{p}_1^T \boldsymbol{M} \boldsymbol{p}_2 = \int_\Omega p_1 \frac{1}{\lambda - \tilde{\lambda}} p_2 d\Omega$$

2.2 Field discretization

To avoid spurious terms in the solution and locking phenomena, the discretization of both fields $\underline{U}$ and p must be chosen accordingly to the LBB condition, or must pass successfully the patch tests, [4,26,27]. In this paper, we use a discontinuous discretization of the pressure p, a continuous one for the displacement $\underline{U}$, and the finite element in all of the forecoming bidimensional examples is the straight-edge triangle with a linear interpolation of the pressure, a quadratic interpolation of the displacement augmented with a third order bubble (P2 bubble - P1 discontinuous). This element is LBB stable and 2d order.

3 Domain decomposition

In order to solve the systems (1) or (2), several strategies can be used.

A direct monolithic factorisation requires a suitable degree of freedom (dof) renumbering scheme and/or a suitable dof pivoting due to the lack of SPD property for the global problem matrix. Such a direct approach suffers from a high computational cost when the size of the problem increases. An iterative scheme can then be of interest to keep these costs affordable. Among them, partitioning techniques can be applied: for instance, a physical splitting between displacement and pressure dof, or a geometric splitting of the domain in subdomains. We are interested herein in such methods: the domain is partitioned into non overlapping subdomains, the displacement-pressure coupled problem remains coupled on each of them. If the pressure is not subjected to continuity conditions, "gluing" requirements on the interface between subdomains is only concerned with a null displacement jump.

The algorithms we are developing are based on the dual multilevel domain decomposition FETI [20]. More precisely, two versions of this method, the so-called FETI and FETI-DP formulations, are extended to get the FETI-I and FETI-DPI solution schemes, both for incompressible or nearly incompressible problems.

The proposed test problems used throughout this paper concern an idealized 2D incompressible polymer flow through an extrusion die, treated with plane

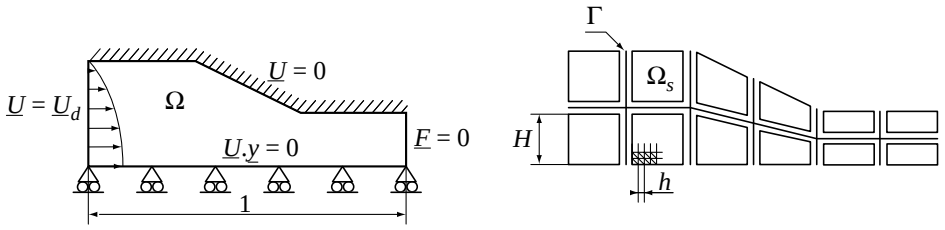

Fig. 1. Decomposition in subdomains Ω_s, and a global interface Γ

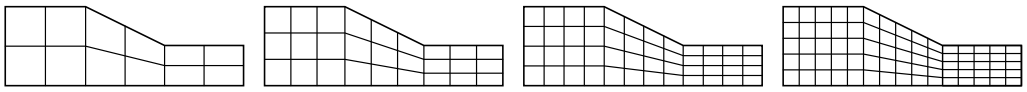

Fig. 2. Decomposition in $n = 12,\ 27,\ 48,\ 75$ subdomains

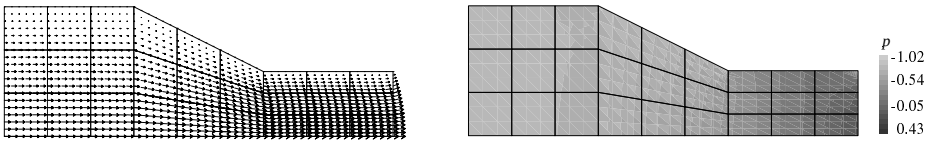

Fig. 3. Solution to the die problem with $n = 27$ subdomains and $h/H = 1/6$; left: displacement field, right: field of variable p

strain assumption (remark that with plane stress, the incompressibility condition does not prescribe any constraint on the in-plane kinematics). Due to symmetry, only half of the problem is modelled on Figure 1. A parabolic displacement field is prescribed on the left, while there is a free surface on the right. The structure is decomposed into n subdomains Ω_s, $s = 1, \ldots, n$, and a global interface Γ between them. h denotes a characteristic size of the finite elements, and H, the size of the subdomains. With an overall length of the problem set to 1, the subdomain size H varies from 1/6 to 1/15 when the number of subdomains increases from 12 to 75, on Figure 2. Figure 3 presents the solution obtained with $n = 27$ subdomains and a mesh refinement $h/H = 1/6$.

3.1 Extension of the FETI method to (nearly) incompressible problems

The FETI method is a *dual* domain decomposition method: the discrete kinematic "gluing" condition between subdomains is dualized *via* Lagrange multipliers [20,7]. When using a conforming finite element mesh for all the domain Ω, the Lagrange multipliers $\boldsymbol{\lambda}$ are defined at each node of the global interface Γ between subdomains. They represent the corresponding nodal force acting from a subdomain s on a subdomain s' (when using the convention $s < s'$ to uniquely define the sign of $\boldsymbol{\lambda}$). Therefore, each subdomain s needs a signed boolean matrix $\boldsymbol{B}^{(s)}$ to map the global force vector $\boldsymbol{\lambda}$ onto the forces acting on its boundary: $-\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}$. The kinematic constraint on the jump of displacement

between neighboring subdomains on the global interface is therefore:

$$\sum_s \boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{3}$$

The finite element equilibrium of a subdomain s, submitted to the external generalized forces $\boldsymbol{f}^{(s)}$ and the action of the multipliers, is:

$$\boldsymbol{K}_H^{(s)} \begin{bmatrix} \boldsymbol{u}^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{(s)} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{B}^{(s)^T} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda} \qquad \text{with} \qquad \boldsymbol{K}_H^{(s)} = \begin{bmatrix} \widetilde{\boldsymbol{K}}^{(s)} & \boldsymbol{C}^{(s)^T} \\ \boldsymbol{C}^{(s)} & -\boldsymbol{M}^{(s)} \end{bmatrix} \tag{4}$$

$\boldsymbol{K}_H^{(s)}$ is symmetric, but not positive and not obviously definite: for floating subdomains, i.e. subdomains with no prescribed displacement, or with internal mechanisms, the kernel of $\boldsymbol{K}_H^{(s)}$ contains rigid body motions. The set of such rigid body motions and null pressure of such a subdomain s is denoted by $\boldsymbol{R}^{(s)}$.

The first step to build the FETI-I method is to condense the unknowns $\boldsymbol{u}^{(s)}$ and $\boldsymbol{p}^{(s)}$ onto the multiplier $\boldsymbol{\lambda}$: provided that the right hand side of (4) is orthogonal to $\boldsymbol{R}^{(s)}$, i.e.

$$\boldsymbol{R}^{(s)^T} \left\{ \begin{bmatrix} \boldsymbol{f}^{(s)} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{B}^{(s)^T} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda} \right\} = 0 \tag{5}$$

the solution is obtained with an indefinite rigid body motion field using one of the pseudo-inverses $\boldsymbol{K}_H^{(s)^+}$ of $\boldsymbol{K}_H^{(s)}$: (4) is equivalent to

$$\begin{bmatrix} \boldsymbol{u}^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \boldsymbol{K}_H^{(s)^+} \left\{ \begin{bmatrix} \boldsymbol{f}^{(s)} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{B}^{(s)^T} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda} \right\} + \boldsymbol{R}^{(s)}\boldsymbol{\alpha}^{(s)}$$

$\boldsymbol{\alpha}^{(s)}$ are the arbitrary coordinates of the rigid body motion basis $\boldsymbol{R}^{(s)}$. Using this expression in the constraint (3) leads to

$$\boldsymbol{F}\boldsymbol{\lambda} - \boldsymbol{G}\boldsymbol{\alpha} - \boldsymbol{d} = \boldsymbol{0}$$

with

$$\boldsymbol{F} = \sum_s \begin{bmatrix} \boldsymbol{B}^{(s)} & \boldsymbol{0} \end{bmatrix} \boldsymbol{K}_H^{(s)^+} \begin{bmatrix} \boldsymbol{B}^{(s)^T} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{d} = \sum_s \begin{bmatrix} \boldsymbol{B}^{(s)} & \boldsymbol{0} \end{bmatrix} \boldsymbol{K}_H^{(s)^+} \begin{bmatrix} \boldsymbol{f}^{(s)} \\ \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{G} = \left[\begin{bmatrix} \boldsymbol{B}^{(1)} & \boldsymbol{0} \end{bmatrix} \boldsymbol{R}^{(1)} \ldots \begin{bmatrix} \boldsymbol{B}^{(n)} & \boldsymbol{0} \end{bmatrix} \boldsymbol{R}^{(n)} \right] \qquad \boldsymbol{\alpha}^T = \left[\boldsymbol{\alpha}^{(1)^T} \ldots \boldsymbol{\alpha}^{(n)^T} \right]$$

In order to complete the problem, the equations (5) are needed for each subdomain. Assembling them leads to:

$$\boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{e} \qquad \text{with} \qquad \boldsymbol{e} = \left[\boldsymbol{R}^{(1)^T} \begin{bmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{0} \end{bmatrix} \dots \boldsymbol{R}^{(n)^T} \begin{bmatrix} \boldsymbol{f}^{(n)} \\ \boldsymbol{0} \end{bmatrix} \right] \tag{6}$$

Finally, the dual problem is to find $(\boldsymbol{\lambda}, \boldsymbol{\alpha})$ such that:

$$\begin{bmatrix} \boldsymbol{F} & -\boldsymbol{G} \\ -\boldsymbol{G}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ -\boldsymbol{e} \end{bmatrix} \tag{7}$$

A key point is that, though we use a mixed formulation, $\boldsymbol{F}$ is still symmetric positive (not definite, especially when there are redundancies in kinematic constraints (3)).

To prove it, consider first the initial formulation (1), and an arbitrary multiplier field $\boldsymbol{\lambda}$; for the subdomain s, define

$$\begin{bmatrix} \boldsymbol{v}^{(s)} \\ \boldsymbol{\pi}^{(s)} \end{bmatrix} = \boldsymbol{K}_H^{(s)^+} \begin{bmatrix} \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} \\ \boldsymbol{0} \end{bmatrix}$$

Then,

$$\begin{aligned} \boldsymbol{\lambda}^T\boldsymbol{F}^{(s)}\boldsymbol{\lambda} &= \left[\boldsymbol{B}^{(s)}\boldsymbol{\lambda}^T \ \boldsymbol{0} \right] \boldsymbol{K}_H^{(s)^+} \begin{bmatrix} \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} \\ \boldsymbol{0} \end{bmatrix} \\ &= \left[\boldsymbol{B}^{(s)}\boldsymbol{\lambda}^T \ \boldsymbol{0} \right] \begin{bmatrix} \boldsymbol{v}^{(s)} \\ \boldsymbol{\pi}^{(s)} \end{bmatrix} = \left[\boldsymbol{B}^{(s)}\boldsymbol{\lambda}^T \ \boldsymbol{0} \right] \begin{bmatrix} \boldsymbol{v}^{(s)} \\ -\boldsymbol{\pi}^{(s)} \end{bmatrix} \\ &= \left[\boldsymbol{v}^{(s)^T} \ \boldsymbol{\pi}^{(s)^T} \right] \boldsymbol{K}_H^{(s)} \begin{bmatrix} \boldsymbol{v}^{(s)} \\ -\boldsymbol{\pi}^{(s)} \end{bmatrix} = \boldsymbol{v}^{(s)^T}\boldsymbol{K}^{(s)}\boldsymbol{v}^{(s)} + \boldsymbol{\pi}^{(s)^T}\boldsymbol{N}^{(s)}\boldsymbol{\pi}^{(s)} \end{aligned}$$

As $\boldsymbol{K}^{(s)}$ and $\boldsymbol{N}^{(s)}$ are positive, $\boldsymbol{F}^{(s)}$ and $\boldsymbol{F} = \sum_s \boldsymbol{F}^{(s)}$ are positive too. The next step is to recall that the generalized formulation (2) is strictly equivalent to the previous one for the local solution $\boldsymbol{u}^{(s)}$, and $\boldsymbol{p}^{(s)} = \lambda \boldsymbol{h}^{(s)}/(\lambda - \tilde{\lambda})$, so the positivity is ensured for this formulation also, as soon as $\tilde{\boldsymbol{\lambda}} \neq \boldsymbol{\lambda}$.

Therefore, a projected conjugate gradient is applicable to solve (7). A projection is required to enforce the admissibility condition (6) for $\boldsymbol{\lambda}$. The projector is for instance $\boldsymbol{P} = \boldsymbol{1} - \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T$. The corresponding overall algorithm

Table 1
The projected conjugate gradient algorithm

Initialize			
	$\boldsymbol{\lambda}^0$	$=$	$\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{e}$
	$\boldsymbol{r}^0$	$=$	$\boldsymbol{d}-\boldsymbol{F}\boldsymbol{\lambda}^0$
Iterate $k=1,\ 2,\ \cdots$ until convergence			
Project	$\boldsymbol{w}^{k-1}$	$=$	$\boldsymbol{P}^T\boldsymbol{r}^{k-1}$
Precondition	$\boldsymbol{z}^{k-1}$	$=$	$\bar{\boldsymbol{F}}^{-1}\boldsymbol{w}^{k-1}$
Re-project	$\boldsymbol{y}^{k-1}$	$=$	$\boldsymbol{P}\boldsymbol{z}^{k-1}$
Conjugate	ζ^k	$=$	$\dfrac{\boldsymbol{y}^{k-1^T}\boldsymbol{w}^{k-1}}{\boldsymbol{y}^{k-2^T}\boldsymbol{w}^{k-2}}\quad(\zeta^1=0)$
Search direction	$\boldsymbol{p}^k$	$=$	$\boldsymbol{y}^{k-1}+\zeta^k\boldsymbol{p}^{k-1}\quad(\boldsymbol{p}^1=\boldsymbol{y}^0)$
Matrix-vector product	$\boldsymbol{q}^k$	$=$	$\boldsymbol{F}\boldsymbol{p}^k$
Line search	η^k	$=$	$\dfrac{\boldsymbol{p}^{k-1^T}\boldsymbol{w}^{k-1}}{\boldsymbol{p}^{k^T}\boldsymbol{q}^k}$
Update	$\boldsymbol{\lambda}^k$	$=$	$\boldsymbol{\lambda}^{k-1}+\eta^k\boldsymbol{p}^k$
Residual	$\boldsymbol{r}^k$	$=$	$\boldsymbol{r}^{k-1}-\eta^k\boldsymbol{q}^k$

does not differ from the classical FETI method; in order to be self-contained, it is recalled in Table 1.

One has to notice that

- at each iteration, the algorithm requires 2 projections;
- each projection is a coarse problem on the entire domain with the rigid body motion coefficients $\boldsymbol{\alpha}$ as unknowns;
- the computational kernel is the matrix-vector product $\boldsymbol{F}\boldsymbol{v}$. $\boldsymbol{F}$ is never explicitly assembled; the product is performed in parallel as it requires independent Neumann problems on each subdomains;
- the preconditioner $\bar{\boldsymbol{F}}^{-1}$ is intended to be easily parallelizable. Its design is recalled in a following section.

This approach is a straightforward extension of the FETI method to mixed problems with a discontinuous discretized pressure field. Using a continuous approximation of the pressure would require a "gluing" in the pressure field at the interface. The operator $\boldsymbol{F}$ would be no more positive, and so, an other iteration scheme is required (GMRes for instance); this feature is out of the scope of the present paper.

An other situation is very similar to the one we consider herein: the case where the pressure is chosen to be continuous over any subdomain, but discontinuous

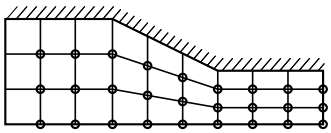

Fig. 4. Coarse nodes for the die problem with $n = 27$ subdomains

throughout the interface as discussed in [17]. In such a case, the discretized solution obviously depends on the domain decomposition.

3.2 Extension of the dual-primal version to (nearly) incompressible problems

Recently, a modified version of the FETI method have been designed under the FETI-DP acronym (for FETI Dual-Primal) [21,28]. Using an upgraded gluing kinematical condition, it allows the local rigidities of the subdomains to be regular, and needs only one coarse problem resolution per iteration. The key point is the definition of "corner" nodes that will constitute the coarse nodes of the problem. A simple way to define them is to use the geometric corners of the subdomains in 2D, but other possibilities exist, as soon as clamping these nodes leads to a regular Neumann problem on every subdomain, and to a regular coarse problem [29].

For the example of the Figure 1, and $n = 27$ subdomains, the coarse nodes are represented on Figure 4.

For (nearly) incompressible problems, it begins also with the splitting of the local dof (per subdomain) into local corner dof $\boldsymbol{u}_c^{(s)}$ and the remaining ones $\boldsymbol{u}_r^{(s)}$. Prescribing corner continuity between subdomains is equivalent to extract $\boldsymbol{u}_c^{(s)}$ from a unique global vector of corner dof $\boldsymbol{u}_c$: $\boldsymbol{u}_c^{(s)} = \boldsymbol{B}_c^{(s)}\boldsymbol{u}_c$. The remaining kinematic continuity conditions on interfaces is still expressed with a Lagrange multiplier $\boldsymbol{\lambda}_r$ and a boolean mapping matrix $\boldsymbol{B}_r^{(s)}$. The variational form corresponding to the equilibrium of the subdomains is:

$$\forall\ \boldsymbol{u}_r^{(s)\star},\ \boldsymbol{p}^{(s)\star},\ \boldsymbol{u}_c^{(s)\star} = \boldsymbol{B}_c^{(s)}\boldsymbol{u}_c^{\star},$$

$$\begin{bmatrix} \boldsymbol{u}_c^{(s)\star} \\ \boldsymbol{u}_r^{(s)\star} \\ \boldsymbol{p}^{(s)\star} \end{bmatrix}^T \boldsymbol{K}_H^{(s)} \begin{bmatrix} \boldsymbol{u}_c^{(s)} \\ \boldsymbol{u}_r^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_c^{(s)\star} \\ \boldsymbol{u}_r^{(s)\star} \\ \boldsymbol{p}^{(s)\star} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{f}_c^{(s)} \\ \boldsymbol{f}_r^{(s)} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{u}_c^{(s)\star} \\ \boldsymbol{u}_r^{(s)\star} \\ \boldsymbol{p}^{(s)\star} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B}_r^{(s)T} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda}_r$$

$$\text{with} \quad \boldsymbol{K}_H^{(s)} = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{cc}^{(s)} & \widetilde{\boldsymbol{K}}_{cr}^{(s)} & \boldsymbol{C}_c^{(s)T} \\ \widetilde{\boldsymbol{K}}_{rc}^{(s)} & \widetilde{\boldsymbol{K}}_{rr}^{(s)} & \boldsymbol{C}_r^{(s)T} \\ \boldsymbol{C}_c^{(s)} & \boldsymbol{C}_r^{(s)} & -\boldsymbol{M}^{(s)} \end{bmatrix}$$

It can be split in the local part per subdomain s:

$$\boldsymbol{K}_{H_{rr}}^{(s)} \begin{bmatrix} \boldsymbol{u}_r^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_r^{(s)} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{B}_r^{(s)^T} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda}_r - \begin{bmatrix} \widetilde{\boldsymbol{K}}_{rc}^{(s)} \\ \boldsymbol{C}_c^{(s)} \end{bmatrix} \boldsymbol{B}_c^{(s)} \boldsymbol{u}_c$$

$$\text{with} \quad \boldsymbol{K}_{H_{rr}}^{(s)} = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{rr}^{(s)} & \boldsymbol{C}_r^{(s)^T} \\ \boldsymbol{C}_r^{(s)} & -\boldsymbol{M}^{(s)} \end{bmatrix} \tag{8}$$

and the remaining coarse part which is assembled on the entire domain:

$$\left(\sum_s \boldsymbol{B}_c^{(s)^T} \widetilde{\boldsymbol{K}}_{cc}^{(s)} \boldsymbol{B}_c^{(s)} \right) \boldsymbol{u}_c + \sum_s \boldsymbol{B}_c^{(s)^T} \begin{bmatrix} \widetilde{\boldsymbol{K}}_{cr}^{(s)} & \boldsymbol{C}_c^{(s)^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_r^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \sum_s \boldsymbol{B}_c^{(s)^T} \boldsymbol{f}_c^{(s)}$$

With a correct choice of local corner dof, $\boldsymbol{K}_{H_{rr}}^{(s)}$ is regular and (8) is equivalent to:

$$\begin{bmatrix} \boldsymbol{u}_r^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \boldsymbol{K}_{H_{rr}}^{(s)\ -1} \left\{ \begin{bmatrix} \boldsymbol{f}_r^{(s)} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{B}_r^{(s)^T} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda}_r - \begin{bmatrix} \widetilde{\boldsymbol{K}}_{rc}^{(s)} \\ \boldsymbol{C}_c^{(s)} \end{bmatrix} \boldsymbol{B}_c^{(s)} \boldsymbol{u}_c \right\} \tag{9}$$

The kinematic constraints on the interface depends only on $\boldsymbol{u}_r^{(s)}$ dof:

$$\sum_s \begin{bmatrix} \boldsymbol{B}_r^{(s)} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_r^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = \boldsymbol{0}$$

With the previous expression of the local solution, (3.2) and (9) lead to:

$$\begin{bmatrix} \boldsymbol{F}_{rr} & \boldsymbol{F}_{rc} \\ \boldsymbol{F}_{cr} & -\boldsymbol{K}_{cc}^{\star} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_r \\ \boldsymbol{u}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{d}_r \\ -\boldsymbol{f}_c^{\star} \end{bmatrix} \tag{10}$$

with

$$\boldsymbol{F}_{rr} = \sum_s \left[\boldsymbol{B}_r^{(s)}\ \boldsymbol{0}\right] {\boldsymbol{K}_{Hrr}^{(s)}}^{-1} \begin{bmatrix} {\boldsymbol{B}_r^{(s)}}^T \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{d}_r = \sum_s \left[\boldsymbol{B}_r^{(s)}\ \boldsymbol{0}\right] {\boldsymbol{K}_{Hrr}^{(s)}}^{-1} \begin{bmatrix} \boldsymbol{f}_r^{(s)} \\ \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{F}_{rc} = \sum_s \left[\boldsymbol{B}_r^{(s)}\ \boldsymbol{0}\right] {\boldsymbol{K}_{Hrr}^{(s)}}^{-1} \begin{bmatrix} \widetilde{\boldsymbol{K}}_{rc}^{(s)} \\ \boldsymbol{C}_c^{(s)} \end{bmatrix} \boldsymbol{B}_c^{(s)} \qquad \boldsymbol{F}_{cr} = \boldsymbol{F}_{rc}^T$$

$$\boldsymbol{K}_{cc}^{\star} = \sum_s {\boldsymbol{B}_c^{(s)}}^T \boldsymbol{K}_{cc}^{(s)\star} \boldsymbol{B}_c^{(s)} \qquad \boldsymbol{K}_{cc}^{(s)\star} = \widetilde{\boldsymbol{K}}_{cc}^{(s)} - \left[\widetilde{\boldsymbol{K}}_{cr}^{(s)}\ {\boldsymbol{C}_c^{(s)}}^T\right] {\boldsymbol{K}_{Hrr}^{(s)}}^{-1} \begin{bmatrix} \widetilde{\boldsymbol{K}}_{rc}^{(s)} \\ \boldsymbol{C}_c^{(s)} \end{bmatrix}$$

$$\boldsymbol{f}_c^{\star} = \sum_s {\boldsymbol{B}_c^{(s)}}^T \boldsymbol{f}_c^{(s)\star} \boldsymbol{B}_c^{(s)} \qquad \boldsymbol{f}_c^{(s)\star} = \boldsymbol{f}_c^{(s)} - \left[\widetilde{\boldsymbol{K}}_{cr}^{(s)}\ {\boldsymbol{C}_c^{(s)}}^T\right] {\boldsymbol{K}_{Hrr}^{(s)}}^{-1} \begin{bmatrix} \boldsymbol{f}_r^{(s)} \\ \boldsymbol{0} \end{bmatrix}$$

The last step is the condensation of the corner dof $\boldsymbol{u}_c$ on the Lagrange multipliers $\boldsymbol{\lambda}_r$ in (10), $\boldsymbol{u}_c = {\boldsymbol{K}_{cc}^{\star}}^{-1}(\boldsymbol{f}_c^{\star} + \boldsymbol{F}_{cr}\boldsymbol{\lambda}_r)$, to get:

$$\boldsymbol{F}^{\star}\boldsymbol{\lambda}_r = \boldsymbol{d}^{\star} \quad \text{with} \quad \boldsymbol{F}^{\star} = \boldsymbol{F}_{rr} + \boldsymbol{F}_{rc}{\boldsymbol{K}_{cc}^{\star}}^{-1}\boldsymbol{F}_{cr} \quad \text{and} \quad \boldsymbol{d}^{\star} = \boldsymbol{d}_r - \boldsymbol{F}_{rc}{\boldsymbol{K}_{cc}^{\star}}^{-1}\boldsymbol{f}_c^{\star} \tag{11}$$

Notice that the matrix problem $\boldsymbol{F}^{\star}$ is still symmetric positive. The proof that $\boldsymbol{F}_{rr}$ is positive is the same as the one previously used for FETI-I algorithm. To prove that $\boldsymbol{K}_{cc}^{\star}$ is positive, let us consider

$$\begin{bmatrix} \boldsymbol{v}_c^{(s)} \\ \boldsymbol{v}_r^{(s)} \\ \boldsymbol{\pi}^{(s)} \end{bmatrix} = {\boldsymbol{K}_H^{(s)}}^{+} \begin{bmatrix} \boldsymbol{f}_c^{(s)} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{K}_H^{(s)} \begin{bmatrix} \boldsymbol{v}_c^{(s)} \\ \boldsymbol{v}_r^{(s)} \\ \boldsymbol{\pi}^{(s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_c^{(s)} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{K}_{cc}^{(s)\star}\boldsymbol{v}_c^{(s)} = \boldsymbol{f}_c^{(s)}$$

As it has already been proved that

$$\left[{\boldsymbol{f}_c^{(s)}}^T\ \boldsymbol{0}\ \boldsymbol{0}\right] {\boldsymbol{K}_H^{(s)}}^{+} \begin{bmatrix} \boldsymbol{f}_c^{(s)} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \geq 0$$

one gets

$$\left[{\boldsymbol{f}_c^{(s)}}^T\ \boldsymbol{0}\ \boldsymbol{0}\right] \begin{bmatrix} \boldsymbol{v}_c^{(s)} \\ \boldsymbol{v}_r^{(s)} \\ \boldsymbol{\pi}^{(s)} \end{bmatrix} = {\boldsymbol{f}_c^{(s)}}^T\boldsymbol{v}_c^{(s)} = {\boldsymbol{v}_c^{(s)}}^T\boldsymbol{K}_{cc}^{(s)\star}\boldsymbol{v}_c^{(s)} \geq 0$$

So, $\boldsymbol{K}_{cc}^{(s)\star}$ is positive, and $\boldsymbol{K}_{cc}^{\star}$ is positive too. To check if $\boldsymbol{K}_{cc}^{\star}$ is regular, let us look for a displacement field such that $\boldsymbol{u}_c^T\boldsymbol{K}_{cc}^{\star}\boldsymbol{u}_c = 0$. As all $\boldsymbol{K}_{cc}^{(s)\star}$

are positive, this is equivalent to $\boldsymbol{u}_c^{(s)^T} \boldsymbol{K}_{cc}^{(s)\star} \boldsymbol{u}_c^{(s)} = 0$ for any subdomain s. As all $\boldsymbol{K}_{rr}^{(s)}$ are regular: (i) $\boldsymbol{K}_{cc}^{(s)^\star}$ has rigid body motions (restricted to local c dof) as a kernel, so $\boldsymbol{u}_c^{(s)}$ must be the restriction to local c dof of a rigid body motion $\boldsymbol{u}^{(s)}$; and (ii) progressing from a subdomain to its neighbors, if *sufficiently* connected by some local c dof, one can conclude that all $\boldsymbol{u}^{(s)}$ are the restriction on the subdomains of the same global rigid body motion $\boldsymbol{u}$. If the initial problem is regular, the only admissible rigid body motion is $\boldsymbol{u} = \boldsymbol{0}$, so $\boldsymbol{u}_c = 0$ too. This concludes to the regularity of $\boldsymbol{K}_{cc}^\star$. There are therefore less severe conditions on the choice of corner modes to ensure regularity of $\boldsymbol{K}_{cc}^\star$ in such a way that less corner nodes are required. They are discussed in [29] and are directly applicable to the present case. Notice that, in 2D, when each subdomain shares with every neighbor that has a common edge at least two corner nodes, the regularity is guaranteed, though the number of corner nodes is not minimal. This is the case for the following examples in this paper.

To discuss the definiteness of $\boldsymbol{F}^\star$, let us now consider

$$\boldsymbol{\lambda}_r^T \boldsymbol{F}^\star \boldsymbol{\lambda}_r = \boldsymbol{\lambda}_r^T \boldsymbol{F}_{rr} \boldsymbol{\lambda}_r + (\boldsymbol{\lambda}_r^T \boldsymbol{F}_{rc}) \boldsymbol{K}_{cc}^{\star\,-1} (\boldsymbol{F}_{cr} \boldsymbol{\lambda}_r) = 0$$

Due to the positivity of $\boldsymbol{F}_{rr}$ and $\boldsymbol{K}_{cc}^\star$, this leads to $\boldsymbol{\lambda}_r^T \boldsymbol{F}_{rr} \boldsymbol{\lambda}_r = 0$. With

$$\begin{bmatrix} \boldsymbol{v}_r^{(s)} \\ \boldsymbol{\pi}^{(s)} \end{bmatrix} = \boldsymbol{K}_{H_{rr}}^{(s)\,-1} \begin{bmatrix} \boldsymbol{B}_r^{(s)^T} \boldsymbol{\lambda}_r \\ \boldsymbol{0} \end{bmatrix}$$

as for FETI-I, it leads to $\sum_s \boldsymbol{v}_r^{(s)^T} \widetilde{\boldsymbol{K}}_{rr}^{(s)} \boldsymbol{v}_r^{(s)} + \boldsymbol{\pi}^{(s)^T} \boldsymbol{N}^{(s)} \boldsymbol{\pi}^{(s)} = 0$. With $\widetilde{\boldsymbol{K}}_{rr}^{(s)}$ and $\boldsymbol{N}^{(s)}$ SPD, we can conclude to $\boldsymbol{v}_r^{(s)} = \boldsymbol{0}$ and $\boldsymbol{\pi}^{(s)} = \boldsymbol{0}$. So, $\boldsymbol{B}_r^{(s)^T} \boldsymbol{\lambda}_r = \boldsymbol{0}$ for any subdomain s. If there is no redundant multiplier (i.e. if each multiplier enforces an independant gluing condition), this allows us to conclude to $\boldsymbol{\lambda}_r = \boldsymbol{0}$. So, $\boldsymbol{F}^\star$ is definite, and therefore, is SPD. Notice that with redundant multipliers, $\boldsymbol{F}^\star$ is not definite any more, but this does not prevent a conjugate gradient algorithm to converge.

The FETI-DPI algorithm is a classical conjugate gradient applied to the problem (11). The algorithm is then similar with the one in Table 1, but no more projection is needed.

One has to notice that

- at each iteration, this algorithm requires the solution of only one global coarse problem consisting in finding the displacement of corner dof, during matrix-vector product $\boldsymbol{F}^\star \boldsymbol{v}$;
- as previously, the choice of the preconditioner is discussed in the next section.

3.3 A first choice of preconditioner and the corresponding comparisons

The first idea for preconditioning is to re-use the classical preconditioners in the case of a compressible problem. The so-called Dirichlet preconditioner is fully parallel and solves a Dirichlet-like problem per subdomain (with a prescribed displacement on the boundary):

$$\bar{\boldsymbol{F}}_D^{-1} = \sum_s \boldsymbol{W}\boldsymbol{B}^{(s)} \begin{bmatrix} \boldsymbol{S}_{bb}^{(s)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{B}^{(s)^T}\boldsymbol{W} \tag{12}$$

$\bar{\boldsymbol{F}}_D^{-1}$ applies to a kinematics jump on the interface and returns a generalized force on the same interface. $\boldsymbol{W}$ is a diagonal matrix storing in each of its entries the inverse of the multiplicity of an interfaced dof, and the subscript b denotes the dof that lie on the boundary of the subdomain, and the subscript i, the remaining ones. In particular, the standard rigidity matrix can be split in:

$$\widetilde{\boldsymbol{K}}^{(s)} = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{bb}^{(s)} & \widetilde{\boldsymbol{K}}_{bi}^{(s)} \\ \widetilde{\boldsymbol{K}}_{ib}^{(s)} & \widetilde{\boldsymbol{K}}_{ii}^{(s)} \end{bmatrix}$$

and a compressible preconditioner is: $\boldsymbol{S}_{bb}^{(s)} = \widetilde{\boldsymbol{K}}_{bb}^{(s)} - \widetilde{\boldsymbol{K}}_{bi}^{(s)}\widetilde{\boldsymbol{K}}_{ii}^{(s)^{-1}}\widetilde{\boldsymbol{K}}_{ib}^{(s)}$. This preconditioner is a function of the coefficient $\tilde{\lambda}$, or equivalently of the coefficient $\tilde{\nu} = \frac{1}{2}\tilde{\lambda}/(\tilde{\lambda}+\mu)$, used to compute the stiffness $\widetilde{\boldsymbol{K}}^{(s)}$.

This preconditioner can be applied to each of the FETI-I and FETI-DPI methods (for the second one, the matrix $\boldsymbol{B}_r^{(s)}$ is used and b dof are a partition of the r dof; there is no residual belonging to c nodes because there is no kinematic jump on these nodes all along iterations).

To compare the proposed methods to the standard FETI and FETI-DP ones, the test case proposed in Figure 1 is used. The number of subdomains is $n = 48$ (which corresponds to $H = 1/12$) and a refinement of the mesh such that $h/H = 1/6$. The different algorithms have been implemented with Matlab™.

Figure 5 and Table 2 report the number of iterations required to reach convergence. The convergence criteria is $\|\boldsymbol{w}^k\|/\|\boldsymbol{d}\| \leq 10^{-7}$. This number of iterations is plotted versus the Poisson's ratio of the real material.

For compressible cases ($\nu < 0.5$) both classical methods and adapted methods (with a mixed formulation) have similar performances; the methods suited to incompressible problems are then not competitive with the classical ones for this regime, as their finite element problems are larger, due to the pressure unknowns. For the mixed formulations, the results are plotted for different values of the parameter $\tilde{\nu}$. When $\tilde{\nu}$ approaches 0.5, the bad conditioning of the preconditioner decreases global performance, and the number of iterations

Table 2
Compared convergences (number of iterations) versus Poisson's ratio

Poisson's ratio ν	0.3	0.4	0.45	0.49	0.499	0.4995	0.5
FETI	35	39	46	83	168	183	-
FETI-I $\tilde{\nu} = 0$	35	37	40	46			50
FETI-I $\tilde{\nu} = 0.29$	35	36	38	43			47
FETI-I $\tilde{\nu} = 0.44$	42	43	44	49			52
FETI-I $\tilde{\nu} = 0.48$	56	57	59	67			72
FETI-DP	23	25	29	50	77	89	-
FETI-DPI $\tilde{\nu} = 0$	24	27	28	33			36
FETI-DPI $\tilde{\nu} = 0.29$	22	24	26	31			34
FETI-DPI $\tilde{\nu} = 0.44$	28	27	27	33			37
FETI-DPI $\tilde{\nu} = 0.48$	39	38	37	40			47

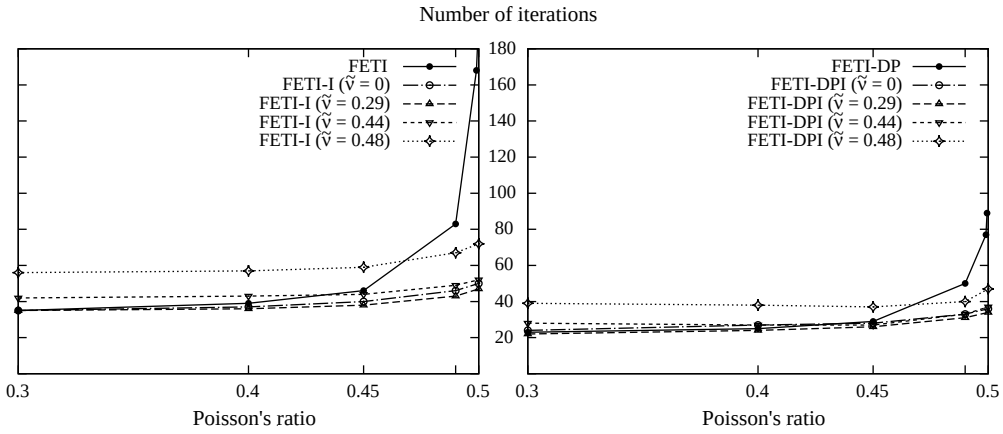

Fig. 5. Compared convergences (number of iterations) versus Poisson's ratio; left: for FETI like methods; right: for FETI-DP like methods

is higher than for the other cases. Taking $\tilde{\nu} = 0.29$ or even $\tilde{\nu} = 0$ (classical mixed formulation) leads to good results.

For nearly incompressible cases ($\nu \approx 0.5$), the classical methods exhibit difficulties to converge because the problem to solve is equivalent to a penalized one, and so, is ill-conditioned. For FETI-I and FETI-DPI methods, the convergence is preserved, even for the purely incompressible case ($\nu = 0.5$).

The characteristics of the original FETI methods, for compressible linear elasticity problems, concern the condition number κ of the iterations on the interface problem: it is asymptotically bounded as $\kappa = \mathcal{O}(1 + \log \frac{H}{h})^m$. Usually $m = 3$, or $m = 2$ with an appropriate projector, see [30].

Table 3
Optimality test for $n = 75$ subdomains

h/H	1/3	1/6	1/10
FETI (compressible)	33	37	39
FETI-I (incompressible)	49	50	52
FETI-IA (incompressible)	38	39	40
FETI-DP (compressible)	20	23	26
FETI-DPI (incompressible)	34	36	39
FETI-DPIA (incompressible)	21	24	26

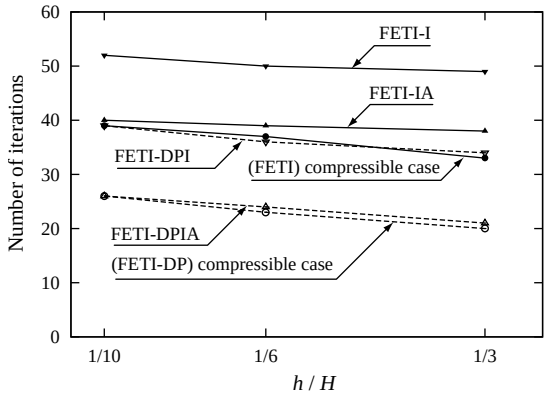

Fig. 6. Number of iterations depending of the problem size h/H (for $n = 75$ subdomains)

3.3.1 Optimality test

The optimality test consists in keeping the number n of subdomains constant while increasing the size of the problem, i.e. decreasing the ratio h/H. The algorithm is optimal if the number of iteration weakly depends on h/H. Figure 6 and Table 3 report this number of iterations versus the ratio h/H. The different curves correspond to an incompressible problem ($\nu = 0.5$) and to $\tilde{\nu} = 0.3$ for FETI-I and FETI-DPI methods. In order to compare to a reference, the number of iterations for a compressible case ($\nu = 0.3$) and standard FETI is also provided. FETI-IA and FETI-DPIA entries correspond to augmented versions of the algorithms that will be discussed in a later section.

When using $n = 75$ subdomains, and $h/H = 1/10$, the problem possesses 15 000 elements, 45 000 pressure dof and about 61 000 displacement dof (without the bubble dof which has been condensed at the element level).

All the tested approaches are optimal; nevertheless, for the incompressible case, the convergence is not as efficient as for the compressible case: the number of iterations is about 35% higher. Dual-primal versions of the methods need systematically less iterations, but their costs are not necessarily the same. The

Table 4
Numerical scalability test for $h/H = 1/10$

$H(n)$	1/6 (12)	1/9 (27)	1/12 (48)	1/15 (75)
FETI (compressible)	31	35	37	39
FETI-I (incompressible)	38	46	49	52
FETI-IA (incompressible)	34	36	38	40
FETI-DP (compressible)	17	22	24	26
FETI-DPI (incompressible)	24	33	37	39
FETI-DPIA (incompressible)	17	22	25	26

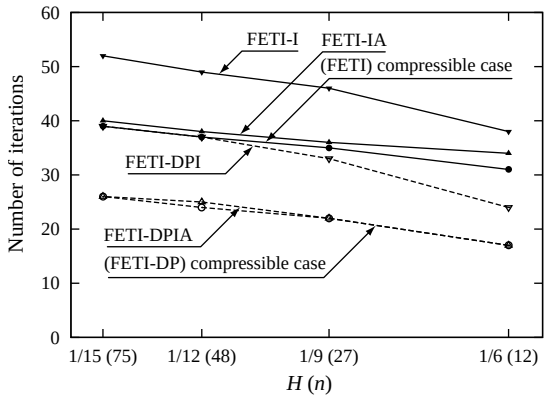

Fig. 7. Number of iterations depending of the number of subdomains (for $h/H = 1/10$)

section 5 will discuss this last point.

3.3.2 Numerical scalability test

The numerical scalability test consists in keeping a local problem of constant size (h/H constant) while increasing the number of subdomains (decreasing H), and so, increasing the size of the global problem. Figure 7 and Table 4 report the results for the same kind of situations than for the previous test.

The scalability is similar for all the methods. The previous remarks concerning the number of iterations are still applicable.

3.4 A second Dirichlet-like preconditioner

The next idea for a preconditioner is to use the same mixed formulation as for the direct problem. The form of the preconditioner is still the same, but now:

$$\boldsymbol{S}_{bb}^{(s)} = \widetilde{\boldsymbol{K}}_{bb}^{(s)} - \begin{bmatrix} \widetilde{\boldsymbol{K}}_{bi}^{(s)} & \boldsymbol{C}_b^{(s)} \end{bmatrix} {\boldsymbol{K}_{H_{ii}}^{(s)}}^{-1} \begin{bmatrix} \widetilde{\boldsymbol{K}}_{ib}^{(s)} \\ {\boldsymbol{C}_b^{(s)}}^T \end{bmatrix} \quad \text{with} \quad \boldsymbol{K}_{H_{ii}}^{(s)} = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{ii}^{(s)} & {\boldsymbol{C}_i^{(s)}}^T \\ \boldsymbol{C}_i^{(s)} & -\boldsymbol{M}^{(s)} \end{bmatrix} \tag{13}$$

$\boldsymbol{S}_{bb}^{(s)}$ does not depend any more on $\tilde{\nu}$ so, for compressible cases, the results are expected to be similar to those of the previous preconditioner.

For incompressible case, this preconditioner cannot be applied since $\boldsymbol{K}_{H_{ii}}^{(s)}$ is singular; the next section will discuss this topic.

For the (nearly) incompressible case, using this preconditioner is not obviously efficient: the residual (kinematic jump on interface) can have components with a significant volume variation per subdomain. Due to its nearly incompressibility nature, the preconditioner will return a force field on the boundary with a high hydrostatic component, and so, will impair convergence. To exemplify this point, let us consider again the test case of the die problem with a nearly incompressible material ($\nu = 0.4995$), $n = 48$ subdomains and $h/H = 1/6$. We use FETI-I method with $\tilde{\nu} = 0.3$ and different preconditioners: $\bar{\boldsymbol{F}}_C^{-1}$ for compressible preconditioner of section 3.3, and $\bar{\boldsymbol{F}}_H^{-1}$ for this last preconditioner using (13). The convergence rate is related to the condition number κ of the iteration matrix $\bar{\boldsymbol{F}}_D^{-1}\boldsymbol{P}^T\boldsymbol{F}\boldsymbol{P}$. Once we get rid of null eigenvalues corresponding to the unbalanced eigenvectors, and to the redundant gluing conditions, we obtain: $\kappa(\boldsymbol{P}^T\boldsymbol{F}\boldsymbol{P}) = 586$, $\kappa(\bar{\boldsymbol{F}}_C^{-1}\boldsymbol{P}^T\boldsymbol{F}\boldsymbol{P}) = 255$ and $\kappa(\bar{\boldsymbol{F}}_H^{-1}\boldsymbol{P}^T\boldsymbol{F}\boldsymbol{P}) = 1114$. As the condition number is higher for the nearly incompressible preconditioner, the convergence is expected to be lower.

This last preconditioner is therefore not useful if used alone. Nevertheless, its design is a motivation for the augmentation of the algorithms, in the next section.

4 Augmentation of the algorithms

Up to now, the basis of the FETI-I and FETI-DPI algorithms are very similar to the original FETI and FETI-DP ones because they do not take into account the particularities of the problem to be solved. This feature could improve the performance of the algorithms. For instance, for an incompressible case, one may use an incompressible preconditioner. This can be used only if the residual

to which this preconditioner is applied, satisfies the constant volume condition per subdomain. Such a constraint is an augmentation.

This technique has been previously used to accelerate the convergence by prescribing additional constraints to the successive iterates of the algorithm [31,32,28]. The drawback is the increase of the coarse problem size. If the acceleration counterbalances this overhead of cost, the performance of the overall algorithm is improved. It is now known in the litterature that an augmentation is mandatory for FETI-DP to be scalable for second order 3D problems.

We propose in this paper an augmentation suited to the kind of problems we are dealing with. As a first step, this augmentation is built for the incompressible case.

4.1 Augmentation of FETI-I algorithm for incompressible cases

In this situation, if one wishes to use an incompressible preconditioner, the residual it is applied to must satisfy a solvability condition: the volume of the subdomain s which is completely surrounded by neighbors (or prescribed displacement boundary) must be constant when a displacement $\underline{V}$ is prescribed by the preconditioner on its boundary. Let us denote by $\underline{U}$ its prolongation inside the subdomain Ω_s ($\underline{U}_{|\partial\Omega_s} = \underline{V}$); the condition is:

$$\int_{\Omega_s} \mathrm{Tr}\,\boldsymbol{\varepsilon}(\underline{U}) d\Omega = \int_{\partial\Omega_s} \underline{V} \cdot \underline{n} dS = 0$$

where $\underline{n}$ is the outward unit normal vector on the boundary. Once discretized, this condition is equivalent to ${\boldsymbol{p}_1^{(s)}}^T \boldsymbol{C}^{(s)} \boldsymbol{u}^{(s)} = 0$ with a uniform pressure field $\boldsymbol{p}_1^{(s)}$ on the whole subdomain (for instance equal to unity). The constraint matrix is then easily computed from the coupling matrix $\boldsymbol{C}^{(s)}$: $\boldsymbol{c}^{(s)} = {\boldsymbol{p}_1^{(s)}}^T \boldsymbol{C}^{(s)}$.

The Dirichlet preconditioner (12) and (13) acts on the projected residual: $\boldsymbol{P}^T\boldsymbol{r} = \boldsymbol{P}^T \sum_s \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)}$. For the subdomain s, ${\boldsymbol{B}^{(s)}}^T \boldsymbol{W} \boldsymbol{P}^T \boldsymbol{r}$ are the displacements computed in the beginning of the preconditioning step, so the constraint must be enforced on them: ${\boldsymbol{Q}^{(s)}}^T \boldsymbol{P}^T \sum_s \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = 0$ with ${\boldsymbol{Q}^{(s)}}^T = \boldsymbol{c}^{(s)} {\boldsymbol{B}^{(s)}}^T \boldsymbol{W}$. With $\boldsymbol{Q} = \left[\boldsymbol{Q}^{(1)} \dots \boldsymbol{Q}^{(n)}\right]$, which has a full column rank, the n additional constraints are

$$\boldsymbol{Q}^T \boldsymbol{P}^T \sum_s \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = 0 \tag{14}$$

As this constraint is obviously satisfied for the solution of the problem (for which $\boldsymbol{P}^T \sum_s \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = 0$), it can be prescribed to any subdomain, even if it

possesses a free boundary. Doing this avoids the detection of the subdomains that have a singular Neumann problem during preconditioning.

To enforce these constraints, an additional multiplier $\boldsymbol{\gamma}$ enriches the previous ones [31,32], and satisfies automatically the admissibility condition: $\boldsymbol{\lambda} + \boldsymbol{PQ\gamma}$ replaces $\boldsymbol{\lambda}$ in the previous FETI-I algorithm (4), (3), (5).

$\boldsymbol{\gamma}$ can be interpreted as a uniform pressure value per subdomain. Keeping it in a coarse problem allows the solution to satisfy the additional constraints at each iteration, and so, is expected to improve convergence.

With the equations (14), the overall problem to be solved is:

$$\begin{bmatrix} \boldsymbol{F} & \boldsymbol{FPQ} & -\boldsymbol{G} \\ \boldsymbol{Q}^T\boldsymbol{P}^T\boldsymbol{F} & \boldsymbol{L} & \boldsymbol{0} \\ -\boldsymbol{G}^T & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{Q}^T\boldsymbol{P}^T\boldsymbol{d} \\ -\boldsymbol{e} \end{bmatrix} \qquad \text{with} \qquad \boldsymbol{L} = \boldsymbol{Q}^T\boldsymbol{P}^T\boldsymbol{FPQ}$$

To build the corresponding algorithm, the following ways are equivalent: using the FETI2 framework [32] with the interpretation of an additional coarse space correction with the regular matrix $\boldsymbol{L}$, or, as it is done herein, condensing the new unknowns $\boldsymbol{\gamma}$ on the other ones to get:

$$\begin{bmatrix} \boldsymbol{F}^\star & -\boldsymbol{G} \\ -\boldsymbol{G}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d}^\star \\ -\boldsymbol{e} \end{bmatrix}$$

with $\boldsymbol{F}^\star = \boldsymbol{F} - \boldsymbol{FPQL}^{-1}\boldsymbol{Q}^T\boldsymbol{P}^T\boldsymbol{F}$ and $\boldsymbol{d}^\star = \boldsymbol{d} - \boldsymbol{FPQL}^{-1}\boldsymbol{Q}^T\boldsymbol{P}^T\boldsymbol{d}$, and apply the previous FETI-I algorithm to this new problem. the overall algorithm is the same as in Table 1; the difference in the implementation is the need for the additional coarse problem (with matrix $\boldsymbol{L}$) within the matrix-vector product $\boldsymbol{F}^\star\boldsymbol{p}$. The preconditioning step allows to apply the eventually singular (because incompressible) Dirichlet preconditioner to the residual: if s is a subdomain with all of its boundary subjected to a prescribed displacement $\boldsymbol{v}^{(s)} = {\boldsymbol{B}^{(s)}}^T \boldsymbol{W}\boldsymbol{P}^T \sum_s \boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)}$, the Dirichlet problem to be solved on this subdomain is:

$$\boldsymbol{K}^{(s)}_{H_{ii}} \begin{bmatrix} \boldsymbol{v}_i^{(s)} \\ \boldsymbol{p}^{(s)} \end{bmatrix} = - \begin{bmatrix} \widetilde{\boldsymbol{K}}^{(s)}_{ib} \\ {\boldsymbol{C}^{(s)}_b}^T \end{bmatrix} \boldsymbol{v}_b^{(s)}$$

If $\boldsymbol{K}^{(s)}_{H_{ii}}$ is singular, its kernel is exactly a uniform pressure $\begin{bmatrix} \boldsymbol{0} & {\boldsymbol{p}_1^{(s)}}^T \end{bmatrix}^T$. As the right hand side is orthogonal to this kernel as soon as the augmentation condition is satisfied, the solution is then defined up to an undetermined uniform pressure field per subdomain. This field is not required to be determined: first, only forces on the boundary are extracted from the preconditioned solution

and, second, this correction will be automatically added in the augmented coarse problem within the multiplier $\boldsymbol{\gamma}$.

4.2 Augmentation of FETI-DPI algorithm for incompressible cases

The same procedure can be applied to the FETI-DPI algorithm. Briefly, the additional constraint is this time: $\boldsymbol{Q}^T \sum_s \boldsymbol{B}_r^{(s)} \boldsymbol{u}_r^{(s)} = 0$ with $\boldsymbol{Q}^{(s)^T} = \boldsymbol{c}_r^{(s)} \boldsymbol{B}_r^{(s)^T} \boldsymbol{W} = \boldsymbol{p}_1^{(s)^T} \boldsymbol{C}_r^{(s)} \boldsymbol{B}_r^{(s)^T} \boldsymbol{W}$ because at the current iteration, the correction to the solution is performed with no residual on coarse nodes and so, the additional constraint is only dealing with r dof. The corresponding problem to be solved is:

$$\begin{bmatrix} \boldsymbol{F}_{rr} & \boldsymbol{F}_{rr}\boldsymbol{Q} & \boldsymbol{F}_{rc} \\ \boldsymbol{Q}^T\boldsymbol{F}_{rr} & \boldsymbol{Q}^T\boldsymbol{F}_{rr}\boldsymbol{Q} & \boldsymbol{Q}^T\boldsymbol{F}_{rc} \\ \boldsymbol{F}_{cr} & \boldsymbol{F}_{cr}\boldsymbol{Q} & -\boldsymbol{K}_{cc}^\star \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_r \\ \boldsymbol{\gamma} \\ \boldsymbol{u}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{d}_r \\ \boldsymbol{Q}^T\boldsymbol{d}_r \\ -\boldsymbol{f}_c^\star \end{bmatrix}$$

with a condensation of the coarse dof $\begin{bmatrix} \boldsymbol{\gamma}^T & \boldsymbol{u}_c^T \end{bmatrix}^T$ on $\boldsymbol{\lambda}_r$:

$$\boldsymbol{F}^\star \left(\boldsymbol{F}_{rr} - \begin{bmatrix} \boldsymbol{F}_{rr}\boldsymbol{Q} & \boldsymbol{F}_{rc} \end{bmatrix} \bar{\boldsymbol{K}}_{cc}^{\star\,-1} \begin{bmatrix} \boldsymbol{Q}^T\boldsymbol{F}_{rr} \\ \boldsymbol{F}_{cr} \end{bmatrix} \right) \boldsymbol{\lambda}_r = \boldsymbol{d}_r - \begin{bmatrix} \boldsymbol{F}_{rr}\boldsymbol{Q} & \boldsymbol{F}_{rc} \end{bmatrix} \bar{\boldsymbol{K}}_{cc}^{\star\,-1} \begin{bmatrix} \boldsymbol{Q}^T\boldsymbol{d}_r \\ -\boldsymbol{f}_c^\star \end{bmatrix}$$

$$\text{with} \qquad \bar{\boldsymbol{K}}_{cc}^\star = \begin{bmatrix} \boldsymbol{Q}^T\boldsymbol{F}_{rr}\boldsymbol{Q} & \boldsymbol{Q}^T\boldsymbol{F}_{rc} \\ \boldsymbol{F}_{cr}\boldsymbol{Q} & -\boldsymbol{K}_{cc}^\star \end{bmatrix}$$

The augmented coarse space is then a mixed coarse problem whose unknowns are both the corner dof and the uniform pressure values.

To check if this coarse problem is LBB stable, in particular to avoid spurious pressure oscillations, let us consider the condensation of $\boldsymbol{K}_{cc}^\star$ on coarse pressure dof $\boldsymbol{\gamma}$: $\boldsymbol{Q}^T\boldsymbol{F}_{rr}\boldsymbol{Q} + \boldsymbol{Q}^T\boldsymbol{F}_{rc}\bar{\boldsymbol{K}}_{cc}^{\star\,-1}\boldsymbol{F}_{cr}\boldsymbol{Q} = \boldsymbol{Q}^T\boldsymbol{F}^\star\boldsymbol{Q}$. If $\boldsymbol{F}^\star$ is SPD (especially when there is no redundant multiplier), this condensed matrix is symmetric positive. Moreover, if we consider $\boldsymbol{\gamma}^T\boldsymbol{Q}^T\boldsymbol{F}^\star\boldsymbol{Q}\boldsymbol{\gamma} = 0$, we obtain $\boldsymbol{Q}\boldsymbol{\gamma} = \boldsymbol{0}$. As $\boldsymbol{Q}$ is full column rank, one gets $\boldsymbol{\gamma} = \boldsymbol{0}$, so the condensed matrix is SPD too. Its minimum eigenvalue is then bounded away from zero, and the coarse matrix problem is stable.

As already mentioned, the augmentation consists in prescribing a constraint that will be satisfied at convergence, because it enforces the residual to be orthogonal to a wider subspace: $\boldsymbol{Q}^T \sum_s \boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)} = 0$. The present augmentation can obviously been applied to nearly incompressible cases as well, without any modification of the algorithm.

Now, the different possibilities are the following, when the material is nearly incompressible or purely incompressible:

- for the non-augmented algorithms FETI-I and FETI-DPI: the preconditioner can be compressible or nearly incompressible;
- for the augmented algorithms denoted with FETI-IA and FETI-DPIA: the preconditioner can be compressible, nearly incompressible or incompressible.

The compressible Dirichlet preconditioner has already been tested for FETI-I and FETI-DPI algorithms. To compare with FETI-IA and FETI-DPIA, the results have already been given in Tables 3 and 4. Obviously, the additional constraints improve convergence. The results are very similar to the compressible case, in terms of number of iterations.

As already mentioned, the treatment of compressible problems with the proposed algorithms is not efficient. For nearly incompressible problems and incompressible problems, the performance in terms of the number of iterations with different preconditioners can be found in Table 5. The tested problem is the die problem with $n = 48$ subdomains, $h/H = 1/6$ and for $\tilde{\nu} = 0.3$. As it is not clearly interesting to use an incompressible preconditioner for nearly incompressible problems or nearly incompressible preconditioners for incompressible problems, the corresponding results are not reported.

Clearly, in each case, the convergence rate in increased when switching from FETI to FETI-DP, and with the augmentation of the algorithms. Concerning the influence of the preconditioner, the situation is not so clear: the compressible preconditioner (due to its regularity) is more efficient than other ones. Even with purely incompressible problems with augmentation, the use of compressible preconditioner is to be recommended. So, this is now the choice we made, in particular in the next section.

Table 5
Comparison of convergences for different algorithms (NI = nearly incompressible, I = incompressible, C = compressible)

		NI problems			I problem
		$\nu = 0.49$	$\nu = 0.499$	$\nu = 0.4995$	$\nu = 0.5$
NI preconditioner	FETI-I	49	77	80	
NI preconditioner	FETI-IA	37	39	39	
NI preconditioner	FETI-DPI	52	68	69	
NI preconditioner	FETI-DPIA	22	23	23	
I preconditioner	FETI-IA				40
I preconditioner	FETI-DPIA				25
C preconditioner	FETI-I	43	47	47	47
C preconditioner	FETI-IA	34	36	37	37
C preconditioner	FETI-DPI	31	34	34	34
C preconditioner	FETI-DPIA	21	22	22	22

5 Implementation and complexity analysis

A complexity analysis is a way to estimate the costs of algorithms; nevertheless it should be used only for an estimation of the main trends rather than a proof of efficiency. It is not able to measure the effective parallel potentialities of an algorithm, nor its ability to take into account a particular architecture of a parallel machine to tune the algorithm (as for unlooping, vectorizing or locality of datas) and the cost of data access in memory. On the positive side, the floating point operation count does not interfere with the performance of a particular operating system and compiler; it is close to the algorithmic part and the main choices done when implementing it (full entry matrix, skyline solver...). It provides also information on how the cost scales with respect to the parameters of the problem solved. For instance, we choose to consider problems with a regular decomposition into subdomains ($m \times m$ subdomains in 2D, and $m \times m \times m$ subdomains in 3D); and regular meshes for each subdomain. This allows to determine the structure of the sparse matrices involved, independently of the dof renumbering scheme used.

We considered herein banded symmetric matrices, and we assumed that the renumbering scheme, the solver for local subdomain non SPD problems, and the eventual pivoting scheme, see [33], does not destroy the bandwidth. This assumption is optimistic for large number of dof, as the probability to detect a null pivot that does not correspond to a rigid body motion, increases. In particular, when needed, the detection of rigid body motions in subdomains

will be preferably performed with a geometric inspection [34,35].

5.1 Matrix structures

In all of the following cost estimations, we separated the initialization phase of the various algorithms (computation of elementary matrices; local factorizations; computation of coarse matrix; corresponding right hand side, and its factorization; factorization of the preconditioner) and the cost of an iteration (local solves for matrix-vector product of conjugate gradient, and for preconditioner; coarse solves; global operation on interface fields).

The assembling of matrices and vectors, or disassembling of global vectors (at element or subdomain levels) have been neglected as soon as they involve boolean matrices $\boldsymbol{B}^{(s)}, \boldsymbol{B}_c^{(s)}, \boldsymbol{B}_r^{(s)} \ldots$

For FETI-I, the coarse space matrix $\boldsymbol{G}^T\boldsymbol{G}$ is symmetric and possesses the connectivity of neighboring subdomains. For FETI-DPI, $\boldsymbol{K}_{cc}^{\star}$ has the same connectivity as a mesh with corner nodes, and subdomains only connected with these nodes as superelements. Concerning augmented algorithms, the structure of the coarse problem is more complex because superlements are also connected by an 'internal' dof corresponding to the coarse pressure $\boldsymbol{\gamma}$. Nevertheless, it still possesses a sparse pattern [36], and with the same assumption as for the subdomain stiffness factorization, the costs are estimated with the corresponding banded matrix storage, and costs are estimated accordingly. For FETI-IA, the second level coarse problem matrix $\boldsymbol{L} = \boldsymbol{Q}^T\boldsymbol{P}^T\boldsymbol{F}\boldsymbol{P}\boldsymbol{Q}$ is full and symmetric, due to the long distance coupling of the projection operator. For the tested structure of the die problem (with $h/H = 1/6$ and $n = 48$ subdomains) the corresponding morse pattern is also plotted on the previous expression.

5.2 Implementation choices

Concerning implementation efficiency, the idea is to store the information (rather than recompute it on-the-fly several times when iterating) as soon as it concerns only fields on the boundary that need only to be stored distributedly at the subdomain level.

For instance, for FETI-DPI, during initialization phase, the vectors

$$\boldsymbol{V}^{(s)} = {\boldsymbol{K}_{H_{rr}}^{(s)}}^{-1} \begin{bmatrix} \boldsymbol{K}_{cr}^{(s)} \\ \boldsymbol{C}_c^{(s)} \end{bmatrix}$$

are precomputed with the resolution of as many local systems as there are corner nodes touching the subdomain s (and stored only on the boundary dof b for iteration phase). Then, $\boldsymbol{K}_{cc}^{(s)\star} = \boldsymbol{K}_{cc}^{(s)} - \left[\boldsymbol{K}_{cr}^{(s)}\ \boldsymbol{C}_c^{(s)^T}\right] \boldsymbol{V}^{(s)}$, $\boldsymbol{f}_c^{(s)\star} = \boldsymbol{f}_c^{(s)} - \boldsymbol{V}^{(s)^T} \boldsymbol{f}_r^{(s)}$ and for matrix vector product $\boldsymbol{F}^\star \boldsymbol{\lambda}_r = \boldsymbol{F}_{rr}\boldsymbol{\lambda}_r + \boldsymbol{F}_{rc}\boldsymbol{K}_{cc}^{\star\,-1}\boldsymbol{F}_{cr}\boldsymbol{\lambda}_r$, with:

$$\boldsymbol{F}_{rr}^s \boldsymbol{\lambda}_r = \left[\boldsymbol{B}_r^{(s)}\ \boldsymbol{0}\right] \boldsymbol{K}_{H_{rr}}^{(s)\,-1} \begin{bmatrix} \boldsymbol{\lambda}_r^s \\ \boldsymbol{0} \end{bmatrix} \qquad \text{with} \qquad \boldsymbol{\lambda}_r^s = \boldsymbol{B}_r^{(s)^T} \boldsymbol{\lambda}_r$$

$$\boldsymbol{F}_{cr}^s \boldsymbol{\lambda}_r = \boldsymbol{B}_c^{(s)^T} \boldsymbol{V}^{(s)^T} \boldsymbol{\lambda}_r^s \qquad\qquad \boldsymbol{u}_c = \boldsymbol{K}_{cc}^{\star\,-1} \boldsymbol{F}_{rc} \boldsymbol{\lambda}_r = \boldsymbol{K}_{cc}^{\star\,-1} \sum_s \boldsymbol{F}_{rc}^s \boldsymbol{\lambda}_r$$

$$\boldsymbol{F}_{rc}\boldsymbol{u}_c = \sum_s \boldsymbol{B}_r^{(s)} \boldsymbol{V}^{(s)} \boldsymbol{u}_c^{(s)} \qquad \text{with} \qquad \boldsymbol{u}_c^{(s)} = \boldsymbol{B}_c^{(s)} \boldsymbol{u}_c$$

For FETI-DPIA, the entries of $\boldsymbol{F}_{cr}\boldsymbol{Q}$ are computed as follows: the column j is

$$\sum_s \boldsymbol{B}_c^{(s)^T} \boldsymbol{V}^{(s)^T} \begin{bmatrix} \boldsymbol{B}_r^{(s)^T} \boldsymbol{Q}^{(j)} \\ \boldsymbol{0} \end{bmatrix}$$

and is computed only with the contributions of the subdomains s neighbors of the subdomain j (due to the term $\boldsymbol{B}_r^{(s)^T}\boldsymbol{B}_r^j$ in $\boldsymbol{B}_r^{(s)^T}\boldsymbol{Q}^{(j)}$, the other contributions are null). The term on line i and column j of $\boldsymbol{Q}^T\boldsymbol{F}_{rr}\boldsymbol{Q}$ is therefore:

$$\sum_s \left[\boldsymbol{Q}^{(i)^T} \boldsymbol{B}_r^{(s)}\ \boldsymbol{0}\right] \boldsymbol{A}^{sj} \qquad \text{with} \qquad \boldsymbol{A}^{sj} = \boldsymbol{K}_{H_{rr}}^{(s)\,-1} \begin{bmatrix} \boldsymbol{B}_r^{(s)^T} \boldsymbol{Q}^{(j)} \\ \boldsymbol{0} \end{bmatrix}$$

Similarly, $\boldsymbol{A}^{sj}$ is stored on a subdomain s database only for neighbors j, and only for local boundary dof b. Then

$$\bar{\boldsymbol{F}}^\star \boldsymbol{\lambda}_r = \boldsymbol{F}_{rr}\boldsymbol{\lambda}_r - \left[\boldsymbol{F}_{rr}\boldsymbol{Q}\ \boldsymbol{F}_{rc}\right] \bar{\boldsymbol{K}}_{cc}^{\star\,-1} \begin{bmatrix} \boldsymbol{Q}^T\boldsymbol{F}_{rr} \\ \boldsymbol{F}_{cr} \end{bmatrix} \boldsymbol{\lambda}_r$$

$$\text{with} \quad \boldsymbol{F}_{rr}\boldsymbol{Q}\boldsymbol{\mu} = \sum_s \left[\boldsymbol{B}_r^{(s)}\ \boldsymbol{0}\right] \sum_j \boldsymbol{A}^{sj}\boldsymbol{\mu}_j$$

and similar computations of $\boldsymbol{F}_{rr}\boldsymbol{\lambda}_r$ and $\boldsymbol{F}_{cr}\boldsymbol{\lambda}_r$.

5.3 Numerical Results

With the previous scenario, the cost estimation can be plotted with respect to the parameters H and h/H for each of the proposed algorithms. In 2D on

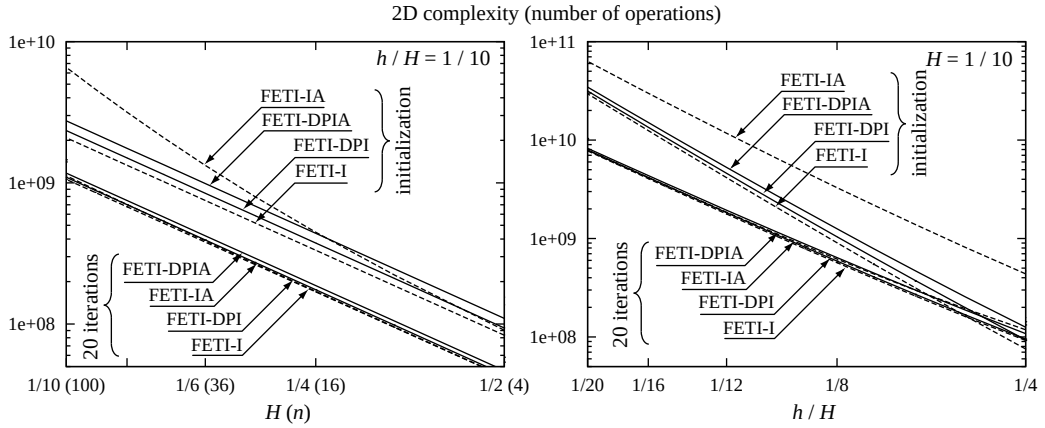

Fig. 8. Complexity estimation for 2D problems

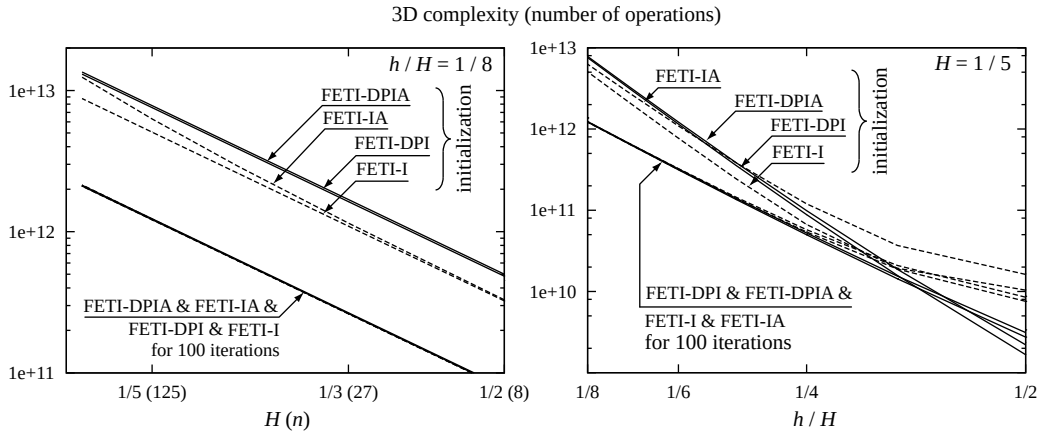

Fig. 9. Complexity estimation for 3D problems

Figure 8, the initialization phase has been separated from the iterations, and in order to have similar costs, 20 iterations are considered. FETI-I and FETI-IA are plotted with dashed lines, while DP versions are plotted with plain lines. To give an idea of the behavior of the methods, a 3D cost estimation has been done as well and the results are plotted on Figure 9 for the initialization phase, and for 100 iterations.

Both for the initialization and iteration phases, we can consider as a first step that the costs of the previous algorithms arise from different sources:

- the coarse problem: the cost is related first to the number of subdomains, and second, to the structure of the coarse problem itself;
- the subdomain problems: the cost is mainly related to the size of the local problems on subdomains; it is driven by the coefficient h/H;
- additive operations like dot products on a global interface field, which can be performed at subdomain level with few assembly or disassembly operations.

Usually, the costs of the third category are negligible when compared to the

Table 6
Comparison of complexities (in Mflop) for different algorithms

	number of iterations	init. cost	iter. cost	total cost
	n_{it}	c_1	c_2	$c_1 + c_2 n_{it}$
FETI-I	47	151	6.24	445
FETI-IA	37	395	6.69	642
FETI-DPI	34	180	6.51	401
FETI-DPIA	22	225	7.19	383

previous ones.

If the amount of operations that are done at the subdomain level is sufficiently large when compared to the cost related to the coarse problem (i.e. if h/H is sufficiently small), both the initialization and iteration costs have similar evolutions when h/H decreases. Due to the higher fill-in in the coarse problem of FETI-IA, the initialization cost grows very rapidly when the number of subdomain increases. Therefore, FETI-IA is probably not the most efficient approach to use, for the cost estimation, as well as for the complexity of implementation.

Concerning iteration phase, the cost is almost the same for all of the approaches, especially for 3D case. Concerning initialization phase, the cost increases from FETI-I, to FETI-DPI, and up to FETI-DPIA. To compare these approaches, let us recall the convergence results for 2D incompressible case, with compressible preconditioner, of Table 5. Table 6 evaluates the relative costs with the previous complexity analysis ($n = 48$, $H = n^{-0.5}$, $h/H = 1/6$). It appears that for the proposed test case, using dual-primal version is more efficient, as when an augmentation is used. Moreover, as it has been previously highlighted in original papers on FETI-DP [21], it consumes less iterations and is more robust than the previous FETI and FETI2 versions.

6 Summary and conclusions

The framework of standard FETI and FETI-DP versions with or without augmentation has not been modified. To do so, we derived specific implementation for the mixed local problems per subdomain, without changing gluing condition between them, thanks to a discontinuous pressure discretization. This feature let us to expect an easy integration of the extension for incompressible or nearly incompressible material with other developments of the FETI method, especially for dynamics problems [37] and contact problems [38].

Without augmentation of the algorithm, optimality and numerical scalability have been obtained in 2D. They are expected to pertain in 3D for the FETI-I method, but not for the FETI-DPI without augmentation, as it has been previously noticed for compressible problems [28]. If the proposed augmentation (only one constraint per subdomain) is not sufficient enough to recover the numerical scalability in 3D, there will be no difficulty to add the previously developed augmentations [28], to the specific incompressible one proposed herein. With this specific augmentation, the rigid body motion for FETI-I (and corner modes for FETI-DPI) as well as a uniform pressure per subdomain is propagated globally at each iteration. The performances obtained are then similar to the one obtained for compressible problems.

Of course, though the use of uniform pressure for the augmentation is required in the case of purely incompressible behavior, other additional stress modes may be used, for instance the generalized Trefftz basis.

The increase in size of the coarse problem leads to improved convergence rate, but increases also the cost of the coarse problem (a complexity analysis is required to predict the gain in performances) and impairs the parallel part of the implementation. With a large number of subdomains, a particular care of the coarse problem implementation is required as it is a bottleneck for the parallelization.

References

[1] I. Babuska, M. Suri, Locking effects in the finite element approximation of elasticity problems, Numerische Mathematik 62 (1992) 439–463.

[2] R. Bramley, D. Pelletier, Iterative methods and formulations for incompressible FEM flow solvers, in: Proceedings of the 34th Aerospace Sciences Meeting and Exhibit, 1996, aIAA Paper 96-0890.

[3] A. C. Bauer, A. K. Patra, Performance of parallel preconditioners for adaptive *hp* FEM discretization of incompressible flows, Communications in Numerical Methods in Engineering 18 (2002) 305–313.

[4] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Vol. 15 of Computational Mathematics, Springer, 1991.

[5] R. Stenberg, M. Suri, Mixed *hp* finite element methods for problems in elasticity and Stokes flow, Numerische Mathematik 72 (1996) 367–389.

[6] P. Le Tallec, Domain decomposition methods in computational mechanics, in: Computational Mechanics Advances, Vol. 1, North-Holland, 1994.

[7] C. Farhat, F.-X. Roux, Implicit parallel processing in structural mechanics, in: J. T. Oden (Ed.), Computational Mechanics Advances, Vol. 2, North-Holland, 1994.

[8] L. F. Pavarino, A. Toselli (Eds.), Recent Developments in Domain Decomposition Methods, no. 23 in Lecture Notes in Computational Science and Engineering, Springer, 2002.

[9] A. Klawonn, L. F. Pavarino, A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems, Numerical Linear Algebra with Applications 7 (2000) 1–25.

[10] J. Mandel, Balancing domain decomposition, Communications in Applied Numerical Methods 9 (1993) 233–241.

[11] L. C. Cowsar, J. Mandel, M. F. Wheeler, Balancing domain decomposition for mixed finite elements, Mathematics of Computation 64 (1995) 989–1015.

[12] P. Le Tallec, A. Patra, Non-overlapping domain decomposition methods for adaptive *hp* approximations of the Stokes problem with discontinuous pressure fields, Computer Methods in Applied Mechanics and Engineering 145 (1997) 361–379.

[13] L. F. Pavarino, O. B. Widlund, Balancing Neumann-Neumann methods for incompressible Stokes equations, Communication on Pure and Applied Mathematics 55 (2002) 302–335.

[14] F. Léné, C. Rey, Some strategies to compute elastomeric lamified composite structures, Composite Structures 54 (2001) 231–241.

[15] P. W. Grant, M. F. Webster, X. Zhang, Coarse grain parallel finite element simulations for incompressible flows, International Journal for Numerical Methods in Engineering 41 (1998) 1321–1337.

[16] O. Axelsson, V. A. Barker, M. Neytcheva, B. Polman, Solving the Stokes problem on a massively parallel computer, Mathematical Modelling and Analysis 6 (2001) 7–27.

[17] C. Calgaro, J. Laminie, On the domain decomposition method for the generalized Stokes problem with continuous pressure, Numerical Methods for Partial Differential Equations 16 (2000) 84–106.

[18] J. H. Bramble, J. E. Pasciak, A domain decomposition technique for Stokes problems, Applied Numerical Mathematics 6 (1990) 251–261.

[19] M. Ainsworth, S. Sherwin, Domain decomposition preconditioners for p and *hp* finite element approximations of Stokes equations, Computer Methods in Applied Mechanics and Engineering 175 (1999) 243–266.

[20] C. Farhat, F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, International Journal for Numerical Methods in Engineering 32 (1991) 1205–1227.

[21] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, D. Rixen, FETI-DP: a dual-primal unified FETI method - part I: a faster alternative to the two-level FETI method, International Journal for Numerical Methods in Engineering 50 (7) (2001) 1523–1544.

[22] P. Ladevèze, P. Marin, J. P. Pelle, G. L. Gastine, Accuracy and optimal meshes in finite element computation for nearly incompressible materials, Computer Methods in Applied Mechanics and Engineering 94 (1992) 303–315.

[23] V. Girault, P. A. Raviart, Finite Element Methods for Navier-Stokes Equations, Vol. 5 of Computational Mathematics, Springer, 1986.

[24] O. Pironneau, Finite Element Methods for Fluids, Wiley, 1989.

[25] L. R. Herrmann, Elasticity equations for incompressible and nearly incompressible materials by a variational theorem, AIAA Journal 3 (1965) 1896–1900.

[26] O. C. Zienkiewicz, S. Qu, R. L. Taylor, S. Nakazawa, The patch test for mixed formulations, International Journal for Numerical Methods in Engineering 23 (1986) 1873–1883.

[27] E. N. Dvorkin, On the convergence of incompressible finite element formulations, Engineering Computations 18 (3-4) (2001) 539–556.

[28] C. Farhat, M. Lesoinne, K. Pierson, A scalable dual-primal domain decomposition method, Numerical Linear Algebra with Applications 7 (2000) 687–714.

[29] M. Lesoinne, A FETI-DP corner selection algorithm for three-dimensional problems, in: Proceedings of the 14th International Conference on Domain Decomposition Methods, Mexico, 2002.

[30] A. Klawonn, O. B. Widlund, FETI and Neumann-Neumann iterative substructuring methods: connections and new results, Communications in Pure and Applied Mathematics 54 (2001) 57–90.

[31] C. Farhat, P. S. Chen, F. Risler, F.-X. Roux, A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers, International Journal for Numerical Methods in Engineering 42 (1998) 257–288.

[32] C. Farhat, K. Pierson, M. Lesoinne, The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems, Computer Methods in Applied Mechanics and Engineering 184 (2000) 333–374.

[33] I. S. Duff, The solution of augmented systems, in: D. F. Griffiths, G. A. Watson (Eds.), Numerical Analysis 1993: Proceedings of the 15th Dundee Conference, Pitman Research Notes in Mathematics, Longman Scientific and Technical, 1994.

[34] C. Farhat, M. Géradin, On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures, International Journal for Numerical Methods in Engineering 41 (4) (1998) 675–696.

[35] M. Papadrakakis, Y. Fragakis, An integrated geometric-algebraic method for solving semi-definite problems in structural mechanics, Computer Methods in Applied Mechanics and Engineering 190 (2001) 6513–6532.

[36] K. Pierson, A family of domain decomposition methods for the massively parallel solution of computational mechanics problems, Phd thesis, University of Colorado at Boulder, Aerospace Engineering (2000).

[37] C. Farhat, P. S. Chen, J. Mandel, A scalable Lagrange multiplier based domain decomposition method for implicit time-dependent problems, International Journal for Numerical Methods in Engineering 38 (1995) 3831–3854.

[38] D. Dureisseix, C. Farhat, A numerically scalable domain decomposition method for the solution of frictionless contact problems, International Journal for Numerical Methods in Engineering 50 (2001) 2643–2666.