UNIVERSITÉ DE LIÈGE
FACULTÉ DES SCIENCES APPLIQUÉES

Substructuring and Dual Methods in Structural Analysis

Daniel J. Rixen

Aspirant F.N.R.S.
MSc in Aerospace Vehicle Design
Ingénieur Civil Electricien-Mécanicien (Aérospatiale)

Thèse présentée en vue de l'obtention du titre de
Docteur en Sciences Appliquées
de l'Université de Liège

Janvier 1997

À Yann, Gilles et Bettina

Remerciements

Une thèse représente une étape importante dans le cheminement scientifique d'un chercheur. Elle consacre une recherche de plusieurs années et en même temps révèle l'infinie ignorance qu'il reste à éclairer.

Mener à bien un tel travail demande plus que beaucoup de café et de temps de calcul. Je voudrais remercier toutes les personnes qui m'ont soutenu au cours de ces années, à commencer par les Professeurs Michel Géradin de l'Université de Liège et Charbel Farhat de l'Université du Colorado à Boulder dont la passion pour la recherche fut un formidable encouragement. Qu'ils soient remerciés ici pour les discussions toujours édifiantes et les conseils bienveillants.

Je suis reconnaissant à François-Xavier Roux (ONERA, France) et Patrick LeTallec (INRIA, France) pour leurs commentaires avisés ainsi qu'à François Hemez et toute l'équipe du Professeur Farhat pour leur collaboration lors de mes visites à Boulder.

Je remercie également les membres du jury, le Professeur P. Beckers, Président du jury, les Professeurs J. Donea, P. de Marneffe et S. Cescotto (Université de Liège), Ngyuen V. Hien (FUNDP), R. Keunings (UCL) et D. Roose (KUL) qui ont eu l'amabilité de s'intéresser à mon travail. Nul doute que leur analyse critique et leurs commentaires judicieux permettront d'améliorer mon travail futur.

Pour l'ambiance agréable et joyeuse qui a régné au LTAS durant mon travail de thèse, j'aimerais exprimer toute ma reconnaissance aux membres du service de Dynamique des Structures et de Vibrations et Identifications des Structures. Merci à Danielle, Albert, Bao, Igor et Pierre pour les échanges scientifiques pleins d'intérêt, et à Cécile, notre secrétaire tant appréciée.

Enfin, il est important de souligner le support du Fonds National de la Recherche Scientifique sans qui ce travail n'aurait sans doute pu être mené à bien.

Contents

List of Figures

List of Tables

List of Algorithms

Résumé

Dans la plupart des domaines de la science moderne, le besoin de puissance de calcul ne fait que s'amplifier avec l'avènement de nouvelles techniques d'analyse et la complexification grandissante des modèles utilisés, par exemple en météorologie, en génie chimique ou en imagerie. Le domaine de l'analyse des structures en est un autre exemple : alors que les ingénieurs se sont longtemps contentés de modèles simples et souvent analytiques, l'apparition des ordinateurs a suscité la mise au point de nouveaux algorithmes de calcul et des formulations nouvelles pour les systèmes mécaniques sont apparues. Aujourd'hui, résoudre des problèmes complexes tels que l'analyse non-linéaire du déploiement d'une antenne de satellite ou le calcul couplé du comportement aéroélastique d'une aile d'avion sont choses communes. En même temps, les méthodes classiques de résolution atteignent souvent leurs limites pour des modèles impliquant parfois plus d'un million de variables.

Deux problèmes importants se posent aujourd'hui aux ingénieurs désireux d'utiliser des modèles complexes. Le premier provient de la difficulté de construire le modèle numérique. En analyse par éléments finis, par exemple, définir un maillage comprenant plusieurs dizaines de milliers de noeuds implique que le modèle soit maillé par des techniques automatiques et, éventuellement, par différentes équipes travaillant sur un même projet. Le second est de résoudre les systèmes d'équations en un temps de calcul raisonnable.

Afin de faciliter la création d'un maillage et de pouvoir le modifier sans trop d'efforts, des techniques particulières sont apparues permettant l'assemblage de différentes parties d'une même structure avec éventuellement des maillages non-compatibles. De même, le besoin de réduire le coût de calcul de ces modèles a amené les ingénieurs à mettre au point des techniques de réduction du nombre de variables. En calcul des structures, les techniques les plus prisées font appel aux méthodes de sous-structuration dont le but est de formuler des approximations pour des sous-ensembles structuraux par un nombre réduit de variables. Enfin, des ordinateurs de plus en plus puissants sont utilisés. Des moyens de calculs très sophistiqués sont aujourd'hui accessibles et en particulier certains super-ordinateurs atteignent une grande puissance de calcul en utilisant plusieurs processeurs pour une même tâche. Ces ordinateurs, appelés parallèles, ne délivrent leur pleine puissance que si un bon usage peut être fait de leurs différents processeurs. Une méthode efficace pour utiliser rationnellement les machines multi-processeurs est de subdiviser le domaine de modélisation en sous-domaines qui sont pris en charge par des processeurs différents.

Pour les raisons mentionnées ci-dessus, les techniques de sous-structuration sont au centre des préoccupations de bien des chercheurs et ingénieurs de bureaux d'études. Notre travail a pour but de contribuer au développement des méthodes appropriées pour traiter l'assemblage de sous-structures, pour construire des modèles avec un nombre réduit de variables et pour résoudre efficacement des systèmes de grandes tailles sur ordinateurs parallèles.

Dans la première partie du travail, nous discutons les méthodes permettant d'assembler différentes sous-structures et, en particulier, nous étudions celles permettant de traiter les maillages non-compatibles. Nous présentons les méthodes mathématiques et numériques relatives aux systèmes contraints par les conditions de compatibilité d'interface (chapitre 2). Nous étudions alors au chapitre 3 plus en détail la méthode des multiplicateurs de Lagrange et nous montrons comment elle définit une formulation hybride du problème d'interface: les sous-structures sont définies en termes de degrés de liberté de déplacement alors que le recollement des interfaces est exprimé par un problème aux forces de contraintes sur les interfaces. Dans le chapitre 4, cette approche est mise à profit pour définir une nouvelle technique de réduction modale du problème d'interface plus avantageuse que les méthodes de réduction spectrale basées sur une formulation primale (Complément de Schur).

Par la suite, utilisant des fonctions simples pour modéliser de manière approchée les contraintes d'interfaces, nous formulons un problème réduit bien approprié pour assembler avec un faible nombre de variables les sous-domaines d'une structure (chapitre 5). Cette technique est efficace pour les problèmes homogènes avec des solutions régulières mais apparaît mal adaptée pour des problèmes hétérogènes. Nous proposons aux chapitres 6 et 7 une nouvelle technique consistant à construire une solution fortement compatible à partir de solutions faiblement assemblées obtenues pour le système réduit. Cette méthode est basée sur une approche énergétique et définit une technique de lissage de la solution. La combinaison d'une formulation hybride réduite du problème d'interface et de notre technique de lissage forme un outil efficace pour assembler avec un faible nombre de variables d'interface des sous-structures hétérogènes, conformes ou non. Toutefois, pour des structures dont l'interface est fortement irrégulière telles que les treillis, il n'est pas possible de réduire efficacement le problème d'interface en utilisant des fonctions simples de distribution de contraintes. Nous proposons alors de construire l'espace d'approximation des contraintes d'interface de manière itérative en faisant usage des forces implicitement obtenues en définissant une solution lissée (chapitre 8). Cette technique itérative est appliquée avec succès au problème particulièrement difficile d'un treillis de barres et est en fait applicable à tout problème d'assemblage de sous-structures homogènes, hétérogènes, ayant des maillages conformes ou non.

L'algorithme itératif qui découle des techniques de réduction et de lissage est une généralisation de la méthode FETI (découpage de maillage et recollement des sous-domaines) développée pour la résolution par décomposition en sous-domaines de systèmes de grande taille. La méthode FETI s'applique donc à l'assemblage par multiplicateurs de Lagrange de maillages conformes obtenus par décomposition en sous-domaines et est particulièrement bien adaptée pour le calcul parallèle.

Dans la partie II de notre travail, nous abordons plus particulièrement les problèmes liés à la méthode FETI pour l'analyse des structures. Au chapitre 10, nous présentons succinctement les particularités du calcul parallèle en décrivant les machines multi-processeurs et en discutant des algorithmes adaptés à ces machines. Les méthodes de décomposition en sous-domaines sont alors considérées plus en détail. Elles consistent à traiter de manière indépendante et avec des algorithmes standards les sous-domaines d'une même structure et à résoudre le problème d'interface par des algorithmes itératifs. Cette approche nous paraît la mieux adaptée pour le calcul parallèle car elle permet de distribuer efficacement le travail aux multiples processeurs. De plus, la formulation par sous-domaine permet de construire un schéma d'itération efficace sur les variables d'interface grâce au fait que des préconditionneurs de grande qualité et parallèles par nature peuvent être construits à partir des opérateurs locaux. Dans ce travail, nous avons considéré la formulation duale du problème d'interface. Toutefois, la plupart des développements présentés ici sont directement transposables aux méthodes primales.

L'algorithme du Gradient Conjugué qui forme la base de la résolution par la méthode FETI est résumé au chapitre 11 et les spécificités liées au caractère particulier du problème dual d'interface sont discutées. En particulier, nous proposons une technique d'itération pour traiter des systèmes semi-définis et nous montrons comment modifier la méthode FETI afin de résoudre les problèmes de structures ayant des modes rigides globaux de déplacement.

Par la suite (chapitre 12), nous proposons une légère modification des préconditionneurs

classiques de la méthode FETI (Dirichlet et opérateur diagonalisé) en pondérant les contributions des sous-domaines par les multiplicités des interfaces afin d'assurer la cohérence entre ces préconditionneurs et leur interprétation mécanique. L'effet négatif du rapport de forme des sous-domaines sur la convergence des itérations pour le problème d'interface est analysé au chapitre 13. La nécessité d'utiliser le préconditionneur de Dirichlet apparaît dans divers cas pratiques tels que les problèmes de coques par exemple. Constatant le surcoût important lié à ce préconditionneur, nous développons un nouveau préconditonneur de type Dirichlet mais pour lequel l'équilibre interne des sous-domaines n'est satisfait qu'approximativement (chapitre 14). De cette façon, le préconditionneur de Dirichlet approché est plus efficace que le préconditionneur diagonalisé et implique un coût de calcul moindre que le préconditionneur de Dirichlet exact.

Nous abordons au chapitre 15 le sujet difficile du préconditionnement efficace des problèmes de structures hautement hétérogènes. Nous illustrons d'abord l'inefficacité des préconditionneurs classiques pour de telles structures et nous proposons une adaptation simple dans le cas particulier où tous les sous-domaines sont uniformes. La technique de lissage développée dans la première partie est utilisée afin de définir un cadre global pour construire des préconditionneurs adaptés aux cas hétérogènes. Nous construisons ainsi un préconditionneur de Dirichlet avec lissage incluant un problème de grille grossière dans lequel les interfaces sont considérées comme des macro-variables. La bonne efficacité de cette technique est démontrée sur plusieurs exemples, mais son coût calcul prohibitif nous amène à proposer des variantes moins onéreuses. En particulier, nous définissons une technique de lissage basée sur une grille grossière super-diagonalisée permettant de traiter efficacement, et pour un effort calcul mineur, les problèmes présentant de fortes hétérogénéités localisées au voisinage des interfaces et les problèmes de coques. Dans ce même chapitre, nous montrons également comment évaluer un résidu primal pour les itérations FETI.

Au chapitre 16, nous montrons comment la méthode FETI peut être appliquée pour l'analyse dynamique des structures (analyse des vibrations libres et calcul de la réponse harmonique et transitoire). Il apparaît que cette méthode est bien adaptée aux problèmes de la dynamique des structures si des techniques particulières sont utilisées. Enfin, nous terminons notre présentation en discutant des opportunités nouvelles qu'offrent les méthodes de décomposition en sous-domaines pour la résolution des problèmes non-linéaires. Une attention particulière est portée sur les stratégies de calcul permettant d'inclure dans la méthode FETI des contraintes non-linéaires (chapitre 17).

L'ensemble de ce travail démontre la grande souplesse des méthodes itératives pour l'analyse des structures et, en particulier, la méthode FETI apparaît comme une technique mature et hautement efficace pour remplacer les méthodes classiques de factorisation pour le calcul parallèle. Les méthodes proposées dans ce travail sont innovantes et mieux adaptées au contexte de l'analyse numérique moderne. Nul doute que les méthodes de décomposition en sous-domaines resteront encore longtemps un domaine de prédilection pour mathématiciens et, de par leur lien évident avec la mécanique des structures, une source d'inspiration pour les ingénieurs de demain.

Introduction

Science and engineering are undergoing some radical changes as far as modelization and numerical methods are concerned. Indeed, the need for ever more refined models and for rapid solution schemes is boosting the development of new problem formulations. Novel numerical techniques are imagined to handle the models reaching a level of complexity one wouldn't have dared to dream of only ten years ago. Among the scientific domains that have benefit the most of the ever increasing effectiveness of modelization and computing, we can list meteorology, chemistry, acoustics and image processing. Aerospace engineering is not the last field where tremendous research effort is going on in order to improve the quality of structural and fluid models and to provide engineers with adaptable and powerful computing means.

In structural analysis, two key issues keep busy a large community of engineers and mathematicians. The first concerns the study of discretization techniques that render more flexible the classical approaches such as finite element methods. Industrial finite element discretizations can include several thousands of nodes and handling such numerical models requires that it be easier to create the models and to modify part of the meshing. A general way to improve the flexibility of finite element modelization is to construct submodels of structural components so that they can be handled by different teams and so that automatic mesh generators can be applied to them. Providing then specific techniques that allow to assemble even non-matching grids greatly enhances the analysis capabilities for structural prototyping for instance.

Another problem is that very large models are usually built in order to perform detailed analyses and it is therefore not unusual to face the problem of solving systems of equations with several millions of variables. Obviously it is crucial for the engineer who analyses a prototype or for a meteorologist who prepares a weather forecast that a solution be found as fast as possible. To that purpose, one can on one hand reduce the model size by decreasing in some clever way the number of unknowns without deteriorating too much the accuracy of the analysis and on the other hand, one should apply efficient solution algorithms and run the problem on more powerful computers. A very efficient and well understood method for reducing the size of a model consists in subdividing the model and defining some approximate representation of the solution within each subpart. Decomposing a structural model was also found to result in a neat strategy for setting-up well defined subtasks suitable for parallel computing, namely for distributing the computational work on several processors in order to speed-up the solution process.

The two challenges described above are different and yet similar. Assembling different substructures created independently is related to *domain integration* whereas subdividing a model for building reduced models or for exploiting the power of parallel computers is seen as *domain decomposition*. Domain integration often requires connecting substructures with non-matching meshes while, for domain decomposition, the subdomains are created artificially by subdividing a

global mesh and the grids are thus matching.

The same assembly techniques can be used for both issues and therefore investigating the assembling problems within a unique framework is an exciting research subject. In particular, the mathematical formulations and numerical algorithms pertaining to substructuring can benefit from considering as equivalent the problem of gluing together independently modeled subparts and the problem of tearing apart global models for reduction purposes and for parallel computing. The most interesting point though is that the mechanical aspects underlying domain decomposition and domain integration are closely related. Improving our understanding of the structural mechanics of subdomain coupling and mastering the mathematical and algorithmic tools for efficiently handling the assembling problem is a critical challenge for modern engineering and computation.

Our presentation is therefore subdivided into two parts. In part I, we first discuss the substructuring methods related to the assembly of general substructured problems and the issues of domain integration and efficient model reduction for general structural analysis are next addressed. In part II, we investigate the more specific problem of solving large structural systems in parallel by domain decomposition.

Part I

Two-field hybrid method for the assembly of substructures

Chapter 1

Assembling substructures: an introduction

In modern scientific computation, the effort for setting-up computational models and for solving the associated equations has grown together with the ever increasing complexity and refinement of the computational models. Intricate finite element models are constructed in particular for structural analysis either by applying automatic mesh generators to basic parts or by subdividing the model and spreading it among the different design teams. Therefore, the assembling of substructures has become a crucial issue [128].

Substructuring is also applied when only part of a structure is modified and when we want to reuse the finite element model of the unchanged components. Typically in the design of an aircraft, the fuselage and the wings are generally meshed separately. The basic design then evolves into a family of aircrafts with a common wing while the fuselage is entirely redesigned. The setup of the structural model then generally requires the assembly of substructures with non-matching meshes along the common edges: the interfaces between the independently designed sub-parts are said to be non-conforming because the shape functions defining the finite elements along the interface do not satisfy the interface displacement compatibility since the nodal displacements are not coincident.

Non-matching grids also appear naturally if subparts of a model are sliding such as in contact problems. Another important example of assembly of non-matching finite element sub-models appears in adaptive refinement and in global/local analysis [7, 49, 54, 185]: in order to improve the accuracy of the solution of a finite element analysis, the meshing is refined or the shape functions are enriched in the neighborhood of a particular sub-region (e.g. in stress concentration regions). One technique then consists in applying the boundary conditions (loads or displacements) derived from the global analysis onto the boundary of the newly meshed sub-region [132]. This method known as "zooming" is not satisfactory since it does not account for the coupling between the locally refined substructure and the global coarse mesh. The entire coupled system must be re-analyzed and the definition of the compatibility between local and global sub-regions is a key issue.

As will be seen in the application examples, the analysis of substructured models with non-conforming interfaces leads to good accuracy within each substructure, but since in the general case the compatibility can not be strictly enforced, the solution might be approximate in the vicinity of the interfaces. The sometimes poor quality of the results on the interface does not affect the solution away from it (Saint-Venant's principle) as long as the global behavior of the structure is well represented in the assembled model. If the solution is also to be accurate in the vicinity of the interface, as it is for the detailed stress analysis around the wing/fuselage intersection, the mesh should be locally modified in order to build a new mesh with matching nodes. The interface is then said to be conforming and the classical convergence properties of the finite element theory apply. Nevertheless, modifying the mesh is a tedious task when dealing with large and intricate

models so that working with non-matching meshes is indeed very appealing.

Conforming interfaces naturally appear when a model is decomposed into sub-domains, and the assembly is then trivial: it consists in the classical finite element assembling where a global numbering is introduced so that the assembled model is constructed by simple localization of the element nodes. However, the system may also be kept in its substructured form e.g. for parallel computing reasons, in which case the matching of the interface displacements must be explicitly enforced.

One assembling technique consisting in introducing Lagrange multipliers allows to reduce the number of variables defining the interface solution by expressing only a weak interface compatibility. Reducing the interface unknowns is licit if the interface displacement is smooth enough to be represented by a small number of variables. The computation of the solution is then cheaper. Whenever a global analysis is to be performed on a very fine mesh, the reduction of the interface number of variables is primordial. For example, when performing a modal analysis of a wing, one would like to make use of the very fine model built for the static analysis [76], but the level of refinement needed for stress analysis is definitely too high for computing the first few low frequencies needed for the purpose of aeroelastic computation for instance. Knowing that solving an eigenfrequency problem is very costly, reducing the model might be the only way to run a modal analysis on a fine mesh. One method is then to coarsen the mesh, e.g. by introducing simple constraints between neighboring nodes in order to build macro elements [130]. Another method is to project the solution space of the substructures onto a reduced subspace: the subspace may include the statically exact solution (static condensation or Guyan reduction) and the substructure modes: the most popular modal synthesis methods are the Craig-Bampton and the McNeal-Rubin methods [91, 154, 181]. A further reduction of the model may then be obtained by projecting the interface displacements onto interface modes associated with the primal interface operator [22]. Computing the exact interface modes is however expensive. Another method to reduce the interface displacement is then to express the interface compatibility in a weak form as will be shown later [76] .

In Chapter 2, we will discuss the different ways of defining interface compatibility for matching and non-matching meshes. The mathematical tools for adding the compatibility constraints to the set of substructure equations will be briefly presented. The two-field hybrid method based on the Lagrange multiplier approach will be discussed in Chapter 3.

Modal reduction methods for the internal and interface degrees of freedom are outlined in Chapter 4. In particular, we present a new spectral approximation method for the dual interface operator. In Chapter 5 we will show how to formulate a reduced interface dual problem by assuming simple distribution functions for the interface stresses. Reduced two-field hybrid methods are good at finding accurate solution for smooth static and dynamic problems, but the method will appear to be inappropriate to dealing with heterogeneous systems.

A new post-processing method for the displacements resulting from weak compatibility constraints is then presented in Chapter 6: a strong compatibility is re-enforced by a smoothing scheme based on a Rayleigh-Ritz procedure so as to take account of the heterogeneities appearing between the connecting substructures. First, the smoothing method will be discussed for the interface between two subdomains and in Chapter 7 it is generalized for an arbitrary number of substructures. Several examples will be presented in order to assess the efficiency of the proposed smoothing. To conclude part I, methods for improving the reduced interface problem iteratively will be discussed and we will show how they lead in a natural way to conjugate gradient algorithms which are the subject of part II.

Chapter 2

Compatibility of substructure interfaces

The finite element solutions generally used in practice are based on kinematically admissible or hybrid formulations, and inter-element compatibility is ensured by defining for each connector a unique global displacement variable. When substructures are assembled, the displacement compatibility between subdomains, which is an essential condition for applying kinematically admissible principles, cannot always be ensured exactly and the interface compatibility has to be reformulated.

First, the different ways of expressing the interface compatibility between two substructures will be reviewed. Special issues related to multiple substructures will then be addressed. In section 2.3 we will present the mathematical and numerical tools for embedding the compatibility constraints in the substructure equations. Finally we discuss and motivate our choice of the two-field hybrid method.

2.1 Compatibility constraints between two substructures

Let us assume that we want to assemble two substructures called $\Omega^{(1)}$ and $\Omega^{(2)}$, and we call Γ_I the interface on which they connect. The interface is said *conforming* if the finite element mesh is *matching*. When the mesh is *non-matching*, the interface is said *non-conforming*, meaning that the displacements defined along the interface on each subdomain can not be set equal because of one of the following reasons:

- the element vertex nodes do not correspond;
- the element vertex nodes correspond but the element shape functions are different;
- the connected elements represent different structural approximations (e.g. beam element connected to a volume element).

Note that we will not discuss the case of overlapping domains, although it might be of a practical interest. For instance, when designing the mesh of the subparts, the substructures may not intersect exactly on a common interface: part of the structure in the vicinity of the interface may belong to several substructures and the meshings on the overlapping region are in general non-matching. The compatibility of the meshing must then be defined on the overlapping part and special methods must be used. The non-matching and overlapping meshes are usually tackled using Lagrange multiplier formulations [112] so that in effect, the compatibility can be expressed in a way similar to the non-overlapping case.

2.1.1 Non-conforming interface

The interface Γ_I can be a point, a curve or a surface and we will always assume that the *geometric compatibility* is satisfied [132], i.e. the substructure boundaries defined on each side of an interface by the shape of the undeformed finite element edges are geometrically identical. This might be obvious, but it is not always the case. If we assume for one moment that the finite elements on one side are isoparametric and linear, their edges are straight; curves and curved surfaces will be approximated by piecewise linear segments. Hence, if all elements along Γ_I are linear, the edges of $\Omega^{(1)}$ and $\Omega^{(2)}$ can be equal on a curved interface only if they have matching vertices.

The question is now how to define the displacement compatibility on the interface Γ_I when the nodes of the meshing do not correspond. At first, one could try to impose the exact compatibility of the displacements. Consider the simple interface described in Fig. 2.1 were $\Omega^{(1)}$ has 5 nodes along Γ_I whereas $\Omega^{(2)}$ has a boundary of 3 nodes. We assume that $\Omega^{(1)}$ derives from a compatible meshing with linear elements that has been refined so that the middle nodes are matching. Fig. 2.1(a) shows a p-refinement (the elements of $\Omega^{(1)}$ are quadratic) and Fig. 2.1(b) describes a h-refinement (the initial meshing has been subdivided).

To enforce exact compatibility of the interface, we must impose that the segments defined by the displacement of the nodes 1-2-3 and 3-4-5 remain straight. We thus enforce two linear constraints on the boundary $\partial\Omega^{(1)}$ which in effect eliminate the interface degrees of freedom (d.o.f.) introduced by the refinement and the displacements are linear on both sides of Γ_I (Fig. 2.1). Let us observe two important points here:

- First, when a substructure is refined without moving the nodes of the coarse grid, it is possible to enforce exact continuity on the interface by imposing the interface field to be equal to the displacement field on the edge of the substructure having the lower number of d.o.f. (coarser side). This substructure is called the *master* for Γ_I while the substructure with a higher number of d.o.f. on which constraints are defined is called the *slave* side.

- The finite element model on the refined side is stiffened due to the constraining of the boundary degrees of freedom. However, this stiffening is local since only the displacement along Γ_I is affected. Moreover, the stiffening is acceptable: the boundary is given the same number of d.o.f. as the coarsest side.

Let us now consider the case of linear elements when their nodes on the interface do not match at all as it happens when assembling substructures modeled by independent teams or by automatic mesh generators, or when nodes are relocated in an adaptive mesh (Fig. 2.2). It is clear that exact compatibility of the interface displacement can be obtained only if the entire substructure boundary is constrained to remain straight, meaning that instead of being piecewise linear, the displacement would be linear along the entire interface. Clearly, this results in a tremendous stiffening of the finite element models on both sides of Γ_I.

From the very simple discussion above, we see that, except for refined meshes, no exact compatibility can be defined between non-matching substructures without introducing an unacceptable stiffening of the entire model. Therefore, several methods for defining an approximate compatibility have been developed and will now be summarized.

A reference displacement field is defined on Γ_I with respect to which the interface compatibility will be described. The reference displacement field u_{Γ_I} will be written in a general way as

$$\boldsymbol{u}_{\Gamma_I}(\eta) = \boldsymbol{N}_{\Gamma_I}(\eta)\boldsymbol{u}_{\Gamma_I} \tag{2.1}$$

where $\boldsymbol{u}_{\Gamma_I}$ is a unicolumn matrix collecting the n_{Γ_I} reference displacement parameters and $\boldsymbol{N}_{\Gamma_I}$ is the shape function matrix of dimension $n_d \times n_{\Gamma_I}$, n_d being the dimension of the interface displacement field (thus also the number of degrees of freedom defined on a node). The variable η represents the set of coordinates associated with the interface boundary.

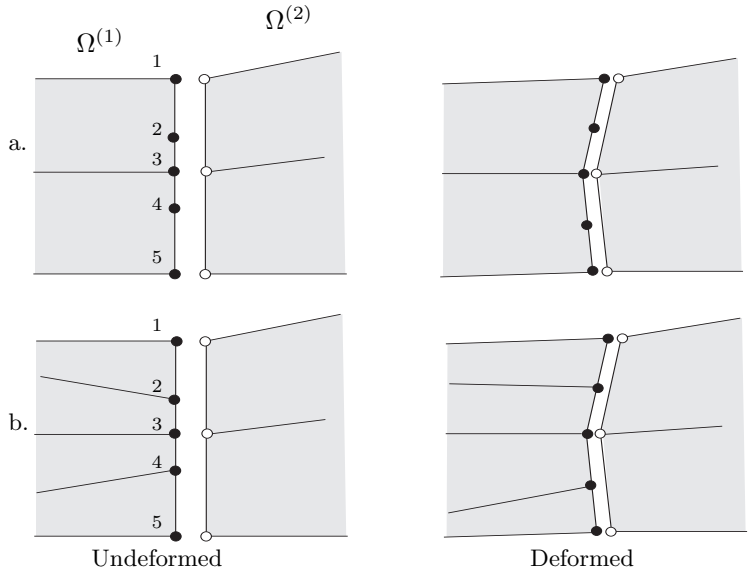


Figure 2.1: Non-conforming interface of refined meshing

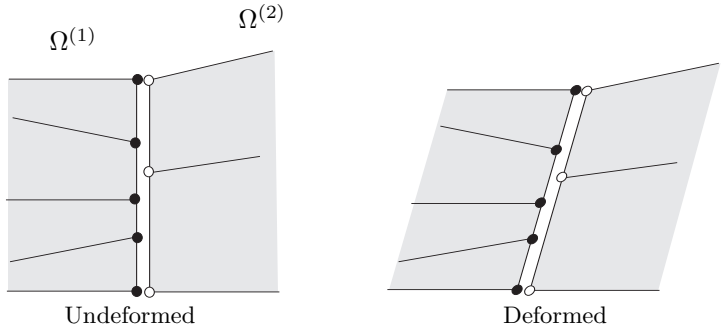


Figure 2.2: General non-conforming interface

Node collocation method

The first idea that comes to mind is to impose that the nodes of the substructure boundaries remain on the reference interface surface defined by the reference displacement u_{Γ_I} (Fig. 2.3(a)). We will thus write a set of $n_b^{(1)}$ and $n_b^{(2)}$ constraints for the boundary displacements $u_b^{(1)}$ and $u_b^{(2)}$ of the substructures $\Omega^{(1)}$ and $\Omega^{(2)}$ respectively, $n_b^{(1)}$ and $n_b^{(2)}$ being the number of boundary d.o.f. of the substructures. Calling $\boldsymbol{N}_{\Gamma_I}(\eta_n)$ the value of the shape function at the coordinates of node n and calling (s) the generic substructure number, the boundary d.o.f. $\boldsymbol{u}_{b,n}^{(s)}$ belonging to substructure $\Omega^{(s)}$ must satisfy

$$\boldsymbol{u}_{b,n}^{(s)} = \boldsymbol{N}_{\Gamma_I}(\eta_n)\boldsymbol{u}_{\Gamma_I} \qquad s = 1, 2 \tag{2.2}$$

and in a general manner we will write the constraints related to the collocation of the boundary nodes onto the reference surface as

$$\boldsymbol{u}_b^{(s)} = \boldsymbol{C}^{(s)}\boldsymbol{u}_{\Gamma_I} \qquad s = 1, 2 \tag{2.3}$$

$\boldsymbol{C}^{(s)}$ being a matrix of dimension $n_b^{(s)} \times n_{\Gamma_I}$. In order for the reference displacement field to be uniquely defined by $\boldsymbol{u}_b^{(1)}$ and $\boldsymbol{u}_b^{(2)}$, we will assume that the number of d.o.f. defined for the reference displacement is lower or equal to the minimum of $n_b^{(1)}$ and $n_b^{(2)}$, i.e. [7, 60]

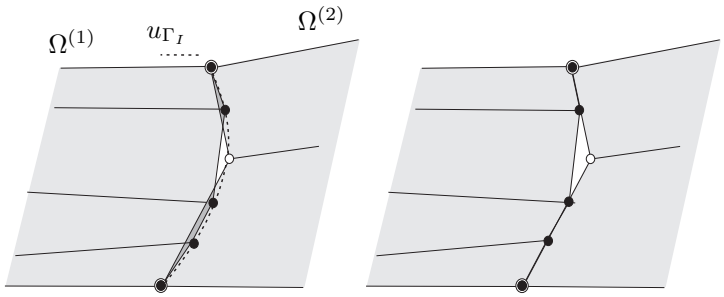
$$n_{\Gamma_I} \leq \min\left(n_b^{(1)}, n_b^{(2)}\right) \tag{2.4}$$

A particular case of the collocation method appears if the substructure of minimum boundary nodes is taken as reference. The substructure having the finer meshing is then the slave side and its boundary nodes are required to lie on the master boundary defined by the reference side (Fig. 2.3(b)). This procedure is very similar to the method discussed above for enforcing exact compatibility between refined and non-refined substructures, but here compatibility is exact only at the nodes of the slave boundary. The compatibility is then approximated by the set of constraints

$$\boldsymbol{u}_b^{(slave)} = \boldsymbol{C}\boldsymbol{u}_b^{(master)} \tag{2.5}$$

where $\boldsymbol{C}$ is built by evaluating the boundary shape functions on $\partial\Omega^{(master)}$ at the position of the nodes of $\partial\Omega^{(slave)}$.

In the collocation method, the boundaries are "glued" only point-wise which, except for the special case of mesh refinement, is far from optimum as will be seen later. Moreover, the collocation of the nodes corresponds to a strong nodal compatibility and the assembling constraints are highly dependent on the node distribution on Γ_I. The method described below relaxes the nodal compatibility requirement by enforcing it only in a least squares sense.



a. With reference field b. Master and slave

Figure 2.3: Collocation of nodes

Discrete least squares compatibility

If we look at the reference displacement $\boldsymbol{u}_{\Gamma_I}$ defined by a number of variable $n_{\Gamma_I} \leq \min(n_b^{(1)}, n_b^{(2)})$ as an interpolation of the nodal displacements on $\partial\Omega^{(1)}$ and $\partial\Omega^{(2)}$, the best approximation in the least-squares sense of the nodal displacement mismatch is found when the parameters $\boldsymbol{u}_{\Gamma_I}$ satisfy the minimum conditions

$$\frac{\partial}{\partial \boldsymbol{u}_{\Gamma_I,i}} \left\{ \left(\boldsymbol{u}_b^{(s)} - \boldsymbol{C}^{(s)} \boldsymbol{u}_{\Gamma_I} \right)^T \left(\boldsymbol{u}_b^{(s)} - \boldsymbol{C}^{(s)} \boldsymbol{u}_{\Gamma_I} \right) \right\} = 0 \qquad \begin{matrix} i = 1, \dots n_{\Gamma_I} \\ s = 1, 2 \end{matrix} \tag{2.6}$$

Thus, the discrete least squares formulation is expressed by the constraints

$$\boldsymbol{C}^{(s)^T} \boldsymbol{u}_b^{(s)} = \boldsymbol{C}^{(s)^T} \boldsymbol{C}^{(s)} \boldsymbol{u}_{\Gamma_I} \qquad s = 1, 2 \tag{2.7}$$

This is clearly a weakening of the collocation constraints: the compatibility conditions (2.7) are the projection of the collocation constraints (2.3) on the columns of $\boldsymbol{C}^{(s)}$. If we recall the assumption (2.4), the square matrices $\boldsymbol{C}^{(s)^T} \boldsymbol{C}^{(s)}$ of dimension $n_{\Gamma_I} \times n_{\Gamma_I}$ are regular. Obviously, as for the collocation method, the boundary on the coarser side of Γ_I can be taken as reference in which case the compatibility constraints become [132]

$$\boldsymbol{C}^T \boldsymbol{u}_b^{(slave)} = \boldsymbol{C}^T \boldsymbol{C} \boldsymbol{u}_b^{(master)} \tag{2.8}$$

The least squares method leads to a less rigid linkage between the substructures, but as in the collocation of nodal displacements, the constraints are still derived from nodal compatibility which means that the substructures are also connected only point-wise.

Continuous least squares formulation

Instead of defining weak compatibility nodal constraints as in the discrete least squares method, one can consider the exact continuous compatibility requirement

$$\boldsymbol{u}_b^{(1)}(\eta) = \boldsymbol{u}_{\Gamma_I}(\eta) = \boldsymbol{u}_b^{(2)}(\eta) \tag{2.9}$$

which, according to the discussion of the beginning of this section, cannot be enforced exactly in general without excessive stiffening of the model. Thus, we will derive again a least squares formulation but this time relative to the continuous compatibility (2.9). To relax the compatibility condition, let us consider once more the reference field $\boldsymbol{u}_{\Gamma_I}(\eta)$ as an interpolation of the boundary displacements defined along Γ_I by the substructures shape functions. The best approximation is then obtained by searching for the minimum displacement gap along the interface, i.e by minimizing the $\mathbb{L}^2(\Gamma_I)$ norm of the compatibility error:

$$\frac{\partial}{\partial \boldsymbol{u}_{\Gamma_I,i}} \int_{\Gamma_I} \left(\boldsymbol{u}_b^{(s)}(\eta) - \boldsymbol{u}_{\Gamma_I}(\eta) \right)^T \left(\boldsymbol{u}_b^{(s)}(\eta) - \boldsymbol{u}_{\Gamma_I}(\eta) \right) d\eta \qquad \begin{matrix} i = 1, \dots n_{\Gamma_I} \\ s = 1, 2 \end{matrix} \tag{2.10}$$

The substructure boundary displacements are discretized in the finite element model by

$$\boldsymbol{u}_b^{(s)}(\eta) = \boldsymbol{N}_b^{(s)}(\eta) \boldsymbol{u}_b^{(s)} \tag{2.11}$$

where $\boldsymbol{N}_b^{(s)}$, the shape function matrix of dimension $n_d \times n_b^{(s)}$, is defined piece-wise on the element edges and where $\boldsymbol{u}_b^{(s)}$ are the boundary d.o.f. of $\Omega^{(s)}$. The continuous least squares formulation (2.10) then results in the n_{Γ_I} constraints

$$\boldsymbol{b}^{(s)} \boldsymbol{u}_b^{(s)} = \boldsymbol{D} \boldsymbol{u}_{\Gamma_I} \qquad s = 1, 2 \tag{2.12}$$

where the matrices $\boldsymbol{b}^{(s)}$ of dimension $n_{\Gamma_I} \times n_b^{(s)}$ are computed by integrating the product of the reference and the finite element shape functions

$$\boldsymbol{b}^{(s)} = \int_{\Gamma_I} \boldsymbol{N}_{\Gamma_I}^T \boldsymbol{N}_b^{(s)} d\eta \tag{2.13}$$

and the square matrix $\boldsymbol{D}$ of dimension $n_{\Gamma_I} \times n_{\Gamma_I}$ is given by

$$\boldsymbol{D} = \int_{\Gamma_I} \boldsymbol{N}_{\Gamma_I}^T \boldsymbol{N}_{\Gamma_I} d\eta \tag{2.14}$$

According to (2.12), the compatibility constraint between 2 substructures can be written as

$$\boldsymbol{b}^{(1)} \boldsymbol{u}_b^{(1)} = \boldsymbol{b}^{(2)} \boldsymbol{u}_b^{(2)} \tag{2.15}$$

which explicitly defines the constraints between the interface displacements of the substructures without reference to the parameters $\boldsymbol{u}_{\Gamma_I}$. Nevertheless, the shape functions $\boldsymbol{N}_{\Gamma_I}$ of the reference displacement field are embedded in the definition of $\boldsymbol{b}^{(s)}$.

2.1.2 Matching meshes

In the case of conforming interfaces, nodal collocation and continuous compatibility are equivalent: according to the very definition of matching meshes, the displacement field is compatible on the edges of the elements if and only if the nodal displacements coincide. The interface compatibility constraints are then given by identifying the corresponding nodes on the edges connecting on Γ_I. If we call $\boldsymbol{b}^{(s)}$ a Boolean square matrix expressing the renumbering of the boundary d.o.f. of $\Omega^{(s)}$ into a global interface numbering, the nodal constraints are as in (2.15). If we also define a matrix $\boldsymbol{B}^{(s)}$ of dimension $n_b^{(s)} \times n^{(s)}$ where $n^{(s)}$ is the total number of d.o.f. of $\Omega^{(s)}$ stored in $\boldsymbol{u}^{(s)}$, and if we assume that the columns of $\boldsymbol{B}^{(s)}$ corresponding to internal d.o.f. are zero whereas the columns associated with the interface nodes correspond to the submatrix $\boldsymbol{b}^{(s)}$, the interface compatibility (2.15) can be put in the form

$$\boldsymbol{B}^{(1)} \boldsymbol{u}^{(1)} = \boldsymbol{B}^{(2)} \boldsymbol{u}^{(2)} \tag{2.16}$$

Let us note that in practice, the localization matrices $\boldsymbol{b}^{(s)}$ and $\boldsymbol{B}^{(s)}$ are never explicitly constructed: the expression $\boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)}$ simply represents the extraction and the reordering of the interface d.o.f. from the complete set.

The localization operators are usually replaced by a system of pointers which are defined when expressing the nodal splitting. When the substructures are artificially built by a decomposition algorithm, the localization data are part of the results of the decomposition.

The Boolean localization matrices $\boldsymbol{B}^{(s)}$ can also be derived from the so-called algebraic partitioning of a global structural model [141]. If we assume that the structure Ω has been subdivided into substructures $\Omega^{(s)}$, we call $\boldsymbol{K}_d$ the global structural matrix which diagonal blocks are the substructures matrices $\boldsymbol{K}^{(s)}$

$$\boldsymbol{K}_d = \begin{bmatrix} \boldsymbol{K}^{(1)} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{K}^{(N_s)} \end{bmatrix} \tag{2.17}$$

where N_s is the number of substructures. The assembled matrix $\boldsymbol{K}$ is then obtained by summing up the contributions of matching nodes and can be formally expressed by

$$\boldsymbol{K} = \boldsymbol{L}^T \boldsymbol{K}_d \boldsymbol{L} \tag{2.18}$$

where $\boldsymbol{L}$ is the *assembly Boolean matrix*.

The constraint equations are then found by computing the null space of $\boldsymbol{L}$. To illustrate this issue, let us consider the case of two substructures: we have

$$\begin{bmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \end{bmatrix} = \boldsymbol{L}\boldsymbol{u} \tag{2.19}$$

where $\boldsymbol{u}$ is the vector of assembled d.o.f. Assuming that the interface d.o.f. are numbered last in $\Omega^{(1)}$ and that they are numbered first in $\Omega^{(2)}$, the assembly Boolean matrix can be formally

written as

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{2.20}$$

The null space of $\boldsymbol{L}^T$ can be found be inspection:

$$\text{null}(\boldsymbol{L}^T) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \\ -\boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^{(1)^T} \\ \\ -\boldsymbol{B}^{(2)^T} \\ \\ \end{bmatrix} = \boldsymbol{B}^T \tag{2.21}$$

where we call $\boldsymbol{B}$ the constraining matrix such that (2.16) writes

$$\boldsymbol{B} \begin{bmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \end{bmatrix} = \boldsymbol{0} \tag{2.22}$$

2.2 Compatibility between multiple substructures

In the general case when N_s substructures are to be assembled, some portions of the interface may connect more than two grids. The *multiplicity* of an interface Γ_I will be defined as *the number of substructures connecting on* Γ_I. We will discuss here some particular issues for the intersection point of more than 2 substructures, called *cross-point*.

Fig. 2.4 shows the connection of three substructures on a cross-point of Γ_I. For non-matching meshings, Fig. 2.4(a) depicts the case when one substructure has no node defined on the cross-point. The compatibility at the cross-point will be satisfied if we use a nodal collocation formulation and if $\partial\Omega^{(1)}$ can be chosen as the master side on ${\Gamma_I}^{12} \cup {\Gamma_I}^{13}$, ${\Gamma_I}^{sr}$ denoting the interface between $\Omega^{(s)}$ and $\Omega^{(r)}$. However if $\partial\Omega^{(2)}$ and $\partial\Omega^{(3)}$ are the masters on ${\Gamma_I}^{12}$ and ${\Gamma_I}^{13}$ respectively, no exact compatibility will be enforced on the cross-point. If a node is defined on the cross-point for every substructure (Fig. 2.4(b)), then exact compatibility will be satisfied in a collocation formulation. When applying a least squares method for this case, one could decide to treat the cross-point as a special interface portion and impose nodal matching on the cross-point while the remaining Γ_I is handled separately. In practice however, the cross-point is not considered apart and it is generally assumed to belong to all ${\Gamma_I}^{sr}$ so the cross-point does not require any specific care [3]: the cross-point is considered as a negligible subspace of Γ_I.

For matching meshes (Fig. 2.4(c)), exact compatibility is ensured over Γ_I simply by collocating the nodes as usual. For instance, using the subscript c for the cross-point nodes, we can write

$$\boldsymbol{u}_c^{(1)} = \boldsymbol{u}_c^{(2)} \tag{2.23}$$
$$\boldsymbol{u}_c^{(2)} = \boldsymbol{u}_c^{(3)} \tag{2.24}$$
$$\boldsymbol{u}_c^{(3)} = \boldsymbol{u}_c^{(1)} \tag{2.25}$$

Obviously the last constraint is redundant and may be omitted. We will see later in this work that in order to simplify the programming of the assembling algorithms and, more importantly, for numerical efficiency, the complete set of redundant constraints is considered in practice.

2.3 Numerical methods for constrained systems

The compatibility constraints discussed in section 2.1 and 2.2 must be added to the substructure equations to form the assembled system for the entire structure. In this section we will review the mathematical tools and the numerical methods for solving the constrained problem arising from

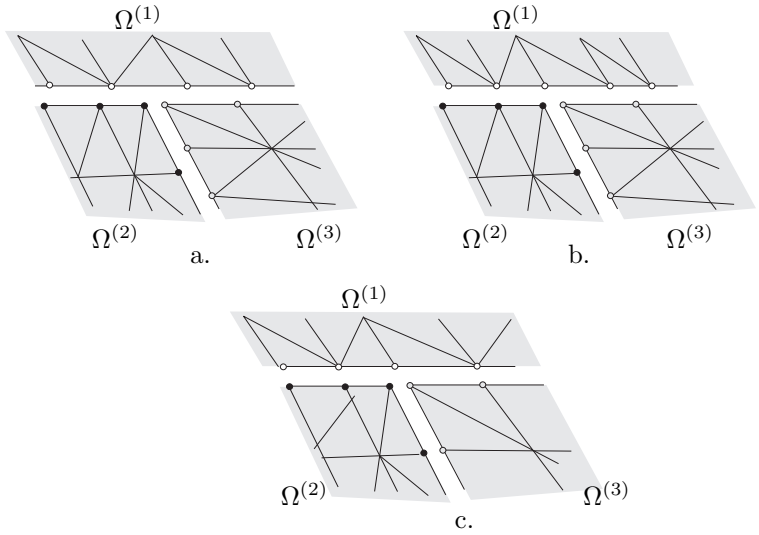


Figure 2.4: Cross-point on an interface

the assembly. The subject of constrained equations has been extensively studied in the context of optimization [18], contact problems [119] and in multi-body analysis [28]. The scope of this section is to summarize the most popular methods and to show how they apply to the assembly problem.

2.3.1 Weak form in structural analysis and finite element discretization

In structural analysis, the equilibrium equations are generally expressed in the form of a variational principle in order to formulate approximation methods. The linearized structural problem for a structure Ω has a solution u which is deduced from the weak form [12, 175]

$$a(u,v)_\Omega = l(v)_\Omega \qquad \forall v \in \mathbb{V}_{\Gamma_d}(\Omega) = \left\{ v \in \mathbb{H}^1(\Omega), v|_{\Gamma_D} = \overline{u} \right\} \tag{2.26}$$

where Γ_D is the portion of $\partial\Omega$ with the Dirichlet boundary condition $u = \overline{u}$ and $\mathbb{H}^1(\Omega)$ is the Sobolev space of the functions that are square integrable as well as their first partial derivatives. The functional $a(u,v)_\Omega$ defined on $\mathbb{V}_{\Gamma_d}(\Omega) \times \mathbb{V}_{\Gamma_d}(\Omega)$ is a bilinear continuously elliptic form which for the linear elastostatic problem in $\mathbb{R}^3$ is given by (using implicit summation)

$$a(u,v)_\Omega = \int_\Omega u_{(i,j)} c_{ijkl} v_{(k,l)} d\Omega \qquad i,j,k,l = 1 \ldots 3 \tag{2.27}$$

where c_{ijkl} are the components of Hooke's tensor and

$$u_{(i,j)} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{2.28}$$

are the components of the linear deformation tensor.

The $\mathbb{V}_{\Gamma_d}$-ellipticity of $a(u,v)_\Omega$ ensures the uniqueness of the solution. The functional $l(v)_\Omega$ is the continuously linear form

$$l(v)_\Omega = \int_\Omega f_i v_i d\Omega + \int_{\Gamma_N} t_i v_i d\Gamma_N \tag{2.29}$$

f and t being respectively the volume force field in Ω and the surface traction on $\partial\Omega$. The boundary Γ_N is the portion of $\partial\Omega$ with Neumann boundary conditions.

Assuming that the bilinear form $a(u,v)$ is symmetric (which is obvious for the linear elastostatic case (2.27)), and defining the functional

$$\mathcal{J}(v)_\Omega = \frac{1}{2}a(v,v)_\Omega - l(v)_\Omega \tag{2.30}$$

it is a classical result that the weak form (2.26) is equivalent to the minimization problem:

$$\mathcal{J}(u)_\Omega = \min_{v\in\mathbb{V}_{\Gamma_d}(\Omega)} \mathcal{J}(v)_\Omega \tag{2.31}$$

Noting that $\frac{1}{2}a(v,v)_\Omega$ and $l(v)_\Omega$ correspond respectively to the deformation energy and to the work of the external loads, the latter form expresses the kinematically admissible principle of minimum total potential energy for structural mechanics.

When introducing a finite element discretization, the best approximation solution of the structural problem is deduced from

$$\mathcal{J}(u_h)_\Omega = \min_{v_h\in\mathbb{V}_{h,\Gamma_d}} \mathcal{J}(v_h)_\Omega \qquad\qquad \mathbb{V}_{h,\Gamma_d}(\Omega) = \left\{v\in\mathbb{H}^1_h(\Omega), v|_{\Gamma_D}=0\right\} \tag{2.32}$$

where $\mathbb{V}_{h,\Gamma_d}(\Omega)$ denotes the finite element space of kinematically admissible displacements (or velocities for non-linear mechanics).

If the structure Ω is subdivided in N_s substructures $\Omega^{(s)}$ such that $\Omega = \bigcup_s \Omega^{(s)}$, the weak form (2.26) is equivalent to

$$\sum_{s=1}^{N_s} a(u,v)_{\Omega^{(s)}} = \sum_{s=1}^{N_s} l(v)_{\Omega^{(s)}} \qquad \forall v\in\mathbb{V}_{\Gamma_d}(\Omega) \tag{2.33}$$

We define the local spaces of restrictions

$$\mathbb{V}^{(s)}_{\Gamma_d} = \left\{v^{(s)} = v|_{\Omega^{(s)}}, v\in\mathbb{V}_{\Gamma_d}(\Omega)\right\} \tag{2.34}$$

and the global interface $\Gamma_I = (\cup_s \partial\Omega^{(s)})\backslash\partial\Omega$ are decomposed into faces Γ_I^{sr} separating neighboring subdomains $\Omega^{(s)}$ and $\Omega^{(r)}$. The space of admissible displacements can then be defined as the product space $\prod_s \mathbb{V}^{(s)}_{\Gamma_d}$ when the interface compatibility is satisfied, i.e.

$$\mathbb{V}_{\Gamma_d}(\Omega) = \left\{(v^{(s)})_{s=1\dots N_s} \in \prod_s \mathbb{V}^{(s)}_{\Gamma_d}, v^{(s)}|_{\Gamma_I^{sr}} = v^{(r)}|_{\Gamma_I^{sr}},\ \forall r,s = 1\dots N_s\right\} \tag{2.35}$$

The finite element problem for a decomposed domain can now be stated as

$$\mathcal{J}(u_h^{(s)})_{\Omega^{(s)}} = \min_{v_h^{(s)}\in\mathbb{V}^{(s)}_{h,\Gamma_d}} \mathcal{J}(v_h^{(s)})_{\Omega^{(s)}} \qquad\qquad s = 1\dots N_s \tag{2.36}$$

and in order for the displacement field to be kinematically admissible, that is for $\prod_{s=1}^{N_s} \mathbb{V}^{(s)}_{h,\Gamma_d}$ to represent $\mathbb{V}_{h,\Gamma_d}$, the substructure displacements must satisfy the compatibility on the interfaces:

$$u_h^{(s)}|_{\Gamma_I^{sr}} = u_h^{(r)}|_{\Gamma_I^{sr}} \tag{2.37}$$

2.3.2 The Lagrangian multipliers method

The Lagrangian multipliers technique is a classical and widely used method for dealing with constrained systems (e.g. in optimization [18]). The theoretical study of the method consisting in enforcing essential boundary conditions in a Finite Element discretization through Lagrange multipliers was done in [12] and its application was discussed e.g. in [142]. Later on, more theoretical results came out for the use of Lagrange multipliers for gluing subdomains of non-matching elements [3,47]. In our discussion, we will show how the method applies when enforcing compatibility along the interface edges.

In the Lagrangian multipliers method, the weak form (2.33) and the partitioned definition (2.35) of $\mathbb{V}_{\Gamma_d}(\Omega)$ are transformed into the following variational expression [175]:

$$\bullet \sum_{s=1}^{N_s} a(u^{(s)}, v^{(s)})_{\Omega^{(s)}} - \sum_{s=1}^{N_s} l(v^{(s)})_{\Omega^{(s)}} + \sum_{s=1}^{N_s} \sum_{\substack{\Gamma_I^{sr}\neq\emptyset \\ s<r}} \int_{\Gamma_I^{sr}} \lambda^{sr} \left(v^{(s)}|_{\Gamma_I^{sr}} - v^{(r)}|_{\Gamma_I^{sr}} \right) d\Gamma_I^{sr} = 0 \qquad \forall v^{(s)} \in \prod_s \mathbb{V}^s_{\Gamma_d}, \quad r,s = 1 \dots N_s \tag{2.38a}$$

$$\bullet\ u^{(s)}|_{\Gamma_I^{sr}} - u^{(r)}|_{\Gamma_I^{sr}} = 0 \qquad \forall \Gamma_I^{sr} \neq \emptyset \tag{2.38b}$$

where λ^{sr} is defined in the dual space of the trace of $\mathbb{V}^{(s)}_{\Gamma_d}$ on Γ_I^{sr}.

The energy functional associated to a substructure is defined by

$$\begin{aligned} \mathcal{J}(v^{(s)})_{\Omega^{(s)}} &= \frac{1}{2} a(v^{(s)}, v^{(s)})_{\Omega^{(s)}} - l(v^{(s)})_{\Omega^{(s)}} \\ &= \frac{1}{2} \int_{\Omega^{(s)}} v_{(i,j)} c_{ijkl} v_{(i,j)} d\Omega^{(s)} \\ &\quad - \int_{\Omega^{(s)}} f_i^{(s)} v_i^{(s)} d\Omega^{(s)} - \int_{\Gamma_N^{(s)}} t_i^{(s)} v_i^{(s)} d\Gamma_N^{(s)} \end{aligned} \tag{2.39}$$

The weak form (2.38) of the substructured system is then equivalent to finding the stationary point $(u^{(s)}, \lambda^{(s)})$ of the two-field Lagrangian function [67]

$$\mathcal{L}(v^{(1)}, v^{(2)}, \dots v^{(N_s)}, \mu^{12}, \dots) = \sum_{s=1}^{N_s} \mathcal{J}(v^{(s)})_{\Omega^{(s)}} + \sum_{s=1}^{N_s} \sum_{\substack{\Gamma_I^{sr}\neq\emptyset \\ s<r}} \int_{\Gamma_I^{sr}} \mu^{sr} \left(v^{(s)}|_{\Gamma_I^{sr}} - v^{(r)}|_{\Gamma_I^{sr}} \right) d\Gamma_I^{sr} \tag{2.40}$$

The stationarity of $\mathcal{L}$ leads to the local linear equilibrium equations [91]

$$\sum_{i=1}^{3} \frac{\partial \sigma_{ij}^{(s)}}{\partial x_i} + f_j^{(s)} = 0 \qquad \text{in } \Omega^{(s)} \qquad j = 1,2,3 \tag{2.41a}$$

$$t_j^{(s)} = \sum_{i=1}^{3} n_i^{(s)} \sigma_{ij}^{(s)} \qquad \text{on } \Gamma_N^{(s)} \tag{2.41b}$$

$$\lambda_j^{sr} = \pm \sum_{i=1}^{3} n_i^{(s)} \sigma_{ij}^{(s)} \qquad \text{on } \Gamma_I^{sr} \tag{2.41c}$$

$$v^{(s)}|_{\Gamma_I^{sr}} = v^{(r)}|_{\Gamma_I^{sr}} \tag{2.41d}$$

where $\Gamma_N^{(s)}$ is the portion of $\partial\Omega^{(s)} \cap \partial\Omega$ with Neumann boundary conditions, $\sigma_{ij} = c_{ijkl} u_{(k,l)}$ are the components of the stress tensor and n_i are the components of the outward normal to the boundary. Eq. (2.41c) shows that the Lagrange multipliers are the dual quantity conjugate to the interface displacements, that is the interface connection stresses that enforce the compatibility constraints (2.41d). This formulation is called *hybrid* because it is based on the explicit expression of the interface stresses and subdomain displacements. It will be studied in more detail in the next chapter.

The general theory of the Lagrange multipliers method (called μ-method for the sake of conciseness) as presented so far for enforcing compatibility between substructures can be applied in the context of the finite element discretization in two different ways.

We can first consider that, starting from the continuous variational principle (2.39) we introduce a finite element discretization for each subdomain $\Omega^{(s)}$ and we introduce distribution functions associated with the Lagrange multipliers field on the interfaces. Expressing the stationarity of $\mathcal{L}$ with respect to the finite element variables and with respect to the parameters of the interface stress distribution leads to a consistent weak form of the equilibrium equations and of the interface compatibility constraints.

A second approach will consist in introducing discrete Lagrange multipliers that will handle discrete compatibility constraints such as the nodal collocation relations. This variant will be called discrete μ-method.

Lagrange multipliers for the continuous compatibility constraints

Considering the general formulation of the variational principle (2.39), a discretization of the substructures equilibrium and of the interface compatibility can be found by defining shape functions $\boldsymbol{N}^{(s)}$ for the displacement field on every sub-domain $\Omega^{(s)}$ and distribution functions $\boldsymbol{\Lambda}^{sr}$ for the interface stresses on Γ_I^{sr} (i.e. the Lagrange multipliers field). In matrix notation we can write

$$\boldsymbol{u}^{(s)}(x) = \boldsymbol{N}^{(s)}(x)\boldsymbol{u}^{(s)} \tag{2.42}$$

$$\boldsymbol{\lambda}^{sr}(\eta) = \boldsymbol{\Lambda}^{sr}(\eta)\boldsymbol{\lambda}^{sr} \tag{2.43}$$

where x are the intra-subdomain coordinates and η are the coordinates defining the substructures interfaces[1]. The stationarity of the Lagrangian function $\mathcal{L}$ (2.40) then leads to the algebraic system

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + {\boldsymbol{B}_c^{(s)}}^T\boldsymbol{\lambda} = \boldsymbol{f}^{(s)} \qquad s = 1, \dots N_s \tag{2.44a}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}_c^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{2.44b}$$

where $\boldsymbol{K}^{(s)}$ and $\boldsymbol{f}^{(s)}$ are the usual stiffness matrices and generalized forces, $\boldsymbol{\lambda}$ contains the interface lagrange multipliers $\boldsymbol{\lambda}^{sr}$, and the constraint matrices $\boldsymbol{B}_c^{(s)}$ are given by

$$\boldsymbol{B}_c^{(s)} = \left[\begin{array}{ccccccc} \boldsymbol{0} & \dots & \boldsymbol{b}_c^{(sr)} & \boldsymbol{0} & \dots & \boldsymbol{b}_c^{(st)} & \dots \end{array} \right] \tag{2.45}$$

$$\boldsymbol{b}_c^{(sr)} = \pm \int_{\Gamma_I^{sr}} {\boldsymbol{\Lambda}^{sr}}^T \boldsymbol{N}^{(s)} d\eta \tag{2.46}$$

The sign in (2.46) is alternated for the matrices $\boldsymbol{b}_c^{(sr)}$ and $\boldsymbol{b}_c^{(rs)}$ pertaining to Γ_I^{sr}.

The term ${\boldsymbol{B}_c^{(s)}}^T\boldsymbol{\lambda}$ in (2.44a) represents the generalized finite element forces arising from the interface stress distribution. Clearly, when the finite element shape functions are identical on each side of an interface, i.e. for matching grids, $\boldsymbol{b}^{(sr)} = -\boldsymbol{b}^{(rs)}$ and Eq. (2.44b) is a general weak form of the interface compatibility. Indeed, from the very definition of the variational principle (2.40), the interface compatibility is weighted by the Lagrange multipliers distribution functions.

[1] Only one set of Lagrange multipliers is defined between two subdomains so that the notations $\boldsymbol{\lambda}^{sr}$ and $\boldsymbol{\lambda}^{rs}$ are equivalent.

For non-conforming interfaces, the shape functions on the substructure edges connected on an interface Γ_I^{sr} are not identical and the constraint matrices $\boldsymbol{b}_c^{(rs)}$ are specific to each side of an interface. Considering the case of two substructures, we see that the constraint (2.44b) is equivalent to the constraints (2.15) derived from the continuous least square formulation (section 2.1.1): the definitions (2.13) and (2.46) are equivalent if the reference displacement field shape function $\boldsymbol{N}_{\Gamma_I}$ is equal to the stress distribution function $\boldsymbol{\Lambda}^{12}$.

Hence we conclude that *the μ-method applied to the interface compatibility is equivalent to defining a weak compatibility by minimizing the $\mathbb{L}^2$ norm of the displacement jump between the substructure edges and a reference displacement field having as a shape function $\boldsymbol{\Lambda}^{12}$.*

When discussing the expression of the compatibility constraints for the collocation and least squares methods in section 2.1, we have shown that the number of parameters defining the reference interface displacement should not exceed the number of d.o.f. on the coarser side of the interface so that the interface parameters are uniquely defined. In the μ-method, the field of Lagrange multipliers $\boldsymbol{\lambda}^{rs}(\eta)$ and the finite element shape functions should also be properly selected. In theory, $\boldsymbol{\Lambda}^{rs}$ and $\boldsymbol{N}^{(s)}$ should be constructed so as to satisfy the Babuška-Brezzi condition (also known as the *inf-sup* condition) [11, 47, 98] in order to ensure that the finite element and Lagrange multipliers discretization converges to the exact solution as the discretization is refined. Unfortunately, it is difficult to verify the so-called *inf-sup* condition on practical problems.

This issue is discussed in detail for the *Mortar* method in fluid analysis [1,3,17]. In the Mortar method the Lagrange multiplier field is taken equal to the trace on Γ_I^{sr} of the displacement shape functions on one side of the interface and for the end points of interface segments, the shape functions of the last two nodes are associated with only one multiplier.

For polynomial stress distribution functions, the Babuška-Brezzi is satisfied if the degree of the polynomial approximation of the Lagrange multiplier interconnecting two substructures $\Omega^{(s)}$ and $\Omega^{(r)}$ remains lower than the square root of the local size of the coarsest discretization on Γ_I^{sr} when the discretization is refined [98]. This sufficient condition being too strong in practice, we will only satisfy a necessary condition for the *inf-sup* criterion: the approximation functions $\boldsymbol{\Lambda}^{sr}$ will be selected such that Eqs. (2.44) are solvable. While such an approach is less than perfect from a mathematical viewpoint, it allows to construct useful and powerful assembly and model reduction algorithms [76, 77]. It means for instance that the number of Lagrange multipliers should not exceed the number of d.o.f. of the coarsest mesh on the interface. This is similar to the condition for building collocation or least squares constraints for non-matching grids (see page 18). Note that satisfying the Babuška-Brezzi condition is important especially for highly irregular cases, but for fairly smooth problems as encountered in practice, good results can be obtained even if the condition is not strictly satisfied [11].

A variant of the μ-method is found if the interface compatibility between two substructures is expressed with respect to an intermediate third displacement field $u_{\Gamma_I^{sr}}$. This technique is known as a *three-field formulation*. Two independent spaces of discretization of the Lagrange multipliers must be defined, one between the reference third field and each substructure [7, 8, 86]. The Lagrangian function (2.40) becomes

$$\mathcal{L} = \sum_{s=1}^{N_s} \mathcal{J}(v^{(s)})_{\Omega^{(s)}} + \sum_{s=1}^{N_s} \sum_{\Gamma_I^{sr}\neq\emptyset} \int_{\Gamma_I^{sr}} \lambda^{s-r} \left(v^{(s)}|_{\Gamma_I^{sr}} - v_{\Gamma_I^{sr}} \right) d\Gamma_I^{sr} \tag{2.47}$$

Calling $\boldsymbol{N}_{\Gamma_I^{sr}}$ the shape functions of the reference displacement $u_{\Gamma_I^{sr}}$ and noting $\boldsymbol{\Lambda}^{s-r}$ the approximation field of the stresses between $\partial\Omega^{(s)}$ and the reference displacement $u_{\Gamma_I^{sr}}$ on Γ_I^{sr}, the variation of the latter expression with respect to $v^{(s)}$, $v_{\Gamma_I^{sr}}$ and λ^{s-r} leads to a three-field system of equations. For the two-subdomain case we find

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + \boldsymbol{Q}^{(s)^T}\boldsymbol{\lambda}^{s-r} = \boldsymbol{f}^{(s)} \qquad s,r=1,2\ ,\ s\neq r \tag{2.48a}$$

$$\boldsymbol{G}^{1-2^T}\boldsymbol{\lambda}^{1-2} + \boldsymbol{G}^{2-1^T}\boldsymbol{\lambda}^{2-1} = \boldsymbol{0} \tag{2.48b}$$

$$\boldsymbol{Q}^{(s)}\boldsymbol{u}^{(s)} + \boldsymbol{G}^{s-r}\boldsymbol{u}_{\Gamma_I^{12}} = \boldsymbol{0} \tag{2.48c}$$

where $\boldsymbol{\lambda}$ collects the multipliers $\boldsymbol{\lambda}^{s-r}$ and, if we assume that boundary d.o.f. are numbered last in $\boldsymbol{u}^{(s)}$, $\boldsymbol{Q}^{(s)}$ and $\boldsymbol{G}^{(s)}$ are defined by

$$\boldsymbol{Q}^{(s)} = \left[\begin{array}{cc} \boldsymbol{0} & \boldsymbol{q}^{s-r} \end{array} \right] \tag{2.49a}$$

$$\boldsymbol{q}^{s-r} = \int_{\Gamma_I^{sr}} \boldsymbol{\Lambda}^{s-r^T} \boldsymbol{N}^{(s)} d\eta \tag{2.49b}$$

$$\boldsymbol{G}^{s-r} = -\int_{\Gamma_I^{sr}} \boldsymbol{\Lambda}^{s-r^T} \boldsymbol{N}_{\Gamma_I^{sr}} d\eta \tag{2.49c}$$

Eq. (2.48b) expresses the equilibrium between the stress fields acting on the interface and (2.48c) are the compatibility condition between the substructures boundary displacements and the reference displacement.

Discrete Lagrange multipliers

The exact interface compatibility constraint for matching grids can be replaced by equating the nodal displacements connected on the interface. For non-matching meshes, weak compatibility conditions can be constructed by collocation or discrete least squares formulations. In those cases, the interface compatibility requirements are expressed after the finite element discretization has been independently introduced in each substructure, whereas for the continuous μ-method discussed above, the finite element and the interface discretization were introduced simultaneously in the variational principle. Therefore, when the compatibility is described by discrete conditions on the nodal displacements in the form (2.3), (2.5) or (2.7), we will introduce discrete Lagrange multipliers, each Lagrange multiplier being associated with one discrete compatibility constraint.

Let us write the discrete interface compatibility constraints under the general form

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{2.50}$$

The Lagrangian function is then written as

$$\mathcal{L}(\boldsymbol{v}^{(s)}, \boldsymbol{\mu}) = \sum_{s=1}^{N_s} \left(\frac{1}{2} \boldsymbol{v}^{(s)^T} \boldsymbol{K}^{(s)} \boldsymbol{v}^{(s)} - \boldsymbol{v}^{(s)^T} \boldsymbol{f}^{(s)} \right) + \boldsymbol{\mu}^T \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{v}^{(s)} \tag{2.51}$$

From the stationarity of $\mathcal{L}$ we deduce the system of equations

$$\boldsymbol{K}^{(s)} \boldsymbol{u}^{(s)} + \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} = \boldsymbol{f}^{(s)} \qquad s = 1, \dots N_s \tag{2.52a}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{2.52b}$$

which is similar to (2.44) except that now the constraint matrices $\boldsymbol{B}_c^{(s)}$ are replaced by $\boldsymbol{B}^{(s)}$ and are no longer defined by the integral of the product of finite element shape functions and stress distributions as in (2.46).

Matching grids The matrices $\boldsymbol{B}^{(s)}$ for matching grids express the equality of the interface nodal displacements. The conjugate Lagrange multipliers are the nodal connection forces: $\boldsymbol{B}^{(s)}$ are Boolean matrices. Let us then consider the square constraint matrices $\boldsymbol{b}_c^{(sr)}$ (2.46) obtained from the continuous μ-method when the number of Lagrange multipliers $\boldsymbol{\lambda}^{sr}$ defined on a conforming interface is equal to the number of d.o.f. on Γ_I^{sr}. If $\boldsymbol{\Lambda}^{sr}$ are independent fields, i.e. if $\boldsymbol{b}_c^{(sr)}$ is non-singular, we can define the change of variables

$$\boldsymbol{\lambda}^{sr} = \boldsymbol{b}_c^{(sr)^{-1}} \, \overline{\boldsymbol{\lambda}}^{sr} \tag{2.53}$$

Replacing in (2.44) shows that the constraint matrices associated to $\overline{\boldsymbol{\lambda}}^{sr}$ are boolean and therefore the continuous μ-method is equivalent to nodal collocation of matching nodes if the space of discretization of λ^{sr} is complete.

Non-conforming interface In the case of collocation for non-matching grids, the term $\boldsymbol{B}^{(s)}\boldsymbol{\lambda}$ in (2.52a) represents the generalized forces associated to discrete loads applied at the collocation points. Thus, a constant stress field will not be correctly transmitted across the interfaces since stresses can be transfered only through the collocation points. Hence we can conclude that for the collocation formulation applied to non-conforming interfaces, the hybrid model will not pass the finite element patch test [8,86]. This conclusion can also be drawn when observing that, from a kinematical viewpoint, compatibility is enforced only point-wise. The same reasoning can also be held for the discrete least squares method. Note that in the particular case of mesh refinement (see Fig. 2.1), exact compatibility can be enforced through nodal collocation without excessive stiffening: the constant stress criteria of the patch test will then be satisfied. In the latter case, nodal collocation can be efficient [180].

The weak compatibility formulation obtained from the continuous μ-method is based on a global definition of the continuity of interface displacements and the associated nodal forces are consistently derived from the interface stress distribution and from the finite element shape functions. Hence a constant stress field will be correctly modeled by the hybrid formulation (2.44): the hybrid model is comparable to a model made of hybrid finite elements and passes the patch test.

2.3.3 Augmented Lagrangian methods

The Lagrangian function (2.40) for a constrained system can be modified by adding a quadratic function of the constraints. For the problem of substructures assembling, the augmented Lagrangian function is

$$\begin{aligned}\mathcal{L}^a(v^{(s)},\mu^{sr}) &= \sum_{s=1}^{N_s}\mathcal{J}(v^{(s)})_{\Omega^{(s)}} + \sum_{s=1}^{N_s}\sum_{\substack{\Gamma_I^{sr}\neq\emptyset\\ s<r}}\int_{\Gamma_I^{sr}}\lambda^{sr}\phi|_{\Gamma_I^{sr}}d\Gamma_I^{sr}\\ &\quad+\sum_{s=1}^{N_s}\sum_{\substack{\Gamma_I^{sr}\neq\emptyset\\ s<r}}\int_{\Gamma_I^{sr}}\frac{p}{2}\phi|^2_{\Gamma_I^{sr}}d\Gamma_I^{sr}\end{aligned} \tag{2.54}$$

where $\phi|_{\Gamma_I^{sr}}=0$ is the compatibility constraint, i.e.

$$\phi|_{\Gamma_I^{sr}} = v^{(s)}|_{\Gamma_I^{sr}} - v^{(r)}|_{\Gamma_I^{sr}} \tag{2.55}$$

and p is a strictly positive number (possibly a function of the interface coordinates). The supplementary term $p\phi|^2_{\Gamma_I^{sr}}/2$ is a penalization term that clearly associates a high cost to the violation of the compatibility constraints and increases the convexity of the functional for the displacement variables so to facilitate the search of the minimum of $\mathcal{L}$ for $v^{(s)}$. Note that the augmented Lagrangian function $\mathcal{L}^a$ has the same stationary point as $\mathcal{L}$: expressing the stationarity of $\mathcal{L}^a$ with respect to μ leads to the compatibility condition $\phi|_{\Gamma_I^{sr}}=0$ and thus, at the stationary point the penalization term is null. The local equations obtained by minimizing $\mathcal{L}^a$ with respect to $v^{(s)}$ are identical to (2.41a) for the substructure internal equilibrium, but the interface equilibrium is now expressed by

$$p\,\left(v_j^{(s)}|_{\Gamma_I^{sr}} - v_j^{(r)}|_{\Gamma_I^{sr}}\right) \pm \lambda_j^{sr} = \sum_{i=1}^{3} n_i^{(s)}\sigma_{ij}^{(s)} \qquad \text{on } \Gamma_I^{sr}, \qquad \begin{matrix} s=1,\dots N_s \\ j=1,2,3 \end{matrix} \tag{2.56}$$

This yields the same solution as (2.41c) since $\phi|_{\Gamma_I^{sr}}=0$, but the system to be solved is different: *the penalization term introduces a fictitious distributed spring of stiffness p along the interface in parallel with the connecting forces λ^{sr}.*

Assuming that the constraint matrix $\boldsymbol{B}=[\boldsymbol{B}^{(1)}\dots\boldsymbol{B}^{(N_s)}]$ is given either by collocation or by the integral form (2.46), the discretized augmented Lagrangian function $\mathcal{L}^a$ writes

$$\mathcal{L}^a(\boldsymbol{v}^{(s)},\boldsymbol{\mu}^{sr}) = \frac{1}{2}\boldsymbol{v}_d^T\boldsymbol{K}_d\boldsymbol{v}_d - \boldsymbol{v}_d^T\boldsymbol{f}_d + \frac{p}{2}\boldsymbol{v}_d^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{v}_d + \boldsymbol{v}_d^T\boldsymbol{B}^T\boldsymbol{\mu} \tag{2.57}$$

where the d subscript corresponds to the quantities relative to the block diagonal structural matrix $\boldsymbol{K}_d$ (see Eq. (2.17)). The stationarity criterion leads to the algebraic system

$$(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})\boldsymbol{u}_d + \boldsymbol{B}^T\boldsymbol{\lambda} = \boldsymbol{f}_d \tag{2.58a}$$

$$\boldsymbol{B}\boldsymbol{u}_d = \boldsymbol{0} \tag{2.58b}$$

The symmetric matrix $p\boldsymbol{B}^T\boldsymbol{B}$ that is added to $\boldsymbol{K}_d$ is equivalent to discrete springs between the interface nodes. In particular if $\boldsymbol{B}$ is the Boolean localization matrix for matching grids, the term $p\boldsymbol{B}^T\boldsymbol{B}$ is diagonal and represents the fictitious springs connecting the d.o.f. of two matching nodes (Fig. 2.5). While in practice the penalty factor p is high in order to obtain a significant "convexification" of $\mathcal{L}^a$, the definite positivity of matrix $(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})$ is improved but the conditioning of the system of equations (2.58a–2.58b) deteriorates because of the difference of order of magnitude between the internal and boundary stiffnesses and between the augmented stiffness matrix and the constraint matrices $\boldsymbol{B}$. This latter difficulty can easily be overcome by introducing a scaling factor ρ for the Lagrange multipliers such that $\boldsymbol{\lambda} = \rho\overline{\boldsymbol{\lambda}}$. The augmented system finally writes

$$(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})\boldsymbol{u}_d + \rho\boldsymbol{B}^T\overline{\boldsymbol{\lambda}} = \boldsymbol{f}_d \tag{2.59a}$$

$$\rho\boldsymbol{B}\boldsymbol{u}_d = \boldsymbol{0} \tag{2.59b}$$

Let us stress once more that the solution of the augmented system is identical to the solution of the hybrid system (2.44). A highly interesting consequence of the penalization is the dramatic speed-up of iterative solution schemes applied to (2.59). To illustrate this point, two iterative algorithms relative to the augmented system will shortly be discussed.

Usawa iterations

A first highly effective iteration method can be found if we iterate on the Lagrange multipliers. Assuming that the augmented stiffness matrix $(\boldsymbol{K}_d+p\boldsymbol{B}^T\boldsymbol{B})$ has been factored and eliminating the displacement unknowns in the augmented system, the interface problem relative to the Lagrange multiplier is given by

$$\rho\tilde{\boldsymbol{F}}\boldsymbol{\lambda} = \rho\boldsymbol{B}(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{f}_d \tag{2.60}$$

$$\text{where} \qquad \tilde{\boldsymbol{F}} = \left(\boldsymbol{B}(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\right) \tag{2.61}$$

Let us note that in general the block diagonal matrix $\boldsymbol{K}_d$ is singular, the reason being that some of the substructures of the model may be floating and their stiffness matrix may thus be singular. However the augmented matrix can be assumed to be regular since it corresponds to the global stiffness matrix of the substructured problem with springs connecting the interfaces. From this we deduce that the penalty term also regularizes the subdomains equilibrium problem.

Based on (2.60) we define the simple iteration scheme

$$\boldsymbol{\lambda}_{k+1} = \left(\boldsymbol{I} - \rho\tilde{\boldsymbol{F}}\right)\boldsymbol{\lambda}_k - \rho\boldsymbol{B}(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{f}_d \tag{2.62a}$$

$$= \boldsymbol{\lambda}_k + \rho\boldsymbol{B}(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})^{-1}\left(\boldsymbol{f}_d - \boldsymbol{B}^T\boldsymbol{\lambda}_k\right) \tag{2.62b}$$

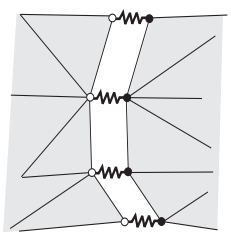


Figure 2.5: Fictitious springs in augmented systems

In general, the scaling factor ρ is taken equal to p. For a thorough discussion on the choice of ρ and on the convergence of the iteration scheme, the reader is referred to [94]. If the penalization factor p is high enough, the convergence is very rapid. Indeed, it can be shown that [35]- [94, Chap.3]

$$\lim_{p\to\infty} p\tilde{\boldsymbol{F}}\boldsymbol{B} = \lim_{p\to\infty} \left(\boldsymbol{B}(\frac{1}{p}\boldsymbol{K}_d + \boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T \right) \boldsymbol{B} = \boldsymbol{B} \tag{2.63}$$

meaning that the condition number of the iteration operator tends to 1 in the range of $\boldsymbol{B}$ as $p \to \infty$.

The rapid convergence of this iteration method can also be understood by focusing on its mechanical interpretation: assuming for the initial step that $\boldsymbol{\lambda}_0$ is null, the first iteration leads to taking as interface stresses $\boldsymbol{\lambda}_1$ the forces applied by the fictitious springs onto the interface nodes when the external forces are applied on the system. After the first iteration, the Lagrange multipliers $\boldsymbol{\lambda}_1$ are a good estimation of the exact connecting forces and the higher the penalization factor p, the better the approximation of $\boldsymbol{\lambda}$: indeed, if $p \to \infty$, the fictitious springs enforce the exact compatibility. In the next iteration steps, the corrections to $\boldsymbol{\lambda}$ are given by the stresses in the fictitious springs when $\boldsymbol{\lambda}_k$ is applied on the interface.

This iterative scheme is of the same type as the so-called Usawa's iteration [2, 119] which is characterized by iterations on the Lagrange multipliers while solving for the primal variables explicitly within each iteration. When applied to the augmented hybrid system, the algorithm (2.62) is highly efficient, although it remains to properly choose the penalty factor. For the iteration scheme to convergence rapidly, p should tend to ∞, but if p is to high the numerical condition of the augmented matrix $\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B}$ deteriorates: if the stiffness of the fictitious springs is excessive, the stiffnesses of the augmented system will be poorly balanced.

If we set $p = 0$, the augmented Lagrangian method degenerates into the μ-method and the iteration scheme (2.62) must be improved by introducing a Conjugate Gradient algorithm. The term $\boldsymbol{B}^T\boldsymbol{B}$ needs not to be considered in this case, but the local stiffness matrices are no longer regularized so that special care must be taken for the definition of a consistent subdomain equilibrium if the model includes floating substructures. These issues will be discussed in Chapter 3.

Three-field iteration scheme

The penalization introduces an explicit coupling between the substructures d.o.f. via the interface nodes: the matrix $(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})$ is no longer block diagonal which undermines any concurrent solution scheme. A variant method better suited to parallel computation was proposed in [94,175]. The main idea is to introduce a third reference displacement field on Γ_I so that the penalization is not explicitly introduced between the substructures, but it is defined for the compatibility constraints relative to the intermediate field $u_{\Gamma_I^{sr}}$. Considering the case of two subdomains and assuming that the constraints between the substructure displacements and the reference field are given by

$$\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{u}|_{\Gamma_I} \qquad\qquad s = 1, 2 \tag{2.64}$$

the augmented Lagrangian function for the three-field problem is

$$\begin{aligned} \mathcal{L}^a(\boldsymbol{v}^{(s)}, \boldsymbol{v}|_{\Gamma_I}, \boldsymbol{\mu}^{sr}) \;=\; & \sum_{s=1}^{2} \left(\frac{1}{2}\boldsymbol{v}^{(s)^T}\boldsymbol{K}^{(s)}\boldsymbol{v}^{(s)} - \boldsymbol{v}^{(s)^T}\boldsymbol{f}^{(s)} \right) \\ & + \sum_{\substack{s=1 \\ r\neq s}}^{2} \mu^{s-r^T} \left(\boldsymbol{C}^{(s)}\boldsymbol{v}^{(s)} - \boldsymbol{v}|_{\Gamma_I} \right) \\ & + \frac{p}{2} \sum_{s=1}^{2} \left(\boldsymbol{C}^{(s)}\boldsymbol{v}^{(s)} - \boldsymbol{v}|_{\Gamma_I} \right)^T \left(\boldsymbol{C}^{(s)}\boldsymbol{v}^{(s)} - \boldsymbol{v}|_{\Gamma_I} \right) \end{aligned} \tag{2.65}$$

The stationarity of $\mathcal{L}^a$ with respect to $\boldsymbol{v}^{(s)}$, $\boldsymbol{v}|_{\Gamma_I}$ and $\boldsymbol{\mu}^{sr}$ yields

$$\left(\boldsymbol{K}^{(s)} + p\boldsymbol{C}^{(s)^T}\boldsymbol{C}^{(s)}\right)\boldsymbol{u}^{(s)} - p\boldsymbol{C}^{(s)^T}\boldsymbol{u}|_{\Gamma_I} + \boldsymbol{C}^{(s)^T}\boldsymbol{\lambda}^{s-r} = \boldsymbol{f}^{(s)} \tag{2.66a}$$

$$-\sum_{s=1}^{2} p\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)} + 2p\boldsymbol{u}|_{\Gamma_I} - \sum_{\substack{s=1\\ r\neq s}}^{2}\boldsymbol{\lambda}^{s-r} = \boldsymbol{0} \tag{2.66b}$$

$$\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)} - \boldsymbol{u}|_{\Gamma_I} = \boldsymbol{0} \qquad s = 1,2 \tag{2.66c}$$

The first equations (2.66a) are the augmented equilibrium equations of the substructures and we note again that the regularization term $\boldsymbol{C}^{(s)^T}\boldsymbol{C}^{(s)}$ allows to define the inverse of the augmented stiffness. The second set of equations (2.66b) expresses the equilibrium of the Lagrange multipliers acting on the reference field: if we substitute the compatibility constraints (2.66c) in (2.66b) we find $\boldsymbol{\lambda}^{1-2} + \boldsymbol{\lambda}^{2-1} = 0$.

Applying the same reasoning as for the development of the iteration scheme (2.62), we can define the following recurrent relations

$$\boldsymbol{\lambda}^{s-r}_{k+1} = \boldsymbol{\lambda}^{s-r}_{k} + p\left(\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)}_k - \boldsymbol{u}|_{\Gamma_I,k}\right) \tag{2.67}$$

where $\boldsymbol{u}^{(s)}_k$ and $\boldsymbol{u}|_{\Gamma_I,k}$ are solution of the system (2.66a–2.66b) with $\boldsymbol{\lambda}^{s-r} = \boldsymbol{\lambda}^{s-r}_k$. Unfortunately, the solution of this subsystem introduces a coupling between all subdomains as in the iteration scheme (2.62). To avoid this coupling, the subsystem (2.66a–2.66b) will be solved by a block relaxation: for $l = 1, \ldots,$

$$\left(\boldsymbol{K}^{(s)} + p\boldsymbol{C}^{(s)^T}\boldsymbol{C}^{(s)}\right)\boldsymbol{u}^{(s)}_{k,l} = p\boldsymbol{C}^{(s)^T}\boldsymbol{u}|_{\Gamma_I,k,l-1} + \boldsymbol{C}^{(s)^T}\boldsymbol{\lambda}^{sr}_k + \boldsymbol{f}^{(s)} \tag{2.68a}$$

$$2p\boldsymbol{u}|_{\Gamma_I,k,l} = \sum_{s=1}^{2} p\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)}_{k,l} + \sum_{\substack{s=1\\ r\neq s}}^{2}\boldsymbol{\lambda}^{s-r}_k \tag{2.68b}$$

If these iterations are limited to a single step and if the solution of the first equation (2.68a) is used to define an intermediate estimation of the Lagrange multipliers, the overall iteration scheme amounts to

$$\left(\boldsymbol{K}^{(s)} + p\boldsymbol{C}^{(s)^T}\boldsymbol{C}^{(s)}\right)\boldsymbol{u}^{(s)}_k = p\boldsymbol{C}^{(s)^T}\boldsymbol{u}|_{\Gamma_I,k-1} + \boldsymbol{C}^{(s)^T}\boldsymbol{\lambda}^{sr}_k + \boldsymbol{f}^{(s)} \tag{2.69a}$$

$$\boldsymbol{\lambda}^{s-r}_{k+\frac{1}{2}} = \boldsymbol{\lambda}^{s-r}_{k} + p\left(\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)}_k - \boldsymbol{u}|_{\Gamma_I,k-1}\right) \tag{2.69b}$$

$$2p\boldsymbol{u}|_{\Gamma_I,k} = \sum_{s=1}^{2} p\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)}_k + \sum_{\substack{s=1\\ r\neq s}}^{2}\boldsymbol{\lambda}^{s-r}_{k+\frac{1}{2}} \tag{2.69c}$$

$$\boldsymbol{\lambda}^{s-r}_{k+1} = \boldsymbol{\lambda}^{s-r}_{k+\frac{1}{2}} + p\left(\boldsymbol{C}^{(s)}\boldsymbol{u}^{(s)}_k - \boldsymbol{u}|_{\Gamma I,k}\right) \tag{2.69d}$$

The convergence analysis and the improvements of this algorithm are discussed in [94, 175].

2.3.4 Penalty method and perturbed Lagrangian functions

The μ-method and the augmented Lagrange method have proved to be very effective in practical applications. However introducing dual variables increases the number of unknowns in the problem and requires that the solution scheme be modified. So let us assume that the Lagrange multipliers

in the augmented Lagrangian function (2.57) are set to zero. Only the penalty term remains and we obtain

$$\mathcal{L}^p(\boldsymbol{v}^{(s)}) = \frac{1}{2}\boldsymbol{v}_d^T\boldsymbol{K}_d\boldsymbol{v}_d - \boldsymbol{v}_d^T\boldsymbol{f}_d + \frac{p}{2}\boldsymbol{v}_d^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{v}_d \tag{2.70}$$

The solution is then given by minimizing $\mathcal{L}^p$ which is equivalent to the solution of a non constrained problem. The quadratic penalty term prescribes a high cost to non compatible solutions, but the solution will satisfy the compatibility constraints only for $p \to \infty$. From a mechanical view point, the substructures are linked only by inter-subdomain springs of stiffness p. While the penalty factor p can only be given a value limited by the deterioration of the conditioning of the augmented structural operator, the compatibility will be satisfied in an approximate manner. When applied to real assembly problems, this technique has shown to be difficult to tune properly [132] although it remains popular in the domain of contact analysis [168].

Let us consider once more the basic μ-method. A variant of the μ-method can be found if we introduce a regularization term for the hybrid system of equation. To this end we define a perturbed Lagrangian function by adding to $\mathcal{L}$ a term quadratic in the Lagrange multipliers

$$\mathcal{L}^{per}(\boldsymbol{v}^{(s)}, \boldsymbol{\mu}^{sr}) = \frac{1}{2}\boldsymbol{v}_d^T\boldsymbol{K}_d\boldsymbol{v}_d - \boldsymbol{v}_d^T\boldsymbol{f}_d + \boldsymbol{v}_d^T\boldsymbol{B}^T\boldsymbol{\mu} - \frac{\varepsilon}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} \tag{2.71}$$

where ε is a small positive scalar such that when $\varepsilon \to 0$, $\mathcal{L}^{per} \to \mathcal{L}$. The stationarity condition for $\mathcal{L}^{per}$ leads to

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + {\boldsymbol{B}^{(s)}}^T\boldsymbol{\lambda} = \boldsymbol{f}^{(s)} \qquad s = 1, \dots N_s \tag{2.72a}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)} - \varepsilon\boldsymbol{\lambda} = \boldsymbol{0} \tag{2.72b}$$

While the system of equations arising from a pure μ-method contains null diagonal terms for the Lagrange multipliers and might therefore require special strategies when the entire system is factored, the system of equation above is regularized in the sense that it adds non-zero contribution of the Lagrange multipliers to the compatibility constraints. The exact compatibility will be obtained only when $\varepsilon \to 0$. When applied for assembling finite elements meshings, the perturbed Lagrangian method defines the compatibility within a tolerance depending on ε and also on the node distribution [132].

Rearranging Eq. (2.72b) we see that the Lagrange multipliers are related to the compatibility error by

$$\boldsymbol{\lambda} = \frac{1}{\varepsilon}\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)} = \frac{1}{\varepsilon}\boldsymbol{B}\boldsymbol{u}_d \tag{2.73}$$

and substituting this result in the definition of the perturbed Lagrangian functional (2.71), we find

$$\begin{aligned} \mathcal{L}^{per}(\boldsymbol{v}^{(s)}, \boldsymbol{\lambda}^{sr}) &= \frac{1}{2}\boldsymbol{v}_d^T\boldsymbol{K}_d\boldsymbol{v}_d - \boldsymbol{v}_d^T\boldsymbol{f}_d + \frac{1}{2\varepsilon}\boldsymbol{v}_d^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{v}_d \\ &= \mathcal{L}^p\left(p = \frac{1}{2\varepsilon}\right) \end{aligned} \tag{2.74}$$

We thus conclude that the perturbed Lagrangian formulation comes down to a penalty method. One interesting feature however of the perturbed method is that the constraints can be consistently defined by introducing a Lagrange multiplier field.

Iteration schemes based on the penalty method for linear and non-linear problems are discussed in [18]. In particular for contact problems where the constraints matrix $\boldsymbol{B}$ changes according to the contact state, a special technique taking advantage of the Sherman-Morisson formula can be devised in order to get a cheap evaluation of $(\boldsymbol{K}_d + p\boldsymbol{B}^T\boldsymbol{B})^{-1}$ when $\boldsymbol{K}_d$ has been factored [168].

Finally we note that the penalty and perturbed Lagrangian methods can also be applied in a straightforward manner to enforce the compatibility constraints relative to a third reference displacement field [168]: the system of equations are then similar to (2.66a,2.66b) with $\boldsymbol{\lambda}^{s-r} = \boldsymbol{0}$.

2.3.5 Explicit elimination of constraints

The compatibility constraints arising from the assembly procedure are equalities. They may thus be directly used to define a set of independent d.o.f. The assembled equations are then derived from the variational principles applied to the admissible d.o.f. In doing so, no supplementary dual variables are needed as in the Lagrangian type methods, and the constraints are exact.

The simplest way to eliminate the compatibility constraints appears for the collocation methods [8]. When a reference field is defined on an interface Γ_I^{sr} between $\Omega^{(s)}$ and $\Omega^{(r)}$, the collocation formulation of the interface compatibility leads to the constraints (2.3)

$$\boldsymbol{u}_b^{(s)} = \boldsymbol{C}^{(s)} \boldsymbol{u}_{\Gamma_I} \tag{2.3}$$

which also hold in the multiple substructure case if we assume that $\boldsymbol{u}|_{\Gamma_I}$ collects all the reference interface displacements. Calling $\boldsymbol{u}_i^{(s)}$ the internal d.o.f. of $\Omega^{(s)}$, the set of variables $[\dots \boldsymbol{u}_i^{(s)} \dots \boldsymbol{u}|_{\Gamma_I}]$ can be chosen as the independent and admissible d.o.f. Assuming that the boundary d.o.f. of every substructure are numbered last, the energy functional for the substructured system can be written in the following unconstrained form:

$$\begin{aligned}
&\mathcal{J}(\dots \boldsymbol{u}_i^{(s)} \dots \boldsymbol{u}|_{\Gamma_I}) = \\
&\quad \sum_{s=1}^{N_s} \frac{1}{2} \begin{bmatrix} \boldsymbol{u}_i^{(s)^T} & (\boldsymbol{C}^{(s)} \boldsymbol{u}|_{\Gamma_I})^T \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{ii}^{(s)} & \boldsymbol{K}_{ib}^{(s)} \\ \boldsymbol{K}_{bi}^{(s)} & \boldsymbol{K}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(s)} \\ \boldsymbol{C}^{(s)} \boldsymbol{u}|_{\Gamma_I} \end{bmatrix} \\
&\quad - \sum_{s=1}^{N_s} \begin{bmatrix} \boldsymbol{u}_i^{(s)^T} & (\boldsymbol{C}^{(s)} \boldsymbol{u}|_{\Gamma_I})^T \end{bmatrix} \boldsymbol{f}^{(s)}
\end{aligned} \tag{2.75}$$

where the subscripts i and b respectively refer to the substructures interior and boundary. The assembled system of equation is then found be minimizing $\mathcal{J}$with respect to $\boldsymbol{u}_i^{(s)}$ and $\boldsymbol{u}|_{\Gamma_I}$:

$$\begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{ib}^{(1)} \boldsymbol{C}^{(1)} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{ii}^{(N_s)} & \boldsymbol{K}_{ib}^{(N_s)} \boldsymbol{C}^{(N_s)} \\ \boldsymbol{C}^{(1)^T} \boldsymbol{K}_{bi}^{(1)} & \dots & \boldsymbol{C}^{(N_s)^T} \boldsymbol{K}_{bi}^{(N_s)} & \sum_s \boldsymbol{C}^{(s)^T} \boldsymbol{K}_{bb}^{(s)} \boldsymbol{C}^{(s)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \vdots \\ \boldsymbol{u}_i^{(N_s)} \\ \boldsymbol{u}|_{\Gamma_I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \vdots \\ \boldsymbol{f}_i^{(N_s)} \\ \sum_s \boldsymbol{C}^{(s)^T} \boldsymbol{f}_b^{(s)} \end{bmatrix} \tag{2.76}$$

The diagonal term associated with the interface displacements $\boldsymbol{u}|_{\Gamma_I}$ is the sum of the mappings of the boundary stiffnesses from the subdomains boundary d.o.f. to the reference interface displacements.

In the general case when the compatibility constraints are expressed in the form

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{B} \boldsymbol{u}_d = \boldsymbol{0} \tag{2.77}$$

the direct elimination is not always straightforward.

A first approach is to compute the null space of $\boldsymbol{B}$ in order to define a minimum and independent set of admissible d.o.f. If we call $\boldsymbol{B}^*$ the null space of $\boldsymbol{B}$, i.e.

$$\boldsymbol{B} \boldsymbol{B}^* = \boldsymbol{0} \tag{2.78}$$

we can define a set of admissible d.o.f. $\tilde{\boldsymbol{u}}$ by

$$\boldsymbol{u}_d = \boldsymbol{B}^* \tilde{\boldsymbol{u}} \tag{2.79}$$

and the associated assembled stiffness matrix is

$$\boldsymbol{K} = \boldsymbol{B}^{*^T} \boldsymbol{K}_d \boldsymbol{B}^* \tag{2.80}$$

The dimension $\tilde{n}$ of $\tilde{\boldsymbol{u}}$ is the maximum number of independent d.o.f. that can be defined for the assembled system. This technique has been applied in [30] for constrained equations arising from multibody simulation.

In general the computation of the null space of $\boldsymbol{B}$ is not cheap. In the particular case of matching interfaces, we have shown in section 2.1.2 that $\boldsymbol{B}$ is obtained by computing the null space of the assembly Boolean matrix and Eq. (2.21) can be written as

$$\boldsymbol{B}\boldsymbol{L} = \boldsymbol{0} \tag{2.81}$$

Hence we have $\boldsymbol{B}^* = \boldsymbol{L}$ showing that, for conforming interfaces, the null space of $\boldsymbol{B}$ can be found by inspection and corresponds to the assembly Boolean matrix. The explicit elimination of the constraints is then equivalent to the finite element assembly method expressed by (2.18).

We note that the expressions (2.18) and (2.80) are at the heart of the primal domain decomposition methods such as the Schur Complement method [175]. Indeed, if the internal d.o.f. are condensed on the interface boundary and calling $\boldsymbol{b}^{(s)*}$ the submatrix of $\boldsymbol{B}^*$ associated with the boundary d.o.f., the assembled stiffness condensed on the interface is given by

$$\boldsymbol{S} = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)*^T} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)*} \tag{2.82}$$

where

$$\boldsymbol{S}_{bb}^{(s)} = \boldsymbol{K}_{bb}^{(s)} - \boldsymbol{K}_{bi}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} \tag{2.83}$$

is the Schur complement of subdomain $\Omega^{(s)}$. For matching subdomains, $\boldsymbol{b}^*$ is the assembly Boolean matrix of the interface d.o.f. and (2.82) is the primal interface operator as defined in [175].

A second approach for explicitly eliminating the constraints is found by constructing the symmetric projection operator $\boldsymbol{P}$

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{B}^T \left(\boldsymbol{B}\boldsymbol{B}^T \right)^{-1} \boldsymbol{B} \tag{2.84}$$

which projects onto the subspace of admissible displacements since

$$\boldsymbol{B}\boldsymbol{P} = \boldsymbol{0} = \boldsymbol{P}\boldsymbol{B}^T \tag{2.85}$$

It is assumed that $\boldsymbol{B}^T$ has full column rank. We thus define the admissible d.o.f.

$$\boldsymbol{u}_d = \boldsymbol{P}\tilde{\boldsymbol{u}}_d \tag{2.86}$$

the subscript d indicating that $\tilde{\boldsymbol{u}}_d$ has the same dimension as $\boldsymbol{u}_d$. Introducing this definition in the variational principle leads to the system

$$\boldsymbol{P}\boldsymbol{K}_d\boldsymbol{P}\ \tilde{\boldsymbol{u}}_d = \boldsymbol{P}\boldsymbol{f}_d \tag{2.87}$$

Obviously $\tilde{\boldsymbol{u}}_d$ is not a minimum set of variables and the solution of (2.87) is not unique since $\boldsymbol{P}\boldsymbol{K}_d\boldsymbol{P}$ is singular and $\boldsymbol{B}^T$ is its null space. Nevertheless (2.87) has a solution since $\boldsymbol{P}\boldsymbol{f}_d$ is in the space orthogonal to $\boldsymbol{B}^T$. Hence the solution can be written as $\tilde{\boldsymbol{u}}_d + \boldsymbol{B}^T\boldsymbol{\alpha}$ and $\boldsymbol{u} = \boldsymbol{P}\tilde{\boldsymbol{u}}_d$ showing that the final displacement solution is independent of the components lying in the range of $\boldsymbol{B}^T$ and is thus unique.

The projection technique is not efficient when a direct elimination of $\boldsymbol{P}\boldsymbol{K}_d\boldsymbol{P}$ is applied: the number of d.o.f. is not minimum and, although setting-up $\boldsymbol{P}$ is straightforward for matching grids

($\boldsymbol{B}$ is Boolean), the explicit computation of $\boldsymbol{P}\boldsymbol{K}_d\boldsymbol{P}$ is very demanding especially in terms of storage. Moreover special algorithmic strategies must be implemented in order to deal with the high order of singularity of the projected diagonal stiffness matrix. Nevertheless, if the projected system is solved iteratively for instance with a Conjugate Gradient algorithm, the projected diagonal stiffness matrix needs not be constructed and the multiplication by this operator is a naturally parallel task.

Note that the iterative algorithm based on (2.87) is also related to the so-called Primal Schur method (see Part II).

2.4 Discussion of the assembly methods

The discretization of the interface compatibility requirements and the mathematical formulations of the constrained equations are numerous and sometimes very different from the point of view of the modeling error and algorithmic efficiency. In this section we will discuss these issues keeping in mind that we are looking for assembly techniques applicable to matching and non-matching interfaces, which have a potential for defining a reduced set of interface connectors and which are well suited to designing parallel solution strategies.

First, let us remind that the nodal collocation method and its discrete least squares variant do not pass the patch test for general non-matching grids: as explained in 2.3.2, the compatibility is enforced only point-wise and connecting forces are defined in a discrete sense so that, in this case, constant strains cannot be correctly transmitted across an interface. Collocation formulations satisfy the patch test only in the particular case of non-matching grids originating from the refinement of matching meshes and also for matching grids. For those reasons, although the collocation methods are used in commercial codes such as MSc/Nastran [8], we will favor the continuous least-squares formulations which are equivalent to distributed Lagrange multiplier methods (see section 2.3.2): they are consistent with the general theory of hybrid finite elements and are flexible for defining the interface connectors. This is also the method applied in the finite element code SAMCEF [132].

The methods summarized in this chapter can all be adapted to a three-field formulation, namely when a reference displacement field is defined on the interface and when the compatibility is expressed with respect to it. This reference field can then be taken as the uniquely defined interface displacement field, while the substructure displacements may be different. An interesting application where the third field is important happens to be in contact analysis: if the facing substructures on a contact surface are modeled by finite elements, the normal to the contact surface which defines the contact conditions may not be uniquely defined at the nodes whereas if a smooth reference field is introduced, a reference normal direction can be constructed and the gap function can be properly defined [168].

Another advantage of the three-field formulation appears when the node distribution on the interface is very irregular: the interface problem is generally ill-conditioned and defining the compatibility in terms of reference displacements $\boldsymbol{u}|_{\Gamma_I}$ can improve this conditioning. We should also note that the three field formulation leads to a parallel iterative algorithm associated with the augmented Lagrangian method (section 2.3.3).

On the other hand, making use of a third interface displacement field increases the number of interface unknowns and if for instance a Lagrange multiplier method is applied, two independent connecting force distributions must be introduced in addition to the third field. This implies an increase of the computational cost when solving for the interface variables. It is also noteworthy to observe that a three-field formulation is algebraically equivalent to a two displacement field method: if the displacement field in $\Omega^{(r)}$ and the Lagrange multipliers $\boldsymbol{\lambda}^{r-s}$ on one side of the reference displacement are condensed in the system of equations (2.48) we find a system similar to (2.44) which expresses the compatibility between the $\boldsymbol{u}|_{\Gamma_I}$ and the displacements on $\Omega^{(s)}$. The structural matrix associated with $\boldsymbol{u}|_{\Gamma_I}$ is then equivalent to the stiffness matrix of a hybrid finite element defined on $\partial\Omega^{(r)}$.

For our applications, we will assume that no reference field is defined which facilitates the study of the assembling techniques. However, the developments presented in this work can be generalized to the three-field formulation so that our choice is not really restrictive.

Explicit elimination methods for a non-conforming interface are difficult to implement since they require computing a null space or constructing a projection operator. For matching interfaces, enforcing the compatibility exactly by the Boolean localization matrices is equivalent to classical assembly: it does not allow any weakening of the compatibility.

For those reasons and because Lagrange multiplier methods lead to a nice dual expression of the interface problem which is well-suited to iterative schemes and thus to parallelization as will be seen later, the constraints will not be handled by direct elimination methods. Mathematical formulations using a penalization term will also be discarded: the penalization and the perturbed Lagrangian methods can enforce the compatibility only up to a certain tolerance which depends on the penalty factor and which, for numerical reasons, can not be taken as high as wanted. The augmented Lagrangian method leads to exact compatibility but as the other methods with a penalty term, it introduces explicit coupling between neighboring subdomains which is an obstacle to concurrent computation. For the particular case when the augmented Lagrangian method is applied to the three-field formulation, a naturally parallel algorithm can be found but there remains the expense of supplementary interface unknowns relative to the three fields on the interface. More work should be pursued to investigate the actual parallel efficiency of this latter strategy.

Finally, let us note that all the Lagrange multiplier methods summarized in this chapter also apply when the subdomains are overlapping: Lagrange multipliers can be used to define the compatibility over the entire overlap region whether it is conforming or not [112, 115].

In conclusion we will choose for this work the Lagrange multiplier method applied to the continuous compatibility constraint when the interface is non-conforming or whenever a relaxed compatibility is looked after. When exact compatibility is required for matching grids, we will apply the Lagrange multiplier method to the discrete nodal matching conditions.

Chapter 3

Two-field hybrid method for the analysis of substructured systems

In this section the two-field hybrid method presented in the previous section 2.3.2 is reconsidered and the associated interface problem is outlined in more detail.

3.1 Saddle point formulation of hybrid methods

The Lagrangian multiplier method applied to the assembling of substructured systems requires the stationarity of the two-field Lagrangian function (2.40)

$$\mathcal{L}(v^{(1)}, v^{(2)}, \dots v^{(N_s)}, \mu^{12}, \dots) = \sum_{s=1}^{N_s} \mathcal{J}(v^{(s)})_{\Omega^{(s)}} + \sum_{s=1}^{N_s} \sum_{\substack{\Gamma_I^{sr} \neq \emptyset \\ s<r}} \int_{\Gamma_I^{sr}} \mu^{sr} \left(v^{(s)}|_{\Gamma_I^{sr}} - v^{(r)}|_{\Gamma_I^{sr}} \right) d\Gamma_I^{sr} \tag{2.40}$$

where the local energy functions are defined by (2.39)

$$\mathcal{J}(v^{(s)})_{\Omega^{(s)}} = \frac{1}{2} a(v^{(s)}, v^{(s)})_{\Omega^{(s)}} - l(v^{(s)})_{\Omega^{(s)}} \tag{2.39}$$

For linear or linearized mechanical problems, a represents the internal deformation energy and l is the energy of the external forces[1]. The compatibility of the local displacement fields $v^{(s)}$ with respect to the Dirichlet boundary conditions on $\partial\Omega_D^{(s)}$ is an essential condition for (2.40) whereas the interface compatibility is a natural condition for $\mathcal{L}$. The variational calculus applied on (2.40) yields the local equilibrium equations (2.41a) and the continuous interface compatibility conditions (2.41d). The Lagrange multiplier field corresponds to the dual field of inter-subdomain connecting forces.

The solution $(u^{(s)},\ \lambda)$ satisfying the stationarity of $\mathcal{L}$ is a saddle point defined by [18]

$$\mathcal{L}(u^{(s)}, \mu) \leq \mathcal{L}(u^{(s)}, \lambda) \leq \mathcal{L}(v^{(s)}, \lambda) \tag{3.1}$$

for all admissible $v^{(s)}$ and μ. The left-hand inequality requires that the term associated with μ in $\mathcal{L}$ be null which yields the interface compatibility. The inequality on the right-hand side expresses the classical minimum property of the total energy in each subdomain when the exact interface forces are applied.

[1] For a discussion on the two-field hybrid formulation for non-symmetric operators, see [182]

Introducing a finite element discretization in the subdomains and defining distribution functions for the Lagrange multipliers, the Lagrangian functional writes

$$\mathcal{L}(\boldsymbol{v}^{(s)},\boldsymbol{\mu}) = \sum_{s=1}^{N_s}\left(\frac{1}{2}\boldsymbol{v}^{(s)^T}\boldsymbol{K}^{(s)}\boldsymbol{v}^{(s)} - \boldsymbol{v}^{(s)^T}\boldsymbol{f}^{(s)}\right) + \boldsymbol{\mu}^T\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{v}^{(s)} \tag{3.2}$$

where $\boldsymbol{v}^{(s)}$ are the local d.o.f., $\boldsymbol{\mu}$ are the Lagrange multiplier parameters, $\boldsymbol{K}^{(s)}$ and $\boldsymbol{f}^{(s)}$ are the usual finite element stiffness matrices and load vectors, and $\boldsymbol{B}^{(s)}$ are the constraints matrices computed by integrating the product $\boldsymbol{\Lambda}^{sr^T}\boldsymbol{N}^{(s)}$ as in (2.46). When exact compatibility is enforced for matching grids, $\boldsymbol{B}^{(s)}$ are Boolean matrices and the Lagrange multipliers are the connecting nodal forces. The stationarity condition then leads to the algebraic system of equations

$$\begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_d \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_d \\ \boldsymbol{0} \end{bmatrix} \tag{3.3}$$

where the d subscript stands as before for the block diagonal operators and vectors, and $\boldsymbol{B} = [\boldsymbol{B}^{(1)} \ \ldots \ \boldsymbol{B}^{(N_s)}]$ is the global constraints matrix.

The hybrid principle expressed by the Lagrangian function (3.2) can be transformed into a kinematically admissible or a statically admissible formulation. If we assume that the local d.o.f. $\boldsymbol{v}^{(s)}$ satisfy the interface compatibility, that is $\boldsymbol{v}_d = \boldsymbol{B}^*\tilde{\boldsymbol{v}}$ where $\boldsymbol{B}^* = \text{null}(\boldsymbol{B})$, the Lagrangian functional becomes

$$\mathcal{L}(\tilde{\boldsymbol{v}}) = \frac{1}{2}\tilde{\boldsymbol{v}}^T\ \boldsymbol{B}^{*^T}\boldsymbol{K}_d\boldsymbol{B}^*\tilde{\boldsymbol{v}} - \tilde{\boldsymbol{v}}^T\boldsymbol{B}^{*^T}\boldsymbol{f}_d \tag{3.4}$$

where $\tilde{\boldsymbol{v}}$ is a set of independent d.o.f., $\boldsymbol{B}^{*^T}\boldsymbol{K}_d\boldsymbol{B}^*$ is the assembled stiffness matrix and $\boldsymbol{B}^{*^T}\boldsymbol{f}_d$ is the assembled force vector. The Lagrangian function comes down to the energy functional of a kinematically admissible principle. We see that the Hermitian matrix of $\mathcal{L}(\tilde{\boldsymbol{v}})$ is the assembled stiffness matrix which is positive definite (or positive semi-definite in case the structure has rigid body modes). The displacement solution thus corresponds to the *minimum* of $\mathcal{L}$ with respect to the displacements that satisfy the interface compatibility. This corresponds to the right-hand side inequality in (3.1).

If we assume that the subdomains equilibrium is verified, the first equations of the system (3.3) yields

$$\boldsymbol{v}_d = \boldsymbol{K}_d^{-1}\left(\boldsymbol{f}_d - \boldsymbol{B}^T\boldsymbol{\mu}\right) \tag{3.5}$$

where we assume for now that the local stiffness matrices $\boldsymbol{K}^{(s)}$ are regular. The Lagrangian functional becomes

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}) &= \frac{1}{2}\boldsymbol{v}_d^T\left(\boldsymbol{f}_d - \boldsymbol{B}^T\boldsymbol{\mu}\right) - \boldsymbol{v}_d^T\boldsymbol{f}_d + \boldsymbol{\mu}^T\boldsymbol{B}\boldsymbol{v}_d \\
&= -\frac{1}{2}\boldsymbol{v}_d^T\boldsymbol{f}_d + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{B}\boldsymbol{v}_d \qquad (3.6a)\\
&= -\frac{1}{2}\boldsymbol{f}_d^T\boldsymbol{K}_d^{-1}\boldsymbol{f}_d - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{B}\boldsymbol{K}_d^{-1}\boldsymbol{B}^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{B}\boldsymbol{K}_d^{-1}\boldsymbol{f}_d \\
&= Cst - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{F}_I\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{d} \qquad (3.6b)
\end{aligned}$$

where

$$\boldsymbol{F}_I = \boldsymbol{B}\boldsymbol{K}_d^{-1}\boldsymbol{B}^T \tag{3.7}$$

$$\boldsymbol{d} = \boldsymbol{B}\boldsymbol{K}_d^{-1}\boldsymbol{f}_d \tag{3.8}$$

and where Cst is a constant. This statically admissible formulation has as Hermitian matrix $-\boldsymbol{F}_I$. Since the interface flexibility matrix $\boldsymbol{F}_I$ is at least positive semi-definite according to (3.7), we can

state that if the subdomains equilibria are satisfied, the solution of the interface force corresponds to the supremum of $\mathcal{L}$ (see left-hand side of (3.1)). If we then define the energy functionals

$$\mathcal{E}(\boldsymbol{v}_d) = \sup_{\boldsymbol{\mu}} \mathcal{L}(\boldsymbol{v}_d, \boldsymbol{\mu}) \tag{3.9}$$

$$\mathcal{C}(\boldsymbol{\mu}) = \inf_{\boldsymbol{v}_d} \mathcal{L}(\boldsymbol{v}_d, \boldsymbol{\mu}) \tag{3.10}$$

the saddle point principle can be written as

$$\mathcal{L}(\boldsymbol{u}_d, \boldsymbol{\lambda}) = \inf_{\boldsymbol{v}_d} \sup_{\boldsymbol{\mu}} \mathcal{L}(\boldsymbol{v}_d, \boldsymbol{\mu}) = \inf_{\boldsymbol{v}_d} \mathcal{E}(\boldsymbol{v}_d) \tag{3.11}$$

$$= \sup_{\boldsymbol{\mu}} \inf_{\boldsymbol{v}_d} \mathcal{L}(\boldsymbol{v}_d, \boldsymbol{\mu}) = \sup_{\boldsymbol{\mu}} \mathcal{C}(\boldsymbol{\mu}) \tag{3.12}$$

A thorough mathematical analysis of the saddle point problem for the two-field hybrid method was presented in [67].

3.2 The dual interface problem

The hybrid system of equation (3.3) arising from the saddle point problem can be solved by a direct factorization applied onto the entire hybrid operator. This however is not straightforward. Since the operator comes from a saddle point problem, it is no longer definite: the displacements being associated with a positive definite Hessian matrix, the corresponding pivots will be positive (or null if floating modes exist) while the pivots associated with the Lagrange multipliers are negative or null being said that the relevant Hessian matrix is negative semi-definite.

As a consequence, full pivoting must be applied when factorizing the hybrid operator. For instance, if the elimination is first carried out on the $\boldsymbol{K}_d$ block, there will appear as many pivots as there are local rigid body modes of floating subdomains. Pivoting must thus be applied so that null pivots are temporarily rejected at the end. If a block factorization is used, this might be a problem and an a priori renumbering should be applied in order to mix displacement d.o.f. and Lagrange multipliers in the hybrid operator. Note that this renumbering can be obtained as a by-product when minimizing the bandwidth of the hybrid operator.

A different way to solve the hybrid system (3.3) consists in eliminating the displacement d.o.f. and write the interface problem in terms of the Lagrange multipliers. This approach will now be detailed.

3.2.1 The local Neumann problem

Following the reasoning that led to the statically admissible formulation (3.6b), we solve first the independent local equilibrium problems in each subdomain. The local equilibrium equations correspond to local Neumann problems where we assume that the interface loads are known. If we note that some substructures may not have enough Dirichlet boundary conditions to be non-floating when the interface is a Neumann boundary, the local displacements are computed by

$$\boldsymbol{u}^{(s)} = \boldsymbol{K}^{(s)^+} \left(\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} \right) - \boldsymbol{R}^{(s)} \boldsymbol{\alpha}^{(s)} \qquad s = 1, \dots N_s \tag{3.13}$$

where $\boldsymbol{R}^{(s)}$ is the null space of $\boldsymbol{K}^{(s)}$ and $\boldsymbol{\alpha}^{(s)}$ are the associated amplitudes. $\boldsymbol{K}^{(s)^+}$ is the inverse of the stiffness matrix if $\Omega^{(s)}$ is non-floating. In the case of floating subdomains, $\boldsymbol{K}^{(s)}$ is singular: $\boldsymbol{K}^{(s)^+}$ is then a pseudo-inverse that satisfies

$$\boldsymbol{K}^{(s)} \boldsymbol{K}^{(s)^+} \boldsymbol{K}^{(s)} = \boldsymbol{K}^{(s)} \tag{3.14a}$$

$$\boldsymbol{K}^{(s)^+} \boldsymbol{K}^{(s)} \boldsymbol{K}^{(s)^+} = \boldsymbol{K}^{(s)^+} \tag{3.14b}$$

The computation of a pseudo-inverse can be performed by the temporary links method [91] where the lines and columns relative to a zero pivot are zeroed, the rigid body modes $\boldsymbol{R}^{(s)}$ being obtained as a by-product of the factorization.

Another method to handle singular local operators and which is more robust consists in constructing the translation and rotation rigid body modes by simple geometric inspection and in finding the linear combinations of possible rigid body modes that satisfy the Dirichlet boundary conditions [61]. The computation of the pseudo-inverse is then facilitated by knowing a priori the number of zero pivots. This procedure requires special care though for handling the so-called kinematical modes or local mechanisms, that is zero energy modes which are different from the pure rigid body modes and which cannot be detected by simple inspection. In practice however zero pivots associated with kinematical modes are easy to detect during the factorization.

A singular local Neumann problem has a solution only if the right-hand side $\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}$ is in the range of $\boldsymbol{K}^{(s)}$, namely if the external loads and the interface connecting forces are self-equilibrated with respect to the rigid body modes of the subdomain. The amplitude $\boldsymbol{\alpha}^{(s)}$ of the local rigid body modes will thus be determined by the following conditions on the Lagrange multipliers

$$\boldsymbol{R}^{(s)^T}\left(\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} - \boldsymbol{f}^{(s)}\right) = \boldsymbol{0} \qquad s = 1, \dots N_s \tag{3.15}$$

Having defined the general solution for a singular local Neumann problem, let us transform the Lagrangian function as we did earlier in (3.6) to define a statically admissible approach. Substituting the general expression (3.13) in (3.2) and considering the amplitudes $\boldsymbol{\alpha}^{(s)}$ of the local rigid modes as global displacement unknowns, the Lagrangian function writes

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\alpha}) &= \sum_{s=1}^{N_s}\left(\frac{1}{2}\boldsymbol{u}^{(s)^T}\boldsymbol{K}^{(s)}\boldsymbol{K}^{(s)^+}(\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s)^T}\boldsymbol{\mu}) - \boldsymbol{u}^{(s)^T}\boldsymbol{f}^{(s)}\right) + \boldsymbol{\mu}^T\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)} \\
&= \frac{1}{2}\sum_{s=1}^{N_s}\left(-\boldsymbol{f}^{(s)^T}\boldsymbol{K}^{(s)^+}\boldsymbol{f}^{(s)} - \boldsymbol{\mu}^T\boldsymbol{B}^{(s)}\boldsymbol{K}^{(s)^+}\boldsymbol{B}^{(s)^T}\boldsymbol{\mu}\right) \\
&\quad + \sum_{s=1}^{N_s}\boldsymbol{\mu}^T\boldsymbol{B}^{(s)}\boldsymbol{K}^{(s)^+}\boldsymbol{f}^{(s)} - \sum_{s=1}^{N_s}\boldsymbol{\alpha}^{(s)^T}\boldsymbol{R}^{(s)^T}(\boldsymbol{B}^{(s)^T}\boldsymbol{\mu} - \boldsymbol{f}^{(s)})
\end{aligned} \tag{3.16}$$

This result is similar to (3.6b) where we assumed that no floating substructures exist, but here the Lagrangian function does not correspond to a pure statically admissible functional since the rigid body displacements remain in the functional. The stationarity condition on $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\alpha})$ with respect to the $\boldsymbol{\alpha}^{(s)}$ yields the self-equilibrium condition (3.15). Hence the self-equilibrium requirement is a natural condition for the saddle point principle.

3.2.2 The dual interface problem

In order to express the interface problem in the dual form, let us substitute $\boldsymbol{u}^{(s)}$ in the compatibility equations $\boldsymbol{B}\boldsymbol{u}_d = \boldsymbol{0}$ by their expression (3.13) in terms of the Lagrange multipliers. Taking account of the self-equilibrium condition (3.15), the dual interface problem is given by [67]

$$\begin{bmatrix} \boldsymbol{F}_I & \boldsymbol{G}_I \\ \boldsymbol{G}_I^T & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix} \tag{3.17}$$

where

$$\boldsymbol{F}_I = \boldsymbol{B}\boldsymbol{K}_d^+\boldsymbol{B}^T = \sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{K}^{(s)^+}\boldsymbol{B}^{(s)^T} \tag{3.18a}$$

$$\boldsymbol{G}_I = \begin{bmatrix} \boldsymbol{B}^{(1)}\boldsymbol{R}^{(1)} & \dots & \boldsymbol{B}^{(N_f)}\boldsymbol{R}^{(N_f)} \end{bmatrix} \tag{3.18b}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^{(1)^T} & \dots & \boldsymbol{\alpha}^{(N_f)^T} \end{bmatrix}^T \tag{3.18c}$$

$$\boldsymbol{d} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{K}^{(s)^+} \boldsymbol{f}^{(s)} \tag{3.18d}$$

$$\boldsymbol{e} = \left[\begin{array}{cccc} \boldsymbol{f}^{(1)^T} \boldsymbol{R}^{(1)} & \ldots & \boldsymbol{f}^{(N_f)^T} \boldsymbol{R}^{(N_f)} \end{array} \right]^T \tag{3.18e}$$

and N_f is the number of floating substructures. Clearly, the interface problem (3.17) also derives from the stationarity of the Lagrangian function $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\alpha})$ (3.16).

Some techniques have been proposed to regularize the subdomain equations so to avoid the problem of computing the pseudo-inverse matrices and the null space associated with floating substructures. The first one consists in introducing a penalization term (see section 2.3.3). This leads to a coupling of all subdomains and is therefore not considered here.

A different technique proposed in [66] for conforming interfaces regularizes the local Neumann problem by transferring some diagonal stiffness from one interface node to its corresponding node on the interface. The so-called regularized saddle point problem leads to a simplification of the method since it avoids to include the subdomain rigid body modes in the interface problem. However, as will be seen later (Part II), including the local rigid body modes is of crucial importance for an iterative scheme to have good convergence rate when applied to the dual interface problem. Hence if the interface problem is solved by a direct elimination method, regularization might be a good choice, but whenever an iterative method is used the null spaces of the local problems have to be included in the interface system, even though the rigid body modes amplitudes slightly increase the dimension of the interface system (3.17).

Noting n_f the total number of floating modes, the matrix $\boldsymbol{G}_I$ is of dimension $n_\lambda \times n_f$ and it maps the local rigid body modes onto the dual interface space. When the global structure is non floating, one can prove that $\boldsymbol{G}_I$ has full column rank [67]. This issue will be discussed in Section 11.5 of Chap. 11, Part II.

Premultiplying the first set of equations (3.17) by $\boldsymbol{G}_I^T$, we find

$$\boldsymbol{G}_I^T \boldsymbol{G}_I \boldsymbol{\alpha} = \boldsymbol{G}_I^T (\boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}) \tag{3.19}$$

showing that if no global modes of zero energy exist, the amplitudes $\boldsymbol{\alpha}$ are uniquely defined by $\boldsymbol{\lambda}$.

The interface flexibility operator $\boldsymbol{F}_I$ is in general positive semi-definite because at cross-points redundant compatibility constraints can be defined as explained in section 2.2. In that case, there exist non-zero sets $\tilde{\boldsymbol{\Lambda}}$ of Lagrange multipliers such that the associated interface loads are zero:

$$\boldsymbol{B}^T \tilde{\boldsymbol{\Lambda}} = \boldsymbol{0} \tag{3.20}$$

One could choose to work with a non redundant set of compatibility constraints but in practice this is not necessary and even more the redundancies in cross-point compatibility constraints are crucial for the convergence of the iterative schemes applied to the dual problem as will be explained in Part II.

Even though redundant constraints are included, a solution $\boldsymbol{\lambda}$ exists because by the definitions (3.18b,3.18d), $\boldsymbol{d}$ and $\boldsymbol{G}_I$ are in the range of $\boldsymbol{B}$ and hence in the range of $\boldsymbol{F}_I$. This solution is known up to the null space $\tilde{\boldsymbol{\Lambda}}$ of $\boldsymbol{B}^T$ so that the solution writes $\lambda + \tilde{\boldsymbol{\Lambda}}\boldsymbol{\gamma}$. The displacement solution (3.13) is however independent of the $\tilde{\boldsymbol{\Lambda}}$ and is therefore unique (if no global modes of zero energy exist).

The dual interface has a clear mechanical interpretation. If we observe that $\boldsymbol{d}$ is the interface compatibility error, i.e. the displacement gap expressed in the dual space, when the external forces are applied and the Lagrange multiplier are set to zero, the dual interface problem consists in finding the interface forces that close the gap $\boldsymbol{d}$ and which satisfy the self-equilibrium of the floating subdomains. The operator $\boldsymbol{F}_I$ is the interface flexibility matrix relating the interface displacement gap to the inter-subdomain forces. From a mechanical view point, the interface flexibility expressed by (3.18a) corresponds to the assembly of the subdomain flexibilities in the sense that the operator $\boldsymbol{B}$ maps the flexibility in the dual space and the sum yields the displacement

error. The assembly of the subdomain flexibilities is equivalent in a dual way to the assembly of the interface forces when assembling the subdomain stiffnesses to form the assembled stiffness matrix as described by (2.18) or (2.80). Incidentally, let us note the similarities between (2.18) or (2.80) and (3.18a) where $\boldsymbol{B}^*$ or $\boldsymbol{L}$ correspond to $\boldsymbol{B}$ and where the local stiffnesses correspond to the local flexibilities.

The interface flexibility of each subdomain can also be expressed by the pseudo-inverse of the subdomain Schur complement (2.83) so that the assembled interface flexibility writes [67]

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{K}^{(s)^+} \boldsymbol{B}^{(s)^T} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_{bb}^{(s)^+} \end{bmatrix} \boldsymbol{B}^{(s)^T} \tag{3.21}$$

and it becomes obvious that $\boldsymbol{F}_I$ is the dual of the Schur complement operator (2.82). Defining $\boldsymbol{b}^{(s)}$ as the restriction of $\boldsymbol{B}^{(s)}$ to the boundary d.o.f., we can also write

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \boldsymbol{b}^{(s)^T} \tag{3.22}$$

The dual interface problem (3.17) can be solved by direct methods, but this requires to explicitly form the operator $\boldsymbol{F}_I$ and special care must be taken because the system is not definite in general. When the number n_λ of Lagrange multipliers is not small, $\boldsymbol{F}_I$ can not be formed and an iterative scheme such as the Conjugate Gradient algorithm should be used with a projection for handling the self-equilibrium condition (see Part II).

We will now discuss the two-field hybrid method and the dual interface problem for the modal and transient response analysis of substructured problems.

3.3 Modal analysis of substructured systems

The dynamics of a substructured problem is governed by the local dynamic equations which in the discretized form are

$$\boldsymbol{M}^{(s)} \ddot{\boldsymbol{u}}^{(s)}(t) + \boldsymbol{C}^{(s)} \dot{\boldsymbol{u}}^{(s)}(t) + \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s)}(t) + \boldsymbol{B}^{(s)^T} \boldsymbol{u}^{(s)}(t) = \boldsymbol{f}^{(s)}(t) \tag{3.23}$$
$$\boldsymbol{u}^{(s)}(0) \quad \text{and} \quad \dot{\boldsymbol{u}}^{(s)}(0) \qquad \text{given}$$

where t is the time, $\dot{\boldsymbol{u}}^{(s)}$ and $\ddot{\boldsymbol{u}}^{(s)}$ are respectively the first and second order time derivatives of the displacements $\boldsymbol{u}^{(s)}$, and $\boldsymbol{M}^{(s)}$ and $\boldsymbol{C}^{(s)}$ are respectively the mass and damping matrices of substructure $\Omega^{(s)}$. The interface compatibility constraints are then expressed by

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)}(t) = \boldsymbol{0} \tag{3.24}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \dot{\boldsymbol{u}}^{(s)}(t) = \boldsymbol{0} \tag{3.25}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \ddot{\boldsymbol{u}}^{(s)}(t) = \boldsymbol{0} \tag{3.26}$$

Only one of the constraints above has to be satisfied. Indeed, let us assume for instance that the compatibility constraints are imposed on the accelerations $\ddot{\boldsymbol{u}}$ and that the initial displacements $\boldsymbol{u}^{(s)}(0)$ and velocities $\dot{\boldsymbol{u}}^{(s)}(0)$ are compatible, i.e.

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)}(0) = \boldsymbol{0} \tag{3.27}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \dot{\boldsymbol{u}}^{(s)}(0) = \boldsymbol{0} \tag{3.28}$$

The displacement and velocity compatibility at any time t is then deduced from the integration of the acceleration compatibility equation (3.26).

The free vibration analysis of a substructured system assumes that no external forces are applied and that the displacements are harmonic. In the assembled primal form, the free vibrations are governed by the eigenproblem

$$\left(\boldsymbol{K} - \omega^2 \boldsymbol{M}\right) \tilde{\boldsymbol{u}} = \boldsymbol{0} \tag{3.29}$$

where we assume for the sake of simplicity that the system is undamped and where $\tilde{\boldsymbol{u}}$ is a minimum set of compatible d.o.f.. ω is the circular frequency of the free vibration[2]. $\boldsymbol{K}$ and $\boldsymbol{M}$ are the assembled stiffness and mass matrices defined by

$$\boldsymbol{K} = \boldsymbol{B}^{*T} \boldsymbol{K}_d \boldsymbol{B}^* \tag{3.30}$$

$$\boldsymbol{M} = \boldsymbol{B}^{*T} \boldsymbol{M}_d \boldsymbol{B}^* \tag{3.31}$$

$\boldsymbol{B}^*$ being the null space of $\boldsymbol{B}^T$ (see Eqs. (2.78–2.80)).

Using the same block diagonal notations as before, the dual form of the eigenvalue problem for free vibrations writes

$$\left(\begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} - \omega^2 \begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{u}_d \\ \boldsymbol{\lambda} \end{bmatrix} = \boldsymbol{0} \tag{3.32}$$

This hybrid eigenvalue problem is of dimension $n_\lambda + \sum_s n^{(s)}$, where n_λ is the number of interface constraints and $n^{(s)}$ is the number of d.o.f. in $\Omega^{(s)}$. The hybrid matrices in (3.32) are indefinite and it can be shown [75] that the solutions are the $\sum_s n^{(s)} - n_\lambda$ eigenmodes of the assembled problem which we call $(\omega_k, [\boldsymbol{u}_{d,k}^T \; \boldsymbol{\lambda}_k^T])$ and that $2n_\lambda$ supplementary eigensolutions of infinite eigenvalue can be associated with the Lagrange multipliers. Indeed, if we look at the compatibility equations in (3.32) as a dynamic system associated with the Lagrange multipliers but with zero mass, we can find the $2n_\lambda$ solutions [27, 75]

$$(-\infty, \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e}_1 \end{bmatrix}), (+\infty, \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e}_1 \end{bmatrix}), \ldots (-\infty, \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e}_{n_\lambda} \end{bmatrix}), (+\infty, \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e}_{n_\lambda} \end{bmatrix}), \tag{3.33}$$

where $\boldsymbol{e}_i$ defines the unit vector having 1 for its ith term.

The orthonormality relations for the finite eigensolutions associated with the symmetric hybrid eigenproblem (3.32) are

$$\begin{bmatrix} \boldsymbol{u}_{d,k}^T & \boldsymbol{\lambda}_k^T \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{d,l} \\ \boldsymbol{\lambda}_l \end{bmatrix} = \omega_k^2 \delta_{kl} \tag{3.34}$$

$$\begin{bmatrix} \boldsymbol{u}_{d,k}^T & \boldsymbol{\lambda}_k^T \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{d,l} \\ \boldsymbol{\lambda}_l \end{bmatrix} = \delta_{kl} \tag{3.35}$$

which, taking account of the compatibility constraints, yield

$$\sum_{s=1}^{N_s} {\boldsymbol{u}_k^{(s)}}^T \boldsymbol{K}^{(s)} \boldsymbol{u}_l^{(s)} = \omega_k^2 \delta_{kl} \tag{3.36}$$

$$\sum_{s=1}^{N_s} {\boldsymbol{u}_k^{(s)}}^T \boldsymbol{M}^{(s)} \boldsymbol{u}_l^{(s)} = \delta_{kl} \tag{3.37}$$

The orthonormality in the latter form is totally equivalent to the orthonormality relations of the classical assembled system.

The free vibration problem (3.32) is generally solved by Krylov subspace iteration methods such as the subspace iteration or the Lanczos method [91]. In those algorithms, the structural

[2]The same type of eigenproblem arises in stability analysis. The mass matrix is then replaced by a geometric stiffness matrix.

eigensolutions of lowest eigenvalues — which are of practical interest — are computed by successive inverse iterations which basically involve the solution of

$$\begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d^{n+1} \\ \boldsymbol{\lambda}^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d^{n} \\ \boldsymbol{\lambda}^{n} \end{bmatrix} \tag{3.38}$$

the subscript n standing for the iteration number. An inverse iteration such as (3.38) is equivalent to a static problem. At every iteration one can write the dual interface problem

$$\begin{bmatrix} \boldsymbol{F}_I & \boldsymbol{G}_I \\ \boldsymbol{G}_I^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{n+1} \\ \boldsymbol{\alpha}^{n+1} \end{bmatrix} = \begin{bmatrix} \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{K}^{(s)^+} \left(\boldsymbol{M}^{(s)} \boldsymbol{u}^{(s),n} \right) \\ \left[\boldsymbol{R}^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{u}^{(s),n} \right] \end{bmatrix} \tag{3.39}$$

The singularity of $\boldsymbol{K}_d$ can be eliminated in the inverse iteration by applying a shift to the eigenspectrum: defining the shifted eigenvalues $(\omega^2 - \mu^2)$ in (3.32), the local stiffnesses in the inverse iteration are then replaced by $\boldsymbol{K}_d - \mu^2 \boldsymbol{M}_d$ which is regular if μ does not coincide with an eigenvalue. The shifting technique is often used in practice for computing the eigenspectrum in the vicinity of a given eigenfrequency μ, but applying this method for removing the local singularities is not recommended and this for two reasons. First the shift must be well separate from an eigenvalue and should be small enough compared to the first eigenvalues in order not to modify the frequency range captured by the inverse iteration. Hence the shifting factor μ can be properly chosen only if one already knows a good estimate of the eigenfrequencies [91]. Secondly, as will be explained in part II, when an iterative solution scheme is applied for solving the static system (3.39) in the inverse iteration, the local floating modes provide an efficient coarse grid that accelerates convergence.

Calling $\boldsymbol{Z}^{(s)}(\omega) = \boldsymbol{K}^{(s)} - \omega^2 \boldsymbol{M}^{(s)}$ the dynamic impedance matrix of the subdomain $\Omega^{(s)}$, the eigenvalue problem can be put in the equivalent form

$$\begin{bmatrix} \boldsymbol{Z}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d \\ \boldsymbol{\lambda} \end{bmatrix} = \boldsymbol{0} \tag{3.40}$$

where $\boldsymbol{Z}_d$ is the block diagonal impedance matrix. The associated dual interface eigenproblem is

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{Z}^{(s)^{-1}}(\omega) \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} = \boldsymbol{B} \boldsymbol{Z}_d(\omega)^{-1} \boldsymbol{B}^T \boldsymbol{\lambda} = \boldsymbol{0} \tag{3.41}$$

This form is of little practical use since it yields a non-linear eigenproblem in terms of ω.

For harmonic forced response analysis though, $\omega = \omega_{ex}$ is the frequency of the harmonic excitation and calling $\boldsymbol{p}^{(s)}$ the amplitude of the harmonic excitation, the amplitudes of the forced response are given by

$$\begin{bmatrix} \boldsymbol{Z}_d(\omega_{ex}) & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_d \\ \boldsymbol{0} \end{bmatrix} \tag{3.42}$$

and the interface problem writes

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{Z}^{(s)^{-1}}(\omega_{ex}) \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{Z}^{(s)^{-1}} \boldsymbol{p}^{(s)} \tag{3.43}$$

Forced harmonic response analysis thus consists in solving a static-like problem. If ω_{ex} is not zero and if it does not coincide with an eigenfrequency of an isolated subdomain, the local impedance operators are not singular since the stiffness matrices are regularized by the inertia contributions.

3.4 Transient response analysis

The two-field hybrid method can also be applied for transient response analysis, namely to find the solution of (3.23) as a function of time for a given applied load $\boldsymbol{f}^{(s)}(t)$. Two methods can be applied for computing the time response. First, for linear systems, the dynamic response can be approximated by a mode superposition method, that is by expressing the solution as a series of modal components. This technique requires the computation of a set of structural eigenmodes. Applying the two-field hybrid method leads to new procedures for the computation of the eigensolutions, and for computing the acceleration corrections [91]. The required procedures have been described in the two previous sections and the normal equations associated with the mode superposition are no different from the normal equations obtained from the assembled system.

When a time-stepping algorithm is used for computing the transient response, the numerical time integration is performed by an explicit or an implicit algorithm [91]. The two-field hybrid form of the time dependent problem (3.23) can be put in the following matrix form:

$$\begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_d \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_d \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_d \\ \boldsymbol{0} \end{bmatrix} \tag{3.44}$$

The compatibility constraints in (3.44) are associated with the displacements as it is the most natural choice. However, from a mathematical viewpoint, it is equivalent to impose the compatibility constraints onto the velocities or onto the accelerations as explained in section 3.3. From a numerical viewpoint though, there is an essential difference as analyzed in [75]. Indeed, as explained in section 3.3, $2n_\lambda$ eigensolutions of infinite eigenvalue are associated with the free vibration equation corresponding to (3.44). These infinite eigensolutions can be physically interpreted as the dynamic modes associated with the Lagrange multipliers for which according to (3.44) a stiffness term is defined but no inertia. It is well known that for conditionally stable time-integration algorithms, the maximum time step depends on the highest frequency of the model and therefore when infinite eigensolutions are present no stable time-step can be chosen. Even for unconditionally stable schemes such as the Newmark method [91], the presence of infinite spurious eigensolutions destroys the stability of the time-integration since the appearance of the infinite eigenmodes is triggered in practice by round-off errors [75].

There are several ways to overcome the problem of numerical instability when solving the constrained differential algebraic equations (3.44) [75]. It is common use to introduce numerical damping in the integration schemes in order to dampen out the spurious infinite eigensolutions [28,75]. Unfortunately, although numerical damping can be designed in order to affect only slightly the dominant spectrum of the response [32], it always adversely affects the accuracy of the solution.

Numerical stability can also be ensured when by-passing the dynamics of the Lagrange multipliers by reshaping the algorithms and using momentum variables [75]. A third technique is to associate the constraints to the acceleration terms. The compatibility constraints being linear, computing their second derivative is straightforward and we can write

$$\begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_d \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_d \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_d \\ \boldsymbol{0} \end{bmatrix} \tag{3.45}$$

The dynamics of the Lagrange multipliers is characterized by zero stiffness and non-zero inertia: the spurious modes are now associated with zero eigensolutions so that no instability appears.

3.4.1 Explicit time integration

Explicit integration schemes do not require the solution of a system of equation because the solution at a time step t_n explicitly depends on the previously computed time steps. Only matrix multiplications and vector updates are implied. For instance, considering for the sake of simplicity that the system is undamped, the central difference algorithm [91] leads to the following explicit

scheme:

$$\boldsymbol{u}_d^n = \boldsymbol{u}_d^{n-1} + h^n \dot{\boldsymbol{u}}_d^{n-1/2} \tag{3.46}$$

$$\begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_d^n \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_d^n \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d^n \\ \boldsymbol{\lambda} \end{bmatrix} \tag{3.47}$$

$$\dot{\boldsymbol{u}}_d^{n+1/2} = \dot{\boldsymbol{u}}_d^{n-1/2} + h^{n+1/2} \ddot{\boldsymbol{u}}_d^n \tag{3.48}$$

where the superscript n refers to the time step t^n, h^n is the corresponding time step and $h^{n+1/2} = \frac{h^n + h^{n+1}}{2}$. When the mass is assumed to be diagonal, the central difference scheme in its assembled form leads to an explicit algorithm, but we observe that its dual form (3.47) is not explicit since the associated interface problem writes

$$\left(\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} {\boldsymbol{M}^{(s)}}^{-1} {\boldsymbol{B}^{(s)}}^T \right) \boldsymbol{\lambda} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} {\boldsymbol{M}^{(s)}}^{-1} \left(\boldsymbol{f}^{(s)_n} - \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s)_n} \right) \tag{3.49}$$

First we note that the local mass matrices are generally definite positive and if they are diagonal ${\boldsymbol{M}^{(s)}}^{-1}$ are directly accessible which simplifies the expression of the dual problem. We must stress that the dual problem is not diagonal in general so that the integration method in its dual form is no longer trivially explicit. Nevertheless, we will explain in part II, Chap. 16 how to solve (3.49) in a simple and cheap way.

For explicit integration algorithms to be numerically stable, the time step should not be higher then a value depending on the highest eigenfrequency included in the system. For instance, when using the central difference scheme, the time step must fulfil the requirement imposed by the so-called Courant's condition stipulating that $h^n \leq 2/\omega_{cr}$, ω_{cr} being an upper bound of the model eigenfrequency spectrum. Taking as cut-off frequency the highest frequency of any isolated element, this condition means that no wave should travel more than the distance between two elements during one time-step [91].

Courant's condition generally requires to use very small time steps which are really needed only when one is interested in the computation of wave propagation such as in impact analysis. For most engineering applications however the time step prescribed by the Courant's condition is much to small compared to the time scale of the phenomenon one wants to analyze and thus the number of time steps required for a given time interval is very high. As a consequence, even though one iteration may be inexpensive, computing the response might be unnecessarily demanding. Therefore, it is generally preferable to use an implicit integration scheme which is unconditionally stable, meaning that the time step can be chosen as large as wanted. It is generally determined according to the required integration accuracy.

3.4.2 Implicit time integration

The two-field hybrid method applied to implicit time integration will be presented for the commonly used one-step Newmark formula. However any implicit scheme can be handled in the same manner. Assuming a constant time step, the Newmark formula writes [91]

$$\dot{\boldsymbol{u}}^{(s)_{n+1}} = \dot{\boldsymbol{u}}^{(s)_n} + \frac{h}{2} \left(\ddot{\boldsymbol{u}}^{(s)_n} + \ddot{\boldsymbol{u}}^{(s)_{n+1}} \right) \tag{3.50a}$$

$$\boldsymbol{u}^{(s)_{n+1}} = \boldsymbol{u}^{(s)_n} + h\dot{\boldsymbol{u}}^{(s)_n} + \frac{h^2}{4} \left(\ddot{\boldsymbol{u}}^{(s)_n} + \ddot{\boldsymbol{u}}^{(s)_{n+1}} \right) \tag{3.50b}$$

The time discretization of the dynamic equations (3.44) results in

$$\begin{bmatrix} \boldsymbol{D}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_d^{n+1} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_d^{n+1} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{C}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_d^{*n+1} \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d^{*n+1} \\ \boldsymbol{\lambda} \end{bmatrix} \tag{3.51}$$

where $\boldsymbol{D}_d$ is the block diagonal matrix of the local iteration matrices defined by

$$\boldsymbol{D}^{(s)} = \boldsymbol{M}^{(s)} + \frac{h}{2}\boldsymbol{C}^{(s)} + \frac{h^2}{4}\boldsymbol{K}^{(s)} \tag{3.52}$$

$\boldsymbol{u}_d^{*^{n+1}}$ and $\dot{\boldsymbol{u}}_d^{*^{n+1}}$ are respectively the displacement and velocity predictors

$$\boldsymbol{u}_d^{*^{n+1}} = \boldsymbol{u}_d^n + h\dot{\boldsymbol{u}}_d^n + \frac{h^2}{4}\ddot{\boldsymbol{u}}_d^n \tag{3.53a}$$

$$\dot{\boldsymbol{u}}_d^{*^{n+1}} = \dot{\boldsymbol{u}}_d^n + \frac{h}{2}\ddot{\boldsymbol{u}}_d^n \tag{3.53b}$$

The dual interface problem associated to (3.51) writes:

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{D}^{(s)^{-1}}\boldsymbol{B}^{(s)^T}\,\boldsymbol{\lambda} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{D}^{(s)^{-1}}\left(\boldsymbol{f}^{(s)_{n+1}} - \boldsymbol{C}^{(s)}\dot{\boldsymbol{u}}^{*(s)_{n+1}} - \boldsymbol{K}^{(s)}\boldsymbol{u}^{*(s)_{n+1}}\right) \tag{3.54}$$

Hence, the solution of the implicit time-stepping equations is entirely similar to the solution of a static problem except that the local stiffness operators are replaced by the iteration matrices $\boldsymbol{D}^{(s)}$. Since the local mass matrices are generally regular, the local dynamic stiffnesses $\boldsymbol{D}^{(s)}$ are also regular and no auxiliary null space problem is associated to (3.54).

3.4.3 Initial acceleration

When starting a time-stepping algorithm, one must first evaluate the initial accelerations knowing the initial displacements and velocities. For instance, considering the Newmark formulae (3.50a-3.50b), the computation of the displacement and velocity predictors for $n = 1$ requires the knowledge of $\ddot{\boldsymbol{u}}^{(s)^0}$. The initial acceleration is determined by solving the problem

$$\begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{u}}_d^0 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_d^0 \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{C}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{u}}_d^0 \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_d^0 \\ \boldsymbol{\lambda} \end{bmatrix} \tag{3.55}$$

This system is similar to the equations of explicit time integration and therefore, if an explicit scheme is used, the initial acceleration can be found easily. However when an implicit scheme is used and when the mass matrices are not diagonal, solving for $\ddot{\boldsymbol{u}}^{(s)_0}$ requires to perform an elimination on $\boldsymbol{M}_d$ which might be costly.

If the cost of computing the initial acceleration is too high, it is common use to set $\ddot{\boldsymbol{u}}^{(s)^0} = \boldsymbol{0}$. In that case, an error is introduced at the time t^0: indeed, making the assumption of zero initial acceleration is equivalent to introducing a parasitic initial force equal and opposite to the out of equilibrium damping, elastic and applied forces:

$$\boldsymbol{f}_r^{(s)^0} = \boldsymbol{C}^{(s)}\dot{\boldsymbol{u}}^{(s)^0} + \boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)^0} - \boldsymbol{f}^{(s)^0} \tag{3.56}$$

The energy introduced in each sub-domain by this parasitic force when the Newmark scheme is used is equal to $\frac{1}{2}(\boldsymbol{u}^{(s)^1} - \boldsymbol{u}^{(s)^0})\boldsymbol{f}_r^{(s)^0}$ [91]. Hence the error due to the zero initial acceleration assumption decreases when the time step is reduced, but when no numerical or structural damping is present, the error remains during the entire transient response. In [110] the error introduced when truncating the starting conditions is studied and it is shown that in that case integration schemes that are intrinsically of the order of accuracy $O(h^2)$ become of the order of $O(h)$. Hence it is important to compute the initial acceleration if possible.

Some approximation methods could be used in order to compute a good estimate of $\ddot{\boldsymbol{u}}^{(s)^0}$. For instance one could use a diagonal approximation of the local mass matrices. Another approximation would be to compute the initial accelerations only subdomain by subdomain, neglecting the interface coupling. However in that case, the interface compatibility of the initial acceleration is not satisfied which would lead to a drift of the compatibility constraints over the integration interval.

Chapter 4

Modal reduction of substructures

4.1 Craig-Bampton modal reduction: primal and dual formulation

For industrial applications, computing the eigenmodes and eigenfrequencies of a model by solving the successive static problems (3.38) in the inverse iterations is very costly, especially if modal analysis is performed on a very fine mesh designed for static analysis purposes. In the latter case, the degree of refinement of the meshing is unnecessarily high for computing the low frequency eigenspectrum. Reduction methods have therefore been developed. The displacement fields in the substructures are approximated by a set of Ritz functions or in the context of finite element models, the space of the d.o.f. is approximated by a lower dimension subspace of displacement modes. The most popular method consists in defining a spectral approximation of the subdomain operator by a combination of local eigenmodes.

In the Craig-Bampton method, the eigenmodes of the substructures clamped on the interface are considered together with the interface static modes. Assuming that the interface boundary d.o.f. are numbered first, the local displacements are approximated by [91]

$$\boldsymbol{u}^{(s)} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} & \boldsymbol{\Phi}_i^{(s)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_b^{(s)} \\ \boldsymbol{\eta}^{(s)} \end{bmatrix} \tag{4.1}$$

The subscripts i and b respectively refer to the internal and boundary d.o.f. and $\boldsymbol{\eta}^{(s)}$ are the amplitudes of the first orthonormal eigenmodes of the clamped substructure that are stored in $\boldsymbol{\Phi}_i^{(s)}$ of dimension $n_i^{(s)} \times n_\phi^{(s)}$:

$$\boldsymbol{K}_{ii}^{(s)} \boldsymbol{\Phi}_i^{(s)} = \boldsymbol{M}_{ii}^{(s)} \boldsymbol{\Phi}_i^{(s)} \boldsymbol{\Omega}_i^{(s)^2} \tag{4.2}$$

$$\boldsymbol{\Phi}_i^{(s)^T} \boldsymbol{M}_{ii}^{(s)} \boldsymbol{\Phi}_i^{(s)} = \boldsymbol{I} \tag{4.3}$$

$$\boldsymbol{\Phi}_i^{(s)^T} \boldsymbol{K}_{ii}^{(s)} \boldsymbol{\Phi}_i^{(s)} = \boldsymbol{\Omega}_i^{(s)^2} \tag{4.4}$$

and $\boldsymbol{\Omega}_i^{(s)}$ is the diagonal matrix of the circular frequencies associated with the clamped subdomains. Obviously, as the number of eigenmodes $n_\phi^{(s)}$ that approximate the internal dynamic behavior is increased, the eigenspectrum of the reduced model converges towards the eigenspectrum of the initial complete system. Theoretical convergence results can be found in [22].

Substituting in (3.32), the reduced hybrid eigenproblem writes [76]

$$\begin{bmatrix} \boldsymbol{S}_{bb}^{(1)} & \mathbf{0} & . & . & \mathbf{0} & \boldsymbol{b}^{(1)^T} \\ \mathbf{0} & \boldsymbol{\Omega}^{(1)^2} & . & . & . & \mathbf{0} \\ . & . & . & . & . & . \\ . & . & . & \boldsymbol{S}_{bb}^{(N_s)} & \mathbf{0} & \boldsymbol{b}^{(N_s)^T} \\ \mathbf{0} & . & . & \mathbf{0} & \boldsymbol{\Omega}^{(N_s)^2} & \mathbf{0} \\ \boldsymbol{b}^{(1)} & \mathbf{0} & . & \boldsymbol{b}^{(N_s)} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_b^{(1)} \\ \boldsymbol{\eta}^{(1)} \\ . \\ \boldsymbol{u}_b^{(N_s)} \\ \boldsymbol{\eta}^{(N_s)} \\ \boldsymbol{\lambda} \end{bmatrix} = \omega^2 \begin{bmatrix} \overline{\boldsymbol{M}}_{bb}^{(1)} & \overline{\boldsymbol{M}}_{b\phi}^{(1)} & . & . & \mathbf{0} & \mathbf{0} \\ \overline{\boldsymbol{M}}_{\phi b}^{(1)} & \boldsymbol{I} & . & . & . & \mathbf{0} \\ . & . & . & . & . & . \\ . & . & . & \overline{\boldsymbol{M}}_{bb}^{(N_s)} & \overline{\boldsymbol{M}}_{b\phi}^{(N_s)} & \mathbf{0} \\ \mathbf{0} & . & . & \overline{\boldsymbol{M}}_{\phi b}^{(N_s)} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & . & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_b^{(1)} \\ \boldsymbol{\eta}^{(1)} \\ . \\ \boldsymbol{u}_b^{(N_s)} \\ \boldsymbol{\eta}^{(N_s)} \\ \boldsymbol{\lambda} \end{bmatrix} \tag{4.5}$$

where $\boldsymbol{S}_{bb}^{(s)}$ is the stiffness matrix statically condensed onto the interface, i.e. the local Schur complement, $\overline{\boldsymbol{M}}_{bb}^{(s)}$ is the statically condensed mass matrix and $\overline{\boldsymbol{M}}_{b\phi}^{(s)}$ the mass coupling matrix given by

$$\begin{aligned} \overline{\boldsymbol{M}}_{bb}^{(s)} &= \boldsymbol{M}_{bb}^{(s)} - \boldsymbol{M}_{bi}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} - \boldsymbol{K}_{bi}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{M}_{ib}^{(s)} \\ &\quad + \boldsymbol{K}_{bi}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{M}_{ii}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} \end{aligned} \tag{4.6}$$

$$\overline{\boldsymbol{M}}_{b\phi}^{(s)^T} = \overline{\boldsymbol{M}}_{\phi b}^{(s)} = \boldsymbol{\Phi}^{(s)^T} \left(\boldsymbol{M}_{ib}^{(s)} - \boldsymbol{M}_{ii}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} \right) \tag{4.7}$$

The reduced eigenproblem (4.5) is the two-field hybrid formulation of the well-known Craig-Bampton modal reduction method where the interface Schur complements are split among the substructures and where the coupling is expressed by the interface compatibility constraints.

An inverse iteration for the reduced eigenproblem (4.5) involves the solution of a set of trivial local diagonal systems

$$\boldsymbol{\Omega}^{(s)^2}{}_{n+1} \boldsymbol{\eta}^{(s)_{n+1}} = \overline{\boldsymbol{M}}_{\phi b}^{(s)} \boldsymbol{u}_b^{(s)_n} + \boldsymbol{\eta}^{(s)_n} \qquad s = 1, \ldots N_s \tag{4.8}$$

and the dual interface problem writes

$$\boldsymbol{u}_b^{(s)_{n+1}} = \boldsymbol{S}_{bb}^{(s)^+} \left(\overline{\boldsymbol{M}}_{bb}^{(s)} \boldsymbol{u}_b^{(s)_n} + \overline{\boldsymbol{M}}_{b\phi}^{(s)} \boldsymbol{\eta}^{(s)_n} - \boldsymbol{b}^{(s)^T} \boldsymbol{\lambda}^{n+1} \right) - \boldsymbol{R}^{(s)} \boldsymbol{\alpha}^{(s)_{n+1}} \tag{4.9}$$

$$\begin{bmatrix} \boldsymbol{F}_I & \boldsymbol{G}_I \\ \boldsymbol{G}_I^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{n+1} \\ \boldsymbol{\alpha}^{n+1} \end{bmatrix} = \begin{bmatrix} \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \left(\overline{\boldsymbol{M}}_{bb}^{(s)} \boldsymbol{u}_b^{(s)_n} + \overline{\boldsymbol{M}}_{b\phi}^{(s)} \boldsymbol{\eta}^{(s)_n} \right) \\ \begin{bmatrix} \vdots \\ \boldsymbol{R}_b^{(s)^T} \left(\overline{\boldsymbol{M}}_{bb}^{(s)} \boldsymbol{u}_b^{(s)_n} + \overline{\boldsymbol{M}}_{b\phi}^{(s)} \boldsymbol{\eta}^{(s)_n} \right) \\ \vdots \end{bmatrix} \end{bmatrix} \tag{4.10}$$

where $\boldsymbol{F}_I$ is the interface static flexibility (3.22)

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \boldsymbol{b}^{(s)^T} \tag{3.22}$$

computed by mapping the inverse of the Schur complements onto the dual space and $\boldsymbol{G}_I$ is as defined in (3.18b). Note that, if enough interface boundary d.o.f. exist, the null space $\boldsymbol{R}_b^{(s)^T}$ of the

Schur complement $\boldsymbol{S}_{bb}^{(s)}$ is equal to the boundary restriction of the null space $\boldsymbol{R}^{(s)}$ of $\boldsymbol{K}^{(s)}$ since the former results from a static condensation of the latter.

The primal assembled form of the reduced eigenproblem (4.5) is found by setting $\boldsymbol{u}_b^{(s)} = \boldsymbol{b}^{(s)^*} \tilde{\boldsymbol{u}}_b$, where $\boldsymbol{b}^{(s)^*}$ is the restriction to $\Gamma_I^{(s)}$ of the null space of $\boldsymbol{B}$ (see (2.78)):

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{S} & \boldsymbol{0} & . & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}^{(1)^2} & . & . \\ . & . & . & \boldsymbol{0} \\ \boldsymbol{0} & . & \boldsymbol{0} & \boldsymbol{\Omega}^{(N_s)^2} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}}_b \\ \boldsymbol{\eta}^{(1)} \\ . \\ \boldsymbol{\eta}^{(N_s)} \end{bmatrix} \\ = \quad & \omega^2 \begin{bmatrix} \overline{\boldsymbol{M}} & \overline{\overline{\boldsymbol{M}}}_{b\phi}^{(1)} & . & \overline{\overline{\boldsymbol{M}}}_{b\phi}^{(N_s)} \\ \overline{\overline{\boldsymbol{M}}}_{\phi b}^{(1)} & \boldsymbol{I} & . & \boldsymbol{0} \\ . & . & . & . \\ \overline{\overline{\boldsymbol{M}}}_{\phi b}^{(N_s)} & \boldsymbol{0} & . & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}}_b \\ \boldsymbol{\eta}^{(1)} \\ . \\ \boldsymbol{\eta}^{(N_s)} \end{bmatrix} \end{aligned} \tag{4.11}$$

where $\boldsymbol{S}$ is the assembled Schur complement (2.83), $\overline{\boldsymbol{M}}$ is the assembled and statically condensed stiffness matrix

$$\overline{\boldsymbol{M}} = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)*^T} \overline{\boldsymbol{M}}_{bb}^{(s)} \boldsymbol{b}^{(s)*} \tag{4.12}$$

and $\overline{\overline{\boldsymbol{M}}}_{b\phi}^{(s)} = \boldsymbol{b}^{*^T} \overline{\boldsymbol{M}}_{b\phi}^{(s)}$. For conforming substructures, (4.11) is equal to the original Craig-Bampton expression.

The modal reduction scheme described above has several advantages. First, the internal d.o.f. of the substructures are not explicitly computed during the inverse iteration since they are approximated by the reduced set of eigenmodes to which, thanks to the orthonormality of the eigensolutions, the trivial problem (4.8) is associated. Another practical use of this modal reduction method appears when the storage space can not handle the entire subdomain matrices: modal reduction allows then to work with the small Schur complement matrices and with a limited number of subdomain eigensolutions. Finally, the modal reduction schemes can be used as modal synthesis methods: the static coupling modes and the local eigenmodes can be taken from modal and structural testing.

The dual interface problem (4.10) arising from the modal reduction is similar to the original dual interface system (3.39) for the inverse iteration: the dimension of the interface problem is unchanged. But the right-hand side in the reduced problem is now function of the reduced set of local clamped eigenmodes. However, as a reduction technique for the primal eigenproblem (3.29) or for its dual formulation (3.32), the Craig and Bampton method described here is not inexpensive: it requires the computation of local eigensolutions and the computation of the Schur complements and of the condensed mass matrices. We also observe that the primal and dual interface problems associated with the modal reduction algorithm are of the same size as the original one. Indeed, no reduction of the boundary d.o.f. has been introduced since we use the exact static coupling modes relative to the subdomain static solution when Dirichlet boundary conditions are assumed on the interface. Observing that the interface problem must be solved for every eigenvalue iteration, the advantage of having only a small size interface problem becomes clear. We next discuss different techniques for reducing the number of interface variables by defining spectral approximations of the primal and dual interface operators.

4.2 Spectral expansion of the Schur complement

Reducing the space of the boundary d.o.f. by defining an approximation subspace for the coupling modes is a very attractive idea. In [22] it was proposed to use as coupling modes the eigenmodes

associated with the Steklov-Poincaré operator of the interface. For discrete systems, this corresponds to computing the spectral approximation of the primal interface operator consisting in the assembled Schur complement $\boldsymbol{S}$ defined by (2.82).

Let us assume that the set of independent displacements $\tilde{\boldsymbol{u}}$ can be split into local internal d.o.f. $\tilde{\boldsymbol{u}}_i^{(s)}$ and global interface d.o.f. such that

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_b \\ \tilde{\boldsymbol{u}}_i^{(1)} \\ \vdots \\ \tilde{\boldsymbol{u}}_i^{(N_s)} \end{bmatrix} \tag{4.13}$$

We define the static modes

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{I} \\ \begin{bmatrix} -\boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)*} \end{bmatrix} \\ \vdots \end{bmatrix} \tilde{\boldsymbol{u}}_b \tag{4.14}$$

where the boundary restrictions $\boldsymbol{b}^{(s)*}$ of the null space of $\boldsymbol{B}$ are Boolean for conforming interfaces. Substituting (4.14) in the primal eigenproblem (3.29), we find the statically condensed eigenproblem

$$\boldsymbol{S}\tilde{\boldsymbol{u}}_b = \omega^2 \overline{\boldsymbol{M}} \tilde{\boldsymbol{u}}_b \tag{4.15}$$

where $\boldsymbol{S}$ is the assembled global Schur complement defined in (2.82) and where $\overline{\boldsymbol{M}}$ is the assembled statically condensed mass matrix (4.12). The eigenmodes of the statically condensed system are stored in the columns of $\boldsymbol{\Phi}_b$ and the associated eigenvalues are the diagonal of $\boldsymbol{\Omega}_b^2$ so that the associated orthonormality relations write

$$\boldsymbol{\Phi}_b^T \boldsymbol{S} \boldsymbol{\Phi}_b = \boldsymbol{\Omega}_b^2 \tag{4.16}$$

$$\boldsymbol{\Phi}_b^T \overline{\boldsymbol{M}} \boldsymbol{\Phi}_b = \boldsymbol{I} \tag{4.17}$$

Using the n_{ϕ_b} first eigenmodes to approximate the boundary d.o.f., $\boldsymbol{\Phi}_b$ is of dimension $\tilde{n}_b \times n_{\phi_b}$ and the modal reduction basis is defined as

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{\Phi}_b & \boldsymbol{0} \\ \begin{bmatrix} -\boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)*} \boldsymbol{\Phi}_b \end{bmatrix} & \begin{bmatrix} \boldsymbol{\Phi}_i^{(1)} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{\Phi}_i^{(N_s)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_b \\ \begin{bmatrix} \boldsymbol{\eta}^{(s)} \\ \vdots \end{bmatrix} \end{bmatrix} \tag{4.18}$$

The reduced eigenproblem in the primal form then writes

$$\left(\begin{bmatrix} \boldsymbol{\Omega}_b^2 & \boldsymbol{0} \\ \boldsymbol{0} & \left[\boldsymbol{\Omega}_i^{(s)^2}\right] \end{bmatrix} - \omega^2 \begin{bmatrix} \boldsymbol{I} & \left[\boldsymbol{\Phi}_b^T \overline{\overline{\boldsymbol{M}}}_{b\phi}^{(s)}\right] \\ \left[\overline{\overline{\boldsymbol{M}}}_{\phi b}^{(s)} \boldsymbol{\Phi}_b\right] & [\boldsymbol{I}] \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\eta}_b \\ \left[\boldsymbol{\eta}^{(s)}\right] \end{bmatrix} = \boldsymbol{0} \tag{4.19}$$

Solving the inverse iterations associated with (4.19) includes only diagonal matrices and is thus trivial. Theoretical analysis of this reduction method [23, 24] shows that the convergence of the eigensolution of the reduced system towards the eigensolutions of the non-reduced eigenproblem depends on the number of local and interface modes included in the representation. The essential theoretical result in [23, 24] is that the convergence of this reduction method does not depend on the smoothness of the solution, that is for instance on the heterogeneities present in the system. This property is obtained at the cost of computing the eigenmodes of the local and interface problems.

A way of reducing the cost of computing the interface eigenmodes is to approximate the exact interface Schur complement by assuming that the d.o.f. pertaining to the interior of the subdomains are fixed. This corresponds to using the eigenmodes of the eigenproblem lumped on the interface

$$\boldsymbol{K}_{bb}\boldsymbol{\Phi}_{b,l} = \boldsymbol{M}_{bb}\boldsymbol{\Phi}_{b,l}\boldsymbol{\Omega}^2_{b,l} \tag{4.20}$$

where the subscript l indicates the eigensolutions derive from a lumped eigenproblem. Using these lumped modes for reducing the interface problem is not effective in practice because a subset including the first modes $\boldsymbol{\Phi}_{b,l}$ is a poor approximation of the interface displacements of the eigenmodes for the entire system.

An intermediate approach then consists in building a good modal interface basis by considering the interface eigenproblem for the statically condensed stiffness matrix and the lumped mass:

$$\boldsymbol{S}\boldsymbol{\Phi}_{b,m} = \boldsymbol{M}_{bb}\boldsymbol{\Phi}_{b,m}\boldsymbol{\Omega}^2_{b,m} \tag{4.21}$$

where the subscript m indicates that the eigenproblem assumes a lumped mass matrix. The reduced eigenproblem is then similar to (4.19) except that the diagonal block $\boldsymbol{\Phi}^T_{b,m}\overline{\boldsymbol{M}}_{bb}\boldsymbol{\Phi}_{b,m}$ is no longer equal to the identity. This mass lumping variant of the interface reduction method is cost effective since it does not require to work with the statically condensed mass matrix for computing the eigenmodes $\boldsymbol{\Phi}_{b,m}$ in (4.21). The theoretical convergence results relative to the Steklov-Poincaré reduction have not been generalized to this mass lumping variant and in particular we cannot guarantee that the convergence of this variant does not depend on the solution smoothness.

Let us illustrate the solution resulting from the interface reduction method based on the spectral approximation of the Schur complement with two simple cases. The structure we consider is a plane stress square domain, clamped on one side and subdivided into 4 square subdomains (Fig. 4.1). Each subdomain is modeled by 10×10 four noded square finite elements. In the first case (a) we assume that the structure is homogeneous, whereas in the second case (b) 7 elements in subdomain 1 have a Young's modulus 10 times higher and 2 elements in subdomain 4 have a Young's modulus 10 times lower. The eigenmodes of the second problem are therefore less smooth then those of the homogeneous problem.

The aim of these numerical tests being to assess the convergence of the interface reduction, we will assume that no reduction is introduced for the internal d.o.f. The boundary d.o.f. are approximated by a truncated basis containing the n_{ϕ_b} lower modes $\boldsymbol{\Phi}_b$ or $\boldsymbol{\Phi}_{b,m}$, depending if the statically condensed or the lumped mass matrix is considered for the interface eigenproblem. The eigenfrequencies of the structures are computed for n_{ϕ_b} varying from 5 to 40, the number of assembled boundary d.o.f. being 82. The relative error for the eigenfrequencies number 1, 5, 10 and 20 is plotted in Fig. 4.2 as a function of n_{ϕ_b}. The relative error is computed by

$$\Delta\omega_i = \frac{\omega_{i,reduced} - \omega_{i,non-reduced}}{\omega_{i,non-reduced}} \tag{4.22}$$

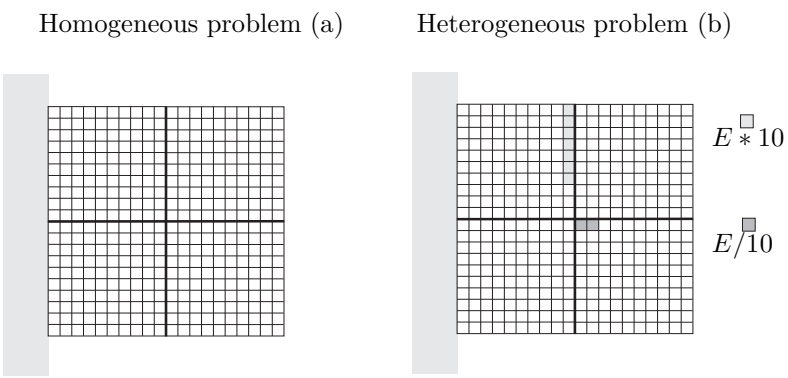


Figure 4.1: Homogeneous and heterogeneous plane stress problem

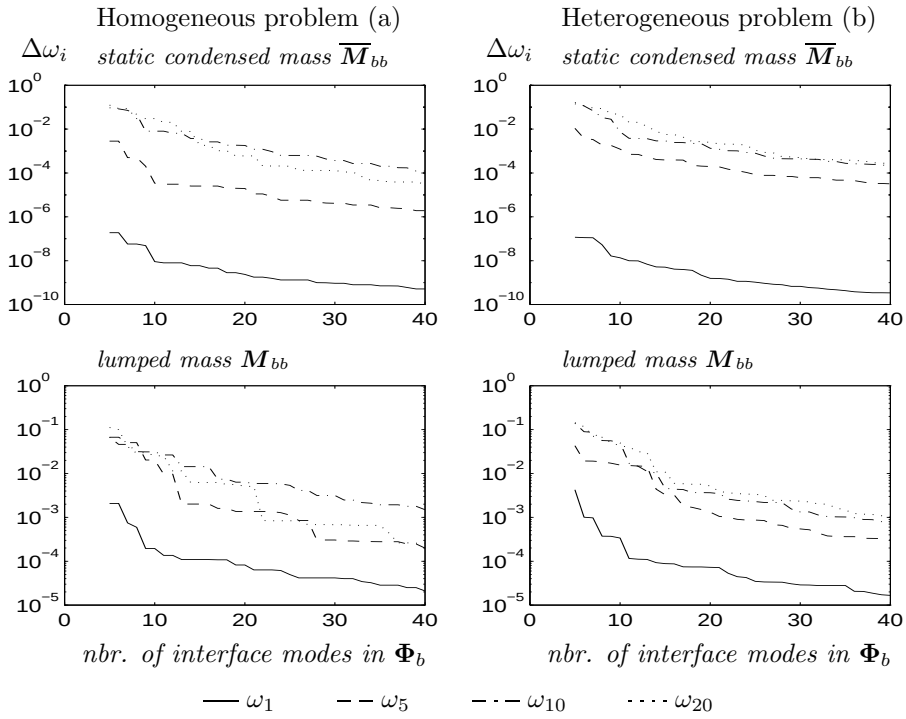


Figure 4.2: Convergence of the Steklov-Poincaré reduction

The plotted results show the very good approximation of the eigensolutions when the interface modes associated with the statically condensed mass matrix are considered. For the lumped mass variant, the error is higher for a same n_{ϕ_b} but the approximate solution remains good even for a low number of interface modes. The convergence of the eigensolutions when n_{ϕ_b} increases is rapid: the relative error on the first 20 eigenfrequencies is lower than 1 percent for n_{ϕ_b} equal to 13 when the condensed mass is considered and for n_{ϕ_b} equal to 17 when the mass is lumped. Note that the effect of mass lumping is significant only for the first eigenfrequencies. Comparing the results of the homogeneous and of the heterogeneous cases, we see that, as predicted by the theory [24], the convergence is unaffected by the smoothness of the solution.

Let us observe that reducing the interface d.o.f. leads to overestimating the eigenfrequencies, which is a consequence of the kinematical admissibility of the approximation, meaning that the reduction scheme introduces additional constraints on the displacement field.

The interface reduction method presented in this section is efficient but costly since it requires to compute the lower spectrum of the system statically condensed on the interface. Moreover, being based on a reduction of the interface displacement field and thus on a compatible field, this technique can not be transposed as such to a dual formulation of the interface problem. In what follows we will present an important simplification of the exact Steklov-Poincaré reduction which leads in a natural way to a reduction method for the dual formulation.

4.3 An approximation of the Schur complement spectrum

Let us consider once more the statically condensed eigenproblem (4.15) and let us define the interface static force modes associated to the modes $\boldsymbol{\Phi}_b$

$$\boldsymbol{Y}_b = \boldsymbol{S}\boldsymbol{\Phi}_b \tag{4.23}$$

The interface eigenproblem (4.15) can then be put in the equivalent form

$$\overline{\boldsymbol{M}}\boldsymbol{S}^{-1}\boldsymbol{Y}_b = \boldsymbol{Y}_b\boldsymbol{\Omega}_b^{2^{-1}} \tag{4.24}$$

In the latter expression we have assumed that $\boldsymbol{S}$ is regular, namely that no global rigid body modes exist for the assembled system. In the case global floating modes exist, one should consider a general More-Penrose inverse of $\boldsymbol{S}$ and add to the computed modes the global rigid body modes.

The eigenmodes associated with the lower spectrum of (4.15) correspond to the eigenmodes of the high spectrum of (4.24). The advantage of considering the eigenproblem associated to the interface forces is that the eigensolutions of interest for approximating the dominant spectrum of $\boldsymbol{S}$ can be computed by a non-inverse iteration scheme. The problem is then to obtain $\boldsymbol{S}^{-1}$ in order to form the force eigenproblem (4.24). Obviously if we compute exactly the inverse of $\boldsymbol{S}$ we haven't gained anything compared to the original Steklov-Poincaré reduction. The idea presented in [39] is then to approximate the exact inverse of the Schur complement by the Neumann preconditioner as defined for the primal iterative interface problem [125, 175].

4.3.1 Approximating the inverse of the assembled Schur Complement

The approximation of $\boldsymbol{S}^{-1}$ is set up as follows. Let us search for an approximate solution of the static problem

$$\boldsymbol{S}\boldsymbol{x} = \boldsymbol{y} \tag{4.25}$$

where the assembled Schur complement results from the assembling (2.82)

$$\boldsymbol{S} = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)*^T} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)*} \tag{2.82}$$

Let us recall that $\boldsymbol{b}^{(s)*}$ is the restriction to the interface d.o.f. of subdomain $\Omega^{(s)}$ of $\boldsymbol{B}^*$, the null space of $\boldsymbol{B}$. For matching grids, $\boldsymbol{B}^*$ is equivalent to the Boolean assembling matrix $\boldsymbol{L}$.

Let us also consider a weighting matrix $\boldsymbol{D}^{(s)}$ that defines a partition of unity such that

$$\sum_{s=1}^{N_s} \boldsymbol{b}^{(s)*^T} \boldsymbol{D}^{(s)} \boldsymbol{b}^{(s)*} = \boldsymbol{I} \tag{4.26}$$

For matching grids, the simplest choice for $\boldsymbol{D}^{(s)}$ is to take the diagonal matrix of the inverse of the d.o.f. multiplicity. An approximation of $\boldsymbol{x}$ is then found in each subdomain by distributing the interface forces $\boldsymbol{y}$:

$$\boldsymbol{y}^{(s)} = \boldsymbol{D}^{(s)} \boldsymbol{b}^{(s)*} \boldsymbol{y} = \boldsymbol{Q}^{(s)} \boldsymbol{y} \tag{4.27}$$

where $\boldsymbol{Q}^{(s)} = \boldsymbol{D}^{(s)} \boldsymbol{b}^{(s)*}$. Then $\boldsymbol{x}$ is approximated in subdomain $\Omega^{(s)}$ by the solution of the local Neumann problems

$$\boldsymbol{S}_{bb}^{(s)} \boldsymbol{x}^{(s)} = \boldsymbol{y}^{(s)} \qquad s = 1, \ldots N_s \tag{4.28}$$

If $\boldsymbol{S}_{bb}^{(s)}$ is singular (floating subdomain), a solution exist only if $\boldsymbol{y}^{(s)}$ is self-equilibrated, that is if $\boldsymbol{y}^{(s)}$ is orthogonal to the null space of $\boldsymbol{S}_{bb}^{(s)}$. Hence we will replace $\boldsymbol{y}$ and $\boldsymbol{y}^{(s)}$ respectively by $\overline{\boldsymbol{y}}$ and $\overline{\boldsymbol{y}}^{(s)}$ such that

$$\boldsymbol{R}_b^{(s)^T} \overline{\boldsymbol{y}}^{(s)} = \boldsymbol{0} \qquad \forall s, \boldsymbol{S}_{bb}^{(s)} \text{ singular} \tag{4.29}$$

$$\overline{\boldsymbol{y}}^{(s)} = \boldsymbol{Q}^{(s)} \overline{\boldsymbol{y}} \tag{4.30}$$

Defining

$$\boldsymbol{G} = \left[\begin{array}{ccc} \cdots & \boldsymbol{Q}^{(s)^T} \boldsymbol{R}_b^{(s)} & \cdots \end{array} \right] \tag{4.31}$$

the self-equilibrium condition writes[1]

$$\boldsymbol{G}^T\overline{\boldsymbol{y}} = \boldsymbol{0} \tag{4.32}$$

and we thus define

$$\overline{\boldsymbol{y}} = \left(\boldsymbol{I} - \boldsymbol{S}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{S}\boldsymbol{G})^{-1}\boldsymbol{G}^T\right)\boldsymbol{y} = \boldsymbol{P}\boldsymbol{y} \tag{4.33}$$

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{S}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{S}\boldsymbol{G})^{-1}\boldsymbol{G}^T \tag{4.34}$$

$\boldsymbol{P}$ is a projection matrix that projects $\boldsymbol{y}$ in a subspace of the boundary forces whose distribution onto each subdomains is self-equilibrated with respect to the subdomains floating modes. This technique is known as *subdomain balancing*, as was first proposed in [125] in the context of primal iterative decomposition methods. Note that $(\boldsymbol{G}^T\boldsymbol{S}\boldsymbol{G})$ is regular if the global structure is non-floating.

The local solutions are given by

$$\boldsymbol{x}^{(s)} = \boldsymbol{S}_{bb}^{(s)^+}\boldsymbol{Q}^{(s)}\overline{\boldsymbol{y}} + \boldsymbol{R}_b^{(s)}\boldsymbol{\beta}^{(s)} \tag{4.35}$$

where, as before, $\boldsymbol{S}_{bb}^{(s)^+}$ is a generalized inverse if $\boldsymbol{S}_{bb}^{(s)}$ is singular or it is equal to $\boldsymbol{S}_{bb}^{(s)^{-1}}$ if $\boldsymbol{S}_{bb}^{(s)}$ is regular. $\boldsymbol{\beta}^{(s)}$ are the amplitudes of the local rigid body modes. Obviously the solutions $\boldsymbol{x}^{(s)}$ are in general not identical on each side of an interface and an approximate unique solution is computed by weighting the subdomain solutions

$$\boldsymbol{x} = \sum_{s=1}^{N_s} \boldsymbol{Q}^{(s)^T}\left(\boldsymbol{S}_{bb}^{(s)^+}\boldsymbol{Q}^{(s)}\overline{\boldsymbol{y}} + \boldsymbol{R}_b^{(s)}\boldsymbol{\beta}^{(s)}\right) \tag{4.36}$$

and the rigid body amplitudes are defined by imposing that the interface forces corresponding to $\boldsymbol{x}$ are self-equilibrated, i.e.

$$\boldsymbol{G}^T\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0} \tag{4.37}$$

We get

$$\boldsymbol{x} = \boldsymbol{P}^T\left(\sum_{s=1}^{N_s} \boldsymbol{Q}^{(s)^T}\boldsymbol{S}_{bb}^{(s)^+}\boldsymbol{Q}^{(s)}\right)\boldsymbol{P}\boldsymbol{y} \tag{4.38}$$

The approximation of the inverse of $\boldsymbol{S}$ is finally given by

$$\tilde{\boldsymbol{S}}^{-1} = \boldsymbol{P}^T\left(\sum_{s=1}^{N_s} \boldsymbol{Q}^{(s)^T}\boldsymbol{S}_{bb}^{(s)^+}\boldsymbol{Q}^{(s)}\right)\boldsymbol{P} \tag{4.39}$$

4.3.2 An approximate modal basis

We store in $\boldsymbol{Y}_b$ the n_{Y_b} highest modes of the eigenproblem

$$\overline{\boldsymbol{M}}\tilde{\boldsymbol{S}}^{-1}\boldsymbol{Y}_b = \boldsymbol{Y}_b\boldsymbol{\Omega}_b^2 \tag{4.40}$$

According to the definition of the interface forces $\boldsymbol{Y}_b$, the eigenmodes are then approximated by

$$\tilde{\boldsymbol{\Phi}}_b = \tilde{\boldsymbol{S}}^{-1}\boldsymbol{Y}_b \tag{4.41}$$

and an approximation to the subspace span($\boldsymbol{\Phi}_b$) is found by taking the subspace span($\tilde{\boldsymbol{S}}^{-1}\boldsymbol{Y}_b$).

From (4.39) and (4.34) we note that $\tilde{\boldsymbol{S}}^{-1}\boldsymbol{Y}_b$ is orthogonal to $\boldsymbol{S}\boldsymbol{G}$, namely to the interface forces associated with the local $\boldsymbol{R}^{(s)}$. The eigensolutions of the structure not being a priori orthogonal

[1] Note the similarity between the set up of the approximate inverse of the primal Schur operator and the construction of the dual interface problem. For instance, $\boldsymbol{G}$ plays the same role as $\boldsymbol{G}_I$.

to $\boldsymbol{SG}$, one explicitly adds to the reduction space the subspace span($\boldsymbol{G}$) so that the reduction basis finally writes

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \tilde{\boldsymbol{\Phi}}_b & \boldsymbol{0} \\ \begin{bmatrix} -\boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{*(s)} \tilde{\boldsymbol{\Phi}}_b \end{bmatrix} & \begin{bmatrix} \boldsymbol{\Phi}_i^{(1)} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{\Phi}_i^{(N_s)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_b \\ \begin{bmatrix} \boldsymbol{\eta}^{(s)} \end{bmatrix} \end{bmatrix} \tag{4.42}$$

$$\tilde{\boldsymbol{\Phi}}_b = \begin{bmatrix} \tilde{\boldsymbol{S}}^{-1} \boldsymbol{Y}_b & \boldsymbol{G} \end{bmatrix} \tag{4.43}$$

The reduced eigenproblem of the assembled system is then similar to the reduced eigenproblem (4.19) for the exact Steklov-Poincaré, except that the blocks associated with the interface mode amplitudes are no longer diagonal:

$$\left(\begin{bmatrix} \tilde{\boldsymbol{\Phi}}_b^T \boldsymbol{S} \tilde{\boldsymbol{\Phi}}_b & \boldsymbol{0} \\ \boldsymbol{0} & \left[\boldsymbol{\Omega}_i^{(s)^2}\right] \end{bmatrix} - \omega^2 \begin{bmatrix} \tilde{\boldsymbol{\Phi}}_b^T \overline{\boldsymbol{M}} \tilde{\boldsymbol{\Phi}}_b & \left[\tilde{\boldsymbol{\Phi}}_b^T \overline{\overline{\boldsymbol{M}}}_{b\phi}^{(s)}\right] \\ \left[\overline{\overline{\boldsymbol{M}}}_{\phi b}^{(s)} \tilde{\boldsymbol{\Phi}}_b\right] & [\boldsymbol{I}] \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\eta}_b \\ \left[\boldsymbol{\eta}^{(s)}\right] \end{bmatrix} = \boldsymbol{0} \tag{4.44}$$

As for the exact Steklov-Poincaré reduction, the statically condensed mass matrix in the interface force eigenproblem (4.40) can be replaced by the cheap lumped matrix $\boldsymbol{M}_{bb}$.

4.3.3 Application examples

The approximate representation of the Steklov-Poincaré spectrum has been applied for computing the eigenfrequencies of the homogeneous and heterogeneous problems depicted in Fig. 4.1. The relative error on the eigenfrequencies are presented in Fig. 4.3 as a function of the number of modes in $\boldsymbol{Y}_b$.

The result curves show that the approximate Steklov-Poincaré reduction based on the Neumann preconditioner is not as effective as the exact method (Fig. 4.2), but still yields good results even for a small number of interface modes. The convergence rate towards the non-reduced eigenfrequencies is similar to the convergence rate for the exact method.

We also note that when the condensed matrix $\overline{\boldsymbol{M}}$ is replaced by its lumped counterpart, the relative error on the eigenfrequencies is only slightly increased. Comparing the homogeneous and the heterogeneous cases, we see that the accuracy of the eigenfrequencies is only slightly affected by the irregularity of the eigenmodes and the convergence is slower for the heterogeneous case.

4.4 Modal reduction of the dual interface operator

The approximate Steklov-Poincaré reduction presented above is efficient and requires to solve local Neumann problems whereas the exact Steklov-Poincaré method requires to solve successive Neumann problems for the assembled interface. Let us observe that the solution of local Neumann problems is the basis of the dual interface formulation so that the approximate Steklov-Poincaré reduction leads in a natural way to a reduction technique for the dual interface operator $\boldsymbol{F}_I$. Indeed, there is a close similarity between the Neumann preconditioner $\tilde{\boldsymbol{S}}^{-1}$ and the dual interface operator $\boldsymbol{F}_I$: $\tilde{\boldsymbol{S}}^{-1}$ is obtained by assembling in a primal sense the weighted subdomain flexibilities and $\boldsymbol{F}_I$ is built by assembling them in a dual sense.

Considering the dual eigenproblem with a Craig-Bampton modal reduction (4.5) and its corresponding inverse iteration (4.10), we see that in order to reduce the interface problem, the Lagrange multipliers field must be reduced. A modal basis for the Lagrange multipliers can be

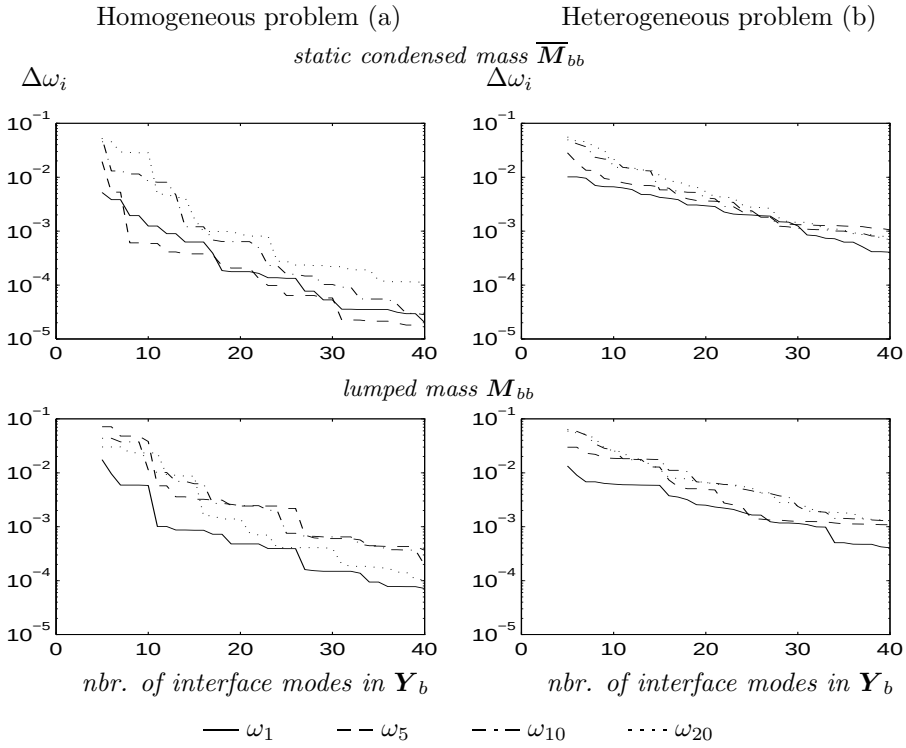


Figure 4.3: Convergence of the approximate Steklov-Poincaré reduction

found by writing the eigenproblem related to $\boldsymbol{F}_I$,

$$\boldsymbol{P}\boldsymbol{F}_I\boldsymbol{P}\tilde{\boldsymbol{\Lambda}} = \tilde{\boldsymbol{\Lambda}}\boldsymbol{\mu} \tag{4.45}$$

$$\boldsymbol{\Lambda} = \boldsymbol{P}\tilde{\boldsymbol{\Lambda}} \tag{4.46}$$

where $\boldsymbol{P}$ is the projector

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{G}_I(\boldsymbol{G}_I^T\boldsymbol{G}_I)^{-1}\boldsymbol{G}_I^T \tag{4.47}$$

such that $\boldsymbol{G}_I^T\boldsymbol{P} = \boldsymbol{0}$, hence $\boldsymbol{G}_I^T\boldsymbol{\Lambda} = \boldsymbol{0}$. Assuming that the eigensolutions are normalized, we have the orthogonality relations

$$\boldsymbol{\Lambda}^T\boldsymbol{F}_I\boldsymbol{\Lambda} = \boldsymbol{\mu} \tag{4.48}$$

Having stressed the similarities between $\tilde{\boldsymbol{S}}^{-1}$ and $\boldsymbol{F}_I$, and recalling that the high spectrum of $\tilde{\boldsymbol{S}}^{-1}$ was used for building a reduced spectral basis in Section 4.3, we consider the n_Λ *last eigensolutions* of (4.45) to build a modal reduction basis for the Lagrange multiplier field: we state

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\zeta} \tag{4.49}$$

The term $\boldsymbol{\lambda}_0$ represents the component of the solution $\boldsymbol{\lambda}$ that lies in the space span($\boldsymbol{G}_I$) and which is orthogonal to the subspace span($\boldsymbol{\Lambda}$) (note the similarity with $\boldsymbol{G}$ in the approximation space $\tilde{\boldsymbol{\Phi}}_b$ (4.43)). Substituting in the inverse iterations of the general eigenproblem (3.38), we find

$$\begin{bmatrix} \boldsymbol{K}_d & \boldsymbol{B}^T\boldsymbol{\Lambda} \\ \boldsymbol{\Lambda}^T\boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d^{n+1} \\ \boldsymbol{\zeta}^{n+1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}^T\boldsymbol{\lambda}_0^{n+1} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_d^{n} \\ \boldsymbol{\zeta}^{n} \end{bmatrix} \tag{4.50}$$

In this system, we see that the compatibility constraints are now projected onto the subspace span($\boldsymbol{\Lambda}$) which means that they have been weakened according to the reduced Lagrange multiplier representation. The reduced dual interface problem is deduced from (4.50):

$$\boldsymbol{\Lambda}^T\boldsymbol{B}\boldsymbol{K}_d^+\boldsymbol{B}^T\boldsymbol{\Lambda}\,\boldsymbol{\zeta}^{n+1} = \boldsymbol{\Lambda}^T\boldsymbol{B}\boldsymbol{K}_d^+\left(\boldsymbol{M}_d\boldsymbol{u}_d^n - \boldsymbol{B}^T\boldsymbol{\lambda}_0^{n+1}\right) \tag{4.51a}$$

$$\boldsymbol{G}_I^T\boldsymbol{\lambda}_0^{n+1} = \left[\boldsymbol{R}^{(s)^T}\boldsymbol{M}^{(s)}\boldsymbol{u}^{(s)_n}\right] \tag{4.51b}$$

The second relation pertaining to the self-equilibrium condition is satisfied if we define a particular solution

$$\boldsymbol{\lambda}_0^{n+1} = \boldsymbol{G}_I(\boldsymbol{G}_I^T\boldsymbol{G}_I)^{-1}\left[\boldsymbol{R}^{(s)^T}\boldsymbol{M}^{(s)}\boldsymbol{u}^{(s)_n}\right] \tag{4.52}$$

and according to the orthonormality relations (4.48), the reduced interface problem (4.51a) is the diagonal system

$$\boldsymbol{\mu}\zeta^{n+1} = \boldsymbol{\Lambda}^T\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{K}^{(s)^+}\left(\boldsymbol{M}^{(s)}\boldsymbol{u}^{(s)_n} - \boldsymbol{B}^{(s^T)}\boldsymbol{\lambda}_0^{n+1}\right) \tag{4.53}$$

Eqs. (4.52-4.53) define the reduced interface problem for the inverse iteration.

If the reduced modal basis for the Lagrange multipliers is now introduced in the dual Craig-Bampton expression (4.5), the reduced dual interface iteration including modal reduction of the subdomains displacements writes:

$$\boldsymbol{\lambda}_0^{n+1} = \boldsymbol{G}_I(\boldsymbol{G}_I^T\boldsymbol{G}_I)^{-1}\left[\boldsymbol{R}_b^{(s)^T}\left(\overline{\boldsymbol{M}}_{bb}^{(s)}\boldsymbol{u}_b^{(s)_n} + \overline{\boldsymbol{M}}_{b\phi}^{(s)}\boldsymbol{\eta}^{(s)_n}\right)\right] \tag{4.54a}$$

$$\boldsymbol{\mu}\zeta^{n+1} = \boldsymbol{\Lambda}^T\sum_{s=1}^{N_s}\boldsymbol{b}^{(s)}\boldsymbol{S}_{bb}^{(s)^+}\left(\overline{\boldsymbol{M}}_{bb}^{(s)}\boldsymbol{u}_b^{(s)_n} + \overline{\boldsymbol{M}}_{b\phi}^{(s)}\boldsymbol{\eta}^{(s)_n} - \boldsymbol{b}^{(s)^T}\lambda_0^{n+1}\right) \tag{4.54b}$$

$$\boldsymbol{u}_b^{(s)_{n+1}} = \boldsymbol{S}_{bb}^{(s)^+}\left(\overline{\boldsymbol{M}}_{bb}^{(s)}\boldsymbol{u}_b^{(s)_n} + \overline{\boldsymbol{M}}_{b\phi}^{(s)}\boldsymbol{\eta}^{(s)_n} - \boldsymbol{b}^{(s)^T}\boldsymbol{\lambda}_0^{n+1} - \boldsymbol{b}^{(s)^T}\boldsymbol{\Lambda}\boldsymbol{\zeta}^{n+1}\right) \tag{4.54c}$$

$$\boldsymbol{\Omega}^{(s)^2_{n+1}}\boldsymbol{\eta}^{(s)_{n+1}} = \overline{\boldsymbol{M}}_{\phi b}^{(s)}\boldsymbol{u}_b^{(s)_n} + \boldsymbol{\eta}^{(s)_n} \tag{4.54d}$$

The reduction of the dual interface problem by a modal basis of the Lagrange multipliers weakens the interface compatibility since the constraints $\boldsymbol{B}\boldsymbol{u}_d = \boldsymbol{0}$ are restricted to a subset of constraints $\boldsymbol{\Lambda}^T\boldsymbol{B}\boldsymbol{u}_d = \boldsymbol{0}$. Therefore this reduction technique is assimilated to a statically admissible approach and when the interior displacement d.o.f. are not reduced, this method leads to underestimating the eigenspectrum of the non-reduced system. When a modal reduction is also applied to the subdomains d.o.f. as in (4.54), the corresponding reduction scheme can be understood at the subdomain level as a hybrid approximation technique of the non-reduced original system.

This dual reduction scheme has been applied to the test problems of Fig. 4.1. The relative error of the eigenfrequencies ω is plotted in Fig. 4.4 as a function of the number n_Λ of eigenmodes included in the reduction basis $\boldsymbol{\Lambda}$. We observe that the error on the eigenfrequencies is of the same order of magnitude as the error on the results obtained when approximating the Steklov-Poincaré operator by the Neumann preconditioner. When the interface eigenmodes are irregular as in the

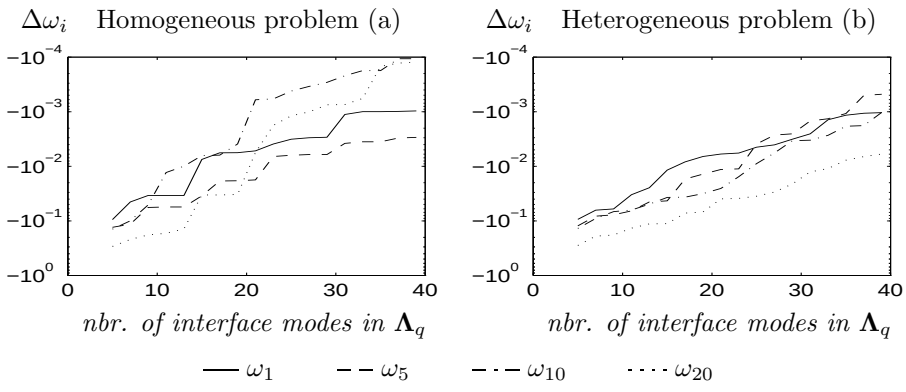


Figure 4.4: Convergence of the dual operator reduction method

heterogeneous case, the reduction of the dual operator leads to a slightly less accurate solution than when the interface is smooth.

The essential lessons to be drawn from this method are the followings:

1. A reduction scheme for the dual interface operator can be formulated in a way similar to the approximation method for spectral reduction of the Schur complement (Section 4.3). The interesting point here is that the mechanically meaningful eigensolutions of the dual interface operator correspond to the high eigenvalues and can be computed with a non-inverse iteration scheme, and hence the dual operator $\boldsymbol{F}_I$ need not be explicitly constructed and factored.

2. This new interface spectral reduction method opens new possibilities for modal reduction. For instance, it is noteworthy to observe that in the spectrally reduced dual eigenproblem (4.52-4.53), the subdomain displacements $\boldsymbol{u}^{(s)}$ can be approximated not only by fixed boundary modes such as in the Craig-Bampton method expressed in Eqs. (4.54), but one could as well use free boundary local eigenmodes since interface boundary nodes are no longer required to define a unique set of interface displacements.

3. This reduction method demonstrates that the eigenspectrum of $\boldsymbol{F}_I$ which participates to the solution of a dual problem corresponds to the high end eigenspectrum. Therefore if we apply an iterative solution scheme to a dual interface problem (see Part II), and considering that iterative schemes capture the high end of the eigenspectrum first, the convergence towards the exact solution will be excellent when working with a dual formulation.

Chapter 5

Reducing the interface problem and weak interface compatibility

As presented in chapter 2, the compatibility constraints on the interface between substructures care for exact compatibility when the grids are matching or for a weaker compatibility when the interface is non-conforming so as to represent the global structure with the maximum of accuracy. In some particular cases, the interface compatibility constraints can be weakened in order for instance to reduce the number of interface variables so that when the substructures are spread among different processors, the solution of the coupling interface problem requires a minimum of communications.

In Section 4.4 we showed how the Lagrange multiplier field can be efficiently represented by a modal basis and how the corresponding eigenproblem is reduced. The modal reduction technique of Section 4.4 requires the computation of the eigenmodes of $\boldsymbol{F}_I$ and is thus expensive. In this chapter we will discuss the reduction of the interface problem by approximating the Lagrange multiplier field with simple distribution functions.

The idea of using the two-field hybrid formulation for building a reduced interface problem was proposed and analyzed in [47]: the Lagrange multiplier field enforcing the interface compatibility is approximated by force distribution functions defined by a limited number of parameters (for instance polynomials). This technique has been applied to practical engineering problems in statics and dynamics in [60, 76, 77].

Approximating the interface connecting loads by a small set of distribution functions takes advantage of knowing a priori some characteristics of the interface solution such as its smoothness. It is clear that for several academic and practical structural problems, one can assume that the interface solution is smooth in terms of displacements and stresses.

The effect of approximating the interface coupling is generally negligible for simple displacement modes such as the first eigensolutions of the global system. What's more, according to Saint Venant's principle, any alteration of the exact solution in the vicinity of the interface should not influence the solution away from the interface provided the modeling of the interface coupling allows a good representation of the global behavior of the system. For instance, the reduction method should satisfy at least the patch test so that constant stress fields are correctly transmitted across the interfaces. As explained in 2.3.2, when a Lagrange multiplier field is defined on a continuous interface, equilibrium is satisfied along all interface edges and constant stress is transmitted correctly if the Lagrange multipliers functions contain at least the constant field representation.

5.1 The reduced dual interface problem

Let us assume that along the interface segments Γ_I^{sr} continuous Lagrange multiplier fields are defined by $\mathbf{\Lambda}^{sr,p}$ where the superscript p indicates that the distribution functions are defined with a small number n_{λ^p} of variables:

$$\lambda^{sr}(\eta) \simeq \mathbf{\Lambda}^{sr,p}(\eta)\boldsymbol{\lambda}^{sr,p} \tag{5.1}$$

Introducing these distribution functions in the Lagrangian function (2.40), we find

$$\begin{aligned}\mathcal{L}(\boldsymbol{v}^{(s)},\boldsymbol{\mu}^p) &= \sum_{s=1}^{N_s}\mathcal{J}(v^{(s)})_{\Omega^{(s)}} \\ &\quad + \sum_{s=1}^{N_s}\sum_{\substack{\Gamma_I^{sr}\neq\emptyset \\ s<r}} \boldsymbol{\mu}^{sr,p^T}\int_{\Gamma_I^{sr}} \mathbf{\Lambda}^{sr,p^T}\left(\boldsymbol{N}^{(s)}\boldsymbol{v}^{(s)} - \boldsymbol{N}^{(r)}\boldsymbol{v}^{(r)}\right)d\Gamma_I^{sr}\end{aligned}$$

where $\boldsymbol{N}^{(s)}$ is as before the matrix of displacement shape functions for subdomain $\Omega^{(s)}$. The discretized equations then write

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + \boldsymbol{B}^{(s),p^T}\boldsymbol{\lambda}^p = \boldsymbol{f}^{(s)} \qquad s=1,\dots N_s \tag{5.2a}$$

$$\sum_{s=1}^{N_s}\boldsymbol{B}^{(s),p}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{5.2b}$$

where the constraint matrices are computed by

$$\boldsymbol{B}^{(s),p} = \left[\begin{array}{ccccccc}\boldsymbol{0} & \dots & \boldsymbol{b}^{(sr),p} & \boldsymbol{0} & \dots & \boldsymbol{b}^{(st),p} & \dots\end{array}\right] \tag{5.3}$$

$$\boldsymbol{b}^{(sr),p} = \pm\int_{\Gamma_I^{sr}}\mathbf{\Lambda}^{sr,p^T}\boldsymbol{N}^{(s)}d\eta \tag{5.4}$$

The reduced dual interface problem writes

$$\left[\begin{array}{cc}\boldsymbol{F}_I^p & \boldsymbol{G}_I^p \\ \boldsymbol{G}_I^{pT} & \boldsymbol{0}\end{array}\right]\left[\begin{array}{c}\boldsymbol{\lambda}^p \\ \boldsymbol{\alpha}^p\end{array}\right] = \left[\begin{array}{c}\displaystyle\sum_{s=1}^{N_s}\boldsymbol{B}^{(s),p}\boldsymbol{K}^{(s)^+}\boldsymbol{f}^{(s)} \\ \boldsymbol{e}\end{array}\right] \tag{5.5}$$

where the reduced dual interface operator is defined by

$$\boldsymbol{F}_I^p = \sum_{s=1}^{N_s}\boldsymbol{B}^{(s),p}\boldsymbol{K}^{(s)^+}\boldsymbol{B}^{(s),p^T} \tag{5.6}$$

and

$$\boldsymbol{G}_I^p = \left[\begin{array}{ccc}\dots & \boldsymbol{B}^{(s),p}\boldsymbol{R}^{(s)} & \dots\end{array}\right] \tag{5.7}$$

If the number of Lagrange multipliers $\boldsymbol{\lambda}^p$ is small, the dual interface operator can be explicitly constructed by computing once for all the results $\boldsymbol{K}^{(s)^+}\boldsymbol{B}^{(s),p^T}$ and the factorization of $\boldsymbol{F}_I^p$ involves only a small computational cost. Doing this might be advantageous if the structural problem must be solved for several right-hand sides.

In order to analyze the accuracy of the reduced two-field hybrid method (5.5), we will take as reference solution the solution of the two-field hybrid system when the interface compatibility is enforced in the strongest sense. For matching grids it means that the reference solution $(\boldsymbol{u}^{(s)},\boldsymbol{\lambda}^{sr})$ is found when a discrete Lagrange multiplier is associated with every interface d.o.f., which is equivalent to solving the fully assembled system. As exposed in section 2.3.2, the interfaces are then exactly compatible.

For non-conforming interfaces, we define the reference solution as the solution of a hybrid system when the interface compatibility is enforced with a maximum of Lagrange multiplier variables. In our applications, we do not verify the required Brezzi-Babuška condition but we will satisfy the necessary condition that the interface problem has a solution: the number of interface dual unknowns is limited by the number of d.o.f. defined on the coarsest side of the interface. The reference solution for non-matching grids will be called *the n_{λ^q} solution*, n_{λ^q} standing for the number of Lagrange variables that define the strongest compatibility constraint.

We will call $(\boldsymbol{u}^{(s)}, \boldsymbol{\lambda}^p)$ the saddle point verifying

$$\mathcal{L}(\boldsymbol{u}^{(s)}, \boldsymbol{\lambda}^p) = \inf_{\boldsymbol{v}^{(s)}} \sup_{\boldsymbol{\mu}^p} \mathcal{L}\left(\boldsymbol{v}^{(s)}, \mu^{rs} = \boldsymbol{\Lambda}^{rs,p}\boldsymbol{\mu}^{rs,p}\right) \tag{5.8}$$

and the reference solution satisfies

$$\mathcal{L}(\boldsymbol{u}^{(s),ref}, \boldsymbol{\lambda}) = \inf_{\boldsymbol{v}^{(s)}} \sup_{\boldsymbol{\mu}^q} \mathcal{L}\left(\boldsymbol{v}^{(s)}, \boldsymbol{\mu}^{rs} = \boldsymbol{\Lambda}^{rs,q}\boldsymbol{\mu}^{rs,q}\right) = \mathcal{L}(\boldsymbol{u}^{(s)}, \boldsymbol{\lambda}^q) \tag{5.9}$$

where, for conforming interfaces, $n_{\lambda^q} = n^{(s)}_{\Gamma_I^{sr}} = n^{(r)}_{\Gamma_I^{sr}}$ and for non-conforming interfaces $n_{\lambda^q} = \min(n^{(s)}_{\Gamma_I^{sr}}, n^{(r)}_{\Gamma_I^{sr}})$.

The reduced compatibility constraints (5.2b) express a weak compatibility since the continuous compatibility constraints in the Lagrangian functions are projected onto the subspace $\boldsymbol{\Lambda}^{sr,p}$. Therefore, according to the saddle point principle (3.1), we can write

$$\mathcal{L}(\boldsymbol{u}^{(s)}, \boldsymbol{\lambda}^p) \leq \mathcal{L}(\boldsymbol{u}^{(s),ref}, \boldsymbol{\lambda}) \tag{5.10}$$

From a mechanical viewpoint, reducing the number of compatibility constraints increases the number of effective d.o.f. of the system and thus decreases the global stiffness. To prove this point, let us consider the domain-wise energy norm of the displacement error, namely of the difference between the displacement $\boldsymbol{u}^{(s)}$ computed for a weak interface compatibility and the strongly compatible displacements $\boldsymbol{u}^{(s),ref}$ obtained for the reference interface compatibility. Considering that $\boldsymbol{u}^{(s)}$ satisfy the Eqs. (5.2a-5.2b), we find

$$\begin{aligned}
\sum_{s=1}^{N_s} \left\|\boldsymbol{u}^{(s)} - \boldsymbol{u}^{(s),ref}\right\|^2_{\boldsymbol{K}^{(s)}} &= \sum_{s=1}^{N_s} \left(\boldsymbol{u}^{(s)} - \boldsymbol{u}^{(s),ref}\right)^T \boldsymbol{K}^{(s)} \left(\boldsymbol{u}^{(s)} - \boldsymbol{u}^{(s),ref}\right) \\
&= \sum_{s=1}^{N_s} \Big(\boldsymbol{u}^{(s)^T} \left(\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s),p}\boldsymbol{\lambda}^p\right) \\
&\qquad + \boldsymbol{u}^{(s),ref^T} \left(\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s),q}\boldsymbol{\lambda}\right) \\
&\qquad - 2\boldsymbol{u}^{(s),ref^T} \left(\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s),p}\boldsymbol{\lambda}^p\right)\Big)
\end{aligned} \tag{5.11}$$

Taking account of the weak and the reference compatible conditions verified by $\boldsymbol{u}^{(s)}$ and $\boldsymbol{u}^{(s),ref}$, and noting that $\boldsymbol{u}^{(s),ref}$ also satisfies the weaker compatibility, we finally obtain

$$\left\|\boldsymbol{u}^{(s)} - \boldsymbol{u}^{(s),ref}\right\|^2_{\boldsymbol{K}^{(s)}} = \sum_{s=1}^{N_s} \boldsymbol{u}^{(s)^T} \boldsymbol{f}^{(s)} - \sum_{s=1}^{N_s} \boldsymbol{u}^{(s),ref^T} \boldsymbol{f}^{(s)} \geq 0 \tag{5.12}$$

This clarifies what we mean when we state that using a weak interface compatibility softens the structural model: the work produced by the external loads is overestimated when the interface displacement is weakly compatible.

5.2 Choosing the interface stress distribution functions

If one knows a priori that the interface stress field is smooth, it is natural to take as stress distribution simple functions defined over every interface segment Γ_I^{sr}. The stress distribution functions used in practice can be one of the followings.

Polynomials

The simplest choice for $\boldsymbol{\Lambda}^{sr,p}$ is to use polynomials having as support the interface Γ_I^{sr}. In that case, a sufficient condition can be found for satisfying the Brezzi-Babuška condition ensuring the convergence of the hybrid approximation method [47, 98]: when refining the discretization of the problem, the polynomial degree p of the Lagrange multiplier field on Γ_I^{sr} should not grow quicker than the square root of the number of interface displacement d.o.f. For practical problems, this condition is too restrictive and as already detailed before we will satisfy only the necessary solvability condition by assuming that the maximum degree p that one can choose is smaller than the number of d.o.f. on the coarsest side.

A well-known bottle-neck when increasing the polynomial degree in a polynomial representation is that the conditioning number of the approximation equations rapidly deteriorates, so that in practice the polynomial degree is limited to a value of 7 or 8. Indeed, for higher polynomial degrees, the constraints matrices $\boldsymbol{b}^{sr,p}$ are very badly conditioned and no valuable numerical solution can be computed for the interface problem.

Tchebichev polynomials and Fourier series

To overcome the ill-conditioning of polynomials, one can use orthonormal base functions such as Tchebichev polynomials or Fourier series. Unfortunately, since the columns of $\boldsymbol{b}^{sr,p}$ are computed by integrating $\boldsymbol{\Lambda}^{sr,p^T}\boldsymbol{N}^{(s)}$, the columns of $\boldsymbol{b}^{sr,p}$ are not orthogonal and the condition number also deteriorates. Nevertheless the number of approximation functions one can use is higher when the distribution functions are orthogonal.

Piecewise polynomials

In order to increase the number of Lagrange multiplier parameters while using low order polynomials and thus ensure a good conditioning of the dual interface problem, one can define piecewise polynomials along Γ_I^{sr} and increase the discretization of λ by increasing the number of segments on which low-order polynomials are defined. However piecewise polynomials do not form a very good basis to approximate smooth functions since the approximation function is discontinuous. The Mortar method [1, 17, 112] is a particular case of piecewise polynomials where the Lagrange multiplier field is equal to the displacement shape functions on one side of the interface.

Smooth piecewise polynomials

In order to take advantage of the good conditioning of piecewise polynomials and to form smooth approximation bases, one can impose the continuity of the piecewise polynomials as well as the continuity of their derivatives. For instance in [77], piecewise cubic Bessel interpolations are used.

The Lagrange multiplier distribution functions described here above have proven very efficient when the stress distribution along the interface is smooth. In the case of coefficient jumps (Young modulus ...) along the interface or when external non-smooth loads are applied onto the interface, the interface solution is irregular and the number of interface dual variables might become high if a good accuracy is required.

One can use the eigensolutions of $\boldsymbol{F}_I$ as in the modal reduction method discussed in Section 4.4. As shown in that section, this technique is efficient but costly since it requires to compute

the eigensolutions of $\boldsymbol{F}_I$. Nevertheless, the physically meaningful eigenmodes of $\boldsymbol{F}_I$ are associated with the higher eigenvalues and can be computed by rather cheap non-inverse iterations.

Other interface load distributions can also be included in the Lagrange distribution functions to enhance the accuracy of the representation. For instance if a previous similar structural analysis has been carried out, the interface stress field of previous solutions can be taken as test functions. One could also inject into the functional basis any interface stress field available from measurements.

5.3 Orthonormalization of constraint matrices

As explained in the previous section, using polynomial functions as interface stress distribution causes the dual interface problem to be ill-conditioned. One of the remedies is then to define piecewise polynomials. An interesting feature of defining piecewise functions is that, as in a finite element meshing, the control points of the support segments can be placed so as to define small segments on portions where the interface stresses are very irregular. However a drawback of this technique is that computing the constraints matrices requires cumbersome piecewise integration. This seriously complicates the implementation of such methods.

Therefore, in our examples we will use another technique: simple polynomial or sinusoidal functions will be used over the interface Γ_I^{sr}, but the constraints matrices will be orthonormalized after construction. Noticing that the conditioning of the interface operator $\boldsymbol{F}_I^p = \boldsymbol{B}^p \boldsymbol{K}_d^+ \boldsymbol{B}^{p^T}$ directly depends on the conditioning of $\boldsymbol{B}^p \boldsymbol{B}^{p^T}$, one defines a change of variable for $\boldsymbol{\lambda}^p$ such that the columns of the constraint matrices are orthonormal.

Let us consider the constraint matrices $\boldsymbol{b}^{rs,p}$ and $\boldsymbol{b}^{sr,p}$ associated to the Lagrange multiplier $\lambda^{sr,p} = \boldsymbol{\Lambda}^{sr,p} \boldsymbol{\lambda}^p$ defined between the substructures $\Omega^{(s)}$ and $\Omega^{(r)}$. The corresponding compatibility constraints are

$$\boldsymbol{b}^{sr,p} \boldsymbol{u}_b^{(s)} + \boldsymbol{b}^{rs,p} \boldsymbol{u}_b^{(r)} = \boldsymbol{0} \tag{5.13}$$

We then consider the orthonormal expansion of $\boldsymbol{b}^{sr,p^T}$ for instance

$$\boldsymbol{b}^{sr,p^T} = \boldsymbol{Q}^{sr,p^T} \boldsymbol{R}^{sr,p} \tag{5.14}$$

where $\boldsymbol{R}^{sr,p}$ is a square matrix that is regular if $\boldsymbol{b}^{sr,p^T}$ is full column rank and where the columns of $\boldsymbol{Q}^{sr,p^T}$ are orthonormal:

$$\boldsymbol{Q}^{sr,p} \boldsymbol{Q}^{sr,p^T} = \boldsymbol{I} \tag{5.15}$$

Defining then the variable change

$$\overline{\boldsymbol{\lambda}}^{sr,p} = \boldsymbol{R}^{sr,p} \boldsymbol{\lambda}^{sr,p} \tag{5.16}$$

the compatibility constraints associated with $\overline{\boldsymbol{\lambda}}^{sr,p}$ write

$$\boldsymbol{Q}^{sr,p} \boldsymbol{u}_b^{(s)} + \boldsymbol{R}^{sr,p^{-1}} \boldsymbol{b}^{rs,p} \boldsymbol{u}_b^{(r)} = \boldsymbol{0} \tag{5.17}$$

For matching interfaces, $\boldsymbol{b}^{sr,p} = -\boldsymbol{b}^{rs,p}$ and the modified compatibility constraints write

$$\boldsymbol{Q}^{sr,p} \boldsymbol{u}_b^{(s)} - \boldsymbol{Q}^{sr,p} \boldsymbol{u}_b^{(r)} = \boldsymbol{0} \tag{5.18}$$

In practice, when for instance polynomial distribution functions are used, the orthogonalization process degenerates, namely when the columns of $\boldsymbol{b}^{sr,p}$ are nearly linearly dependent, it is numerically impossible to orthogonalize them all. This orthogonalization technique improves the numerical conditioning of $\boldsymbol{F}_I^p$ and enables us to use polynomial distributions of higher degree. In our examples we found that with orthogonalization, p can be taken as high as 12.

5.4 Averaging of interface displacements

5.4.1 Building a compatible approximate solution

The solution of a reduced two-field hybrid system exhibits a weak compatibility along the subdomains interface Γ_I and the solution of the assembled system is perturbed by the extra interface flexibility. Indeed, calling n_{λ^q} the number of constraints defined on Γ_I for the reference solution, the effective number of interface d.o.f. for the reference solution is $\sum_s n_b^{(s)} - n_{\lambda^q}$ whereas the number of effective interface d.o.f. for the reduced problem is $\sum_s n_b^{(s)} - n_{\lambda^p}$, $n_{\lambda^p} < n_{\lambda^q}$. In practice, one observes that the error in the reduced system solution is partly due to the approximation of the interface stress field, but for smooth problems the very origin of the solution error lies in the poor interface compatibility. From this observation, as proposed in [47, 77], an improved solution can be constructed based on the solution of the reduced hybrid system by defining an *a posteriori* compatible solution. For matching grids, a compatible solution can be achieved by applying a simple averaging scheme.

Let us consider the case of two subdomains for which a weakly compatible solution $\boldsymbol{u}^{(1)}$ and $\boldsymbol{u}^{(2)}$ has been computed such that

$$\boldsymbol{b}^{(1),p}\boldsymbol{u}_b^{(1)} + \boldsymbol{b}^{(2),p}\boldsymbol{u}_b^{(2)} = \boldsymbol{0} \tag{5.19}$$

and let us assume that the d.o.f. are numbered such that the reference compatibility writes

$$\boldsymbol{u}_b^{(1)} - \boldsymbol{u}_b^{(2)} = \boldsymbol{0} \tag{5.20}$$

Calling $\boldsymbol{\delta}_I$ the compatibility error of the solution of the reduced hybrid system

$$\boldsymbol{\delta}_I = \boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)} \tag{5.21}$$

the reference compatibility can be achieved by defining the averaged interface solutions $\hat{\boldsymbol{u}}_b^{(s)}$

$$\hat{\boldsymbol{u}}_b^{(1)} = \boldsymbol{u}_b^{(1)} + \frac{1}{2}\boldsymbol{\delta}_b \tag{5.22a}$$

$$\hat{\boldsymbol{u}}_b^{(2)} = \boldsymbol{u}_b^{(2)} - \frac{1}{2}\boldsymbol{\delta}_b \tag{5.22b}$$

From these definitions, it is clear that

$$\hat{\boldsymbol{u}}_b^{(1)} = \frac{1}{2}\left(\boldsymbol{u}_b^{(1)} + \boldsymbol{u}_b^{(2)}\right) = \hat{\boldsymbol{u}}_b^{(2)} \tag{5.23}$$

and the reference exact compatibility is satisfied. When more then two subdomains intersect, the averaged solution is defined by summing up the subdomains solutions and by dividing by the interface multiplicity.

In [47, 77], the authors also advocated back-propagating the effect of the above interface correction to the interior of each substructure, which implies the additional post-processing of each computed displacement field inside $\Omega^{(s)}$ by solving the Dirichlet problem where the averaged interface displacement is imposed:

$$\begin{bmatrix} \boldsymbol{K}_{ii}^{(s)} & \boldsymbol{K}_{ib}^{(s)} \\ \boldsymbol{K}_{bi}^{(s)} & \boldsymbol{K}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}}_i^{(s)} \\ \boldsymbol{u}_b^{(s)} = \hat{\boldsymbol{u}}_b^{(s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(s)} \\ \hat{\boldsymbol{f}}_b^{(s)} \end{bmatrix} \tag{5.24}$$

$\hat{\boldsymbol{f}}_b^{(s)}$ are the interface forces recovered from the solution of

$$\hat{\boldsymbol{u}}_i^{(s)} = \boldsymbol{K}_{ii}^{(s)^{-1}}\left(\boldsymbol{f}_i^{(s)} - \boldsymbol{K}_{ib}^{(s)}\hat{\boldsymbol{u}}_b^{(s)}\right) \tag{5.25}$$

$$= \boldsymbol{u}_i^{(s)} \pm \boldsymbol{K}_{ii}^{(s)^{-1}}\left(\boldsymbol{K}_{ib}^{(s)}\frac{\boldsymbol{\delta}_I}{2}\right) \tag{5.26}$$

In [47] this "patching" technique is proved not to deteriorate the asymptotic convergence rate of the approximate solution. As will be shown in the simple and homogeneous applications of Section 5.5, even if only very poor interface stress distribution functions are used, the averaging scheme leads to very good solutions when the interface solution is smooth. However, the optimality of this averaging scheme can be questioned in the presence of heterogeneities as will be shown in further applications.

5.4.2 Averaging of vibration eigenmodes

The averaging procedure defined here above for the weakly compatible static solution resulting from a reduced hybrid system can also be adapted for improving the results of the reduced hybrid eigenproblem. A reduced hybrid vibration eigenproblem is obtained by replacing the constraint matrices $\boldsymbol{B}$ by the reduced constraint matrices $\boldsymbol{B}^{(s),p}$ in the hybrid eigenproblem (3.32) and in the dual interface problem (3.39) inherent to an inverse iteration step.

The eigensolutions of such a reduced hybrid eigenproblem are weakly compatible on the interface. Two important consequences then follow: first the eigenvalues are underestimated compared to the eigenvalues of the reference problem, and secondly if the eigensolutions are utilized for constructing a mode superposition method for transient response analysis, the weak interface compatibility will be passed on to the transient solution. In practice, this means that the results of dynamic analysis with reduced interface compatibility are of very poor accuracy.

As in the static case, one observes that the approximate eigensolutions are fairly inaccurate mainly because of the spurious interface flexibility associated with the weakening of the compatibility. Therefore, averaging the eigenmodes appears to be an efficient way of enhancing the eigensolutions accuracy.

If the averaging technique is applied to the eigensolutions of the reduced hybrid eigenproblem [69], the averaged solutions no longer satisfy the orthogonality properties. The averaging step should thus be embedded in the solution procedure of the eigenproblem. Considering that all common eigensolvers are based on inverse iterations, we will explain how the averaging should be used in that case. The idea is to ensure that each and every iterate of the solution algorithm, i.e. the iterates of the basic inverse iteration and the subspace vectors, satisfy the reference strong compatibility so that the converged iterates defined in the strongly compatible basis are themselves compatible. The kernel of a typical inverse iteration step will be as follows:

- Solve a reduced hybrid system for a new iterate:

$$\begin{bmatrix} \boldsymbol{F}_I^p & \boldsymbol{G}_I^p \\ \boldsymbol{G}_I^{pT} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{p,n+1} \\ \boldsymbol{\alpha}^{p,n+1} \end{bmatrix} = \begin{bmatrix} \sum_{s=1}^{N_s} \boldsymbol{B}^{(s),p} \boldsymbol{K}^{(s)^+} \boldsymbol{M}^{(s)} \hat{\boldsymbol{u}}^{(s),n} \\ \left[\boldsymbol{R}^{(s)^T} \boldsymbol{M}^{(s)} \hat{\boldsymbol{u}}^{(s),n} \right] \end{bmatrix} \tag{5.27}$$

- Compute a weakly compatible new iterate:

$$\boldsymbol{u}^{(s),n+1} = \boldsymbol{K}^{(s)^+} \left(\boldsymbol{M}^{(s)} \hat{\boldsymbol{u}}^{(s),n} - \boldsymbol{B}^{(s),p} \boldsymbol{\lambda}^{p,n+1} \right) + \boldsymbol{R}^{(s)} \boldsymbol{\alpha}^{(s)p,n+1} \tag{5.28}$$

- Define a strongly compatible new iterate $\hat{\boldsymbol{u}}^{(s),n+1}$ by the averaging scheme. For two-subdomain, we write

$$\hat{\boldsymbol{u}}_b^{(1),n+1} = \hat{\boldsymbol{u}}_b^{(2),n+1} = \frac{\boldsymbol{u}_b^{(1),n+1} + \boldsymbol{u}_b^{(2),n+1}}{2} \tag{5.29a}$$

$$\hat{\boldsymbol{u}}_i^{(s),n+1} = \boldsymbol{K}_{ii}^{(s)^{-1}} \left(\boldsymbol{M}^{(s)} \boldsymbol{u}^{(s),n} - \boldsymbol{K}_{ib}^{(s)} \hat{\boldsymbol{u}}_b^{(s),n+1} \right) \tag{5.29b}$$

The averaged eigensolutions are strongly compatible on the interface Γ_I but satisfy the interface equilibrium only approximately. The combination of reducing the dual interface space and then averaging can be seen as a kinematically admissible approach and hence the averaged eigensolutions have overestimated eigenfrequencies.

5.4.3 Averaging the transient response solution

When a mode superposition method is used, the averaged eigensolutions can be used without any modification to the usual normal equations method since orthonormality is enforced within the eigenproblem solver.

When direct integration is used to time-integrate the reduced two-field hybrid equations for the dynamic response of a system, an averaging step should be included after each time-step solution obtained by (3.54) or (3.51). When using the Newmark integration scheme, it means that the solution $\ddot{\boldsymbol{u}}_d^{n+1}$ of (3.51) should be averaged in the same way as the displacements are averaged in the static problem. For the two-subdomains case we write:

$$\hat{\ddot{\boldsymbol{u}}}_b^{(1)_{n+1}} = \frac{1}{2}\left(\ddot{\boldsymbol{u}}_b^{(1)_{n+1}} + \ddot{\boldsymbol{u}}_b^{(2)_{n+1}}\right) = \hat{\boldsymbol{u}}_b^{(2)_{n+1}} \tag{5.30a}$$

$$\hat{\ddot{\boldsymbol{u}}}_i^{(s)_{n+1}} = \ddot{\boldsymbol{u}}_i^{(s)_{n+1}} \pm \boldsymbol{D}_{ii}^{(s)^{-1}}\left(\boldsymbol{D}_{ib}^{(s)}\frac{\ddot{\boldsymbol{u}}_b^{(2)_{n+1}} - \ddot{\boldsymbol{u}}_b^{(1)_{n+1}}}{2}\right) \tag{5.30b}$$

Hence every new iteration step will start from a strongly compatible solution and finally the entire transient solution will be compatible.

Remark: interface averaging and group implicit integration

A particular time-integration algorithm can be deduced from the reduced hybrid method with averaging if we consider the limit case when $n_{\lambda^p} = 0$, i.e. when no interface stress distribution is defined on Γ_I. The inter-subdomain compatibility is then entirely enforced by the averaging procedure applied after each time step. For the two-subdomain case, the resulting scheme then looks as follows:

predictor
$$\boldsymbol{u}^{(s)*_{n+1}} = \boldsymbol{u}^{(s)_n} + h\dot{\boldsymbol{u}}^{(s)_n} + \frac{h^2}{4}\ddot{\boldsymbol{u}}^{(s)_n} \tag{5.31a}$$

$$\dot{\boldsymbol{u}}^{(s)*_{n+1}} = \dot{\boldsymbol{u}}^{(s)_n} + \frac{h}{2}\ddot{\boldsymbol{u}}^{(s)_n} \tag{5.31b}$$

solve
$$\boldsymbol{D}^{(s)}\ddot{\boldsymbol{u}}^{(s)_n+1} = \boldsymbol{f}^{(s)_n+1} - \boldsymbol{C}^{(s)}\dot{\boldsymbol{u}}^{(s)*_{n+1}} - \boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)*_{n+1}} \tag{5.31c}$$

averaging
$$\hat{\ddot{\boldsymbol{u}}}_b^{(1)_{n+1}} = \hat{\ddot{\boldsymbol{u}}}_b^{(2)_n+1} = \frac{1}{2}\left(\ddot{\boldsymbol{u}}_b^{(1)_{n+1}} + \ddot{\boldsymbol{u}}_b^{(2)_{n+1}}\right) \tag{5.31d}$$

$$\hat{\ddot{\boldsymbol{u}}}_i^{(s)_{n+1}} = \ddot{\boldsymbol{u}}_i^{(s)_{n+1}} \pm \boldsymbol{D}_{ii}^{(s)^{-1}}\left(\boldsymbol{D}_{ib}^{(s)}\frac{\ddot{\boldsymbol{u}}_b^{(2)_{n+1}} - \ddot{\boldsymbol{u}}_b^{(1)_{n+1}}}{2}\right) \tag{5.31e}$$

correction
$$\boldsymbol{u}^{(s)_{n+1}} = \boldsymbol{u}^{(s)*_{n+1}} + \frac{h^2}{4}\hat{\ddot{\boldsymbol{u}}}^{(s)_{n+1}} \tag{5.31f}$$

$$\dot{\boldsymbol{u}}^{(s)_{n+1}} = \dot{\boldsymbol{u}}^{(s)*_{n+1}} + \frac{h}{2}\hat{\ddot{\boldsymbol{u}}}^{(s)_{n+1}} \tag{5.31g}$$

This iteration scheme solves an implicit integration step independently within each subdomain and data must be communicated from one subdomain to another only for the averaging. For this reason, this algorithm is highly suitable for parallel computation. It can be seen as a *group implicit* integration scheme where the subdomains are integrated with an implicit Newmark algorithm whereas the inter-subdomain behavior is computed explicitly. If the averaging takes account of the relative inertia [100, 133], the inter-subdomain integration corresponds to a central difference scheme.

This implicit-explicit scheme has shown to be an unconditionally stable concurrent procedure [133], but unfortunately the accuracy of such methods is questionable since in [83] it was proven that this algorithm is only conditionally consistent: it transmits information only between neighboring subdomains within one time step. In [83] it is shown that if the spatial and time discretization are refined, the consistency of the scheme requires that the time-step goes to zero faster than the spatial step. For practical time-steps, one observes that the conditional consistency severely delays the wave front in hyperbolic problems which results in poor accuracy.

When n_{λ^p} is not zero, at least a weak compatibility is imposed at every time step within the implicit integration and thus information is exchanged among all subdomains. Therefore, if the reduced discretization of the interface stresses is rich enough, the reduced hybrid system with averaging does not suffer the same accuracy problems as the group implicit schemes.

5.5 Applications

We will present some applications that illustrate the efficiency of the reduced two-field hybrid method and of the averaging procedure. First we will apply the method to the linear static analysis of 2D structures. Then results of a modal and transient analysis will be presented.

5.5.1 Static analysis

As first example we consider the static analysis of an unsymmetric two-subdomain ($\Omega^{(1)}$, $\Omega^{(2)}$) beam problem [83,151]. Each substructure is clamped at one end, and the left one is subjected to horizontal and vertical point loadings (Fig. 5.1). Both substructures are discretized using 4-node plane stress elements with two degrees of freedom per node. Each finite element model contains 126 interior and 18 interface d.o.f.

In this problem, the substructures have matching interfaces. Therefore, they can be strongly glued together with discrete Lagrange multipliers introduced at each of the 18 interface d.o.f. When this option is chosen, we call the resulting displacement solution the reference solution ($\boldsymbol{u}^{ref}$).

Low-order polynomial Lagrange multipliers can also be used to enforce in a weaker sense the interface continuity conditions. For example, Fig. 5.2 depicts the computed horizontal (u_{b_x}) and vertical (u_{b_y}) displacements along the interface boundary Γ_I when polynomials of degree $p = 0, 1, 2, 3$ are used to approximate the interface tractions.

In order to measure the quality of the solution, we will compute a relative energy norm of the displacement errors for the solution of the reduced hybrid system [47] which is equally suitable for the weakly compatible and for the averaged solution. Calling $\boldsymbol{u}^{(s)}$, $\hat{\boldsymbol{u}}^{(s)}$ and $\boldsymbol{u}^{(s),ref}$ respectively the displacement solution computed by the reduced hybrid system, the averaged solution and the reference solution, we define

$$\varepsilon = \frac{\sqrt{\sum_{s=1}^{N_s} \left\| \boldsymbol{u}^{(s),ref} - \boldsymbol{u}^{(s)} \right\|^2_{\boldsymbol{K}^{(s)}}}}{\sqrt{\sum_{s=1}^{N_s} \left\| \boldsymbol{u}^{(s),ref} \right\|^2_{\boldsymbol{K}^{(s)}}}} \tag{5.32}$$

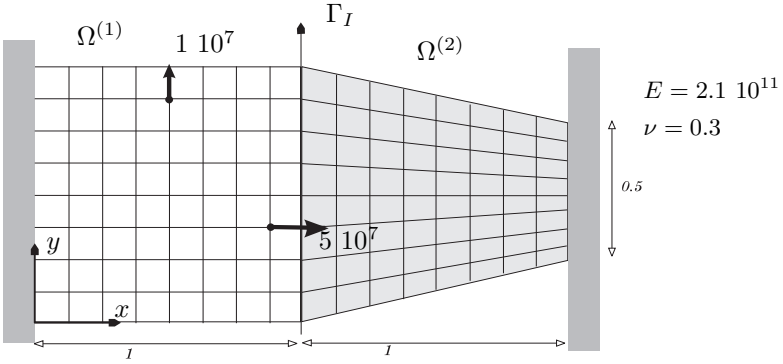


Figure 5.1: Clamped-clamped unsymmetric two-substructure beam problem

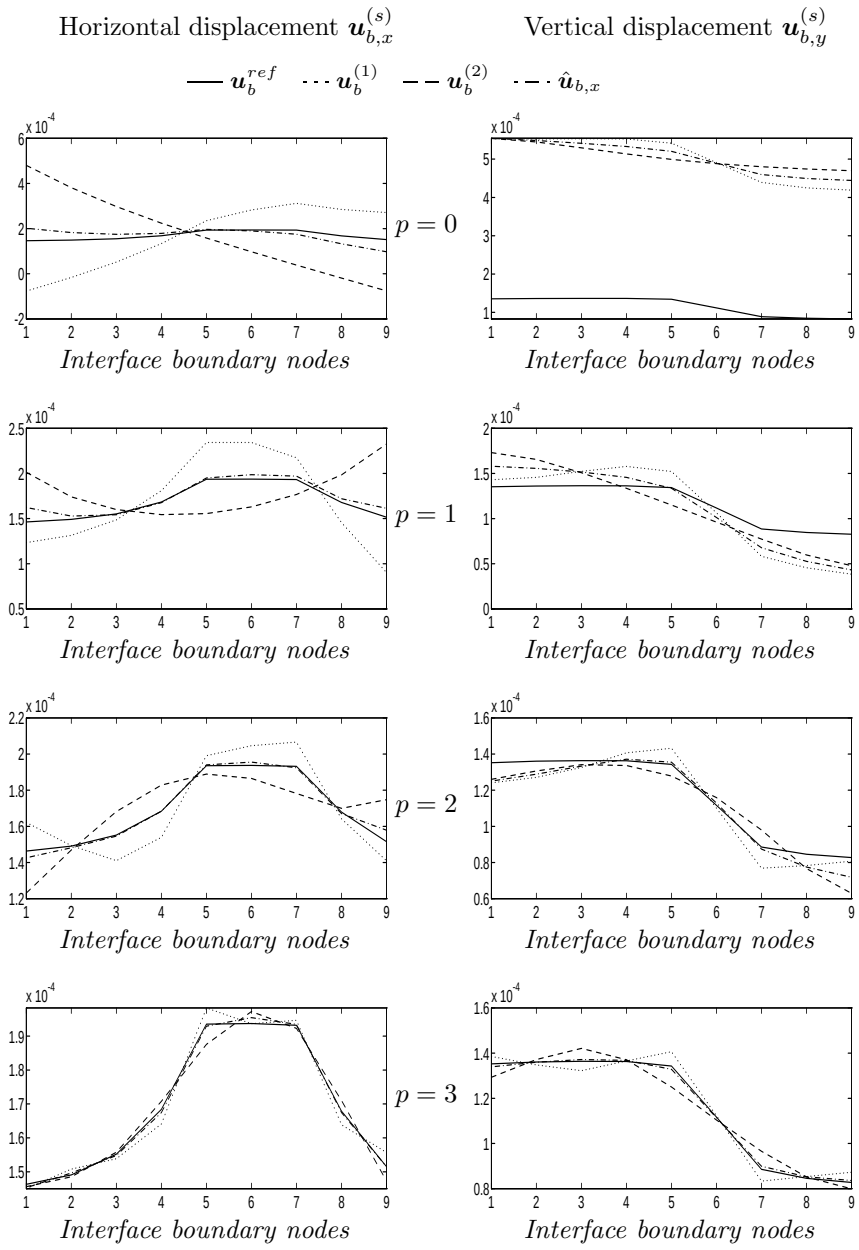


Figure 5.2: Unsymmetric beam: interface displacement solutions for reduced hybrid system

$$\hat{\varepsilon} = \frac{\sqrt{\sum_{s=1}^{N_s} \left\| \boldsymbol{u}^{(s),ref} - \hat{\boldsymbol{u}}^{(s)} \right\|^2_{\boldsymbol{K}^{(s)}}}}{\sqrt{\sum_{s=1}^{N_s} \left\| \boldsymbol{u}^{(s),ref} \right\|^2_{\boldsymbol{K}^{(s)}}}} \tag{5.33}$$

The results for the unsymmetric beam are given in Table 5.1.

order of pol.	$p=0$	$p=1$	$p=2$	$p=3$	$p=4$
ε	0.48	0.26	0.13	0.071	0.047
$\hat{\varepsilon}$	0.63	0.13	0.046	0.015	0.010

Table 5.1: Energy norm of solution error (unsymmetric beam)

From the result curves in Fig. 5.2 and from Table 5.1, the reader can observe that:

- a second-order polynomial approximation of the Lagrange multipliers generates for this problem 6 interface unknowns only, and the corresponding averaged displacement solution is in excellent agreement with the exact solution that requires 18 interface unknowns.
- in all cases (except for the highly inaccurate case $p = 0$), the averaged substructure displacement solutions are more accurate than the non-averaged ones and the energy norm of the error converges much faster when the solution is averaged.

Next we analyze the homogeneous clamped beam model considered for eigenvalue analysis in section 3.3: as shown in Fig. 5.3 each of the four subdomains are modeled by 10×10 plane stress elements, and vertical and horizontal loads are applied on the upper right edge of the structure. The domain-wise energy norms of the displacement errors defined by (5.32-5.33) are listed in Table 5.2 when polynomials of order p are taken as stress distribution functions on every interface Γ_I^{sr}, the polynomial order being equal to $p = 0, 1, 2, 3, 4, 5$.

As for the unsymmetric beam, Table 5.2 shows that the averaged solution converges more rapidly to the reference solution as p increases. Moreover, we observe that when p is equal to 2, the relative energy norm of the displacement error is less than 1 percent. In this case 3 Lagrange multipliers are defined on each of the 4 interfaces (the cross-point being neglected for the approximate compatibility definition) and for each displacement direction u_x and u_y: hence $n_{\lambda^p} = 24$. Observing that the number of independent compatibility constraints for exact interface

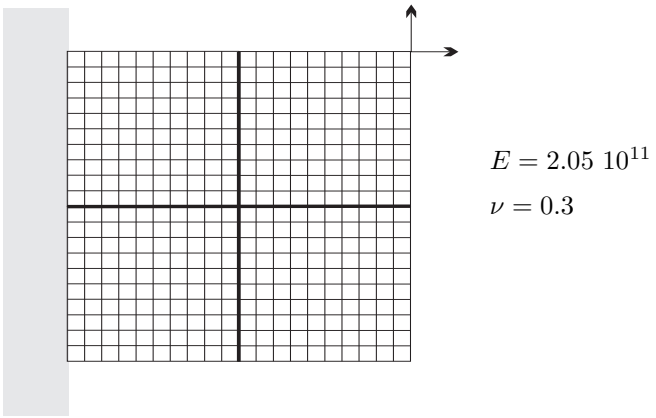


Figure 5.3: Homogeneous clamped beam problem

order of pol.	$p=0$	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
ε	0.42	0.055	$12\ 10^{-3}$	$2.8\ 10^{-3}$	$8.7\ 10^{-4}$	$51\ 10^{-5}$
$\hat{\varepsilon}$	0.84	0.039	$5.5\ 10^{-3}$	$1.1\ 10^{-3}$	$2.0\ 10^{-4}$	$9.9\ 10^{-5}$

Table 5.2: Energy norm of solution error (homogeneous clamped beam)

compatibility is $n_\lambda = 84$, we see that good accuracy can be achieved with a significantly reduced number of interface variables.

If we refine the meshing in every subdomain so that they are modeled by 20×20 finite elements, the energy norm of the displacement errors listed in Table 5.3 are totally similar to the results obtained for the coarser grid. For the refined grid, the number of independent constraints associ-

order of pol.	$p=0$	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
ε	0.39	0.054	$12\ 10^{-3}$	$2.9\ 10^{-3}$	$6.6\ 10^{-4}$	$3.0\ 10^{-4}$
$\hat{\varepsilon}$	0.87	0.044	$6.9\ 10^{-3}$	$1.6\ 10^{-3}$	$2.6\ 10^{-4}$	$1.0\ 10^{-4}$

Table 5.3: Energy norm of solution error (clamped beam, refined mesh)

ated with the reference solution is $n_\lambda = 164$. We thus conclude that the reduced hybrid method allows to accurately represent the solution in the global structure depending only on the solution itself and not on the underlying finite element mesh. This proves the tremendous efficiency of the method in terms of number of interface unknowns reduction.

More applications examples can be found in [77].

5.5.2 Modal analysis

As an example of modal analysis when reducing the dual interface problem, let us consider again the unsymmetric beam structure modeled by plane stress elements and decomposed into two subdomains (Fig. 5.1). The 12 first eigensolutions of the structure are first computed without introducing any averaging in the inverse iterations. The computed circular eigenfrequencies ω corresponding to a polynomial interface stress distribution of order $p = 1$ and $p = 2$ are plotted in Fig. 5.4 together with the reference eigenfrequencies ω^{ref} obtained for the fully assembled problem. The relative eigenfrequency error is also shown.

From Fig. 5.4 we deduce that even for a polynomial order as low as $p = 1$ ($n_{\lambda^p} = 4$) the first 7 eigenfrequencies are approximated within a relative error of 1 percent, and for $p = 2$ ($n_{\lambda^p} = 6$) all first 12 eigenfrequencies have this precision. The eighth mode obtained with $p = 1$ has an eigenfrequency with a small relative error of $4.2\ 10^{-2}$. However if we visualize the horizontal displacements of the associated eigenmode (Fig. 5.5) we observe that it is far from any admissible mode of the assembled system: the interface compatibility is really poor and the computed mode is totally spurious.

We then consider again the same reduced hybrid eigenvalue problem but this time the eigensolutions are computed by adding an averaging step after each inverse iteration. The eigensolutions are computed with a subspace algorithm [91] and the eigenfrequencies are plotted in Fig. 5.6. This time the eigenfrequencies are overestimated, but the relative error on the eigenfrequencies is one order of magnitude lower. We thus conclude that introducing an averaging procedure within the inverse iteration based eigensolvers significantly improves the solution.

5.5.3 Transient response analysis

Let us consider once more the unsymmetric beam example and let us assume now that horizontal and vertical forces are applied stepwise (Fig. 5.7) during a time interval $T_8/3 = 7.6\ 10^{-5}s$, where T_8 is the period of the eighth eigensolution of the associated free-vibration problem. Note that

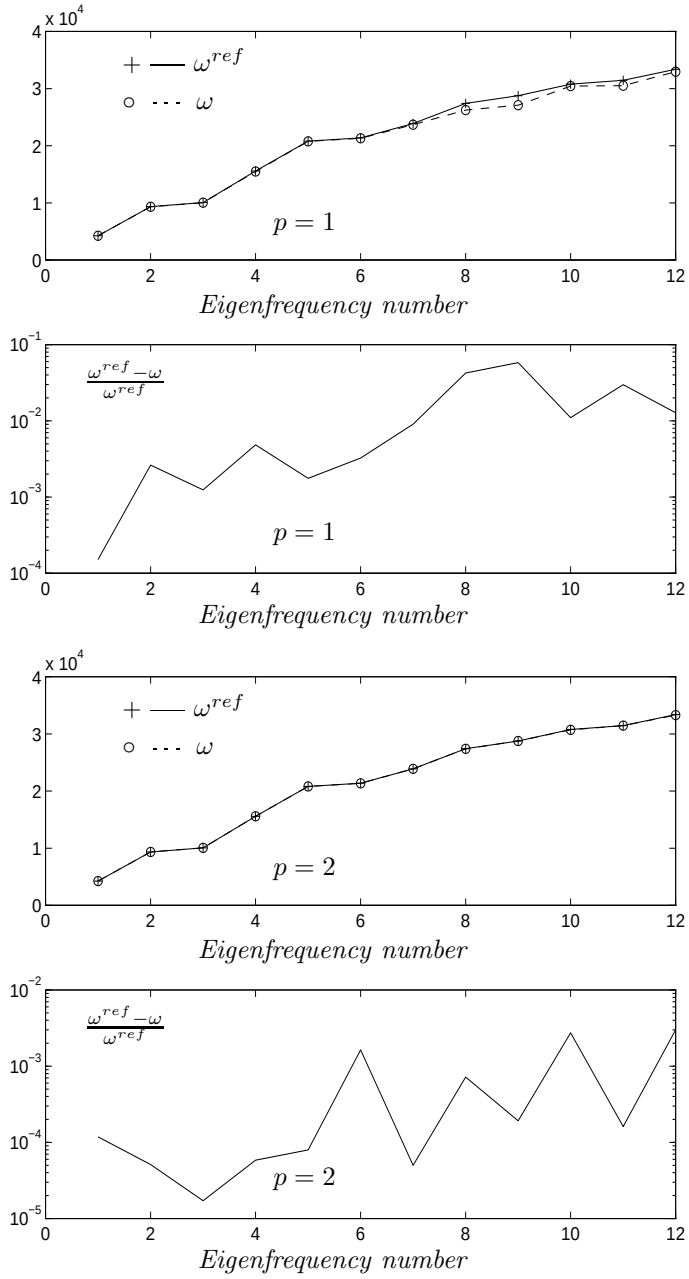


Figure 5.4: Eigenfrequencies of the unsymmetric beam

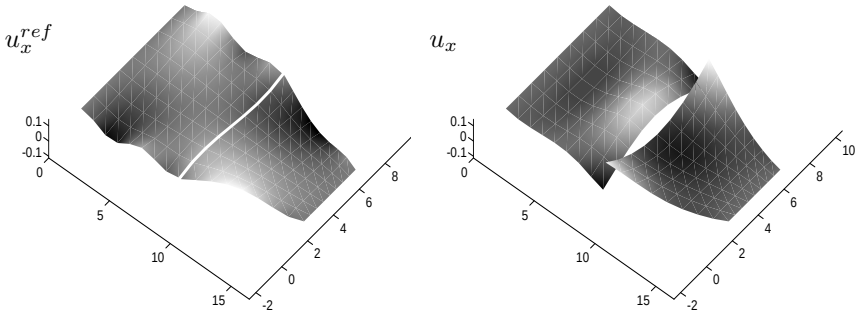


Figure 5.5: Spurious eigenmode of the unsymmetric beam (non-averaged mode 8 with $p = 1$)

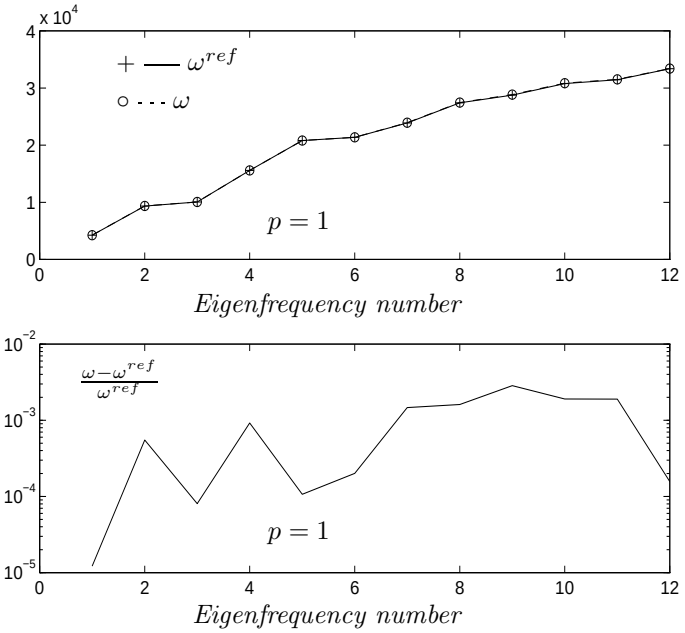


Figure 5.6: Eigenfrequencies of the unsymmetric beam (with averaging)

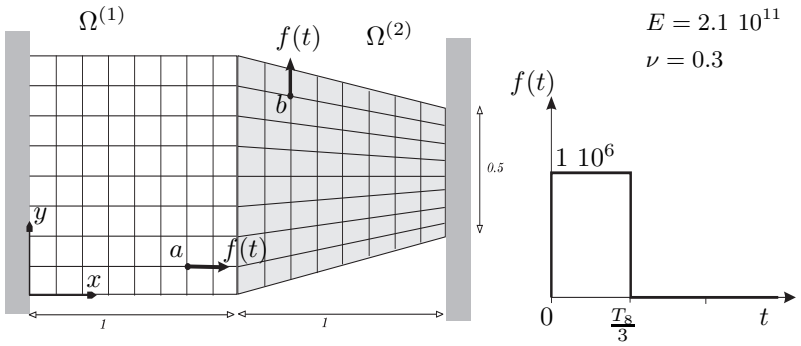


Figure 5.7: Unsymmetric beam with dynamic loading

this loading defines a fairly tricky problem since its eigenfrequency content is such that it excites higher eigenmodes that are very badly approximated by the reduced interface problem (see the discussion in the previous section). The time-step we use for the Newmark (trapezoidal) scheme is equal to $T_8/21$.

The transient response is computed by a reduced hybrid method where the interface stress distribution is a polynomial of order as low as $p = 1$. The vertical displacement of the point b (Fig. 5.7) where the vertical load is applied is plotted in Fig. 5.8: the non-averaged solution of the reduced hybrid system is compared with the reference solution, that is the fully assembled solution. The same figure also shows the transient response computed when averaging is applied after each time step.

We observe that only the low frequency content of the solution is well represented by the reduced hybrid solution when no averaging is used. This effect is a direct consequence of the poor representation of the higher eigensolutions when $p = 1$ and when no averaging is used. When averaging is applied, the solution of the reduced hybrid system is very close to the reference solution which shows the effectiveness of an averaging scheme for transient response computation.

5.6 Effect of substructure heterogeneities

The averaging procedure outlined in Eqs. (5.23-5.26) is justifiable only when the two substructures have roughly the same stiffness, which is unfortunately not true for all applications. Hence, in the remainder of part I of this work, we will pursue a generalization of the patching (or averaging) procedure proposed in [47, 77] to the case of arbitrary substructures. However, before discussing such an extension, we begin with highlighting the effect of substructure heterogeneities on the accuracy of the displacement solutions computed by the two-field hybrid method.

For this purpose, we consider a modified version of the two-substructure unsymmetric beam described in section 5.5, Fig. 5.1, where the right substructure $\Omega^{(2)}$ is assumed to have a Young modulus equal to α times the value of that of the left substructure $\Omega^{(1)}$ ($\alpha = E^{(2)}/E^{(1)}$). We focus on second- and third-order polynomial approximations ($p = 2, 3$) of the Lagrange multipliers because these have been shown to produce excellent solutions in the homogeneous case $\alpha = 1$ (see Fig. 5.2). For all three values $\alpha = 1.0, 0.4, 0.2$, the horizontal and vertical substructure displacements are computed using the two-field hybrid method described in section 5.1, and their traces on Γ_I are graphically depicted in Fig. 5.9 ($p = 2$) and Fig. 5.10 ($p = 3$).

For $\alpha = 1.0$ – that is, for the case of a homogeneous two-substructure problem – the traces on Γ_I of the substructure displacements computed via the two-field hybrid method without any averaging procedure are shown to be nearly equally apart from the exact solution. For $\alpha \neq 1.0$ — that is, for the case of a heterogeneous two-substructure problem — the trace on the interface boundary of the displacement field associated with the stiffer substructure is shown to be closer to the exact solution than that associated with the softer one. For $\alpha = 0.2$ and $p = 3$, the reader can observe that the computed interface displacement of the stiffer substructure is almost equal to the exact solution, while that of the softer substructure is far from it. Therefore, the simple averaging procedure is clearly optimal in the homogeneous case, and sub-optimal in the heterogeneous case.

To further illustrate the effect of subdomain heterogeneities, let us consider the truss frame depicted in Fig. 5.11 [77]. The bars that compose the two subdomains of this structure are identical, but the right substructure $\Omega^{(2)}$ has extra horizontal bars which greatly increases its overall stiffness.

The interface Γ_I is assumed to be the line running across the connection points between the two substructures. The nodal connection forces are assumed to lie on a polynomial curve of degree $p = 1, 3, 5$. Here, the reduced constraint matrices which are generally obtained by integrating the product of the interface displacement shape functions and stress distribution (Eq. (5.4)) are simply constructed by evaluating the polynomials at the connection points. The horizontal and vertical displacements computed by the reduced hybrid method are plotted in Fig. 5.12 and they are compared to the reference solution of the fully assembled problem.

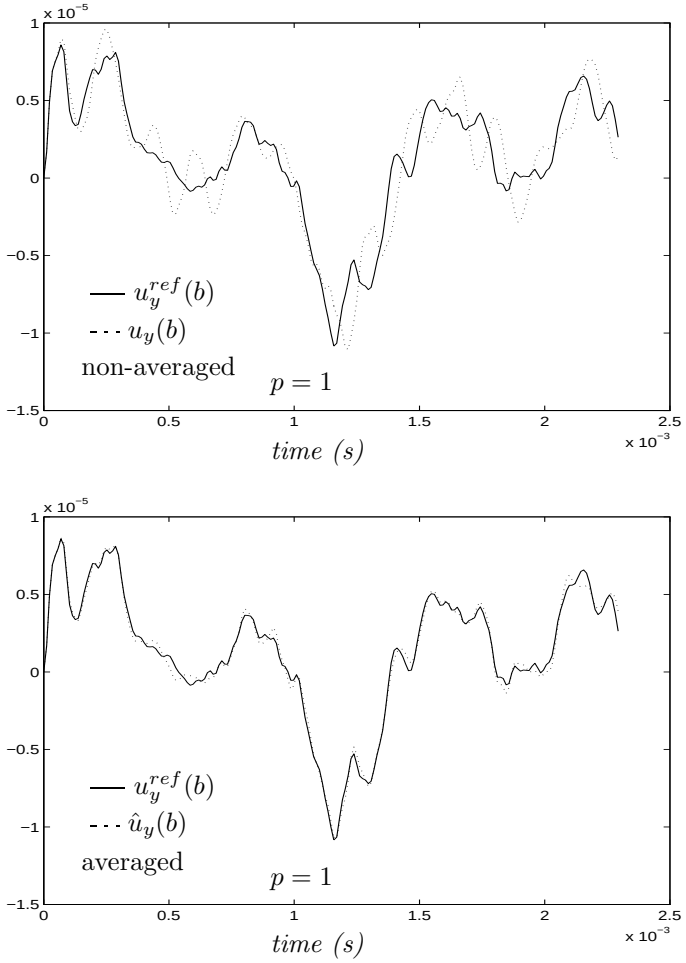


Figure 5.8: Unsymmetric beam: transient response computed by a reduced hybrid system

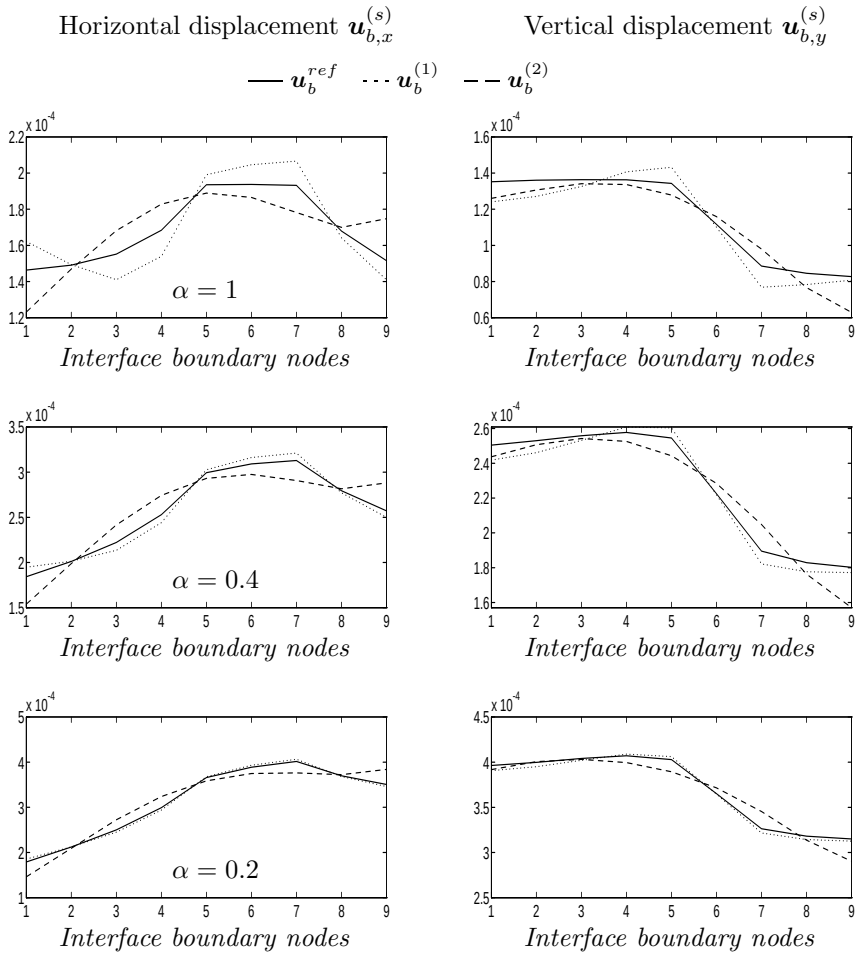


Figure 5.9: Heterogeneous unsymmetric beam: interface displacements computed by a reduced hybrid system ($p = 2$)

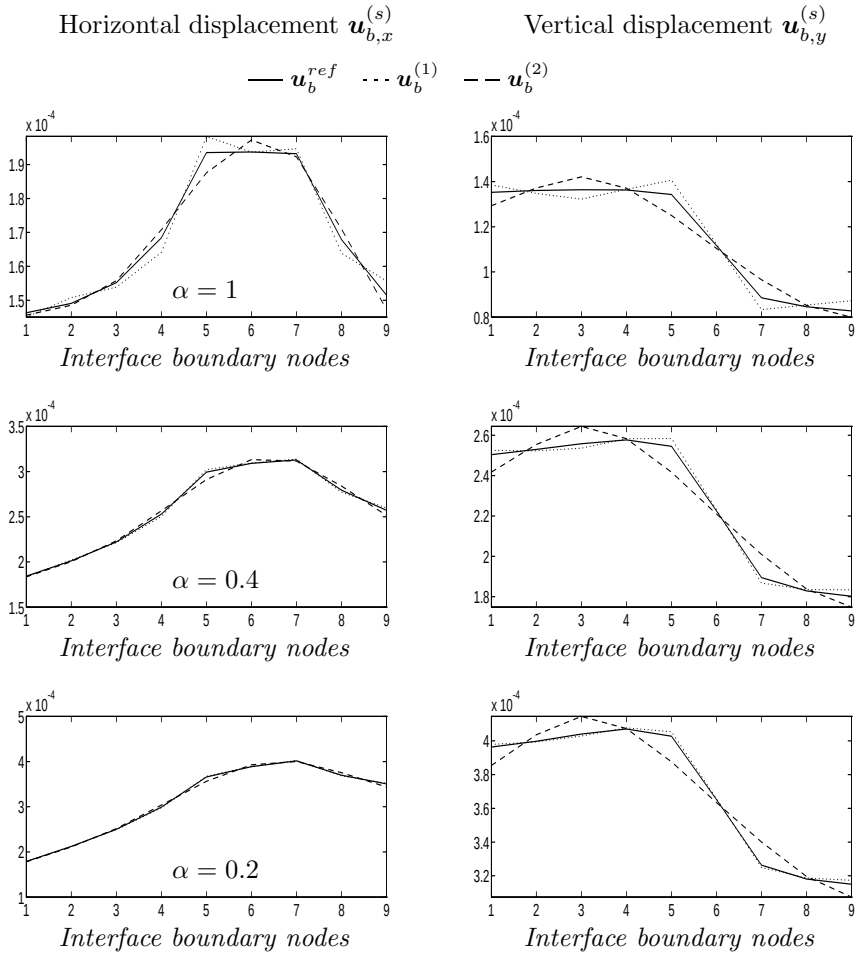


Figure 5.10: Heterogeneous unsymmetric beam: interface displacements computed by a reduced hybrid system ($p = 3$)

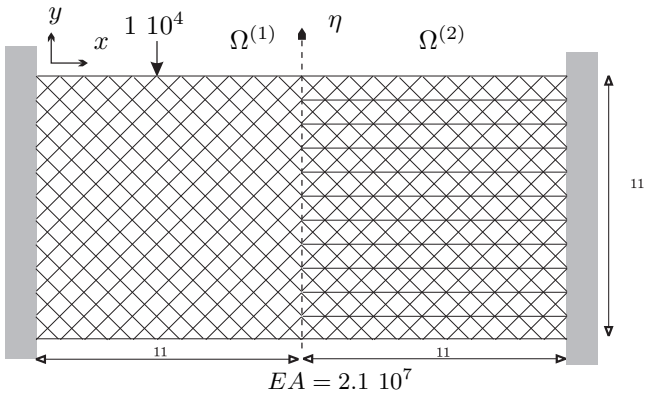


Figure 5.11: Truss frame structure

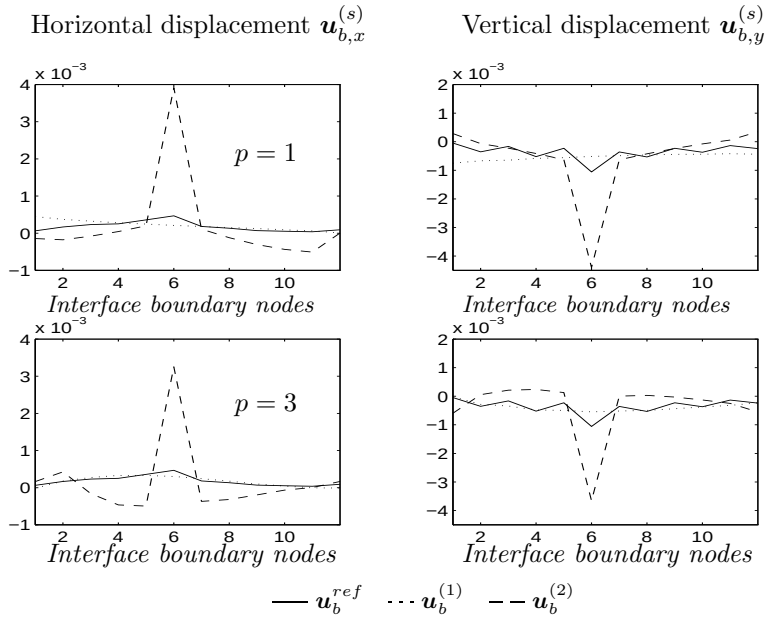


Figure 5.12: Computed interface displacements for the truss frame structure

The results clearly highlight two points:

- The connection loads on the interface of the truss frame can not be accuratly approximated by polynomial functions. In fact, for this truss problem, the interface solution is far from smooth: the interface displacements (Fig. 5.12) and the associated interface connection loads (Fig. 5.13) are clearly irregular.
- Even though reducing the interface problem by polynomials for this kind of structure is far from optimum, the result curves for the interface displacements of $\Omega^{(2)}$, namely on the stiffest side, are fairly close to the reference solution.

We thus conclude that the simple averaging scheme is far from optimum for heterogeneous problems, but if we can define a method that determines a new averaging scheme and which gives more importance to the stiffest side when defining an a posteriori compatible solution, one could take advantage of the subdomain heterogeneities to define good approximate solutions. Such a technique will be developed in the next chapters, first for the two-subdomain case with a conforming interface and then for the general case of N_s subdomains having conforming or non-conforming interfaces.

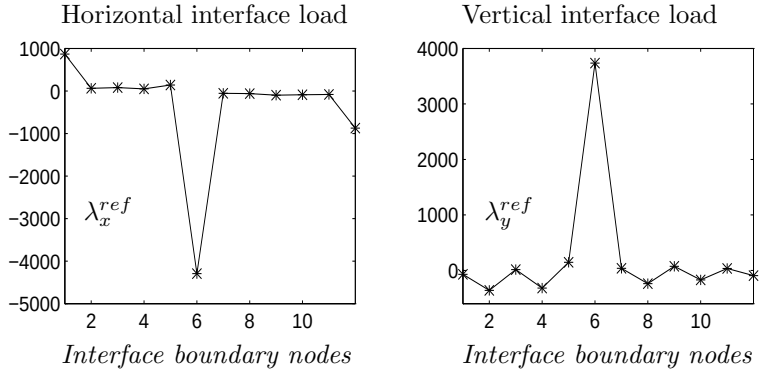


Figure 5.13: Interface loads of the assembled truss frame

Chapter 6

A Rayleigh-Ritz based smoothing procedure - the case of two conforming subdomains

In order not to obscure the main idea of the smoothing method by the complexity of the notation needed for a problem with an arbitrary number of substructures and with non-matching interfaces, we consider first the case of two heterogeneous substructures with conforming discrete interfaces. The general case of a system including multiply ($N_s > 2$) and arbitrarily connected substructures with non-matching interfaces will be treated in the next chapter.

6.1 From a statically admissible to a kinematically compatible system

Using as before the subscripts i and b for the internal and boundary d.o.f. and assuming that the internal d.o.f. are numbered first, the two-field hybrid system for two subdomains can be written as

$$\begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{K}_{ib}^{(1)^T} & \boldsymbol{K}_{bb}^{(1)} & \mathbf{0} & \mathbf{0} & \boldsymbol{b}^{(1),p^T} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{K}_{ii}^{(2)} & \boldsymbol{K}_{ib}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{K}_{ib}^{(2)^T} & \boldsymbol{K}_{bb}^{(2)} & \boldsymbol{b}^{(2),p^T} \\ \mathbf{0} & \boldsymbol{b}^{(1),p} & \mathbf{0} & \boldsymbol{b}^{(2),p} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b^{(1)} \\ \boldsymbol{u}_i^{(2)} \\ \boldsymbol{u}_b^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} \\ \boldsymbol{f}_i^{(2)} \\ \boldsymbol{f}_b^{(2)} \\ \mathbf{0} \end{bmatrix} \tag{6.1}$$

The averaging procedure summarized in Eqs. (5.23) for two substructures sharing a matching interface can be generalized as follows:

$$\hat{\boldsymbol{u}}_b^{(1)} = \hat{\boldsymbol{u}}_b^{(2)} = a\boldsymbol{u}_b^{(2)} + b\boldsymbol{u}_b^{(1)} \tag{6.2}$$

Here, a and b are real numbers that are constant along the edge connecting the two substructures. We impose that $a + b = 1$ so that when the non-averaged solution satisfies the reference compatibility, i.e $\boldsymbol{u}_b^{(2)} = \boldsymbol{u}_b^{(1)}$, the averaging procedure does not modify the solution. The generalized

averaging procedure thus writes

$$
\begin{aligned}
\hat{\boldsymbol{u}}_b^{(1)} = \hat{\boldsymbol{u}}_b^{(2)} &= \hat{\boldsymbol{u}}_I = (1-a)\boldsymbol{u}_b^{(1)} + a\boldsymbol{u}_b^{(2)} && \text{(6.3a)}\\
\hat{\boldsymbol{u}}_i^{(s)} &= \boldsymbol{K}_{ii}^{(s)^{-1}}(\boldsymbol{f}_i^{(s)} - \boldsymbol{K}_{ib}^{(s)}\hat{\boldsymbol{u}}_b^{(s)}) && \text{(6.3b)}\\
&= \boldsymbol{u}_i^{(s)} - \boldsymbol{K}_{ii}^{(s)^{-1}}\boldsymbol{K}_{ib}^{(s)}(\hat{\boldsymbol{u}}_b^{(s)} - \boldsymbol{u}_b^{(s)}) && \text{(6.3c)}\\
& \qquad s \;=\; 1,\; 2
\end{aligned}
$$

Let $\boldsymbol{\delta}_I$ denote as before the displacement jump on Γ_I defined as

$$\boldsymbol{\delta}_I = \boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)} \tag{6.4}$$

From the relations above, it follows that $\hat{\boldsymbol{u}}_b^{(1)}$ and $\hat{\boldsymbol{u}}_b^{(2)}$ can be re-written as

$$
\begin{aligned}
\hat{\boldsymbol{u}}_b^{(1)} &= \boldsymbol{u}_b^{(1)} + \Delta\boldsymbol{u}_b^{(1)} && \text{(6.5)}\\
\hat{\boldsymbol{u}}_b^{(2)} &= \boldsymbol{u}_b^{(2)} + \Delta\boldsymbol{u}_b^{(2)} && \text{(6.6)}\\
& && \text{(6.7)}
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta\boldsymbol{u}_b^{(1)} &= a\boldsymbol{\delta}_I && \text{(6.8)}\\
\Delta\boldsymbol{u}_b^{(2)} &= -(1-a)\boldsymbol{\delta}_I && \text{(6.9)}
\end{aligned}
$$

This procedure is called *smoothing* as opposed to averaging where the weighting factor a is taken as 0.5. Before discussing the optimal selection of the parameter a, we note that the smoothing procedure (6.3) destroys equilibrium while enforcing the desired strong compatibility on Γ_I. However, because the effect of the interface smoothing is back-propagated to the internal d.o.f. (see Eq. (6.3b)), a force residual is introduced only on the interface Γ_I. Hence after post-processing, the substructure equations of equilibrium become

$$
\begin{bmatrix}
\boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} & \boldsymbol{0}\\
\boldsymbol{K}_{ib}^{(1)^T} & \boldsymbol{K}_{bb}^{(1)} + \boldsymbol{K}_{bb}^{(2)} & \boldsymbol{K}_{ib}^{(2)}\\
\boldsymbol{0} & \boldsymbol{K}_{ib}^{(2)^T} & \boldsymbol{K}_{ii}^{(2)}
\end{bmatrix}
\begin{bmatrix}
\hat{\boldsymbol{u}}_i^{(1)}\\ \hat{\boldsymbol{u}}_I\\ \hat{\boldsymbol{u}}_i^{(2)}
\end{bmatrix}
=
\begin{bmatrix}
\boldsymbol{f}_i^{(1)}\\ \boldsymbol{f}_b^{(1)} + \boldsymbol{f}_b^{(2)}\\ \boldsymbol{f}_i^{(2)}
\end{bmatrix}
+
\begin{bmatrix}
\boldsymbol{0}\\ \boldsymbol{r}_b\\ \boldsymbol{0}
\end{bmatrix}
\tag{6.10}
$$

From Eq. (6.1) and Eq. (6.10), and after some algebraic manipulations, it follows that the interface residual caused by the proposed smoothing procedure can be written as

$$\boldsymbol{r}_b = \boldsymbol{S}_{bb}^{(1)}\Delta\boldsymbol{u}_b^{(1)} + \boldsymbol{S}_{bb}^{(2)}\Delta\boldsymbol{u}_b^{(2)} \tag{6.11}$$

where $\boldsymbol{S}_{bb}^{(s)}$ denote the Schur complement matrices.

Clearly, this last equation shows that the interface residual $\boldsymbol{r}_b$ is directly related and only related to the displacement corrections resulting from enforcing a posteriori a strong compatibility on Γ_I. This residual can also be expressed in terms of the displacement jump $\boldsymbol{\delta}_I$ as follows

$$\boldsymbol{r}_b = \boldsymbol{r}_b(a) = \left(a\boldsymbol{S}_{bb}^{(1)} + (a-1)\boldsymbol{S}_{bb}^{(2)}\right)\boldsymbol{\delta}_I \tag{6.12}$$

In summary, after $\boldsymbol{u}^{(1)}$ and $\boldsymbol{u}^{(2)}$ have been obtained from the solution of Eqs. (6.1) that are associated with the two-field hybrid method, the jump $\boldsymbol{\delta}_I$ is readily available, and the parameter a of the smoothing procedure (6.3) proposed for enforcing the desired strong compatibility along Γ_I should be selected as to minimize the undesired side-effect $\boldsymbol{r}_b$.

6.2 Energy minimization

Rather than minimizing directly some norm of the interface residual $\boldsymbol{r}_b$, we propose to adopt a Rayleigh-Ritz approach where the post-processed displacement solutions $\hat{\boldsymbol{u}}^{(1)}(a)$ and $\hat{\boldsymbol{u}}^{(2)}(a)$ given in Eqs. (6.3a) are viewed as *kinematically admissible fields* parameterized by a, and we propose to minimize the corresponding energy of the global system. For the two-substructure problem discussed above, the total energy associated with the compatible smoothed solution can be written as

$$\begin{aligned}\mathcal{E}(a) &= \frac{1}{2}\left[\begin{array}{ccc} \hat{\boldsymbol{u}}_i^{(1)^T} & \hat{\boldsymbol{u}}_I^T & \hat{\boldsymbol{u}}_i^{(2)^T} \end{array}\right] \left[\begin{array}{ccc} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} & 0 \\ \boldsymbol{K}_{ib}^{(1)^T} & \boldsymbol{K}_{bb}^{(1)} + \boldsymbol{K}_{bb}^{(2)} & \boldsymbol{K}_{ib}^{(2)} \\ 0 & \boldsymbol{K}_{ib}^{(2)^T} & \boldsymbol{K}_{ii}^{(2)} \end{array}\right] \left[\begin{array}{c} \hat{\boldsymbol{u}}_i^{(1)} \\ \hat{\boldsymbol{u}}_I \\ \hat{\boldsymbol{u}}_i^{(2)} \end{array}\right] \\ &\quad - \left[\begin{array}{ccc} \hat{\boldsymbol{u}}_i^{(1)^T} & \hat{\boldsymbol{u}}_I^T & \hat{\boldsymbol{u}}_i^{(2)^T} \end{array}\right] \left[\begin{array}{c} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} + \boldsymbol{f}_b^{(2)} \\ \boldsymbol{f}_i^{(2)} \end{array}\right] \end{aligned} \tag{6.13}$$

which, in view of Eq. (6.10) can also be expressed as

$$\mathcal{E}(a) = \frac{1}{2}\left[\begin{array}{ccc} \hat{\boldsymbol{u}}_i^{(1)^T} & \hat{\boldsymbol{u}}_I & \hat{\boldsymbol{u}}_i^{(2)^T} \end{array}\right] \left[\begin{array}{c} -\boldsymbol{f}_i^{(1)} \\ -\boldsymbol{f}_b^{(1)} - \boldsymbol{f}_b^{(2)} + \boldsymbol{r}_b \\ -\boldsymbol{f}_i^{(2)} \end{array}\right] \tag{6.14}$$

Using Eq. (6.1), Eqs. (6.3) and Eq. (6.11), we finally obtain

$$\mathcal{E}(a) = Cst - a\boldsymbol{\delta}_I^T \boldsymbol{S}_{bb}^{(2)} \boldsymbol{\delta}_I + \frac{1}{2} a^2 \boldsymbol{\delta}_I^T (\boldsymbol{S}_{bb}^{(1)} + \boldsymbol{S}_{bb}^{(2)}) \boldsymbol{\delta}_I \tag{6.15}$$

where Cst is an expression that does not depend on a. Differentiating $\mathcal{E}$ with respect to a and enforcing the condition

$$\frac{d\mathcal{E}}{da} = -\boldsymbol{\delta}_I^T \boldsymbol{S}_{bb}^{(2)} \boldsymbol{\delta}_I + a\boldsymbol{\delta}_I^T (\boldsymbol{S}_{bb}^{(1)} + \boldsymbol{S}_{bb}^{(2)}) \boldsymbol{\delta}_I = 0 \tag{6.16}$$

gives

$$\boxed{\begin{aligned} a &= \frac{k^{(2)}}{k^{(1)} + k^{(2)}} && \text{(6.17a)} \\ k^{(1)} &= \boldsymbol{\delta}_I^T \boldsymbol{S}_{bb}^{(1)} \boldsymbol{\delta}_I = (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)})^T \boldsymbol{S}_{bb}^{(1)} (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)}) && \text{(6.17b)} \\ k^{(2)} &= \boldsymbol{\delta}_I^T \boldsymbol{S}_{bb}^{(2)} \boldsymbol{\delta}_I = (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)})^T \boldsymbol{S}_{bb}^{(2)} (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)}) && \text{(6.17c)} \end{aligned}}$$

We deduce from (6.17) that the smoothing parameter for the two subdomain case is such that $0 \leq a \leq 1$. From a physical viewpoint, the generalized smoothing procedure described by Eqs. (6.3) consists in treating the two substructures as two linear springs in series, computing the displacement jump at their point of connection, and redistributing this jump among both springs according to their "*global interface stiffnesses*" $k^{(1)}$ and $k^{(2)}$. Such a procedure was previously suggested in [67] within the context of the FETI (Finite Element Tearing and Interconnecting) method [79, 82], and its preconditioned conjugate gradient solver. However, Eqs. (6.17) give for the first time rational values of the substructure spring constants $k^{(1)}$ and $k^{(2)}$ that clearly contain the effect of material properties and mesh resolution. These constants can be described as the *Schur-complement norms* of the displacement jump at the substructure interfaces.

Remarks

- The Schur-complement matrices $\boldsymbol{S}_{bb}^{(s)}$ do not need to be explicitly computed. Only their action on the displacement jump vector should be evaluated [67].
- If the two substructures and their finite element models are identical, from Eqs. (6.17) it follows that $k^{(1)} = k^{(2)}$ and $a = 1/2$, which shows that the generalized smoothing procedure defined by Eqs. (6.3,6.17) includes the averaging scheme (5.23) as a particular case.
- As stated before, a is chosen here to be constant along the edge Γ_I. However, one could also discretize a in some nontrivial polynomial space, or ultimately, introduce a different smoothing parameter a for each interface d.o.f. Such alternative strategies can only be more accurate but are surely computationally more expensive than the present one.
- When the stiffness of the interface problem varies significantly from one direction to another, we recommend using different smoothing parameters for the different directional components of the interface displacement field.
- The convergence of the weakly compatible solution of the reduced hybrid system towards the reference solution when the subspace of the interface Lagrange multipliers is enriched is monotonic since the reduced method results from the saddle point principle (5.10). When a smoothing process is applied a posteriori onto the weakly compatible solution, the combination of the two-field hybrid method and the smoothing technique can be seen globally as a kinematically admissible approach. However, the smoothed solution does not derive from a simultaneous variational principle on the two-field hybrid unknowns and on the smoothing parameter and therefore the monotonic convergence of the smoothed solution can not be stated. Nevertheless, for practical applications, good convergence of the smoothed solution is observed and, most importantly, the quality of the smoothed solution measured by the energy norm of the displacement error was found to be much better than for the non-smoothed solution.

Finally, it is worthwhile noting that

$$\begin{aligned} \boldsymbol{\delta}_I^T \boldsymbol{r}_b &= \boldsymbol{\delta}_I^T (a\boldsymbol{S}_{bb}^{(1)} + (a-1)\boldsymbol{S}_{bb}^{(2)})\boldsymbol{\delta}_I \\ &= \boldsymbol{\delta}_I^T \frac{k^{(2)}\boldsymbol{S}_{bb}^{(1)} - k^{(1)}\boldsymbol{S}_{bb}^{(2)}}{k^{(1)} + k^{(2)}}\boldsymbol{\delta}_I \\ &= \frac{k^{(2)}k^{(1)} - k^{(1)}k^{(2)}}{k^{(1)} + k^{(2)}} \\ &= 0 \end{aligned} \tag{6.18}$$

which stresses the fact that

- the smoothing procedure described in Eqs. (6.3) is equivalent to a Rayleigh-Ritz method where the displacement shape functions are governed by the parameter a.
- the dual quantities (residual interface loads introduced by the proposed smoothing procedure and displacement corrections) do not produce work.

6.3 Accuracy improvement

6.3.1 Static analysis

In order to illustrate the beneficial effects of the proposed smoothing procedure, we consider first the heterogeneous two-substructure problem of the unsymmetric beam discussed in section 5.6 Fig. 5.1, apply the two-field hybrid method, and post-process the computed substructure displacements using formulae (6.3,6.17). The obtained results are summarized in Fig. 6.1 and Fig.

6.2 for $\alpha = E_2/E_1 = 1.0, 0.4, 0.2$ and $p = 2, 3$. The corresponding smoothing parameters are given in the figures. The relative energy norm of the displacement error before and after smoothing are listed in Table 6.1 (see the definitions (5.32-5.33)).

pol.order	$p=2$			$p=3$		
E_2/E_1	$\alpha=1.0$	$\alpha=0.4$	$\alpha=0.2$	$\alpha=1.0$	$\alpha=0.4$	$\alpha=0.2$
ε	0.13	0.091	$66\ 10^{-3}$	$7.1\ 10^{-2}$	$48\ 10^{-3}$	$36\ 10^{-3}$
$\hat{\varepsilon}$ (smoothed)	0.045	0.027	$1.5\ 10^{-3}$	$1.2\ 10^{-2}$	$7.1\ 10^{-3}$	$4.3\ 10^{-3}$

Table 6.1: Energy norm of solution error (heterogeneous unsymmetric beam)

Clearly, the results reported in Fig. 6.1 and Fig. 6.2 show that, unlike the original results obtained by the two-field hybrid method (Fig. 5.9 and Fig. 5.10), the smoothed horizontal and vertical interface displacement fields are in perfect agreement with the exact solution, which demonstrates the importance of the smoothing procedure (6.3) for heterogeneous substructure problems.

Moreover, it appears from Table 6.1 that, for a given polynomial order p of the interface stress distribution, the more heterogeneous the substructure problem is, the more accurate the smoothed two-field hybrid solution is.

As next example, we apply the smoothing procedure to the weakly compatible displacements obtained for the truss frame of Fig. 5.11. In this structure, the horizontal and vertical global stiffnesses are different, mainly because of the horizontal bars. Hence we define two independent smoothing variables a_x and a_y associated respectively with the horizontal and vertical displacements. We consider polynomial interface stress distributions of order $p = 1, 2, 3, 4$. The relative energy norm of the displacement error is given in Table 6.2.

pol.order	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
ε (non-smoothed)	0.83	0.75	0.73	0.68	0.67
$\hat{\varepsilon}$ (smoothed)	0.27	0.18	0.14	0.13	0.14

Table 6.2: Smoothing applied to the truss frame

The results show that

- the convergence of the reduced hybrid solution is very slow. This issue was already discussed in section 5.6 where we showed that the interface stress distribution was irregular and could therefore not be well approximated by a polynomial representation.
- the smoothing technique significantly improves the quality of the solution. Indeed, the relative energy norm of the error is reduced by three and the accuracy reached by the smoothed solution can not be obtained without smoothing for any polynomial order. Observing the smoothed and non-smoothed interface displacement when $p = 3$ for instance (Fig. 6.3), we see that the smoothing technique can efficiently improve the weakly compatible solution in giving more weight to the solution on the stiffest side.

6.3.2 Dynamic analysis

For modal analysis, the smoothing procedure is applied to every iterate of the inverse iteration. Thus for heterogeneous structures, the averaging steps (5.29) are replaced by the smoothing procedure (6.3).

Let us consider again the unsymmetric beam example in Fig. 5.1 where the Young modulus of the right subdomain is multiplied by $\alpha = 0.2$. We then compute the first 12 eigenfrequencies with

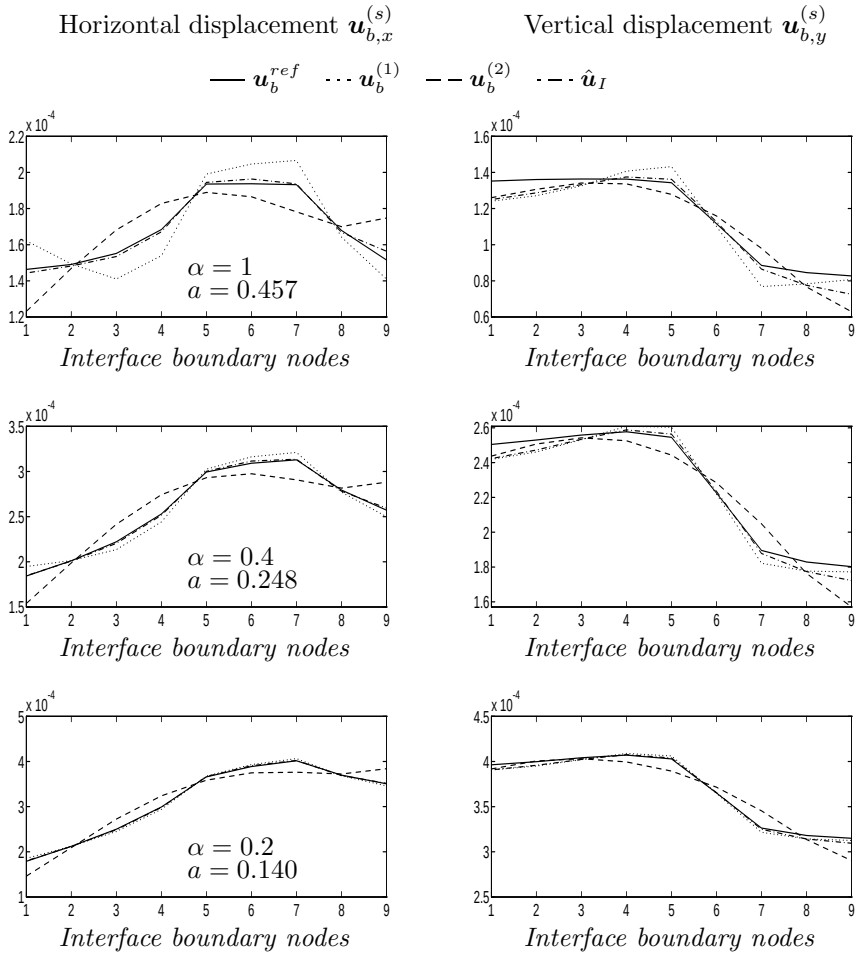


Figure 6.1: Smoothing results for the heterogeneous unsymmetric beam ($p = 2$)

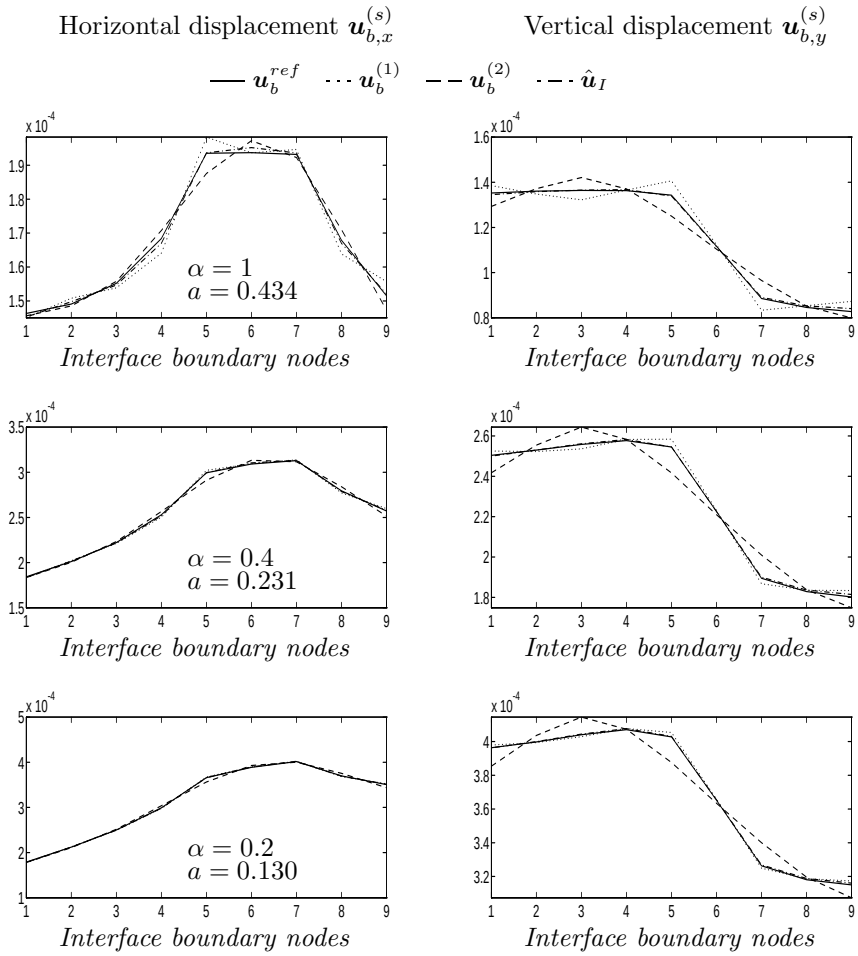


Figure 6.2: Smoothing results for the heterogeneous unsymmetric beam ($p = 3$)

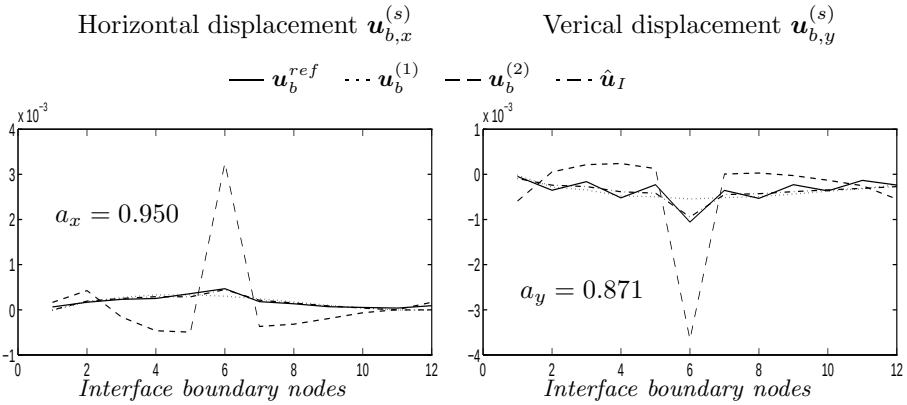


Figure 6.3: Smoothing of the truss displacements ($p = 3$)

a polynomial Lagrange multiplier of order $p = 1$. The results obtained with a subspace iteration algorithm are plotted in Fig. 6.4 when no smoothing is applied and then when the iterates are smoothed. The computed eigenfrequencies are compared to the reference solution and the relative eigenfrequency errors are also shown.

Fig. 6.4 shows that the smoothing procedure applied to the eigensolutions enhances the accuracy of the computed eigensolutions of the heterogeneous problem: the relative error on the eigenfrequencies is reduced by one order of magnitude. We can thus state that the smoothing procedure is as optimum for heterogeneous problem as averaging for homogeneous problems.

We then apply the smoothing technique to the time integration of the dynamic problem exposed in section 5.5.3, Fig. 5.7, where we assume that the right subdomain has its Young modulus multiplied by 0.2, i.e. $\alpha = E_2/E_1 = 0.2$. In order for the dynamic response to be comparable to the dynamic response of the homogeneous case (Fig. 5.8), we assume again that a step load is applied during a time interval of $T_8/3$, where T_8 is now the period of the eighth eigenmode of the heterogeneous problem and the time step is equal to $T_8/21$. Let us remind that this problem is a difficult one since the frequency content of the loading is such that it excites eigenmodes that are very badly approximated by the reduced two-field formulation.

First we show in Fig. 6.5 the transient solution computed when no smoothing is applied. It clearly points out that only the low frequency content of the transient response is correctly approximated while its high spectrum content is very much altered due to the high residual interface flexibility introduced by the weak compatibility.

Fig. 6.5 then shows the dynamic response obtained for a Lagrange polynomial order $p = 1$ when smoothing the transient solution at each time step. We also plot the acceleration curve to show the high level of accuracy achieved by the reduced integration algorithm with smoothing for a polynomial order as low as 1, that is when only 4 interface unknowns are defined.

The difference between the reference solution and the smoothed transient response is barely noticeable on the displacement and acceleration curves. In Fig. 6.6, we also give the value of the averaging parameter a. The value of the averaging parameter is optimized at every time step and thus varies according to the interface compatibility error after every integration step.

For transient response analysis, the smaller the time step, the smaller the compatibility error on the interface after a time step and thus the smaller the interface correction for smoothing the solution. To illustrate this point, let us reconsider the heterogeneous dynamic problem here above and integrate the transient response with a time-step multiplied by 7, i.e. $h = T_8/3$, which is the maximum practicle time-step one should use since it corresponds to the interval during which the step load is applied. The resulting displacement curve is shown in Fig. 6.7 when $p = 1$ and when smoothing is applied. The results indicate that the error between the reduced hybrid solution with smoothing and the reference solution is now more significant than for the smaller time-step. As predicted, the corrections brought to the transient response by the smoothing steps are more important. However, the overall solution remains very close to the reference solution.

We conclude that the smoothing technique is highly efficient also for dynamic analysis: it dramatically improves the accuracy of the solution computed by a reduced two-field hybrid method and therefore makes it possible to use a low number of interface unknowns while keeping good accuracy. Although the size of the time-step affects the precision of the smoothing post-processing, the solution remains of good quality for time step values commonly used in implicit time-integration.

6.4 A low cost alternative for repeated problems

When smoothing is required repetitively as in transient problems or in inverse iteration based eigensolvers, it becomes interesting to consider a "cheaper" variant of procedure (6.3) where the effect of the interface smoothing is not back-propagated to the internal d.o.f. After applying the compatibility correction to the sole interface (Eq. (6.3a)) the internal nodes are assumed to be

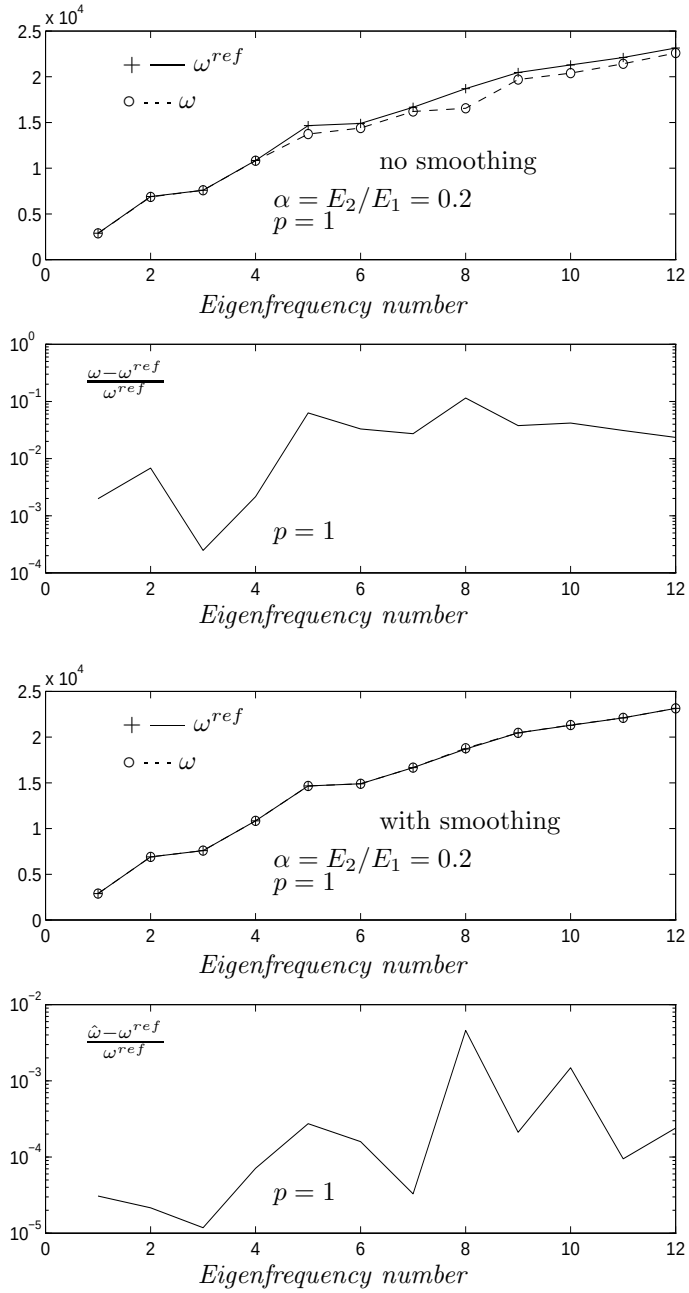


Figure 6.4: Eigenfrequencies of the heterogeneous unsymmetric beam ($p = 1, \alpha = 0.2$)

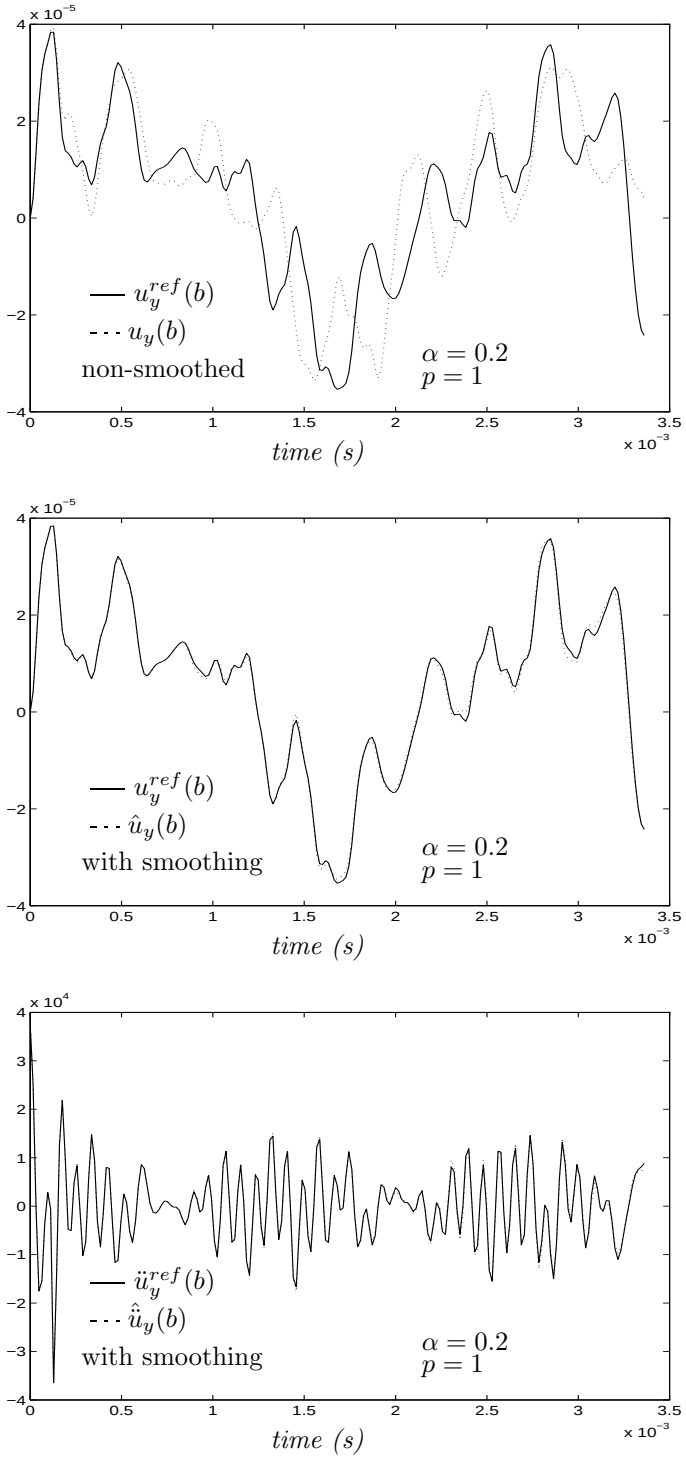


Figure 6.5: Transient response of the heterogeneous unsymmetric beam ($p = 1, \alpha = 0.2$)

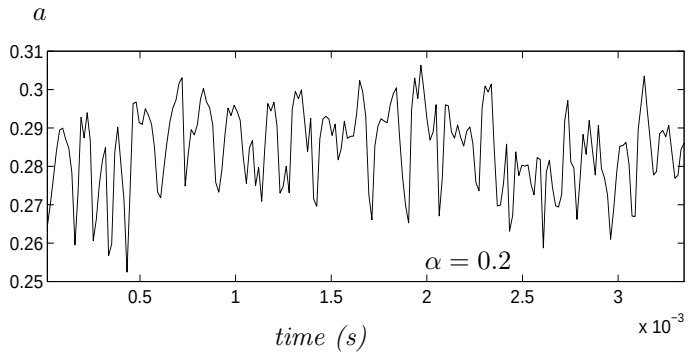


Figure 6.6: Smoothing parameter of the transient response ($p = 1, \alpha = 0.2$)

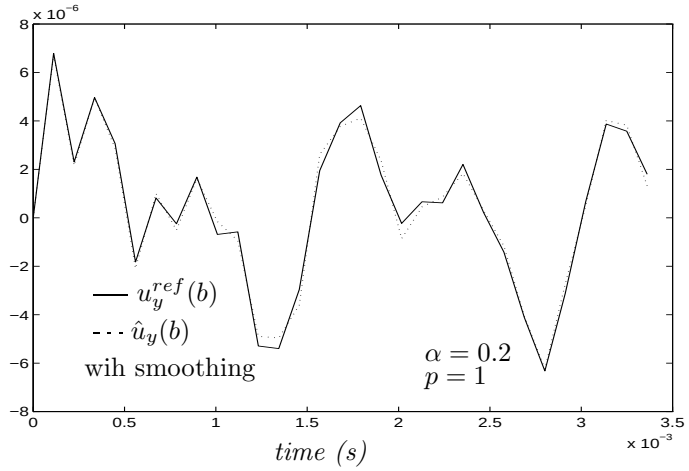


Figure 6.7: Transient response of the heterogeneous unsymmetric beam with large time step

fixed, namely Eq. (6.3b) is replaced by

$$\hat{\boldsymbol{u}}_i^{(s)} = \boldsymbol{u}_i^{(s)} \tag{6.19}$$

Hence the interface corrections are not back-propagated to the subdomain interiors and a force residual appears on the interface *and* on the internal nodes directly connected to the interface nodes. The equilibrium equation (6.10) after smoothing now writes

$$\begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} & 0 \\ \boldsymbol{K}_{ib}^{(1)^T} & \boldsymbol{K}_{bb}^{(1)} + \boldsymbol{K}_{bb}^{(2)} & \boldsymbol{K}_{ib}^{(2)} \\ 0 & \boldsymbol{K}_{ib}^{(2)^T} & \boldsymbol{K}_{ii}^{(2)} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}}_i^{(1)} \\ \hat{\boldsymbol{u}}_I \\ \hat{\boldsymbol{u}}_i^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} + \boldsymbol{f}_b^{(2)} \\ \boldsymbol{f}_i^{(2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_i^{(1)} \\ \boldsymbol{r}_b \\ \boldsymbol{r}_i^{(2)} \end{bmatrix} \tag{6.20}$$

where

$$\boldsymbol{r}_b = \boldsymbol{K}_{bb}^{(1)} \Delta \boldsymbol{u}_b^{(1)} + \boldsymbol{K}_{bb}^{(2)} \Delta \boldsymbol{u}_b^{(2)} \tag{6.21}$$

$$\boldsymbol{r}_i^{(s)} = \boldsymbol{K}_{ib}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{6.22}$$

Following the same variational approach as in Section 6.2, such a strategy leads to a smoothing procedure similar to Eqs. (6.17) but where the Schur-complement matrices $\boldsymbol{S}_{bb}^{(s)}$ are replaced by the *lumped* interface stiffness matrices $\boldsymbol{K}_{bb}^{(s)}$ [67]. The following lumped averaging parameter are found

$$a_L = \frac{k_L^{(2)}}{k_L^{(1)} + k_L^{(2)}} \tag{6.23a}$$

$$k_L^{(1)} = \boldsymbol{\delta}_I^T \boldsymbol{K}_{bb}^{(1)} \boldsymbol{\delta}_I = (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)})^T \boldsymbol{K}_{bb}^{(1)} (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)}) \tag{6.23b}$$

$$k_L^{(2)} = \boldsymbol{\delta}_I^T \boldsymbol{K}_{bb}^{(2)} \boldsymbol{\delta}_I = (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)})^T \boldsymbol{K}_{bb}^{(2)} (\boldsymbol{u}_b^{(2)} - \boldsymbol{u}_b^{(1)}) \tag{6.23c}$$

The computational advantages of the above lumped strategy are obvious given that $\boldsymbol{K}_{bb}$ is readily available and only simple matrix-vector multiplications are needed to evaluate the averaging parameter a_L, whereas the solution (6.17) requires the solution of a Dirichlet problem for each subdomain. Moreover, it is our experience that for practical time-steps, smoothing with the lumped strategy (6.19–6.23) retains almost the same order of accuracy as smoothing with the less economical but complete procedure (6.17).

As a first example we will compute the first 12 eigenfrequencies of the unsymmetric heterogeneous beam (Fig. 5.1) already analyzed in the previous section with the complete smoothing procedure. Applying now the lumped smoothing to every iterate, the eigenfrequencies obtained by a subspace iteration algorithm for a polynomial degree $p = 1$ are as plotted in Fig. 6.8. Comparing those results with the eigenfrequencies obtained by the complete smoothing scheme (Fig. 6.4), we observe that the lumping of the smoothing steps slightly deteriorates the accuracy of the solution, although compared to the non-smoothed solution, the accuracy improvement with this cheap lumped variant remains significant.

We then apply the lumped smoothing procedure to the transient response computation of this heterogeneous unsymmetric beam. Introducing a lumped smoothing correction after each time-integration step, the transient response is very similar to the reference solution (Fig. 6.9) and comparing this result to the solution obtained with a complete smoothing method (Fig. 6.5), we conclude that although the lumped procedure introduces more force residuals in the system, the loss of accuracy for the transient response computation is small.

The fact that the precision of the solution is nearly the same when using a smoothing method with a complete Dirichlet solution and a lumped smoothing variant can be explained as follows. Although the time-step can be large as far as the stability of implicit time-stepping algorithms is

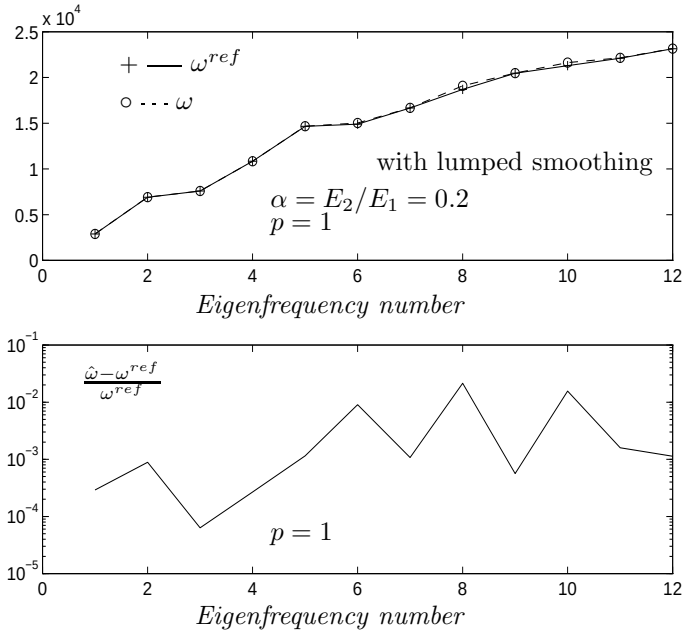


Figure 6.8: Eigenfrequencies of the unsymmetric heterogeneous beam (lumped smoothing)

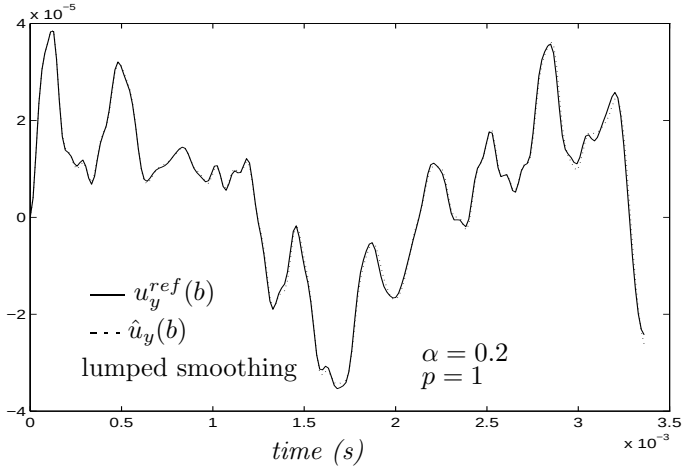


Figure 6.9: Transient response of the heterogeneous unsymmetric beam with lumped smoothing

concerned [91], the time step is generally small enough to accurately capture the participation of the low eigenspectrum of the response. This means that the time-stepping matrix of the classical Newmark scheme

$$\boldsymbol{D}^{(s)} = \boldsymbol{M}^{(s)} + \frac{h}{2}\boldsymbol{C}^{(s)} + \frac{h^2}{4}\boldsymbol{K}^{(s)} \tag{6.24}$$

is dominated by the mass matrix. If we then observe that in practice the mass matrix is diagonally dominant or lumped, we have that $\boldsymbol{D}^{(s)} \simeq \boldsymbol{M}^{(s)}_{bb}$ so that the lumped smoothing for this operator is similar to the complete method requiring the solution of a Dirichlet problem associated to $\boldsymbol{D}^{(s)}$. This justifies the use of the lumped strategy in most dynamic response applications.

In the next section, we further analyze the accuracy of the time response computed with an implicit algorithm and a smoothing procedure: the energy balance of the implicit time-integration scheme will be studied.

6.5 Energy balance of the time-integration scheme with smoothing

It is well-known that the Newmark scheme, i.e. the trapezoidal rule, preserves the total energy of linear systems, and as a consequence it corresponds to a stable integration scheme [91]. When smoothing is performed between two time-steps, the energy conservation property is somewhat destroyed as will be shown.

For the simplicity of the presentation, zero damping is assumed. We call $\mathcal{T}+\mathcal{V}$ the total energy of the system, i.e. the sum of the kinetic and potential energies. When using the trapezoidal time-integration rule, the variation of the total energy of a substructure $\Omega^{(s)}$ when advancing the solution from t_n to t_{n+1} is given by [91]

$$\begin{aligned}\left[\mathcal{T}^{(s)} + \mathcal{V}^{(s)}\right]_{t_n}^{t_{n+1}} &= \frac{1}{2}\Big(\dot{\boldsymbol{u}}^{(s)^T_{n+1}}\boldsymbol{M}^{(s)}\dot{\boldsymbol{u}}^{(s)_{n+1}} + \boldsymbol{u}^{(s)^T_{n+1}}\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)_{n+1}} \\ &\qquad -\dot{\boldsymbol{u}}^{(s)^T_{n}}\boldsymbol{M}^{(s)}\dot{\boldsymbol{u}}^{(s)_{n}} - \boldsymbol{u}^{(s)^T_{n}}\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)_{n}}\Big) \\ &= \left(\boldsymbol{u}^{(s)_{n+1}} - \boldsymbol{u}^{(s)_n}\right)^T \frac{\boldsymbol{f}^{(s)_n} - \boldsymbol{B}^{(s),p}\boldsymbol{\lambda}^n + \boldsymbol{f}^{(s)_{n+1}} - \boldsymbol{B}^{(s),p}\boldsymbol{\lambda}^{n+1}}{2}\end{aligned} \tag{6.25}$$

Summing up the domain-wise energy balance, and taking account of the compatibility relations satisfied by the solution of a two-field hybrid system, we find the energy balance for the complete structure

$$[\mathcal{T}+\mathcal{V}]_{t_n}^{t_{n+1}} = \sum_{s=1}^{N_s}\left(\boldsymbol{u}^{(s),n+1} - \boldsymbol{u}^{(s),n}\right)^T \frac{\boldsymbol{f}^{(s),n} + \boldsymbol{f}^{(s),n+1}}{2} \tag{6.26}$$

and we note that the interface Lagrange multipliers do not produce any work. Eq. (6.26) shows that the trapezoidal rule applied to a two-field hybrid system satisfies the energy conservation since the total energy variation is equal to the work produced by the external forces.

We now assume that weak compatibility constraints are enforced by a reduced set of Lagrange multipliers, and that a smoothing procedure is applied at each time-step to the solution of the two-field system. In order to evaluate the energy balance in this case, we note that the smoothed solution satisfying the equilibrium relations (6.10) or (6.20) can be understood as solutions of the reference hybrid system when the residual force $\boldsymbol{r}$ is applied to the system in addition to the external forces. Hence, considering that the residual force $\boldsymbol{r}$ inherent to the smoothing process acts as an external excitation, the energy balance for the smoothed solution writes

$$\left[\hat{\mathcal{T}} + \hat{\mathcal{V}}\right]_{t_n}^{t_{n+1}} = \frac{1}{2}\left(\begin{bmatrix}\hat{\boldsymbol{u}}_i^{(1)_{n+1}} \\ \hat{\boldsymbol{u}}_I^{n+1} \\ \hat{\boldsymbol{u}}_i^{(2)_{n+1}}\end{bmatrix} - \begin{bmatrix}\hat{\boldsymbol{u}}_i^{(1)_n} \\ \hat{\boldsymbol{u}}_I^{n} \\ \hat{\boldsymbol{u}}_i^{(2)_n}\end{bmatrix}\right)^T \left(\boldsymbol{f}^n + \boldsymbol{f}^{n+1} + \begin{bmatrix}\boldsymbol{r}_i^{(1)_n} + \boldsymbol{r}_i^{(1)_{n+1}} \\ \boldsymbol{r}_b^n + \boldsymbol{r}_b^{n+1} \\ \boldsymbol{r}_i^{(2)_n} + \boldsymbol{r}_i^{(2)_{n+1}}\end{bmatrix}\right) \tag{6.27}$$

where $\boldsymbol{r}_i^{(s)} = \mathbf{0}$ for every time-step when the complete smoothing scheme (6.3) is applied. We thus conclude that the residual force inherent to any smoothing post-processing introduces a spurious energy variation in the system which is equal to

$$\left[\hat{\mathcal{T}} + \hat{\mathcal{V}}\right]_{t_{n_{spurious}}}^{t_{n+1}} = \frac{1}{2}\left(\begin{bmatrix} \hat{\boldsymbol{u}}_i^{(1)_{n+1}} \\ \hat{\boldsymbol{u}}_I^{n+1} \\ \hat{\boldsymbol{u}}_i^{(2)_{n+1}} \end{bmatrix} - \begin{bmatrix} \hat{\boldsymbol{u}}_i^{(1)_n} \\ \hat{\boldsymbol{u}}_I^{n} \\ \hat{\boldsymbol{u}}_i^{(2)_n} \end{bmatrix}\right)^T \begin{bmatrix} \boldsymbol{r}_i^{(1)_n} + \boldsymbol{r}_i^{(1)_{n+1}} \\ \boldsymbol{r}_b^{n} + \boldsymbol{r}_b^{n+1} \\ \boldsymbol{r}_i^{(2)_n} + \boldsymbol{r}_i^{(2)_{n+1}} \end{bmatrix} \tag{6.28}$$

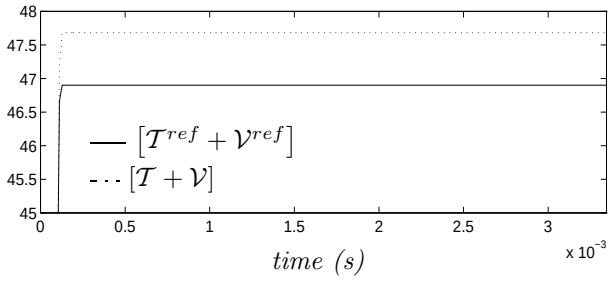
To illustrate this point let us consider once more the unsymmetric beam example when the Young modulus of the right-hand side subdomain is equal to 0.2 times the Young modulus of the left-hand side domain. The transient response of this structure has been computed in sections 6.3 and 6.4 for a polynomial Lagrange multiplier of order $p = 1$ and for a time step equal to $T_8/21$. The energy curves are given in Fig. 6.10 when no smoothing is applied, when the complete smoothing is used, and when the lumped smoothing is considered. The variation of the total energy is compared to the total energy of the non-substructured system.

From the result curves in Fig. 6.10 we make the following observations:

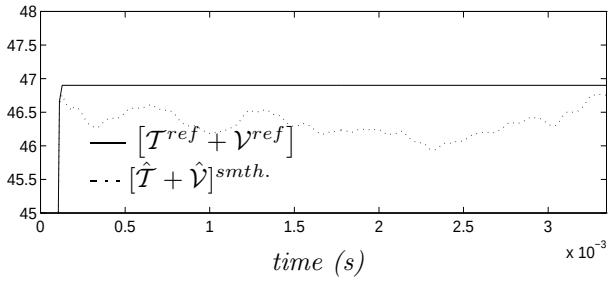
- The total energy of the reference solution remains constant after the applied loads vanish as predicted by the energy conservation property of the trapezoidal integration rule. The total energy associated with the non-smoothed solution (Fig. 6.10(a)) also remains constant as predicted by Eq. (6.26), but at a level 1.6 percent higher than the reference curve. This clearly indicates that the trapezoidal rule also conserves the total energy for the reduced interface formulation, but due to the weaker interface stiffness the energy accumulated by the structure is overestimated.

- The total energy of the smoothed solution fluctuates slightly but does not cause any instability (Fig. 6.10(b)–6.10(c)). The energy variation is due to the spurious energy produced by the residual forces at every time step and this spurious energy for the complete smoothing and for the lumped variant are of similar order of magnitude.

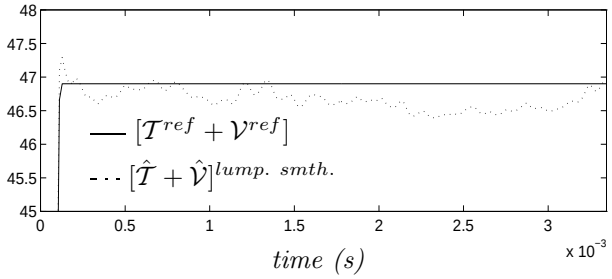
In practice, we recommend that the spurious energy (6.28) be monitored and the user should increase the number of Lagrangian variables whenever the accumulation of spurious energy becomes significant compared to the total energy of the structure. Another possible remedy to the accumulation of spurious energy is to introduce numerical damping by replacing the trapezoidal rule by a generalized-α method [32]: it damps out the high frequency fluctuations due to smoothing without affecting significantly the main spectrum of the response.



(a) No smoothing



(b) Complete smoothing



(c) Lumped smoothing

Figure 6.10: Total energy of the transient response of the heterogeneous unsymmetric beam ($p = 1$, $\alpha = 0.2$)

Chapter 7

General smoothing procedure for multiple conforming and non-conforming substructures

When the given substructures have non-matching discrete interfaces, the issue of strong (discrete) continuity on Γ_I does not arise because in that case the substructure displacement fields are not necessarily computed on the same physical interface nodes. However, we will show that even in that case, smoothing is beneficial and can be interpreted as a means of enforcing a posteriori an interface continuity that is stronger than that resulting from the solutions generated by the two-field hybrid method with a small number of interface unknowns (such as when low order Lagrange multipliers are used).

For example, if the Lagrange multipliers needed for enforcing the continuity constraints across a certain interface are approximated with a p-th order polynomial, the substructure displacement fields computed by the two-field method can be post-processed to satisfy

$$\int_{\Gamma_I^{sr}} \lambda^{sr,q}(v^{(s)} - v^{(r)})\, d\Gamma \;=\; 0 \tag{7.1}$$

where $\lambda^{sr,q}$ is the Lagrangian multiplier field approximated by a q-th order polynomial with $q > p$. In the sequel, we refer to this post-processing paradigm as a n_{λ^q}-th order smoothing procedure, n_{λ^q} referring to the number of unknowns defined for the reference solution.

7.1 Conventions and mathematical preliminaries

As before $\boldsymbol{b}^{(s)}$ denotes the restriction of a constraint matrix $\boldsymbol{B}^{(s)}$ to the interface boundary $\Gamma_I^{(s)} = \partial\Omega^{(s)} \backslash \partial\Omega$ of a given substructure $\Omega^{(s)}$. Taking once more the convention that the boundary d.o.f. are numbered last, we have

$$\boldsymbol{B}^{(s)} = \left[\begin{array}{cc} \boldsymbol{0} & \boldsymbol{b}^{(s)} \end{array} \right] \tag{7.2}$$

The interface boundary of each substructure can be broken into *edges*:

- An edge is defined as a collection of neighboring interface nodes.
- Each substructure interface node is assigned to one and only one substructure edge.

Note that this definition is independent from the dimensionality of the interface (Fig. 7.1). Consequently, the traditional crosspoint that is often referred to in the domain decomposition

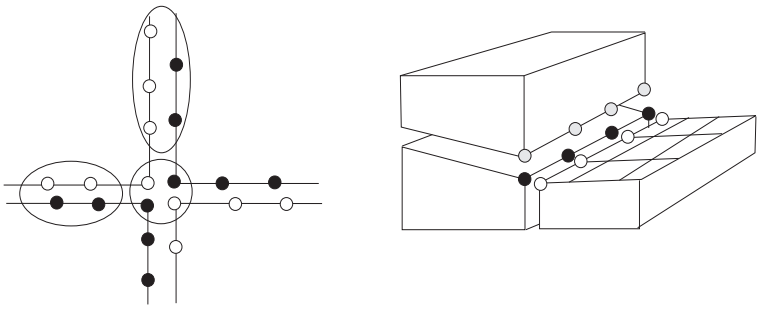


Figure 7.1: Edges and cross-points for two- and three-dimensional substructure problems

literature as a "vertex" for two-dimensional problems is treated here as an edge with a single interface node (Fig. 7.1). For three-dimensional problems, the crosspoint is an edge with an arbitrary number of interface nodes. It follows that $\boldsymbol{b}^{(s)}$ can be partitioned as

$$\boldsymbol{b}^{(s)} = \left[\begin{array}{cccc} \boldsymbol{b}^{(s),i} & \boldsymbol{b}^{(s),j} & \cdots & \boldsymbol{b}^{(s),l} \end{array} \right] \tag{7.3}$$

where $\boldsymbol{b}^{(s),j}$ is the restriction of $\boldsymbol{b}^{(s)}$ to the j-th edge of $\Gamma_I^{(s)}$. Usually, the continuity constraints across an edge j are enforced independently from those across an edge $j \neq l$, and therefore one has

$$\boldsymbol{b}^{(s),j^T} \boldsymbol{b}^{(s),l} = \boldsymbol{0} \qquad j \neq l \tag{7.4}$$

Moreover, an edge j can be shared by multiple substructures, and therefore $\boldsymbol{b}^{(s),j}$ can also be partitioned as

$$\boldsymbol{b}^{(s),j} = \left[\begin{array}{c} \boldsymbol{0} \\ \vdots \\ \boldsymbol{b}^{(s,q),j} \\ \boldsymbol{0} \\ \boldsymbol{b}^{(s,r),j} \\ \vdots \\ \boldsymbol{0} \end{array} \right] \tag{7.5}$$

where $\boldsymbol{b}^{(s,q),j}$ is the component of $\boldsymbol{b}^{(s),j}$ pertaining to the interconnectivity of substructures $\Omega^{(s)}$ and $\Omega^{(q)}$ along edge j. Hence the support of $\boldsymbol{b}^{(s,q),j}$ is the interface portion $\Gamma_I^{sq,j}$ defined between $\Omega^{(s)}$ and $\Omega^{(q)}$ and restricted to the edge j.

Next, we define the *multiplicity* m_i of an edge i as *the number of substructures it interconnects*. Hence, $m_i \geq 2$, and for a crosspoint $m_i > 2$. In particular, we note that for conforming interfaces, the following holds

$$\boldsymbol{b}^{(s,q),j^T} \boldsymbol{b}^{(s,q),j} = \boldsymbol{I} = \boldsymbol{b}^{(s,q),j} \boldsymbol{b}^{(s,q),j^T} \tag{7.6}$$

$$\boldsymbol{b}^{(s),j^T} \boldsymbol{b}^{(s),j} = (m_i - 1)\boldsymbol{I} \tag{7.7}$$

where $\boldsymbol{I}$ is the identity matrix of dimension $n_b^{(s),j} \times n_b^{(s),j}$, $n_b^{(s),j}$ being the number of interface d.o.f. on edge j belonging to $\Omega^{(s)}$. The first equation (7.6) states that for a conforming interface, $\boldsymbol{b}^{(s,q),j}$ defines a bi-univocal relation between the interface d.o.f. and the nodal Lagrange multipliers that connect $\Omega^{(s)}$ and $\Omega^{(q)}$ on edge j. The second relation (7.7) specifies that if we consider a conforming edge $\Gamma_I^{(s),j}$, the number of Lagrange multipliers defined for each interface d.o.f. is equal to the number of subdomains connecting on $\Gamma_I^{(s),j}$ minus one. Obviously, for this statement to be valid we assume that Lagrange multipliers are explicitly defined between all the subdomains

connecting on Γ_I^{sq}. As discussed earlier in section 2.2, the compatibility constraints on a cross point ($m_i > 2$) are redundant, but this redundancy will prove to be crucial for defining a consistent smoothing procedure.

Finally, we introduce the unsigned equivalents of $\boldsymbol{B}^{(s)}$, $\boldsymbol{b}^{(s)}$, $\boldsymbol{b}^{(s),j}$ and $\boldsymbol{b}^{(s,q),j}$, and designate them with a tilde superscript. For example, we have

$$\tilde{\boldsymbol{b}}^{(s,q),j} = \int_{\Gamma^{sq,j}} \boldsymbol{\Lambda}^{sq,j^T} \boldsymbol{N}^{(s)} d\Gamma \tag{7.8}$$

where $\boldsymbol{\Lambda}^{sq,j^T}$ are the stress distribution functions between $\Omega^{(s)}$ and $\Omega^{(r)}$ on edge j.

We assume that along an edge j, the same approximation is used for the Lagrange multipliers interconnecting any pair of substructures $\Omega^{(s)}$, $\Omega^{(r)}$ and $\Omega^{(q)}$, i.e.

$$\boldsymbol{\Lambda}^{sr,j}(\eta) = \boldsymbol{\Lambda}^{sq,j}(\eta) = \boldsymbol{\Lambda}^{rq,j}(\eta) \qquad \forall s,r,q : \Gamma_I^{(s)} \ni j, \Gamma_I^{(r)} \ni j, \Gamma_I^{(q)} \ni j \tag{7.9}$$

To emphasize this point, we use in the sequel the notation $\boldsymbol{\Lambda}^j$ and $\boldsymbol{b}^{(s,-),j}$ rather than $\boldsymbol{\Lambda}^{sq,j}$ and $\boldsymbol{b}^{(s,q),j}$.

Using the above definitions and notation, it follows that the compatibility equations between the substructure interfaces can be written in the general case as

$$\tilde{\boldsymbol{b}}^{(s,-),j} \boldsymbol{u}_b^{(s),j} - \tilde{\boldsymbol{b}}^{(r,-),j} \boldsymbol{u}_b^{(r),j} = \boldsymbol{0} \qquad \forall j,\ \forall \Gamma_I^{sr} \ni j \tag{7.10}$$

where $\boldsymbol{u}_b^{(s),j}$ is the trace of the displacement field on the substructure edge $\Gamma_I^{(s),j}$.

In order to satisfy at least a necessary condition for the Babuška-Brezzi condition, we assume that the number of interface unknowns defining the Lagrange multiplier interconnecting two substructures $\Omega^{(s)}$ and $\Omega^{(q)}$ is not superior to the local size of the coarsest discretization of the interface Γ_I^{sq}. Hence $\tilde{\boldsymbol{b}}^{(s,-),j}$ is a square matrix or a rectangular matrix with fewer rows than columns. Let $\tilde{\boldsymbol{b}}^{*^{(s,-),j}}$ denote its null space

$$\tilde{\boldsymbol{b}}^{(s,-),j}\ \tilde{\boldsymbol{b}}^{*^{(s,-),j}} = \boldsymbol{0} \tag{7.11}$$

Assuming that the approximation functions $\boldsymbol{\Lambda}^j$ are independent, the matrix product $[\tilde{\boldsymbol{b}}^{(s,-),j}\ \tilde{\boldsymbol{b}}^{(s,-),j^T}]$ is non-singular. Hence, any vector $\boldsymbol{x}^{(s),j}$ relative to the boundary of $\Omega^{(s)}$ and to edge j can be decomposed into a component residing in the null space of $\tilde{\boldsymbol{b}}^{(s,-),j}$, and a component belonging to the range of its transpose

$$\boldsymbol{x}^{(s),j} = \boldsymbol{B}^{(s,-),j} \boldsymbol{y} + \tilde{\boldsymbol{b}}^{*^{(s,-),j}} \boldsymbol{z} \tag{7.12}$$

where

$$\boldsymbol{B}^{(s,-),j} = \tilde{\boldsymbol{b}}^{(s,-),j^T} \left[\tilde{\boldsymbol{b}}^{(s,-),j}\ \tilde{\boldsymbol{b}}^{(s,-),j^T} \right]^{-1} \tag{7.13}$$

and where the amplitudes of the components are given by

$$\boldsymbol{y} = \tilde{\boldsymbol{b}}^{(s,-),j} \boldsymbol{x}^{(s),j} \tag{7.14}$$

$$\boldsymbol{z} = \left[\tilde{\boldsymbol{b}}^{*^{(s,-),j^T}}\ \tilde{\boldsymbol{b}}^{*^{(s,-),j}} \right]^{-1} \tilde{\boldsymbol{b}}^{*^{(s,-),j^T}} \boldsymbol{x}^{(s),j} \tag{7.15}$$

From a mechanical view point, the null space $\tilde{\boldsymbol{b}}^{*^{(s,-),j}}$ represents the interface displacement modes which are not constrained by the compatibility requirements. Clearly, for the exact compatibility constraints on conforming interfaces, $\tilde{\boldsymbol{b}}^{(s,-),j}$ is square and non singular, so that the null space is empty and for the associated Boolean constraint matrices we have $\tilde{\boldsymbol{b}}^{(s,-),j^T}\ \tilde{\boldsymbol{b}}^{(s,-),j} = \boldsymbol{I}$ (see Eq. (7.6)). The decomposition (7.12) then becomes

$$\boldsymbol{x}^{(s),j} = \tilde{\boldsymbol{b}}^{(s,-),j^T} \boldsymbol{y} = \tilde{\boldsymbol{b}}^{(s,-),j^T} \tilde{\boldsymbol{b}}^{(s,-),j} \boldsymbol{x}^{(s),j} \tag{7.16}$$

The vector $\boldsymbol{y}$ in Eqs. (7.12) and (7.16) represents the projection of $\boldsymbol{x}^{(s),j}$ onto the dual space of the constraints functions $\boldsymbol{\Lambda}^j$.

For non-conforming interfaces, the interface compatibility is not fully enforced so that interface displacements modes $\tilde{\boldsymbol{b}}^{*^{(s,-),j}}$ that are not restrained by the compatibility requirements exist and contribute to the spurious interface flexibility discussed in section 5.1. The vector $\boldsymbol{z}$ in Eq. (7.12) then represents the amplitudes of those non-restrained interface modes.

7.2 The generalized smoothing procedure

If the Lagrange multipliers needed for enforcing the continuity constraints across an edge $\Gamma_I^{sr,j}$ are approximated with n_{λ^p} distribution functions, the substructure displacement fields computed using the reduced two-field method (5.2a-5.2b-5.5) will satisfy

$$\tilde{\boldsymbol{b}}_p^{(s,-),j}\ \boldsymbol{u}_b^{(s),j} - \tilde{\boldsymbol{b}}_p^{(r,-),j}\ \boldsymbol{u}_b^{(r),j} = \boldsymbol{0} \tag{7.17}$$

In Eq. (7.17) above, the subscript p is used to emphasize the dependence of $\boldsymbol{b}^{(s,-),j}$ and $\boldsymbol{b}^{(r,-),j}$ on the n_{λ^p}-th order approximation. If $\boldsymbol{b}_q^{(s,-),j}$ is the constraint matrix associated with a n_{λ^q}-th order approximation of the Lagrange multipliers and $n_{\lambda^q} > n_{\lambda^p}$, the substructure displacement fields computed by using a n_{λ^p}-th order approximation of the Lagrange multipliers will satisfy

$$\tilde{\boldsymbol{b}}_q^{(s,-),j}\ \boldsymbol{u}_b^{(s),j} - \tilde{\boldsymbol{b}}_q^{(r,-),j}\ \boldsymbol{u}_b^{(r),j} = \boldsymbol{r}_q^{(s,r),j}\ \neq\ \boldsymbol{0} \tag{7.18}$$

where $\boldsymbol{r}_q^{(s,r),j}$ can be interpreted as a compatibility residual along the edge $\Gamma_I^{sr,j}$. It becomes therefore clear that for substructure problems with non-matching interfaces, the objective of a n_{λ^q}-th order generalized smoothing procedure (see the introduction of this Chapter) is to drive the compatibility residual $\boldsymbol{r}_q^{(s,r),j}$ to zero — that is, to compute substructure displacement fields $\hat{\boldsymbol{u}}_b^{(s),j}$ and $\hat{\boldsymbol{u}}_b^{(r),j}$ that are compatible with respect to a higher-order weighting matrix than the one used for enforcing equilibrium (or with respect to the boolean matrix in the case of compatible interfaces)

$$\tilde{\boldsymbol{b}}_q^{(s,-),j}\ \hat{\boldsymbol{u}}_b^{(s),j} - \tilde{\boldsymbol{b}}_q^{(r,-),j}\ \hat{\boldsymbol{u}}_b^{(r),j} = \boldsymbol{0} \tag{7.19}$$

In the sequel, we drop the p and q subscripts for simplicity.

First, we note that because the same approximation is used for the Lagrange multipliers interconnecting any pair of substructures $\Omega^{(s)}$ and $\Omega^{(q)}$ (see Eq. (7.9)), the subvector of Lagrange multipliers associated with an edge j has the same size in all the substructures where it is evaluated. Therefore, one can construct the edge field ν^j defined as a weighted sum of the edge boundary displacements projected on the Lagrange multiplier space:

$$\boldsymbol{\nu}^j = \sum_{s:\Gamma_I^{(s)}\ni j} \beta^{(s),j} \tilde{\boldsymbol{b}}^{(s,-),j}\ \boldsymbol{u}_b^{(s),j} \tag{7.20}$$

where $\{s : \Gamma_I^{(s)} \ni j\}$ stands for all subdomains s having an interface on edge j. The real coefficients $\beta^{(s),j}$ will play the same role as the averaging parameter a introduced in Eqs. (6.3a). Clearly, for any combination of $\beta^{(s),j}$, ν^j is not specific to any substructure connected to edge j. Hence, it is a prime candidate for constructing a smoothed substructure displacement field $\hat{\boldsymbol{u}}_b^{(s),j}$ that verifies the stronger compatibility equations (7.19). We observe that the reference compatibility constraints (7.19) states that the projections of the interface displacements in the dual space of the Lagrange multipliers are identical. We declare that this unique dual projection is $\boldsymbol{\nu}^j$, namely we impose that the component of $\hat{\boldsymbol{u}}_b^{(s),j}$ residing in the range of $\tilde{\boldsymbol{b}}^{(s,-),j^T}$ be equal to $\boldsymbol{\nu}^j$, and that the one belonging to $\tilde{b}^{*^{(s,-),j}}$ be the same as that obtained from the decomposition (7.12) of $\boldsymbol{u}_b^{(s),j}$:

$$\hat{\boldsymbol{u}}_b^{(s),j} \quad = \quad \boldsymbol{B}^{(s,-),j}\nu^j + \tilde{\boldsymbol{b}}^{*^{(s,-),j}}\ \boldsymbol{\mu}^j \tag{7.21}$$

$$\boldsymbol{\mu}^j = \left[\tilde{\boldsymbol{b}}^{*(s,-),j^T}\ \tilde{\boldsymbol{b}}^{*(s,-),j}\right]^{-1}\ \tilde{\boldsymbol{b}}^{*(s,-),j^T}\ \boldsymbol{u}_b^{(s),j} \tag{7.22}$$

Using Eq. (7.11), the reader can easily verify that the proposed smoothed substructure displacement $\hat{\boldsymbol{u}}_b^{(s),j}$ verifies the objective compatibility equations (7.19). Finally, from Eqs. (7.20–7.21) and the principle of back-propagation introduced in Section 6.1, it follows that the generalized smoothing procedure can be written for an arbitrary substructure problem as

$$\hat{\boldsymbol{u}}_b^{(s),j} = \sum_{r:\Gamma_I^{(r)}\ni j} \beta^{(r),j}\ \boldsymbol{B}^{(s,-),j}\tilde{\boldsymbol{b}}^{(r,-),j}\boldsymbol{u}_b^{(r),j} + \tilde{\boldsymbol{b}}^{*(s,-),j}[\tilde{\boldsymbol{b}}^{*(s,-),j^T}\ \tilde{\boldsymbol{b}}^{*(s,-),j}]^{-1}\ \tilde{\boldsymbol{b}}^{*(s,-),j^T}\ \boldsymbol{u}_b^{(s),j} \tag{7.23a}$$

$$\hat{\boldsymbol{u}}_i^{(s)} = \boldsymbol{K}_{ii}^{(s)^{-1}}\left(\boldsymbol{f}_i^{(s)} - \boldsymbol{K}_{ib}^{(s)}\hat{\boldsymbol{u}}_b^{(s)}\right) \tag{7.23b}$$

It remains to find the optimal values of the coefficients $\beta^{(s),j}$. For this purpose, we follow conceptually the Rayleigh-Ritz approach that was presented in Section 6.2.

7.3 Energy minimization

The coefficients $\beta^{(s),j}$ in (7.23) are constrained to satisfy the consistency condition

$$\sum_{s:\Gamma_I^{(s)}\ni j} \beta^{(s),j} = 1 \tag{7.24}$$

so that if the solution $\boldsymbol{u}^{(s)}$ of the reduced two-field hybrid system already satisfies the reference compatibility, the smoothed solution remains identical to the non-smoothed result. Indeed, let us assume that $\boldsymbol{u}^{(s)}$ verifies the objective compatibility (7.19). The smoothed interface solution (7.23a) then writes:

$$\hat{\boldsymbol{u}}_b^{(s),j} = \sum_{r:\Gamma_I^{(r)}\ni j} \beta^{(r),j}\ \boldsymbol{B}^{(s,-),j}\tilde{\boldsymbol{b}}^{(r,-),j}\boldsymbol{u}_b^{(r),j} + \tilde{\boldsymbol{b}}^{*(s,-),j}\boldsymbol{\mu}^j \tag{7.25}$$

$$= \sum_{r:\Gamma_I^{(r)}\ni j} \beta^{(r),j}\ \boldsymbol{B}^{(s,-),j}\tilde{\boldsymbol{b}}^{(s,-),j}\boldsymbol{u}_b^{(s),j} + \tilde{\boldsymbol{b}}^{*(s,-),j}\boldsymbol{\mu}^j \tag{7.26}$$

Using the consistency condition (7.24), the definition (7.22) of $\boldsymbol{\mu}^j$ and the decomposition formula (7.12), we find the desired result

$$\hat{\boldsymbol{u}}_b^{(s),j} = \boldsymbol{B}^{(s,-),j}\tilde{\boldsymbol{b}}^{(s,-),j}\boldsymbol{u}_b^{(s),j} + \tilde{\boldsymbol{b}}^{*(s,-),j}\boldsymbol{\mu}^j = \boldsymbol{u}_b^{(s),j} \tag{7.27}$$

Let $\Delta\boldsymbol{u}_b^{(s),j}$ be defined as

$$\Delta\boldsymbol{u}_b^{(s),j} = \hat{\boldsymbol{u}}_b^{(s),j} - \boldsymbol{u}_b^{(s),j} \tag{7.28}$$

From Eqs. (7.23), the consistency condition (7.24) and the decomposition (7.12), it follows that the displacement correction is computed by

$$\Delta\boldsymbol{u}_b^{(s),j} = -\sum_{\substack{r:\Gamma_I^{(r)}\ni j \\ r\neq s}} \beta^{(r),j}\ \boldsymbol{B}^{(s,-),j}\boldsymbol{r}^{(s,r),j} \tag{7.29}$$

where $\boldsymbol{r}^{(s,r),j}$ is the interface compatibility error defined by (7.18). In the particular case of conforming interfaces, the above expression simplifies to

$$\Delta \boldsymbol{u}_b^{(s),j} = -\tilde{\boldsymbol{b}}^{(s,-),j^T} \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \beta^{(r),j}\ \boldsymbol{r}^{(s,r),j} \tag{7.30}$$

For an arbitrary substructure problem, the total energy can be written as

$$\mathcal{E}(\beta^{(s),j}) = \sum_{s=1}^{N_s} \left\{ \frac{1}{2} \hat{\boldsymbol{u}}^{(s)^T} \boldsymbol{K}^{(s)} \hat{\boldsymbol{u}}^{(s)} - \hat{\boldsymbol{u}}^{(s)^T} \boldsymbol{f}^{(s)} \right\} \tag{7.31}$$

where no Lagrange multiplier term is involved since the smoothed solution defines a displacement field which is kinematically admissible for the reference compatibility. If the effect of interface smoothing is back-propagated to the substructure interiors as in Eq. (7.23b), $\mathcal{E}(\beta^{(s),j})$ can be re-written by making use of Eqs. (5.2a,5.2b,7.23,7.29) as

$$\begin{aligned}
\mathcal{E}(\beta^{(s),j}) &= \sum_{s=1}^{N_s} \left\{ \frac{1}{2} \boldsymbol{u}^{(s)^T} \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s)} - \boldsymbol{u}^{(s)^T} \boldsymbol{f}^{(s)} \right\} && (7.32) \\
&\quad + \sum_{s=1}^{N_s} \left\{ \frac{1}{2} \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{K}^{(s)} \Delta \boldsymbol{u}^{(s)} + \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s)} - \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{f}^{(s)} \right\} \\
&= -\sum_{s=1}^{N_s} \frac{1}{2} \boldsymbol{u}^{(s)^T} (\boldsymbol{f}^{(s)} + \boldsymbol{B}^{(s),p} \boldsymbol{\lambda}^p) + \sum_{s=1}^{N_s} \frac{1}{2} \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{K}^{(s)} \Delta \boldsymbol{u}^{(s)} \\
&\quad - \sum_{s=1}^{N_s} \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{B}^{(s),p} \boldsymbol{\lambda}^p && (7.33) \\
&= -\sum_{s=1}^{N_s} \frac{1}{2} \boldsymbol{u}^{(s)^T} (\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s),p} \boldsymbol{\lambda}^p) + \sum_{s=1}^{N_s} \frac{1}{2} \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{K}^{(s)} \Delta \boldsymbol{u}^{(s)} \\
&\quad - \sum_{s=1}^{N_s} \hat{\boldsymbol{u}}^{(s)^T} \boldsymbol{B}^{(s),p} \boldsymbol{\lambda}^p && (7.34)
\end{aligned}$$

Noting that the smoothed solution satisfies the reference compatibility and thus also the weaker compatibility $\boldsymbol{B}^{(s),p^T} \hat{\boldsymbol{u}}^{(s)} = \boldsymbol{0}$, we finally find

$$\begin{aligned}
\mathcal{E}(\beta^{(s),j}) &= Cst + \sum_{s=1}^{N_s} \frac{1}{2} \Delta \boldsymbol{u}^{(s)^T} \boldsymbol{K}^{(s)} \Delta \boldsymbol{u}^{(s)} && (7.35) \\
&= Cst + \sum_{s=1}^{N_s} \frac{1}{2} \Delta \boldsymbol{u}_b^{(s)^T} \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} && (7.36)
\end{aligned}$$

where

$$Cst = -\sum_{s=1}^{N_s} \frac{1}{2} \boldsymbol{u}^{(s)^T} \boldsymbol{f}^{(s)} \tag{7.37}$$

is a constant term independent of the smoothing parameters $\beta^{(s),j}$. On the other hand, if the effect of interface smoothing is not back-propagated to the substructure interiors (lumped smoothing), the smoothing of the subdomain interiors (7.23b) is replaced by

$$\hat{\boldsymbol{u}}_i^{(s)} = \boldsymbol{u}_i^{(s)} \tag{7.38}$$

$\mathcal{E}(\beta^{(s),j})$ will have a similar expression as (7.36) but with $\boldsymbol{K}_{bb}^{(s)}$ replacing every occurrence of $\boldsymbol{S}_{bb}^{(s)}$.

If the consistency condition (7.24) is enforced on interface j by a set of multipliers τ_j, minimizing the energy with respect to the smoothing parameters $\beta^{(q),j}$ yields

$$\sum_{\substack{s:\Gamma_I^{(s)}\ni j \\ s\neq q}} \sum_{i:\Gamma_I^{(s)}\ni i} \sum_{\substack{p:\Gamma_I^{(p)}\ni i \\ p\neq s}} k_s^{(q),j;(p),i}\ \beta^{(p),i} = \tau_j \quad \text{(7.39a)}$$

$$\forall[(q),j] : \Gamma_I^{(q)} \ni j$$

$$\sum_{s:\Gamma_I^{(s)}\ni j} \beta^{(s),j} = 1 \quad \text{(7.39b)}$$

where s describes the subdomains connected to the edge j, i are the edges of subdomain s and p designates the subdomains attached to $\Omega^{(s)}$ on i. The global stiffness coefficients are defined by

$$k_s^{(q),j;(p),i} = \left(\boldsymbol{B}^{(s,-),j}\boldsymbol{r}^{(s,q)}\right)^T [\boldsymbol{S}_{bb}^{(s)}]_{j,i} \left(\boldsymbol{B}^{(s,-),i}\boldsymbol{r}^{(s,p)}\right) \quad (7.40)$$

and $[\boldsymbol{S}_{bb}^{(s)}]_{j,i}$ is the Schur-complement of $\boldsymbol{K}^{(s)}$ associated with the edges j and i. If we call $\boldsymbol{f}_s^{(q),j}$ the force produced on the boundary of $\Omega^{(s)}$ when a displacement $\boldsymbol{r}^{(s,q),j}$ is applied on $\Gamma_I^{sq,j}$, the remaining portion of $\Gamma_I^{(s)}$ being clamped, $k_s^{(q),j;(p),i}$ is the work produced by $\boldsymbol{f}_s^{(q),j}$ for a displacement $\boldsymbol{r}^{(s,p),i}$ on $\Gamma_I^{sp,i}$.

Hence, the smoothing parameters $\beta^{(s),j}$ are given by the solution of an auxiliary "coarse" problem. If the consistency constraints (7.24) are enforced explicitly — that is, the redundant smoothing parameters are eliminated during the setup of the auxiliary problem — the size of this auxiliary problem becomes roughly equal to the number of edges in the substructure problem.

There is no question that the system of equations (7.39) is difficult to read. However, its implementation is remarkably simple.

Remarks

- The terminology of "smoothing" comes from the multigrid context where a coarse grid is used to solve for the global behavior of structures while the fine components of the solution are obtained by considering the fine grid. Calling "smoothing" our post-processing of the weakly compatible solution is thus licit if one considers the analogy between the coarse grid (7.39) of the subdomain stiffnesses and the coarse grid used in the multigrid context: considering the subdomains of the substructured model as macro-elements of a coarse grid, the auxiliary problem (7.39) can be interpreted as defining the global behavior of the model. However, whereas the multigrid coarse grid defines only smooth global corrections to the fine grid solution by defining a classical finite element coarse grid, the corrections brought to the weakly compatible solutions such as defined by (7.29) are functions of the interface compatibility error.

- As noted for the two-substructure smoothing, one can also define more than one smoothing coefficient per subdomain and per edge. One could define a different set of smoothing parameters for each type of displacement, for instance one for horizontal displacements and another for vertical displacements, or for displacements and rotations. This increases the coarse grid size, but might significantly improve the accuracy of the smoothed solution. Especially in cases when the global stiffness of the subdomains is very dependent on the type of d.o.f. under consideration, introducing an interface smoothing parameter per displacement direction increases the number of "macro" d.o.f. and generally results in a dramatic improvement of the solution.

- For homogeneous substructures, one could choose not to solve the coarse grid problem (7.39) for determining the smoothing parameters $\beta^{(s),j}$, and determine a solution satisfying the reference compatibility by a simple averaging scheme as presented in section 5.4 for the two-subdomain case. Considering the definition (7.23) of $\hat{\boldsymbol{u}}^{(s)}$ and the consistency condition (7.24), a compatible solution can be defined by taking for the parameters $\beta^{(s),j}$ the same value

$$\beta^{(s),j} = \frac{1}{m_j} \qquad \forall s : \Gamma_I^{(s)} \ni j \tag{7.41}$$

  m_j being the multiplicity of the edge. By the very definition of the multiplicity the consistency condition (7.24) is then verified:

$$\sum_{s:\Gamma_I^{(s)} \ni j} \frac{1}{m_j} = \frac{m_j}{m_j} \tag{7.42}$$

  We thus generalize the averaging procedure (5.23) to multiple conforming and non-conforming substructures by substituting (7.41) into the correction formulae (7.23):

$$\Delta \boldsymbol{u}_b^{(s),j} = -\frac{1}{m_j} \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{B}^{(r,-),j} \boldsymbol{r}^{(s,r),j} \tag{7.43}$$

  where $\boldsymbol{B}^{(r,-),j} = \tilde{\boldsymbol{b}}^{(r,-),j^T}$ for conforming interfaces.

7.4 Eliminating the cross-point smoothing parameters

A major inconvenience of the general smoothing procedure as described in this chapter is that for cross-points ($m_i > 2$) one must define m_i smoothing parameters. The number of smoothing parameters and thus the dimension of the coarse grid may become large (although it still remains much smaller then the full interface problem). Especially in 3-dimension problems, the multiplicity of a cross-point may become very high: if we consider e.g. the simple decomposition of a cube into sub-cubes, the cross-points are of multiplicity $m_i = 4$ for intersection lines and of multiplicity $m_i = 8$ for intersection points.

In practice, a cross-point defines a negligible portion of the entire interface, namely the shape functions pertaining to a corner d.o.f. have a support domain which measure goes to zero as the substructure meshes are refined. Therefore, one should consider if the smoothing parameters on the cross-points significantly participate to the improvement of the overall solution. In some very special cases where the cross-points correspond to mechanical singularities such as stress concentration points, the associated smoothing parameters clearly play an important role in defining a good smoothed solution. However in most practical situations, the dimension of the smoothing coarse grid can be kept very low by applying the following approximate smoothing technique.

The approximation of the smoothing method is still based on a strongly compatible smoothed solution defined by (7.23), but instead of considering the smoothing parameters $\beta^{(s),j)}$ pertaining to a cross-point as variables, a first technique consists in giving to the cross-point parameters the value $1/m_j$ which corresponds to applying a simple averaging scheme on all edges of multiplicity superior to 2. Another and more efficient approach is to link the cross-point parameters to the smoothing parameters defined on the neighboring edges of multiplicity $m_i = 2$. As an example let us consider the case of a cross-point c of multiplicity $m_c = 4$ in a 2-dimension problem (Fig. 7.2). One can impose that the cross-point parameters $\beta^{(s),c}$ are given by

$$\beta^{(1),c} = \left(\beta^{(1),1} + \beta^{(1),2}\right)/4 \tag{7.44a}$$

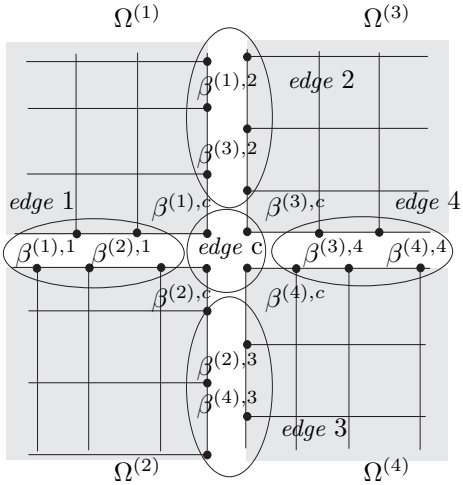


Figure 7.2: Cross-point of a 2-dimensional problem

$$\beta^{(2),c} = \left(\beta^{(2),1} + \beta^{(2),3}\right)/4 \tag{7.44b}$$

$$\beta^{(3),c} = \left(\beta^{(3),2} + \beta^{(3),4}\right)/4 \tag{7.44c}$$

$$\beta^{(4),c} = \left(\beta^{(4),3} + \beta^{(4),4}\right)/4 \tag{7.44d}$$

The consistency condition on the cross-point parameters is then automatically satisfied since the adjacent edges also satisfy a unit sum constraint. The subdomain stiffnesses relative to the cross-point is thus redistributed among the adjacent edges and a small dimension coarse grid can be derived by minimizing the total energy with respect to the remaining smoothing parameters of the edges of multiplicity equal to 2.

7.5 A case study: the four-subdomain problem

To better understand the smoothing procedure developed in the previous sections for multiple substructures, we will apply the smoothing technique to the clamped-free beam considered in section 5.5.1 and pictured in Fig. 7.3. Each structure is modeled by 10×10 plane stress and linear elements. We assume that the Young modulus of the upper right subdomain, called $\Omega^{(3)}$, is 10 times smaller than the Young modulus of the three other subdomains. On Fig. 7.3 we also show the five interface edges.

Approximating the connecting loads between the subdomains by polynomial functions of the order of $p = 2$, we define $p + 1 = 3$ interface unknowns per interface and per direction. Hence neglecting the cross-point compatibility for the reduced problem, we solve a reduced two-field hybrid method with $n_{\lambda^p} = 3 \times 2 \times 4 = 24$ which is small compared to the total number of discrete reference compatibility constraints $n_\lambda = 84$. Fig. 7.4 plots the non-smoothed displacements on the horizontal and vertical interfaces obtained with the reduced system and compares them with the reference displacements relative to the fully assembled problem. The result curves reveal that the interface compatibility is unsatisfactory and that only low credit should be given to the solution along the boundary of the soft subdomain $\Omega^{(3)}$.

We then apply the smoothing procedure as defined by (7.23). Using the edge numbering of Fig. 7.3, we define the smoothed solutions by

$$\hat{\boldsymbol{u}}_b^{(1),1} = \beta^{(1),1}\boldsymbol{u}_b^{(1),1} + \beta^{(2),1}\boldsymbol{u}_b^{(2),1} = \hat{\boldsymbol{u}}_b^{(2),1} \tag{7.45a}$$

$$\hat{\boldsymbol{u}}_b^{(1),2} = \beta^{(1),2}\boldsymbol{u}_b^{(1),2} + \beta^{(3),2}\boldsymbol{u}_b^{(3),2} = \hat{\boldsymbol{u}}_b^{(3),2} \tag{7.45b}$$

$$\hat{\boldsymbol{u}}_b^{(2),3} = \beta^{(2),3}\boldsymbol{u}_b^{(2),3} + \beta^{(4),3}\boldsymbol{u}_b^{(4),3} = \hat{\boldsymbol{u}}_b^{(4),3} \tag{7.45c}$$

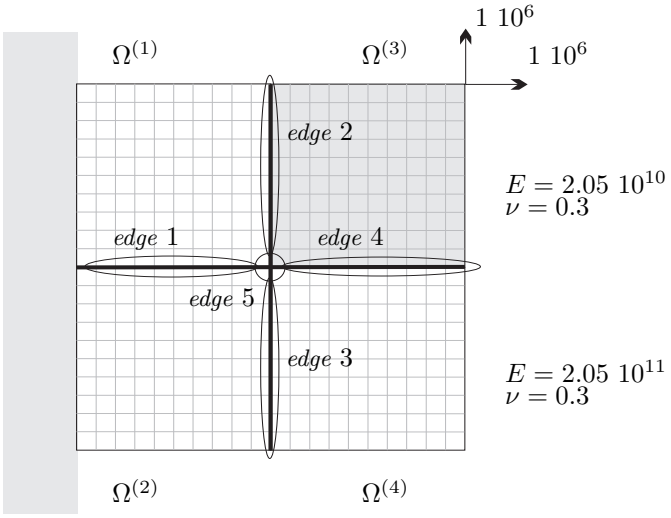
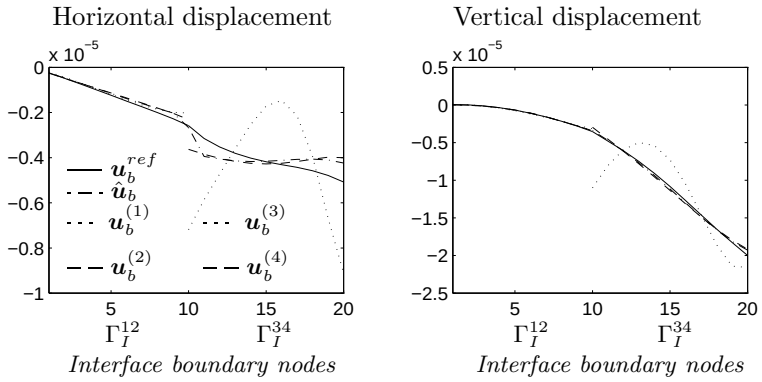
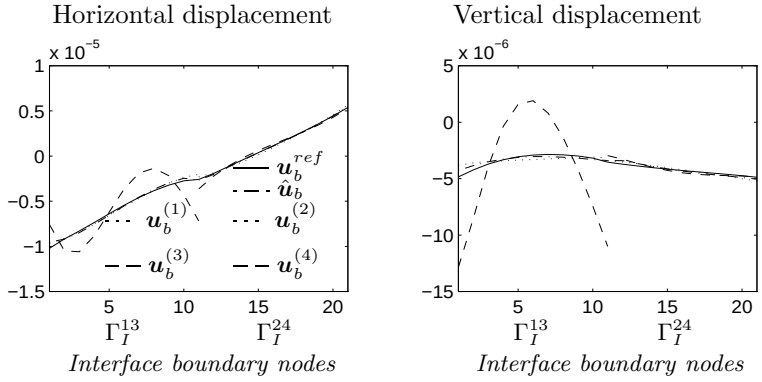


Figure 7.3: Clamped beam example with four subdomains



(a) Horizontal interface



(b) Vertical interface

Figure 7.4: Weakly compatible and smoothed interface displacements of the clamped beam problem ($p = 2$ on all Γ_I^{sr})

$$\hat{\boldsymbol{u}}_b^{(3),4} = \beta^{(3),4}\boldsymbol{u}_b^{(3),4} + \beta^{(4),4}\boldsymbol{u}_b^{(4),4} = \hat{\boldsymbol{u}}_b^{(4),4} \tag{7.45d}$$

$$\begin{aligned}\hat{\boldsymbol{u}}_b^{(1),5} &= \beta^{(1),5}\boldsymbol{u}_b^{(1),5} + \beta^{(2),5}\boldsymbol{u}_b^{(2),5} + \beta^{(3),5}\boldsymbol{u}_b^{(3),5} + \beta^{(4),5}\boldsymbol{u}_b^{(4),5} \\ &= \hat{\boldsymbol{u}}_b^{(2),5} = \hat{\boldsymbol{u}}_b^{(3),5} = \hat{\boldsymbol{u}}_b^{(4),5}\end{aligned} \tag{7.45e}$$

where we assume that the ordering of the boundary nodes on each side of an edge exactly correspond. To determine the smoothing coefficients $\beta^{(s),j}$ we compute the subdomain global stiffness coefficients $k_s^{(q),j;(p),i}$ by (7.40). For instance the stiffness coefficients relative to the first subdomain $\Omega^{(1)}$ are computed by

$$k_1^{(2),1;(2),1} = \boldsymbol{r}^{(1,2),1^T}[\boldsymbol{S}_{bb}^{(1)}]_{1,1}\boldsymbol{r}^{(1,2),1} \tag{7.46a}$$

$$k_1^{(3),2;(3),2} = \boldsymbol{r}^{(1,3),2^T}[\boldsymbol{S}_{bb}^{(1)}]_{2,2}\boldsymbol{r}^{(1,3),2} \tag{7.46b}$$

$$k_1^{(2),1;(3),2} = \boldsymbol{r}^{(1,2),1^T}[\boldsymbol{S}_{bb}^{(1)}]_{1,2}\boldsymbol{r}^{(1,3),2} = k_1^{(3),2;(2),3} \tag{7.46c}$$

$$k_1^{(4),5;(4),5} = \boldsymbol{r}^{(1,4),5^T}[\boldsymbol{S}_{bb}^{(1)}]_{5,5}\boldsymbol{r}^{(1,4),5} \tag{7.46d}$$

$$k_1^{(4),5;(3),2} = \boldsymbol{r}^{(1,4),5^T}[\boldsymbol{S}_{bb}^{(1)}]_{5,2}\boldsymbol{r}^{(1,3),2} = k_1^{(3),2;(4),5} \tag{7.46e}$$

$$k_1^{(4),5;(2),1} = \boldsymbol{r}^{(1,4),5^T}[\boldsymbol{S}_{bb}^{(1)}]_{5,1}\boldsymbol{r}^{(1,2),1} = k_1^{(2),1;(4),5} \tag{7.46f}$$

where $\boldsymbol{r}^{(1,s),j} = \boldsymbol{u}_b^{(1),j} - \boldsymbol{u}_b^{(s),j}$ is the compatibility error between $\Omega^{(1)}$ and a neighboring subdomain $\Omega^{(s)}$ on edge j. We then set up the coarse grid (7.39) and solve for the smoothing coefficients. We find:

$$\beta^{(1),1} = 0.28 \qquad \beta^{(2),1} = 0.72 \tag{7.47a}$$

$$\beta^{(1),2} = 0.95 \qquad \beta^{(3),2} = 0.05 \tag{7.47b}$$

$$\beta^{(2),3} = 0.55 \qquad \beta^{(4),3} = 0.45 \tag{7.47c}$$

$$\beta^{(3),4} = 0.05 \qquad \beta^{(4),4} = 0.95 \tag{7.47d}$$

$$\beta^{(1),5} = 0.66 \qquad \beta^{(2),5} = 0.02 \qquad \beta^{(3),5} = 0.04 \qquad \beta^{(4),5} = 0.28 \tag{7.47e}$$

and the smoothed displacements are found by replacing in (7.45). The smoothed interface solutions are plotted in Fig. 7.4

Finally we can assess the efficiency of the smoothing procedure by computing the relative energy norm of the displacement error for the smoothed and the non-smoothed solution defined by Eqs. (5.11,5.32,5.33):

$$\varepsilon = 7.31\ 10^{-2} \tag{7.48}$$

$$\hat{\varepsilon} = 2.64\ 10^{-2} \tag{7.49}$$

From these error measures and from Fig. 7.4 we conclude that the smoothing procedure is very efficient in selecting the weightings of the weakly compatible solutions for defining a strongly compatible smoothed result: determining the smoothing coefficients by minimizing the total energy of the post-processed solution leads to choosing the solution according to the subdomain global stiffnesses. In this case for instance, considering the values of the smoothing coefficients (7.47) we see that the solution along the boundary of the soft subdomain $\Omega^{(3)}$ is given very low weightings as we expected.

Remark

- To the cross-point, defined in this example as the edge number 5, corresponds four smoothing variables $\beta^{(s),5}, s = 1, \ldots 4$. These smoothing parameters are actually redundant since these 4 parameters must define 2 displacements, that is u_x and u_y of the cross-point, and are constrained to have a unit sum. Hence the cross-point leads to $4 - 2 - 1 = 1$ singularity in the coarse grid problem. Nevertheless, one can show that the coarse grid has a solution which is not unique, but which yields a unique smoothed displacement field. Therefore, the

only precaution one must take is to solve the coarse grid with an algorithm that handles singularities. In our case we chose to solve the coarse grid by explicitly substituting the unit sum constraints. The coarse grid is then solved either by a factorization in which case the parameters corresponding to zero pivots are set to zero, or by a basic Conjugate Gradient.

- As a consequence of the redundancy of the cross-point parameters, the associated parameters are not uniquely defined. This explains why the parameter $\beta^{(2),5}$ is so small in Eq. (7.47e).

If we now assume that the cross-point smoothing parameters are linked to the smoothing parameters defined on the edges of multiplicity equal to 2 (see discussion in Section 7.4), the size of the coarse grid is reduced from 12 (with 5 consistency constraints) to 8 (with 4 consistency constraints). Thus the number of independent smoothing variables is reduced from 7 to 4. The smoothed interface displacements are very similar to those shown on Fig. 7.4 and the energy norm of the displacement error in now equal to $\hat{\varepsilon} = 2.67\ 10^{-2}$ which is only slightly higher than the energy norm of the displacement error (7.49) when the complete smoothing procedure is applied. Therefore even in this case where the cross-point is a region of stress concentration, eliminating the cross-point smoothing variables in the coarse grid does not significantly affect the accuracy of the smoothed solution.

7.6 Treatment of a physical cross-point in the presence of non-conforming interfaces

Assigning a substructure interface node to one and only one substructure edge j (see Eq. (7.4)), and using the same approximation for all the Lagrange multiplier fields interconnecting any pair of substructures $\Omega^{(s)}$ and $\Omega^{(q)}$ along the edge j, are fundamental to the implementation of the generalized smoothing procedure proposed in this work. The impact of these two prerequisites on the treatment of a potential physical crosspoint in a substructure problem with non-matching interfaces is as follows (in the case of matching interfaces, our smoothing procedure does not require a special treatment for the crosspoint).

Consider the "T" three-substructure problem with incompatible interfaces depicted in Fig. 7.5. The two points A and B are the substructure representatives of the same physical crosspoint. In this particular but revealing example, points A and B have the same coordinates, but point C has different ones. In order to respect the two smoothing prerequisites outlined above, we recommend two approaches.

In a first approach, two points A' and B' are inserted in the substructures $\Omega^{(1)}$ and $\Omega^{(2)}$, respectively. Point A' is positioned as to match the first node at the left of point C in substructure $\Omega^{(3)}$, and point B' is positioned as to match point C. The interface boundary Γ_I is then decomposed into 3 edges corresponding to Γ_I^{12}, Γ_I^{13} and Γ_I^{23} (Fig. 7.5a.) The advantage of this approach is that it requires few edges, but on the other hand it does not enforce a strong compatibility of the displacement fields between each of the points A and B and any corresponding point on $\Gamma_I^{(3)}$ — that is, at the physical crosspoint. Depending on the smoothness of the solution at the physical crosspoint, and the distances between A' and B', this may or may not be a problem.

In a second approach, an additional point C' is inserted at the left of C in $\Omega^{(3)}$ and positioned as to match the points A and B (Fig. 7.5b). This approach generates more edges than the previous one — in particular, each of the points A, B, and C' is now an edge — but enforces compatibility at the physical crosspoint.

7.7 Applications

Throughout this section, smoothing refers to the "complete" smoothing procedure determined by Eqs. (7.23,7.39). We will present some relevant results of the smoothing procedure applied to the static analysis of two-dimensional and three-dimensional structures. Obviously, the generalized

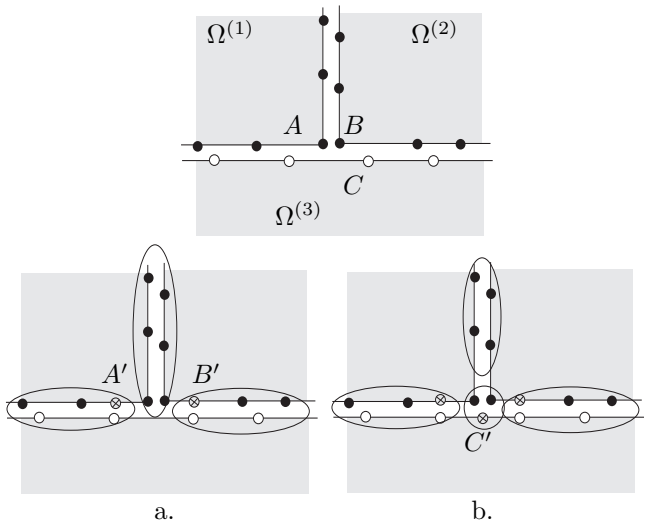


Figure 7.5: A "T" three-substructure problems with a physical crosspoint

smoothing procedure can also be applied to free vibration and dynamic response analysis in a way similar to the methods presented in section 6.3.2 for the two-subdomain case. However, no particular difficulty appears in comparison with the two-subdomain dynamic case. Therefore we will limit our examples to static analyses only.

7.7.1 Global/local analysis of a plate problem

First, we demonstrate the efficiency of the two-field hybrid method and smoothing procedure with the global/local analysis of the uniformly pressure-loaded plate problem discussed in [8]. For this clamped plate, the maximum stresses are at the mid-sides of the plate. For obvious symmetry reasons, only a quarter of the structure is discretized and analyzed using 3-node DKT (Discrete Kirchhoff Triangle) plate elements with 3 d.o.f. per node [14].

The finite element meshes of the global and local models are depicted in Fig. 7.6. They contain 143 and 320 active d.o.f., respectively, and have non-matching interfaces. The local substructure has been specifically designed to capture the maximum Von Mises stress point. For comparison purposes, a converged finite element solution is computed on a uniformly refined mesh containing 7203 active d.o.f. We refer to the latter solution as the "exact" solution.

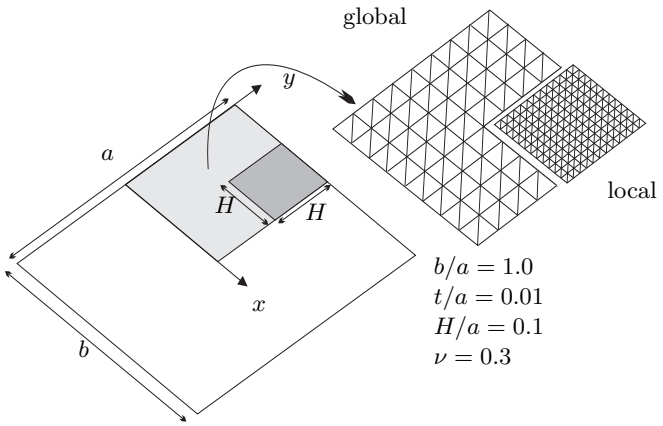


Figure 7.6: Global/local analysis of a pressure-loaded plate [8]

Since in a Kirchhoff plate element the effect of the normal rotation is included in the finite element representation of the transverse displacement, only the continuity of the tangential rotational and vertical displacement d.o.f. are explicitly enforced across the global/local substructure interfaces, using distinct p_θ-th and p_w-th order polynomial Lagrange multipliers, respectively. The interface Γ_I is sub-divided in two interface segments, one parallel to x and the other parallel to y. The constraint matrices are computed as follows: for the rotation tangent to the interfaces, we define the compatibility constraint

$$\int_{\Gamma_I} \Lambda_\theta^T \boldsymbol{N}_\theta d\Gamma \ \boldsymbol{\theta}_t = \mathbf{0} \tag{7.50}$$

where $\boldsymbol{\theta}_t$ are the tangential rotation d.o.f., Λ_θ is a polynomial function of order p_θ and $\boldsymbol{N}_\theta$ are the linear shape functions $\boldsymbol{\theta}_t$ on the edges of the DKT elements. For the vertical displacements we define the compatibility

$$\int_{\Gamma_I} \Lambda_w^T \boldsymbol{N}_w d\Gamma \ \left[\begin{array}{c} \boldsymbol{w} \\ \boldsymbol{\theta}_n \end{array} \right] = \mathbf{0} \tag{7.51}$$

where $\boldsymbol{\theta}_n$ and $\boldsymbol{w}$ are respectively the rotation normal to the interface and the vertical displacements, Λ_w is a polynomial function of order p_w and $\boldsymbol{N}_w$ are the cubic shape functions associated with the Kirchhoff beam assumption along the edges of the DKT elements. The reference solution has a compatibility defined with polynomials of degree $q_\theta = 3$ and $q_w = 7$ on each interface segment.

The results of the global/local analysis using the two-field hybrid method with and without a smoothing are compared in Table 7.1 with the exact solution (converged finite element solution on the 7203 d.o.f. mesh). In this table, w^{max} and σ^{max} denote respectively the maximum vertical deflection and the maximum Von Mises stress.

	$\{p_\theta, p_w\}$	$\{0,0\}$	$\{0,1\}$	$\{1,0\}$	$\{1,1\}$	$\{1,2\}$
$\frac{w_{max}}{w_{max}^{exact}}$	no smoothing	1.012	1.012	1.011	1.011	1.011
	smoothing	1.011	1.011	1.011	1.011	1.011
$\frac{\sigma_{max}}{\sigma_{max}^{exact}}$	no smoothing	0.994	0.995	0.998	0.998	0.998
	smoothing	0.997	0.998	0.998	0.998	0.998

Table 7.1: Global/local analysis of a pressure-loaded plate

As expected, the two-field hybrid method applied to the non-matching meshing defines only weak compatibility constraints and thus slightly overestimates the displacements and underestimates the stresses. Clearly, even for $\{p_\theta, p_w\} = 0, 0$ — that is, for a constant approximation of the Lagrange multipliers along the interface between the global and local models — the results of the two-field hybrid method are in excellent agreement with the exact solution. Note that in our case, $\{p_\theta, p_w\} = \{0, 0\}$ implies that only 4 Lagrange multipliers are used to connect the global and local model interfaces. In reference [8], a variant three-field hybrid method was used to solve this problem: six interface "pseudo-nodes" and at least 18 Lagrange multipliers are employed to produce a similar accuracy. However, such impressive results are possible only because the displacement solution of the pressure-loaded clamped plate happens to be smooth at the global/local model interfaces. Nevertheless, this also means that the two-field hybrid method is well suited and therefore should be used for this class of applications.

7.7.2 Analysis of a checkerboard elastomer/steel structure

Next, we consider the plane stress analysis of a two-dimensional structure assembled in a checkerboard fashion from six independently designed cast iron, steel, and rubber squared subcomponents of unit length. The global system is clamped at one end, and subjected to horizontal and vertical point loads at the top of the other end (Fig. 7.7). The rubber substructures are characterized by a Young modulus $E^{(1)} = E^{(5)} = E^{(6)} = 5.0 \times 10^7$ and a Poisson ratio $\nu^{(1)} = \nu^{(5)} = \nu^{(6)} = 0.48$,

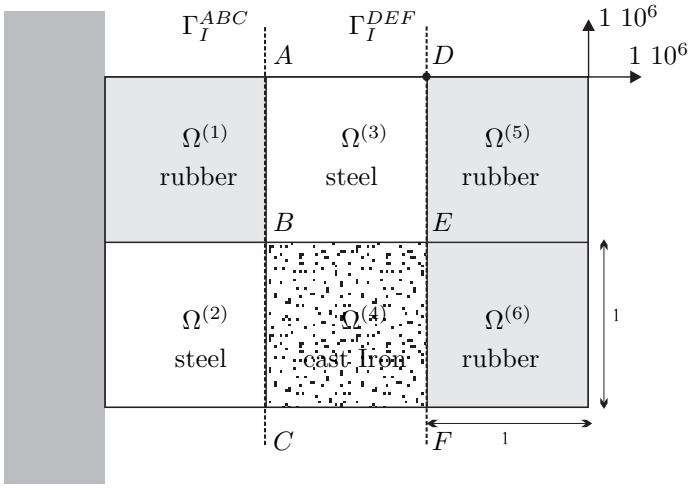


Figure 7.7: A checkerboard elastomer/steel structure

the cast iron subcomponent by $E^{(4)} = 1.025 \times 10^{11}$ and $\nu^{(4)} = 0.30$, and the steel substructures by $E^{(2)} = E^{(3)} = 2.05 \times 10^{11}$ and $\nu^{(2)} = \nu^{(3)} = 0.3$. Hence, of particular interest in this problem is the high degree of heterogeneity measured by $\alpha = E^{(1)}/E^{(2)} = 2.44 \times 10^{-4}$, and the presence of several cross-points between extremely stiff and extremely flexible substructures.

We focus on two scenarios where (a) all substructures are discretized into 15×15 4-node plane stress elements with 2 d.o.f. per node and have matching interfaces, and (b) the rubber subcomponents are discretized into 15×15 elements and the cast iron and steel substructures into 10×10 4-node plane stress elements, and therefore have non-matching interfaces. In scenario (a), the compatibility between the substructure interfaces can be enforced in an exact manner using discrete Lagrange multipliers at every interface d.o.f. However, even in this scenario, we utilize here polynomial Lagrange multipliers in order to illustrate their convergence properties for problems where their use is motivated by the reduction of the computational size of the interface system [47,77], rather than by the non-matching nature of the substructure interfaces. In both scenarios, we apply the two-field hybrid method with and without smoothing. We monitor the quality of the computed solutions at the substructure interfaces $\Gamma_I^{ABC} = \Gamma_I^{13} \bigcup \Gamma_I^{24}$ and $\Gamma_I^{DEF} = \Gamma_I^{35} \bigcup \Gamma_I^{46}$. Note that in this problem the cross-points are high stress concentration points so that eliminating the cross-points smoothing parameters from the coarse grid (see Section 7.4) is not a good approach in this case.

First, we discuss the solution obtained for scenario (a) where all the subcomponents have the same discretization and matching interfaces. We report in Fig. 7.8-7.9 the interface displacements computed by the two-field hybrid method, with and without smoothing, for $p = 1, 2, 3$ on every interface Γ_I^{sr}, together with the reference solution. We plot separately the interface displacement fields in the rubber subcomponents and those in the steel and cast iron substructures, because these displacement fields have significantly different magnitudes. The following observations are noteworthy:

- The effects of the strong heterogeneities of this assembly problem are clearly demonstrated by the oscillatory behavior of the non-smoothed solution at the interfaces of the flexible subcomponents and their crosspoints.

- With and without smoothing, the two-field hybrid method is shown to converge when the polynomial degree p is increased.

- However, for an equal p, the two-field hybrid method with smoothing is shown to be significantly more accurate than without smoothing, especially in the rubber subcomponents and

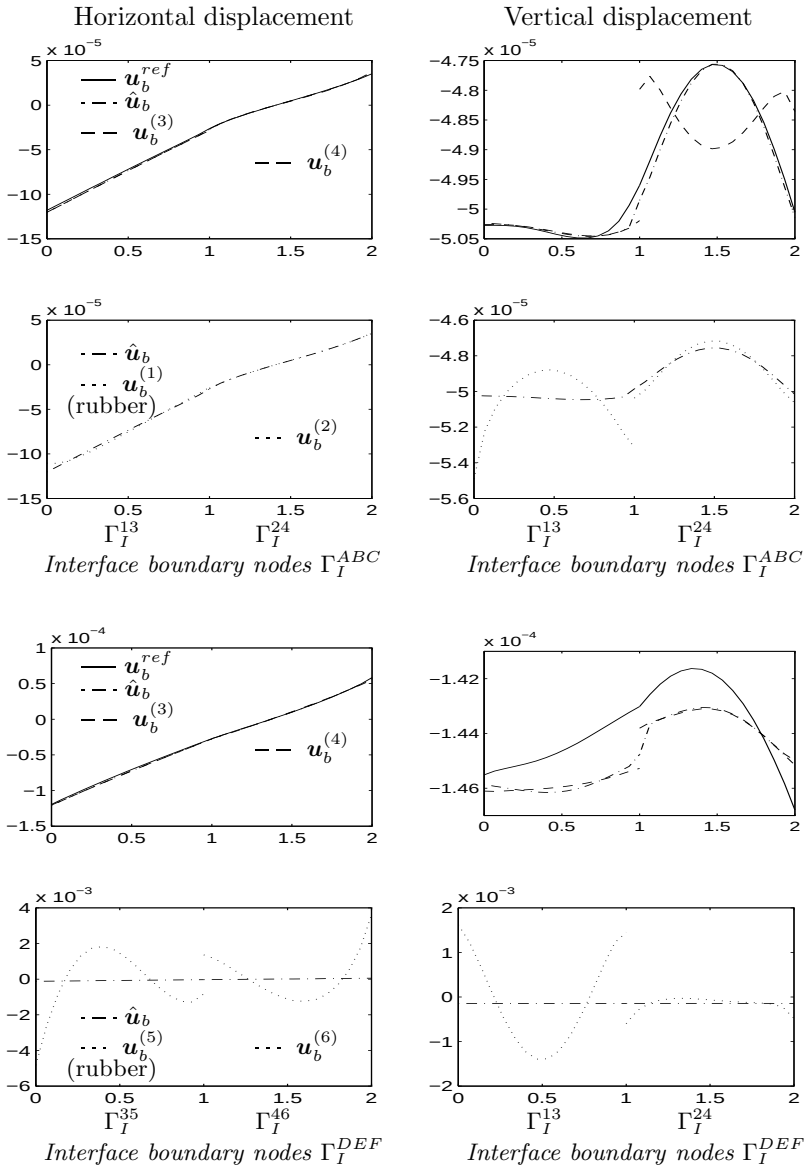


Figure 7.8: Conforming checkerboard elastomer/steel structure - $p = 1$

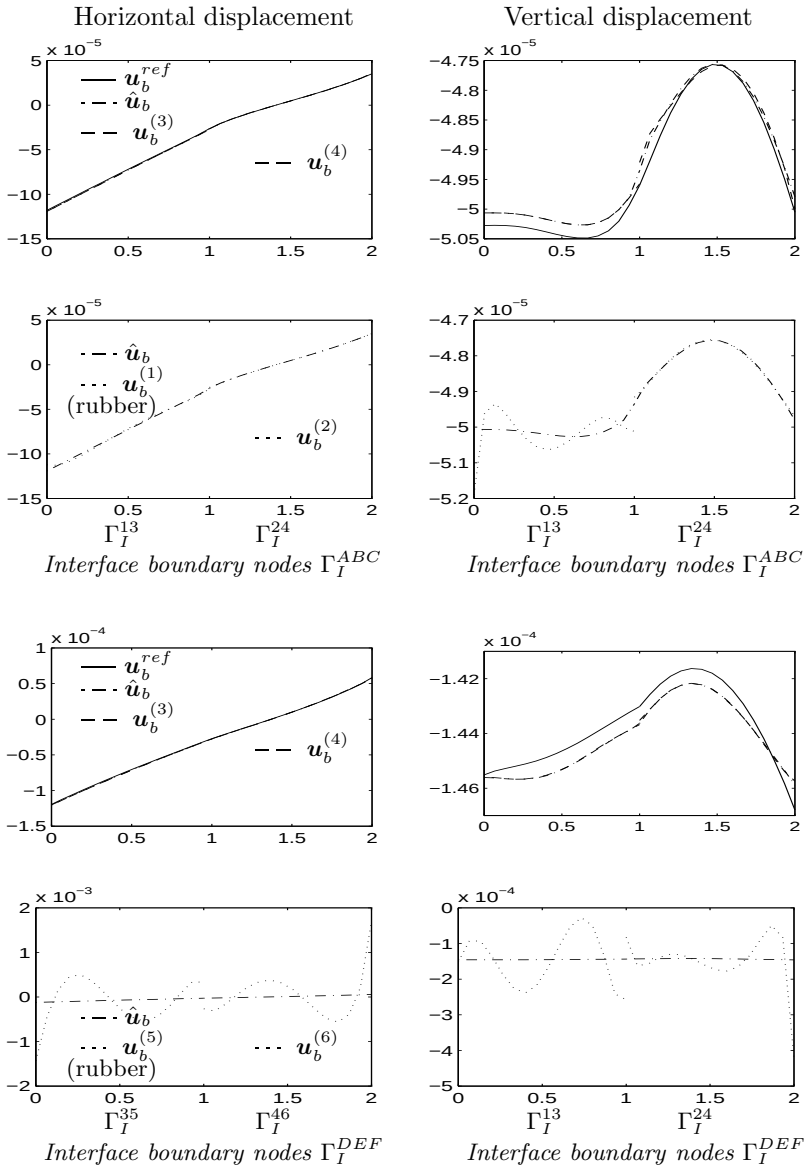


Figure 7.9: Conforming checkerboard elastomer/steel structure - $p = 2$

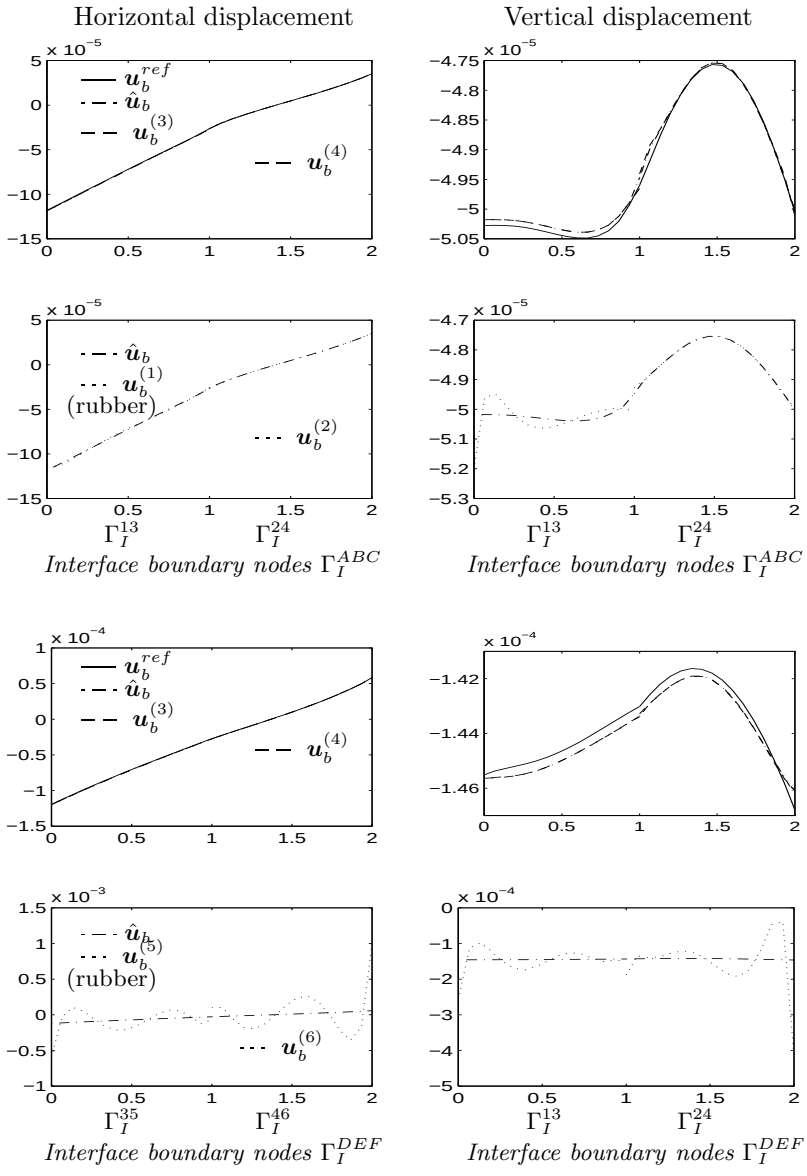


Figure 7.10: Conforming checkerboard elastomer/steel structure - $p = 3$

at the crosspoint E. Without smoothing, the trace of the assembled solution on Γ_I^{DEF} is oscillatory. With smoothing, it is in nearly perfect agreement with the exact solution, even for a polynomial degree as low as $p = 1$,that is for a total number of Lagrange multipliers $n_{\lambda^p} = 28$. The number of discrete constraints for the reference solution being 230, this example clearly shows the advantage of using a reduced interface formulation in conjunction with our efficient smoothing procedure.

Table 7.2 reports the energy norms ε and $\hat{\varepsilon}$ (5.32-5.33) of the displacement errors between the reference solution and the non-smoothed solution and between the reference solution and the smoothed solution. According to these energy norms, the smoothed solution not only is

polyn. degree	$p = 1$	$p = 2$	$p = 3$
ε	0.077	0.029	0.017
$\hat{\varepsilon}$	0.018	0.0036	0.0013

Table 7.2: Checkerboard structure: energy norm of displacement error

characterized by a higher level of accuracy, but its convergence towards the reference solution is quicker.

Next, we focus on scenario (b) where some substructure interfaces do not match because the rubber subcomponents are discretized into 15×15 plane stress elements, while the cast iron and steel subsystems are discretized into 10×10 4-node plane stress elements. We employ the two-field hybrid method with $p = 1, 2, 3$ and apply a smoothing procedure where the reference compatibility is established with interface polynomials of order 9.

We report in Table 7.3 the energy norm of the displacement error ε and $\hat{\varepsilon}$ for the non-smoothed and the smoothed solution respectively. The energy norm gives more importance to displacement

polyn. degree	$p = 1$	$p = 2$	$p = 3$
ε	0.077	0.029	0.017
$\hat{\varepsilon}$	0.020	0.0047	0.0023

Table 7.3: Non-conforming checkerboard structure: energy norm of displacement error

errors appearing in region of high stiffness, and the results of Table 7.3 somewhat hide the dramatic improvement the smoothing procedure can bring to the weakly compatible solution in the very soft rubber subdomains. Let us consider for instance the upper extremety node D of the interface Γ_I^{35} (Fig. 7.7) which turns out to be one of the most challenging point for the interface compatibility. In Table 7.4 we give the relative displacement error for the horizontal and vertical displacement computed at D on the boundary of $\Omega^{(5)}$. From Table 7.4 we see that the accuracy of the smoothed displacements on the boundary of the rubber subdomain $\Omega^{(5)}$ is one or two orders of magnitude better than the precision one can achieve with the non-smoothed solution of the reduced hybrid system.

Clearly, the results reported in Tables 7.3 and 7.4 demonstrate the convergence of the two-field hybrid method and highlight the effectiveness of the smoothing procedure for this problem with heterogeneous and non-conforming substructures. In particular, from Table 7.3 we note that the two-field hybrid method with $p = 1$ and smoothing generates an overall solution as accurate as with $p = 3$ and without smoothing.

Finally, we note that for this substructure assembly problem, the displacement solutions computed by the two-field hybrid method with smoothing are almost identical for both conforming and non-conforming discretizations, with less than 1% relative difference at the control points.

	polyn. degree	$p=1$	$p=2$	$p=3$
non-smoothed	$\frac{\lvert u_x^{(5)^{ref}}(D)-u_x^{(5)}(D)\rvert}{\lvert u_x^{(5)^{ref}}(D)\rvert}$	39.	11.	3.9
	$\frac{\lvert u_y^{(5)^{ref}}(D)-u_y^{(5)}(D)\rvert}{\lvert u_y^{(5)^{ref}}(D)\rvert}$	12.	0.035	0.76
smoothed	$\frac{\lvert u_x^{(5)^{ref}}(D)-\hat{u}_x^{(5)}(D)\rvert}{\lvert \hat{u}_x^{(5)^{ref}}(D)\rvert}$	0.58	0.18	0.080
	$\frac{\lvert u_y^{(5)^{ref}}(D)-\hat{u}_y^{(5)}(D)\rvert}{\lvert \hat{u}_y^{(5)^{ref}}(D)\rvert}$	0.15	0.0088	0.022

Table 7.4: Non-conforming checkerboard structure: relative displacement error at point D of Fig. 7.7

7.7.3 Wing-box structure discretization for aeroelastic analysis

In most realistic aeroelastic computations, the fluid and structure discretizations have non-matching interfaces. For example, in wing flutter problems, the fluid surface grid is typically finer than the mesh of the skin of the structure, at least in the flow direction. One way of handling this situation is to use a staggered procedure for solving the coupled fluid/structure problem [143] and a geometric pairing algorithm [123] for transmitting the flow pressure and structural displacement fields across both sides of the fluid/structure interface. An alternative approach is to design a skin discretization that matches the fluid surface grid, which on one hand simplifies the coupling of the fluid and structure problems, but one the other hand induces non-matching interfaces between some structural components — for example, between the skin and the stiffeners. In the latter case, the two-field hybrid method with smoothing can be employed for solving the structural problem as illustrated in this section.

Since detailed aeroelastic computations are beyond the scope of this work, we focus here on a demonstration problem. For this purpose, we consider the idealization of a wing-box structure depicted in Fig. 7.11. The lower and upper surfaces, defined in this example as two separate substructures Ω^1 and Ω^4, represent the lower and upper skins of a wing. They are clamped at one end, free at the other. The two vertical substructures, Ω^2 and Ω^3, represent two of its stiffeners. The upper and lower skins are subjected to an aerodynamic pressure load whose linear distribution is graphically depicted in Fig. 7.12. Four concentrated forces representing the gravity forces are also applied beneath the stiffeners at the engine locations. All four substructures have the same material properties (Fig. 7.11), and are discretized using three-noded triangular DKT based shell

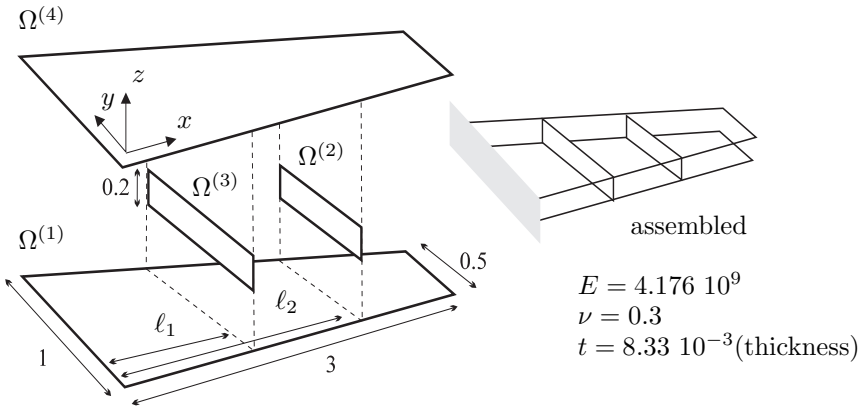


Figure 7.11: Idealization of a wing-box structure: four substructures with non-matching interfaces

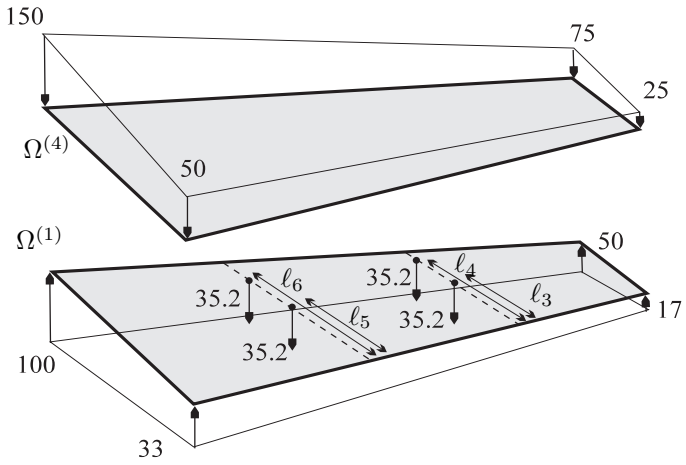


Figure 7.12: Load distribution on the wing-box

elements with six degrees of freedom per node. Three assembly configurations are considered. In configuration A, the model substructures are fully conforming (Fig. 7.13(a)). The skins $\Omega^{(1)}$ and $\Omega^{(4)}$ are modeled by 256 elements and 864 d.o.f., the stiffeners $\Omega^{(2)}$ and $\Omega^{(3)}$ contain 64 elements and 270 d.o.f. each.

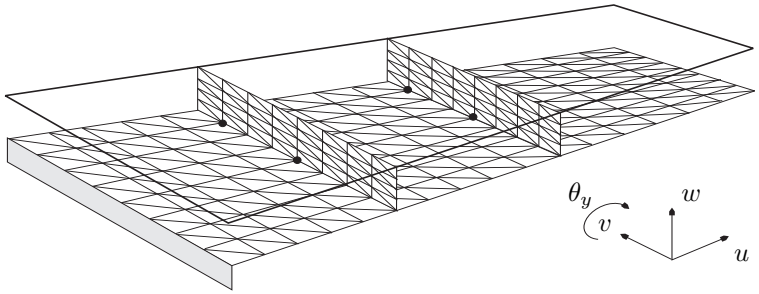
In configuration B, the substructures have non-matching interfaces but intersect only along element edges (Fig. 7.13(b)). The skin meshing is unchanged compared to A, but the stiffener $\Omega^{(2)}$ close to the wing tip contains 40 elements and 180 d.o.f., and $\Omega^{(3)}$ contains 48 elements and 210 d.o.f. In configuration C, the interface edges of the front and back stiffeners *traverse* the elements of the upper and lower skins (Fig. 7.13(c)). In that case, $\Omega^{(2)}$ contains 64 elements and 270 d.o.f., and the other substructures have the same discretization as in configuration B.

The two non-conforming configurations B and C are representative of typical assembly situations. The specific values of the geometric design parameters ℓ_i, $i = 1, \ldots, 6$ (see Fig. 7.11-7.12) are given in Table 7.5 for configurations A, B and C.

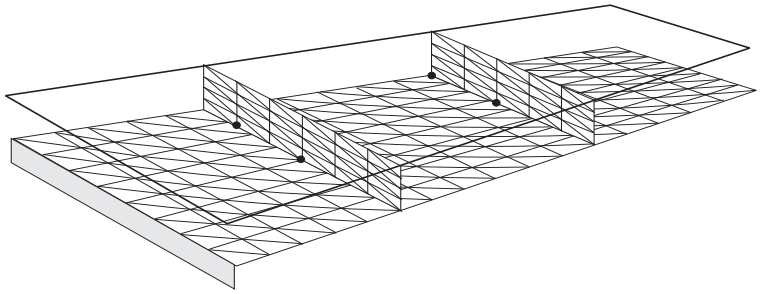
parameter	Configuration A	Configuration B	Configuration C
ℓ_1	0.9375	0.9375	1.0000
ℓ_2	1.8750	1.8750	2.0000
ℓ_3	0.3438	0.4125	0.3333
ℓ_4	0.6016	0.6875	0.5833
ℓ_5	0.4219	0.4218	0.4166
ℓ_6	0.7383	0.7032	0.6944

Table 7.5: Specification of the geometric design variables

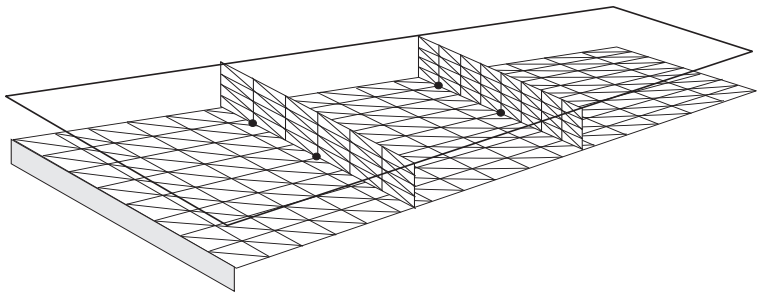
Let u, v, w, and θ_y denote respectively the two in-plane displacements, the transverse displacement, and the tangential rotation of the wing-box structure (tangent to a substructure interface) at a substructure boundary interface node (Fig. 7.13). The compatibilty will be enforced separatly for v and θ_y by considering polynomial functions. The displacements u and w are treated in a special manner for this problem since on the interface boundaries they represent the transverse plate displacements, i.e. u is an out of plane displacement in the stiffeners and a membrane d.o.f. for the skins while w is a out of plane displacement for the skins and a membrane d.o.f. for the stiffeners. Since the kinematic assumption on the DKT element edges is equivalent to the Kirchhoff assumption for beams without shear deformation, the rotations θ_x and θ_z must be considered on the interfaces as the derivatives of the displacements w and u respectively. Therefore, we will



(a) Configuration A (conforming)



(b) Configuration B (non-conforming)



(c) Configuration C (non-conforming)

Figure 7.13: Subdomain meshings for configurations A, B and C

define two independent sets of Lagrange multipliers for the pair (u, θ_z) on one hand and for (w, θ_x) on the other hand. The shape functions associated with these displacements in the DKT elements are the cubic Hermitian polynomials. Taking for instance the interface Γ_I^{12} between the lower skin and the exterior stiffener in the conforming configuration A (Fig. 7.13), we observe that 9 nodes lie on each side of the interface which means that for each pair (u, θ_z) and (w, θ_x) one can use up to 18 Lagrange multipliers for defining the reduced compatibility. Having said (section 5.2) that polynomials can be used only up to a limited degree for conditioning reasons, we will use cosine series as interface force distribution functions so to guarantee good numerical conditioning up to a higher number of interface variables:

$$\lambda(\eta) = \sum_{i=1}^{n_c} a_i \cos\left(\frac{\eta}{H_I}\frac{(i-1)}{\pi}\right) \tag{7.52}$$

where n_c is the number of cosine harmonics used to represent the interface stress distribution. The block of the constraint matrices associated with harmonic i is computed by

$$\boldsymbol{b}_i^{(s)} \begin{bmatrix} u \\ \theta_z \end{bmatrix} \text{or} \begin{bmatrix} w \\ \theta_x \end{bmatrix} = \int_{\Gamma_I} \sum_{elem.} \cos\left(\frac{\eta}{H_I}\frac{(i-1)}{\pi}\right) \boldsymbol{N}_{Herm}^{(s)} d\Gamma_I \begin{bmatrix} u \\ \theta_z \end{bmatrix} \text{or} \begin{bmatrix} w \\ \theta_x \end{bmatrix} \tag{7.53}$$

where $\boldsymbol{N}_{Herm}^{(s)}$ are the Hermitian polynomials on the DKT edges. The formulation (7.53) combined with a constraint orthogonalization technique (Section 5.3) ensures a good numerical conditioning for a number of interface parameters up to 18.

If we now consider the configuration C where the interfaces run across the elements of the skins, the interface displacements on the side of the skins are dependent on all the nodal displacements relative to the elements traversed by the interface. The displacement shape functions within those DKT skin elements are not exactly Hermitian polynomials since the Kirchhoff kinematic assumptions are exact only on the element edges [14, 129]: the displacements shape functions for the interior of DKT elements are independently defined for the translations and rotations although they are implicitly linked by the edge kinematical conditions. To simplify the computation of the constraining matrices for configuration C we will assume that the the Kirchhoff assumptions also hold inside the DKT elements. The membrane displacement v in the DKT and the rotation θ_y are assumed to be linearly variant on the element.

Expressing the displacement shape functions and performing the integration of the constraint matrices (7.53) is not an easy task. We used the symbolic computation package MAPLE to analytically compute the constraint matrices and to generate the corresponding routines.

In this example, the problem heterogeneities are not induced by different substructure material properties, but by different substructure geometries, orientations, and mesh resolutions. When applying our generalized smoothing procedure (7.23,7.39), we will first define one smoothing parameter per interface and per subdomain as in the formulation presented in this chapter. Then, observing that the interface stiffness might be different depending on the displacement direction, we will apply the smoothing technique with 4 smoothing parameters per interface and per edge: on every edge $\Gamma^{sr,j}$, we define $\beta_v^{(s),j}$, $\beta_{\theta_y}^{(s),j}$, $\beta_{u,\theta_z}^{(s),j}$, and $\beta_{w,\theta_x}^{(s),j}$ associated respectively with the displacement v, the tangential rotation θ_y and the plate displacements (u, θ_z) and (w, θ_x). When only one smoothing parameter is defined per interface, the stiffness coarse grid comprises only 4 independent variables and when 4 smoothing parameters are defined per interface, the dimension of the coarse grid is equal to 16 which remains small compared to the number of interface d.o.f.

To illustrate the results we have obtained, we give the displacement curves at the interface Γ_I^{12} between the lower skin and the exterior stiffener. Reminding that two point loads are applied on this interface, the reader easily understands that this interface is one of the most critical ones in terms of compatibility enforcement since the real stress distribution along that edge is rather irregular.

For the conforming configuration A the number of interface Lagrange multipliers used for defining the reduced interface problem is given in Table 7.6 and the interface displacements are

plotted in Fig. 7.14: we have plotted separately the non-smoothed solution and the solution post-processed with a smoothing procedure based on one parameter or 4 parameters per interface. Remember that the reference solution corresponds to the fully assembled solution in this case where the interfaces are conforming.

Interface	type of d.o.f.	n_{λ^p}	n_{λ^q}
All Γ_I^{sr}	v	2	9
	θ_y	3	9
	(u, θ_z)	5	18
	(w, θ_x)	5	18

Table 7.6: Number of Lagrange multipliers for the reduced and the reference aeroelastic problem (conforming configuration A)

From Fig. 7.14 we see that the smoothing procedure defines a good approximate solution by attributing the right weights to the weakly compatible solution an the interface. If we consider for instance the horizontal displacements u and v, we see on Fig. 7.14 that the weakly compatible solution on the boundary of the skin $\Omega^{(1)}$ is very close to the exact solution whereas in the stiffener which is much softer for u the weakly compatible displacement is erroneous. Considering now the vertical displacement w, we see that the situation is exactly opposite: for the vertical displacement, the stiffeners are stiff and the skins are soft. Therefore the weak solution w in the stiffener is now close to the reference solution (Fig. 7.14). For the rotation θ_y tangent to the interface, the skins and the stiffener offer the same interface stiffness so that the non-smoothed solutions are roughly equally apart from the assembled solution on both sides of the interface.

When only one smoothing parameter is defined per interface and per subdomain, the smoothing procedure will define smoothing parameters so as to minimize the energy associated with the interface residual. It is clear from the discussion above that the smoothing should give more weight to the skin solution when u is considered and to the stiffener solution for when w is concerned. When only one smoothing parameter is defined for the interface, the energy principle inherent to the smoothing technique will give similar weights to the skin and stiffener solution to minimize the error globally. Table 7.7 gives the values of the smoothing parameters in the skin substructures: we see that $\beta^{(1)} = 0.66$ and thus $\beta^{(2)} = 0.44$ on Γ_I^{12}. The resulting smoothed solution is not satisfactory neither for the vertical displacement nor for the horizontal displacements (Fig. 7.14).

If we now define one smoothing parameter per displacement direction, we see from Table 7.7 that the smoothing parameters are optimal: for the horizontal displacement u in the skin $\beta^{(1)}_{u,\theta_z} = 0.99$, for the vertical displacement w in the skin, $\beta^{(1)}_{w,\theta_x} = 0.04$ and for the tangential rotation θ_y we find $\beta_{\theta_y} = 0.64$. Fig. 7.14 clearly shows that when 4 smoothing parameters are used per interface, the smoothing process defines optimum smoothed solutions which are very close to the reference solution in each direction. Let us remind that this accurate solution was found for a total number of Lagrange multipliers equal to 60 (see Table 7.6) while the number of exact compatibility conditions is 216.

We now consider configuration B where the meshing of the skins is unchanged and the meshing of the stiffeners is coarsened (Fig. 7.13). The interface are no longer conforming although the stiffeners and skins still intersect on the edges of the DKT elements. The number of Lagrange multipliers defined on the interfaces is given in Table 7.8. Again we show the displacement results obtained on Γ_I^{12} when no smoothing is applied, when smoothing is performed with one parameter per interface and when 4 smoothing parameters per interface are defined (Fig. 7.15).

Since configuration B is non-conforming, the reference solutions on the interfaces are not exactly compatible (see for instance the tangental rotation θ_y in Fig. 7.15). Comparing the results for configuration B (Fig. 7.15) to the results of conforming configuration A (Fig. 7.14) we observe that the reference solution are not exactly equal due to the fact that the engine gravity loads are placed at different nodal locations (Fig. 7.13). Nevertheless, because the results of

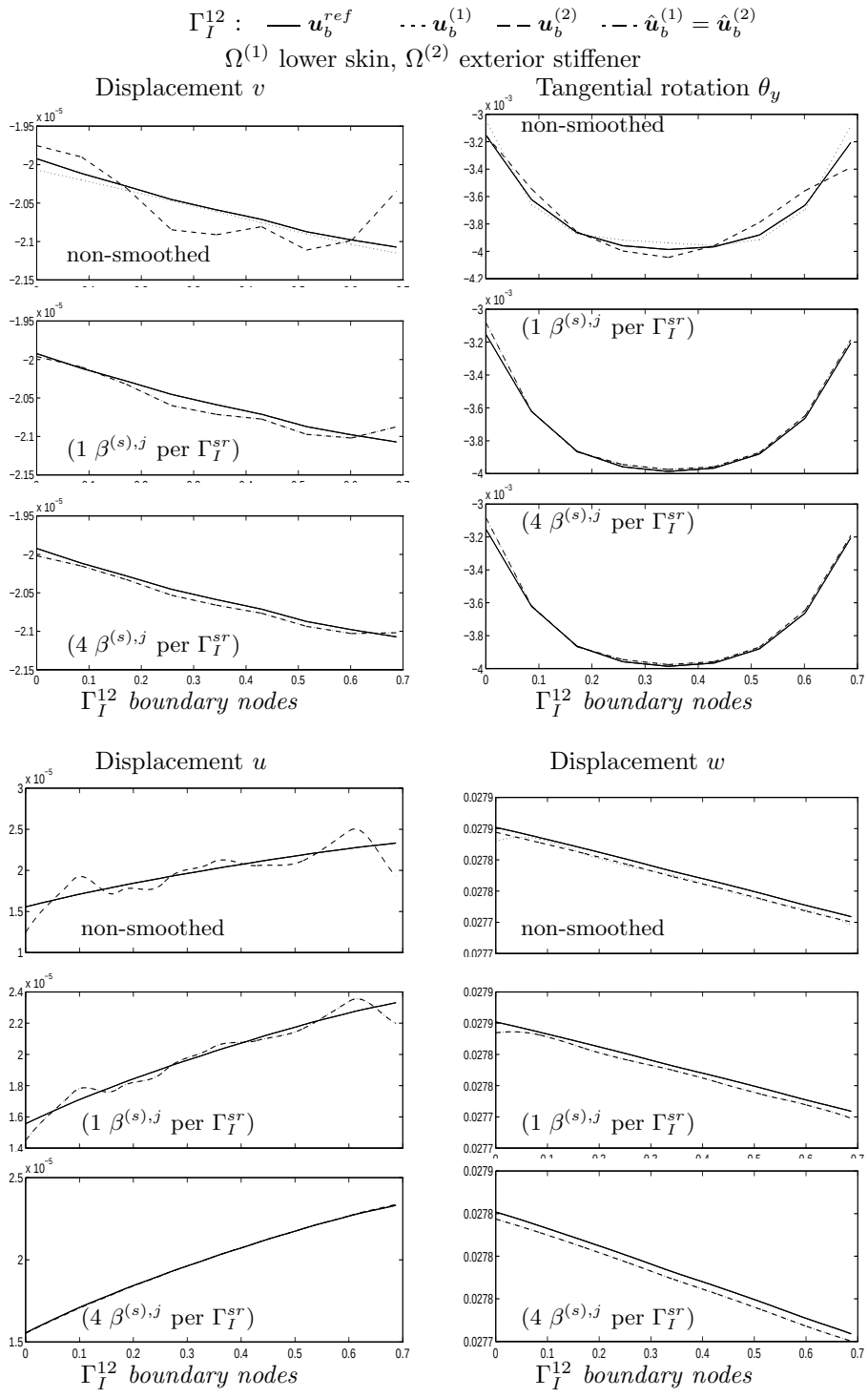


Figure 7.14: Displacements on Γ_I^{12} for configuration A

Interface	1 β per Γ_I^{sr}	Configuration			4 β per Γ_I^{sr}	Configuration		
		A	B	C		A	B	C
Γ_I^{12}	$\beta^{(1)}$	0.66	0.93	0.12	$\beta_v^{(1)}$	0.84	0.81	0.99
					$\beta_{\theta_y}^{(1)}$	0.64	0.75	0.96
					$\beta_{u,\theta_z}^{(1)}$	0.99	0.99	0.99
					$\beta_{w,\theta_x}^{(1)}$	0.04	0.02	0.07
Γ_I^{24}	$\beta^{(4)}$	0.91	0.62	0.21	$\beta_v^{(4)}$	0.80	0.80	0.99
					$\beta_{\theta_y}^{(4)}$	0.59	0.63	0.97
					$\beta_{u,\theta_z}^{(4)}$	0.99	0.99	0.99
					$\beta_{w,\theta_x}^{(4)}$	0.04	0.02	0.06
Γ_I^{13}	$\beta^{(1)}$	0.81	0.97	0.96	$\beta_v^{(1)}$	0.78	0.73	0.99
					$\beta_{\theta_y}^{(1)}$	0.62	0.72	0.75
					$\beta_{u,\theta_z}^{(1)}$	0.99	0.99	0.99
					$\beta_{w,\theta_x}^{(1)}$	0.01	0.01	0.07
Γ_I^{34}	$\beta^{(4)}$	0.83	0.96	0.97	$\beta_v^{(4)}$	0.75	0.73	0.99
					$\beta_{\theta_y}^{(4)}$	0.65	0.73	0.64
					$\beta_{u,\theta_z}^{(4)}$	0.99	0.99	0.99
					$\beta_{w,\theta_x}^{(4)}$	0.01	0.01	0.11

Table 7.7: smoothing coefficients in skin panels

Interface	type of d.o.f.	n_{λ^p}	n_{λ^q}
Γ_I^{12}	v	2	6
and	θ_y	3	6
Γ_I^{24}	(u, θ_z)	5	12
	(w, θ_x)	5	12
Γ_I^{13}	v	2	7
and	θ_y	3	7
Γ_I^{34}	(u, θ_z)	6	14
	(w, θ_x)	6	14

Table 7.8: Number of Lagrange multipliers for the reduced and the reference aeroelastic problem (non-conforming configuration B)

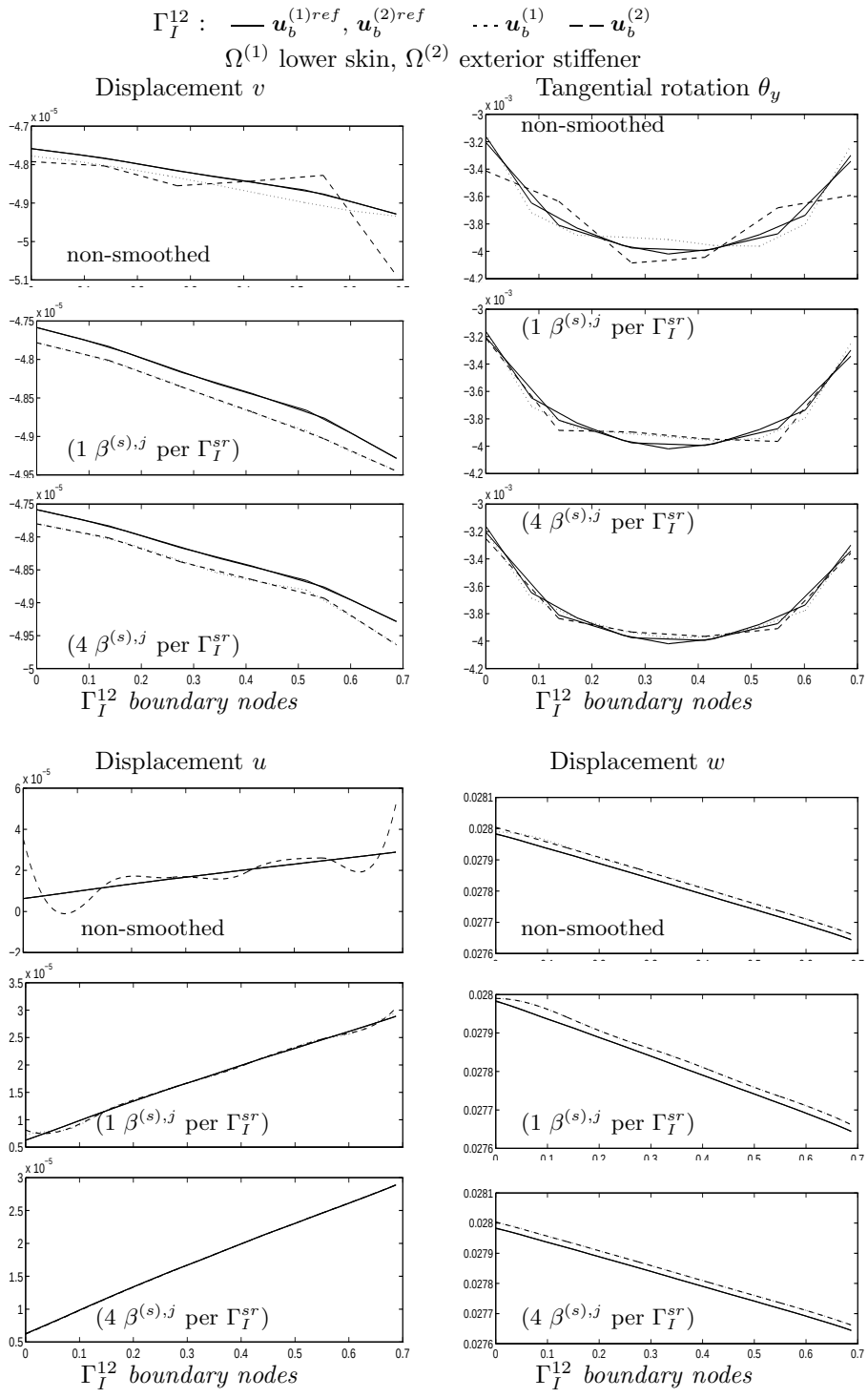


Figure 7.15: Displacements on Γ_I^{12} for configuration B

configurations A and B are very similar and because the interface compatibility is nearly exact (Fig. 7.15), we have good confidence that the reference solutions obtained for the non-conforming mesh B are accurate. We also observe that the so-called reference solution is clearly shown to have an almost zero jump across the substructure interfaces. This is not a trivial result given that the substructures of this problem have non-matching interfaces.

Analyzing now the effect of smoothing on the weakly compatible solution, the discussion made for the configuration A still applies: from Fig. 7.15 and from Table 7.7 giving the values of the smoothing parameters, we observe that the smoothing procedure results in fairly bad displacement predictions when only one smoothing coefficient per interface is used but the algorithm yields accurate solutions when one smoothing parameter is associated with each interface displacement direction. In the latter case, the smoothed solution is essentially defined by the skin when the horizontal displacements u and v are considered and by the stiffener when the vertical displacement w is concerned. Once more an accurate smoothed solution was found with a fairly small number of Lagrange multipliers: the reduced number of Lagrange multipliers n_{λ^p} is equal to 64 which is small considering that the reference stronger compatibility is defined for $n_{\lambda^q} = 156$ (see Table 7.8).

Finally we perform the static analysis on configuration C. The intersection between skin and stiffeners are highly non-conforming considering that the stiffener boundaries traverse the DKT elements of the skins (Fig. 7.13). The number of Lagrange multipliers for setting up the reduced interface problem and the reference system is given in Table 7.9.

Interface	type of d.o.f.	n_{λ^p}	n_{λ^q}
Γ_I^{12}	v	2	9
and	θ_y	4	9
Γ_I^{24}	(u, θ_z)	8	18
	(w, θ_x)	8	18
Γ_I^{13}	v	2	7
and	θ_y	3	7
Γ_I^{34}	(u, θ_z)	7	14
	(w, θ_x)	7	14

Table 7.9: Number of Lagrange multipliers for the reduced and the reference aeroelastic problem (non-conforming configuration C)

The displacements computed whith and whithout smoothing on Γ_I^{12} are plotted in Fig. 7.16. The displacement curves show that spurious wriggles appear for the tangential rotation θ_y on the interface in the lower skin panel. These wrinkles can be explained by the assumption we made for the displacement shape functions for the interior of the DKT elements: the interface displacements on the skins are interior displacements for the DKT elements and we have made the approximation that the corresponding displacement shape functions are cubic Hermitian polynomials as on the DKT edges. This approximation appears to perturb the formulation of the interface compatibility constraints. Although the quality of the solutions requires that the results be considered with caution, the smoothing procedure combines the weakly compatible interface displacements in the best possible manner.

To conclude this example and to show once more the efficiency of our proposed smoothing procedure, we give in Table 7.10 the energy norms of the displacement errors ε defined by (5.32) for the non-smoothed solution and the energy norm $\hat{\varepsilon}$ defined by (5.33) when the solution is smoothed.

Table 7.10 stresses once more the important lessons to be drawn from this wing-box example:

- The Rayleigh-Ritz based smoothing procedure developped in this work appears to be very efficient: when assembling plates or shells, it allows to compute very accurate solutions with

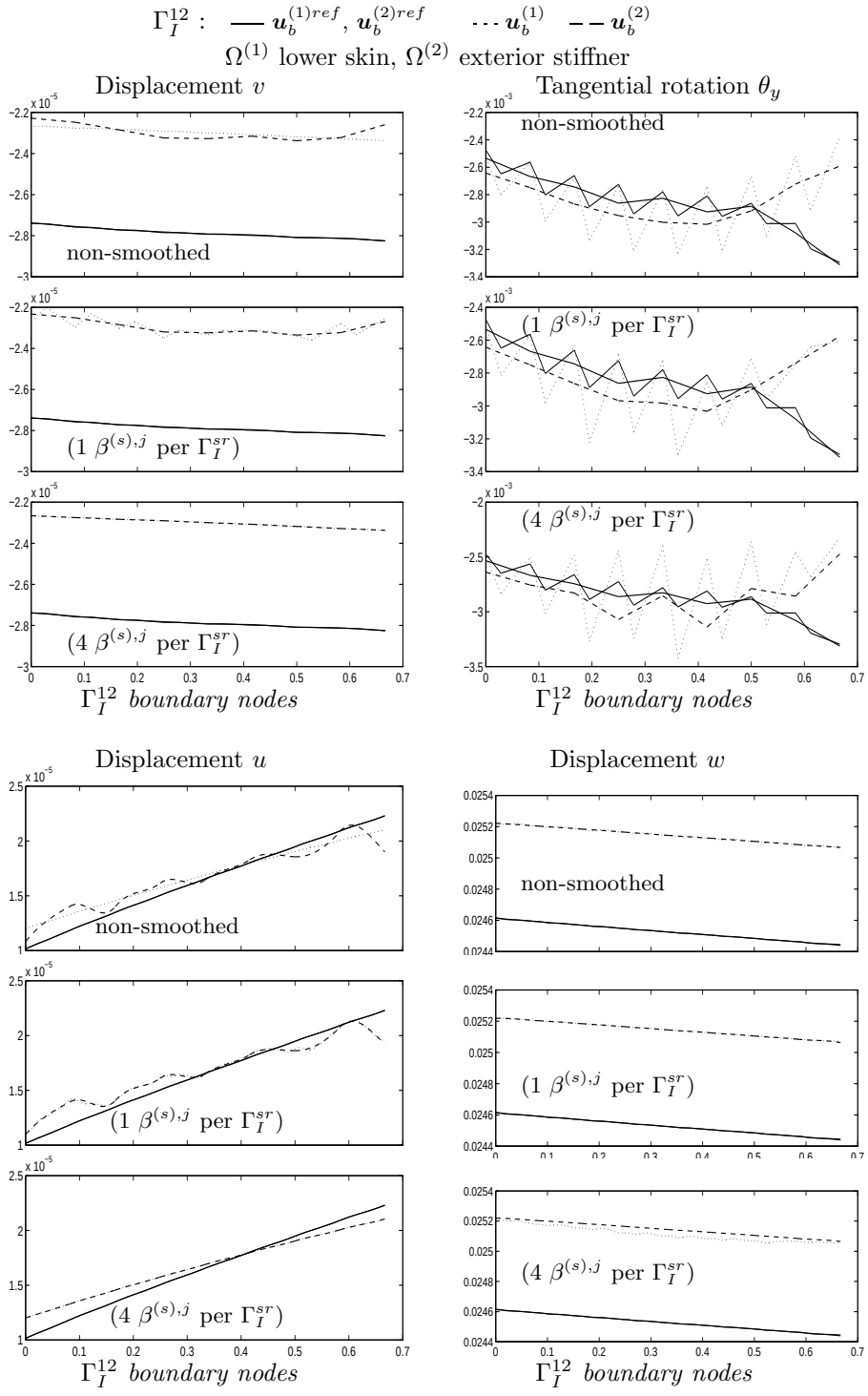


Figure 7.16: Displacements on Γ_I^{12} for configuration C

	relative energy norm		
Configuration	ε no smoothing	$\hat{\varepsilon}\|_1$ smoothing with 1 $\beta^{(s),j}$	$\hat{\varepsilon}\|_4$ smoothing with 4 $\beta^{(s),j}$
A	0.0284	0.1182	0.0051
B	0.0332	0.1438	0.0186
C	0.1619	0.3343	0.2529

Table 7.10: Relative energy norms for the aeroelastic problem

a number of interface variables of only a fraction of the number of Lagrange multipliers defining the reference compatibility.

- In the case when plate-like substructures with different orientations must be assembled, one smoothing parameter must be associated with each displacement direction. The smoothing procedure then takes account of the difference of stiffnesses per subdomain and per direction.
- When an interface traverses a DKT element, special care should be given to the correct estimation of the shape functions for the interior of the element. This might be a difficult issue in practice. Hybrid Reissner-Mindlin plate elements should be easier to connect in this case.

7.8 Smoothing vs. higher-order Lagrange multipliers

Solving a substructure assembly problem by a two-field hybrid method with low-order polynomial or piece-wise polynomial Lagrange multipliers and smoothing the computed solution with the Rayleigh-Ritz based procedure proposed in this work is an alternative to solving the same problem with the same two-field hybrid method but using higher-order Lagrange multipliers. The merits of this alternative are as follows.

- When the two-field hybrid method is used with n_{λ^p}-th order Lagrange multipliers and the corresponding solution is post-processed with a n_{λ^q}-th order smoothing procedure where $n_{\lambda^q} > n_{\lambda^p}$ — for example, $n_{\lambda^q} = 3n_{\lambda^p}$— the size of the interface problem associated with Eqs. (5.2a-5.2b)
$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s),p} \boldsymbol{K}^{(s)^{-1}} \boldsymbol{B}^{(s),p^T} \boldsymbol{\lambda} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s),p} \boldsymbol{K}^{(s)^{-1}} \boldsymbol{f}^{(s)} \tag{7.54}$$
is much smaller than in the case where n_{λ^q} Lagrange multipliers are used — for example, 3 times smaller when $n_{\lambda^q} = 3n_{\lambda^p}$. Of course, this size is also much smaller than that of the interface problem generated by the mortar method (piece-wise linear Lagrange multipliers on all edges of the interface elements). This computational efficiency is particularly interesting when the two-field hybrid method is used as a mechanism for designing a model reduction algorithm [47, 77].
- In general, one does not know a priori the lowest polynomial degree n_{λ^p} for which a highly accurate solution is obtained. Opting for an adaptive strategy such as that described in [77] for finding the optimal degree n_{λ^p} requires an error criterion and an iterative process (see next section). On the other hand, selecting arbitrarily n_{λ^p}, for example $n_{\lambda^p} = 3$, and using a n_{λ^q}-th order smoothing procedure produces an excellent solution as demonstrated in the applications of Sections 6.3 and 7.7 while bypassing the need for an a posteriori error criterion and bypassing iterations.

Chapter 8

Accuracy improvement of the reduced two-field solution

8.1 Iterative refinement procedure for Lagrange multipliers

In the previous chapter we have shown how to obtain accurate results with a low number of interface Lagrange multipliers and by applying an efficient smoothing post-processing. When applying these methods to practical problems for a given reduced number n_{λ^p} of Lagrange multipliers, the accuracy obtained may not be satisfactory. As proposed in [77], an iterative procedure can then be employed in order to improve the quality of the solution up to a given accuracy.

The iterative procedure presented in [77] increases the number of Lagrange multipliers (i.e. the polynomial degree or the number of subintervals for piecewise low order polynomials) until the convergence of a certain measure of the solution error is lower than a given tolerance ϵ. In [77], the convergence criterion was taken as

$$\left\| \hat{\boldsymbol{u}}_j^{(s),k+1} - \hat{\boldsymbol{u}}_j^{(s),k} \right\|_\infty < \epsilon \left\| \hat{\boldsymbol{u}}_j^{(s),k} \right\|_\infty \qquad , s = 1, \dots \ Ns \ , \ j =, \dots d \tag{8.1}$$

where d is the number of d.o.f. per node. This criterion indicates that when increasing the number of interface unknowns from iteration k to $k+1$, the maximum relative improvement should be smaller than a given tolerance ϵ, and this should be true for all displacement directions j. Other convergence criteria could be devised by monitoring for instance the equilibrium residual $\boldsymbol{r}$ after smoothing (6.12).

When increasing the number of Lagrange multipliers n_{λ^p}, a new line of the constraints matrices $\boldsymbol{B}^{(s)}$ must be computed. Updating $\boldsymbol{B}^{(s)}$ requires only to perform the integral of the new interface distribution function times the boundary shape functions. For the interface dual operator $\boldsymbol{F}_I^p = \boldsymbol{B}^p \boldsymbol{K}_d^+ \boldsymbol{B}^{p^T}$, adding compatibility constraint leads to computing new terms relative to the supplementary interface unknowns, which implies the solution of new local Neumann problems, i.e. one forward and backward substitution if the local operators are triangularized once for all. The block of $\boldsymbol{F}_I^p$ pertaining to the previously defined Lagrange multipliers remains unchanged. The solution of the dual interface problem then requires a new elimination on the updated $\boldsymbol{F}_I^p$ the cost of which should be marginal as long as the number of interface variables n_{λ^p} remains small compared to the number of interface d.o.f.

This iterative refinement of the dual space of the interconnection stresses was applied with success on structures such as a solid rocket booster [77] and it was shown to be very efficient as a concurrent algorithm for solving large models on multi-processor computers.

All examples published in [47, 60, 77, 149, 151] were solved efficiently with a low number of interface variables and thus demonstrate the effectiveness of combining a reduced two-field hybrid method, a smoothing post-processing and an iterative accuracy improvement for solving a

substructured problem concurrently. Nevertheless, as highlighted by the truss frame example (Fig. 5.11) considered in sections 5.6 and 6.3.1, the efficiency of this new solution algorithm is questionable when the solution on the interface is irregular (see e.g. Fig. 5.13).

When the interface solutions are not smooth, we violate the very fundamental assumption on which the representation of the interface stress distribution by a limited set of a simple functions is based [47]. It does not mean that the theory developed for the reduced two-field hybrid method does not apply in the case of irregular interface solutions, but the number of stress distribution functions necessary to correctly approximate the interface stress field is so high that it entails no significant reduction of the interface problem.

To illustrate this point, let us apply an iterative solution scheme to the truss frame example of Fig. 5.11 when smoothing with two sets of smoothing parameters, one for the horizontal displacements and one for the vertical displacements. The order p of the interface polynomials is increased from $p = 0$ to $p = 10$. Note that the ultimate case $p = 11$ can not be computed because in that case the constraint matrices become extremely ill-conditioned. However, for $p = 11$, that is for $n_{\lambda^p} = 24$, the dual space is completely represented and we would find the exact solution of the fully assembled system. On Fig. 8.1, we give three different measures of the convergence of the smoothed solution when the polynomial degree of the Lagrange multipliers is increased: in Fig. 8.1(a) we plot the maximum relative variation for the horizontal and vertical displacements (see (8.1)) and in Fig. 8.1(b) we plot the relative energy of the displacement error (5.33) for the smoothed solution and (5.32) for the non-smoothed solution. In Fig. 8.1(c) we give the interface residual force after smoothing:

$$\frac{\|\boldsymbol{r}_x\|_2}{\|\boldsymbol{f}\|_2} \quad \text{and} \quad \frac{\|\boldsymbol{r}_y\|_2}{\|\boldsymbol{f}\|_2} \tag{8.2}$$

Fig. 8.1(b) comparing the non-smoothed and the smoothed solution shows once more the better accuracy associated with the smoothed solution. Nevertheless, from Fig. 8.1, it is clear that even by substantially increasing the polynomial degree of the interface stress distribution, the accuracy of the solution can not be significantly improved. Let us recall that the reason for this is that the exact stress distribution function on the discrete interface (see Fig. 5.13) can not by any means be represented by simple functions. To tackle the problem of reducing the interface problem when the interface solution is highly irregular, we will propose an alternative iteration scheme.

8.2 A new series of interface loads: the primal-dual iterations

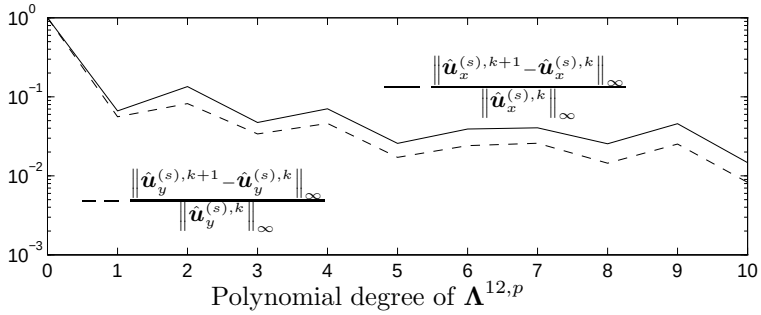
To be able to correctly represent a non-smooth interface stress distribution such as in a truss frame, new trial functions for setting up the Lagrange multiplier field must be related to the very mechanical behavior of the interface. In other words, the stress functions input in the two-field variational principle should be linked to actual interface stress functions of the structure. The best candidate for defining physically meaningful trial functions is clearly the interface stress associated with a smoothed solution.

Let us begin with solving the local Neumann problems with zero interface stress: assuming for the sake of simplicity that the local operators are regular, we write

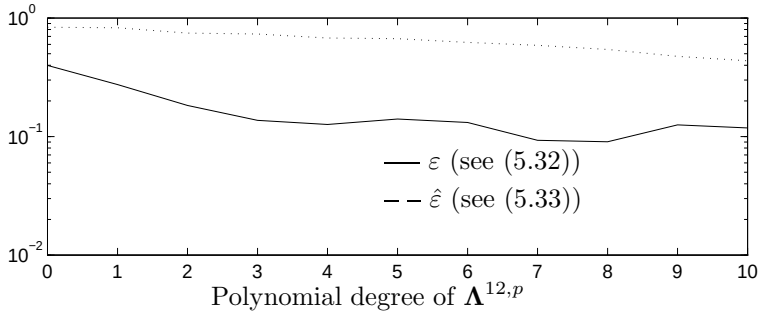
$$\boldsymbol{u}^{(s),0} = {\boldsymbol{K}^{(s)}}^{-1} \boldsymbol{f}^{(s)} \tag{8.3}$$

The interface compatibility of this initial solution is totally erroneous since no compatibility constraint has been enforced so far. The interface compatibility error, i.e. the displacement jump, is given by

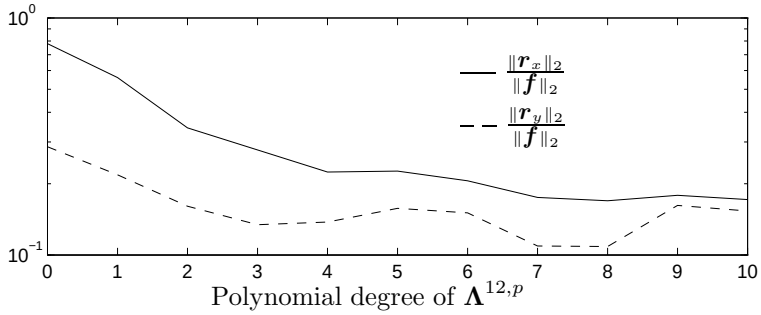
$$\boldsymbol{r}^0 \quad = \quad \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s),0} \tag{8.4}$$



(a) Relative norm of displacement variation during iteration



(b) Relative energy norm of displacement error



(c) Relative force residual

Figure 8.1: Iterative improvement for the truss-frame solution

$$= \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} {\boldsymbol{K}^{(s)}}^{-1} \boldsymbol{f}^{(s)} = \boldsymbol{d} \tag{8.5}$$

where $\boldsymbol{B}^{(s)}$ are the constraints matrices of dimension $n_{\lambda^q} \times n^{(s)}$ defining the reference compatibility we want to achieve iteratively with a number of stress distribution functions $n_{\lambda^p} << n_{\lambda^q}$.

Applying then a smoothing scheme such as defined by (7.23,7.39) to this solution we obtain the smoothed iterate $\hat{\boldsymbol{u}}^{(s),0}$ such that

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \hat{\boldsymbol{u}}^{(s),0} = \boldsymbol{0} \tag{8.6}$$

and we can define the associated domain-wise interface loads

$$\Delta \boldsymbol{f}^{(s),0} = \left(\boldsymbol{K}_{bi}^{(s)} \hat{\boldsymbol{u}}_i^{(s),0} + \boldsymbol{K}_{bb}^{(s)} \hat{\boldsymbol{u}}_b^{(s),0} \right) - \boldsymbol{f}_b^{(s)} \tag{8.7}$$

The interface load $\Delta \boldsymbol{f}^{(s),0}$ represents the interconnection forces associated with the smoothed solution. It can be expressed in terms of displacement corrections $\Delta \boldsymbol{u}^{(s),0} = \hat{\boldsymbol{u}}^{(s),0} - \boldsymbol{u}^{(s),0}$: considering that $\hat{\boldsymbol{u}}^{(s),0}$ satisfies the internal equilibrium 7.23b and that $\boldsymbol{u}^{(s),0}$ verifies the local equilibrium (8.3), we find

$$\Delta \boldsymbol{f}^{(s),0} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}^{(s),0} \tag{8.8}$$

$\boldsymbol{S}_{bb}^{(s)}$ being as before the Schur complement of the interface. On an interface edge Γ_I^j of multiplicity m_j and connecting the subdomains named $s, r, q \ldots$, we thus define m_j interface loads, one on each side of the interface. These interface loads are in fact the reaction forces arising from the solution of the Dirichlet problem inherent to the smoothing procedure and is a natural candidate to represent the interconnection stresses. One dilemma remains though: the interface loads do not equilibrate one another unless the exact solution is found and therefore the question is how to define a unique Lagrange multiplier field between two subdomains $\Omega^{(s)}$ and $\Omega^{(r)}$ connecting on the edge Γ_I^j. From a mechanical view point, one should give more weight to the interface force defined on the softer side and therefore we will define an average interconnection force $\boldsymbol{\lambda}^{sr,j^0}$ in a way exactly dual to the procedure for defining the interface displacement correction in the smoothing scheme (7.29):

$$\boldsymbol{\lambda}^{sr,j^0} = \beta^{(r),j} {\boldsymbol{B}^{(s,-),j}}^T \Delta \boldsymbol{f}^{(s),0} - \beta^{(s),j} {\boldsymbol{B}^{(r,-),j}}^T \Delta \boldsymbol{f}^{(r),0} \tag{8.9}$$

where $\boldsymbol{B}^{(s,-),j}$ is defined in (7.13) and for matching meshes the latter expression simplifies to

$$\boldsymbol{\lambda}^{sr,j^0} = \beta^{(r),j} \ \boldsymbol{b}^{(s,-),j} \Delta \boldsymbol{f}^{(s),0} + \beta^{(s),j} \ \boldsymbol{b}^{(r,-),j} \Delta \boldsymbol{f}^{(r),0} \tag{8.10}$$

The approximations $\boldsymbol{\lambda}^{sr,j^0}$ on each interface edge define a first approximate interface nodal load and we write

$$\boldsymbol{\Lambda}^0 = \boldsymbol{\lambda}^0 \tag{8.11}$$

$$\boldsymbol{\lambda} \simeq \boldsymbol{\Lambda}^0 \eta^0 \tag{8.12}$$

$$\boldsymbol{B}^1 = {\boldsymbol{\Lambda}^0}^T \boldsymbol{B} \tag{8.13}$$

where η^0 is the amplitude of the Lagrange multiplier subspace $\boldsymbol{\Lambda}^0$ and $\boldsymbol{B}^1$ is a reduced constraint matrix.

The iteration for $k = 1, ...$ then goes on as follows:

- Find the solution of the reduced two-field hybrid system relative to $\boldsymbol{B}^k$

$$\boldsymbol{u}^{(s),k} = {\boldsymbol{K}^{(s)}}^{-1} (\boldsymbol{f}^{(s)} - {\boldsymbol{B}^{(s),k}}^T \boldsymbol{\eta}^k) \tag{8.14}$$

$$\boldsymbol{F}_I^k \boldsymbol{\eta}^k = \boldsymbol{d}^k \tag{8.15}$$

$$\boldsymbol{\lambda} = \boldsymbol{\Lambda}^k \boldsymbol{\eta}^k \tag{8.16}$$

where

$$\boldsymbol{F}_I^k = \boldsymbol{\Lambda}^{k^T} \boldsymbol{F}_I \boldsymbol{\Lambda}^k \tag{8.17}$$

$$= \sum_{s=1}^{N_s} \boldsymbol{B}^{(s),k} \boldsymbol{K}^{(s),-1} \boldsymbol{B}^{(s),k^T} \tag{8.18}$$

$$\boldsymbol{d}^k = \boldsymbol{\Lambda}^{k^T} \boldsymbol{d} \tag{8.19}$$

$$= \sum_{s=1}^{N_s} \boldsymbol{B}^{(s),k} \boldsymbol{K}^{(s)^{-1}} \boldsymbol{f}^{(s)} \tag{8.20}$$

- Smooth the resulting weakly compatible solution to define $\hat{\boldsymbol{u}}^{(s),k}$ and compute a correction of the Lagrange multipliers

$$\Delta \boldsymbol{f}^{(s),k} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}^{(s),k} \tag{8.21}$$

$$\boldsymbol{\lambda}^{sr,j^k} = \beta^{(r),j} \boldsymbol{B}^{(s,-),j^T} \Delta \boldsymbol{f}^{(s),k} - \beta^{(s),j} \boldsymbol{B}^{(r,-),j^T} \Delta \boldsymbol{f}^{(r),k} \tag{8.22}$$

- Update the constraint matrices

$$\boldsymbol{\Lambda}^{k+1} = [\boldsymbol{\Lambda}^k \boldsymbol{\lambda}^k] \tag{8.23}$$

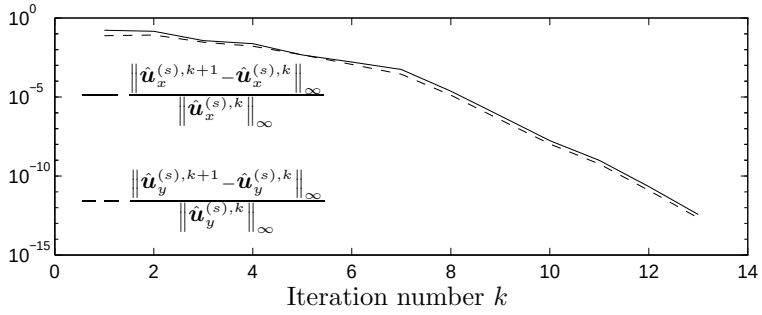
$$\boldsymbol{B}^{k+1} = \boldsymbol{\Lambda}^{k+1^T} \boldsymbol{B} \tag{8.24}$$

$$= \begin{bmatrix} \boldsymbol{B}^k \\ \boldsymbol{\lambda}^{k^T} \boldsymbol{B} \end{bmatrix} \tag{8.25}$$

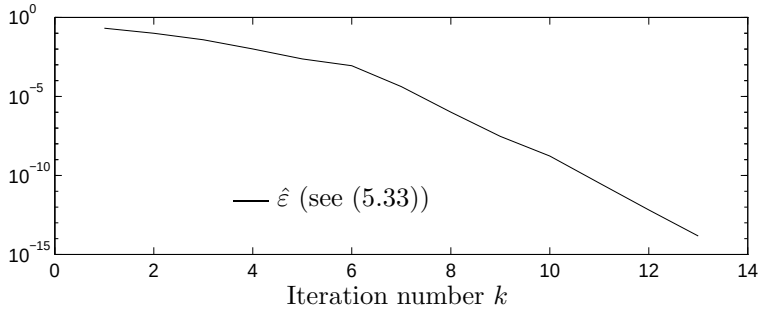
This iteration scheme is similar to the iteration procedure discussed in section 8.1 for refining the Lagrange multipliers by increasing the polynomial degree. The main difference is that now the interface loads introduced in the successive approximate constraint matrices $\boldsymbol{B}^{(s),k}$ are no longer defined with simple mathematical functions, but are derived from the interface force residuals at every iteration. We will call this iteration scheme the *primal-dual* iteration: at every iteration a reduced dual interface problem is solved and it is followed by a smoothing which involves the solution of a primal problem where exact compatibility and weak equilibrium is enforced. Since the smoothing procedure was shown to be highly efficient in defining a good approximate solution and because the subspace for the Lagrange multipliers is built on actual interface loads, the iterative scheme presented here should be efficient even when the interface solution is irregular.

To test the primal-dual iteration, let us consider again the truss frame of Fig. 5.11. In the previous section 8.1, we showed that no simple reduced series of functions could correctly represent the irregular connecting loads on the interface of the truss (Fig. 8.1). We now apply the iterative scheme described above. The convergence can be measured as previously be the relative norm of the maximum displacement corrections (8.1), by the force residual at the end of each iterate (8.2) or by the energy norm $\hat{\varepsilon}$ of the displacement error compared to the reference solution. We will also monitor the compatibility residual $\boldsymbol{r}^k$, i.e. the displacement gap before smoothing. The results are plotted in Fig. 8.2.

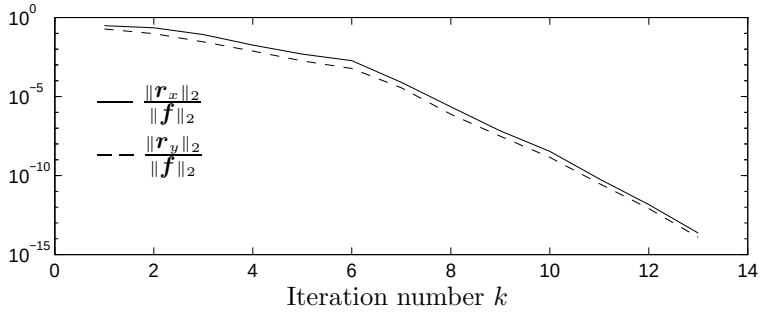
Fig. 8.2 indicates that after 13 iterations the iteration must be stopped because machine precision has been attained. In order to compare the convergence to the convergence of the polynomial iterations in Fig. 8.1, let us note that for the polynomial iteration the total number of interface unknowns is given by $n_{\lambda^p} = 2(p+1)$ whereas for the present iteration method the number of columns in $\boldsymbol{B}^{(s),k}$ is equal to the iteration number k. We thus conclude that the ultimate precision is achieved with the primal-dual iteration for $n_{\lambda^p} = 13$ while very poor accuracy was obtained with the polynomial iteration even for $p = 10$ that is for $n_{\lambda^p} = 22$. In fact, for the truss frame, the maximum accuracy achieved with an iterative polynomial scheme is obtained by the primal-dual iteration already after two iterations!



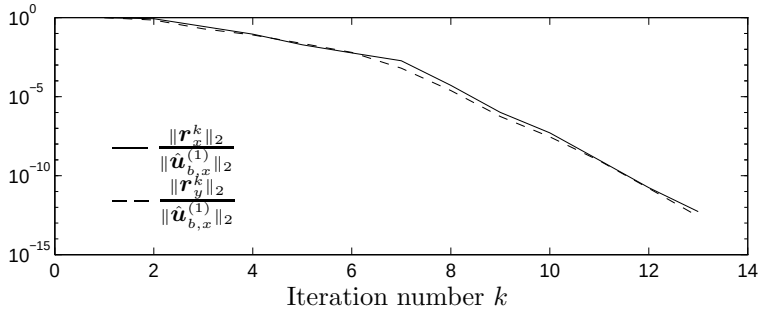
(a) Relative norm of displacement variation during iteration



(b) Relative energy norm of displacement error



(c) Relative force residual



(d) Compatibility error before smoothing

Figure 8.2: Primal-dual iteration for the truss-frame solution

8.3 Towards a Conjugate Gradient like algorithm

The polynomial iterations of Section 8.1 and the primal-dual iteration method can be improved from a numerical efficiency view point if we note that unnecessary computation is performed every time the dual interface operator is factored anew during an iteration k. Reducing the computational effort for solving the interface problem is essential noting that this step involves data exchange between the different substructures and is thus the main point where communications are needed between processors when the substructure operators are split among several processors.

If $\boldsymbol{F}_I^k$ is factored, it is clear that the triangularization of the next dual operator $\boldsymbol{F}_I^{k+1}$ could be obtained by making use of the factorization of $\boldsymbol{F}_I^k$. An alternative but equivalent method can be found if every new column of the constraint matrix $\boldsymbol{B}^{k+1^T}$ is orthogonalized to the previous first k columns with respect to $\boldsymbol{F}_I$: this results in a diagonal and thus trivial dual interface operator $\boldsymbol{F}_I^{k+1}$.

The iteration results obtained when orthogonalizing the successive columns of $\boldsymbol{B}^k$ are completely equivalent to the results obtained when the dual problem is solved by factorizing $\boldsymbol{F}_I^k$ since the subspace in which the approximate interface loads are defined remains unchanged. However, orthogonalization mainly requires to perform dot products which are very well suited to parallel computation whereas the parallelization of an elimination algorithm is not straightforward [36, 67, 68].

The iteration scheme can be summarized as follows:

- Initialize
  - Solve the local Neumann problems

$$\boldsymbol{\lambda} = \boldsymbol{0} \tag{8.26}$$

$$\boldsymbol{u}^{(s),0} = \boldsymbol{K}^{(s)^{-1}} \boldsymbol{f}^{(s)} \tag{8.27}$$

$$\boldsymbol{r}^0 = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s),0} \tag{8.28}$$

$$\tag{8.29}$$

  - Compute a smoothed displacement by (7.29,7.39) and

$$\Delta \boldsymbol{f}^{(s),0} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}^{(s),0} \tag{8.30}$$

$$\boldsymbol{\lambda}^{sr,j^0} = \beta^{(r),j} \boldsymbol{B}^{(s,-),j^T} \Delta \boldsymbol{f}^{(s),0} - \beta^{(s),j} \boldsymbol{B}^{(r,-),j^T} \Delta \boldsymbol{f}^{(r),0} \tag{8.31}$$

- Iterate until convergence
  - If $k > 0$, orthogonalize the new stress vector:

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^k - \sum_{m=0}^{k-1} \frac{\boldsymbol{\lambda}^{k^T} \boldsymbol{F}_I \boldsymbol{\lambda}^m}{\boldsymbol{\lambda}^{m^T} \boldsymbol{F}_I \boldsymbol{\lambda}^m} \boldsymbol{\lambda}^m \tag{8.32}$$

  - Compute the interface Lagrange multipliers

$$\boldsymbol{\eta}^k = \frac{\boldsymbol{\lambda}^{k^T} \boldsymbol{d}}{\boldsymbol{\lambda}^{k^T} \boldsymbol{F}_I \boldsymbol{\lambda}^k} \tag{8.33}$$

$$\boldsymbol{\lambda} = \boldsymbol{\lambda} + \boldsymbol{\eta}^k \boldsymbol{\lambda}^k \tag{8.34}$$

  - $k = k + 1$, compute displacements and compatibility error

$$\boldsymbol{u}^{(s),k} = \boldsymbol{K}^{(s)^{-1}} (\boldsymbol{f}^{(s)} - \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}) \tag{8.35}$$

$$\boldsymbol{r}^k = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s),k} \tag{8.36}$$

– Compute a smoothed displacement by (7.29,7.39) and

$$\Delta \boldsymbol{f}^{(s),k} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}^{(s),k} \tag{8.37}$$

$$\boldsymbol{\lambda}^{sr,j^k} = \beta^{(r),j} \boldsymbol{B}^{(s,-),j^T} \Delta \boldsymbol{f}^{(s),k} - \beta^{(s),j} \boldsymbol{B}^{(r,-),j^T} \Delta \boldsymbol{f}^{(r),k} \tag{8.38}$$

- End

Comparing this iteration scheme to the basic primal-dual iteration presented in Section 8.2, the computations are kept to a minimum and the main steps include the solution of local Neumann and Dirichlet problems, and dot products like $\boldsymbol{\lambda}^{k^T}(\boldsymbol{F}_I \boldsymbol{\lambda}^m)$ where we assume that the result $\boldsymbol{F}_I \boldsymbol{\lambda}^m$ has been stored. In this form, the primal-dual iteration scheme is well adapted to parallel computing.

In fact, as will be discussed in Part II, this iteration scheme is equal to a Preconditioned Conjugate Gradient algorithm applied to the dual interface problem. For conforming substructures it corresponds to the Finite Element Tearing and Interconnecting (FETI) method presented in [67, 82]. For non-conforming meshes, the primal-dual iteration generalizes the FETI method. Recently, this technique has been applied for solving domain decomposition problems in fluid mechanics [117] where the non-conforming interfaces were assembled by a Mortar formulation [3, 17, 112].

Chapter 9

Conclusions of Part I

The reduced two-field hybrid method presented in [47,77] has been improved by adding a smoothing procedure that post-processes the weakly compatible solutions and enforces the reference compatibility. The smoothing involves the solution of a coarse grid problem where the substructures are considered as macro-elements. Our smoothing procedure generalizes the averaging or patching technique proposed in [47,77] in that it takes account of the substructure stiffnesses for computing a patched solution and it is applicable to conforming and non-conforming grids. Moreover, multiply connected substructures and interfaces of multiplicity equal to two are handled alike in a unified theory.

The efficiency of the smoothing procedure for computing accurate solutions of substructured problems with only a few interface Lagrange multipliers and smoothing coefficients has been illustrated in the examples for static analysis, free vibration and transient response analysis. In particular we have shown that our smoothing procedure is well suited for highly heterogeneous problems such as a checkerboard structure or a stiffened panel.

In all cases where the interface solution is smooth, i.e. where the actual interface stresses can be accurately represented by means of a few distribution functions, the combination of the reduced two-field hybrid method and an efficient smoothing scheme appears to form a highly concurrent and accurate solution method. A prescribed accuracy can be achieved at low cost by performing a few iterations where the number of interface variables is increased.

The reduced two-field technique when interface stresses are modeled with distribution functions is not applicable to problems with irregular interfaces (e.g. truss frames). In the latter case, a generalization of the method was proposed, based on the good estimate of the interface loads deriving from the smoothing procedure. The resulting primal-dual iterative method proved to be highly effective for solving the difficult problem of a truss frame.

We have mentioned the equivalence between the primal-dual iterations and the FETI method for conforming substructures. In part II, the FETI method will be presented in more detail under the point of view of a domain-decomposition method for parallel computing. We will exploit the mechanical understanding we gathered from our work on substructures assembly (and in particular from the smoothing method) for enhancing the efficiency of FETI, namely by ensuring good convergence of the interface iterations for highly heterogeneous structures.

Part II

Dual domain decomposition methods for efficient parallel computing

Chapter 10

Parallel computers and parallel computing

Ever since partial differential equations arising in science and engineering have been discretized for solving complex problems, scientists and engineers have been dreaming of ever more powerful computation means. Discretization methods and solution algorithms have been steadily improved and adapted to new computational resources. With the advent of computers, finite element formulations and specific numerical methods such as the frontal or skyline solvers became a major research issues and were brought to be widely used engineering tools.

In order to increase the speed of the computing resources, the CPU clock frequency has been multiplied by several orders of magnitude and new CPU technologies appeared. For instance, subtasks were defined for CPU operations so that they can be executed concurrently on different operands at a time: this led to pipelining and vector processing which are sometimes referred to as data parallelism [46]. Another technique for increasing the computing power is then to use several processors. Major computer companies took this path in the eighties, not the least of which are Cray, IBM, DEC and many others. The tremendous development of computer networking has also led to the idea of coupling commonly available computers such as workstations in order to form powerful computer clusters.

Together with the new parallel machines, new algorithms came about in the late eighties and today parallel computing is more than ever a major research topic in the engineering and scientific community. It is beyond the scope of this text to detail all the challenging and complex issues relating to parallel computers and to concurrent algorithms. Nevertheless, for the reader to understand the full extend of our work and to apprehend the importance of the new developments presented here, we will briefly discuss the context in which our research takes place. First we will recall the basics of parallel computers and then the relevant algorithmic issues will be revised.

10.1 Parallel computers

Parallel computers are characterized by their number of processors (granularity), the way processors are connected, the way memory is organized and by the organization of the operations [46,68].

Parallel computers can comprise a small number of powerful and expensive processors (2 to 32) or they can be made of a high number of simple and inexpensive ones (up to several thousands). They are called respectively *coarse grained* or *fine grained*. Fine grained computers are well suited for performing numerous simple and decoupled tasks such as in artificial intelligence whereas coarse grained computers are better adapted to performing parallel but possibly coupled operations. Coarse grained computers are typically used for engineering applications.

Each processor can be linked to all others by a cross-switch which works as a switchboard that establishes communication between any two processors. Another method is to connect all

processors by a high speed bus (or ring if the bus is closed). The connections to a bus are simple since all communications use the same cable, but contentions appear when several processors use the bus simultaneously. For fine grained computers, the processors are usually connected only to the nearest neighbors as if they were nodes of a more or less complex communication mesh: the connection scheme is simple and communications are rapid between neighboring processors, but when a processor communicates to another processor very distant in the mesh, the information must be passed on from one processor to the other and communication is slow, although meshes can be designed so to minimize the inter-processor distance (e.g. hypercube design). Hybrid connection schemes are often used in order to optimize communication speed, for instance by building clusters of a small number of inter-connected processors and connecting the clusters through a communication mesh.

Memory management is another primordial issue in parallel computers. RAM memory can be a single memory shared by all processors in which case one can define global variables and variables private to a specified processor. The advantage is that data must not be transfered from one processor to another, but memory contention appears when more than one processor accesses a variable simultaneously. Each processor can also be given its own local memory in which case memory contention problems do not exist, but data must be sent between processors. The actual trend is to build physically distributed memory which appears as shared to the users: the memory management is then ensured by the operating system. However, for the computation to be efficient, the user must in general manage the data flow himself in order to enforce data locality.

Data exchange between local memories of different processors is usually performed by message passing. In order to standardize the inter-processor communication commands on all machines and thus ensure portability of parallel codes, several communication standards have been defined, the most popular being PVM (Parallel Virtual Machine) [15, 120] which is to be superseded by MPI. PVM and MPI can be used for setting up communications between the processors of a parallel machine as well as between parallel or sequential machines of different types grouped in a cluster.

A thorough discussion on parallel machine architecture and operating modes can be found e.g. in [19, 46, 68].

10.2 Parallel algorithms

The fundamental question in parallel computing is "Do we compute faster if we run the job on more processors?". The answer to this question is not as obvious as it seems. Everybody knows that hiring 10 people to work on a same project does not guarantee that the final result will be produced in a tenth of the time it requires for a single person to accomplish the same job. The efficiency of parallel computing as the efficiency of team working will depend mainly on whether the work can be organized in independent and non-consecutive sub-tasks. Indeed, if no major part of the work can be done concurrently or if data communications makes up a significant part of the working time, the potential for parallelization is inexistent. Hence, parallel performance is strongly related to the way computation is organized, namely to the algorithms used to achieve a given result, and efficiency also depends on the ability of the processors to work together.

10.2.1 Measuring parallel efficiency

Calling t_1 the time required to perform a job consisting in a number N of operations, and calling t_p the overall time for the same job to be accomplished by p processors, the *speed-up* is defined as [46, 68]

$$S_p = \frac{t_1}{t_p} \tag{10.1}$$

and the *parallel efficiency* will be given by

$$E_p = \frac{S_p}{p} \tag{10.2}$$

The parallel efficiency indicates how efficiently the computing resources are used and in general $E_p < 1$ due to extra communication and algorithmic overheads inherent to the parallel computation.

Remark

- There is one common case when $E_p > 1$: considering again the example of a work team, imagine that all the documents needed for performing the task are so voluminous that they do not fit on one desk. Splitting the work and distributing the associated documents is then beneficial not only because more people are put at work but also because every worker can accommodate all the needed files on his desk. Exactly the same situation appears with parallel computers: if the data do not fit in the RAM memory of a single processor whereas the data needed for the sub-tasks can be stored entirely in each RAM of the multiple processors, significant speed-up will be achieved just because no temporary storage on memories with slower access time is necessary (see e.g. [182]).

Parallel speed-up and parallel efficiency are adversely affected by the existence of sub-tasks which are inherently sequential. Let us assume that in general the N operations are subdivided in $N = N_1 + N_2$, where N_1 is the number of operations performed at a speed of s_1 operations per unit time (e.g. in Mflops) and N_2 is the number of operations having an execution speed s_2. The total computing time is $t = N_1/s_1 + N_2/s_2$ and the overall computing speed s is

$$s = \frac{N}{t} = \frac{N}{N_1/s_1 + N_2/s_2} \tag{10.3}$$

Eq. (10.3) is known as *Amdahl's law* [6].

For parallel computation, let us neglect for simplicity's sake the communication time and let as call f_{seq} the portion of work N_{seq}/N performed sequentially at a speed s_{seq} and $(1 - f_{seq})$ the part N_{par}/N that runs in parallel at a speed $s_{par} = ps_{seq}$. Amdahl's law (10.3) then yields the overall computing speed

$$s = \frac{s_{seq}}{f_{seq} + (1 - f_{seq})/p} \tag{10.4}$$

and the parallel speed-up and efficiency are

$$S_p = \frac{N/s_{seq}}{N/s} = \frac{p}{pf_{seq} + (1 - f_{seq})} \tag{10.5}$$

$$E_p = \frac{S_p}{p} = \frac{1}{pf_{seq} + (1 - f_{seq})} \tag{10.6}$$

These relations are known as *Ware's law* [184]: they state that parallel speed-up is limited by $S_\infty = 1/f_{seq}$ and efficiency tends towards zero as the number of processors p increases. This is a very pessimistic result for the potentiality of parallel computing since if we assume for instance that 90 percent of the work can be fully run in parallel ($f_{seq} = 0.1$), a speed up of 4.7 can be theoretically achieved with 8 processors and for a very large number of processors p, S_p is limited to 10!

This very pessimistic conclusion derives from the assumption that the amount of work to be delivered by p processors remains constant. On distributed memory parallel machines, memory space is generally proportional to the number of processors p and in many scientific computing problems the problem size scales with the available processing power. Let us then consider the inverse of Amdahl's paradigm: let us analyze how long a parallel program would have taken on

a sequential machine. Calling f^*_{seq} the serial work fraction on a parallel system with p processors and calling t_{par} the total parallel CPU time, the fictive sequential CPU time is

$$t_{seq} = f^*_{seq} t_{par} + p(1 - f^*_{seq}) t_{par} \tag{10.7}$$

and the parallel speed-up and efficiency are

$$S_p = \frac{t_{seq}}{t_{par}} = f^*_{seq} + p(1 - f^*_{seq}) \tag{10.8}$$

$$E_p = \frac{S_p}{p} = \frac{f^*_{seq}}{p} + (1 - f^*_{seq}) \tag{10.9}$$

This model was proposed by Gustafson *et al.* [99]. It leads to the following conclusion: assuming that the amount of work increases proportionally to the number of processors p and further assuming that the fraction of sequential work remains constant, speedup is proportional to p. Parallel efficiency becomes independent of p as p increases and tends towards $(1 - f^*_{seq})$.

In practice communication time should be included in speed-up models. For a full discussion on speed-ups and efficiencies, see e.g. [46, 68].

10.2.2 Numerical and parallel scalability

In the definitions of parallel speed-up and efficiency (previous section), it is implicitly assumed that to a given problem corresponds a constant number N of operations. For actual algorithms this is not always true. Indeed, by reorganizing in parallel subtasks the way the final result will be computed can lead to an increase in the number of operations, hence jeopardizing the searched after parallel efficiency. The property of an algorithm to incur a constant (or nearly constant) number of operations when the number of processors is increased is known as *numerical scalability*.

Although numerical scalability is an essential property for the algorithm to be efficiently parallel, it will lead to *parallel scalability*, i.e. it will deliver a larger speed-up for a larger number of processors only if the communication cost and management overheads are small. Unfortunately, numerical scalability is usually obtained by transferring data among processors and thus numerical and parallel scalability are generally conflicting objectives [67].

For iterative solution schemes, the number of operations to achieve convergence towards the result depends on the cost per iteration and on the number of iterations. To achieve numerical scalability, the number of iterations should remain nearly constant when the number of processors is increased and the number of operations per iteration should be unchanged. For practical iterative algorithms, this can be obtained only by propagating informations at every iteration, hence degrading parallel scalability.

One of the most common and time consuming routines in engineering computations, and probably the most challenging to parallelize consists in the solution algorithm of large system of linear equations. Solving a large linear system is the very heart of non-linear problems, of dynamic analyses and even of popular eigensolvers. The importance of designing robust and parallel algorithms for solving large linear systems is thus overwhelming.

Algorithms can exploit parallelism in three different ways [69]:

Functional decomposition approaches decompose the solution space in subspaces. If the subspaces are independent, the process is naturally parallel: examples of this category are Fourier analysis of axisymmetric and cyclic structures [52].

Operator decomposition consists in decomposing the governing operator and compute the solution by operating on the different pieces.

Domain decomposition methods operate separately on subdomains of the entire structure so that the operations can be split among processors in a natural way.

The first category requires only minor adaptation work for use on parallel machines. The two others will be briefly discussed in the next sections.

10.2.3 Algebraic partitioning

Algebraic partitioning schemes fall in the category of operator decompositions. For solving large system of equations, algebraic partitioning consists in modifying the classical factorization schemes (LU, Cholesky, frontal or skyline methods) so to define parallel subtasks.

One method for instance operates by blocks: the diagonal blocks are factored sequentially and updating the out of diagonal submatrices is done in parallel. This blocked factorization has also the advantage of optimizing the cache memory transfers [121]. Another strategy for parallelizing factorization algorithms consists in reorganizing the imbricate loops on rows and columns of the operator [36, 45, 67, 187].

The parallel efficiency of the solution schemes based on algebraic partitioning of the operator is generally good for coarse grained computers (typically $p = 4$). When the number of processors becomes large, the number of communications inherent to these strategies leads to bad speed-ups. Moreover, algebraic partitioning leads to modifying the complete solution algorithm and thus updating a software for parallel computation implies a tremendous re-programming work.

10.2.4 Domain decomposition and global iteration schemes

Global iterative algorithms

Factorization schemes are inherently sequential since they are based on Gauss elimination where the solution is obtained for one variable after the other. A suitable parallel strategy should thus be based on algorithms that search for the solution as a whole, such as in iterative solution schemes.

In iterative algorithms the system operator is not factored and the very heart of the methods consists in multiplying the operator with successive solution directions. This is a naturally parallel task. Let us consider for instance the stiffness matrix for a static problem with conforming interfaces, which assembled form for a problem with N_s subdomains is

$$\boldsymbol{K} = \sum_{s=1}^{N_s} \boldsymbol{L}^{(s)^T} \boldsymbol{K}^{(s)} \boldsymbol{L}^{(s)} \tag{10.10}$$

where $\boldsymbol{L}^{(s)}$ is the Boolean assembly matrix (Sections 2.1.2 and 2.3.5). Multiplying the stiffness operator by a vector then writes

$$\boldsymbol{y} = \boldsymbol{K}\boldsymbol{x} = \sum_{s=1}^{N_s} \boldsymbol{L}^{(s)^T} \boldsymbol{K}^{(s)} \boldsymbol{L}^{(s)} \boldsymbol{x} \tag{10.11}$$

Equation (10.11) indicates that if the stiffness operator has been decomposed according to the contribution of N_s subdomains, the multiplication $\boldsymbol{K}\boldsymbol{x}$ implies that each subdomain performs independently the multiplication $\boldsymbol{y}^{(s)} = \boldsymbol{K}^{(s)}\boldsymbol{L}^{(s)}\boldsymbol{x}$ for the subdomain restriction of $\boldsymbol{x}$, and the result vector is then obtained by assembling the subdomain results, namely $\boldsymbol{y} = \sum_{s=1}^{N_s} \boldsymbol{y}(s)$. Applying the system operator to a vector is thus a naturally parallel task. The assembling of the result requires that the local results be summed up on the subdomain interfaces, which implies communications only between neighboring subdomains.

Iteration methods that operates on subdomain decomposed operators are thus naturally parallel algorithms. Iterative algorithms are not new, but they were left aside in engineering computations because they are less robust than the well known direct elimination methods. Their parallel potentiality has given them new importance and iterative algorithms form a very active research area nowadays. The most popular iterative algorithms are the Conjugate Gradient for symmetric operators and its variants for unsymmetric systems, e.g. biCG, GMRES and QMRES [13, 88, 167, 182]. All these iteration schemes are based on the Krylov subspace theory.

Preconditioners for iterative algorithms

Iterative algorithms are interesting compared to direct elimination schemes only if good convergence is guaranteed independently of the characteristics of the problem. The convergence of the iterative schemes based on Krylov subspaces is dependent on the condition number of the problem, namely on the discrepancy of its eigenvalues. The eigenspectrum of an operator being very much linked to the problem's mathematical model, discretization, mechanical characteristics and topology, iterative methods cannot be used in their basic form since they lack robustness.

Robustness can be obtained if a good approximation of the inverse of the operator is available. It is then used to precondition the successive iterates so that, ideally, the convergence rate is no longer function of the problem conditioning. The efficiency of the preconditioners depends on how accurately they represent the inverse of the system operator. On the other hand, preconditioners should not imply a significant computational effort.

The simplest preconditioner is obtained by approximating the inverse of the operator by the inverse of its diagonal. This technique known as the Jacobi preconditioner is clearly cheap, but its efficiency is poor because it neglects all coupling between the unknowns [114]. A more sophisticated variant of the Jacobi preconditioner is its block version: the inverse operator is approximated by the inverse of its diagonal blocks. This method is more efficient since it accounts for the coupling of unknowns within blocks but still it neglects the inter-block coupling.

Better preconditioners are obtained if more coupling is included in the approximation of the inverse operator. Preconditioners such as the incomplete Cholesky and incomplete LU factorization (ILU) perform a low cost partial elimination by considering for instance only the dominant coupling terms and by limiting the fill-in during factorization [13, 93] (all these incomplete factorization methods are based on SSOR techniques). Yet another preconditioner of the same family is the Element by Element (EBE) preconditioner [108]. Finally we must also mention the existence of less classical preconditioners such as the Tree Separator algorithm [97]. The latter algorithm seems very promising for building a good approximate inverse of the operator at a low cost.

Although the incomplete factorization preconditioning methods have proved to be efficient in ensuring good convergence at an acceptable cost, they are not well suited for parallel computing since they operate on the fully assembled operator. Their block variants have then be considered, each block corresponding to a subdomain. The incomplete factorization is then fully parallel but it leads to neglecting inter-subdomain coupling. Therefore the incomplete blocked preconditioners cannot ensure an acceptable robustness for the iterative algorithms as the number of subdomains is increased.

Multigrid methods

Multigrid methods are somewhat particular iterative methods first developed for fluid problems. Assuming that several imbricate levels of meshing exist, the iteration takes place on the finest grid whereas a kind of preconditioning is performed by solving a coarse problem derived by successive projections onto the coarser grid levels (e.g. [111]). The imbricate levels of discretization can be obtained by successive mesh refinements or by hierarchical finite element methods.

Industrial codes with iterative schemes

The essential issue in iterative algorithms operating on the global operator is to find a good compromise between efficient preconditioning for ensuring robustness, and parallel efficiency. Several popular finite element analysis softwares have been parallelized by using global iterative algorithms [52]. A major example is the parallel version of the code Msc-Nastran in which Jacobi and Incomplete Cholesky preconditioners have been integrated. In general, the Jacobi preconditioner is found to be better in terms of CPU time, the incomplete Cholesky preconditioner being too costly in practice. Hence, being said that Jacobi preconditioners are not very good at handling real complex problems, one can question the numerical scalability and thus the parallel scalability of the softwares based on iterations on the global operator.

10.2.5 Domain decomposition and iterative interface solution schemes

The algebraic partitioning methods are robust but they do not parallelize well because there is now natural parallelism in factorization schemes. On the opposite, global iterative algorithms are inherently parallel, but they are robust only for costly and non-fully parallel preconditioners.

Let us then consider the substructuring methods discussed in part I where an interface problem (dual or primal) is formed based on locally factored operators and let us assume that we associate one processor with every subdomain. If we then apply an iterative solution scheme for solving the interface problem, we obtain a method which hopefully has the combined advantages of algebraic partitioning and global iterations without their inconveniences. Indeed, solving the local subdomain problems by direct elimination is robust and fully parallel, and the interface problem is efficiently parallelized by using iteration methods.

It remains to ensure that solving iteratively the interface problem is robust. First, we note that an interface problem is of lower dimensionality than the global problem and convergence is thus easier to achieve than on the global system. Moreover, we will show in the following chapters that efficient preconditioners can be built based on the local operators.

The domain decomposition with iterative solution of the interface problem has two drawbacks compared to the domain decomposition methods with global iteration presented in the previous section:

- For interface iteration methods, the processor work load depends on the computational cost of factorizing and forward-backward substitutions for a local operator, and it varies according to the local bandwidth and number of variables. Good processor load balancing is thus more difficult to achieve than for the global iteration methods.
- When the subdomain decomposition is modified, the local operator changes and a new local factorization must be performed. For globally iterative schemes, changing the decomposition only requires re-associating the data to the processors.

Two different ways for defining the interface problem between domain decomposed subdomains exist, namely the primal formulation where the interface problem expresses the interface equilibrium in terms of interface displacements (primal variables) and the dual formulation expressing the interface equilibrium in terms of interface stresses (dual variables). These two approaches will be briefly discussed in Section 10.4.

Domain decomposition methods with direct elimination at the local level and iterative solvers for the interface problem have a high potentiality for parallel computing and moreover they are easier to implement than algebraic partitioning methods. Indeed, the subdomains are treated by classical routines that have been optimized for sequential computing, and the iterative solution of the interface problem requires implementing straightforward iterative algorithms. For these reasons, we firmly believe that these domain decomposition methods have a bright future in parallel computing.

Remarks

- In [67] it was found that domain decomposition methods are more efficient than direct elimination even on sequential machines. The reason for this is that subdomain operators have a smaller bandwidth than the global operator. Therefore, it can be interesting to define more subdomains than processors and map more than one subdomain onto every processor.
- On the opposite, for computers with clustered processors, one can associate several clustered processors to one subdomain if an optimum factorization algorithm can be implemented for those processors.
- Domain decomposition methods result in conforming interfaces since they generally define subdomains starting from a fully compatible global grid. Nevertheless, the domain decomposition with iterative interface solution can be generalized to non-conforming interfaces

arising for instance when a subdomain grid is refined. For the relevant discussions we refer the reader to part I of this text.

- In our work, we assume that subdomains are non-overlapping. Several specific algorithms relative to overlapping domains have been proposed, the most popular ones being the additive and multiplicative Schwartz algorithms [94, 115, 175].

10.3 Domain decomposers

The efficiency of a domain decomposed algorithm highly depends on the subdomain decomposition itself. Three major issues must be taken into account when defining a decomposition of a domain [67]:

- The work load must be evenly distributed between all processors, so that idling time is minimized. This issue is referred to as *load balancing*.

- The number of inter-processor communication must be minimized. This implies to minimize the number of interface unknowns as well as the inter-connectivity of the subdomains.

- The aspect ratio of the subdomains must be close to 1, meaning that one should avoid having elongated subdomains. This point is important for ensuring a good conditioning of the interface problem (see Section 13.1).

Finding the optimum decomposition for a large system is far too expensive and in practice, a quasi-optimum decomposition is searched for in two steps [62, 64, 67, 182]:

- A first crude decomposition is computed. This first step is performed by simple and cheap algorithms based on the topology of the structure or on the graph of the mesh and nodes.

- In a second step, the decomposition is optimized according to a cost function including load balancing, interface size and subdomain aspect ratio. A quasi-optimum is search by moving elements in the vicinity of the interface from one subdomain to another according to heuristics such as Simulated Annealing, Tabu Search or Stochastic evolution. As an important by-product of this optimization step, the interface of the quasi-optimum decomposition is geometrically smooth which improves the convergence of the iterative solution scheme.

The optimality of the decomposition depends on the algorithm that will be applied and on the parallel machine it will be run on. All the major algorithms are available together with some tools for estimating the quality of the decomposition in the Top-Domdec software [63].

10.4 The primal and dual domain decomposition method

In domain decomposition methods with iterative solution of the interface problem, independent local problems are solved with the interface variables remaining as unknowns. The subdomain coupling is expressed in terms of the interface unknowns which are then determined iteratively. Depending on whether the interface coupling is expressed in terms of displacements or stresses, the formulation is of the primal or dual type.

It is not the aim of this chapter to make a full description and analysis of primal and dual methods, but we will briefly introduce or recall the concepts necessary for understanding the further developments. Extensive discussion on primal and dual iterative methods can be found in [67, 178].

10.4.1 The primal Schur complement method

Let us assume that one unique set of d.o.f. is defined for the subdomains interface and that all interfaces are conforming. Calling $\boldsymbol{L}_b^{(s)}$ the restriction to interface of subdomain $\Omega^{(s)}$ of the Boolean assembly matrix $\boldsymbol{L}$, the interface boundary nodes of subdomain $\Omega^{(s)}$ are

$$\boldsymbol{u}_b^{(s)} = \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b \tag{10.12}$$

The operator $\boldsymbol{L}_b^{(s)}$ describes the correspondence between the global numbering of interface d.o.f. and the local numbering of $\boldsymbol{u}_b^{(s)}$. The structural problem (e.g. in static analysis) can then be formally written in the subdomain decomposed form

$$\begin{bmatrix} \boldsymbol{K}_{ii}(s)\boldsymbol{K}_{ib}^{(s)} \\ \boldsymbol{K}_{bi}(s)\boldsymbol{K}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(s)} \\ \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(s)} \\ \boldsymbol{f}_b^{(s)} \end{bmatrix} \qquad s = 1, \dots N_s \tag{10.13a}$$

$$\sum_{s=1}^{N_s} \boldsymbol{L}_b^{(s)^T} \left(\boldsymbol{K}_{bi}^{(s)} \boldsymbol{u}_i^{(s)} + \boldsymbol{K}_{bb}^{(s)} \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b \right) = \boldsymbol{L}_b^{(s)^T} \boldsymbol{f}_b^{(s)} \tag{10.13b}$$

Eq. (10.13a) expresses the subdomain equilibrium. Since only one unique set of interface d.o.f. is defined, the interface compatibility is naturally satisfied and Eq. (10.13b) expresses the equilibrium of all the forces acting on the interface.

In order to formulate the interface problem in terms of interface displacements only, we condense in (10.13a) the internal d.o.f. $\boldsymbol{u}_i^{(s)}$ by writing

$$\boldsymbol{u}_i^{(s)} = \boldsymbol{K}_{ii}^{(s)^{-1}} \left(\boldsymbol{f}_i^{(s)} - \boldsymbol{K}_{ib}^{(s)} \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b \right) \tag{10.14}$$

and the equilibrium of the subdomain is reduced to

$$\boldsymbol{S}_{bb}^{(s)} \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b = \boldsymbol{f}_b^{(s)} - \boldsymbol{K}_{bi}^{(s)} \boldsymbol{K}_{ii}^{(s)^{-1}} \boldsymbol{f}_i^{(s)} = \boldsymbol{f}_b^{(s)*} \tag{10.15}$$

where $\boldsymbol{S}_{bb}^{(s)}$ is the Schur complement of subdomain $\Omega^{(s)}$ and $\boldsymbol{f}_b^{(s)*}$ is the statically condensed boundary load.

The interface equilibrium (10.13b) finally writes

$$\left(\sum_{s=1}^{N_s} \boldsymbol{L}_b^{(s)^T} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{L}_b^{(s)} \right) \boldsymbol{u}_b = \sum_{s=1}^{N_s} \boldsymbol{L}_b^{(s)^T} \boldsymbol{f}_b^{(s)*} \tag{10.16}$$

This last relation is the expression of the interface problem in the primal form. The primal interface operator is thus the assembled Schur complement and when the primal interface problem is solved iteratively, multiplying by the Schur complement implies solving local Dirichlet problems and assembling the associated interface loads.

Remarks

- In practice the number of interface boundary nodes for a subdomain is always large enough so that the internal operator $\boldsymbol{K}_{ii}^{(s)}$ is always regular.

- For a non-conforming interface, the Boolean localization operators $\boldsymbol{L}_b^{(s)}$ are replaced by $\boldsymbol{b}^{(s)*}$, the subdomain restriction of the null space of the compatibility constraint matrix (see Section 2.3.5).

10.4.2 Dual Schur complement method (FETI)

We now assume that an independent set of interface boundary d.o.f. $\boldsymbol{u}_b^{(s)}$ is defined for each subdomain and that a unique boundary stress field is defined between any pair of connecting subdomains. The interface equilibrium is then trivially satisfied and the system of equations for the decomposed problem are

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} = \boldsymbol{f}^{(s)} \qquad s = 1, \dots N_s \tag{10.17a}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)^T}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{10.17b}$$

The first set of equations (10.17a) expresses the equilibrium of the subdomains when the interface stresses are $\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ are the Lagrange multipliers energetically conjugate to the compatibility constraints expressed by (10.17b). Solving the local problems in terms of the interface stresses and expressing the self-equilibrium of the loads applied to floating subdomains we find (see Section 3.2)

$$\begin{bmatrix} \boldsymbol{F}_I & \boldsymbol{G}_I \\ \boldsymbol{G}_I^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix} \tag{10.18}$$

where $\boldsymbol{F}_I$ is the dual interface operator formed by assembling the interface flexibilities

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{K}^{(s)^+} \boldsymbol{B}^{(s)^T} \tag{10.19}$$

and $\boldsymbol{G}_I$, $\boldsymbol{d}$ and $\boldsymbol{e}$ are as defined by (3.18b–3.18e).

If this dual form of the interface problem is solved iteratively, local Neumann problems must be solved at every iteration and the compatibility error is found by assembling in a dual way the interface displacements. This domain decomposition method is known as *Finite Element Tearing and Interconnecting* (FETI) [66,67,79,82,156]. It has been successfully implemented in the general finite element package SAMCEF [52].

Remarks

- If the internal d.o.f. are first statically condensed in Eq. (10.17a) we find that the dual interface operator takes the alternative form

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \boldsymbol{b}^{(s)^T} \tag{10.20}$$

  where $\boldsymbol{b}^{(s)}$ is the restriction of $\boldsymbol{B}^{(s)}$ to the boundary d.o.f.. This latter form clearly shows that $\boldsymbol{F}_I$ is the dual counterpart of the assembled Schur complement defined in (10.16). Therefore the FETI method is also sometimes called the dual Schur complement method.

- For conforming interfaces such as those obtained from domain decomposition, the constraint matrices $\boldsymbol{b}^{(s)}$ are signed Boolean localization operators. For non-conforming interfaces, the constraint matrices are consistently defined by integrating the compatibility constraints weighted by the Lagrange multiplier distributions (Eq. (2.46)).

10.4.3 Primal vs. dual Schur method

The primal and dual methods are very similar to one another in that they both solve a subdomain interface problem iteratively, the local problems being solved by direct elimination. Both methods are applicable for conforming and non-conforming interfaces. For the non-conforming case, the dual formulation is straightforward whereas the primal formulation requires to compute the null

space of the constraint matrix $\boldsymbol{B}$ or to project the successive iterates in the subspace of the constraints (Section 2.3.5).

The short summary of the methods as presented in this chapter reveals also other fundamental differences:

- In the dual formulation, additional variables are defined, namely the Lagrange multipliers representing the interface connecting loads. Moreover, for the dual approach, the entire local operator $\boldsymbol{K}^{(s)}$ must be factored, whereas for the primal method the somewhat smaller internal operator $\boldsymbol{K}_{ii}^{(s)}$ is to be factored. Nevertheless, the number of internal d.o.f. is generally close to the total number of d.o.f. in a subdomain and therefore the implication in terms of computational cost is negligible in practice.
- The local operators are singular for floating subdomains, and therefore the dual interface problem is constrained by the self-equilibrium condition on the Lagrange multiplier. This particularity of the dual interface problem makes the iterative solution scheme slightly more complex, but it will be seen that it leads to building a natural coarse grid problem which ensures numerical scalability.

A last essential difference between primal and dual formulations resides in the spectral properties of their interface operator. Recalling the conclusions of Chapter 4, we know that the high eigenspectrum of $\boldsymbol{S}$ is related to the mesh modes and its low eigenspectrum contains all the physically relevant ones. Since the dual operator $\boldsymbol{F}_I$ can be seen as an interface flexibility operator, its eigenspectrum is about the reverse eigenspectrum of $\boldsymbol{S}$ meaning that physically meaningful eigensolutions pertain to the high end of the eigenspectrum of $\boldsymbol{F}_I$. Hence, knowing that any iterative algorithm based on the Krylov subspaces first captures the high end of the eigenspectrum of an operator, we can expect that the dual method will converge much faster than the primal one. However, to be honest, one must amend this statement as follows:

The primal method is preconditioned using a Neumann strategy which is closely related to the dual operator (see Section 4.3 and the balanced method [125]). On the other hand, the dual operator is preconditioned by a Dirichlet based preconditioner (see Section 12.1) which is similar to the Schur complement $\boldsymbol{S}$. Hence the primal and dual preconditioned iterative algorithms are indeed very similar, except that for the primal method, the iteration is performed on the primal operator and preconditioned by a dual-like operator whereas in the FETI method, iterations are performed on the dual problem and preconditioned by a primal-like operator.

In this Part II, we will be concerned with the dual method, namely the FETI algorithm. However, most of the discussion can be directly transposed to the primal method.

10.5 Scope of new developments

In the remaining of Part II, we will be dealing with iterative techniques for the dual interface problem. The dual domain decomposition method has been extensively developed since the beginning of the nineties by Farhat and Roux [59, 67, 79, 82, 156]. Our contribution to this now popular solution method resides in improving the preconditioners and in particular, we propose a new theoretical frame for designing efficient preconditioners specially well suited to heterogeneous problems. Our approach is supported by mathematical developments but it is driven by sound mechanical considerations. Indeed, the major interesting point when working with domain decomposed interface problems is that classical methods such as the Conjugate Gradient algorithm take a direct physical meaning.

Part II is thus organized as follows. In Chapter 11 the Conjugate Gradient method will be revised. We will discuss the particular modifications required for solving the dual interface problem iteratively such as the floating mode coarse grid projection and we will also present a novel strategy for handling structures with global rigid body modes in which case the system is semi-definite.

In Chapter 12, the basic Dirichlet and lumped preconditioners for the iterative solution of the dual interface problem will be discussed. Special attention will be given to the mechanical consistency of these preconditioners. In Chapter 13, the effect of subdomain and finite element aspect

ratio will be analyzed. A new approximate Dirichlet preconditioner is then proposed in Chapter 14 aiming to replace the expensive Dirichlet preconditioner by an approximate preconditioner nearly as efficient as the exact Dirichlet one but less demanding in terms of computational effort.

In Chapter 15 we show the adverse effect of structural heterogeneities on the convergence of the iterative FETI method. We then discuss different ways to adapt the Dirichlet and lumped preconditioners so as to take account of the discrepancy between subdomain stiffnesses. In particular the smoothing procedure developed in Part I for computing an accurate solution from a weakly compatible result will be introduced in the preconditioners. The "checkerboard problem" consisting in structures with juxtaposed uniform subdomains of highly different stiffnesses is discussed. A special iteration technique for the two subdomain problem is then proposed. The Chapter ends by discussing the concept of primal residue and how it should be evaluated.

In Chapter 16 the FETI method is applied to dynamic problems and in Chapter 17 some interesting research directions are presented for using the FETI method in non-linear analysis. Finally, we conclude this work by discussing our contribution to improving the FETI method and thereby to rendering more efficient the use of iterative solution schemes for parallel computing.

Chapter 11

The primal and dual domain decomposition method

In domain decomposition methods with iterative solution of the interface problem, independent local problems are solved with the interface variables remaining as unknowns. The subdomain coupling is expressed in terms of the interface unknowns which are then determined iteratively. Depending on whether the interface coupling is expressed in terms of displacements or stresses, we define a primal or a dual formulation.

It is not the aim of this chapter to make a full description and analysis of primal and dual methods, but we will briefly introduce or recall the concepts necessary for understanding the further developments. Extensive discussion on primal and dual iterative methods can be found in [**?**].

11.1 The primal Schur complement method

Let us assume that one unique set of d.o.f. is defined for the subdomains interface and that all interfaces are conforming. Calling $\boldsymbol{L}_b^{(s)}$ the restriction to interface of subdomain $\Omega^{(s)}$ of the Boolean assembly matrix $\boldsymbol{L}$, the interface boundary nodes of subdomain $\Omega^{(s)}$ are

$$\boldsymbol{u}_b^{(s)} = \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b \tag{11.1}$$

The operator $\boldsymbol{L}_b^{(s)}$ describes the correspondence between the global numbering of interface d.o.f. and the local numbering of $\boldsymbol{u}_b^{(s)}$. The structural problem (e.g. in static analysis) can then be formally written in the subdomain decomposed form

$$\begin{bmatrix} \text{bold}K \\ {}^{c}{}_{ii}(s)\boldsymbol{K}_{ib}^{(s)} \\ \boldsymbol{K}_{bi}(s)\boldsymbol{K}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \text{bold}u \\ {}^{c}{}_{i}^{(s)} \\ \boldsymbol{L}_b^{(s)}\boldsymbol{u}_b \end{bmatrix}$$

Eq. (**??**) expresses the subdomain equilibrium. Since only one unique set of interface d.o.f. is defined, the interface compatibility is naturally satisfied and Eq. (**??**) expresses the equilibrium of all the forces acting on the interface.

In order to formulate the interface problem in terms of interface displacements only, we condense in (**??**) the internal d.o.f. $\boldsymbol{u}_i^{(s)}$ by writting

$$\boldsymbol{u}_i^{(s)} = \boldsymbol{K}_{ii}^{(s)^{-1}} \left(\boldsymbol{f}_i^{(s)} - \boldsymbol{K}_{ib}^{(s)} \boldsymbol{L}_b^{(s)} \boldsymbol{u}_b \right) \tag{11.3}$$

and the equilibrium of the subdomain is reduced to

$$\boldsymbol{S}_{bb}^{(s)}\boldsymbol{L}_b^{(s)}\boldsymbol{u}_b = \boldsymbol{f}_b^{(s)} - \boldsymbol{K}_{bi}^{(s)}\boldsymbol{K}_{ii}^{(s)^{-1}}\boldsymbol{f}_i^{(s)} = \boldsymbol{f}_b^{(s)*} \tag{11.4}$$

where $\boldsymbol{S}_{bb}^{(s)}$ is the Schur complement of subdomain $\Omega^{(s)}$ and $\boldsymbol{f}_b^{(s)*}$ is the statically condensed boundary load.

The interface equilibrium (**??**) finally writes

$$\left(\sum_{s=1}^{N_s}\boldsymbol{L}_b^{(s)^T}\boldsymbol{S}_{bb}^{(s)}\boldsymbol{L}_b^{(s)}\right)\boldsymbol{u}_b = \sum_{s=1}^{N_s}\boldsymbol{L}_b^{(s)^T}\boldsymbol{f}_b^{(s)*} \tag{11.5}$$

This last relation is the expression of the interface problem in the primal form. The primal interface operator is thus the assembled Schur complement and when the primal interface problem is solved iteratively, multiplying by the Schur complement implies solving local Dirichlet problems and assembling the associated interface loads.

Remarks

- In practice the number of interface boundary nodes for a subdomain is large enough so that the internal operator $\boldsymbol{K}_{ii}^{(s)}$ is always regular.
- For a non-conforming interface, the Boolean localization operators $\boldsymbol{L}_b^{(s)}$ are replaced by $\boldsymbol{b}^{(s)*}$, the subdomain restriction of the null space of the compatibility constraint matrix (see Section 2.3.5).

11.2 Dual Schur complement method (FETI)

We now assume that an independent set of interface boundary d.o.f. $\boldsymbol{u}_b^{(s)}$ is defined for each subdomain and that a unique boundary stress field is defined between any pair of connecting subdomains. The interface equilibrium is then trivially satisfied and the system of equations for the decomposed problem are

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} = \boldsymbol{f}^{(s)} \qquad s = 1,\dots N_s \tag{11.6a}$$

$$\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)^T}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{11.6b}$$

The first set of equations (**??**) expresses the equilibrium of the subdomains when the interface stresses are $\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ are the Lagrange multipliers energetically conjugate to the compatibility constraints expressed by (**??**). Solving the local problems in terms of the interface stresses and expressing the self-equilibrium of the loads applied to floating subdomains we find (see Section 3.2)

$$\begin{bmatrix} \boldsymbol{F}_I & \boldsymbol{G}_I \\ \boldsymbol{G}_I^T & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix} \tag{11.7}$$

where $\boldsymbol{F}_I$ is the dual interface operator formed by assembling the interface flexibilities

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{K}^{(s)^+}\boldsymbol{B}^{(s)^T} \tag{11.8}$$

and $\boldsymbol{G}_I$, $\boldsymbol{d}$ and $\boldsymbol{e}$ are as defined by (3.18b–3.18e).

If this dual form of the interface problem is solved iteratively, local Neumann problems must be solved at every iteration and the compatibility error is found by assembling in a dual way the interface displacements. This domain decomposition method is known as Finite Element Tearing and Interconnecting (FETI) [66, 67, 79, 82, 156].

Remarks

- If the internal d.o.f. are first statically condensed in Eq. (**??**) we find that the dual interface operator takes the alternative form

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \boldsymbol{b}^{(s)^T} \tag{11.9}$$

  where $\boldsymbol{b}^{(s)}$ is the restriction of $\boldsymbol{B}^{(s)}$ to the boundary d.o.f.. This latter form clearly shows that $\boldsymbol{F}_I$ is the dual counterpart of the assembled Schur complement defined in (**??**). Therefore the FETI method is also sometimes called the dual Schur complement method.

- For conforming interfaces such as those obtained from domain decomposition, the constraint matrices $\boldsymbol{b}^{(s)}$ are signed Boolean localization operators. For non-conforming interfaces, the constraint matrices are consistently defined by integrating the compatibility constraints wheighted by the Lagrange multiplier distributions (Eq. (2.46)).

11.3 Primal vs. dual Schur method

The primal and dual methods are very similar to one another in that they both solve a subdomain interface problem iteratively, the local problems being solved by direct elimination. Both methods are applicable for conforming and non-conforming interfaces. For the non-conforming case, the dual formulation is straightforward whereas the primal formulation requires to compute the null space of the constraint matrix $\boldsymbol{B}$ or to project the successive iterates in the subspace of the constraints (Section 2.3.5).

The short summary of the methods as presented in this chapter reveals also other fundamental differences:

- In the dual formulation, additional variables are defined, namely the Lagrange multipliers representing the interface connecting loads. Moreover, for the dual approach, the entire local operator $\boldsymbol{K}^{(s)}$ must be factored, whereas for the primal method the somewhat smaller internal operator $\boldsymbol{K}_{ii}^{(s)}$ is to be factored. Nevertheless, the number of internal d.o.f. is generally close to the total number of d.o.f. in a subdomain and therefore the implication in terms of computational cost is negligible in practice.

- The local operators are singular for floating subdomains, and therefore the dual interface problem is constrained by the self-equilibrium condition on the Lagrange multiplier. This particularity of the dual interface problem slightly complexifies the iterative solution scheme for the interface problem, but it will be seen that it leads to building a natural coarse grid problem which ensures numerical scalability.

A last essential difference between primal and dual formulations resides in the spectral properties of their interface operator. Recalling the conclusions of Chapter 4, we know that the high eigenspectrum of $\boldsymbol{S}$ is related to the mesh modes and its low eigenspectrum contains all the physically relevant ones. Since the dual operator $\boldsymbol{F}_I$ can be seen as a interface flexibility operator, its eigenspectrum is about the reverse eigenspectrum of $\boldsymbol{S}$ meaning that physically meaningfull eigensolutions pertain to the high end of the eigenspectrum of $\boldsymbol{F}_I$. Hence, knowing that any iterative algorithm based on the Krylov subspaces first captures the high end of the eigenspectrum of an operator, we can expect that the dual method will converge much faster than the primal one. However, to be honest, one must amend this statement as follows:

The primal method is preconditioned using a Neumann strategy which is closely related to the dual operator (see Section 4.3 and the balanced method [125]). On the other hand, the dual operator is preconditioned by a Dirichlet based preconditioner (see Section 12.1) which is similar to the Schur complement $\boldsymbol{S}$. Hence the priaml and dual preconditioned itertive algorithms are

indeed very similar, except that for the primal method, the iteration is performed on the primal operator and preconditioned by a dual-like operator whereas in the FETI method, iterations are performed on the dual problem and preconditioned by a primal-like operator.

In the remaining of this Part II, we will be deling with the dual method, namely the FETI algorithm. However, most of the discussion can be directly transposed to the primal method.

to

Chapter 12

Conjugate gradient for the dual interface problem

In this chapter we will describe how the Conjugate Gradient applies to the dual Schur complement method. We first present the basic Conjugate Gradient iteration method and then we explain how to integrate in this scheme the self-equilibrium condition on the Lagrange multipliers. We briefly describe how to treat problems with multiple right-hand sides and we will present in Section 11.5 a novel strategy for treating structures with global null energy modes. To conclude this chapter, we will discuss a particular technique for strongly enforcing constraints during the iterations.

12.1 The general Conjugate Gradient

The Conjugate Gradient algorithm can be presented in many different ways [88, 96, 167]. The approach we choose consists in deriving it as an iterative approximation technique for evaluating the stationary point of dual energy functionals.

12.1.1 The Conjugate Gradient Algorithm

Let us first assume that all subdomains are non-floating so that the dual interface problem writes

$$\boldsymbol{F}_I \boldsymbol{\lambda} = \boldsymbol{d} \tag{12.1}$$

The energy functional related to the dual interface problem is then the Lagrangian function (3.6b) and we write that the solution of the dual interface problem minimizes the functional

$$\phi(\boldsymbol{\lambda}) = -\mathcal{L}(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^T \boldsymbol{F}_I \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \boldsymbol{d} \tag{12.2}$$

The conjugate gradient consists in finding the minimum of ϕ by adding at every iteration a new direction to the solution space. Note that this is totally equivalent to improving iteratively the approximation space in a variational sense (see Chapter 8). At iteration k, the Lagrange multipliers will be corrected by adding a new solution direction $\boldsymbol{p}^k$:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^0 + \sum_{i=0}^{k} \eta^i \boldsymbol{p}^i \tag{12.3}$$

$$= \boldsymbol{\lambda}^k + \eta^k \boldsymbol{p}^k \tag{12.4}$$

where $\boldsymbol{\lambda}^0$ is an initial guess and η^k is the amplitude of the new direction in the solution space which is determined by expressing the stationarity of

$$\phi(\boldsymbol{\lambda}^{k+1}) = \phi(\boldsymbol{\lambda}^k) + \frac{1}{2}\eta^{k^2} \boldsymbol{p}^{k^T} \boldsymbol{F}_I \boldsymbol{p}^k + \eta^k \boldsymbol{\lambda}^{k^T} \boldsymbol{F}_I \boldsymbol{p}^k - \eta^k \boldsymbol{p}^{k^T} \boldsymbol{d} \tag{12.5}$$

The direction $\boldsymbol{p}^k$ is taken as the direction of steepest descent for the functional $\phi(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}^k$. Hence,

$$\begin{aligned} \boldsymbol{p}^k &= -\nabla\phi|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^k} && (12.6)\\ &= \boldsymbol{d} - \boldsymbol{F}_I\boldsymbol{\lambda}_k = \boldsymbol{r}^k && (12.7)\end{aligned}$$

where $\boldsymbol{r}^k$ is the residue of the dual interface problem, namely the compatibility error on the interface. If we want the minimization of $\phi(\boldsymbol{\lambda}^{k+1})$ in (11.5) to be independent of the previous minimizations, we must ensure that $\boldsymbol{p}^k$ is orthogonal to the previous directions $\boldsymbol{p}^l$, $l = 0, \ldots k-1$, so that in (11.5) we have ${\boldsymbol{\lambda}^k}^T \boldsymbol{F}_I \boldsymbol{p}^k = 0$. Thus we have

$$\begin{aligned} \boldsymbol{p}^k &= \boldsymbol{r}^k - \sum_{i=0}^{k-1} \beta_i^k \boldsymbol{p}^i && (12.8)\\ \beta_i^k &= \frac{{\boldsymbol{p}^i}^T \boldsymbol{F}_I \boldsymbol{r}^k}{{\boldsymbol{p}^i}^T \boldsymbol{F}_I \boldsymbol{p}^i} && (12.9)\end{aligned}$$

and the minimization of $\phi(\boldsymbol{\lambda}^{k+1})$ in (11.5) yields

$$\eta^k = \frac{{\boldsymbol{p}^k}^T \boldsymbol{r}^k}{{\boldsymbol{p}^k}^T \boldsymbol{F}_I \boldsymbol{p}^k} \qquad (12.10)$$

Algorithm 12.1 Conjugate Gradient iterations for the dual interface problem (non-floating subdomains)

$k = 0$, $\boldsymbol{\lambda}^0 = \boldsymbol{0}$, $\boldsymbol{r}^0 = \boldsymbol{d}$
while $\|\boldsymbol{r}^k\| > \epsilon\|\boldsymbol{d}\|$ **do**
  $\boldsymbol{p}^k = \boldsymbol{r}^k$
  -Orthogonalization of directions-
  for $i = 0, \ldots k-1$ **do**
    $\beta_i^k = \frac{{\boldsymbol{p}^i}^T \boldsymbol{F}_I \boldsymbol{p}^k}{{\boldsymbol{p}^i}^T \boldsymbol{F}_I \boldsymbol{p}^i}$, $\boldsymbol{p}^k := \boldsymbol{p}^k - \beta_i^k \boldsymbol{p}^i$
  end for

  -Minimization-
  $\eta^k = \frac{{\boldsymbol{p}^k}^T \boldsymbol{r}^k}{{\boldsymbol{p}^k}^T \boldsymbol{F}_I \boldsymbol{p}^k}$
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k \boldsymbol{p}^k$
  $\boldsymbol{r}^{k+1} = \boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}^{k+1} = \boldsymbol{r}^k - \eta^k \boldsymbol{F}_I \boldsymbol{p}^k$
  $k = k + 1$

end while

The conjugate gradient is summarized in Algorithm 11.1. We observe that an iteration involves only one multiplication by the dual operator $\boldsymbol{F}_I$, which amounts to solving in parallel one Neumann problem per subdomain. Note that the dual residue $\boldsymbol{r}^{k+1}$ is computed recursively by making use of $\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \eta^k \boldsymbol{F}_I \boldsymbol{p}^k$. Unfortunately, due to propagation of round-off errors, the accuracy of the recursively evaluated residue is degraded and one should from time to time compute an exact residue by $\boldsymbol{r}^{k+1} = \boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}^{k+1}$. This requires to solve an extra set of local Neumann problems. In practice, the recursive formula is used, and one performs an exact update of the residue when the accumulated errors become significant. A full discussion of the problem of correctly evaluating the residue and on the updating strategy is presented in [169].

In Algorithm 11.1, the convergence criterion is set on the dual error $\boldsymbol{r}^k$. However in practice, a tolerance is set on the associated primal residue (see Section 15.7).

12.1.2 Convergence of the Conjugate Gradient

At iteration k, the direction amplitude η^k is such that it minimizes $\phi(\boldsymbol{\lambda}^{k+1})$, hence from (11.5) we find

$$\boldsymbol{p}^{k^T}(\boldsymbol{r}^k - \eta^k \boldsymbol{F}_I \boldsymbol{p}^k) = \boldsymbol{p}^{k^T}\boldsymbol{r}^{k+1} = 0 \tag{12.11}$$

meaning that the residue $\boldsymbol{r}^{k+1}$ is orthogonal to the direction $\boldsymbol{p}^k$. The residue $\boldsymbol{r}^{k+1}$ is also orthogonal to all previous conjugate directions: according to the orthogonality relations between the conjugate directions and to the relation (11.11) satisfied at every iteration we can write

$$\boldsymbol{p}^{j^T}\boldsymbol{r}^{k+1} = \boldsymbol{p}^{j^T}\left(\boldsymbol{r}^{j+1} - \sum_{i=j+1}^{k} \eta^i \boldsymbol{F}_I \boldsymbol{p}^i\right) = 0 \qquad j = 0, \dots k \tag{12.12}$$

Thus after k iterations, the residue $\boldsymbol{r}^{k+1}$ is zero in the subspace $\text{span}\{\boldsymbol{p}^0, \dots \boldsymbol{p}^k\}$. Noting that

$$\text{span}\{\boldsymbol{p}^0, \dots \boldsymbol{p}^k\} = \text{span}\{\boldsymbol{r}^0, \dots \boldsymbol{r}^k\} = \text{span}\{\boldsymbol{d}, \boldsymbol{F}_I\boldsymbol{d}, \dots {\boldsymbol{F}_I}^k\boldsymbol{d}\} \tag{12.13}$$

we conclude that the Conjugate Gradient iteration searches for the minimum of ϕ in the successive Krylov subspaces $\mathcal{K}_k(\boldsymbol{d}, \boldsymbol{F}_I) = \text{span}\{\boldsymbol{d}, \boldsymbol{F}_I\boldsymbol{d}, \dots {\boldsymbol{F}_I}^k\boldsymbol{d}\}$. The convergence rate of the solution is thus function of the condition number κ of $\boldsymbol{F}_I$, namely of the ratio between its highest and lowest eigenvalue [96]:

$$\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^k\|_{\boldsymbol{F}_I} \;\leq\; 2\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\|_{\boldsymbol{F}_I}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \tag{12.14}$$

The exact solution is found when the residue is zero in the entire solution space, that is after $n_{\lambda^e ff}$ iterations, $n_{\lambda^e ff}$ being the effective number of Lagrange multipliers obtained by subtracting from n_λ the number of redundant constraints.

In practice, the iterations are stopped when the residue is lower than a given tolerance and thus the number of iterations directly depends on the convergence rate. For the dual formulation, we have shown in Section 4.4 that the physically meaningful eigensolutions of $\boldsymbol{F}_I$ pertain to the higher end of its eigenspectrum which is represented first in the Krylov subspace. Moreover, the spectrum of $\boldsymbol{F}_I$ is characterized by clustered eigenvalues which is beneficial for convergence. Hence, the dual formulation leads to a super-convergence rate [67].

12.1.3 Recursive orthogonality

Let us substitute the orthogonalization relations (11.8) in Eq. (11.12):

$$\begin{aligned}\boldsymbol{p}^{j^T}\boldsymbol{r}^{k+1} &= \left(\boldsymbol{r}^j - \sum_{i=0}^{j-1}\beta_i^j \boldsymbol{p}^i\right)^T \boldsymbol{r}^{k+1} \\ &= \boldsymbol{r}^{j^T}\boldsymbol{r}^{k+1} = 0 \qquad j = 0, \dots k\end{aligned} \tag{12.15}$$

The orthogonalization coefficient β_i^k between directions $\boldsymbol{p}^k$ and $\boldsymbol{p}^i$, $i = 0, \dots k-1$ in Eqs. (11.9) can now be transformed as follows:

$$\begin{aligned}\beta_i^k &= \frac{\boldsymbol{p}^{i^T}\boldsymbol{F}_I\boldsymbol{r}^k}{\boldsymbol{p}^{i^T}\boldsymbol{F}_I\boldsymbol{p}^i} = \frac{(\boldsymbol{r}^i - \boldsymbol{r}^{i+1})^T\boldsymbol{r}^k}{(\boldsymbol{r}^i - \boldsymbol{r}^{i+1})^T\boldsymbol{p}^i} \\ &= 0 \qquad \text{if} \quad i < k-1 \qquad \text{(12.16a)} \\ &= \frac{\boldsymbol{r}^{k^T}\boldsymbol{r}^k}{\boldsymbol{r}^{k-1^T}\boldsymbol{p}^{k-1}} \quad \text{if} \quad i = k-1 \qquad \text{(12.16b)}\end{aligned}$$

This shows that $\boldsymbol{p}^k$ must be orthogonalized only with respect to the previous direction. Nevertheless, due to round-off errors in finite precision computation, orthogonality is lost after about 10

iterations for practical problems and therefore it is generally recommended to perform a full orthogonalization as shown in Algorithm 11.1 in order not to perturb the convergence of the Conjugate Gradient iterations. Full orthogonalization is costly since it requires storing all directions $\boldsymbol{p}^i$ and $\boldsymbol{F}_I\boldsymbol{p}^i$, and it implies performing additional dot products. Nevertheless, full re-orthogonalization is overall more cost effective [59].

12.2 Preconditioned Conjugate Gradient

The successive directions of descent in the Conjugate Gradient are the steepest descent directions. However, depending on the convexity of the functional ϕ, the steepest direction may not be the most rapid way to the minimum.

From a physically view point, in the Conjugate gradient we use the compatibility error $\boldsymbol{r}^k$ on the interface to approximate the interface stress corrections for $\boldsymbol{\lambda}^k$. The representation of the interface stress would be much more accurate if one uses as corrections an estimate of the interface stresses associated with the compatibility error $\boldsymbol{r}^k$. If one knows for instance the exact inverse of $\boldsymbol{F}_I$, one could compute the exact stress correction corresponding to $\boldsymbol{r}^k$ and convergence would be obtained instantaneously. Obviously, the inverse of $\boldsymbol{F}_I$ is not available, but one should at least use an estimate $\tilde{\boldsymbol{F}}_I^{-1}$ of ${\boldsymbol{F}_I}^{-1}$ so to evaluate a descent direction proportional to a stress and which is closer to the exact stress correction than $\boldsymbol{r}^k$.

We will thus *precondition* the minimization direction $\boldsymbol{p}^k$ provided that we define a preconditioner $\tilde{\boldsymbol{F}}_I^{-1}$ which is an approximate inverse of $\boldsymbol{F}_I$. The conjugate directions are then computed by

$$\boldsymbol{p}^k = \tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{r}^k \tag{12.17}$$

Only the first statement in the iteration loop of Algorithm 11.1 is thus modified. The convergence rate is still given by Eq. (11.14) but the condition number κ to consider is now the condition number of $\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{F}_I$ which tends towards one as the preconditioner tends towards the exact inverse of $\boldsymbol{F}_I$.

The preconditioner must be symmetric in order to ensure recursive orthogonality between residues and directions of descents and thereby ensuring the convergence of the iterations. The preconditioner can however be different at every iteration. The orthogonality relations (11.15) between residues now writes

$$\boldsymbol{r}^{j^T}\tilde{\boldsymbol{F}}_I^{j^{-1}}\boldsymbol{r}^{k+1} = 0 \qquad j = 0, \dots k \tag{12.18}$$

The recursive orthogonality relations (11.16) are still valid only if the the preconditioner is invariant. If the preconditioner changes during the iterations, a full re-orthogonalization is mandatory as in Algorithm 11.1.

12.3 Projected Conjugate Gradient

In practice, when a structure is decomposed, some of the subdomains are floating and we must determine the amplitudes of the floating modes by ensuring that the interface stresses are self-equilibrated. This leads to the now familiar dual interface problem (see Section 3.2)

$$\begin{bmatrix} \boldsymbol{F}_I & \boldsymbol{G}_I \\ \boldsymbol{G}_I^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix} \tag{12.19}$$

This system is not positive definite and a Conjugate Gradient procedure cannot be applied as such since the associated functional

$$\phi(\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \boldsymbol{\lambda}^T\boldsymbol{F}_I\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\boldsymbol{d} + \boldsymbol{\alpha}^T(\boldsymbol{G}_I^T\boldsymbol{\lambda} - \boldsymbol{e}) \tag{12.20}$$

does not define a classical minimum problem.

12.3.1 The PCPG algorithm

The solution space for $\boldsymbol{\lambda}$ is subdivided into the range of $\boldsymbol{G}_I$ and into the subspace orthogonal to $\boldsymbol{G}_I^T$. Introducing a regular symmetric operator $\boldsymbol{Q}$, the image of $\boldsymbol{G}_I$ can be written in a general way as span$\{\boldsymbol{QG}_I\}$ and the solution then writes

$$\boldsymbol{\lambda} = \boldsymbol{QG}_I \left(\boldsymbol{G}_I^T \boldsymbol{QG}_I\right)^{-1} \boldsymbol{G}_I^T \boldsymbol{\lambda} + \boldsymbol{P\lambda} \tag{12.21a}$$

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{QG}_I \left(\boldsymbol{G}_I^T \boldsymbol{QG}_I\right)^{-1} \boldsymbol{G}_I^T \tag{12.21b}$$

where $\boldsymbol{P}$ is a projection operator onto the subspace orthogonal to $\boldsymbol{G}_I$:

$$\boldsymbol{G}_I^T \boldsymbol{P} = \boldsymbol{0} \tag{12.22}$$

As will be discussed later in Section 11.5 ($\boldsymbol{G}_I^T \boldsymbol{QG}_I$) is regular if no floating modes exist. Taking into account the self-equilibrium condition $\boldsymbol{G}_I^T \boldsymbol{\lambda} = \boldsymbol{e}$, we finally write

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^0 + \boldsymbol{P\lambda} \tag{12.23}$$

$$\boldsymbol{\lambda}^0 = \boldsymbol{QG}_I \left(\boldsymbol{G}_I^T \boldsymbol{QG}_I\right)^{-1} \boldsymbol{e} \tag{12.24}$$

In the same manner, the dual interface problem is projected onto the space span$\{\boldsymbol{QG}_I\}$ and onto its orthogonal counterpart. We obtain

$$\boldsymbol{P}^T \boldsymbol{F}_I \boldsymbol{P\lambda} = \boldsymbol{P}^T (\boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}^0) \tag{12.25}$$

$$\boldsymbol{G}_I^T \boldsymbol{QG}_I \boldsymbol{\alpha} = \boldsymbol{G}_I^T \boldsymbol{Q} (\boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}) \tag{12.26}$$

The Conjugate gradient is then applied for the projected compatibility equation (11.25) and the iterations take place in the space of self-equilibrated Lagrange multipliers. Considering (11.26) and (11.21b), the residue of the dual interface problem can be written as

$$\boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}^k - \boldsymbol{G}_I \boldsymbol{\alpha}^k = \boldsymbol{P}^T \left(\boldsymbol{d} - \boldsymbol{F}_I (\boldsymbol{\lambda}^0 + \boldsymbol{P\lambda}^k)\right) \tag{12.27}$$

$$= \boldsymbol{w}^k \tag{12.28}$$

This last relation shows that the actual interface compatibility residue is the residue of the projected compatibility equation.

The FETI algorithm thus consists in a Preconditioned Conjugate Projected Gradient (PCPG) which is summarized in Algorithm 11.2.

12.3.2 Convergence of the PCPG

The Preconditioned Conjugate Projected Gradient is a preconditioned Conjugate Gradient on the operator $\boldsymbol{P}^T \boldsymbol{F}_I \boldsymbol{P}$. Its convergence rate is thus given by (11.14) where the condition number is the ratio between the highest and the lowest non-zero eigenvalue of $\boldsymbol{P}^T \boldsymbol{F}_I \boldsymbol{P}$. Recalling that the highest eigenvalues of $\boldsymbol{F}_I$ are related to the low eigenfrequency modes of the local operators, we understand that the effect of the projector $\boldsymbol{P}$ is to remove the eigenvalues lying in the subspace of $\boldsymbol{G}_I$ and thereby to eliminate part of the high eigenspectrum of $\boldsymbol{F}_I$, thus improving its condition number.

From a mechanical view point, the projection of the dual residue consists in finding the amplitudes of the rigid body mode displacements for the floating subdomains. The projection of the Lagrange multipliers consists in adding to the Lagrange multipliers the interface stresses corresponding to the rigid displacement modes and which ensure the self-equilibrium of the subdomains. Hence, the projection step can be interpreted as computing the corrections of a coarse grid formed by the rigid body displacements of all the floating modes. This coarse grid is called the *floating modes coarse grid*: it propagates the errors across the entire structure and is thus essential for ensuring a good convergence of the FETI method as the number of subdomains increases. Therefore floating modes are essential for the FETI method since they provide a natural and efficient coarse grid.

Algorithm 12.2 Preconditioned Conjugate Projected Gradient iterations

$k = 0$, $\boldsymbol{\lambda}^0 = \boldsymbol{Q}\boldsymbol{G}_I \left(\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{G}_I\right)^{-1} \boldsymbol{e}$,
$\boldsymbol{r}^0 = \boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}^0$
$\boldsymbol{w}^0 = \boldsymbol{P}^T \boldsymbol{r}^0$
while $\|\boldsymbol{w}^k\| > \epsilon \|\boldsymbol{w}^0\|$ **do**
  $\boldsymbol{z}^k = \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k$, $\boldsymbol{p}^k = \boldsymbol{P}\boldsymbol{z}^k$
  -orthogonalization of directions-
  for $i = 0, \ldots k-1$ **do**
    $\beta_i^k = \frac{\boldsymbol{p}^{i^T} \boldsymbol{F}_I \boldsymbol{p}^k}{\boldsymbol{p}^{i^T} \boldsymbol{F}_I \boldsymbol{p}^i}$, $\boldsymbol{p}^k := \boldsymbol{p}^k - \beta_i^k \boldsymbol{p}^i$
  end for

  $\eta^k = \frac{\boldsymbol{p}^{k^T} \boldsymbol{w}^k}{\boldsymbol{p}^{k^T} \boldsymbol{F}_I \boldsymbol{p}^k}$
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k \boldsymbol{p}^k$
  $\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \eta^k \boldsymbol{F}_I \boldsymbol{p}^k$
  $\boldsymbol{w}^{k+1} = \boldsymbol{P}^T \boldsymbol{r}^{k+1}$
  $k = k + 1$
end while

12.3.3 Choosing the right projection operator

We have seen in Eq. (11.21) that at every iteration step the projection of the Lagrange multipliers brings a correction proportional to $\boldsymbol{Q}\boldsymbol{G}_I$. Since $\boldsymbol{G}_I$ are the boundary restrictions of the local floating modes, $\boldsymbol{Q}$ should by a stiffness matrix, and since the correction $\boldsymbol{Q}\boldsymbol{G}_I$ represents the boundary stresses corresponding to the rigid mode displacements, $\boldsymbol{Q}$ should approximate the inverse of $\boldsymbol{F}_I$.

We can thus say that $\boldsymbol{Q}$ plays the role of a preconditioner for the coarse grid displacements. Ideally, one should take for $\boldsymbol{Q}$ the preconditioner utilized for the Conjugate Gradient iterations, namely $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_I^{-1}$. In that case, one verifies that [67]

$$\boldsymbol{P}\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{P}^T = \boldsymbol{P}\tilde{\boldsymbol{F}}_I^{-1} = \tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{P}^T \tag{12.29}$$

which means that one projection can be avoided in this case.

In practice, it is common to choose $\boldsymbol{Q} = \boldsymbol{I}$, which amounts to assuming that $\boldsymbol{F}_I$ is not very different from unity. Even for non-regular problems, one can choose $\boldsymbol{Q} = \boldsymbol{I}$ in order to avoid the cost of preconditioning $\boldsymbol{G}_I$, but making this simplification can inhibit the efficiency of a good preconditioner. Indeed, the first iteration step yields

$$\boldsymbol{p}^0 = \boldsymbol{P}\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{w}^0 \tag{12.30a}$$
$$\eta^0 = \frac{\boldsymbol{p}^{0^T}\boldsymbol{w}^0}{\boldsymbol{p}^{0^T}\boldsymbol{F}_I\boldsymbol{p}^0} \tag{12.30b}$$
$$\boldsymbol{r}^1 = \boldsymbol{r}^0 - \eta^0 \boldsymbol{F}_I \boldsymbol{p}^0 \tag{12.30c}$$

If the preconditioner is a good approximate of the inverse of $\boldsymbol{F}_I$, i.e. $\tilde{\boldsymbol{F}}_I^{-1} \simeq {\boldsymbol{F}_I}^{-1}$, and if $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_I^{-1}$, using relations (11.29) we find

$$\boldsymbol{p}^0 = \tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{P}^T\boldsymbol{w}^0 = \tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{w}^0 \tag{12.31}$$
$$\eta^0 = \frac{\boldsymbol{w}^{0^T}\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{w}^0}{\boldsymbol{w}^{0^T}\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{F}_I\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{w}^0} \simeq 1 \tag{12.32}$$
$$\boldsymbol{r}^1 \simeq \boldsymbol{r}^0 - \boldsymbol{F}_I\tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{w}^0 \simeq \boldsymbol{r}^0 - \boldsymbol{w}^0 \tag{12.33}$$
$$\boldsymbol{w}^1 = \boldsymbol{P}^T\boldsymbol{r}^1 \simeq 0 \tag{12.34}$$

showing that convergence is reached within a few iterations. If however the rigid body displacements are not preconditioned ($\boldsymbol{Q} = \boldsymbol{I}$), and if $\boldsymbol{F}_I$ is very different from $\boldsymbol{I}$ the convergence will be significantly delayed, even though the preconditioner is very close to the exact inverse operator.

12.4 Multiple right-hand sides

When an elimination method is used for solving a system of equations, the factorization of the operator can be reused to solve problems with different right-hand sides and it then requires only a forward and backward substitution which is cheap compared to the factorization. For the iterative method, one must also find a procedure that reuses the previous iterations so to decrease the cost of solving problems with different right-hand sides. This issue is primordial if we want the FETI method to be competitive for static problems, for eigensolvers with inverse iterations and for transient dynamic analysis with time-stepping.

12.4.1 Conjugate directions by projection and orthogonalization

The basic idea is to reuse the descent directions computed for previous problems and already defining a Krylov space. The method was first proposed by Saad in [163] and was applied to the FETI method in [54, 55, 58]. The method proceeds in two steps:

- Search for the solution by projection in the Krylov space available from previous solutions.
- If the residue of the projected solution is not small enough, search for the solution components in the space orthogonal to the initial Krylov space. We therefor perform conjugate iterations where the directions of descent are conjugate to the directions already included in the Krylov space.

Assuming that we have stored the l directions $\boldsymbol{p}^j,\ j = 0, \dots l-1$ obtained from previous solutions, the solution $\boldsymbol{\lambda}^{k+1}$ at iteration k is given by

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^0 + \sum_{j=0}^{l} \eta^j \boldsymbol{p}^j + \sum_{i=l}^{l+k} \eta^i \boldsymbol{p}^i \tag{12.35}$$

$\boldsymbol{\lambda}^0$ is given by (11.24) and defines a set of Lagrange multipliers that equilibrates the applied forces for floating subdomains. The directions $\boldsymbol{p}^j$ in Eq. (11.35) are self-equilibrated, namely $\boldsymbol{G}_I^T \boldsymbol{p}^j = \boldsymbol{0}$. The amplitudes η^j of the solution in the Krylov space of the previous solutions are obtained by setting $k = -1$ and by minimizing the functional ϕ given by (11.20). The directions $\boldsymbol{p}^j$ being conjugate, we find

$$\eta^j = \frac{{\boldsymbol{p}^j}^T (\boldsymbol{d} - \boldsymbol{F}_I \lambda^0)}{{\boldsymbol{p}^j}^T \boldsymbol{F}_I \boldsymbol{p}^j} \qquad j = 1, \dots l-1 \tag{12.36}$$

showing that if $\boldsymbol{F}_I \boldsymbol{p}^j$ are stored, the solution in the Krylov space is obtained by simple projection.

The dual residue associated with the Lagrange multipliers projected in the initial Krylov space is

$$\boldsymbol{w}^0 = \boldsymbol{P}^T \left(\boldsymbol{d} - \boldsymbol{F}_I \lambda^0 - \sum_{j=0}^{l} \eta^j \boldsymbol{F}_I \boldsymbol{p}^j \right) \tag{12.37}$$

The directions generated during the subsequent iteration will then be chosen orthogonal to the initial Krylov space so that ${\boldsymbol{p}^j}^T \boldsymbol{F}_I \boldsymbol{p}^i = 0$. The iteration takes thus place in a reduced space orthogonal to the previously computed directions.

The directions $\boldsymbol{p}^j$ are stored in the columns of matrix $\boldsymbol{P}_l$ which will be updated every time a new direction is generated in the conjugate gradient iteration. Moreover, we assume that the

Algorithm 12.3 Projection and orthogonalization procedure for multiple right-hand sides

$\boldsymbol{P}_l, \quad \boldsymbol{F}_l = \boldsymbol{F}_I \boldsymbol{P}_l \ : \ \boldsymbol{F}_l^T \boldsymbol{P}_l = \boldsymbol{I}$
$k = 0, \boldsymbol{\lambda}^{00} = \boldsymbol{Q}\boldsymbol{G}_I \left(\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{G}_I \right)^{-1} \boldsymbol{e}$
$\boldsymbol{r}^{00} = \boldsymbol{d} - \boldsymbol{F}_I \lambda^0$
$\boldsymbol{w}^{00} = \boldsymbol{P}^T \boldsymbol{r}^{00}$
-Initial projection-
$\boldsymbol{\eta}^0 = \boldsymbol{P}_l^T \boldsymbol{r}^0$
$\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}^{00} + \boldsymbol{P}_l \boldsymbol{\eta}^0$
$\boldsymbol{r}^0 = \boldsymbol{r}^{00} - \boldsymbol{F}_l \boldsymbol{\eta}^0$
$\boldsymbol{w}^0 = \boldsymbol{P}^T \boldsymbol{r}^0$

while $\|\boldsymbol{w}^k\| > \epsilon \|\boldsymbol{w}^{00}\|$ **do**
$\boldsymbol{z}^k = \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k$, $\boldsymbol{p}^{l+k} = \boldsymbol{P}\boldsymbol{z}^k$
-orthogonalization of directions-
$\boldsymbol{\beta}^k = \boldsymbol{F}_l^T \boldsymbol{p}^{l+k}$, $\boldsymbol{p}^{l+k} := \boldsymbol{p}^{l+k} - \boldsymbol{P}_l \boldsymbol{\beta}^k$

$\boldsymbol{p}^{l+k} = \boldsymbol{p}^{l+k} / \sqrt{\boldsymbol{p}^{l+k^T} \boldsymbol{F}_I \boldsymbol{p}^{l+k}}$
$\eta^k = \boldsymbol{p}^{l+k^T} \boldsymbol{w}^k$
$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k \boldsymbol{p}^{l+k}$
$\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \eta^k \boldsymbol{F}_I \boldsymbol{p}^{l+k}$
$\boldsymbol{w}^{k+1} = \boldsymbol{P}^T \boldsymbol{r}^{k+1}$
$k = k + 1$
-Updating $\boldsymbol{P}_l$ and $\boldsymbol{F}_l$-
$\boldsymbol{P}_l = \begin{bmatrix} \boldsymbol{P}_l & \boldsymbol{p}^{l+k} \end{bmatrix}$, $\boldsymbol{F}_l = \begin{bmatrix} \boldsymbol{F}_l & \boldsymbol{F}_I \boldsymbol{p}^{l+k} \end{bmatrix}$

end while

directions are normalized such that ${\boldsymbol{p}^j}^T \boldsymbol{F}_I \boldsymbol{p}^j = 1$. The results of $\boldsymbol{F}_I \boldsymbol{p}^j$ are stored in $\boldsymbol{F}_l = \boldsymbol{F}_I \boldsymbol{P}_l$. The projection-orthogonalization procedure is summarized in Algorithm 11.3.

Algorithm 11.3 shows that finding the solution in the space of the previously computed directions implies only matrix operations and ensuring that the iteration takes place in the space orthogonal to the previous direction can also be put in the form of simple matrix operations. Hence the projection-orthogonalization procedure is cheap and well adapted to vectorized computation. Moreover, since every subdomain stores its associated part of $\boldsymbol{P}_l$ and $\boldsymbol{F}_l$, the associated matrix operations are naturally parallel.

The effect of the projection-orthogonalization procedure is twofold. First the projection gives a good initial solution. Then the orthogonalization of subsequent iteration directions leads to iterating into a reduced space in which exact convergence is achieved after $(n_{\lambda^{eff.}} - l)$ steps and the convergence rate is now function of the highest eigenvalue in the remaining subspace. The Krylov space generated by previous solutions convergences towards the higher end of the eigenspectrum and therefore the remaining subspace has an improved condition number.

Although the dimension of the directions stored in $\boldsymbol{P}_l$ and $\boldsymbol{F}_l$ is equal to n_λ and is thus small compared to the total number of d.o.f., it may happen that there is not enough memory space for storing all successive directions. In that case, partial orthogonalization is performed on a *last in, first out* basis so that only the last column of $\boldsymbol{P}_l$ and $\boldsymbol{F}_l$ is updated.[1]

This projection-orthogonalization method has proved to be very effective in general. We expect this strategy to be especially suitable for solving static problems with multiple load cases, since in that case, the solution space for the interface stresses is generally very similar for all solutions so that making use of previously computed directions yields already very accurate results after the projection step. In practice, one finds that the solutions are obtained with the desired accuracy already after the projection when a space of l directions is available, with $l << n_{\lambda^{eff.}}$. Indeed, for practical static problems, $\boldsymbol{F}_I$ is characterized by a small number of dominant eigensolutions, and once this physically meaningful portion of the eigenspectrum of $\boldsymbol{F}_I$ is captured by the Krylov space, accurate solutions are found in this reduced subspace.

12.4.2 Multiple static load cases

When the right-hand sides are all known from the beginning on, the projection-orthogonalization method can be slightly modified. This case appears for instance in static analysis when all load cases are known (obviously, for time-stepping in dynamic analysis, this is not the case).

In that case, one can use the conjugate directions as soon as they are computed for every right-hand side. Therefore, at every conjugate iteration, we can compute the direction amplitude for each solution simultaneously. This does not fundamentally change the projection strategy, but it leads to organizing the operations in a matrix form better suited to modern computer architectures. The algorithm is then similar to the successive conjugate iterations scheme proposed in [173].

12.4.3 Solving the floating modes coarse grid

In the PCPG for the FETI method, the Lagrange multipliers are projected by $\boldsymbol{P}$ and the residue is projected by $\boldsymbol{P}^T$. These projections imply solving the floating modes coarse grid of the form

$$(\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{G}_I)\boldsymbol{\alpha}^k = \boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{r}^k \tag{12.38}$$

The dimension of this coarse grid is equal to the total number of floating modes and is thus small compared to the total number of interface variables. Nevertheless, as the number of subdomains increases, solving the coarse grid problem by an elimination method is not very efficient in terms of parallelism so that it should be solved iteratively. Since this coarse grid is solved at every iteration, it is essential that it be computed efficiently and therefore a Conjugate Gradient method is used together with the projection-orthogonalization technique. The convergence tolerance ϵ for solving

[1] A more economical strategy in terms of memory requirement can be found [137,159]. It consists in storing only the residues and the coefficients η^k and β_i^k from which the directions $\boldsymbol{p}^k$ and $\boldsymbol{F}_I \boldsymbol{p}^k$ can be deduced.

this coarse grid problem is chosen very small since the accuracy of this solution directly affects the accuracy one can achieve on the FETI iterations.

12.5 C.G. methods for semi-definite systems

In this section we address the problem of solving a semi-definite linear system. This problem arises when solving a static problem in the presence of global rigid body modes and when performing inverse iterations for computing the eigenmodes of a floating structure. First we will discuss the general problem of applying a Conjugate Gradient scheme to a singular system. We then discuss the particular case of the FETI method in the presence of global rigid body modes.

12.5.1 Computing a null space by C.G.

Let us consider the general linear problem

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{12.39}$$

We assume that $\boldsymbol{A}$ is symmetric and semi-definite and we further assume that there exist m independent non-zero solutions $\boldsymbol{u}$ satisfying

$$\boldsymbol{A}\boldsymbol{u}^i = \boldsymbol{0} \qquad i = 1, \dots m \tag{12.40}$$

The general problem (11.39) has a solution if and only if the right-hand side $\boldsymbol{b}$ is in the range of $\boldsymbol{A}$, that is if $\boldsymbol{b}$ is orthogonal to the null space $\text{span}\{\boldsymbol{u}^1, \dots \boldsymbol{u}^m\}$:

$${\boldsymbol{u}^i}^T \boldsymbol{b} = 0 \qquad i = 1, \dots m \tag{12.41}$$

If this latter condition is satisfied, a solution exist and the solution can still be computed by the usual Conjugate Gradient-like algorithms if the iterates remain out of the null space of $\boldsymbol{A}$.

When the system is solved by an elimination method, the null space of $\boldsymbol{A}$, namely the solutions $\boldsymbol{u}^i$ are responsible for zero pivots and they are computed as by-products of the temporary links strategy [91]. Unfortunately, when an iterative scheme is applied, nothing indicates the singularity of the operator and the null space is not directly accessible. In practice, the condition (11.41) should be tested before applying Conjugate Gradient iterations, otherwise one will observe a non-convergence without knowing that in fact there is no solution to the problem.

It is thus essential that before running a Conjugate Gradient on a problem that is possibly singular, one tests the regularity of the system and, if the system is singular, that one computes its null space in order to check if condition (11.41) is satisfied.

The method we propose consists in applying Conjugate Gradient iterations to the null space problem $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0}$. It is equivalent to say that we solve iteratively the problem

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} \tag{12.42}$$

where $\boldsymbol{A}\boldsymbol{y} \neq \boldsymbol{0}$. The right-hand side of this problem is trivially in the range of $\boldsymbol{A}$ and thus a solution can be found by a Conjugate Gradient. If the solution $\boldsymbol{x}$ is equal to $\boldsymbol{y}$, we can state that the operator is regular. Otherwise, we compute a null space component $\boldsymbol{u} = \boldsymbol{y} - \boldsymbol{x}$.

The delicate point in this algorithm is to determine whether the converged solution is equal to $\boldsymbol{y}$ or not. Let us decompose $\boldsymbol{y}$ in its component $\overline{\boldsymbol{y}}$ in the range of $\boldsymbol{A}$ and in its null space components:

$$\boldsymbol{y} = \overline{\boldsymbol{y}} + \sum_{i=1}^{m} b_i \boldsymbol{u}^i \tag{12.43}$$

Hence $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{A}\overline{\boldsymbol{y}}$. The conjugate gradient iterations take place in the Krylov space $\text{span}\{\boldsymbol{A}\boldsymbol{y}, \dots \boldsymbol{A}^k\boldsymbol{y}\}$ and therefore, if no preconditioning is applied, the successive directions are in the range of $\boldsymbol{A}$.

Therefore $\boldsymbol{x}$ converges towards $\overline{\boldsymbol{y}}$ and if the convergence tolerance is set small enough, we can write

$$\boldsymbol{y} - \boldsymbol{x} \simeq \sum_{i=1}^{m} b_i \boldsymbol{u}^i \tag{12.44}$$

We then assume that in $\boldsymbol{y}$ the null space components are well enough represented so that $\| \sum_{i=1}^{m} b_i \boldsymbol{u}^i \| \geq \|\boldsymbol{y}\|/N$, where N is the dimension of $\boldsymbol{y}$. Hence we will state that $\boldsymbol{x}$ is not equal to $\boldsymbol{y}$, namely that $\boldsymbol{x} - \boldsymbol{y}$ yields a null space direction if $\|\boldsymbol{x} - \boldsymbol{y}\| > \|\boldsymbol{y}\|/N$.

When a null space direction is found, a new Conjugate Gradient iteration must be performed with $\boldsymbol{y}$ orthogonal to the already found null directions until no new null direction is found. When the Conjugate Gradient iteration is restarted in order to check for remaining null directions, one uses the projection-orthogonalization technique.

The entire method is summarized in Algorithm 11.4 where we assume that the successive directions are normalized and where for simplicity's sake, no preconditioner is used.

Let us note that:

- In order to detect the null space, the vector $\boldsymbol{y}$ should on one hand contain enough components in the range of $\boldsymbol{A}$ and on the other hand it should contain enough components of the null space. In practice, we built $\boldsymbol{y}$ based on the diagonal coefficients of $\boldsymbol{A}$.

- The conjugate directions generated in this algorithm for checking the singularity of the operator can be efficiently reused for solving the actual problem (11.39) by using the projection-orthogonalization technique.

- If a preconditioner is introduced in the iterations, one must project the preconditioned direction in the space orthogonal to the null space directions already found so not to reintroduce them in the iterations.

- This algorithm is easy to implement since it mainly consists in adding an outer loop to the Conjugate Gradient iterations.

If we assume that $\boldsymbol{A}$ is for instance the assembled stiffness matrix, the strategy presented here indicates how a system with global rigid body modes should be treated when a global iteration scheme is used.

12.5.2 C.G. for the dual interface problem of floating structures

The dual interface problem (11.19) can be singular due to two different reasons. First, the operator $\boldsymbol{F}_I$ can be singular due to redundant constraints. The null space of $\boldsymbol{F}_I$ is then equal to the null space of $\boldsymbol{B}^T$ and the equation

$$\boldsymbol{F}_I \boldsymbol{\lambda} + \boldsymbol{G}_I \boldsymbol{\alpha} = \boldsymbol{d} \tag{12.45}$$

has a solution since according to their definitions (3.18b–3.18d) $\boldsymbol{G}_I$ and $\boldsymbol{d}$ are in the range of $\boldsymbol{B}$. The solution for $\boldsymbol{\lambda}$ is not unique, but the corresponding interface loads $\boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}$ are unique and so are the displacements. The Preconditioned Conjugate Projected Gradient can then be applied if we note that preconditioned and projected directions are never parallel to the null space of $\boldsymbol{B}^T$.

A second reason for the dual interface problem (11.19) to be singular arises when global floating modes exist for the entire structure. Indeed, we will now show that the presence of global rigid body modes implies the singularity of $\boldsymbol{G}_I$.

Let us assume that $\boldsymbol{G}_I$ is not full column rank and that there exist m non-zero solutions $\boldsymbol{\theta}^i$ in the space of the local floating mode amplitudes $\boldsymbol{\alpha}$ such that

$$\boldsymbol{G}_I \boldsymbol{\theta}^i = 0 \qquad i = 1, \dots m \tag{12.46}$$

Algorithm 12.4 Checking for regularity and computing the null space by a Preconditioned Conjugate Gradient

$l = 0$, $\boldsymbol{P}_l = [\,]$, $\boldsymbol{F}_l = [\,]$
$m = 0$, $\boldsymbol{y} : \boldsymbol{A}\boldsymbol{y} \neq \boldsymbol{0}$,
$flag_{nsp} = 1$
while $flag_{nsp} = 1$ **do**
  $k = 0$, $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{y}$,
  -Projection-
    $\boldsymbol{\eta}^0 = \boldsymbol{P}_l^T \boldsymbol{b}$
    $\boldsymbol{x}^0 = \boldsymbol{P}_l \boldsymbol{\eta}^0$, (or $\boldsymbol{x}^0 = \boldsymbol{0}$ if $l = 0$)
    $\boldsymbol{r}^0 = \boldsymbol{b} - \boldsymbol{F}_l \boldsymbol{\eta}^0$, (or $\boldsymbol{r}^0 = \boldsymbol{b}$ if $l = 0$)

  while $\|\boldsymbol{r}^k\| > \epsilon \|\boldsymbol{b}\|$ **do**
    $\boldsymbol{p}^{l+k} = \boldsymbol{r}^k$
    -orthogonalization of directions-
      $\boldsymbol{\beta}^k = \boldsymbol{F}_l^T \boldsymbol{p}^{l+k}$
      $\boldsymbol{p}^{l+k} = \boldsymbol{p}^{l+k} - \boldsymbol{P}_l \boldsymbol{\beta}^k$

    $\boldsymbol{p}^{l+k} = \boldsymbol{p}^{l+k} / \sqrt{\boldsymbol{p}^{l+k^T} \boldsymbol{A} \boldsymbol{p}^{l+k}}$
    $\eta^k = \boldsymbol{p}^{l+k^T} \boldsymbol{r}^k$
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \eta^k \boldsymbol{p}^{l+k}$
    $\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \eta^k \boldsymbol{A} \boldsymbol{p}^{l+k}$
    $k = k + 1$
    -Updating $\boldsymbol{P}_l$ and $\boldsymbol{F}_l$-
      $\boldsymbol{P}_l = \left[\boldsymbol{P}_l \;\; \boldsymbol{p}^{l+k}\right]$, $\boldsymbol{F}_l = \left[\boldsymbol{F}_l \;\; \boldsymbol{A}\boldsymbol{p}^{l+k}\right]$

  end while

  -Check for null directions-
    if $\|\boldsymbol{x}^k - \boldsymbol{y}\| > \|\boldsymbol{y}\|/N$ **then**
      $m = m + 1$
      $\boldsymbol{u}^m = \boldsymbol{x}^k - \boldsymbol{y}$
      -Normalize null direction-
        $\boldsymbol{u}^m = \boldsymbol{u}^m / \sqrt{\boldsymbol{u}^{m^T} \boldsymbol{u}^m}$

      -Restart iteration-
        $\boldsymbol{y} : \boldsymbol{A}\boldsymbol{y} \neq \boldsymbol{0}$
        for $i = 1, \ldots m$ **do**
          $\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{u}^i (\boldsymbol{u}^{i^T} \boldsymbol{y})$
        end for

    else
      $flag_{nsp} = 0$
    end if

end while

From the definition (3.18b) of $\boldsymbol{G}_I$, (11.46) can be written explicitly as

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{R}^{(s)} \boldsymbol{\theta}^{(s),i} = 0 \qquad i = 1, \dots m \tag{12.47}$$

where $\boldsymbol{\theta}^{(s),i}$ is the subset of $\boldsymbol{\theta}^i$ pertaining to subdomain $\Omega^{(s)}$. This last relation expresses that there are m linear combinations of local rigid body modes that satisfy the interface compatibility. Calling $\boldsymbol{u}^i = \boldsymbol{R}^{(s)} \boldsymbol{\theta}^{(s),i}$ the combinations of local floating modes, we can write

$$\boldsymbol{K}^{(s)} \boldsymbol{u}^{(s),i} = \boldsymbol{0} \qquad s = 1, \dots N_s \tag{12.48a}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s),i} = \boldsymbol{0} \qquad i = 1, \dots m \tag{12.48b}$$

The first equation (11.48a) expresses that all subdomains are in a zero stress state and (11.48b) stipulates the interface compatibility. Hence, the system of equations (11.48) is the subdomain form of the global rigid body mode equations and $\boldsymbol{u}^{(s),i}$ are the subdomain restrictions of the rigid body modes $\boldsymbol{u}^i$.

We thus state that when no global rigid body modes exist, $\boldsymbol{G}_I$ is full column rank and $\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{G}_I$ is regular. For a structure with global rigid body modes, $\boldsymbol{G}_I$ is not full rank and its rank deficiency is equal to the number of global floating modes. The dual interface problem is then singular and the self-equilibrium condition on the Lagrange multipliers

$$\boldsymbol{G}_I^T \boldsymbol{\lambda} = \boldsymbol{e} \tag{12.49}$$

can be satisfied only if $\boldsymbol{e}$ is orthogonal to all $\boldsymbol{\theta}^i$. From the definition (3.18e) of $\boldsymbol{e}$ we see that a solution exist only if

$$\boldsymbol{\theta}^{i^T} \boldsymbol{e} = \sum_{s=1}^{N_s} \boldsymbol{\theta}^{(s),i^T} \boldsymbol{R}^{(s)^T} \boldsymbol{f}^{(s)} \tag{12.50a}$$

$$= \sum_{s=1}^{N_s} \boldsymbol{u}^{(s),i^T} \boldsymbol{f}^{(s)} \qquad i = 1, \dots m \tag{12.50b}$$

$$= \boldsymbol{u}^{i^T} \boldsymbol{f} = 0 \tag{12.50c}$$

This condition expresses the self-equilibrium of the external forces with respect to the global rigid body modes. We thus conclude that *the self-equilibrium of the subdomains under the action of the Lagrange multipliers and of the external loads can be satisfied if and only if the external loads are self-equilibrated with respect to the global floating modes.*

Hence, when performing a static analysis, or when applying an inverse iteration to possibly floating structures, one must first determine the global rigid body modes by computing the null space $\boldsymbol{\theta}^i$ of $\boldsymbol{G}_I$, and then, one must check for the self-equilibrium of the applied loads by verifying that $\boldsymbol{e}$ is orthogonal to $\boldsymbol{\theta}^i$. To determine the null space of $\boldsymbol{G}_I$, we note that $\boldsymbol{\theta}^i$ is also the null space of the floating modes coarse grid operator:

$$\boldsymbol{G}_I \boldsymbol{\theta}^i = 0 \qquad \Longleftrightarrow \qquad (\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{G}_I) \boldsymbol{\theta}^i = 0 \qquad i = 1, \dots m \tag{12.51}$$

This result holds if $\boldsymbol{Q}$ is regular. The null space $\boldsymbol{\theta}^i$ can thus be computed by applying a preliminary iteration as explained in the previous section to the floating modes coarse grid. This requires only a minor modification to the iteration scheme for the coarse grid, and the conjugate directions computed in this preliminary step will be used for all subsequent solves. Hence, checking for the regularity of the coarse grid and computing the global rigid modes does incur only a negligible over-cost.

As an example, we consider a square plane stress structure made of 80×80 elements and decomposed into 8×8 subdomains. No boundary conditions are applied to the structure and

therefore 3 global rigid body modes exist. Every subdomain has 3 local floating modes and the associated coarse grid problem is of dimension 192. We compute the null space $\boldsymbol{\theta}^i$ by applying Algorithm 11.4. In Fig. 11.1 we show the convergence curves for the 4 successive conjugate gradient iterations. The first 3 Conjugate Gradients yield the null space $\boldsymbol{\theta}^i$ such that $\boldsymbol{R}^{(s)}\boldsymbol{\theta}^{(s),i}, i = 1, 2, 3$ are the domain restrictions of the two translational and the rotational rigid body modes of the global structure. The fourth Conjugate Gradient iteration checks for the regularity of the coarse grid in the subspace orthogonal to $\boldsymbol{\theta}^i$. On Fig. 11.1 we see that the convergence is monotonic and that thanks to the projection-orthogonalization strategy, the conjugate directions computed for determining the first null direction are re-used for the subsequent runs so that the last 2 null directions are found after only a few iterations.

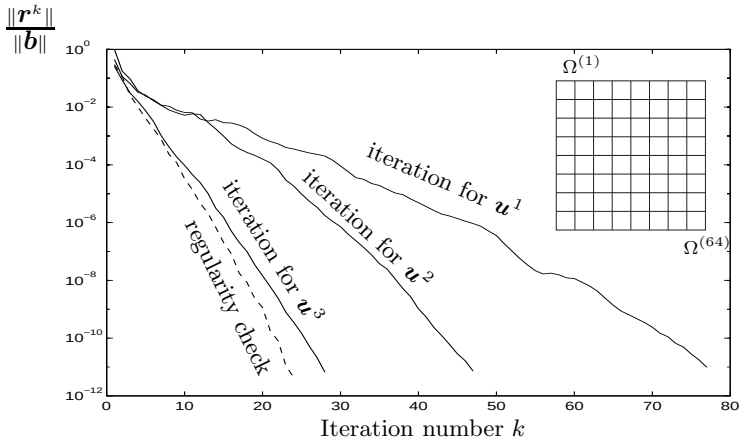


Figure 12.1: Convergence of the Conjugate Gradient for solving the floating modes coarse grid and computing the global rigid body modes

12.6 Auxiliary coarse grid for strong compatibility constraints

For some particular problems, it is important to ensure that exact compatibility is enforced for certain interface d.o.f. which are singular points and which therefore degrade the conditioning of the interface problem. It has been recognized for instance that a corner point for a bi-harmonic problem (plates and shells) is a point of singularity and that convergence of iterative schemes applied to the primal or dual interface problem is badly affected by the presence of so-called Kirchhoff corner forces [78, 124, 176].

In this section we will briefly present the way how a subset of constraints can be strongly enforced at every FETI iteration and we will discuss how the FETI iterations can be organized in order to minimize the computational cost of solving the related auxiliary coarse grid. For a detailed discussion of the method we refer to [72, 73, 78, 124, 176].

12.6.1 Coarse problem for a subset of constraints

Let us assume that we want to strongly enforce a subset of the constraints defined by

$$\boldsymbol{C}^T\boldsymbol{B}\boldsymbol{u}_d = \boldsymbol{C}^T \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{12.52}$$

where $\boldsymbol{C}$ is a matrix of dimension $n_\lambda \times n_c$, n_c being the number of constraints we want to enforce explicitly at every FETI iteration. For instance, when one wants to enforce the corner compatibility

in a shell problem, $\boldsymbol{C}$ is a Boolean matrix pertaining to the set of corner compatibility constraints. At every FETI iteration, one explicitly adds to every Lagrange multiplier iterate the Lagrange multipliers associated with $\boldsymbol{C}$, namely the interface loads taking care of the constraint subset.

For the initial guess of the Lagrange multipliers, the PCPG iterations define $\boldsymbol{\lambda}^0 = \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^0$ where $\boldsymbol{\gamma}^0$ is determined such that $\boldsymbol{G}_I^T\boldsymbol{\lambda}^0 = \boldsymbol{e}$. This yields the expression (11.24). We now require that the initial Lagrange multipliers are such that the constraint subset are satisfied, namely that the initial dual residue $\boldsymbol{w}^0$ is such that

$$\boldsymbol{C}^T\boldsymbol{w}^0 = \boldsymbol{0} \tag{12.53}$$

Hence we set

$$\boldsymbol{\lambda}^0 = \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^0 + \boldsymbol{C}\boldsymbol{\mu}^0 \tag{12.54}$$

where the second term represents the Lagrange multipliers associated with the constraints subset that are to be enforced strongly at every FETI iteration. The coefficients $\boldsymbol{\gamma}^0$ and $\boldsymbol{\mu}^0$ are determined by imposing that $\boldsymbol{\lambda}^0$ satisfies the self-equilibrium and (11.53):

$$\boldsymbol{G}_I^T\boldsymbol{\lambda}^0 = \boldsymbol{e} \tag{12.55a}$$

$$\boldsymbol{C}^T(\boldsymbol{d} - \boldsymbol{F}_I\boldsymbol{\lambda}^0 - \boldsymbol{G}_I\boldsymbol{\alpha}^0) = \boldsymbol{0} \tag{12.55b}$$

and the amplitudes $\boldsymbol{\alpha}^0$ of the local floating modes in (11.55b) is determined by projecting the dual problem onto the subspace of $\boldsymbol{Q}\boldsymbol{G}_I$ (see (11.26)):

$$\boldsymbol{G}_I^T\boldsymbol{Q}(\boldsymbol{d} - \boldsymbol{F}_I\boldsymbol{\lambda}^0 - \boldsymbol{G}_I\boldsymbol{\alpha}^0) = \boldsymbol{0} \tag{12.56}$$

The initial Lagrange multipliers are thus found by solving the problem deduced by substituting (11.54) in (11.55b), (11.56) and (11.55a) respectively:

$$\boldsymbol{N}\begin{bmatrix} \boldsymbol{\mu}^0 \\ \boldsymbol{\gamma}^0 \\ \boldsymbol{\alpha}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{d} \\ \boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix} \tag{12.57}$$

where $\boldsymbol{N}$ is defined by

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{F}_I\boldsymbol{C} & \boldsymbol{C}^T\boldsymbol{F}_I\boldsymbol{Q}\boldsymbol{G}_I & \boldsymbol{C}^T\boldsymbol{G}_I \\ \boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{F}_I\boldsymbol{C} & \boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{F}_I\boldsymbol{Q}\boldsymbol{G}_I & \boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{G}_I \\ \boldsymbol{G}_I^T\boldsymbol{C} & \boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{G}_I & \boldsymbol{0} \end{bmatrix} \tag{12.58}$$

The problem (11.57) is a new coarse problem which includes the local floating modes coarse grid but which also implies the Lagrange multipliers associated to $\boldsymbol{C}$. The coarse grid (11.57) determines the local floating mode amplitudes $\boldsymbol{\alpha}^0$, the amplitudes $\boldsymbol{\gamma}^0$ of the Lagrange multipliers $\boldsymbol{Q}\boldsymbol{G}_I$ and the amplitudes $\boldsymbol{\mu}^0$ of the Lagrange multipliers associated with the constraint subset such that self-equilibrium is satisfied and such that the compatibility is exactly satisfied for the subset of constraints defined by $\boldsymbol{C}$. It is a genuine coarse grid problem in that it amounts to solving a reduced problem over the entire structure.

For the successive iterates in the FETI iterations we then assume that every direction of descent is complemented by the Lagrange multipliers of the constraints subset and by the stresses corresponding to local floating modes:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k\Delta\boldsymbol{\lambda}^k \tag{12.59a}$$

$$\text{where} \qquad \Delta\boldsymbol{\lambda}^k = \boldsymbol{p}^k + \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^k + \boldsymbol{C}\boldsymbol{\mu}^k \tag{12.59b}$$

and where $\boldsymbol{p}^k$ is the preconditioned direction of descent. The coefficients $\boldsymbol{\gamma}^k$ and $\boldsymbol{\mu}^0$ are again determined by imposing that the self-equilibrium is satisfied and that the dual residue is zero for the constraints subset:

$$\boldsymbol{G}_I^T\boldsymbol{\lambda}^{k+1} = \boldsymbol{e} \tag{12.60a}$$

$$\boldsymbol{C}^T(\boldsymbol{d} - \boldsymbol{F}_I\boldsymbol{\lambda}^{k+1} - \boldsymbol{G}_I\boldsymbol{\alpha}^{k+1}) = \boldsymbol{0} \tag{12.60b}$$

and the floating modes amplitude $\boldsymbol{\alpha}^{k+1}$ is obtained by

$$\boldsymbol{G}_I^T \boldsymbol{Q}(\boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{\lambda}^{k+1} - \boldsymbol{G}_I \boldsymbol{\alpha}^{k+1}) = \boldsymbol{0} \tag{12.61}$$

Taking account of (11.55a), and substituting (11.59) in (11.60b), (11.61) and (11.60a), we obtain the coarse grid problem

$$\boldsymbol{N} \begin{bmatrix} \boldsymbol{\mu}^k \\ \boldsymbol{\gamma}^k \\ \Delta\boldsymbol{\alpha}^k \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}^T \boldsymbol{w}^k / \eta^k \\ \boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{w}^k / \eta^k \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{C}^T \boldsymbol{F}_I \boldsymbol{p}^k \\ -\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{F}_I \boldsymbol{p}^k \\ -\boldsymbol{G}_I^T \boldsymbol{p}^k \end{bmatrix} \tag{12.62}$$

where

$$\Delta\boldsymbol{\alpha}^k = \frac{\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k}{\eta^k} \tag{12.63}$$

$$\boldsymbol{w}^k = \boldsymbol{d} - \boldsymbol{F}_I \lambda^k - \boldsymbol{G}_I \boldsymbol{\alpha}^k \tag{12.64}$$

Considering that the successive dual residues $\boldsymbol{w}^k$ satisfy (11.60b) and (11.61), the first term of the right-hand side in the coarse problem (11.62) is null.

When the coarse problem (11.62) has been solved, we can compute $\Delta\boldsymbol{w}^k$ defined by

$$\begin{aligned} \Delta\boldsymbol{w}^k &= -\boldsymbol{F}_I(\boldsymbol{p}^k + \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^k + \boldsymbol{C}\boldsymbol{\mu}^k) - \boldsymbol{G}_I\Delta\boldsymbol{\alpha}^k \\ &= -\boldsymbol{F}_I\Delta\boldsymbol{\lambda}^k - \boldsymbol{G}_I\Delta\boldsymbol{\alpha}^k \end{aligned} \tag{12.65}$$

such that the dual residue can be written as

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \eta^k \Delta\boldsymbol{w}^k \tag{12.66}$$

The coefficient η^k is then determined as before by minimizing the functional $\phi(\boldsymbol{\lambda}^{k+1})$, namely by ensuring that the dual residue $\boldsymbol{w}^{k+1}$ is orthogonal to the descent direction $\Delta\boldsymbol{\lambda}^k$:

$$\eta^k = -\frac{(\Delta\boldsymbol{\lambda}^k)^T \boldsymbol{w}^k}{(\Delta\boldsymbol{\lambda}^k)^T \Delta\boldsymbol{w}^k} \tag{12.67}$$

We can now write the FETI iterations when an auxiliary coarse problem is defined as in Algorithm 11.5. This algorithm generalizes the PCPG algorithm described in Algorithm 11.2: if we set $\boldsymbol{C} = \boldsymbol{0}$ in $\boldsymbol{N}$, solving the coarse problem is totally equivalent to projecting the descent directions by $\boldsymbol{P}$ and projecting the residue by $\boldsymbol{P}^T$ as in Algorithm 11.2. The formulation presented here is similar to the method proposed in [72, 73, 78, 124]. However the algorithm as presented therein includes redundant projection steps whereas in Algorithm 11.5 we fully exploit the results of the coarse grid in order to avoid unnecessary projection by $\boldsymbol{P}$.

Remarks

- Usually, one chooses $\boldsymbol{Q} = \boldsymbol{I}$ so to avoid the cost of computing $\boldsymbol{Q}\boldsymbol{G}_I$ (see discussion in Section 11.3.3).
- In Algorithm 11.5, the symbol "$\longrightarrow$" indicates that in practice the results $\Delta\boldsymbol{\lambda}^k$ and $\Delta\boldsymbol{w}^k$ need not to be computed but result from the solution of the coarse problem.
- The way Algorithm 11.5 is organized indicates that a projection-orthogonalization strategy as presented in Section 11.4 can be applied in a straightforward manner for problems with multiple right-hand sides: one should store the direction of descent $\Delta\boldsymbol{\lambda}^k$ and the dual residue increments $\Delta\boldsymbol{w}^k$.
- The same methodology for building an auxiliary coarse grid can be applied to the primal formulation, namely to the balancing method [125, 176].

Algorithm 12.5 FETI iterations with an auxiliary coarse problem

$k = 0$

solve $\quad \boldsymbol{N} \begin{bmatrix} \boldsymbol{\mu}^0 \\ \boldsymbol{\gamma}^0 \\ \boldsymbol{\alpha}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}^T \boldsymbol{d} \\ \boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix}$

$\longrightarrow \boldsymbol{\lambda}^0 = \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^0 + \boldsymbol{C}\boldsymbol{\mu}^0$

$\longrightarrow \boldsymbol{w}^0 = \boldsymbol{d} - \boldsymbol{F}_I\boldsymbol{\lambda}^0 - \boldsymbol{G}_I\boldsymbol{\alpha}^0$

while $\|\boldsymbol{w}^k\| > \epsilon\|\boldsymbol{w}^0\|$ **do**

$\boldsymbol{p}^k = \tilde{\boldsymbol{F}}_I^{-1}\boldsymbol{w}^k$

-orthogonalization of directions-

for $i = 0, \ldots k-1$ **do**

$\beta_i^k = \frac{(\Delta\boldsymbol{w}^i)^T\boldsymbol{p}^k}{(\Delta\boldsymbol{w}^i)^T\Delta\boldsymbol{\lambda}^i}$, $\boldsymbol{p}^k := \boldsymbol{p}^k - \beta_i^k\Delta\boldsymbol{\lambda}^i$

end for

solve $\quad \boldsymbol{N} \begin{bmatrix} \boldsymbol{\mu}^k \\ \boldsymbol{\gamma}^k \\ \Delta\boldsymbol{\alpha}^k \end{bmatrix} = \begin{bmatrix} -\boldsymbol{C}^T\boldsymbol{F}_I\boldsymbol{p}^k \\ -\boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{F}_I\boldsymbol{p}^k \\ -\boldsymbol{G}_I^T\boldsymbol{p}^k \end{bmatrix}$

$\longrightarrow \Delta\boldsymbol{\lambda}^k = \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^k + \boldsymbol{C}\boldsymbol{\mu}^k$

$\longrightarrow \Delta\boldsymbol{w}^k = -\boldsymbol{F}_I\Delta\boldsymbol{\lambda}^k - \boldsymbol{G}_I\Delta\boldsymbol{\alpha}^k$

$\eta^k = -\frac{(\Delta\boldsymbol{\lambda}^k)^T\boldsymbol{w}^k}{(\Delta\boldsymbol{\lambda}^k)^T\Delta\boldsymbol{w}^k}$

$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k\Delta\boldsymbol{\lambda}^k$

$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \eta^k\Delta\boldsymbol{w}^k$

$k = k + 1$

end while

12.6.2 Solving the auxiliary coarse grid

The coarse grid problem (11.62) that must be solved at every FETI iteration can be put in the form

$$\begin{bmatrix} \boldsymbol{T}^T\boldsymbol{F}_I\boldsymbol{T}^T & \boldsymbol{T}^T\boldsymbol{G}_I \\ \boldsymbol{G}_I\boldsymbol{T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^k \\ \boldsymbol{\gamma}^k \end{bmatrix} \\ \Delta\boldsymbol{\alpha}^k \end{bmatrix} = \begin{bmatrix} -\boldsymbol{T}^T\boldsymbol{F}_I\boldsymbol{p}^k \\ -\boldsymbol{G}_I^T\boldsymbol{p}^k \end{bmatrix} \tag{12.68}$$

where

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{Q}\boldsymbol{G}_I \end{bmatrix} \tag{12.69}$$

This form was proposed in [73,78]: it shows that the auxiliary coarse grid problem can be treated as a reduced secondary FETI problem and hence the FETI algorithm can be applied in a recursive manner, the inner FETI corresponding to the coarse grid and being solved up to a precision higher than the outer FETI iterations. Therefore, this method is also known as the *two-level FETI method*.

Another way to understand how the auxiliary coarse problem should be solved can be found by condensing the variables $\boldsymbol{\gamma}^k$ and $\Delta\boldsymbol{\alpha}^k$ in the system (11.62). Considering the last and then the second set of equations in the coarse grid problem (11.62) we successively obtain

$$\boldsymbol{\gamma}^k = -\left(\boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{G}_I\right)^{-1}\boldsymbol{G}_I^T(\boldsymbol{p}^k + \boldsymbol{C}\boldsymbol{\mu}^k) \tag{12.70a}$$

$$\Delta\boldsymbol{\alpha}^k = -\left(\boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{G}_I\right)^{-1}\boldsymbol{G}_I^T\boldsymbol{Q}\boldsymbol{F}_I(\boldsymbol{p}^k + \boldsymbol{C}\boldsymbol{\mu}^k + \boldsymbol{Q}\boldsymbol{G}_I\boldsymbol{\gamma}^k) \tag{12.70b}$$

Substituting these results in the first set of equations of the coarse problem (11.62), we find the alternative form

$$(\boldsymbol{P}\boldsymbol{C})^T\boldsymbol{F}_I(\boldsymbol{P}\boldsymbol{C})\boldsymbol{\mu}^k = (\boldsymbol{P}\boldsymbol{C})^T\boldsymbol{F}_I\boldsymbol{P}\boldsymbol{p}^k \tag{12.71}$$

From (11.71) we observe that the coarse grid corresponds to a dual interface problem in the subspace of $\boldsymbol{PC}$. The coarse grid can thus be solved by a simple Conjugate Gradient scheme in a subspace of small dimension and with a projection-orthogonalization strategy. The FETI method with an auxiliary coarse problem is therefore solved by two imbricate efficient and parallel iteration loops.

If we now substitute the results (11.70a) and (11.71) in the expression (11.59b) of $\Delta\boldsymbol{\lambda}^k$ we find successively

$$\begin{aligned} \Delta\boldsymbol{\lambda}^k &= \boldsymbol{P}(\boldsymbol{p}^k + \boldsymbol{C}\boldsymbol{\mu}^k) \\ &= \boldsymbol{A}\boldsymbol{P}\boldsymbol{p}^k \end{aligned} \tag{12.72}$$

where

$$\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{PC}\left((\boldsymbol{PC})^T \boldsymbol{F}_I (\boldsymbol{PC})\right)^{-1} (\boldsymbol{PC})^T \boldsymbol{F}_I \tag{12.73}$$

Substituting now $\Delta\boldsymbol{\alpha}^k$ from (11.70b) into the definition (11.65) of $\Delta\boldsymbol{w}^k$ we find

$$\Delta\boldsymbol{w}^k = -\boldsymbol{P}^T \boldsymbol{F}_I \Delta\boldsymbol{\lambda}^k \tag{12.74}$$

Finally, combining (11.72) and (11.74), and noting from definition (11.73) that $\boldsymbol{F}_I\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{F}_I\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{F}_I$, we find

$$\Delta\boldsymbol{w}^k = -\left(\boldsymbol{P}^T\boldsymbol{A}^T\boldsymbol{F}_I\boldsymbol{A}\boldsymbol{P}\right)\boldsymbol{p}^k \tag{12.75}$$

From the definition (11.21b) of $\boldsymbol{P}$, we see that $\boldsymbol{G}_I^T\Delta\boldsymbol{w}^k = \boldsymbol{0}$ and from the definition (11.73) of $\boldsymbol{A}$, it can be verified that $\boldsymbol{C}^T\Delta\boldsymbol{w}^k = \boldsymbol{0}$. We thus conclude that in the FETI method with an auxiliary coarse grid, the iterations take place in a subspace orthogonal to the floating modes and to the subset of constraints. The convergence of the FETI iterations are thus driven by the condition number of the operator $\boldsymbol{F}_I$ projected in this subspace. Therefore, when a few constraints are responsible for the bad conditioning of the interface problem (e.g. corner compatibility in shells), we will include them in the auxiliary coarse grid in order to ensure a good FETI convergence.

Chapter 13

FETI preconditioners for homogeneous structures

Preconditioners for the FETI method are fundamentally different from classical preconditioners: when global iteration is applied to the entire system, preconditioners are based on approximation techniques for the inverse of the global operator (Jacobi, incomplete Cholesky ...), whereas for domain decomposition algorithms when only the interface system is solved iteratively, preconditioning is based on the local operators.

We will first discuss the basic preconditioners suitable for fairly homogeneous problems. The Dirichlet and lumped preconditioners were introduced in [67, 79] and will be shortly presented here. We then discuss the mechanical consistency of these preconditioners and propose to add a scaling to the basic methods in order to take account of interface multiplicity when defining an interface stress correction in the preconditioning step.

13.1 The Dirichlet preconditioner

The dual interface operator $\boldsymbol{F}_I$ arises from the dual assembly of the interface flexibilities and it appears natural to approximate the inverse of $\boldsymbol{F}_I$ by assembling the interface stiffnesses, namely the subdomains Schur complements:

$$\left(\sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \boldsymbol{b}^{(s)^T} \right)^{-1} \boldsymbol{w}^k \simeq \left(\sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \right) \boldsymbol{w}^k \tag{13.1}$$

This amounts to solving local Dirichlet problems (see next section) and is thus called the Dirichlet preconditioner:

$$\tilde{\boldsymbol{F}}_{I,D}^{-1} = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \tag{13.2}$$

This preconditioner is inherently parallel since it approximates the inverse of the sum of the interface flexibilities by the sum of the inverse. From a mathematical view point it is readily understood that this approximation is licit when all components of the sum have the same order of magnitude, but when the interface flexibilities are very different between subdomains, the approximation (12.1) is far from accurate. Therefore, the preconditioners discussed in this chapter will be suitable for homogeneous problems and specific modifications to the basic preconditioners for heterogeneous structures will be discussed in Chapter 15.

The idea behind the Dirichlet preconditioner can be transposed to the primal method to define

the Neumann preconditioner [175]:

$$\left(\sum_{s=1}^{N_s} \boldsymbol{L}_b^{(s)^T} \boldsymbol{S}^{(s)} \boldsymbol{L}_b^{(s)}\right)^{-1} \simeq \left(\sum_{s=1}^{N_s} \boldsymbol{L}_b^{(s)^T} \boldsymbol{S}^{(s)^+} \boldsymbol{L}_b^{(s)}\right) = \tilde{\boldsymbol{S}}^{-1} \tag{13.3}$$

and an auxiliary problem similar to the rigid body coarse grid for FETI must be associated to this Neumann preconditioner in order to determine the local rigid displacements (see Section 4.3 and [125]). The coarse grid problem associated to this Neumann preconditioner is essential for defining a scalable scheme and the resulting method is known as the *balancing method* [125].

Making the assumption of regular discretizations and uniform mesh partitions, the primal method for elasticity problems with the Neumann preconditioner and the balancing method was shown to have a condition number of the order of [125]

$$\kappa(\boldsymbol{S}^{-1}\boldsymbol{S}) = O\left(1 + \log\frac{H}{h}\right) \tag{13.4}$$

where h is the element size and H is the subdomain characteristic dimension.

The condition number of the FETI operator for Poisson or plane stress/strain problems without preconditioning is bounded by [67]

$$\kappa(\boldsymbol{P}^T\boldsymbol{F}_I\boldsymbol{P}) \leq Cst\frac{H}{h} \tag{13.5}$$

where Cst is a constant. This result can be understood if we note that the high spectrum of $\boldsymbol{F}_I$ is related to the lower eigenvalues of the local operators (hence proportional to the subdomain size H) while the lower modes of $\boldsymbol{F}_I$ correspond to the mesh modes function of the mesh size h.

By analogy with the condition number (12.4) of the balanced domain decomposition method one can show that the FETI method with the Dirichlet preconditioner applied to homogeneous problems with regular discretization and uniform mesh partition has a condition number of [65]

$$\kappa(\tilde{\boldsymbol{F}}_{I,D}^{-1}\boldsymbol{P}^T\boldsymbol{F}_I\boldsymbol{P}) = O\left(1 + \log^\beta\frac{H}{h}\right), \qquad \beta \leq 3 \tag{13.6}$$

The conditioning results (12.5) shows that the FETI method is inherently scalable with respect to the number of subdomains (which is related to $1/H$). This property is a consequence of the floating modes coarse grid included in FETI and does not apply to the primal Schur complement method. When a Dirichlet preconditioner is added to the iteration operator, (12.6) highlights the numerical scalability of the FETI method with respect to the mesh size h and with respect to the number of subdomains: the condition number of the FETI method decreases when the number of subdomains in increased and grows only weakly with the mesh refinement.

The numerical scalability of the FETI is obtained thanks to the rigid body coarse grid (the same can be said about the numerical scalability of the balanced method). Since the coarse grid ensuring the numerical scalability for FETI has a small dimension, the cost of the solution of the coarse grid is marginal so that numerical scalability induces larger speedups when the number of processors is increased. This property is known as parallel scalability. From this discussion, it can be stated that the Dirichlet preconditioner is optimum for FETI. The optimality of the FETI method for bi-harmonic problems (plates and shells) is obtained by adding an auxiliary coarse grid problem associated with the corner forces (Section 11.6).

To illustrate the effect of the Dirichlet preconditioner on the eigenspectrum of the interface operator, let us consider the simple plane stress problem of a square domain decomposed into 4 square substructures, each subdomain being modeled by 20×20 elements (Fig. 12.1). We show in Fig. 12.2 the eigenspectrum of the non preconditioned operator $\boldsymbol{P}^T\boldsymbol{F}_I\boldsymbol{P}$ and the eigenspectrum of $\tilde{\boldsymbol{F}}_{I,D}^{-1}\boldsymbol{P}^T\boldsymbol{F}_I\boldsymbol{P}$.[1]

[1] In all eigenspectra shown in this text, we do not consider the zero eigenvalues for the sake of clarity and because they do not belong to the iteration space anyway.

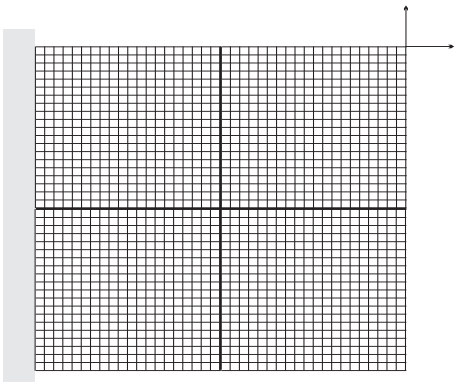


Figure 13.1: Plane stress problem decomposed into 4 subdomains

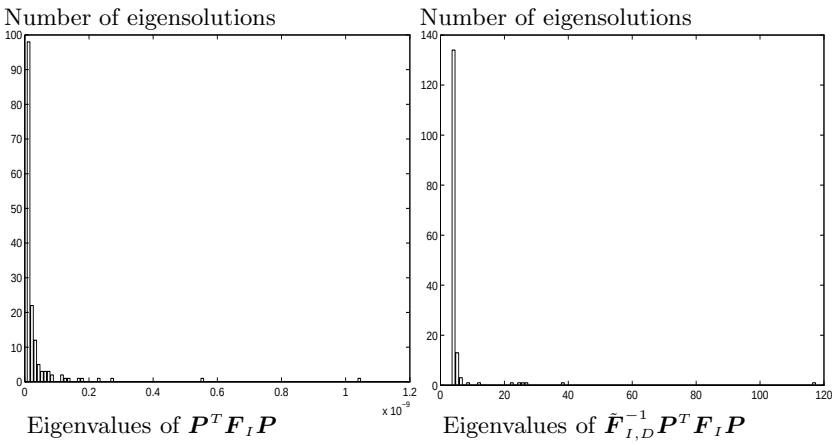


Figure 13.2: Eigenspectrum of the dual operator with and without the Dirichlet preconditioner

Comparing in Fig. 12.2 the eigenspectrum with and whitout preconditioning, we observe that the effect of the preconditioner is to group the eigenvalues around the value 4 and thus to produce a clustered eigenspectrum. As a result, we get a tremendous improvement of the condition number when the optimal Dirichlet preconditioner is used.

13.2 Mechanically consistent Dirichlet preconditioner - Scaling procedure

The role of a preconditioner at iteration $k+1$ of the PCPG is to build a correction to the Lagrange multipliers given an interface compatibility error, i.e. a displacement jump $\boldsymbol{w}^k$.

If we first consider the case of two subdomains and the associated displacement fields $\boldsymbol{u}^{(1),k}$ and $\boldsymbol{u}^{(2),k}$ such that $\boldsymbol{w}^k = \boldsymbol{b}^{(1)}\boldsymbol{u}_b^{(1),k} + \boldsymbol{b}^{(2)}\boldsymbol{u}_b^{(2),k}$, we can in a first step build a compatible interface displacement by defining an average value as proposed in [47, 77] (see Section 5.4):

$$\hat{\boldsymbol{u}}_b^{(1),k} = \frac{1}{2}\left(\boldsymbol{u}_b^{(1),k} - \boldsymbol{b}^{(1)^T}\boldsymbol{b}^{(2)}\boldsymbol{u}_b^{(2),k}\right) \tag{13.7a}$$

$$\hat{\boldsymbol{u}}_b^{(2),k} = \frac{1}{2}\left(\boldsymbol{u}_b^{(2),k} - \boldsymbol{b}^{(2)^T}\boldsymbol{b}^{(1)}\boldsymbol{u}_b^{(1),k}\right) \tag{13.7b}$$

Note that the operator $-\boldsymbol{b}^{(s)^T}\boldsymbol{b}^{(r)}$ simply expresses the correspondence between the numbering of the interface d.o.f. in $\Omega^{(s)}$ and $\Omega^{(r)}$. The interface displacement corrections are

$$\Delta\boldsymbol{u}_b^{(1),k} = \hat{\boldsymbol{u}}_b^{(1),k} - \boldsymbol{u}_b^{(1),k} = -\frac{1}{2}\boldsymbol{b}^{(1)^T}\boldsymbol{w}^k \tag{13.8a}$$

$$\Delta\boldsymbol{u}_b^{(2),k} = \hat{\boldsymbol{u}}_b^{(2),k} - \boldsymbol{u}_b^{(2),k} = -\frac{1}{2}\boldsymbol{b}^{(2)^T}\boldsymbol{w}^k \tag{13.8b}$$

If we assume that the internal d.o.f. are modified to satisfy the internal equilibrium when the interface d.o.f. are corrected, we have

$$\Delta\boldsymbol{u}_i^{(s),k} = -\boldsymbol{K}_{ii}^{(s)^{-1}}\boldsymbol{K}_{ib}^{(s)}\Delta\boldsymbol{u}_b^{(s),k} \tag{13.9}$$

The interface stress corrections on Γ_I^{12} on both sides are then computed by

$$\Delta\boldsymbol{f}_b^{(s),k} = \boldsymbol{S}_{bb}^{(s)}\Delta\boldsymbol{u}_b^{(s),k} \tag{13.10}$$

In general $\Delta\boldsymbol{f}^{(1),k}$ and $\Delta\boldsymbol{f}^{(2),k}$ are not identical so that the interface stress is not uniquely defined. Hence, the interface stress correction will be chosen as an average in the same way as we defined the interface displacement average:

$$\boldsymbol{z}^{k+1} = -\frac{1}{2}\left(\boldsymbol{b}^{(1)}\Delta\boldsymbol{f}_b^{(1),k} + \boldsymbol{b}^{(2)}\Delta\boldsymbol{f}_b^{(2),k}\right) \tag{13.11}$$

This direction of descent for iteration $k+1$ is exactly equivalent to the direction resulting from the Dirichlet preconditioner: combining (12.8) and (12.11), we find

$$\boldsymbol{z}^{k+1} = \frac{1}{4}\sum_{s=1}^{2}\boldsymbol{b}^{(s)}\boldsymbol{S}_{bb}^{(s)}\boldsymbol{b}^{(s)^T}\boldsymbol{w}^k \tag{13.12}$$

When more than 2 subdomains interconnect on an interface, the averaging scheme can be generalized (see Eq. (7.43)): on an edge j of the interface $\Gamma_I^{(s),j}$ connecting m_j subdomains indexed r (Fig. 12.3), the compatible displacement is the average value

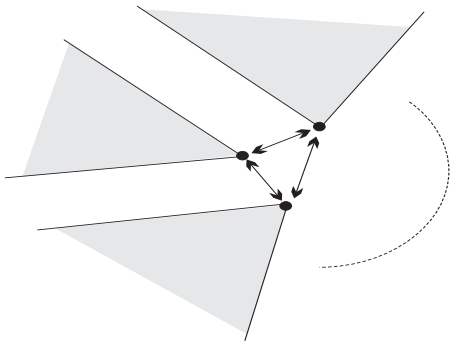


Figure 13.3: Lagrange multipliers on an interface of multiplicity > 2

$$\hat{\boldsymbol{u}}_b^{(s),j^k} = \frac{1}{m_j}\left(\boldsymbol{u}_b^{(s),j^k} - \sum_{\substack{r:\Gamma_I^{(r)}\ni j \\ r\neq s}} \boldsymbol{b}^{(s,-),j^T}\boldsymbol{b}^{(r,-),j}\boldsymbol{u}_b^{(r),j^k}\right) \tag{13.13}$$

where m_j is the multiplicity of the edge and where we use the notations of Section 7.1.

Calling $\boldsymbol{w}^{sr,j^k}$ the part of the dual error $\boldsymbol{w}^k$ pertaining to the compatibility between $\Omega^{(s)}$ and $\Omega^{(r)}$ on the edge j, the displacement correction on Γ_I^j writes

$$\begin{aligned}
\Delta\boldsymbol{u}_b^{(s),j^k} &= \hat{\boldsymbol{u}}_b^{(s),j^k} - \boldsymbol{u}_b^{(s),j^k} = \hat{\boldsymbol{u}}_b^{(s),j^k} - \boldsymbol{b}^{(s,-),j^T}\boldsymbol{b}^{(s,-),j}\boldsymbol{u}_b^{(s),j^k} \\
&= -\frac{1}{m_j}\sum_{\substack{r:\Gamma_I^{(r)}\ni j \\ r\neq s}} \boldsymbol{b}^{(s,-),j^T}\left(\boldsymbol{b}^{(r,-),j}\boldsymbol{u}^{(r),j^k} + \boldsymbol{b}^{(s,-),j}\boldsymbol{u}^{(s),j^k}\right) \\
&= -\frac{1}{m_j}\sum_{\substack{r:\Gamma_I^{(r)}\ni j \\ r\neq s}} \boldsymbol{b}^{(s,-),j^T}\boldsymbol{w}^{sr,j^k} \\
&= -\frac{1}{m_j}\boldsymbol{b}^{(s),j^T}\boldsymbol{w}^k
\end{aligned} \tag{13.14}$$

where we use the notations introduced in Section 7.1. The associated force corrections on the subdomain boundaries are then

$$\Delta\boldsymbol{f}_b^{(s)^k} = \boldsymbol{S}_{bb}^{(s)}\Delta\boldsymbol{u}_b^{(s)^k} \tag{13.15}$$

The corrections for the Lagrange multiplier, i.e. the new descent direction for the PCG, is then given by an averaged redistribution of the force correction onto the Lagrange multipliers:

$$\boldsymbol{z}^{j^{k+1}} = \frac{-1}{m_j}\sum_{r:\Gamma_I^{(r)}\ni j} \boldsymbol{b}^{(r),j}\ \Delta\boldsymbol{f}_b^{(r),j^k} \tag{13.16}$$

In doing so, we ensure that whenever the averaged displacement is equal to the exact solution, the correction of the Lagrange multipliers leads to the exact interface forces on each subdomain. To prove this point, we will assume that the averaged solution is equal to the exact solution and we will show that in that case the interface forces resulting from the averaged interface Lagrange multipliers (12.16) are the exact interface loads.

Assuming that the averaged solution $\hat{\boldsymbol{u}}_b^{(s),j^k}$ is the exact solution satisfying the interface compatibility *and* the interface equilibrium, we write that the associated interface force corrections

have a zero resultant:

$$\sum_{\substack{r:\Gamma_I^{(r)}\ni j \\ r\neq s}} \boldsymbol{b}^{(s),j^T}\boldsymbol{b}^{(r),j}\Delta\boldsymbol{f}_b^{(r),j^k} + \Delta\boldsymbol{f}_b^{(s),j^k} = \boldsymbol{0} \tag{13.17}$$

The load corrections on edge $\Gamma_I^{(s),j}$ associated with the averaged Lagrange multiplier correction are

$$\boldsymbol{b}^{(s),j^T}\boldsymbol{z}^{j^{k+1}} = -\frac{1}{m_j}\boldsymbol{b}^{(s),j^T}\sum_{r:\Gamma_I^{(r)}\ni j}\boldsymbol{b}^{(r),j}\Delta\boldsymbol{f}_b^{(r),j^k} \tag{13.18}$$

Noting that $\boldsymbol{b}^{(s),j^T}\boldsymbol{b}^{(s),j} = m_j - 1$ (see Eq. (7.7)), and taking account of (12.17), we find

$$\begin{aligned}\boldsymbol{b}^{(s),j^T}\boldsymbol{z}^{j^{k+1}} &= -\frac{1}{m_j}\sum_{\substack{r:\Gamma_I^{(r)}\ni j \\ r\neq s}}\boldsymbol{b}^{(s),j^T}\boldsymbol{b}^{(r),j}\Delta\boldsymbol{f}_b^{(r),j^k} - \frac{1}{m_j}(m_j-1)\Delta\boldsymbol{f}_b^{(s),j^k} &(13.19)\\ &= \Delta\boldsymbol{f}_b^{(s),j^k} &(13.20)\end{aligned}$$

which proves that when the interface loads are averaged as in (12.16), the interface stresses are exact when the interface displacements are exact.

From Eqs. (12.14) and (12.16) we obtain the expression of the preconditioned direction

$$\boldsymbol{z}^{j^{k+1}} = \boldsymbol{A}\sum_{s=1}^{N_s}\boldsymbol{b}^{(s)}\boldsymbol{S}_{bb}^{(s)^+}\boldsymbol{b}^{(s)^T}\boldsymbol{A}\boldsymbol{w}^k \tag{13.21}$$

where $\boldsymbol{A}$ is a scaling diagonal matrix representing the weighting by the multiplicities associated with the Lagrange multipliers.

From the discussion above, we observe that in order to be mechanically consistent, the Dirichlet preconditioner as expressed in equation (12.2) should be implemented by scaling first the compatibility residue $\boldsymbol{w}^k$ by the interface multiplicities, and by applying a second scaling onto the descent direction $\boldsymbol{z}^{k+1}$. This scaling procedure is inexpensive and it is our experience that it reduces the number of FETI iterations by 10 to 20 percent for two-dimensional and three-dimensional problems.

Remarks

- The mechanical interpretation of the Dirichlet preconditioner makes it clear that for interfaces of multiplicity $m_i > 2$, a Lagrange multiplier must be associated with each pair of intersecting subdomains even though it might define redundant constraints. It is well known that if the set of interface constraints is not complete, the Dirichlet preconditioner is not efficient. The mechanical reasons for this requirement are that (a.) the displacement jump between each subdomain must be stored in $\boldsymbol{w}^k$ in order to express the displacement correction by the simple expression (12.14), (b.) the distribution (12.16) of the load correction onto the Lagrange multipliers must be equivalent for each subdomain.

- The same reasoning can be applied to the Neumann preconditioner in the primal method. The Neumann preconditioner (12.3) is then modified by applying a scaling exactly as for the Dirichlet preconditioner (see the role of matrix $\boldsymbol{D}^{(s)}$ in the developments of Section 4.3).

To illustrate the effect of the scaling procedure proposed in this section, let us once more consider the eigenspectrum of the dual problem for a square plane stress problem decomposed into 4 square subdomains (Fig. 12.1) when a scaled Dirichlet preconditioner is applied. From Fig. 12.4 and comparing with the case when no scaling is applied (Fig. 12.2), we observe that the

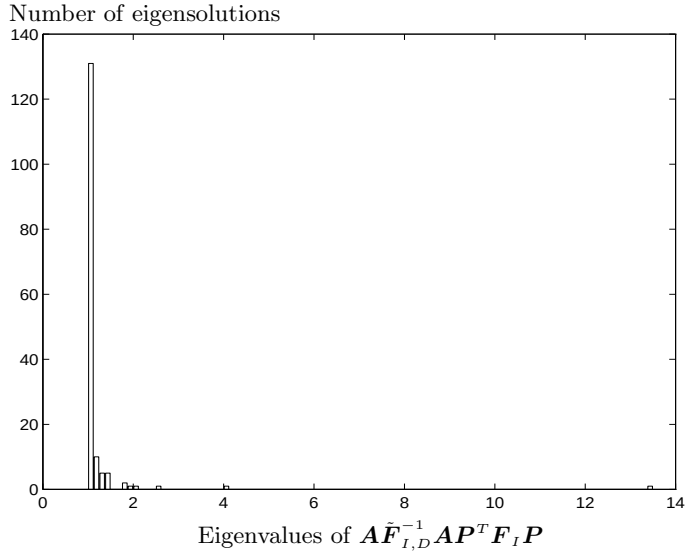


Figure 13.4: Eigenspectrum of a dual operator with the scaled Dirichlet preconditioner

clustering of the eigenvalues around 1 is nearly perfect when scaling is used, except for a small number of isolated eigenvalues slightly detached from the set of eigenvalues of the order of 1. The fact that the majority of the eigenvalues are equal to 1 whereas the cluster of eigenvalues was centered on the value 4 without scaling (Fig. 12.2) clearly demonstrates that the scaling procedure introduced when analyzing the mechanical meaning of the preconditioner allows to define a better approximation of the inverse of $\boldsymbol{F}_I$.

13.3 Lumped preconditioner

The Dirichlet preconditioner discussed in Sections 12.1 and 12.2 is optimal but expensive: the expression (12.1) requires the solution of local Dirichlet problems of the form

$$\begin{bmatrix} \boldsymbol{K}_{ii}^{(s)} & \boldsymbol{K}_{ib}^{(s)} \\ \boldsymbol{K}_{bi}^{(s)} & \boldsymbol{K}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}_i^{(s)} \\ \boldsymbol{b}^{(s)^T}\boldsymbol{w}^k \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \Delta\boldsymbol{f}_b^{(s),k} \end{bmatrix} \tag{13.22}$$

which consists in computing the internal d.o.f. when the (scaled) compatibility residue is imposed as interface displacement and in finding the interface reaction forces $\Delta\boldsymbol{f}_b^{(s),k}$. These local Dirichlet problems thus require factorizing the stiffness matrix pertaining to the internal d.o.f., namely $\boldsymbol{K}_{ii}^{(s)}$. The factorization of $\boldsymbol{K}_{ii}^{(s)}$ could be deduced from the factorization of the local operator $\boldsymbol{K}^{(s)}$ if all the interface d.o.f. were numbered last in the elimination scheme. Retaining the interface d.o.f. when factorizing $\boldsymbol{K}^{(s)}$ is generally incompatible with an optimum numbering and may thus result in a significant increase of the factorization cost. In practice, it is cheaper to perform two separate optimized factorizations for $\boldsymbol{K}^{(s)}$ and for $\boldsymbol{K}_{ii}^{(s)}$ [67]. Another significant cost of the Dirichlet preconditioner is then the forward and backward substitution relative to $\boldsymbol{K}_{ii}^{(s)}$ at every preconditioning.

A low cost preconditioner can be found if we consider that the internal equilibrium of the subdomains is not satisfied when the displacement correction is applied onto the interface. This means that the averaged internal solution is set to $\hat{\boldsymbol{u}}_i^{(s)} = \boldsymbol{u}_i^{(s)}$ and substituting $\Delta\boldsymbol{u}_i^{(s)} = \boldsymbol{0}$ in (12.22), we find

$$\Delta\boldsymbol{f}_b^{(s),k} = \boldsymbol{K}_{bb}^{(s)}\boldsymbol{b}^{(s)^T}\boldsymbol{w}^k \tag{13.23}$$

The Dirichlet preconditioner (12.2) is then replaced by

$$\tilde{\boldsymbol{F}}_{I,L}^{-1} = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{K}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \tag{13.24}$$

where the subscript L indicates that the Schur complements are approximated by the stiffness matrices *lumped* on the interface.

The preconditioner $\tilde{\boldsymbol{F}}_{I,L}^{-1}$ is clearly inexpensive since it requires only at every C.G. iteration multiplying the interface compatibility error $\boldsymbol{b}^{(s)^T} \boldsymbol{w}^k$ by the submatrix $\boldsymbol{K}_{bb}^{(s)}$ which is readily available.

Following the same mechanical interpretation as in section 12.2, where $\boldsymbol{S}_{bb}^{(s)}$ is now replaced by its lumped counterpart, we find that the mechanically consistent lumped preconditioner writes

$$\tilde{\boldsymbol{F}}_{I,L}^{-1} = \boldsymbol{A} \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{K}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{A} \tag{13.25}$$

where $\boldsymbol{A}$ is as before a diagonal scaling matrix containing $1/m_j$, m_j being the multiplicities of the interface edges. As for the Dirichlet preconditioner, the scaling leads to a reduction of 10 to 20 percent of the iteration numbers when applied to the lumped preconditioner for a marginal computing overhead.

Let us consider one last time the academic problem of the square plane stress structure (Fig. 12.1): Fig. 12.5 represents the eigenspectrum of the dual interface operator when the scaled lumped preconditioner (12.25) is applied. Comparing Fig. 12.5 and Fig. 12.4 we clearly see that

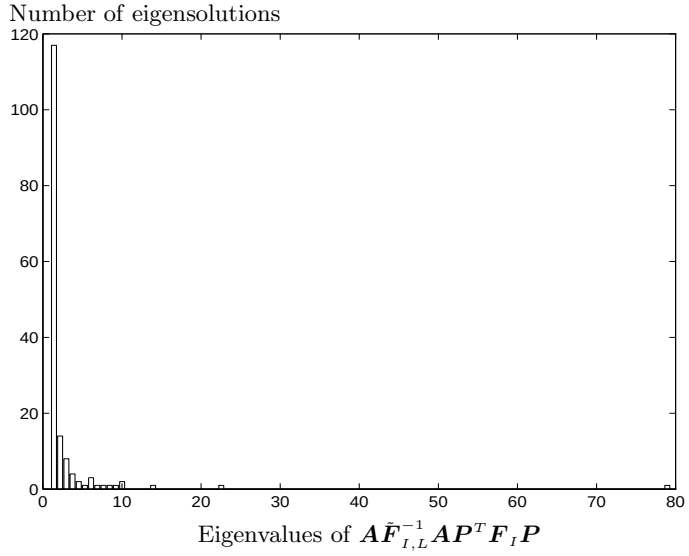


Figure 13.5: Eigenspectrum of a dual operator with the scaled lumped preconditioner

the eigenspectrum for the lumped preconditioner is slightly more spread around the unit eigenvalue, but yet the clustering is significant and the convergence of the FETI method is generally found to be acceptable with this preconditioner.

Remark

- From now on, whenever we talk about Dirichlet and lumped preconditioners we will implicitly assume that a scaling procedure is included.

- For the primal Schur complement method, it is not possible to define a lumped preconditioner as presented here. Indeed, lumping the local operators in the Neumann preconditioner would mean to replace the generalized inverse $\boldsymbol{K}^{(s)+^T}$ by the inverse of the lumped operator $\boldsymbol{K}_{bb}^{(s)^{-1}}$, the lumped operator being regular. Hence, one would loose the rigid body coarse grid problem related to the Neumann preconditioner. The latter being essential for the Neumann method to be properly balanced and thus to ensure scalability.

As outlined in [67], the lumped preconditioner is globally more efficient in most practical applications: compared to the Dirichlet preconditioner, the decrease of the convergence rate, that is the increase of FETI iterations is generally largely compensated by the reduction of the cost of one preconditioned iteration. The reason why the lumped version of the preconditioner works well for a large number of applications will now be discussed.

13.4 Understanding the efficiency of the lumped preconditioner

The fact that the lumped preconditioner works well for the FETI method can be explained as follows.

The behavior of the interior of the subdomains plays an important role for determining the overall interface solution whereas the local non-smooth stress distribution is determined by the mechanical behavior of the subdomains in the vicinity of the interface. At the beginning of the FETI iterations, the dual and primal residues are high and have a smooth distribution along the interfaces: the interface corrections associated with the first iterates should thus account for the global behavior of the subdomains. After a few iterations, the remaining interface residue consists in wriggles and the corrections can be computed by taking into account only the elements in the vicinity of Γ_I. From there, one expects the Dirichlet preconditioner to be more efficient especially for the first FETI iterations and once the low eigenspectrum content of the solution has been found, the lumped preconditioner becomes nearly as efficient as the Dirichlet.

This can be observed for instance for the simple plane stress structure shown in Fig. 12.6. Fig. 12.7 shows the initial dual residue $\boldsymbol{w}^0$ corresponding to a interface load $\boldsymbol{\lambda}^0$. On the same figure we also show the difference between $\boldsymbol{\lambda}^0$ and the exact interface loads, and the load corrections estimated by the Dirichlet and the lumped preconditioners. This simple example demonstrates that in order to obtain a good estimate of the interface load corrections the Dirichlet preconditioner should be used. The lumped preconditioner assumes that all the interior d.o.f. are fixed and the new direction for the FETI iteration is thus far from the exact load error. Remember however that the very low eigencontent of the solution is already solved for in the rigid body coarse grid at every FETI iteration so that the remaining smooth part of the solution to be computed during the iterations is related to a small number of high eigenvalues of $\boldsymbol{F}_I$ (see for instance the few isolated

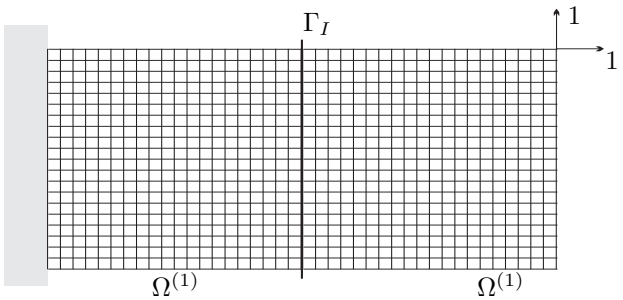


Figure 13.6: Simple two-subdomain problem

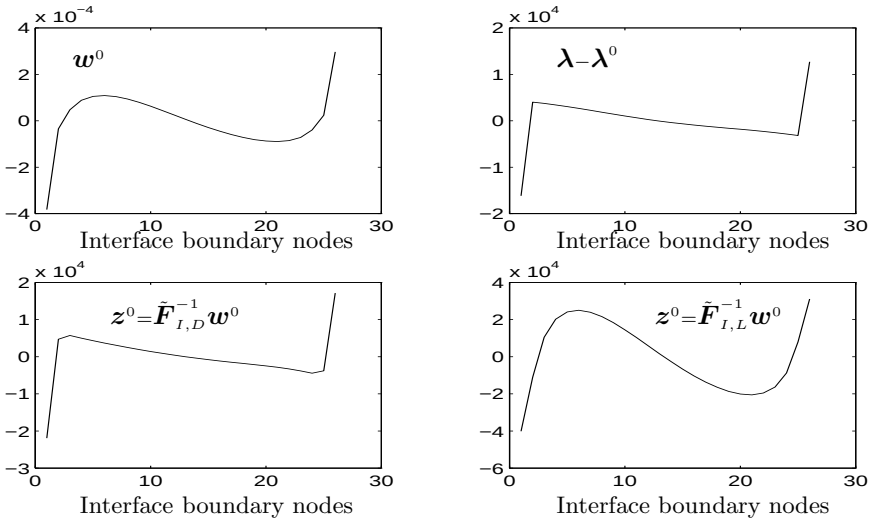


Figure 13.7: Interface load prediction by the Dirichlet and lumped preconditioners at the initial step (horizontal components)

eigenvalues in Fig. 12.5, the higher extremity of the spectrum being eliminated by the projector $\boldsymbol{P}$).

If we now consider the interface compatibility error at a further iteration, e.g. $\boldsymbol{w}^{10}$, we observe on Fig. 12.8 that the interface load of the lumped preconditioner is very similar to the correction defined by the Dirichlet preconditioner. This happens every time the interface residue is oscillatory: at a small distance of the interface these wriggles vanish and thus only the elements in the vicinity of Γ_I are relevant.

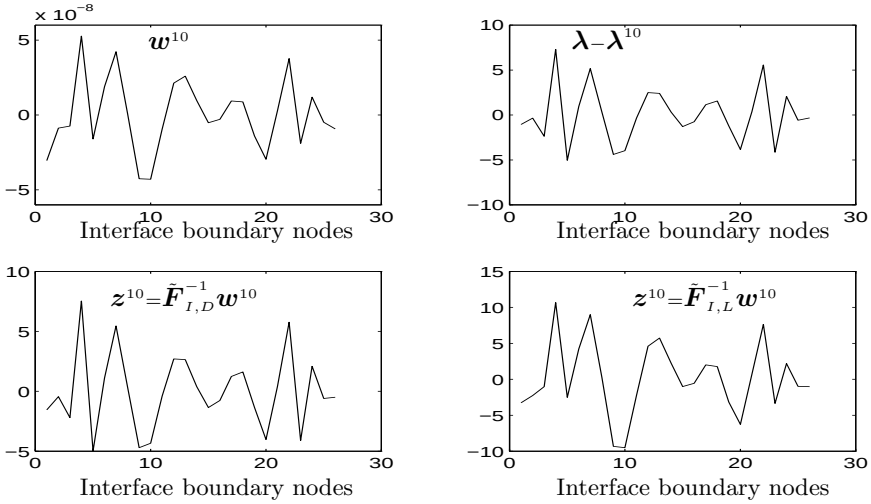


Figure 13.8: Interface load prediction by the Dirichlet and lumped preconditioners at iteration 10 (horizontal components)

In practice, for simple structures (that is for homogeneous structures with regular decompo-

sitions), the lumped preconditioner leads to a number of iterations slightly higher than when the Dirichlet preconditioner is used, but the lumped preconditioner is much cheaper and is thus more efficient overall.

13.5 Preconditioning the floating modes coarse grid problem?

As explained in Section 11.3, the projector $\boldsymbol{P}$ ensures that at every iteration the Lagrange multipliers satisfy the self-equilibrium relative to floating subdomains, and it was shown that its general expression is

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}\boldsymbol{G}_I \left(\boldsymbol{G}_I^T \boldsymbol{G}_I \right)^{-1} \boldsymbol{G}_I \tag{13.26}$$

We argued that the symmetric definite matrix $\boldsymbol{Q}$ should be taken as the identity matrix in order to minimize the computation cost relative to the coarse grid solution. We also showed in Section 11.3.3 that in order to define a consistent coarse grid and to take full advantage of a highly efficient preconditioner $\tilde{\boldsymbol{F}}_I^{-1}$ one should use $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_I^{-1}$.

Having introduced the optimal Dirichlet and the cheap lumped preconditioners, we will rediscuss the preconditioning of the coarse grid problem and analyze in which condition it should be employed.

Let us reconsider the eigenspectrum of the Dirichlet and lumped preconditioner (Fig. 12.4 and Fig. 12.5) for the square plane stress problem decomposed into 4 subdomains (Fig. 12.1). We observe that beside the cluster of eigenvalues in the vicinity of 1 there exists an isolated eigenvalue at the high end of the spectrum. This isolated eigenvalue significantly increases the condition number of the FETI operator. Calling $\boldsymbol{\lambda}^{n_\lambda}$ the eigensolution associated to the highest eigenvalue, we plot in Fig. 12.9 the associated loads $\boldsymbol{f}_b^{(3),n_\lambda} = \boldsymbol{b}^{(3)^T} \boldsymbol{\lambda}^{n_\lambda}$ on the boundary of subdomain $\Omega^{(3)}$ for instance. We clearly observe that the eigensolution $\boldsymbol{\lambda}^{n_\lambda}$ corresponds to boundary forces associated with the rotation rigid body modes of subdomains $\Omega^{(3)}$ and $\Omega^{(4)}$ and their intensity is particularly high on the nodes in the direct vicinity of the cross-point. This indicates that the interface loads associated with the coarse grid correction are not well evaluated during the FETI iterations.

We then include the preconditioners in the rigid body coarse grid problem, namelly by setting $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,D}^{-1}$ and $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,L}^{-1}$ for the Dirichlet and the lumped preconditioner respectively. The preconditioners $\tilde{\boldsymbol{F}}_{I,D}^{-1}$ and $\tilde{\boldsymbol{F}}_{I,L}^{-1}$ are symmetric semi-definite due to possible redundant constraints and thus to the rank deficiency of $\boldsymbol{B}^{(s)}$. Since $\boldsymbol{Q}$ should be symmetric definite, is it licit to set $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,D}^{-1}$ (or $\tilde{\boldsymbol{F}}_{I,L}^{-1}$)? To answer this question we note that the coarse grid operator can be put in the form

$$\boldsymbol{G}_I^T \tilde{\boldsymbol{F}}_{I,D}^{-1} \boldsymbol{G}_I = \boldsymbol{R}_d^T \boldsymbol{B}^T \tilde{\boldsymbol{F}}_{I,D}^{-1} \boldsymbol{B} \boldsymbol{R}_d \tag{13.27}$$

where we use the block diagonal notations with $\boldsymbol{R}_d = \mathrm{diag}(\boldsymbol{R}^{(s)})$ and $\boldsymbol{B} = [\boldsymbol{B}^{(1)} ... \boldsymbol{B}^{(N_s)}]$. Noting that the null space of $\tilde{\boldsymbol{F}}_{I,D}^{-1}$ and $\tilde{\boldsymbol{F}}_{I,L}^{-1}$ is the null space of $\boldsymbol{B}^T$ and that the preconditioners are definite in the image of $\boldsymbol{B}$, we conclude that the preconditioners qualify for defining a definite coarse grid.

From Eq. (12.27), we see that the blocks of the coarse grid operator are computed by

$$[\boldsymbol{G}_I^T \tilde{\boldsymbol{F}}_{I,D}^{-1} \boldsymbol{G}_I]_{sr} = \boldsymbol{R}_b^{(s)^T} \boldsymbol{b}^{(s)^T} \left(\sum_{q=1}^{N_s} \boldsymbol{b}^{(q)} \boldsymbol{S}_{bb}^{(q)} \boldsymbol{b}^{(q)^T} \right) \boldsymbol{b}(r) \boldsymbol{R}^{(r)} \tag{13.28}$$

If we recall that $\boldsymbol{b}^{(s)^T} \boldsymbol{b}^{(q)} = \boldsymbol{0}$ if subdomains $\Omega^{(s)}$ and $\Omega^{(q)}$ do not connect on Γ_I (i.e. they are not neighbors), the form (12.28) shows that computing $\tilde{\boldsymbol{F}}_{I,D}^{-1} \boldsymbol{G}_I$ requires solving a Dirichlet problem only for the rigid body modes of the neighboring subdomains. Therefore, the computation of $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,D}^{-1}$ is a parallel task which implies only a small computational cost.

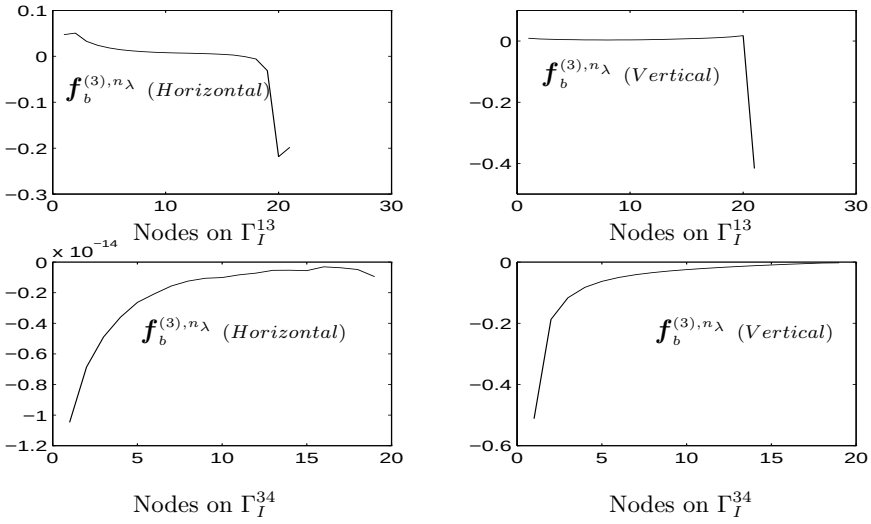


Figure 13.9: Loads on boundary of $\Omega^{(3)}$ associated with the highest eigenvalue of $\boldsymbol{A}\tilde{\boldsymbol{F}}_{I,D}^{-1}\boldsymbol{A}\boldsymbol{P}^T\boldsymbol{F}_I\boldsymbol{P}$

The eigenspectrum of the plane stress square plane when including the preconditioners in the coarse grid is as plotted in Fig. 12.10.

The results in Fig. 12.10 indicate that including the preconditioner in the coarse grid of the rigid body modes does barely affect the eigenspectrum of the lumped preconditioned operator, but for the Dirichlet preconditioner it shifts the isolated high frequency to a value close to the clustered eigenvalues. We thus deduce from this example two important facts:

- The mechanical behavior of a substructured system in the neighborhood of a cross-point can be correctly approximated only with the optimal Dirichlet preconditioner. This is a direct consequence of the fact that in the Dirichlet preconditioner the interior equilibrium is satisfied in the subdomains and thus the strong coupling between boundary d.o.f. via the internal d.o.f. near the cross-point is well represented. On the opposite, for the lumped preconditioner, the internal d.o.f. are assumed to be fixed and the mechanical behavior of the interface at cross-points is severely altered.

- When the cross-point plays an important role in determining the solution of the interface problem, the Dirichlet preconditioner should be used in conjunction with a preconditioning of the floating modes coarse grid, i.e. one should take $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,D}^{-1}$. This happens for instance every time a cross-point corresponds to a mechanical singularity such as a point of stress concentration or a corner of a plate (see Section 11.6). Let us recall that the additional cost for computing $\tilde{\boldsymbol{F}}_{I,D}^{-1}\boldsymbol{G}_I$ is small since the columns of $\boldsymbol{G}_I$ comprise the interface restrictions of the floating modes $\boldsymbol{B}^{(s)}\boldsymbol{R}^{(s)}$ which is non-zero only for the subdomains connected to $\Omega^{(s)}$.

We thus conclude that for the problems for which the Dirichlet preconditioner is optimum, that is for homogeneous structures with regular decompositions, the choice $\boldsymbol{Q} = \boldsymbol{I}$ is generally acceptable and moreover the lumped variant is in most cases more efficient in terms of overall computing time. But whenever the cross-point is a point of singularity, the Dirichlet preconditioner should be used and the coarse grid of the local floating modes should include the Dirichlet preconditioner by setting $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,D}^{-1}$.

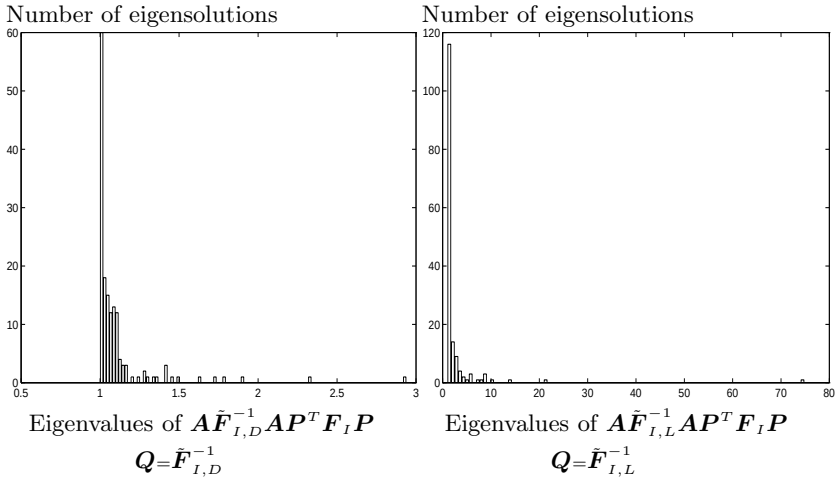


Figure 13.10: Eigenspectrum when preconditioners included in coarse grid

Chapter 14

Impact of subdomain and element aspect ratio on FETI convergence

The optimality of the Dirichlet preconditioner has been proved mathematically in [65] for regular discretizations and uniform mesh partitions. These assumptions are generally not verified in practice and it is thus important to understand how the FETI method operates on irregular meshings and non-uniform subdomain decompositions. In particular we will analyze the effect of subdomain and finite element aspect ratios on the convergence of the FETI iterations.

14.1 Effect of subdomain aspect ratio

The adverse effect of bad subdomain aspect ratios on the convergence of the FETI iterations has already been highlighted in [67, chap.7]. To analyze the very origin of poor conditioning arising from bad subdomain aspect ratios, we will first study the eigenspectrum of problems with bad aspect ratios and then we will discuss the underlying mechanical reasons.

14.1.1 Eigenspectrum of the dual operator

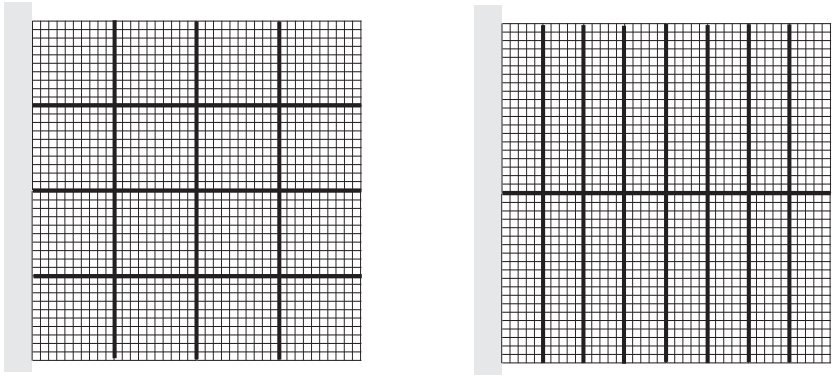
Let us consider a plane stress structure modeled by 40×40 four noded elements: first we assume a uniform partition in 4×4 subdomains (Fig. 13.1(a)), then we use a 8×2 decomposition which defines subdomains of aspect ratios of 1/4 (Fig. 13.1(b)). The eigenspectrum of the dual operator with the Dirichlet and lumped preconditioners are shown in Fig. 13.2 for the two different decompositions.

Analyzing the eigenspectrum of the dual operator given in Fig. 13.2, we observe that several isolated eigenvalues appear and move towards the high end of the eigenspectrum when the aspect ratio of the subdomains is changed from 1 to 1/4. The cluster of eigenvalues around 1 representing the mesh modes remain unchanged. As a consequence the condition number of the FETI iteration will increase. We plot in Fig. 13.3 the subdomain displacement mode related to the highest isolated eigenvalue appearing when the aspect ratio is equal to 1/4. The displacement mode depicted in Fig. 13.3 corresponds to the lower bending mode of a substructure. It is thus clear that the high isolated eigenvalues for the FETI operator appearing when subdomains have bad aspect ratios are related to the local low energy displacement modes.

Let us now investigate the mechanical reasons underlying the adverse effect of subdomains aspect ratios on the iteration of FETI.

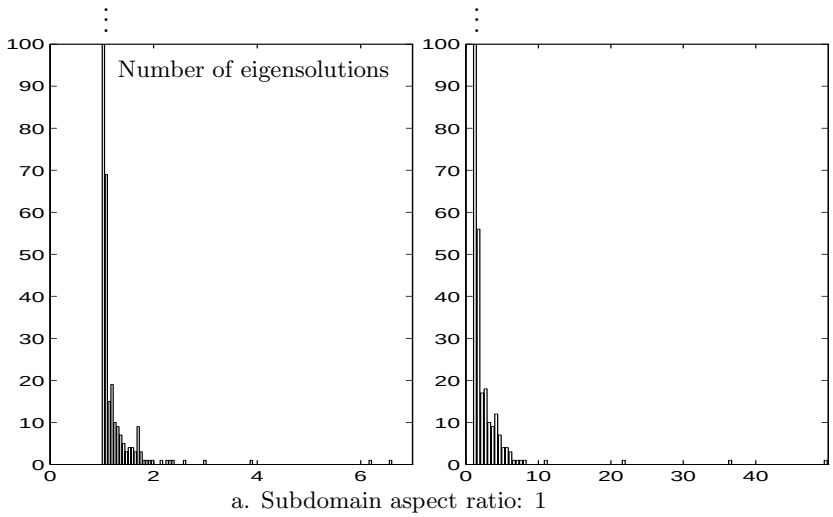
14.1.2 Mechanical behavior of subdomains with bad aspect ratios

To understand the mechanical reasons for the adverse effect of subdomain aspect ratios, let us apply the FETI method to the structure depicted in Fig. 13.4. The model contains 40×40 plane

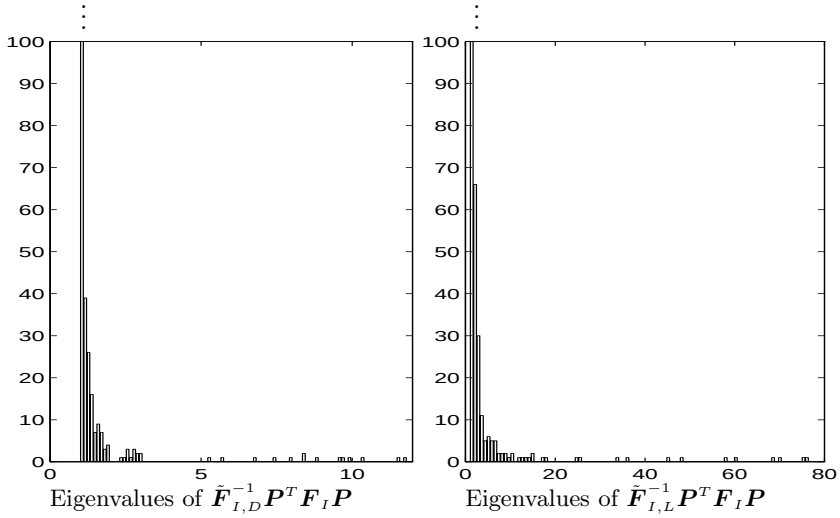


a. Subdomain aspect ratio 1 b. Subdomain aspect ratio 1/4

Figure 14.1: Two decompositions of a square domain



a. Subdomain aspect ratio: 1



b. Subdomain aspect ratio: 1/4

Figure 14.2: Eigenspectrum of the dual operator for two decompositions

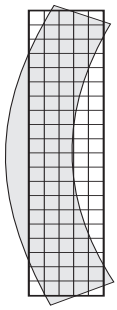


Figure 14.3: Subdomain displacement mode associated with the isolated eigenvalue of FETI for an aspect ratio of 1/4

stress elements and is decomposed into 4 vertical slices. We assume that the horizontal length is successively equal to $l_x = l_y$, $l_x = 4l_y$ and $l_x = 16l_y$ hence defining subdomain aspect ratios of respectively 1/4, 1 and 4. The number of iterations for the FETI method to achieve a relative primal residue [1] of 10^{-6} is given in Table 13.1 for the different aspect ratios and for the Dirichlet and the lumped preconditioners.

Prec.	$l_x = l_y$ Aspect ratio 1/4	$l_x = 4l_y$ Aspect ratio 1	$l_x = 16l_y$ Aspect ratio 4
Dirichlet	11	6	4
lumped	24	11	5

Table 14.1: Number of iteration for the FETI method ($\|\boldsymbol{z}\|/\|\boldsymbol{f}\| < 10^{-6}$)

From Table 13.1 we see that the convergence is very slow when the subdomains aspect ratio is 1/4, but convergence becomes rapid when the aspect ratio is equal to 4. Moreover the lumped preconditioner is inefficient for an aspect ratio of 1/4 whereas it is nearly as effective as the Dirichlet preconditioner for an aspect ratio of 4.

This example demonstrates that the negative effect of subdomains aspect ratio is a matter of distance between interfaces. If we look at the stress distribution on the far right interface Γ_I^{34}

[1] The primal residue will be defined in Section 15.7

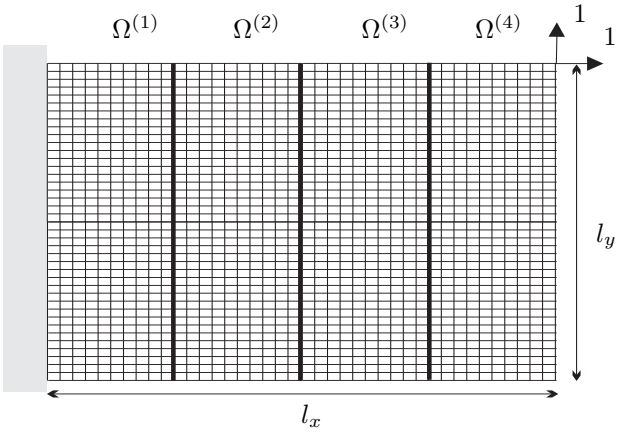


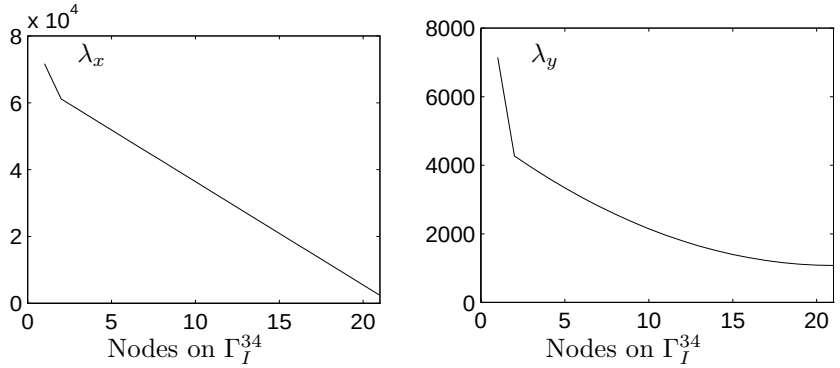
Figure 14.4: Sliced plane stress problem

close to the applied load (Fig. 13.5), we see that for an aspect ratio of 4, the interface stress distribution is smooth whereas for an aspect ratio of 1/4 the interface stresses have a rather irregular distribution. Noting that the stresses depicted in Fig. 13.5 correspond to the external point load on the upper right tip of the structure, we can interpret the results as follows:

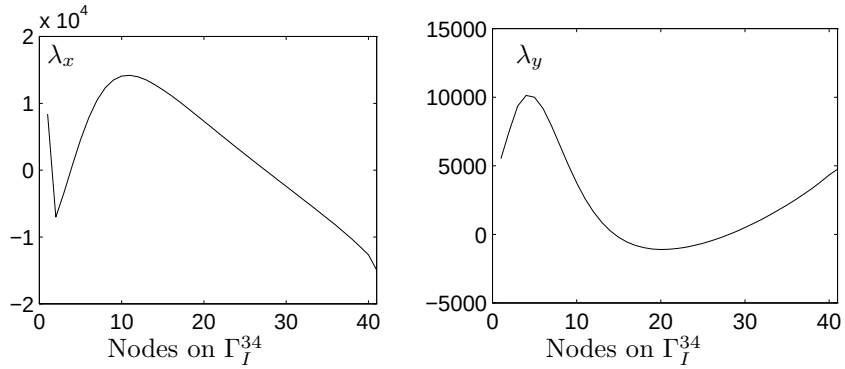
- When $l_x = 16l_y$, the interfaces are far apart from one another: the stresses are diffused so that only the global stress distribution corresponding to low energy modes are passed on to the interfaces while the high spectrum content of the displacements on an interface is not transmitted to the next one. Thus when the distance between interfaces is large, the subomains act as mechanical filter for the oscillating part of the solution (Saint Venant's principle). In that case, an accurate solution is found when only the high eigenspectrum of $\boldsymbol{F}_I$ is captured.
- When $l_x = l_y$, the distance between interfaces is much smaller then in the previous case. The interface solution plotted in Fig. 13.5 then appears much more irregular meaning that the stresses transmitted from one interface to another include the participation of low energy displacement modes as well as high energy components. Hence to built an accurate solution one must include a large number of eigensolutions of $\boldsymbol{F}_I$ having very different eigenvalues. This explains the poor convergence observed for the case when the subdomain aspect ratio is 1/4.

The study of the eigenspectrum of the dual operator (see previous section) and the discussion of the mechanical issues here above lead to the following important statements:

1. When the aspect ratio of all subdomains is close to 1, the distance between interface edges is maximized: the FETI method is then characterized by a good convergence due to the fact that the smooth components of the interface solution related to low energy modes and depending on the coupling of subdomains are well represented by the coarse grid of the floating modes. The local asperities of the solution which depend on the subdomains behavior in the vicinity of the interface edges and which are not prone to be affected by interface couplings are well represented by the Dirichlet and lumped preconditioners. Aspect ratios different from 1 adversely affect the convergence of the FETI method because some interfaces are close to one another and inter-subdomain coupling appears not only for the smooth modes but mechanical coupling between interface edges becomes important even for high energy components of the interface solution. The irregular component of the solution is thus badly approximated during a FETI iteration since the coarse grid takes care only of smooth modes, and the preconditioners responsible for estimating the interface corrections do not include subdomain coupling.
2. For bad aspect ratios, the Dirichlet preconditioner works much better than the lumped variant since the former at least correctly represents the coupling between the interfaces of a subdomain through the action of the internal d.o.f..
3. Knowing that the adverse effect of bad aspect ratios is originated by a bad dispersion of the high energy stress components between interface edges and noting that the coupling between d.o.f. is stronger for a harmonic problem (plane stress) then for a Poisson problem (heat transfer) and even stronger for a bi-harmonic problem such as plates, one can expect that the deterioration of the convergence for FETI with bad aspect ratios is more dramatic for harmonic problems and even more for bi-harmonic problems. This can be observed on the result Tables given in [67, chap.7].
4. The worst case appears when all subdomains have the same bad aspect ratio (such as in the example of Figs. 13.1 and 13.4) since in that case a high number of interface edges are then close to one another, hence the coupling between edges is very high. Thus, if it is not possible to define a mesh partitioning with good aspect ratios, one should at least search for a decomposition with different aspect ratios for neighboring subdomains.



a. Subdomain aspect ratio: 4



b. Subdomain aspect ratio: 1/4

Figure 14.5: Displacement solution on interface Γ_I^{34}

As a consequence, one should try to avoid subdomain aspect ratios significantly different from 1. Special care must therefore be taken when defining the domain decomposition of a structure for instance by including the subdomain aspect ratios in the cost function that is minimized during the optimization of the decomposition [64, 182].

14.2 FETI and finite element aspect ratio

For the FETI method, the mesh modes are related to the lower end of the eigenspectrum of the dual operator $\boldsymbol{F_I}$. These modes are generally not implied in the solution of the interface displacements and are in practice not captured during the FETI iterations. Therefore one can expect that the aspect ratios of the finite elements do not influence the conditioning of the FETI method. Moreover the Dirichlet and lumped preconditioners are extremely efficient in estimating the higher energy modes included in the solution and depending on the meshing. Hence, for the preconditioned FETI method, finite element aspect ratios is not a big issue.

To illustrate this point, let us consider three discretizations of a plane stress square domain as shown in Fig. 13.6. The first case is the reference case where the structure has a regular meshing and is decomposed into 4 square subdomains. The second problem is also decomposed into 4 squares but each subdomain is discretized with finite elements of different aspect ratios: the coarsest subdomain is modeled by 5×5 elements and the finest comprises 30×30 elements. The third example has a non-uniform decomposition and different finite element aspect ratios (all subdomains comprise 20×20 elements). The number of FETI iterations for reaching a relative primal error of 10^{-6} is listed in Table 13.2 for all cases and for the Dirichlet and lumped preconditioners.

	problem a.	problem b.	problem c.
F.E. asp.rat.	1	$1/6, 1, 6$	$1/6, 1, 6$
Subd asp.rat.	1	1	$1/3, 1, 3$
Dirichlet	10	10	13
lumped	21	19	25

Table 14.2: Problems of Fig. 13.6: number of iteration for the FETI method ($\|\boldsymbol{z}\|/\|\boldsymbol{f}\| < 10^{-6}$)

The results in Table 13.2 indicate that the number of FETI iterations is not affected by the finite elements aspect ratios. We also observe that when the subdomain aspect ratio is different from one (problem c.) the number of iteration increases only slightly because all subdomains have a different aspect ratio (see point 4. in the conclusions of Section 13.1.2).

Remarks

- For the primal Schur complement method, the mesh modes constitute the higher eigenspectrum of the operator. Therefore the condition number and the convergence rate are expected to be more sensitive to finite element ratios.
- When an iterative solution scheme is applied to the solution of the fully assembled problem, i.e. when no dual or primal interface problem is defined between subdomains, the convergence is significantly worsened by bad finite element aspect ratios since the assembled operator $\boldsymbol{K}$ has a condition number which directly depends on the shape of the finite elements [21, 43]. It is interesting to note that if we consider each element as a subdomain, the approach consisting in applying a Conjugate Gradient to the assembled system is equivalent to the primal Schur complement method.

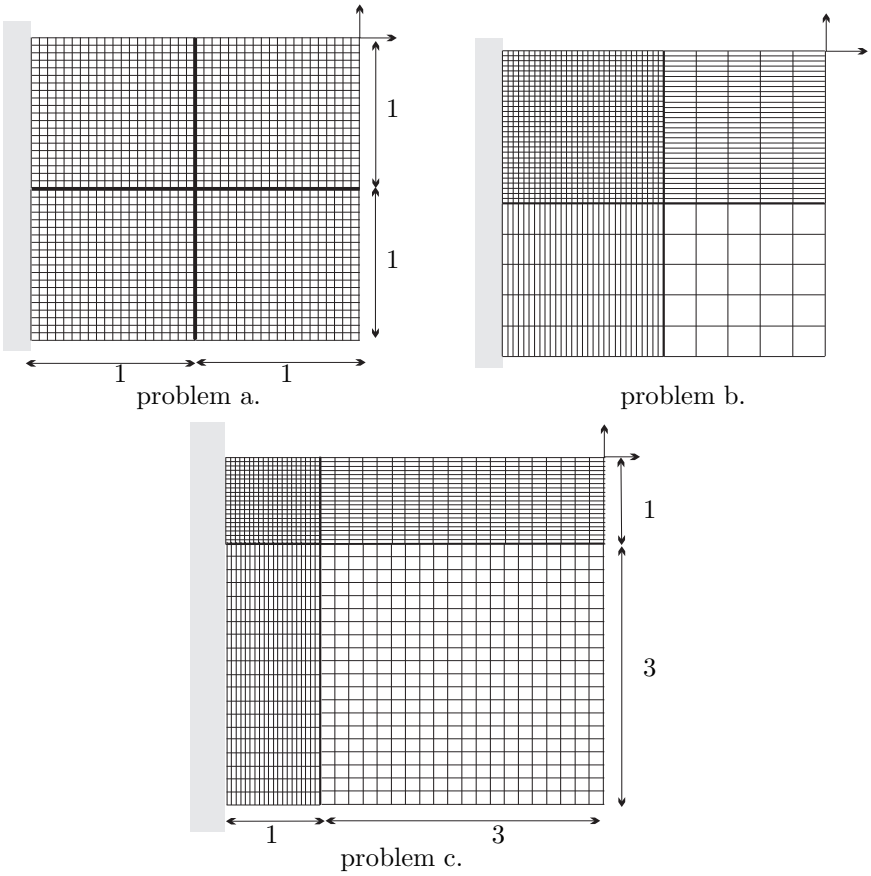


Figure 14.6: Problems with different element aspect ratios

Chapter 15

Approximate Dirichlet preconditioner

As already discussed in Section 12.3, applying a Dirichlet preconditioner to the FETI method requires factorizing the submatrices $\boldsymbol{K}_{ii}^{(s)}$ associated with the interior d.o.f. of each subdomain and performing a forward and backward substitution at every FETI iteration. It is therefore costly in terms of computing effort and storage space. Although its lumped variant is not as efficient in ensuring an optimal convergence, the increase in the number of iterations is often compensated by the low cost of one lumped preconditioning. For several well conditioned applications, the cheap lumped preconditioner leads thus to a better overall computing performance. Nevertheless, for other applications, the lumped preconditioner is enable to ensure an acceptable rate of convergence for the FETI method and the Dirichlet preconditioner is then the most cost effective choice.

From the discussions in 12.4 we understand that the lumped preconditioner is efficient when a good estimate of the interface stress correction can be found by applying the interface displacement gap on the boundaries and by assuming that the subdomain internal d.o.f. are fixed. In other words, the Dirichlet preconditioner must be used every time the interface solution is significantly dependent on the mechanical behavior of the subdomains interior. The cases encountered in practice where the internal behavior of the subdomains must be correctly included in the preconditioners are the followings:

- When strong interaction exists between d.o.f. in the vicinity of cross-points. We have said in Section 13.1.2 (p.186) that interactions between d.o.f. is weak for a Poisson problem, stronger for a harmonic problem and really strong for bi-harmonic problems such as for plate and shells (see also Section 11.6). Further more, as the element size decreases the internal d.o.f. are closer to the boundary Γ_I and their importance for approximating correctly the interface mechanics in the vicinity of a cross-point becomes primordial. Thus the need for a Dirichlet-like preconditioner increases as the discretization is refined.

- When the solution at a cross-point is determining for the solution in the entire structure, we have shown in 12.5 and we will illustrate in Section 15.2 the necessity of including the Dirichlet preconditioner in the rigid body coarse grid, the lumped preconditioner having barely any effect for this purpose.

- If the domain decomposition is characterized by bad subdomain aspect ratios, including the coupling between edges of a subdomain in the evaluation of interface corrections is essential and therefore the Dirichlet preconditioner should be used whenever the subdomain aspect ratios are very different from one (Section 13.1).

- Every time non-homogeneities of the subdomain interiors significantly affects the solution on the interface. This point will be illustrated by an example at the end of this section.

The need for a new preconditioner having an efficiency similar to the Dirichlet preconditioner in the troublesome cases listed here above but which avoids the cost of factorizing $\boldsymbol{K}_{ii}^{(s)}$ appears in practical applications. In this chapter we will present a new Dirichlet-like preconditioner which allows to approximate the Dirichlet preconditioner in a cheap and yet effective way.

15.1 Krylov basis for approximating the internal subdomain operators

Computing a new FETI direction by the Dirichlet preconditioner consists in the two steps

$$\Delta \boldsymbol{f}_b^{(s),k} = \left(\boldsymbol{K}_{bb}^{(s)} - \boldsymbol{K}_{bi}^{(s)} {\boldsymbol{K}_{ii}^{(s)}}^{-1} \boldsymbol{K}_{ib}^{(s)} \right) {\boldsymbol{b}^{(s)}}^T \boldsymbol{w}^k \tag{15.1}$$

$$\boldsymbol{z}^k = \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \Delta \boldsymbol{f}_b^{(s),k} \tag{15.2}$$

The fundamental idea in approximating this preconditioner arises from the observation that it is not necessary to compute an exact solution to the local Dirichlet problem for finding the new directions of the interface stress space, the latter being anyway only an estimate. Our approach thus consists in approximating the solution of the Dirichlet problem

$$\boldsymbol{K}_{ii}^{(s)} \Delta \boldsymbol{u}_i^{(s),k} = \boldsymbol{K}_{ib}^{(s)} {\boldsymbol{b}^{(s)}}^T \boldsymbol{w}^k \tag{15.3}$$

where $\boldsymbol{w}^k$ is the scaled dual residue.

We will approximate the inverse of $\boldsymbol{K}_{ii}^{(s)}$ in the associated Krylov subspace of dimension n, i.e.

$$\mathcal{K}_n(\boldsymbol{x}, \boldsymbol{K}_{ii}^{(s)}) = \text{span} \left\{ \boldsymbol{x}^{(s),0}, \boldsymbol{K}_{ii}^{(s)} \boldsymbol{x}^{(s),0}, \ldots, {\boldsymbol{K}_{ii}^{(s)}}^{n-1} \boldsymbol{x}^{(s),0} \right\} \tag{15.4}$$

$$= \text{span} \left\{ \boldsymbol{x}^{(s),0}, \boldsymbol{x}^{(s),1}, \ldots, \boldsymbol{x}^{(s),n-1} \right\} \tag{15.5}$$

where $\boldsymbol{x}^{(s),0}$ stands for the root vector and $\boldsymbol{x}^{(s),j}$ are the orthonormalized components of the Krylov space stored in the columns of $\boldsymbol{X}^{(s)}$ such that

$${\boldsymbol{X}^{(s)}}^T \boldsymbol{K}_{ii}^{(s)} \boldsymbol{X}^{(s)} = \boldsymbol{I} \tag{15.6}$$

The basis $\boldsymbol{X}^{(s)}$ converges towards the eigenmode subspace associated with the high end of the eigenspectrum of $\boldsymbol{K}_{ii}^{(s)}$.

The solution $\Delta \boldsymbol{u}_i^{(s),k}$ of the Dirichlet problem (14.3) is then searched in the Krylov space by considering the general solution

$$\Delta \boldsymbol{u}_i^{(s),k} = \sum_{j=0}^{n-1} \gamma_j \boldsymbol{x}^{(s),j} \tag{15.7}$$

Substituting this general solution into (14.3) and projecting onto the orthonormal basis $\boldsymbol{X}^{(s)}$ we find

$$\Delta \boldsymbol{u}_i^{(s),k} = \sum_{j=0}^{n-1} \boldsymbol{x}^{(s),j} \boldsymbol{x}^{(s),j^T} \left(\boldsymbol{K}_{ib}^{(s)} {\boldsymbol{b}^{(s)}}^T \boldsymbol{w}^k \right) \tag{15.8}$$

$$= \boldsymbol{X}^{(s)} {\boldsymbol{X}^{(s)}}^T \left(\boldsymbol{K}_{ib}^{(s)} {\boldsymbol{b}^{(s)}}^T \boldsymbol{w}^k \right) \tag{15.9}$$

We thus approximate the inverse of $\boldsymbol{K}_{ii}^{(s)}$ by

$${\boldsymbol{K}_{ii}^{(s)}}^{-1} \simeq \boldsymbol{X}^{(s)} {\boldsymbol{X}^{(s)}}^T \tag{15.10}$$

If we denote by $\boldsymbol{W}_{bn}^{(s)}$ the boundary reactions matrix of dimension $n_b^{(s)} \times n$ associated to the Krylov base directions, i.e.

$$\boldsymbol{W}_{bn}^{(s)} = \boldsymbol{K}_{bi}^{(s)} \boldsymbol{X}^{(s)} \tag{15.11}$$

the approximation of the interface loads in the Dirichlet preconditioner (14.2) writes

$$\Delta \boldsymbol{f}_b^{(s),k} \simeq \left(\boldsymbol{K}_{bb}^{(s)} - \boldsymbol{W}_{bn}^{(s)} \boldsymbol{W}_{bn}^{(s)^T} \right) \boldsymbol{b}^{(s)^T} \boldsymbol{w}^k \tag{15.12}$$

15.2 Adapting the Dirichlet preconditioner during FETI iterations

If we choose as root direction for the Krylov space $\mathcal{K}_n(\boldsymbol{x}, \boldsymbol{K}_{ii}^{(s)})$ the internal loads produced when enforcing the initial scaled compatibility residue $\boldsymbol{w}^0$ on the interface, we write

$$\boldsymbol{x}^0 = \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{w}^0 \tag{15.13}$$

The Krylov basis is then obtained by constructing the successive directions $\boldsymbol{K}_{ii}^{(s)} \boldsymbol{x}^j$ and by orthonormalizing them with respect to one another. This procedure is exactly equivalent to a Conjugate Gradient applied to the internal equilibrium problem (14.3) and where the Krylov vectors are the conjugate directions of descent. The dimension n of the Krylov subspace is then determined by the accuracy we require for the solution $\Delta \boldsymbol{u}_i^{(s),0}$.

For the next FETI iterations $k = 1, ...$, the preconditioned directions are found by using the Krylov basis computed at FETI iteration $k = 0$, namely by Eq. (14.12). This strategy is not efficient in practice because as the FETI iteration progresses, the dual residue on the interface changes (remember that the successive compatibility errors $\boldsymbol{w}^k$ are orthogonal to one another with respect to the preconditioner (Section 11.2, Eq. 11.18)). After a few iterations, the Krylov space generated by the root vector $\boldsymbol{w}^0$ is not adapted anymore to the new Dirichlet problems and one observes that convergence rate becomes similar to the convergence rate obtained with the lumped preconditioner.

Thus at every FETI iteration, we will complement the Krylov basis of $\boldsymbol{K}_{ii}^{(s)}$ so that every $\Delta \boldsymbol{u}_i^{(s),k}$ satisfies the internal equilibrium (14.3) with a relative precision of ϵ_D. The approximate Dirichlet preconditioned direction is thus found by a Conjugate Gradient with a projection-orthogonalization procedure such as presented in Section 11.4 for solving systems with multiple right-hand sides. As a preconditioner for this auxiliary Conjugate Cradient we choose the simplest and cheapest one, namely the Jacoby preconditioner consisting in scaling the new iterates by the inverse of the diagonal of $\boldsymbol{K}_{ii}^{(s)}$.

Finally, we call $\boldsymbol{Y}^{(s)}$ the matrix of dimension $n_i^{(s)} \times n_k$ storing $\boldsymbol{K}_{ii}^{(s)} \boldsymbol{X}^{(s)}$, n_k being the size of the Krylov space successively obtained during FETI iterations. The strategy for solving the internal equilibrium (14.3) when preconditioning the k-th FETI iteration is summarized by the algorithm 14.1.

The Krylov subspace $\mathcal{K}_n$ is enriched every time a conjugate direction iteration is performed in algorithm 14.1. The approximation of the inverse of $\boldsymbol{K}_{ii}^{(s)}$ changes during the FETI iteration and so does the preconditioner. Therefore full orthogonalization is required in the FETI iteration: this is not an issue since full orthogonalization should be used anyway in order to prevent the loss of orthogonality between successive FETI conjugate directions.

It is worthy mentioning that the strategy for approximating the Dirichlet local problems presented in this section can be adapted in a straightforward manner to the primal Schur method. It then consists in solving the local Neumann problems by auxiliary conjugate gradient iterations.

Algorithm 15.1 Computing an approximate solution to the Dirichlet problem

-Projection-

if $k = 0$ **then**

$\Delta \boldsymbol{u}_i^{(s),k} = \boldsymbol{0}$

$\boldsymbol{r}_D = \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{w}^k$

else

$\boldsymbol{\alpha}_D = \boldsymbol{X}^{(s)^T} \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{w}^k$

$\Delta \boldsymbol{u}_i^{(s),k} = \boldsymbol{X}^{(s)} \boldsymbol{\alpha}_D$

$\boldsymbol{r}_D = \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{w}^k - \boldsymbol{Y}^{(s)} \boldsymbol{\alpha}_D$

end if

-Conjugate Iteration-

while $\|\boldsymbol{r}_D\| > \epsilon_D \|\boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{w}^k\|$ **do**

-Diagonal preconditioning-

$\boldsymbol{p}_D = \mathrm{diag}(\boldsymbol{K}_{ii}^{(s)})^{-1} \boldsymbol{r}_D$

-Conjugate directions-

$\boldsymbol{\beta}_D = \boldsymbol{Y}^{(s)^T} \boldsymbol{p}_D$

$\boldsymbol{p}_D = \boldsymbol{p}_D - \boldsymbol{X}^{(s)} \boldsymbol{\beta}_D$

-Minimization-

$\boldsymbol{y}_D = \boldsymbol{K}_{ii}^{(s)} \boldsymbol{p}_D$

$\mu_D = \sqrt{\boldsymbol{p}_D^T \boldsymbol{y}_D}$; $\boldsymbol{y}_D = \boldsymbol{y}_D / \mu$; $\boldsymbol{p}_D = \boldsymbol{p}_D / \mu$

$\eta_D = \boldsymbol{p}_D^T \boldsymbol{r}_D$

$\Delta \boldsymbol{u}_i^{(s),k} = \Delta \boldsymbol{u}_i^{(s),k} + \boldsymbol{p}_D \eta_D$

$\boldsymbol{r}_D = \boldsymbol{r}_D - \boldsymbol{y}_D \eta_D$

-Update the Krylov space-

$\boldsymbol{X}^{(s)} = \left[\boldsymbol{X}^{(s)} \ \boldsymbol{p}_D\right]$; $\boldsymbol{Y}^{(s)} = \left[\boldsymbol{Y}^{(s)} \ \boldsymbol{y}_D\right]$

end while

15.3 Which accuracy for the subdomain internal equilibrium?

The approximate Dirichlet preconditioner we propose is similar to the exact Dirichlet preconditioner except that the internal equilibrium is satisfied only with an accuracy ϵ_D. Obviously if we choose ϵ_D too small, the internal equilibrium will be solved nearly exactly and no gain can be expected compared to the exact Dirichlet. The whole idea is to choose ϵ_D small enough to find a good approximation of the Dirichlet preconditioner so to ensure good convergence for the FETI iterations, but selecting ϵ_D large enough so that the inverse of $\boldsymbol{K}_{ii}^{(s)}$ be represented with a Krylov space of minimum dimension.

To determine the optimal value of ϵ_D, let us apply the FETI method for computing the solution of the simple plane stress problem of Fig. 14.1 decomposed into 4 equal subdomains. Each subdomain is modeled by 10×10 elements. In a first analysis we will assume that the subdomains are square, and we will take successively the values $\epsilon_D = 0.05, 0.1, 0.2, 0.3, 0.5, 0.8$. The convergence of the primal residual is shown in Fig. 14.2(a) together with the convergence curves obtained with the Dirichlet and lumped preconditioners. Fig. 14.2(a) also shows the variation of the dimension of the Krylov space for $\boldsymbol{K}_{ii}^{(3)}$ during the FETI iteration.

The same analysis is then run for the case when the length of the structure in Fig. 14.1 is equal to 4 times its height. The subdomain aspect ratios are now equal to 4. The convergence curves and the dimension of the Krylov space for $\boldsymbol{K}_{ii}^{(3)}$ are depicted in Fig. 14.2(b).

The results of Fig. 14.2 show that for an accuracy of $\epsilon_D = 0.05$, the approximate Dirichlet preconditioner is equivalent to the exact Dirichlet procedure, but in this case the number of Krylov base directions in a subdomain is nearly equal to half the number of internal d.o.f. Such an accuracy thus leads to a cost similar to an exact Dirichlet preconditioner. If we choose a very low accuracy such as $\epsilon_D = 0.8$, only a few directions are needed in the Krylov space but the efficiency of the preconditioner is then close to the efficiency of a lumped preconditioner. The optimum accuracy is thus of the order of 20 percent, i.e. $\epsilon_D = 0.2$ in which case the approximate Dirichlet preconditioner has an efficiency close to what can be obtained with an exact Dirichlet procedure and the dimension of the subdomain Krylov spaces remains small compared to the number of internal d.o.f. For instance, when $\epsilon_D = 0.2$ and when a relative primal residue of 10^{-6} is required, the dimension of the Krylov space in $\Omega^{(3)}$ is more or less a tenth of the dimension of $\boldsymbol{K}_{ii}^{(3)}$ which allows to save computational effort compared to the cost of a full factorization.

Comparing Fig. 14.2(a) and Fig. 14.2(b), we observe once more the adverse effect of bad subdomain aspect ratios, especially when a lumped preconditioner is used. If we choose $\epsilon_D = 0.2$ in the case where the subdomain aspect ratios are equal to 4, the FETI method converges in 16 iterations whereas the lumped preconditioner leads to a convergence in 24 iterations. The cost for this significant convergence improvement mainly lies in computing 30 Krylov directions involving 30 multiplications by $\boldsymbol{K}_{ii}^{(s)}$.

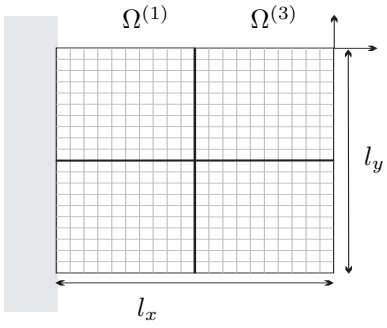
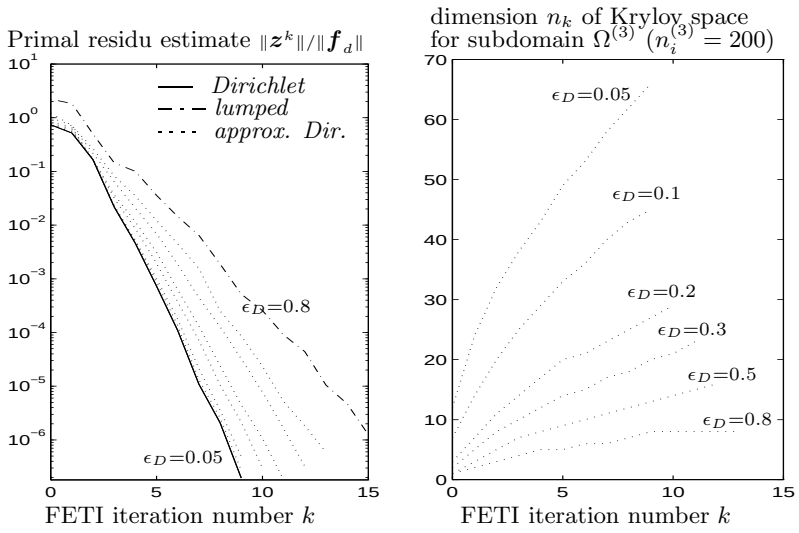
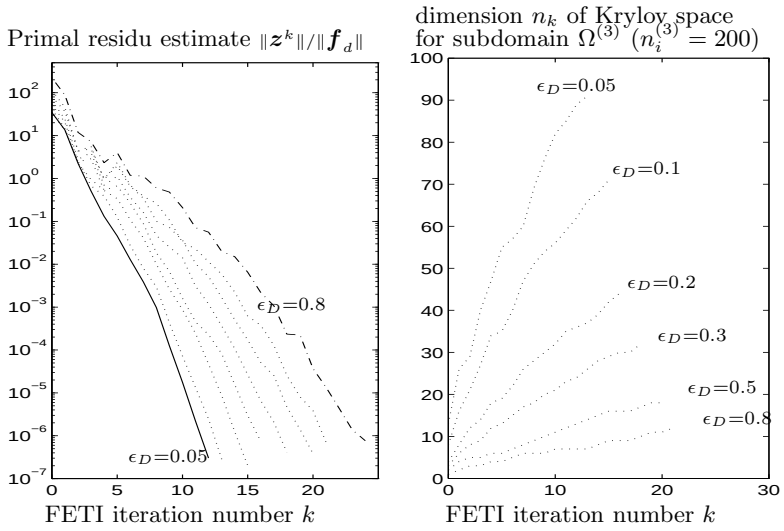


Figure 15.1: Plane stress structure for testing the approximate Dirichlet preconditioner



(a) a. Subdomain aspects ratio of 1



(b) b. Subdomain aspects ratio of 4

Figure 15.2: Convergence of FETI with an approximate Dirichlet preconditioner

15.4 Influence of mesh refinement on the approximate Dirichlet procedure

We will now investigate the effect of mesh refinement on the efficiency of the approximate Dirichlet. Let us consider again the plane stress problem described in Fig. 14.1 when $l_x/l_y = 4$, that is for a subdomain aspect ratio of 4, and let us assume now that each subdomain has a finite element mesh of successively 5×5, 10×10, 20×20, and 30×30. Choosing an accuracy $\epsilon_D = 0.2$ for the approximate Dirichlet procedure, we give in Table 14.1 the number of FETI iterations required to achieve convergence for a relative primal error of 10^{-6}.

Meshing	Preconditioners		
	Dirichlet	approx. Diri $\epsilon_D = 0.2$	lumped
5×5	11	14	18
10×10	12	16	24
20×20	13	20	31
30×30	13	22	36
40×40	13	24	39

Table 15.1: Efficiency of preconditioners for the FETI method ($l_x/l_y = 4$, $\|\boldsymbol{z}\|/\|\boldsymbol{f}\| < 10^{-6}$)

From Table 14.1 we see that as predicted by the theoretical condition number (12.6), the number of FETI iterations with the optimal Dirichlet preconditioner increases only weakly when the mesh size decreases. On the contrary, the efficiency of the lumped preconditioner is badly affected by the mesh refinement. When the approximate Dirichlet preconditioner is considered, we see that the number of iteration grows also weakly.

If we then plot the ratio between the dimension of the Krylov space during the FETI iteration and the number of internal d.o.f. (Fig. 14.3), we see that the relative number of Krylov base directions significantly decreases when the mesh size decreases. For instance, for a subdomain meshing of 40×40 elements, there are 3200 d.o.f. per subdomain and the approximate Dirichlet preconditioner defines only 82 Krylov base vectors, which is less than 3 percent.

The fact that the relative number of directions required to approximate the internal equilibrium with a given accuracy decreases together with the mesh size is a remarkable property. Indeed, this means that for problems with several thousands of d.o.f. per subdomain, the cost of the approximate Dirichlet is small and thus the approximate Dirichlet preconditioner can advantageoulsy replace the exact Dirichlet preconditioner. This remarkable property arises from the fact that in order to evaluate an approximate stress correction on the interface, only the interior d.o.f. located

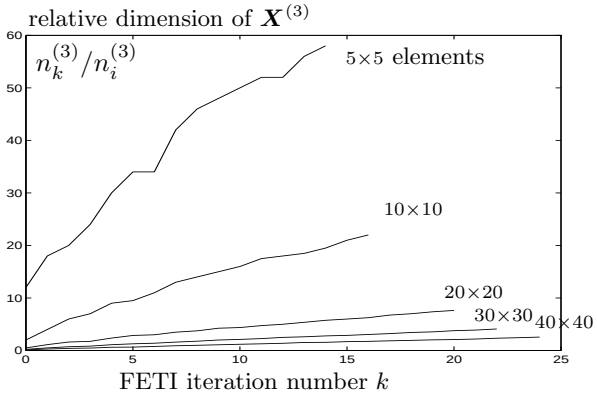


Figure 15.3: Relative number of Krylov directions in $\Omega^{(3)}$ when the mesh size decreases ($l_x/l_y = 4$)

near the interface play a determining role while the d.o.f. in the very interior of the subdomains are important only for defining the response of the subdomain to very smooth interface displacements. Hence, as the mesh is refined, it is not necessary to compute the precise solution for all internal d.o.f.

15.5 Difference with a penetration depth strategy

In order to investigate how the approximate Dirichlet preconditioner operates, let us analyse how the successive Krylov vectors $\boldsymbol{X}^{(s)}$ are constructed. The first direction $\boldsymbol{x}^0$ is given by Eq. (14.13)

$$\boldsymbol{x}^0 = \boldsymbol{K}_{ib}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{w}^0 \tag{14.13}$$

and represents the initial residue of the iterations in Algorithm 14.1. Therefore, $\boldsymbol{x}^0 = \boldsymbol{p}_D^0 = \boldsymbol{r}_D^0$ is the force produced on the internal d.o.f. when the displacement $\boldsymbol{b}^{(s)^T} \boldsymbol{w}^0$ is applied on the boundary of the subdomain $\Omega^{(s)}$. $\boldsymbol{x}^0$ is zero for every internal d.o.f. situated more then one element apart from the interface (see Fig. 14.4). The next Krylov direction is

$$\begin{aligned} \boldsymbol{x}^1 &= \boldsymbol{p}_D^1 && (15.14)\\ &= \boldsymbol{r}_D^1 - \boldsymbol{\beta}_D^1 \boldsymbol{x}^0 && (15.15)\\ &= \boldsymbol{x}^0 - \boldsymbol{K}_{ii}^{(s)} \boldsymbol{x}^0 \eta_D^0 - \boldsymbol{\beta}_D^1 \boldsymbol{x}^0 && (15.16) \end{aligned}$$

and we see that it consists in a linear combination of $\boldsymbol{x}^0$ and $\boldsymbol{K}_{ii}^{(s)} \boldsymbol{x}^0$, the latter term being zero for all internal d.o.f. more than two element levels apart from the interface. The reasoning then goes on showing that as the iteration progresses, more levels of internal d.o.f. are implied (Fig. 14.4).

Thus we can say that the approximate Dirichlet iterations stop when the penetration depth of the solution is sufficient to correctly represent the interface solution for a given interface displacement. We can thus argue that our preconditioner is similar to the procedure recently proposed by Achdou *et al.* in [2]: they propose to define a constant penetration depth such that all the internal d.o.f. further apart from the boundary are considered as fixed. A cheap Dirichlet-like preconditioner is then obtained by factorizing only the part of $\boldsymbol{K}_{ii}^{(s)}$ pertaining to the non-fixed internal d.o.f. and by computing "truncated" local Schur complements.

We have tried the method proposed by Achdou *et al.*, and it turned out to be rather inefficient and in any case less efficient then our approximate Dirichlet preconditioner, and this for the following reasons:

- In order to obtain a significant reduction of the cost when factorizing the "truncated" interior operator, the penetration depth should not be to large. Since the most interior d.o.f. are then considered as fixed, the global behavior of the subdomains is not well represented.

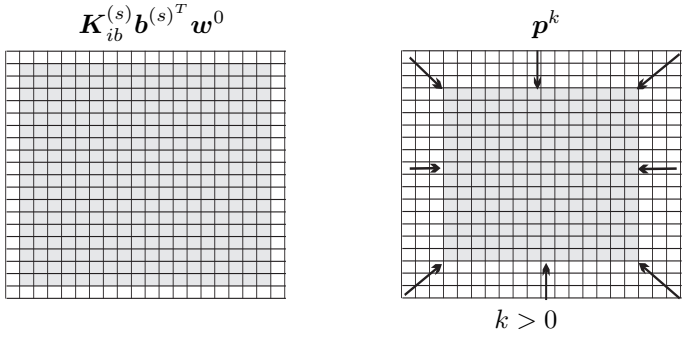


Figure 15.4: Progression of the penetration depth in the approximate Dirichlet method

- The penetration depth must be defined once for all at the begin of the FETI iterations and remains unchanged unless one pays the price for performing a new factorization of the internal operator. It is very tricky to define the optimum penetration depth a priori since it depends on the subdomain discretization and shape. Hence, the penetration depth method is difficult to use in practice.
- Implementing the penetration depth algorithm is not an easy task since it requires to consider the mesh topology in order to define the active internal d.o.f.

All those drawbacks are irrelevant to our approximate method since the penetration depth therein is implicitly defined by the level of accuracy we impose to the internal equations of equilibrium.

Chapter 16

Two-level FETI method

Chapter 17

Preconditioners for heterogeneous problems

17.1 Structural heterogeneities and optimal preconditioning

17.1.1 Heterogeneous problems and FETI convergence

So far we have considered the structures analyzed as homogeneous. Unfortunately, in real life, structures are often non-homogeneous due to one or several of the following reasons:

- The mechanical properties of the subdomains are very different according to the subdomain considered. For instance, the Young's modulus may vary from one subdomain to another.
- The subdomain interfaces coincide with structural intersections so that the orientations of the subdomains of each side of the interface are different. This may cause an important discrepancy in the stiffnesses associated with the interface d.o.f.

To illustrate the different types of heterogeneities encountered in practice, we will consider five exemplary problems. First, let us take a square plane stress domain subdivided into 4 square subdomains. Each subdomain contains 20×20 elements and we consider that two subdomains are made of steel while the two others are made of rubber and thus have a Young's modulus 4098 times smaller than steel, i.e $E_{rubber} = E_{steel}/4098$. In Fig. 15.1 we consider two different configurations: in problem A, the two rubber subdomains are both located on the right-hand side of the structure, whereas in problem B the two rubber subdomain connect only at the cross-point.

Problem C is similar to both problems above except that the structure is now a rectangular plate decomposed into 6 regular subdomains, each of them modeled by 50 DKT (Discrete Kirchhoff Triangle) elements (Fig. 15.2). One of the subdomains is assumed to have a bending flexibility 100 times smaller than the others, due either to smaller thickness or to smaller Young's modulus. For this bi-harmonic problem we assume that an auxiliary corner problem is solved for the cross-points (Section 11.6).

Problem D is a plane stress domain containing soft inclusions (Fig. 15.3). It is modeled by a total of 64×64 elements and the soft inclusions have a Young's modulus 1000 times smaller than the matrix material, i.e. $E_{incl} = E_{matr}/1000$. Several regular decompositions will be considered, namely 2 by 2, 4 by 4 and 8 by 8.

The last example, problem E, illustrates the second type of heterogeneities, that is the case when subdomain interfaces coincide with structural edges and where heterogeneity arises from differences in subdomain orientations. To this purpose we consider again the wing-box structure

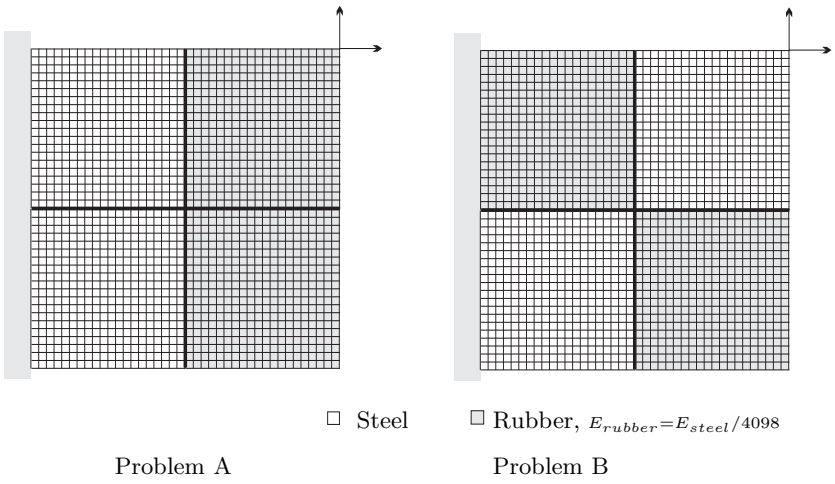


Figure 17.1: Heterogeneous problems A and B: steel and rubber plane stress subdomains

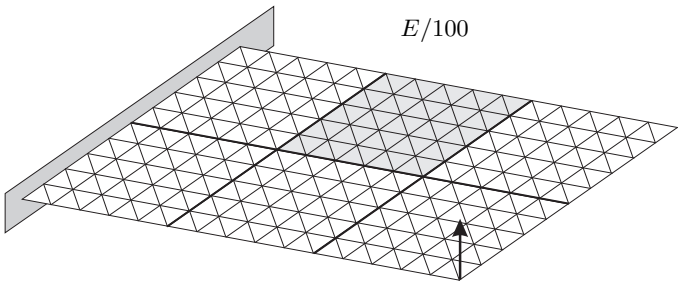


Figure 17.2: Heterogeneous problem C: plate structure

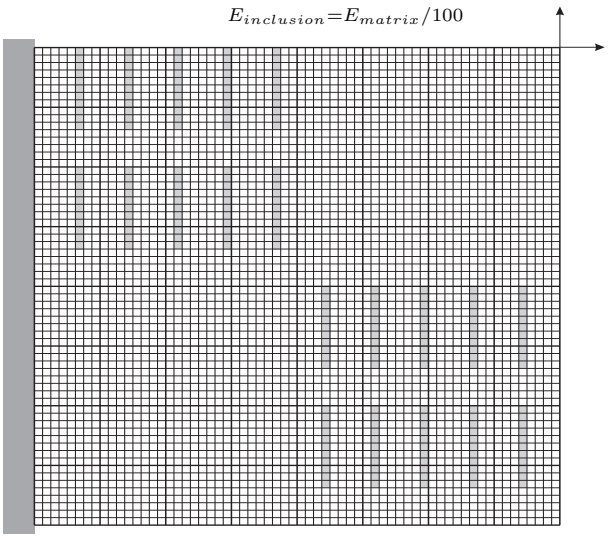


Figure 17.3: Heterogeneous problem D: plane stress domain with soft inclusions

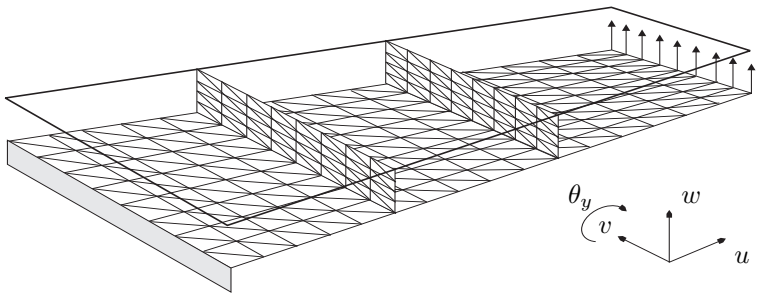


Figure 17.4: Heterogeneous problem E: wing box structure

with conforming discretization analyzed earlier in Part I, Section 7.7.3 where a linear load is applied on the tip of the lower skin (Fig. 15.4).

To show how heterogeneities affect the convergence of the FETI method, we compute the solution of the heterogeneous problems presented here above for a primal residue lower than 10^{-6}, when using a Dirichlet and a lumped preconditioner. The number of FETI iterations are given in Table 15.1, and we also specify the number of iterations for the corresponding homogeneous case (except for the wing-box problem E which does not have any homogeneous counterpart).

	Homogeneous		Heterogeneous	
Problem	Dirichlet	Lumped	Dirichlet	Lumped
A	10	21	68	82
B	10	21	46	44
C	11	19	37	44
D(2×2)	10	25	20	37
D(4×4)	15	29	79	92
D(8×8)	16	25	129	141
E	—	—	122	128

Table 17.1: Effect of heterogeneities on FETI iterations

The results in Table 15.1 clearly indicate that when the subdomains are not homogeneous, the FETI iterations convergence is much slower. The Dirichlet preconditioner which is optimal for homogeneous cases does no longer ensure a good convergence and in the heterogeneous problem B the lumped preconditioner even turns out to be more efficient than the Dirichlet one. We also notice that for the wing-box problem E, the numbers of iterations for the Dirichlet and lumped preconditioners are nearly identical and considering that the associated dual problem is of dimension $n_\lambda = 216$, we understand that the number of iterations for the wing-box is unacceptably high.

In Fig. 15.5 we draw the convergence curves for the homogeneous and heterogeneous problems A and B. The convergence curves of the estimated primal residual are oscillatory for the heterogeneous problem A, which indicates that the Dirichlet and lumped preconditioners do not properly evaluate the interface stress corrections.

Let us recall that the Dirichlet and lumped preconditioners take their origin in the approximation of the inverse of the dual operator by the assembling of the Schur or lumped operators

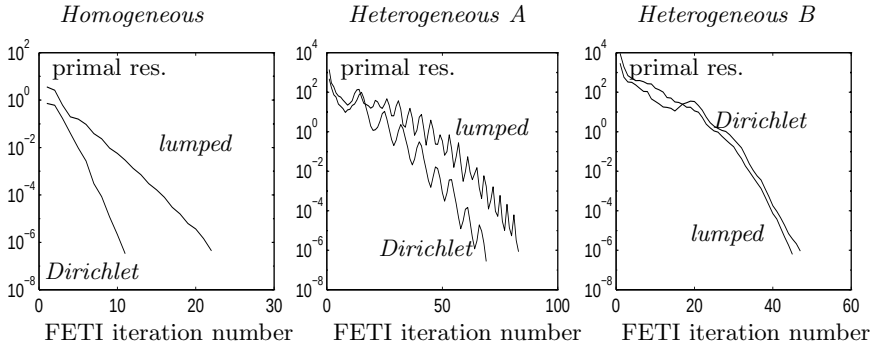


Figure 17.5: Convergence curves for problems A and B

(section 12.1),

$$\boldsymbol{F}_I{}^{-1} = \left(\sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)^+} \boldsymbol{b}^{(s)^T} \right)^{-1} \simeq \sum_{s=1}^{N_s} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \tag{17.1}$$

It is clear that this approximation is valid only if the Schur complements appearing in the assembly are of similar orders of magnitude. In heterogeneous problems, this approximation is no longer correct and we see from Eq. (15.1) that more weight should be given to the stiffer subdomains, that is to the Schur complements of higher order of magnitude. In this section we will present new preconditioning procedures which consist in choosing in an optimum manner the weightings of the different Schur complements in (15.1). From a mechanical view point (Section 12.2), it means that enforcing an averaged interface displacement and averaging the resulting interface stresses to define the descent directions is not the right thing to do when the subdomains are not homogeneous. Considering the subdomains as macro elements, it is clear that the smoothing method developed in part I for post-processing weakly compatible displacements will be the basis of a new strategy for defining optimal preconditioned directions.

17.1.2 Optimum weighted Dirichlet preconditioner for uniform subdomains

In the simple case when the subdomains have uniform properties and where the global structure is heterogeneous because the mechanical properties are different depending on the subdomain, one can assume that all the subdomain Schur complements are similar to a factor. In problems A, B and C for instance, the subdomains have uniform mechanical characteristics and all Schur complements are similar except for the ratio between the subdomains Young's moduli. In that case it is easy to find an optimum Dirichlet preconditioner by adapting the scaling procedure proposed by LeTallec *et al.* in [175] for the primal Schur complement method. It consists in defining a priori inter-subdomain weighting coefficients for each edge in the same way as we defined the preconditioner scaling in Section 12.2 where, for homogeneous structures, we have shown that the optimal weights are the inverses of edge multiplicities.

For heterogeneous problems when $\rho^{(s)}$ defines the specific subdomain stiffness, we weigh the interface displacements and stresses according to the relative stiffnesses. We thus obtain the scaled Dirichlet preconditioner for heterogeneous problems by adapting the relations (12.14–12.15–12.16): using the notations introduced in Section 7.1 in the context of smoothing, we write

$$\Delta \boldsymbol{u}_b^{(s),j^k} = - \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \overline{\rho}^{(r),j} \; \boldsymbol{b}^{(s,-),j^T} \boldsymbol{w}^{(sr),j^k} \tag{17.2a}$$

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{17.2b}$$

$$\boldsymbol{z}^{(sr),j^k} = \Delta\boldsymbol{\lambda}^{sr,j^k} \quad = \quad -\overline{\rho}^{(r),j}\ \boldsymbol{b}^{(s,-),j}\Delta\boldsymbol{f}_b^{(s),j} - \overline{\rho}^{(s),j}\ \boldsymbol{b}^{(r,-),j}\Delta\boldsymbol{f}_b^{(r),j} \tag{17.2c}$$

where $\boldsymbol{w}^{(sr),j^T}$ is the compatibility error on an edge j between subdomains $\Omega^{(s)}$ and $\Omega^{(r)}$. $\overline{\rho}^{(s),j}$ is the relative stiffness of subdomain $\Omega^{(s)}$ with respect to all subdomains connecting on edge j:

$$\overline{\rho}_j^{(s)} = \frac{\rho^{(s)}}{\sum\limits_{r:\Gamma_I^{(r)}\ni j} \rho^{(r)}} \tag{17.3}$$

and $\rho^{(s)}$ is the characteristic subdomain stiffness coefficient (e.g. Young's modulus or thickness). For the homogeneous case, we find that $\rho_j^{(r)} = \rho_j^{(s)}$ for all subdomains connecting on an edge, and thus $\overline{\rho}_j^{(s)} = 1/m_j$. Hence the heterogeneous scaling becomes equivalent to the multiplicity scaling for homogeneous problems. Obviously this heterogeneous scaling procedure is also applicable to the lumped preconditioner in which case $\boldsymbol{S}_{bb}^{(s)}$ is replaced by $\boldsymbol{K}_{bb}^{(s)}$ in Eq. (15.2b).

Applying this heterogeneous scaling to problems A, B and C, the number of FETI iterations are as listed in Table 15.2. From the results of Table 15.2, we observe that the heterogeneous scaling

	Heterogeneous Problems			
	$\tilde{\boldsymbol{F}}_{I,D}^{-1}$		$\tilde{\boldsymbol{F}}_{I,L}^{-1}$	
Problem	$1/m_i$ scaling	$\overline{\rho}_j^{(s)}$ scaling	$1/m_i$ scaling	$\overline{\rho}_j^{(s)}$ scaling
A	68	11	82	25
B	46	4	44	24
C	37	11	44	19

Table 17.2: Heterogeneous scaling for problems A,B and C

procedure derived from the primal scaling proposed by LeTallec *et al.* in [175] is optimum: it leads to convergence rates at least similar to those obtained for the homogeneous problems. For Problem B, we notice that when the heterogeneous scaling is applied to the Dirichlet preconditioner, the required accuracy is obtained already after 4 iterations. This is due to the fact that in Problem B (Fig. 15.1) the two stiffer subdomains are connected only through the cross-point and thus the solution is nearly totally determined by the sole corner connection loads. We see that the Dirichlet preconditioner with the heterogeneous scaling accurately evaluates the corner stresses for this highly heterogeneous case.

The heterogeneous scaling introduced here by modifying the multiplicity scaling method and by adapting the scaling for primal formulations is a cheap and efficient method for preconditioning heterogeneous problems when each subdomain can be given a characteristic stiffness factor $\rho^{(s)}$. Unfortunately this situation arises in practice only for subdomain integration, namely when the subdomains are defined by structural sub-components that can be easily identified according to their specific properties [180]. In practice, when the subdomains result from an automatic mesh decomposer, the subdomains are not uniform and no characteristic subdomain stiffness factor $\rho^{(s)}$ can be defined a priori. For instance, in Problems D and E (Figs. 15.3 and 15.4), no subdomain stiffness can be defined a priori. Therefore, we need to provide a new preconditioning procedure that takes account of general heterogeneities in a self-contained manner.

17.2 Smoothed Dirichlet preconditioners

In order to define a strategy for handling generally heterogeneous domain decomposed problems, we will make use of the smoothing method developed in Part I, Sections 6 and 7. Let us recall that the scaled Dirichlet and lumped preconditioners have been mechanically derived in Section 12.2

by constructing first a compatible interface displacement, by computing the associated interface stresses and by estimating the corresponding Lagrange multipliers. For the homogeneous case, the interface displacements and the Lagrange multipliers were computed based on a simple averaging scheme taking account of the multiplicities of the interface edges. For heterogeneous problems we will define an optimum interface solution and replace the averaging scheme by the smoothing procedure which considers the subdomains as macro-elements. As in part I, we will first consider the case of two subdomains in order not to obscure the fundamentals of the method by the cumbersome notations required for the general case of multiple subdomains. The developments hereafter will be directly derived from the theory presented in Part I, Sections 6 and 7 by assuming here that all interfaces are conforming.

17.2.1 Smoothed Dirichlet preconditioner: the case of two subdomains

In the smoothing method for two-subdomains (Section 6) we assumed that the weakly compatible interface displacements are post-processed in order to define a solution satisfying exactly the compatibility, but verifying only weakly the interface equilibrium. At an iteration k of the FETI method, the subdomain displacements associated with the Lagrange multipliers $\boldsymbol{\lambda}^k$ satisfy the weak compatibility

$$\boldsymbol{p}^{j^T} \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)^k} = \boldsymbol{p}^{j^T} \boldsymbol{w}^k = 0 \qquad j = 0, \dots k \tag{17.4}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)^k} \neq \boldsymbol{0} \tag{17.5}$$

The smoothed interface solution is thus defined by Eq. (6.3) when the numbering of the interface is equal on both sides of the interface. For an arbitrary numbering of the interface d.o.f. we write

$$\hat{\boldsymbol{u}}_b^{(1),k} = (1-a^k)\boldsymbol{u}_b^{(1),k} + a^k \tilde{\boldsymbol{b}}^{(1)^T} \tilde{\boldsymbol{b}}^{(2)} \boldsymbol{u}_b^{(2),k} \tag{17.6a}$$

$$\hat{\boldsymbol{u}}_b^{(2),k} = (1-a^k)\tilde{\boldsymbol{b}}^{(2)^T} \tilde{\boldsymbol{b}}^{(1)} \boldsymbol{u}_b^{(1),k} + a^k \boldsymbol{u}_b^{(2),k} \tag{17.6b}$$

where a^k is the smoothing parameter for the k-th FETI iteration. The displacement corrections associated to the compatibility error for the k-th FETI iteration is obtained as follows:

$$\begin{aligned} \Delta \boldsymbol{u}_b^{(1),k} = \hat{\boldsymbol{u}}_b^{(1),k} - \boldsymbol{u}_b^{(1),k} &= -a^k \boldsymbol{u}_b^{(1),k} - a^k \boldsymbol{b}^{(1)^T} \boldsymbol{b}^{(2)} \boldsymbol{u}_b^{(2),k} \\ &= -a^k \boldsymbol{b}^{(1)^T} \left(\boldsymbol{b}^{(1)} \boldsymbol{u}_b^{(1),k} + \boldsymbol{b}^{(2)} \boldsymbol{u}_b^{(2),k} \right) \\ &= -a^k \boldsymbol{b}^{(1)^T} \boldsymbol{w}^k \end{aligned} \tag{17.7a}$$

$$\begin{aligned} \Delta \boldsymbol{u}_b^{(2),k} = \hat{\boldsymbol{u}}_b^{(2),k} - \boldsymbol{u}_b^{(2),k} &= -(1-a^k) \boldsymbol{b}^{(2)^T} \boldsymbol{b}^{(1)} \boldsymbol{u}_b^{(1),k} - (1-a^k) \boldsymbol{u}_b^{(2),k} \\ &= -(1-a^k) \boldsymbol{b}^{(2)^T} \left(\boldsymbol{b}^{(1)} \boldsymbol{u}_b^{(1),k} + \boldsymbol{b}^{(2)} \boldsymbol{u}_b^{(2),k} \right) \\ &= -(1-a^k) \boldsymbol{b}^{(2)^T} \boldsymbol{w}^k \end{aligned} \tag{17.7b}$$

$$\Delta \boldsymbol{u}_i(s) = -\boldsymbol{K}_{ii}^{(s)} \boldsymbol{K}_{ib}^{(s)} \Delta \boldsymbol{u}_b^{(s),k} \qquad s = 1, 2 \tag{17.7c}$$

The smoothing parameter a^k is then defined by considering the smoothed displacement field as a trial function for an a posteriori Rayleigh-Ritz procedure. Its value is obtained by expressing the corresponding kinematically admissible principle with the total energy given by (6.15)

$$\begin{aligned} \mathcal{E}(a^k) = \; & Cst - a^k \boldsymbol{w}^{k^T} \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)} \boldsymbol{b}^{(2)^T} \boldsymbol{w}^k \\ & + \frac{1}{2} a^{k^2} \boldsymbol{w}^{k^T} \left(\boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)} \boldsymbol{b}^{(1)^T} + \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)} \boldsymbol{b}^{(2)^T} \right) \boldsymbol{w}^k \end{aligned} \tag{17.8}$$

where Cst is a constant for the smoothing parameter. Minimizing $\mathcal{E}(a^k)$ yields

$$a^k = \frac{k^{(2),k}}{k^{(1),k} + k^{(2),k}} \tag{17.9a}$$

$$k^{(1),k} = \boldsymbol{w}^{k^T} \boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)} \boldsymbol{b}^{(1)^T} \boldsymbol{w}^k \tag{17.9b}$$

$$k^{(2),k} = \boldsymbol{w}^{k^T} \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)} \boldsymbol{b}^{(2)^T} \boldsymbol{w}^k \tag{17.9c}$$

Hence we find that the smoothing parameter a^k for computing a compatible displacement field at the FETI iteration k is equal to the relative interface stiffness and the interface stiffnesses are defined as the Schur complement norms of the interface displacement jumps at iteration k.

From the smoothed displacement field we then compute the interface loads associated with the displacement correction, namely

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \qquad s = 1, 2 \tag{17.10}$$

Since only a global correction proportional to $\boldsymbol{w}^k$ is allowed during the smoothing process for the displacement field, the interface equilibrium is generally satisfied only in a weak sense in that the smoothing variational principle enforces

$$\frac{\partial \mathcal{E}(a^k)}{\partial a^k} = 0 \quad \longleftrightarrow \quad \boldsymbol{w}^{k^T} \sum_{s=1}^{2} \tilde{\boldsymbol{b}}^{(s)} \Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{w}^{k^T} \boldsymbol{r}_{\Gamma_I} = 0 \tag{17.11}$$

where $\boldsymbol{r}_{\Gamma_I}$ is the interface equilibrium residual. Thus the interface stresses expressed in the Lagrange multipliers space are not identical, i.e.

$$\tilde{\boldsymbol{b}}^{(1)} \Delta \boldsymbol{f}_b^{(1)} + \tilde{\boldsymbol{b}}^{(2)} \Delta \boldsymbol{f}_b^{(2)} = \boldsymbol{r}_{\Gamma_I}^{(s)} \neq 0 \tag{17.12}$$

To define the Lagrange multipliers associated with the displacement corrections, we will thus compute a weighted sum of $\boldsymbol{b}^{(1)} \Delta \boldsymbol{f}_b^{(1)}$ and $\boldsymbol{b}^{(2)} \Delta \boldsymbol{f}_b^{(2)}$.

From a mechanical viewpoint, more credit should be given to the interface stresses defined on the softer side. Indeed, an error between the exact displacement solution and the smoothed solution leads to a large stress error on the stiffer side and to a small error on the softer side. We therefore take as weights the same weights as those defining the interface displacement corrections (15.7) and we define the Lagrange multiplier corrections as

$$\Delta \lambda^k = -a^k \boldsymbol{b}^{(1)} \Delta \boldsymbol{f}_b^{(1)} - (1 - a^k) \boldsymbol{b}^{(2)} \Delta \boldsymbol{f}_b^{(2)} \tag{17.13}$$

Let us note that using Eqs. (15.7,15.10,15.13), the total energy $\mathcal{E}(a^k)$ defined by (15.8) can be put in the form

$$\mathcal{E}(a^k) = \frac{1}{2} \boldsymbol{w}^{k^T} \Delta \boldsymbol{\lambda}^k(a^k) + Cst \tag{17.14}$$

and thus our definition of the Lagrange multiplier correction (15.13) is such that $\boldsymbol{w}^{k^T} \Delta \lambda^k$ is minimum for a^k given in (15.9).

We then choose as new direction of descent for the FETI iteration $k+1$

$$\boldsymbol{z}^k = \Delta \lambda^k = \left(a^{k^2} \boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)} \boldsymbol{b}^{(1)^T} + (1 - a^k)^2 \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)} \boldsymbol{b}^{(2)^T} \right) \boldsymbol{w}^k \tag{17.15}$$

This amounts to defining the *smoothed Dirichlet preconditioner* as

$$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}} = a^{k^2} \boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)} \boldsymbol{b}^{(1)^T} + (1 - a^k)^2 \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)} \boldsymbol{b}^{(2)^T} \tag{17.16}$$

The smoothed Dirichlet preconditioner includes a weighting with respect to the relative subdomain stiffnesses as experienced by the interface displacements. The weights in the smoothed Dirichlet preconditioner depend on the interface compatibility error and thus vary during the

FETI iterations. Hence, the smoothed Dirichlet preconditioner changes at every iteration so that a full re-orthogonalization should be considered in the conjugate gradient scheme. As already said earlier in the context of approximate Dirichlet methods (Section 14.2), this is not an issue since full reorthogonalization is required in order to avoid the degrading of the conjugacy of the descent directions.

The following remarkable issues must be pointed out:

- If both subdomains are similar, i.e. if $\boldsymbol{K}^{(2)} = \alpha \boldsymbol{K}^{(1)}$, the subdomain stiffness factors in (15.9) are such that $k^{(2),k} = \alpha k^{(1),k}$ and the smoothing parameter a^k is equal to $a^k = \alpha/(1+\alpha)$ which is not dependent on the compatibility error and remains thus unchanged during the FETI iterations. This corresponds to the case of uniform subdomains and the smoothed preconditioner introduced here is then equivalent to the heterogeneous scaling method described in the previous section. However, our smoothed preconditioner is not restricted to the case of uniform subdomains since it characterizes the subdomain stiffnesses by the Schur complement norm of the interface displacement jump.
- While the idea of estimating a local measure of the stiffness of a subdomain to build a scaling matrix for the subdomain preconditioner is not new [67,175], the derivation of the smoother presented in this work sheds new light on the precise treatment of all kinds of stiffness heterogeneities. More importantly, Eqs. (15.9) give for the first time rational estimates $k^{(1)}$ and $k^{(2)}$ of the local measures of the subdomain stiffnesses that clearly contain, among others, the effects of material properties (PDE coefficients), mesh resolution, and aspect ratio.

17.2.2 Smoothed Dirichlet preconditioner for multiple subdomains

The generalization of the smoothed Dirichlet preconditioner to the case of more than two subdomains is fairly straightforward if we use the general smoothing theory presented in Part I, Section 7. Making use of the notation conventions introduced in 7.1 and following the same reasoning as for the two-subdomain case, we define at every FETI iteration k strongly compatible displacements by introducing the displacement corrections $\Delta \boldsymbol{u}^{(s)}$. The smoothed interface displacements are given by (7.23a) where for conforming edges we have $\boldsymbol{B}^{(s,-),j} = \tilde{\boldsymbol{b}}^{(s,-),j^T}$ (Eq. (7.13)):

$$\hat{\boldsymbol{u}}_b^{(s),j} = \sum_{r:\Gamma_I^{(r)} \ni j} \beta^{(r),j}\ \tilde{\boldsymbol{b}}^{(s,-),j^T} \tilde{\boldsymbol{b}}^{(r,-),j} \boldsymbol{u}_b^{(r),j} \tag{17.17}$$

The smoothing parameters are constrained by the unit sum condition (7.24) which defines the consistency condition for the smoothing parameters,

$$\sum_{s:\Gamma_I^{(s)} \ni j} \beta^{(s),j} \ = \ 1 \tag{17.18}$$

The displacement corrections are then obtained from

$$\begin{aligned}
\Delta \boldsymbol{u}_b^{(s),j} &= \hat{\boldsymbol{u}}_b^{(s),j} - \boldsymbol{u}_b^{(s),j} = \hat{\boldsymbol{u}}_b^{(s),j} - \boldsymbol{b}^{(s,-),j^T} \boldsymbol{b}^{(s,-),j} \boldsymbol{u}_b^{(s),j} \\
&= \hat{\boldsymbol{u}}_b^{(s),j} - \sum_{r:\Gamma_I^{(s)} \ni j} \beta^{(r),j} \boldsymbol{b}^{(s,-),j^T} \boldsymbol{b}^{(s,-),j} \boldsymbol{u}_b^{(s),j} \\
&= -\boldsymbol{b}^{(s,-),j^T} \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \beta^{(r),j} \left(\boldsymbol{b}^{(r,-),j} \boldsymbol{u}_b^{(r),j} + \boldsymbol{b}^{(s,-),j} \boldsymbol{u}_b^{(s),j} \right) \\
&= -\boldsymbol{b}^{(s,-),j^T} \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \beta^{(r),j}\ \boldsymbol{w}^{(sr),j^T}
\end{aligned} \tag{17.19a}$$

$$\Delta \boldsymbol{u}_i^{(s)} \;=\; -\boldsymbol{K}_{ii}^{(s)^{-1}} \Delta \boldsymbol{u}_b^{(s)} \qquad\qquad s = 1, \dots N_s \tag{17.19b}$$

The smoothing coefficients are determined by minimizing the total energy (7.36) associated with the compatible smoothed displacements which results in the stiffness coarse (7.39)

$$\sum_{\substack{s:\Gamma_I^{(s)} \ni j \\ s \neq q}} \sum_{i:\Gamma_I^{(s)} \ni i} \sum_{\substack{p:\Gamma_I^{(p)} \ni i \\ p \neq s}} k_s^{(q),j;(p),i} \;\; \beta^{(p),i} \;=\; \tau_j \tag{17.20a}$$

$$\forall [(q),j] : \Gamma_I^{(q)} \ni j$$

$$\sum_{s:\Gamma_I^{(s)} \ni j} \beta^{(s),j} \;=\; 1 \tag{17.20b}$$

where the stiffnesses terms are defined by (7.40) and correspond to the Schur complement norm of the displacement jump on the interface edges:

$$k_s^{(q),j;(p),i} = \left(\boldsymbol{b}^{(s,-),j^T} \boldsymbol{w}^{(sq),j^k} \right)^T [\boldsymbol{S}_{bb}^{(s)}]_{j,i} \left(\boldsymbol{b}^{(s,-),i^T} \boldsymbol{w}^{(sp),i^k} \right) \tag{17.21}$$

We then define the boundary loads associated to the smoothed displacements

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{17.22}$$

The Lagrange multiplier corrections defined between any pair of subdomains connecting on an edge j are computed by assembling in a dual sense the interface loads affected by a weighting factor equal to the corresponding smoothing coefficient:

$$\boldsymbol{z}^{(sr),j^k} = \Delta \boldsymbol{\lambda}^{sr,j^k} = -\beta^{(r),j} \;\boldsymbol{b}^{(s,-),j} \Delta \boldsymbol{f}_b^{(s),j} - \beta^{(s),j} \;\boldsymbol{b}^{(r,-),j} \Delta \boldsymbol{f}_b^{(r),j} \tag{17.23}$$

In summary, considering Eqs. (15.19) and (15.23), the smoothed Dirichlet preconditioner can be expressed as

$$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}} = \sum_{s=1}^{N_s} \boldsymbol{\beta}^{(s),j} \boldsymbol{b}^{(s)} \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{\beta}^{(s),j} \tag{17.24}$$

where $\boldsymbol{\beta}^{(s),j}$ is a diagonal matrix storing for each Lagrange multiplier on edge j between $\Omega^{(s)}$ and $\Omega^{(r)}$ the coefficient $\beta^{(r),j}$. From expression (15.24), we see that the smoothed Dirichlet preconditioner defined here is symmetric. Let us note that:

- As in the case of two-subdomains, one can show that when all local Schur complements are equal to a factor, the smoothed Dirichlet preconditioner comes down to the heterogeneous scaling discussed in Section 15.1.2.
- We note that in this smoothing method, an auxiliary stiffness coarse grid must be solved at every FETI iteration. Solving this system is inexpensive since its dimension is of the order of the number of interface edges, which is generally small compared to the dimension of the dual problem. Nevertheless, setting-up the coarse grid requires to solve one Dirichlet problem per interface edge and per subdomain, which significantly increases the cost compared to the original preconditioner for which a Dirichlet problem is solved per subdomain only. In Section 15.3 we will propose several simplifications for the smoothed Dirichlet preconditioner in order to render it much less demanding.
- Our presentation of the smoothed Dirichlet preconditioner assumes that the smoothing parameters $\beta^{(s),j}$ are unique for all the d.o.f. on an edge j of the interface of $\Omega^{(s)}$. The same smoothing theory could also be derived by defining for instance one smoothing coefficient per interface edge and per direction of displacement. This technique was used for smoothing

the weakly compatible solution of the wing-box structure in Section 7.7.3: we argued that for the wing-box the stiffness discrepancy arises from the different subdomain orientations and that the relative stiffnesses should be account for depending on the displacement direction. Hence, for our Problem E (Fig. 15.4) we must define 4 smoothing parameters per edge, $\beta_v^{(s),j}$ for the tangential displacements, $\beta_{\theta_y}^{(s),j}$ for the rotations tangent to the interface, $\beta_{u,\theta_z}^{(s),j}$ for the horizontal plate displacements and $\beta_{w,\theta_x}^{(s),j}$ for the vertical plate displacements.

- If we define one smoothing parameter per interface d.o.f., one can prove that the smoothed solution is equal to the exact solution since the smoothing is based on a consistent energy minimization. However, in that case the auxiliary stiffness coarse grid is equivalent to the full primal Schur operator. Hence the smoothed Dirichlet preconditioner is then equal to the exact inverse of $\boldsymbol{F}_I$, but computing the smoothing parameters corresponds to solving the full interface problem and is thus much too expensive.

- The FETI method with the smoothed Dirichlet preconditioner is totally equivalent to the primal/dual iteration scheme presented in Sections 8.2 and 8.3

Let us also note that the smoothing procedure introduced here in the Dirichlet preconditioner can be adapted to the balanced Neumann preconditioner for the primal Schur formulation.

To illustrate the efficiency of our smoothed Dirichlet preconditioner, we apply the proposed method to the static analyses of problems D and E.

17.2.3 Applications

We first apply the smoothed Dirichlet preconditioner to Problem D consisting in a plane stress domain with soft inclusions (Fig. 15.3). We note that for the decomposition in 4×4 and 8×8 some cross-points are nodes of the soft inclusions and thus those cross-points are mechanically singular. Hence a corner coarse grid will be added to the FETI iterations (see Section 11.6) for the cross-points. In Table 15.3 we list the number of iterations required for achieving a relative primal residue lower than 10^{-6}. We consider the case when the FETI iterations are performed with or without smoothing in the preconditioner and with or without a corner coarse grid.

decomp.	Heterogeneous problem with soft inclusions			
	No corner coarse		With corner coarse	
	$\tilde{\boldsymbol{F}}_{I,D}^{-1}$	$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$	$\tilde{\boldsymbol{F}}_{I,D}^{-1}$	$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$
2×2	20	17	18	16
4×4	79	40	63	23
8×8	129	93	83	27

Table 17.3: Smoothed Dirichlet preconditioning for problem D

The results in Table 15.3 indicate that adding a corner coarse grid to Problem D is very beneficial: without a corner coarse grid, the number of iterations significantly increases when the number of subdomain is increased, even when a smoothed Dirichlet preconditioner is applied. When a corner coarse grid is included in the FETI iterations, we see from Table 15.3 that the number of iterations grows with the number of subdomains when a basic Dirichlet preconditioner is used. With a smoothed Dirichlet preconditioner the convergence is nearly independent on the decomposition, so that for the 8×8 decomposition, the number of FETI iterations with a smoothed Dirichlet preconditioner is only 1/3 of the number of iterations required when a simple Dirichlet preconditioner is used. Let us recall that for this problem, the simple heterogeneous scaling method is not applicable since no subdomain stiffness can be evaluated a priori.

Let us now perform the static analysis of Problem E. We do not introduce any corner coarse grid, but for this structure we will define 4 independent smoothing coefficients per interface. The stiffness coarse grid in the smoothing step is thus of dimension 16×16, which is small compared to the dimension of the dual operator $\boldsymbol{F}_I$. The number of iterations are given in Table 15.4. Clearly,

Wing-box structure	
$\tilde{\boldsymbol{F}}_{I,D}^{-1}$	$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$
122	32

Table 17.4: Smoothed Dirichlet preconditioning for problem E

the smoothing procedure included in the Dirichlet preconditioner is highly efficient since it reduces the number of iterations by a factor of 4. The smoothing coefficients obtained while solving the stiffness coarse grid problems during the FETI iterations are of the same order of magnitude as the coefficients found when smoothing the weakly compatible solution in Section 7.7.3, Table 7.7: the smoothing coefficients are nearly 1 for the horizontal displacements in the skins, whereas in the stiffeners, the horizontal smoothing coefficients are close to zero, indicating the relative softness of the stiffeners in the horizontal directions. For the vertical directions, the situation is exactly opposite: the vertical smoothing coefficients are very small in the skins but they are close to 1 in the stiffeners.

We thus conclude that the smoothing in the Dirichlet preconditioner takes account in an efficient manner of the stiffness discrepancies between subdomains and between directions of displacements.

17.3 Low cost smoothed preconditioners

The smoothed Dirichlet preconditioner presented in the previous section was shown to be efficient in handling general heterogeneous problems by including the relative interface stiffness in the estimation of the descent directions of the FETI iterations. Unfortunately, this smoothing procedure is expensive: considering the general smoothing equations (15.19) we see that computing the smoothing coefficients $\beta^{(r),j}$ requires solving one local Dirichlet problem per interface edge. This is obviously more expensive than using the classical Dirichlet preconditioner where only one Dirichlet problem is solved per iteration.

In real life applications, the number of interface edges per subdomain can be large. Indeed, remember that every cross-point for instance is an independent edge and thus for multiply connected subdomains the number of Dirichlet problems to be solved at every iteration implies an extra-cost which cancels all the benefits gained from ensuring a good FETI convergence.

Nevertheless, the theory related to smoothing and to the smoothed Dirichlet preconditioner provides a sound framework for developing cheap and efficient heterogeneous preconditioners. In this section we explain how the Dirichlet theory and its smoothing procedure can be modified in order to avoid the extra cost of solving multiple Dirichlet problems.

Remark

- A technique consisting in linking the smoothing coefficients at cross-points to the smoothing parameters on interfaces of multiplicity of 2 was presented in the general section on smoothing procedures (Part I, Section 7.4). Obviously this technique can be readily applied to the smoothing step included in the preconditioner. This modification generally slightly affects the efficiency of the preconditioner, but it dramatically reduces the number of Dirichlet problems to be solved at every FETI iteration since independent smoothing coefficients are defined only on interfaces of multiplicity of 2. Nevertheless, even with this modification, the cost of the smoothed Dirichlet preconditioner remains prohibitive in most practical

situations.

- When an auxiliary coarse grid problem is introduced in the FETI iterations, all the preconditioners presented in this Chapter remain valid.

17.3.1 Smoothed lumped preconditioner

The most obvious simplification one can imagine for the smoothed Dirichlet scheme is to assume that when computing the fully compatible interface displacements, the internal d.o.f. of the subdomain are assumed to be fixed, i.e. Eq. (15.19) now writes

$$\Delta \boldsymbol{u}_b^{(s),j} = -\boldsymbol{b}^{(s,-),j^T} \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \beta^{(r),j} \ \boldsymbol{w}^{(sr),j^T} \tag{17.25a}$$

$$\Delta \boldsymbol{u}_i^{(s)} = \boldsymbol{0} \qquad s = 1, \dots N_s \tag{17.25b}$$

In other words, the effect of interface displacement corrections is not back-propagated to the interior of the subdomains. The interface smoothing coefficients are then obtained by solving a stiffness coarse grid where the interface stiffnesses are now defined as the norm of the dual residue with respect to the lumped stiffness matrix $\boldsymbol{K}_{bb}^{(s)}$ (see Section 7.3).

The interface loads are computed by accepting that the internal equilibrium of the subdomains is not satisfied. The interface equilibrium then yields

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{K}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{17.26}$$

and the Lagrange multiplier corrections are still defined by (15.23). These simplifications thus amount to replacing the Schur complements in the smoothed Dirichlet method by the lumped interface operators. The resulting preconditioner can be written in the general form

$$\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}} = \sum_{s=1}^{N_s} \boldsymbol{\beta}^{(s),j} \boldsymbol{b}^{(s)} \boldsymbol{K}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{\beta}^{(s),j} \tag{17.27}$$

We can make the following comments:

- This preconditioner is similar to the lumped preconditioner introduced in Section 12.3 except that the subdomain contributions are weighted by smoothing coefficients that are determined by a stiffness coarse grid. The stiffness terms in the coarse grid are the lumped stiffness norm of the interface displacement jump and thus represent the stiffness of on interface edge when the internal d.o.f. are clamped. Therefore the preconditioner $\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}}$ defined by (15.27) is called the *smoothed lumped preconditioner*.

- Obviously, when the subdomains are uniform, the smoothing coefficients are equal to the relative subdomain stiffness defined by (15.3) and the smoothed lumped preconditioner is equivalent to the heterogeneous scaled lumped preconditioner defined in Section 15.1.2. Furthermore, for a homogeneous structure, the smoothing coefficients are equal to the edge multiplicities and the smoothed lumped preconditioner is equivalent to the basic lumped preconditioner. *The smoothed lumped preconditioner thus generalizes the lumped preconditioner to general heterogeneous problems.*

- The cost of the smoothing in the lumped preconditioner is very small. Indeed, computing the smoothing corrections (15.25) and setting-up the stiffness coarse grid does not imply any Dirichlet solution but only requires low cost multiplications by the lumped operators. The only cost overhead induced by the smoothing step in the lumped preconditioner is related to solving the stiffness coarse grid, which is negligible if the number of interface dual variables is much larger than the number of interface edges as it is in practice.

We can also define a *smoothed approximate Dirichlet preconditioner* $\tilde{\boldsymbol{F}}_{I,saD}^{k^{-1}}$ by solving the Dirichlet problems in the smoothed Dirichlet preconditioner by a low accuracy iterative scheme as explained in Section 14.2. The resulting preconditioner is midway between the smoothed Dirichlet and the smoothed lumped preconditioners.

To illustrate the efficiency of the smoothed lumped preconditioner $\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}}$, let us consider once more Problems D and E, namely the plane stress structure with soft inclusions (Figs. 15.3) and the wing-box structure (Fig. 15.4). We also test the smoothed approximate Dirichlet preconditioner when the accuracy for the internal equilibrium is set to $\epsilon_D = 0.2$. In Table 15.5 we list the number of FETI iterations needed to achieve a primal residue lower than 10^{-6} for the heterogeneous problems D and E.

Problem	decomp.	$\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}}$	$\tilde{\boldsymbol{F}}_{I,saD}^{k^{-1}}$ $\epsilon_D = 0.2$	$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$
Problem D	2×2	35	26	16
(With corner coarse)	4×4	48	34	23
	8×8	46	36	27
Problem E		52	35	32

Table 17.5: Smoothed lumped and approximate Dirichlet preconditioning for problems D and E

From this table we see that:

- The smoothed lumped preconditioner yields very good convergence rates compared to the non smoothed preconditioners (Table 15.1) and the number of iterations seems to stabilize around 50 for Problem D, which is twice the number of iterations for the smoothed Dirichlet preconditioner. Nevertheless, since a smoothing is very cheap for the lumped preconditioner while it induces a tremendous overcost for the Dirichlet one, one can state that for this case, the smoothed lumped preconditioner is overall very effective.

- The approximate Dirichlet when a smoothing procedure is included leads as expected to convergence rates situated between the exact Dirichlet preconditioner and the lumped one. The dimension of the Krylov space built in each subdomain of Problem D for approximating the inverse of $\boldsymbol{K}_{ii}^{(s)}$ is approximatively equal to 50 for all decompositions. Hence, the approximate scheme can present an advantage compared to Dirichlet preconditioners if the number of interior d.o.f. is very large.

17.3.2 Non-consistent smoothing

The smoothed preconditioners defined so far in this chapter (Dirichlet, lumped and approximate Dirichlet) are all consistent in the sense that the stiffness coarse grid determining the smoothing coefficients and the interface load computation are based on the same lumped or Dirichlet assumptions. Since the smoothed results are then used to define an approximate preconditioned direction for the FETI iterations, one is also allowed to define non-consistent smoothed preconditioners.

For instance let us assume that the stiffness coarse grid is set up by making the assumption of lumping, which is cheap. The smoothing coefficients obtained from the coarse grid are then used as scaling factors for the subsequent Dirichlet preconditioner, exactly as in the Dirichlet preconditioner (15.2c) when heterogeneous scaling is used. In this case the subdomain interior d.o.f. are exactly computed for defining the interface load distribution but they are assumed fixed for computing the smoothing coefficients. As a consequence the cost of this non-consistent Dirichlet preconditioner with a lumped coarse grid has a cost similar to the scaled Dirichlet preconditioner.

As an example, let us consider again Problem D and Problem E and let us perform a static analysis with a Dirichlet preconditioner and a lumped stiffness coarse grid. The results are shown in Table 15.6. For comparison, we also show the figures listed in Table 15.5 for the consistent smoothed Dirichlet and lumped preconditioners.

Problem	decomp.	$\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}}$	$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$ consistent	$\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$ lumped coarse
Problem D	2×2	35	16	17
(With corner coarse)	4×4	48	23	27
	8×8	46	27	28
Problem E		52	32	43

Table 17.6: Non-consistent Dirichlet preconditioner with lumped stiffness coarse grid (problems D and E)

From Table 15.6, we see that the non-consistent Dirichlet preconditioner with a lumped coarse grid is more efficient than the smoothed lumped preconditioner and it is nearly as efficient as the consistent Dirichlet preconditioner for Problem D. For Problem E, the efficiency of the non-consistent Dirichlet is midway between the consistent smoothed Dirichlet preconditioner and the lumped preconditioner. Since non-consistent smoothing for the Dirichlet preconditioner significantly reduces the preconditioning cost, it gives an interesting alternative to the consistent smoothing.

In the next paragraph we present yet another method of non-consistent smoothing which is particularly well adapted to shell problems.

17.3.3 Super-lumping for non-consistent smoothing

For the wing-box problem (Problem E), we have argued that the relative interface stiffness depends on the displacement directions and we therefore use 4 smoothing coefficients per edge. For the plane stress problem E with soft inclusions, one could argue that the relative interface stiffness depends on the position on the interface edges. Indeed, for the portion of the edge where a soft inclusion intersects with the structure matrix, the stiffnesses on both sides of the edge are very different, whereas for the portion of the interface edges with the same material on both sides, the stiffnesses are equal. Taking account of particular non-homogeneous segments on an edge would require at the limit defining one smoothing coefficient per d.o.f. on the interface, which is obviously infeasible since the stiffness coarse grid would be of the same size as the full dual problem. However, there exist one way to associate a smoothing coefficient with every interface d.o.f. while avoiding the need for solving the full interface stiffness problem. The method consists in non-consistent smoothing where the coarse grid is constructed by assuming that all interface d.o.f. are decoupled. This method is described next.

Defining $\boldsymbol{\beta}^{(r),j}$ as the diagonal matrix of smoothing parameters, one per interface d.o.f. and per subdomain, the smoothed displacements (15.17) are now expressed by

$$\hat{\boldsymbol{u}}_b^{(s),j} = \sum_{r:\Gamma_I^{(r)} \ni j} \tilde{\boldsymbol{b}}^{(s,-),j^T} \tilde{\boldsymbol{b}}^{(r,-),j} \boldsymbol{\beta}^{(r),j} \boldsymbol{u}_b^{(r),j} \tag{17.28}$$

The consistency condition (7.24) on the smoothing parameters must now be satisfied for every set of matching d.o.f. and we write

$$\sum_{r:\Gamma_I^{(r)} \ni j} \tilde{\boldsymbol{b}}^{(r,-),j} \boldsymbol{\beta}^{(r),j} \tilde{\boldsymbol{b}}^{(r,-),j^T} = \boldsymbol{I} \qquad \forall \text{ edge } \ j \tag{17.29}$$

Let us define the Boolean operator $\boldsymbol{c}^{(sr),j}$ expressing the correspondence between the d.o.f. numbering on edge j between $\Omega^{(s)}$ and $\Omega^{(r)}$ by

$$\boldsymbol{c}^{(sr),j} = \tilde{\boldsymbol{b}}^{(s,-),j^T} \tilde{\boldsymbol{b}}^{(r,-),j} \tag{17.30}$$

and we state that $\boldsymbol{c}^{(ss),j} = \boldsymbol{I}$. Premultiplying the consistency condition (15.29) by $\boldsymbol{b}^{(s,-),j^T}$, post-multiplying by $\boldsymbol{b}^{(s,-),j}$ and recalling from Eq. (7.6) that $\boldsymbol{b}^{(s,-),j^T} \boldsymbol{b}^{(s,-),j} = \boldsymbol{I}$ for conforming interfaces, we find the alternative form

$$\boldsymbol{\beta}^{(s),j} + \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j} \boldsymbol{\beta}^{(r),j} \boldsymbol{c}^{(sr),j^T} = \boldsymbol{I} \qquad \forall \text{ edge } \ j \tag{17.31}$$

The smoothing corrections can now be computed by using successively the condition (15.31) and the expression (15.28):

$$\begin{aligned}
\Delta \boldsymbol{u}_b^{(s),j} &= \hat{\boldsymbol{u}}_b^{(s),j} - \boldsymbol{u}_b^{(s),j} \\
&= \hat{\boldsymbol{u}}_b^{(s),j} - \left(\boldsymbol{\beta}^{(s),j} + \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j} \boldsymbol{\beta}^{(r),j} \boldsymbol{c}^{(sr),j^T} \right) \boldsymbol{u}_b^{(s),j} \\
&= \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j} \boldsymbol{\beta}^{(r),j} \left(\boldsymbol{u}_b^{(r),j} - \boldsymbol{c}^{(sr),j^T} \boldsymbol{u}_b^{(s),j} \right) \\
&= \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j} \boldsymbol{\beta}^{(r),j} \boldsymbol{b}^{(r,-),j^T} \left(\boldsymbol{b}^{(r,-),j} \boldsymbol{u}_b^{(r),j} + \boldsymbol{b}^{(s,-),j} \boldsymbol{u}_b^{(s),j} \right) \\
&= - \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j} \boldsymbol{\beta}^{(r),j} \ \boldsymbol{c}^{(sr),j^T} \ \boldsymbol{b}^{(s,-),j^T} \ \boldsymbol{w}^{(sr),j^k}
\end{aligned} \tag{17.32}$$

The smoothing coefficients are then determined by a Rayleigh-Ritz procedure. This time the stiffness coarse grid is constructed by considering the total energy associated with the fully diagonal interface restriction of the local operators, i.e.

$$\mathcal{E}(\beta^{(s),j}) = Cst + \frac{1}{2} \sum_{s=1}^{N_s} \Delta \boldsymbol{u}_b^{(s)^T} \boldsymbol{K}_{bb,diag}^{(s)} \ \Delta \boldsymbol{u}_b^{(s)} \tag{17.33}$$

where $\boldsymbol{K}_{bb,diag}^{(s)}$ is the diagonal matrix corresponding to $\boldsymbol{K}_{bb}^{(s)}$. This expression thus differs from the lumped formulation in that not only the coupling of the interface displacements through the subdomain internal d.o.f. is neglected, but even the coupling between interface d.o.f. is now dropped. Therefore, this smoothing procedure is called *super-lumped smoothing*. Substituting (15.32) in the super-lumped energy expression (15.33), we obtain the coarse grid problem (15.20) where the stiffness coefficients are now defined by the diagonal terms of the local operators. We thus find a stiffness problem which is totally decoupled for each set of smoothing parameters pertaining to matching d.o.f. Taking account of the consistency relation (15.31), we find that the smoothing parameters are given by

$$\boldsymbol{\beta}^{(s),j} = \left[\boldsymbol{K}_{bb,diag}^{(s)} \right]_j \left\{ \sum_{r:\Gamma_I^{(r)} \ni j} \boldsymbol{c}^{(sr),j} \left[\boldsymbol{K}_{bb,diag}^{(r)} \right]_j \boldsymbol{c}^{(sr),j^T} \right\}^{-1} \tag{17.34}$$

This last result shows that the smoothing parameter $\beta_n^{(s),j}$ for the d.o.f. n of the subdomain $\Omega^{(s)}$ on edge j is found as follows: get the local diagonal stiffnesses corresponding to the d.o.f. that are constrained to be equal and sum them up. This sum is the assembled diagonal stiffness. The smoothing parameter is then the ratio of the local and assembled diagonal stiffness.

Now that the smoothing coefficients are defined for each d.o.f., the interface loads associated with the smoothed displacements are computed as usual by

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \qquad \text{Dirichlet} \tag{17.35a}$$

or

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{K}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \qquad \text{lumped} \tag{17.35b}$$

and finally the Lagrange multiplier correction is given by

$$\boldsymbol{z}^{(sr),j^k} = \Delta \boldsymbol{\lambda}^{sr,j^k} = -\boldsymbol{b}^{(s,-),j}\ \boldsymbol{\beta}^{(r),j}\ \Delta \boldsymbol{f}_b^{(s),j} - \boldsymbol{b}^{(r,-),j}\ \boldsymbol{\beta}^{(s),j}\ \Delta \boldsymbol{f}_b^{(r,j)} \tag{17.36}$$

We note that:

- In the super-lumped method, each interface d.o.f. u_j is considered as a completely independent variable to which a simple spring of stiffness $\boldsymbol{K}_{jj}^{(s)}$ is associated. The smoothing parameters are then defined in order to solve exactly the problem of closing the displacement gap while satisfying the super-lumped equilibrium (Fig. 15.6). Implementing this super-lumped smoothing is thus straightforward and cheap since it only requires computing the assembled diagonal stiffness for the interface nodes and then computing relative local stiffnesses.

- Comparing the definition of the super-lumped smoothing coefficients (15.34) and the heterogeneous edge scaling coefficients (15.3), we can interpret the super-lumped smoothing coefficients as a heterogeneous scaling at the d.o.f. level where the d.o.f. stiffness is represented by its associated diagonal term. In particular, if the subdomains are uniform, the smoothing coefficients have the same value as the relative subdomain stiffnesses $\overline{\rho}^{(s),j}$ defined in (15.3) and the super-lumped smoothing is equivalent to the heterogeneous scaling. However the super-lumped is not restricted to uniform subdomains as is the heterogeneous scaling procedure.

This super-lumped smoothing has been applied to the Dirichlet, lumped and approximate Dirichlet preconditioners for the test Problems D and E. The number of iterations of the FETI

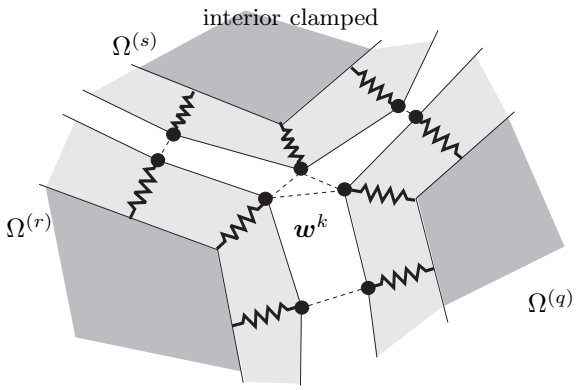


Figure 17.6: Schematic interpretation of super-lumped smoothing

Problem	decomp.	consistent smoothing $\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}}$	consistent smoothing $\tilde{\boldsymbol{F}}_{I,saD}^{k^{-1}}$	consistent smoothing $\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$	super-lumped smoothing $\tilde{\boldsymbol{F}}_{I,sL}^{k^{-1}}$	super-lumped smoothing $\tilde{\boldsymbol{F}}_{I,saD}^{k^{-1}}$	super-lumped smoothing $\tilde{\boldsymbol{F}}_{I,sD}^{k^{-1}}$
Problem D	2×2	35	26	16	36	27	17
(With corner	4×4	48	34	23	47	35	26
coarse)	8×8	46	36	27	44	31	25
Problem E		52	43	32	50	30	25

Table 17.7: Super-lumped smoothing for the FETI preconditioners (problems D and E)

method for achieving a primal residue lower than 10^{-6} are listed in Table 15.7, together with the consistent smoothing counterparts for comparison's sake.

In Table 15.7, we see that for the plane stress problem with soft inclusions (Problem D), the super-lumped smoothing leads to convergence rates very similar to the convergence rates obtained with consistent smoothing. For the wing-box problem (Problem E), the super-lumped smoothing leads to even better convergence rates. Recalling that for this problem the non-smoothed lumped and Dirichlet preconditioners required respectively 128 and 122 iterations (Table 15.1), and being said that the super-lumped smoothing induces only a minor computational overhead, we see that the super-lumped smoothing is a very powerful procedure for reducing the global cost of solving the interface problem, especially for the cases represented by Problems D and E, namely when the classical preconditioners cannot work properly due to high heterogeneities in the vicinity of the interface edges. For shell problems, such as the wing-box structure, our super-lumped smoothing procedure takes care of the stiffness heterogeneities at the d.o.f. level and thus also allows to correctly account for stiffness discrepancies due to subdomain orientation.

In the next section we present more application examples to show the efficiency of super-lumped smoothing.

17.4 Application examples

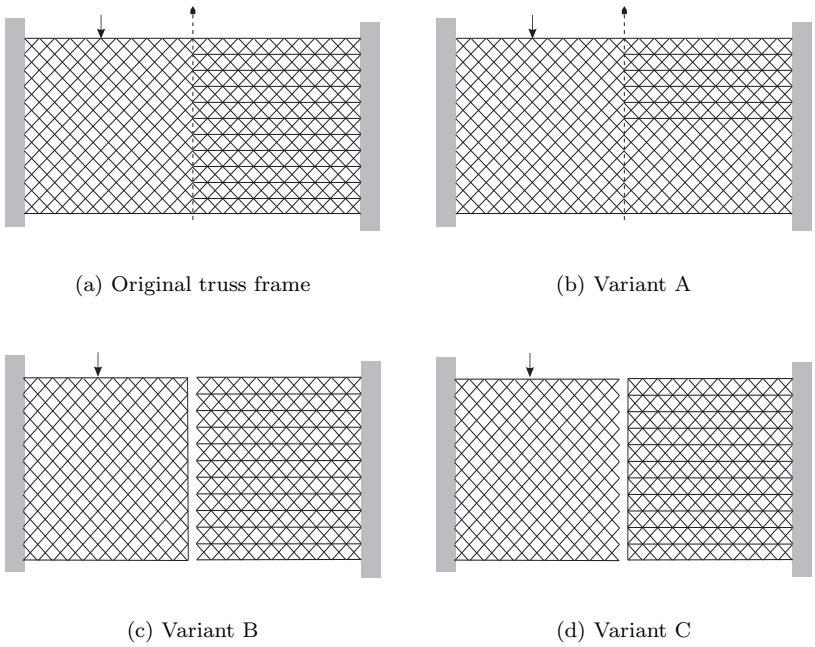
In this section we will first present some results on the efficiency of smoothed preconditioners for a truss structure. We then discuss a practical application consisting in a full model of a wing.

17.4.1 Preconditioners for a truss frame structure

Let us consider the truss frame analyzed earlier in Part I Sections 5.6, 6.3.1 and 8.2 (Fig. 5.11). To investigate different structural layouts for truss frames, we consider the following variants of the original truss frame depicted in Fig 15.7(a). In variant A, we assume that the lower half of the horizontal bars in the right-hand side domain are removed (Fig. 15.7(b)). In variant B, we add vertical bars linking the points on the interface in order to increase the vertical stiffness of the structure and we assume that those vertical reinforcements are part of the left-hand subdomain (Fig. 15.7(c)). Finally, for variant C, we assume that the vertical interface bars belong to the right-hand subdomain (Fig. 15.7(d)). We then apply all possible combinations of the preconditioning methods proposed in this chapter: the preconditioner is either of the lumped type, of the approximate Dirichlet type or of the exact Dirichlet type. The preconditioners can be non-smoothed, consistently smoothed or smoothed in a non-consistent way by a super-lumped scheme. Finally we also consider the particular non-consistent lumped smoothing for the Dirichlet preconditioner. Note that for the smoothed preconditioners, we define two smoothing parameters on each side of the interface, one for the horizontal displacements and one for the vertical ones.

The number of iterations for achieving a relative primal residue lower than 10^{-6} are listed in Table 15.8.

In this table we have underlined the best convergence results and we make the following observations:



(a) Original truss frame

(b) Variant A

(c) Variant B

(d) Variant C

Figure 17.7: Truss frame structure and variants

Precond.	Lumped			Dirichlet			
smoothing	none	cons.	sup.-lump.	none	cons.	lumped	super-lump.
Orig. truss	14	14	14	16	8	13	13
Variant A	15	15	15	14	9	13	13
Variant B	18	19	19	17	10	14	14
Variant C	14	13	13	21	7	17	16

Precond.	Approx. Dirich. ϵ_D=0.2		
smoothing	none	cons.	sup.-lump.
Orig. truss	14	12	12
Variant A	14	13	13
Variant B	17	16	16
Variant C	18	13	14

Table 17.8: FETI convergence for truss frames: comparing different heterogeneous preconditioners

- Noting that the dimension of the interface problem is 24, we can state that the lumped preconditioner and the approximate Dirichlet one are not efficient even when smoothing is applied. This highlights the high coupling between all degrees of freedom for a truss frame structure and thus stresses the fact that in this case an exact Dirichlet preconditioner is required.

- The Dirichlet preconditioner can be rendered efficient only if a consistent smoothing scheme is applied due to the fact that the exact subdomain stiffnesses at the interface can be evaluated only by taking into account the mechanical effect of the subdomain interiors onto the interface boundary. This is an other illustration of the high coupling between the d.o.f.

- The most spectacular improvement obtained by using a smoothing procedure is obtained for the variant C of the truss. In this case, the vertical bars on the interface belong to the right-hand subdomain which is also very much stiffened in the horizontal direction by the horizontal bars. Hence, in variant C, there exists a high difference between the subdomain stiffnesses in both directions. The results show that in that case heterogeneity is beneficial for convergence if however it is efficiently accounted for in the preconditioner.

17.4.2 A realistic wing structural model

In this example we present the results obtained by the research team at Boulder (Farhat *et al.*) on a realistic wing model (Fig. 15.8) when applying our super-lumped smoothing procedure. The skins, spars and ribs are modeled using beam, bar, membrane and shell elements. In a first model, the structure is discretized using 456 nodes and 2736 d.o.f.

In this structure, there are several structural intersections for instance between the stiffeners and the skin. A corner coarse grid is defined for the cross-points. From our discussion on smoothed preconditioners in this chapter, we expect that the super-lumped smoothing method is the most appropriate to ensure good convergence. The first model is decomposed in 2, 4 and 8 subdomains (Fig. 15.9). A Dirichlet preconditioner is applied with a multiplicity scaling and with a super-lumped smoothing. The iteration numbers of FETI are given in Table 15.9 for a relative primal error lower than $5\ 10^{-4}$.

A second refined model of the wing is then defined and the mesh now contains 8370 nodes and 50220 d.o.f. This model is decomposed into 16 and 32 subdomains. The number of FETI iterations are also reported in Table 15.9.

		Dirichlet preconditioner	
	# of subdomains	$1/m_j$ scaling	super-lumped smoothing
Mesh 1	2	56	39
(2736 d.o.f.)	4	122	78
	8	201	136
Mesh 2	16	142	126
(50220 d.o.f.)	32	162	156

Table 17.9: Efficiency of super-lumped smoothing for a realistic wing model

Table 15.9 indicates that for the first wing structural model, the super-lumped smoothing reduces the number of iterations almost by a factor of 2. Let us recall that the super-lumped smoothing incurs only a very minor computational overhead and thus it seems very effective for this problem. For the second meshing, we observe that the FETI convergence is already good even without smoothing and the super-lumped smoothing procedure only slightly enhances the convergence rate. This can be explained by the fact that for the second refined model the interface between subdomains are not coincident with structural edges so that no important stiffness discrepancies arise from subdomain orientations.

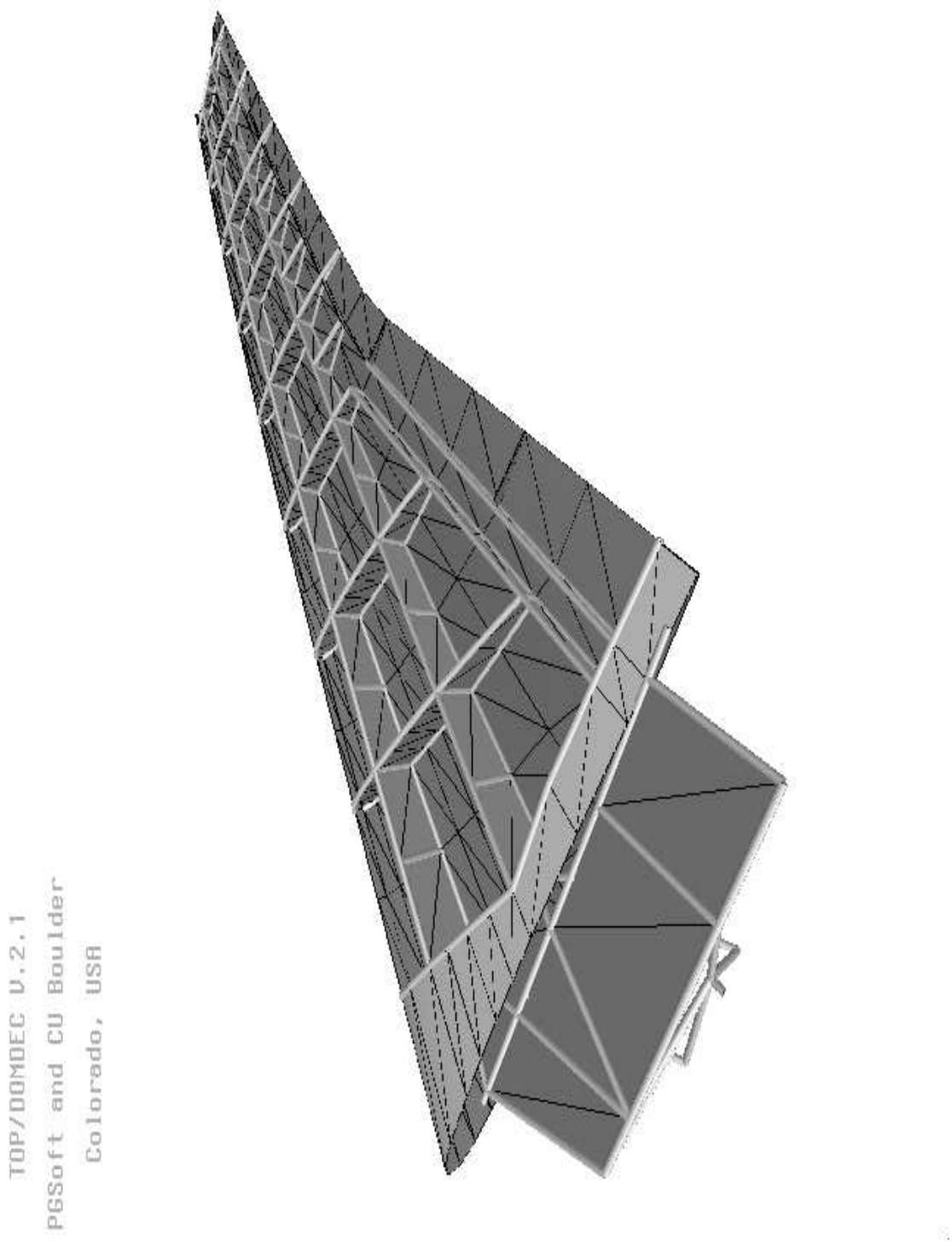


Figure 17.8: A realistic wing model

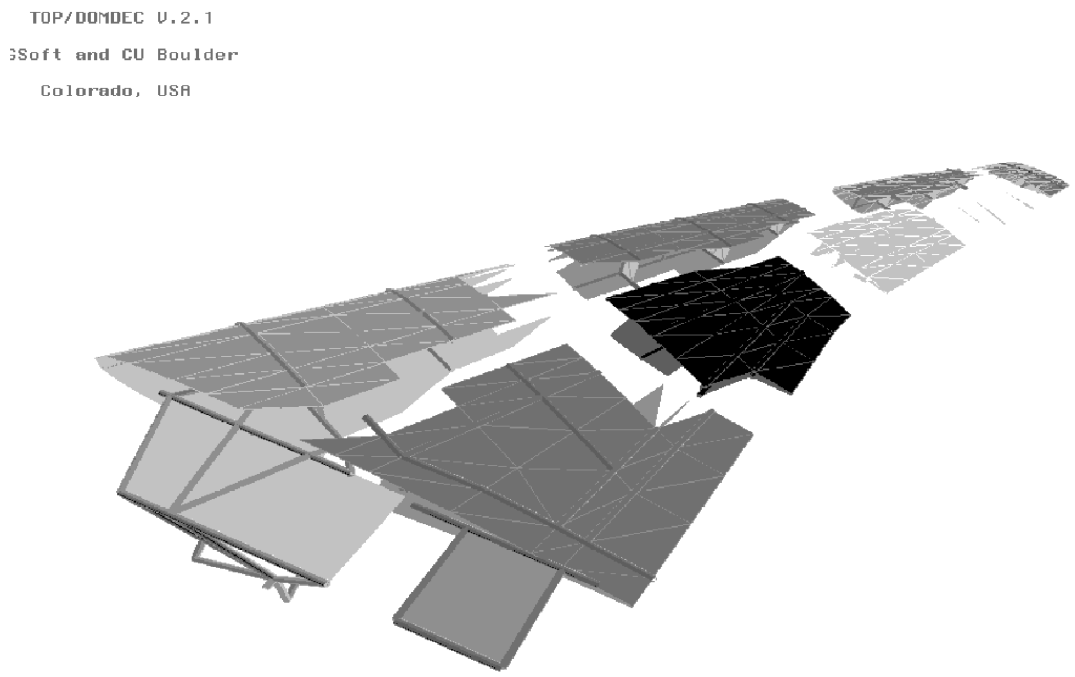


Figure 17.9: Decomposition of the wing model in 8 subdomains

17.5 Correct treatment of the checkerboard problem

Let us consider the very peculiar situation when very soft substructured and very stiff ones are laid out in a checkerboard-like fashion, such as in Problem B described earlier in Fig. 15.1. Here we will consider the case of 16 subdomains forming a checkerboard plane stress problem (Fig. 15.10) and we assume that the stiff domains are made of steel whereas the soft domains are made of rubber which Young's modulus is 4098 times smaller than steel. Each domain contains 10×10 four noded elements.

All subdomains being uniform, we will use heterogeneous scaling (which is equivalent to the super-lumped smoothing for uniform domains). The FETI iterations will be preconditioned by using the lumped preconditioner as well as the approximate and exact Dirichlet ones. Moreover we will assume first that no preconditioning is introduced in the coarse grid of the floating modes ($\boldsymbol{Q} = \boldsymbol{I}$) and then the respective preconditioners will be added to the coarse grid ($\boldsymbol{Q} = \tilde{\boldsymbol{F}}_I^{-1}$).

Table 15.10 gives the number of FETI iterations for a primal residue lower than 10^{-6}: when a Dirichlet preconditioner is used (with heterogeneous scaling) and when the coarse grid related to the local rigid body displacements is preconditioned, the FETI method converges already after 3 iterations!

This very rapid convergence can be explained by noting that in this checkerboard problem, the solution is nearly completely determined by the sole cross-point loads and for the checkerboard layout, the corner loads are mainly determined by the self-equilibrium of the interface loads. In other words, the steel subdomains move as if they were rigid, and since the rigid body displacements are determined exactly by the coarse grid relative to $\boldsymbol{G}_I$, the exact interface forces are found. Since the heterogeneous scaled Dirichlet preconditioner is very good at estimating the interface loads associated with global modes in this highly heterogeneous case, the iterations converge very rapidly. When the coarse grid problem is preconditioned by an approximate Dirichlet method, the number

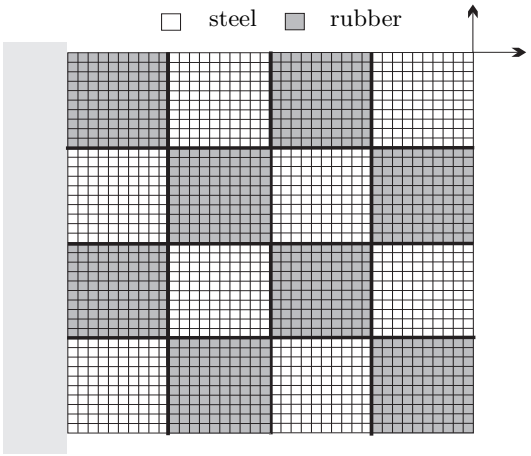


Figure 17.10: The checkerboard problem

Preconditioners (heterogeneous scaling)	Coarse grid preconditioning	
	$\boldsymbol{Q} = \boldsymbol{I}$	$\boldsymbol{Q} = \tilde{\boldsymbol{F}}_I^{-1}$
lumped	38	22
approx. Dirichlet	26	9
Dirichlet	20	3

Table 17.10: Preconditioning the checkerboard problem

of iterations is equal to 9 and for the lumped preconditioner it increases to 22.

When no preconditioning is applied to the coarse grid, i.e. $\boldsymbol{Q} = \boldsymbol{I}$, the convergence rate is similar to the convergence rate obtained for a homogeneous problem, but it does not take advantage of the highly beneficial effect of the coarse grid in this case.

This example illustrates what we already stated earlier in Section 12.5, namely that when the rigid body displacements of the subdomains induce on the interface a peculiar load distribution due for instance to heterogeneities, it is important to precondition the coarse grid by setting $\boldsymbol{Q} = \tilde{\boldsymbol{F}}_{I,D}^{-1}$. Remember that computing $\tilde{\boldsymbol{F}}_{I,D}^{-1}\boldsymbol{G}_I$ requires solving only a small number of local Dirichlet problems (Section 12.5).

17.6 Special iterative method for the two-subdomain case

17.6.1 Quasi-optimality of the smoothed preconditioners

The smoothing method presented in Section 15.2 is based on a Rayleigh-Ritz procedure which implies minimizing the total energy associated with the compatible smoothed displacements. For the two-subdomain case, we have shown that our choice of the correction for Lagrange multipliers (15.13) allows to write the total energy as in (15.14), namely

$$\mathcal{E}(a^k) = \frac{1}{2}\boldsymbol{w}^{k^T}\Delta\boldsymbol{\lambda}^k(a^k) + Cst = \frac{1}{2}\boldsymbol{w}^{k^T}\boldsymbol{z}^k(a^k) + Cst \tag{15.14}$$

and the smoothing parameter a^k solution of the consistent stiffness coarse grid thus satisfies the minimum relation

$$\min_a \boldsymbol{w}^{k^T}\boldsymbol{z}^k(a) \;=\; \boldsymbol{w}^{k^T}\boldsymbol{z}^k(a^k) \tag{17.37}$$

Noting that the projected direction for the FETI iterations is $\boldsymbol{p}^k = \boldsymbol{P}\boldsymbol{z}^k$ and that $\boldsymbol{P}^T\boldsymbol{w}^k = \boldsymbol{0}$ according to the very definition of the dual residue, we can also write

$$\min_a \boldsymbol{w}^{k^T}\boldsymbol{p}^k(a) \;=\; \boldsymbol{w}^{k^T}\boldsymbol{p}^k(a^k) \tag{17.38}$$

These relations can also be generalized for the case of multiple subdomains

$$\begin{aligned} \min_\beta \mathcal{E}(\beta^{(r),j}) &= \min_\beta \boldsymbol{w}^{k^T}\boldsymbol{z}^k(\beta^{(r),j}) && (17.39)\\ &= \min_\beta \boldsymbol{w}^{k^T}\boldsymbol{p}^k(\beta^{(r),j}) = \boldsymbol{w}^{k^T}\boldsymbol{p}^k(\beta^{(r),j^k}) && (17.40) \end{aligned}$$

where $\beta^{(r),j^k}$ are the solutions of the stiffness coarse grid at iteration k. This last relation indicates two important facts:

- The projection by the operator $\boldsymbol{P}$ does not affect the optimality of the preconditioned direction of descent in the sense that the smoothing parameters also ensure that the work produced by the Lagrange multiplier correction $\boldsymbol{p}^k$ with respect to the displacement gap $\boldsymbol{w}^k$ is minimum.

- Considering that at iteration k of the FETI method one minimizes

$$\phi(\boldsymbol{\lambda}^{k+1}) = \boldsymbol{P}^T\left(\frac{1}{2}\boldsymbol{\lambda}^{k+1^T}\boldsymbol{F}_I\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^{k+1^T}\boldsymbol{d}\right) \tag{17.41}$$

  the smoothing method can by interpreted as a quasi-minimization of $\phi(\boldsymbol{\lambda}^{k+1})$ with respect to the smoothing parameters: let us note that the Lagrange multipliers $\boldsymbol{\lambda}^{k+1}$ write

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k\boldsymbol{P}\boldsymbol{z}^k(\beta^{(r),j}) \tag{17.42}$$

  The functional $\phi(\boldsymbol{\lambda}^{k+1})$ can then be put in the form

$$\begin{aligned} \phi(\boldsymbol{\lambda}^{k+1}) &= \boldsymbol{P}^T\left(\frac{1}{2}\boldsymbol{\lambda}^{k^T}\boldsymbol{F}_I\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k^T}\boldsymbol{d}\right.\\ &\qquad \left. + \frac{1}{2}\eta^{k^2}\boldsymbol{p}^{k^T}\boldsymbol{F}_I\boldsymbol{p}^k + \eta^k\boldsymbol{p}^{k^T}(\boldsymbol{F}_I\boldsymbol{\lambda}^k - \boldsymbol{d})\right)\\ &= \phi(\boldsymbol{\lambda}^k) \qquad (17.43)\\ &\quad + \boldsymbol{P}^T\left(\frac{1}{2}\eta^{k^2}\boldsymbol{p}^{k^T}(\beta^{(r),j})\boldsymbol{F}_I\boldsymbol{p}^k(\beta^{(r),j}) + \eta^k\boldsymbol{p}^{k^T}(\beta^{(r),j})\;\boldsymbol{w}^k\right) \end{aligned}$$

  From Eq. (15.40) we conclude that the smoothing procedure minimizes the last term of $\phi(\boldsymbol{\lambda}^{k+1})$ in (15.43). Noting that the expression $\boldsymbol{p}^k$ is quadratic in terms of the smoothing parameters, minimizing the term $\boldsymbol{p}^{k^T}\boldsymbol{F}_I\boldsymbol{p}^k$ would lead to a non-linear problem and is thus not considered. The subsequent Conjugate Gradient step then minimizes $\phi(\boldsymbol{\lambda}^{k+1})$ with respect to the direction amplitude η^k.

We now investigate if it is possible to minimize exactly the functional $\phi(\boldsymbol{\lambda}^{k+1})$ with respect to the smoothing parameters. In other words, we will see if it is possible to use the boundary stresses obtained from the local Dirichlet problems as descent directions for the Conjugate Gradient.

17.6.2 Simultaneous descent directions

Let us consider the case of a two-subdomain decomposition where both subdomains are non-floating. At iteration k, we first solve the local Dirichlet problem consisting in enforcing as

interface displacement the compatibility gap $\boldsymbol{r}^k$ and we compute the boundary reaction forces which write:

$$\Delta \boldsymbol{f}^{(s),k} = \boldsymbol{S}_{bb}^{(s)} \boldsymbol{b}^{(s)^T} \boldsymbol{r}^k \qquad s = 1,2 \tag{17.44}$$

In the smoothing methods presented before we have evaluated an approximate correction to the Lagrange multipliers by computing a weighted inter-subdomain stress based on $\Delta \boldsymbol{f}^{(s),k}$. Let us now consider the boundary loads on each side of the interface as descent directions for the minimization of $\phi(\boldsymbol{\lambda}^{k+1})$ by writing

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^{(1),k} \boldsymbol{b}^{(1)} \Delta \boldsymbol{f}^{(1),k} + \eta^{(2),k} \boldsymbol{b}^{(2)} \Delta \boldsymbol{f}^{(2),k} \tag{17.45}$$

Assuming now that instead of utilizing directly the boundary forces, we use

$$\Delta \boldsymbol{\lambda}^{(1),k} = \boldsymbol{b}^{(1)} \Delta \boldsymbol{f}^{(1),k} + \sum_{s=1}^{2} \sum_{l=0}^{k-1} \alpha^{(s),l} \Delta \boldsymbol{\lambda}^{(s),l} \tag{17.46}$$

$$\Delta \boldsymbol{\lambda}^{(2),k} = \boldsymbol{b}^{(2)} \Delta \boldsymbol{f}^{(2),k} + \alpha^{(2),k} \Delta \boldsymbol{f}^{(1),k} + \sum_{s=1}^{2} \sum_{l=0}^{k-1} \beta^{(s),l} \Delta \boldsymbol{\lambda}^{(s),l} \tag{17.47}$$

such that $\Delta \boldsymbol{\lambda}^{(2),k}$ is orthogonalized with respect to $\Delta \boldsymbol{\lambda}^{(1),k}$ and that both directions are orthogonal to all previous directions:

$$\Delta \boldsymbol{\lambda}^{(1),k^T} \boldsymbol{F}_I \Delta \boldsymbol{\lambda}^{(2),k} = 0 \tag{17.48}$$

$$\Delta \boldsymbol{\lambda}^{(s),k^T} \boldsymbol{F}_I \Delta \boldsymbol{\lambda}^{(r),l} = 0 \qquad r,s = 1,2 \qquad l = 0, \ldots k-1 \tag{17.49}$$

The Lagrange multipliers for iteration k thus write

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^{(1),k} \Delta \boldsymbol{\lambda}^{(1),k} + \eta^{(2),k} \Delta \boldsymbol{\lambda}^{(2),k} \tag{17.50}$$

and the direction amplitudes $\eta^{(s),k}$ are determined by minimizing the functional $\phi(\boldsymbol{\lambda}^{k+1})$ which writes

$$\phi(\boldsymbol{\lambda}^{k+1}) = \frac{1}{2} \boldsymbol{\lambda}^{k+1^T} \boldsymbol{F}_I \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^{k+1^T} \boldsymbol{d} \tag{17.51}$$

$$\begin{aligned} &= \phi(\boldsymbol{\lambda}^k) \\ &\quad + \frac{1}{2} \eta^{(1),k^2} (\Delta \boldsymbol{\lambda}^{(1),k})^T \boldsymbol{F}_I \Delta \boldsymbol{\lambda}^{(1),k} - \eta^{(1),k} (\Delta \boldsymbol{\lambda}^{(1),k})^T \boldsymbol{d} \\ &\quad + \frac{1}{2} \eta^{(2),k^2} (\Delta \boldsymbol{\lambda}^{(2),k})^T \boldsymbol{F}_I \Delta \boldsymbol{\lambda}^{(2),k} - \eta^{(2),k} (\Delta \boldsymbol{\lambda}^{(2),k})^T \boldsymbol{d} \end{aligned} \tag{17.52}$$

The amplitudes $\eta^{(s),k}$ are thus obtained exactly as in a Conjugate Gradient method, namely

$$\eta^{(s),k} = \frac{(\Delta \boldsymbol{\lambda}^{(s),k})^T \boldsymbol{d}}{(\Delta \boldsymbol{\lambda}^{(s),k})^T \boldsymbol{F}_I \Delta \boldsymbol{\lambda}^{(s),k}} \qquad s = 1,2 \tag{17.53}$$

The fundamental feature in this algorithm is that *although at each iteration one minimizes with respect to two descent directions, the cost of an iteration is equivalent to the cost of a normal conjugate gradient direction.* Indeed, recalling that the descent directions are obtained from a Dirichlet problem, we note that

$$\begin{aligned} \boldsymbol{F}_I \boldsymbol{b}^{(1)} \Delta \boldsymbol{f}^{(1),k} &= \left(\boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)^{-1}} \boldsymbol{b}^{(1)^T} + \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)^{-1}} \boldsymbol{b}^{(2)^T} \right) \boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)} \boldsymbol{b}^{(1)^T} \boldsymbol{r}^k \\ &= \boldsymbol{r}^k + \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)^{-1}} \boldsymbol{b}^{(2)^T} \boldsymbol{b}^{(1)} \Delta \boldsymbol{f}^{(1),k} \end{aligned} \tag{17.54}$$

$$\begin{aligned} \boldsymbol{F}_I \boldsymbol{b}^{(2)} \Delta \boldsymbol{f}^{(2),k} &= \left(\boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)^{-1}} \boldsymbol{b}^{(1)^T} + \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)^{-1}} \boldsymbol{b}^{(2)^T} \right) \boldsymbol{b}^{(2)} \boldsymbol{S}_{bb}^{(2)} \boldsymbol{b}^{(2)^T} \boldsymbol{r}^k \\ &= \boldsymbol{b}^{(1)} \boldsymbol{S}_{bb}^{(1)^{-1}} \boldsymbol{b}^{(1)^T} \boldsymbol{b}^{(2)} \Delta \boldsymbol{f}^{(2),k} + \boldsymbol{r}^k \end{aligned} \tag{17.55}$$

We see that applying the dual interface operator to the descent directions requires only one Neumann solution per subdomain!

This Conjugate Gradient with simultaneous directions is summarized in Algorithm 15.1.

Algorithm 17.1 Conjugate Gradient with simultaneous Dirichlet directions (2 subdomains)

$k = 0$, $\boldsymbol{\lambda}^0 = \mathbf{0}$, $\boldsymbol{r}^0 = \boldsymbol{d}$

while $\|\boldsymbol{r}^k\| > \epsilon\|\boldsymbol{d}|$ **do**

$$(\Delta\boldsymbol{\lambda})^{(1),k} = \boldsymbol{b}^{(1)}\boldsymbol{S}_{bb}^{(1)}{\boldsymbol{b}^{(1)}}^T\boldsymbol{r}^k$$
$$(\Delta\boldsymbol{\lambda})^{(2),k} = \boldsymbol{b}^{(2)}\boldsymbol{S}_{bb}^{(2)}{\boldsymbol{b}^{(2)}}^T\boldsymbol{r}^k$$
$$(\boldsymbol{F\lambda})^{(1),k} = \boldsymbol{r}^k + \boldsymbol{b}^{(2)}{\boldsymbol{S}_{bb}^{(2)}}^{-1}{\boldsymbol{b}^{(2)}}^T(\Delta\boldsymbol{\lambda})^{(1),k}$$
$$(\boldsymbol{F\lambda})^{(2),k} = \boldsymbol{r}^k + \boldsymbol{b}^{(1)}{\boldsymbol{S}_{bb}^{(1)}}^{-1}{\boldsymbol{b}^{(1)}}^T(\Delta\boldsymbol{\lambda})^{(2),k}$$

-Orthogonalization with respect to previous directions-

for $l = 1$ to $k-1$ **do**

$$\alpha^{(1),l} = (\boldsymbol{F\lambda})^{(1),l^T}(\Delta\boldsymbol{\lambda})^{(1),k}/(\boldsymbol{\lambda F\lambda})^{(1),l}$$
$$\alpha^{(2),l} = (\boldsymbol{F\lambda})^{(2),l^T}(\Delta\boldsymbol{\lambda})^{(1),k}/(\boldsymbol{\lambda F\lambda})^{(2),l}$$
$$\beta^{(1),l} = (\boldsymbol{F\lambda})^{(1),l^T}(\Delta\boldsymbol{\lambda})^{(2),k}/(\boldsymbol{\lambda F\lambda})^{(1),l}$$
$$\beta^{(2),l} = (\boldsymbol{F\lambda})^{(2),l^T}(\Delta\boldsymbol{\lambda})^{(2),k}/(\boldsymbol{\lambda F\lambda})^{(2),l}$$
$$(\Delta\boldsymbol{\lambda})^{(1),k} = (\Delta\boldsymbol{\lambda})^{(1),k} - \alpha^{(1),l}(\Delta\boldsymbol{\lambda})^{(1),l} - \alpha^{(2),l}(\Delta\boldsymbol{\lambda})^{(2),l}$$
$$(\Delta\boldsymbol{\lambda})^{(2),k} = (\Delta\boldsymbol{\lambda})^{(2),k} - \beta^{(1),l}(\Delta\boldsymbol{\lambda})^{(1),l} - \beta^{(2),l}(\Delta\boldsymbol{\lambda})^{(2),l}$$
$$(\boldsymbol{F\lambda})^{(1),k} = (\boldsymbol{F\lambda})^{(1),k} - \alpha^{(1),l}(\boldsymbol{F\lambda})^{(1),l} - \alpha^{(2),l}(\boldsymbol{F\lambda})^{(2),l}$$
$$(\boldsymbol{F\lambda})^{(2),k} = (\boldsymbol{F\lambda})^{(2),k} - \beta^{(1),l}(\boldsymbol{F\lambda})^{(1),l} - \beta^{(2),l}(\boldsymbol{F\lambda})^{(2),l}$$

end for

-Orthogonalization of simultaneous directions-

$$(\boldsymbol{\lambda F\lambda})^{(1),k} = (\boldsymbol{F\lambda})^{(1),k^T}(\Delta\boldsymbol{\lambda})^{(1),k}$$
$$\alpha^{(2),k} = (\boldsymbol{F\lambda})^{(1),k^T}(\Delta\boldsymbol{\lambda})^{(2),k}/(\boldsymbol{\lambda F\lambda})^{(1),k}$$
$$(\Delta\boldsymbol{\lambda})^{(2),k} = (\Delta\boldsymbol{\lambda})^{(2),k} - \alpha^{(2),k}(\Delta\boldsymbol{\lambda})^{(1),k}$$
$$(\boldsymbol{F\lambda})^{(2),k} = (\boldsymbol{F\lambda})^{(2),k} - \alpha^{(2),k}(\boldsymbol{F\lambda})^{(2),k}$$
$$(\boldsymbol{\lambda F\lambda})^{(2),k} = (\boldsymbol{F\lambda})^{(2),k^T}(\Delta\boldsymbol{\lambda})^{(2),k}$$

-Minimization-

$$\eta^{(1),k} = (\boldsymbol{F\lambda})^{(1),k^T}\boldsymbol{d}/(\boldsymbol{\lambda F\lambda})^{(1),k}$$
$$\eta^{(2),k} = (\boldsymbol{F\lambda})^{(2),k^T}\boldsymbol{d}/(\boldsymbol{\lambda F\lambda})^{(2),k}$$
$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^{(1),k}(\Delta\boldsymbol{\lambda})^{(1),k} + \eta^{(2),k}(\Delta\boldsymbol{\lambda})^{(2),k}$$
$$\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \eta^{(1),k}(\boldsymbol{F\lambda})^{(1),k} - \eta^{(2),k}(\boldsymbol{F\lambda})^{(2),k}$$
$$k = k + 1$$

end while

Remarks

- Let us note that this algorithm is in fact similar to a block Conjugate Gradient algorithm [51] where the blocks are preconditioned by independent Dirichlet problems. Therefore, one can state that block recurrent orthogonality relations exist between the directions of descent. For instance one can show that $\Delta\lambda^{(1),k}$ is orthogonal to $\Delta\lambda^{(1),l}$ for $l = 0, \dots k-2$ and to $\Delta\lambda^{(2),l}$ for $l = 0, \dots k-1$. This simplifies the orthogonalization step, although for numerical stability, we choose to apply a full orthogonalization.

- The convergence criterion as shown in Algorithm 15.1 is set on the dual error. Nevertheless, one could as well define a convergence criterion on the primal error.

This simultaneous iteration scheme has been tested on the two examples shown in Fig. 15.11, namely on the truss frame and on the plane stress unsymmetric beam problem with a heterogeneous interface (2 elements have a Young's modulus 10 times softer than the rest of the structure). In Fig. 15.11(c) and 15.11(d), we show the convergence curves respectively for the truss and for the unsymmetric beam when a simultaneous conjugate iteration is applied. For comparison we have also plotted the convergence curves for the FETI method with a Dirichlet preconditioner and with or without consistent smoothing. We observe that the convergence of the simultaneous conjugate iteration method is even better than the convergence obtained by FETI with smoothing. This result highlights that in the simultaneous conjugate iteration, a really optimal use is made of the solution of the local Dirichlet problems.

It is not sure if this new iteration scheme can be generalized to problems with more than two subdomains and with floating modes, but the results obtained for the two-subdomain case shows that these methods are worth investigating.

17.7 Primal residue evaluation

In Chapter 11, the Conjugate Gradient is presented as an iteration method for solving the dual interface problem and therefore the stopping criterion in the algorithms of Chapter 11 specifies that the relative dual residue $\|\boldsymbol{w}^k\|/\|\boldsymbol{d}\|$ should be smaller than a given tolerance ϵ. This criterion means that we want the displacement gap between subdomains to be $1/\epsilon$ times smaller than the interface gap when only the external loads $\boldsymbol{f}(s)$ and the Lagrange multipliers $\boldsymbol{\lambda}^0$ that self-equilibrate $\boldsymbol{f}^{(s)}$ are applied.

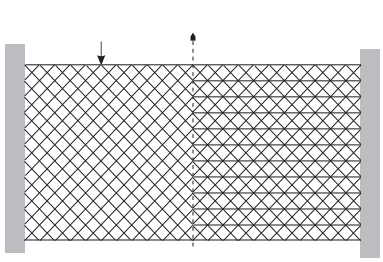
One should however recall that in the dual domain decomposition method, the interface compatibility problem was artificially introduced while the fundamental problem to be solved is the global assembled problem of the type $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$. Hence, it would be more relevant to monitor the error of the assembled problem.

At an iteration k of the FETI method, the interface stresses are given by $\boldsymbol{\lambda}^k$, the local displacements $\boldsymbol{u}^{(s)^k}$ are computed solving the local equilibrium, and the interface compatibility error is deduced from the boundary displacements. Hence, at the end of every FETI iteration, the subdomain equilibrium is exactly satisfied and the interface equilibrium is trivially satisfied by the Lagrange multipliers. Hence, no equilibrium default exist.

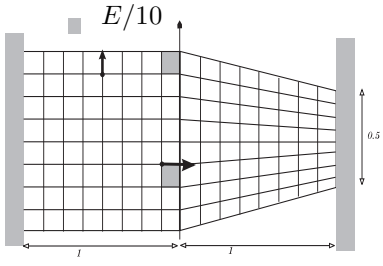
In order to define an equilibrium error, i.e. a primal residue associated with the assembled system, we must first define a compatible interface displacement. As discussed in Chapters 12 and 15, an a posteriori compatible displacement is implicitly defined during the preconditioning step when a mechanically consistent preconditioner is used. For a Dirichlet preconditioner (Section 12.2), the compatible interface displacements $\hat{\boldsymbol{u}}^{(s)^k}$ are computed by averaging the displacements $\boldsymbol{u}_b^{(s)^k}$ (Eq. (12.13) or (12.14)). The internal d.o.f. are then obtained by solving the local equilibrium when the interface displacement corrections are applied, hence by solving local Dirichlet problems. The interface loads on each subdomain boundary are then given by

$$\Delta\boldsymbol{f}_b^{(s)} = \boldsymbol{S}_{bb}^{(s)} \Delta\boldsymbol{u}_b^{(s)} \tag{17.56}$$

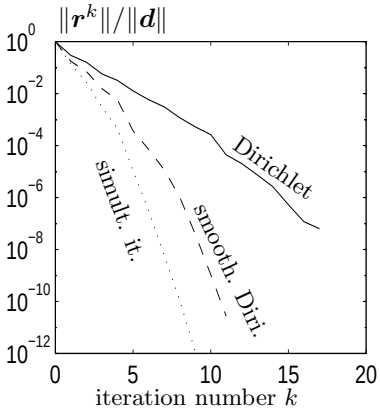
[10pt,a4paper]book bold,PhD,amssymb,rotating,psfig,subfigure



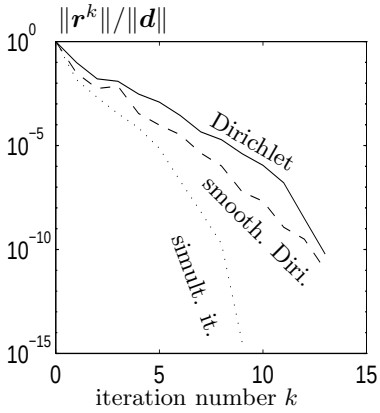
(a) Truss frame structure



(b) Heterogeneous unsymmetric beam



(c) Convergence for the truss frame



(d) Convergence for the beam

Figure 17.11: Results of the simultaneous conjugate iteration method applied to the truss frame and to the heterogeneous unsymmetric beam

In the Dirichlet preconditioner these interface force corrections are then averaged in order to define a Lagrange multiplier correction for the next iteration. However, these interface loads are not equilibrated unless the averaged solution is equal to the exact solution. The interface primal error associated with the averaged displacements is the equilibrium residue which on a boundary edge j of subdomain $\Omega^{(s)}$ is given by

$$\boldsymbol{r}_b^{(s),j} = \Delta \boldsymbol{f}_b^{(s),j} + \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j^T} \Delta \boldsymbol{f}_b^{(r),j} \tag{17.57}$$

where $\boldsymbol{c}^{(sr),j^T}$ defines the numbering correspondence for the boundary d.o.f. on edge j between $\Omega^{(s)}$ and $\Omega^{(r)}$ (Eq. 15.30). The norm of the primal residue on all interface edges is than compared to the norm of the applied force.

When a lumped preconditioner is used, the effect of the interface displacement corrections is not back-propagated into the subdomains interior. The interface reactions resulting from the enforcement of the displacement corrections are then given by (15.56) where the statically condensed interface stiffness $\boldsymbol{S}_{bb}^{(s)}$ is replaced by the lumped stiffness $\boldsymbol{K}_{bb}^{(s)}$. The corresponding primal residue on the interface edges is still given by (15.57) and since for the lumped method the internal equilibrium is no longer satisfied, we must also compute an internal primal residue which is

$$\boldsymbol{r}_i^{(s)} = \boldsymbol{K}_{ib}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{17.58}$$

If smoothing is applied for preconditioning, a set of compatible interface displacements is computed by a smoothing procedure. The interface corrections are than given by (15.2c) or, if a super-lumped smoothing procedure is used, by (15.32). The primal residue is still computed by (15.57) and, for lumped preconditioners, by (15.58).

To summarize, we can say that a primal error can be computed at every FETI iteration by considering the strongly compatible displacements computed in the preconditioning step. In fact, this compatible solution should be taken as the best solution available at iteration k, since we have seen in Part I that an averaged solution for homogeneous problems and a smoothed solution for heterogeneous problems are better solutions than the weakly compatible one (Chapters 5, 6 and 7).

Let us note the close similarity between the computation of the primal residue and the setup of a preconditioned direction. They are both computed from the interface loads deriving from enforcing the interface compatibility. Nevertheless they are different by the fact that the primal residue is the resultant force of all interface reactions whereas the preconditioned Lagrange multipliers actually represent the boundary loads. For instance, if we consider the simple two-subdomain case, and if we further assume that the constraint matrices are $\boldsymbol{b}^{(1)} = -\boldsymbol{I}$ and $\boldsymbol{b}^{(2)} = \boldsymbol{I}$, the primal residue and the preconditioned direction associated to a dual residue $\boldsymbol{w} = \boldsymbol{u}_b^{(2)} - \boldsymbol{u}^{(1)}$ are given by (Dirichlet method)

$$\Delta \boldsymbol{u}^{(1)} = \frac{1}{2}\boldsymbol{w} \qquad \Delta \boldsymbol{u}^{(2)} = -\frac{1}{2}\boldsymbol{w} \tag{17.59}$$

$$\boldsymbol{r}_b^{(1)} = \boldsymbol{r}_b^{(2)} = \Delta \boldsymbol{f}_b^{(1)} + \Delta \boldsymbol{f}_b^{(2)} = \frac{1}{2}(\boldsymbol{S}_{bb}^{(1)} - \boldsymbol{S}_{bb}^{(2)})\boldsymbol{w} \tag{17.60}$$

$$\Delta \boldsymbol{\lambda} = \frac{1}{2}\Delta \boldsymbol{f}_b^{(1)} + \frac{1}{2}(-\Delta \boldsymbol{f}_b^{(2)}) = \frac{1}{4}(\boldsymbol{S}_{bb}^{(1)} + \boldsymbol{S}_{bb}^{(2)})\boldsymbol{w} \tag{17.61}$$

As an example, let us consider again the simplified wing-box structure (Fig. 15.4). Assuming that the super-lumped Dirichlet preconditioner is used, we plot in Fig. 15.12 the variation of the norms of the primal and dual residues, and we also show the norm of the preconditioned directions.

We observe that the primal and dual residues convergence curves are parallel. Moreover, we note that the norm of the preconditioned directions is close to the norm of the primal residue.

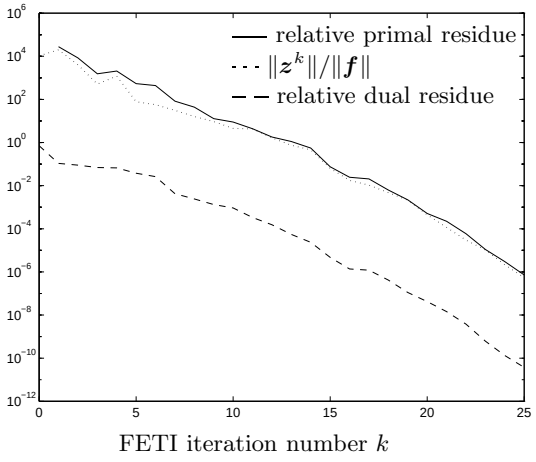


Figure 17.12: Primal and dual residue convergence

This is observed in all practical problems, and therefore the preconditioned direction can be used as a primal residue indicator. We thus use as convergence criterion in the FETI method

$$\|\boldsymbol{z}^k\| < \epsilon \|\boldsymbol{f}\| \tag{17.62}$$

Nevertheless, when the iterations have converged it is important to compute the exact primal residue in order to guarantee the quality of the solution.

17.8 Primal residue evaluation

In Chapter 11, the Conjugate Gradient is presented as an iteration method for solving the dual interface problem and therefore the stopping criterion in the algorithms of Chapter 11 specifies that the relative dual residue $\|\boldsymbol{w}^k\|/\|\boldsymbol{d}\|$ should be smaller than a given tolerance ϵ. This criterion means that we want the displacement gap between subdomains to be $1/\epsilon$ times smaller than the interface gap when only the external loads $\boldsymbol{f}(s)$ and the Lagrange multipliers $\boldsymbol{\lambda}^0$ that self-equilibrate $\boldsymbol{f}^{(s)}$ are applied.

One should however recall that in the dual domain decomposition method, the interface compatibility problem was artificially introduced while the fundamental problem to be solved is the global assembled problem of the type $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$. Hence, it would be more relevant to monitor the error of the assembled problem.

At an iteration k of the FETI method, the interface stresses are given by $\boldsymbol{\lambda}^k$, the local displacements $\boldsymbol{u}^{(s)^k}$ are computed solving the local equilibrium, and the interface compatibility error is deduced from the boundary displacements. Hence, at the end of every FETI iteration, the subdomain equilibrium is exactly satisfied and the interface equilibrium is trivially satisfied by the Lagrange multipliers. Hence, no equilibrium default exist.

In order to define an equilibrium error, i.e. a primal residue associated with the assembled system, we must first define a compatible interface displacement. As discussed in Chapters 12 and 15, an a posteriori compatible displacement is implicitly defined during the preconditioning step when a mechanically consistent preconditioner is used. For a Dirichlet preconditioner (Section 12.2), the compatible interface displacements $\hat{\boldsymbol{u}}^{(s)^k}$ are computed by averaging the displacements $\boldsymbol{u}_b^{(s)^k}$ (Eq. (12.13) or (12.14)). The internal d.o.f. are then obtained by solving the local equilibrium when the interface displacement corrections are applied, hence by solving local Dirichlet problems. The interface loads on each subdomain boundary are then given by

$$\Delta \boldsymbol{f}_b^{(s)} = \boldsymbol{S}_{bb}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{17.63}$$

In the Dirichlet preconditioner these interface force corrections are then averaged in order to define a Lagrange multiplier correction for the next iteration. However, these interface loads are not equilibrated unless the averaged solution is equal to the exact solution. The interface primal error associated with the averaged displacements is the equilibrium residue which on a boundary edge j of subdomain $\Omega^{(s)}$ is given by

$$\boldsymbol{r}_b^{(s),j} = \Delta \boldsymbol{f}_b^{(s),j} + \sum_{\substack{r:\Gamma_I^{(r)} \ni j \\ r \neq s}} \boldsymbol{c}^{(sr),j^T} \Delta \boldsymbol{f}_b^{(r),j} \tag{17.64}$$

where $\boldsymbol{c}^{(sr),j^T}$ defines the numbering correspondence for the boundary d.o.f. on edge j between $\Omega^{(s)}$ and $\Omega^{(r)}$ (Eq. 15.30). The norm of the primal residue on all interface edges is than compared to the norm of the applied force.

When a lumped preconditioner is used, the effect of the interface displacement corrections is not back-propagated into the subdomains interior. The interface reactions resulting from the enforcement of the displacement corrections are then given by (15.56) where the statically condensed interface stiffness $\boldsymbol{S}_{bb}^{(s)}$ is replaced by the lumped stiffness $\boldsymbol{K}_{bb}^{(s)}$. The corresponding primal residue on the interface edges is still given by (15.57) and since for the lumped method the internal equilibrium is no longer satisfied, we must also compute an internal primal residue which is

$$\boldsymbol{r}_i^{(s)} = \boldsymbol{K}_{ib}^{(s)} \Delta \boldsymbol{u}_b^{(s)} \tag{17.65}$$

If smoothing is applied for preconditioning, a set of compatible interface displacements is computed by a smoothing procedure. The interface corrections are than given by (15.2c) or, if a super-lumped smoothing procedure is used, by (15.32). The primal residue is still computed by (15.57) and, for lumped preconditioners, by (15.58).

To summarize, we can say that a primal error can be computed at every FETI iteration by considering the strongly compatible displacements computed in the preconditioning step. In fact, this compatible solution should be taken as the best solution available at iteration k, since we have seen in Part I that an averaged solution for homogeneous problems and a smoothed solution for heterogeneous problems are better solutions than the weakly compatible one (Chapters 5, 6 and 7).

Let us note the close similarity between the computation of the primal residue and the setup of a preconditioned direction. They are both computed from the interface loads deriving from enforcing the interface compatibility. Nevertheless they are different by the fact that the primal residue is the resultant force of all interface reactions whereas the preconditioned Lagrange multipliers actually represent the boundary loads. For instance, if we consider the simple two-subdomain case, and if we further assume that the constraint matrices are $\boldsymbol{b}^{(1)} = -\boldsymbol{I}$ and $\boldsymbol{b}^{(2)} = \boldsymbol{I}$, the primal residue and the preconditioned direction associated to a dual residue $\boldsymbol{w} = \boldsymbol{u}_b^{(2)} - \boldsymbol{u}^{(1)}$ are given by (Dirichlet method)

$$\Delta \boldsymbol{u}^{(1)} = \frac{1}{2}\boldsymbol{w} \qquad \Delta \boldsymbol{u}^{(2)} = -\frac{1}{2}\boldsymbol{w} \tag{17.66}$$

$$\boldsymbol{r}_b^{(1)} = \boldsymbol{r}_b^{(2)} = \Delta \boldsymbol{f}_b^{(1)} + \Delta \boldsymbol{f}_b^{(2)} = \frac{1}{2}(\boldsymbol{S}_{bb}^{(1)} - \boldsymbol{S}_{bb}^{(2)})\boldsymbol{w} \tag{17.67}$$

$$\Delta \boldsymbol{\lambda} = \frac{1}{2}\Delta \boldsymbol{f}_b^{(1)} + \frac{1}{2}(-\Delta \boldsymbol{f}_b^{(2)}) = \frac{1}{4}(\boldsymbol{S}_{bb}^{(1)} + \boldsymbol{S}_{bb}^{(2)})\boldsymbol{w} \tag{17.68}$$

As an example, let us consider again the simplified wing-box structure (Fig. 15.4). Assuming that the super-lumped Dirichlet preconditioner is used, we plot in Fig. 15.12 the variation of the norms of the primal and dual residues, and we also show the norm of the preconditioned directions.

We observe that the primal and dual residues convergence curves are parallel. Moreover, we note that the norm of the preconditioned directions is close to the norm of the primal residue. This is observed in all practical problems, and therefore the preconditioned direction can be used as a primal residue indicator. We thus use as convergence criterion in the FETI method

$$\|\boldsymbol{z}^k\| < \epsilon \|\boldsymbol{f}\| \tag{17.69}$$

Nevertheless, when the iterations have converged it is important to compute the exact primal residue in order to guarantee the quality of the solution.

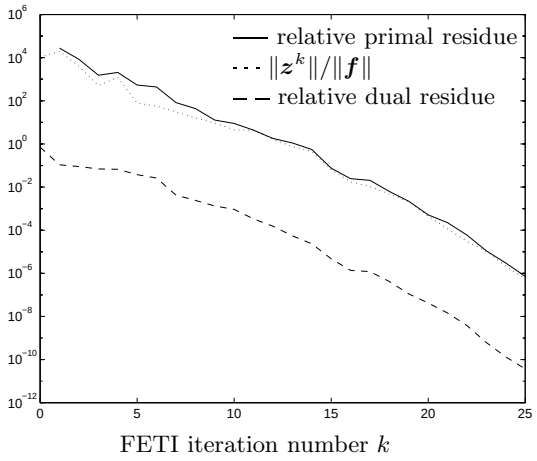


Figure 17.13: Primal and dual residue convergence

Chapter 18

The FETI method for linear dynamic analysis

In this chapter we will discuss some important issues related to the dynamic analysis of structures when the FETI method is utilized. First the computation of free vibrations is tackled. Then we discuss the problem of computing a forced harmonic response. Finally we investigate some particular points concerning the computation of transient a dynamic response.

18.1 Eigenmode computation by inverse iteration methods

When a free vibration analysis is performed on a structure, one is generally interested in the modes of lower eigenfrequencies and inverse iteration techniques such as the subspace iteration or the Lanczos method are used. As already discussed in Section 3.3 of Part I, this amounts to solving at every iteration a dual problem of the form (3.39) which is a static problem where the applied loads are given by the inertia forces of the vibration modes. Hence a FETI method can be used at every inverse iteration together with a projection-orthogonalization technique. As explained earlier in Section 11.4.1 the projection-orthogonalization strategy is very effective in this case since the successive right-hand sides arising during the inverse iteration correspond to inertia forces associated with the iterates. The iterates converge towards the lower eigenspectrum of the structure and therefore a small Krylov space of the dual interface operator is generally needed for representing the associated interface stresses.

The following issues related to the application of FETI to eigensolvers are noteworthy:

- For structures with global floating modes, the strategy presented in Section 11.5 should be applied to find the global rigid body modes, and every iterate of the inverse iterations should be orthogonalized with respect to the global null space [91].
- The subspace iteration method requires to solve a reduced eigenproblem obtained at every inverse iteration by projecting the free vibration problem onto the space of the approximate eigenmodes. When a Lanczos algorithm is used, the interaction problem also requires solving a reduced eigenproblem related to the tri-diagonal matrix of Lanczos coefficients [91]. In both cases the reduced eigenproblem to be solved is small and it can be solved by a classical Jacobi or QR method run by each processor. If however the size of the reduced eigenproblem is no longer small compared to the number of d.o.f. of the global system, one should use parallel algorithms such as the parallel Jacobi method proposed in [144].
- In practice, the Lanczos algorithm is often applied in its blocked form so to enhance its robustness for multiple eigenfrequencies and so to optimize the data flow between primary and secondary memory storage [33, 95]. In that case, the projection and orthogonalization strategy leads to matrix operations which can be optimized for high performance computing.

- When modal reduction or modal synthesis methods are used at the subdomain level in order to reduce the local problem size, the interface problem remains of the same form and no modification is required for the iteration scheme applied to the dual interface problem (see [76, 154] and Chapter 4).

Incorporating the FETI method in an inverse iteration algorithm leads to a cost effective and highly parallel eigensolver. It forms a double iterative scheme, one outer iteration loop for the inverse iterations on the vibration modes and one inner loop pertaining to the interface problem. The outer inverse iteration loop takes place in the Krylov subspace of the global displacements whereas the inner loop iterates onto the Krylov subspace of the interface Lagrange multipliers. The FETI method has been implemented for the Lanczos method in the general Finite Element package SAMCEF [89].

18.2 Harmonic response computation

In a harmonic response analysis of a system, one searches for the amplitude of the forced dynamic response corresponding to a harmonic excitation $\boldsymbol{f}$ with an excitation frequency ω_{ex} lying in a given frequency range. The harmonic equations are found by substituting $\boldsymbol{u}^{(s)} = \boldsymbol{q}^{(s)}\mathrm{e}^{i\omega_{ex}t}$ and $\boldsymbol{f}^{(s)} = \boldsymbol{p}^{(s)}\mathrm{e}^{i\omega_{ex}t}$ in Eq. (3.23). For a system without damping we find

$$\boldsymbol{K}^{(s)}\boldsymbol{q}^{(s)} - \omega_{ex}^2\boldsymbol{M}\boldsymbol{q}^{(s)} = \boldsymbol{p}^{(s)} \tag{18.1}$$

The excitation frequency ω_{ex} is varied so as to sweep the frequency range for which the harmonic response must be computed. This yields a series of static-like systems which can be solved by the FETI method exactly in the same manner as a static problem except that the harmonic operator $(\boldsymbol{K}^{(s)} - \omega_{ex}^2\boldsymbol{M}^{(s)})$ is always regular due to the inertia term if however the excitation frequency is not coincident with an eigenfrequency of the subdomain $\Omega^{(s)}$.

When the excitation frequency ω_{ex} corresponds to the lower eigenspectrum of the assembled system, the harmonic problem can be solved by a mode superposition method by considering as general expression for the assembled displacement vector

$$\boldsymbol{q} = \sum_{i=1}^{N} \eta_i \boldsymbol{u}_i \tag{18.2}$$

where N is the number of assembled d.o.f. and $\boldsymbol{u}_i$ are the eigenmodes of the assembled system satisfying

$$(\boldsymbol{K} - \omega_i^2\boldsymbol{M})\boldsymbol{u}_i = \boldsymbol{0} \qquad i,j = 1,\dots N \tag{18.3}$$

$$\boldsymbol{u}_i^T\boldsymbol{M}\boldsymbol{u}_j = \delta_{ij}\mu_i \qquad \boldsymbol{u}_i^T\boldsymbol{K}\boldsymbol{u}_j = \delta_{ij}\omega_i^2\mu_i \tag{18.4}$$

ω_i are the eigenfrequencies of the assembled system and μ_i are the modal masses. These solutions are found by inverse iterations such as explained in Section 16.1. Substituting the general solution (16.2) in (16.1) and making use of the orthogonality relations (16.4), we find the eigenmode amplitudes

$$\eta_i = \frac{\boldsymbol{u}_i^T\boldsymbol{p}}{\mu_i(\omega_i^2 - \omega_{ex}^2)} \tag{18.5}$$

In practice, only a small set of eigenmodes are included in the general solution (16.2) and a mode acceleration correction is added in order to correctly represent the quasi-static part of the response [91]. Assuming that r modes are retained, the eigenmode amplitudes are given by (16.5) for $i = 1,\dots r$ while for $i = r+1,\dots N$, the modal inertia terms $\mu_i\omega_{ex}^2$ are neglected. The harmonic response thus writes

$$\boldsymbol{q} = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^T\boldsymbol{p}}{\mu_i(\omega_i^2 - \omega_{ex}^2)}\boldsymbol{u}_i + \sum_{i=r+1}^{N} \frac{\boldsymbol{u}_i^T\boldsymbol{p}}{\mu_i\omega_i^2}\boldsymbol{u}_i \tag{18.6}$$

and using the spectral expansion of the assembled stiffness matrix, we finally obtain

$$\boldsymbol{q} = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^T \boldsymbol{p}}{\mu_i(\omega_i^2 - \omega_{ex}^2)} \boldsymbol{u}_i + \boldsymbol{K}^{-1}\boldsymbol{p} - \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^T \boldsymbol{p}}{\mu_i \omega_i^2} \boldsymbol{u}_i \tag{18.7}$$

where we assume for simplicity's sake that the assembled stiffness matrix $\boldsymbol{K}$ is regular. Thus computing an harmonic response with a mode acceleration method requires on one hand performing dot products between the eigenmodes and the excitation amplitudes and on the other hand solving a static problem. Since the eigenmodes $\boldsymbol{u}_i$ are computed by an inverse iteration with a FETI method (see Section 16.1), the static problem $\boldsymbol{K}\boldsymbol{x} = \boldsymbol{p}$ is easily solved by a projection-orthogonalization method using the conjugate directions generated during the inverse iterations. Performing a harmonic analysis with the FETI method thus leads to an efficient and highly parallel solution scheme.

18.3 Transient dynamic response analysis

The transient dynamic response of a system is computed by solving the dynamic system expressed by (3.23). A first solution method consists in expressing the solution as a superposition of eigenmodes such as (16.2) where the modal amplitudes η_i are now functions of time. The problem is then projected onto the subspace of the eigenmodes and the following decoupled normal equations are obtained

$$\ddot{\eta}_i(t) + \omega_i^2 \eta_i(t) = \frac{\boldsymbol{u}_i^T \boldsymbol{f}(t)}{\mu_i} \qquad i = 1, \dots r \tag{18.8}$$

and a mode acceleration correction can be found by solving a static problem exactly as explained in Section 16.2. We thus conclude that using the FETI method to compute the eigenmodes by a mode superposition or a mode acceleration method yields a fully parallel scheme.

When the response has a high eigenspectrum content, namely when the external force $\boldsymbol{f}(t)$ excites a broad band of eigensolutions, and/or when non-diagonal damping is to be included in the system [91], it is preferable to use direct integration methods to advance the solution step by step. As explained in Section 3.4 time-stepping schemes can be implicit or explicit.

When an explicit time-stepping scheme is used, the solution is advanced without solving any system (section 3.4.1). Note that if the explicit time-integration is solved by expressing a dual interface problem, we obtain the system of equations (3.49) which is not trivially diagonal. Nevertheless, since the local operators are diagonal mass matrices, applying the cheap super-lumped smoothing technique together with a lumped preconditioner leads to building an exact preconditioner and thus the problem (3.49) is solved in one iteration.

Explicit time-integration only implies matrix multiplications and it is therefore naturally parallel. Parallel explicit time-integration schemes have been proposed by several authors since they are fairly straightforward and because their application field (e.g. impact analysis) usually requires massive computational resources [83, 92, 131, 164]. Nevertheless, most classical dynamic problems do not require using very small time-steps as in impact dynamics and implicit time-integration methods are then more appropriate.

In the remaining of this section we will discuss same important issues related to performing implicit time-stepping when using the FETI method.

18.3.1 Energy conservation versus interface compatibility

As in Section 3.4.2 we consider the Newmark time-integration formula (3.50) so that at every time-step the system of equation (3.51) must be solved. The FETI method then consists in solving the corresponding dual interface problem (3.54):

$$\overline{\boldsymbol{F}_I}\boldsymbol{\lambda} = \overline{\boldsymbol{d}} \tag{18.9}$$

where

$$\overline{\boldsymbol{F}_I} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{D}^{(s)^{-1}} \boldsymbol{B}^{(s)^T} \tag{18.10}$$

$$\overline{\boldsymbol{d}} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{D}^{(s)^{-1}} \left(\boldsymbol{f}^{(s)_{n+1}} - \boldsymbol{C}^{(s)} \dot{\boldsymbol{u}}^{*(s)_{n+1}} - \boldsymbol{K}^{(s)} \boldsymbol{u}^{*(s)_{n+1}} \right) \tag{18.11}$$

$$\boldsymbol{D}^{(s)} = \boldsymbol{M}^{(s)} + \frac{h}{2} \boldsymbol{C}^{(s)} + \frac{h^2}{4} \boldsymbol{K}^{(s)} \tag{18.12}$$

and $\dot{\boldsymbol{u}}^{*(s)_{n+1}}$ and $\boldsymbol{u}^{*(s)_{n+1}}$ are respectively the velocity and displacement predictors defined by (3.53).

We see that advancing the solution requires solving linear systems where the local operators are the subdomain time-stepping matrices $\boldsymbol{D}^{(s)}$. The FETI iterations are stopped when the primal residue is smaller then a given tolerance and therefore, if the strongly compatible but weakly equilibrated solutions $\hat{\boldsymbol{u}}^{(s)^{n+1}}$ are chosen as displacements for the time step t_{n+1}, the primal residue introduces a spurious energy in the system and the energy conservation property of the Newmark integration scheme is no longer valid (see discussion in Section 6.5).

In order to avoid introducing spurious energy in the system due to residual forces, one can take as displacement solution at every time-step the weakly compatible solution when no averaging or smoothing is applied. This solution satisfies the exact equilibrium equations and therefore no spurious energy is introduced in the system. Nevertheless, in this case, one accepts having a small interface compatibility error at the end of every time-step. This compatibility error corresponds to a gap in the interface accelerations and therefore, even if the dual error at every time-step is small, those errors are accumulated in the velocity and displacement solutions according to the Newmark formula (3.50). Hence, after several time steps, the interface compatibility error in terms of velocities and displacements may become unacceptably high.

In practice, one should use the strongly compatible solution at every time step and accept some spurious energy in the system. In general, since the tolerance on the primal error is small for the FETI iterations, the residual loads and the associated spurious energy do not affect the solution in a significant manner.

18.3.2 On the usefulness of projection-orthogonalization in implicit time-stepping

At every time-step, the interface problem (16.9) is solved by the preconditioned Conjugate Gradient method where the successive conjugate directions can be used in a projection-orthogonalization method (Section 11.4). Nevertheless, as explained in Section 6.4, the time steps used in practice are such that the operators $\boldsymbol{D}^{(s)}$ are dominated by the mass matrix and in general mass matrices are diagonally dominant. Therefore, in the implicit dynamic equations the interface d.o.f. are only weakly coupled. The dual interface operator in (3.54) is then close to diagonal and its eigenvalues are related to the interface assembled masses. As a consequence, no dominant directions exist for the interface Lagrange multiplier solutions and the Krylov subspace relative to $\overline{\boldsymbol{F}}_I$ can correctly represent any solution only if it is nearly complete, i.e. when the number of directions stored for projection and orthogonalization is nearly equal to the number of effective Lagrange multipliers.

For this reason, the projection-orthogonalization strategy might be expensive and it is not yet clear if it is then more efficient to simply perform an iteration anew for every time-step and not apply a projection-orthogonalization strategy. Indeed, since the interface operator $\overline{\boldsymbol{F}}_I$ is nearly diagonal, convergence is rapid for the Conjugate Gradient and it might be more efficient not to use the previously computed directions.

18.3.3 Building an auxiliary coarse grid problem

The main particularity of solving the dynamic system relative to an implicit time step is that the local operators $\boldsymbol{D}^{(s)}$ are regular due to the inertia term. Hence, no natural floating mode coarse grid exist in this case. This might be detrimental for the numerical scalability of the algorithm since the absence of a coarse grid means that the number of FETI iterations will grow when the number of subdomains is increased.

In [56, 73, 74] it has been proposed to reintroduce a coarse grid problem by enforcing that at every FETI iteration the dual residue $\boldsymbol{r}^k = \overline{\boldsymbol{d}} - \overline{\boldsymbol{F}}_I \boldsymbol{\lambda}^k$ be orthogonal to the boundary restriction $\boldsymbol{G}_I$ of the local floating modes, namely the null space of the local stiffness matrices $\boldsymbol{K}^{(s)}$. In this way, as for static problems, the iteration space is the subspace orthogonal to $\boldsymbol{G}_I = [\boldsymbol{B}^{(s)}\boldsymbol{R}^{(s)}]$. Since for the transient dynamic problem, the self-equilibrium condition $\boldsymbol{G}_I^T \boldsymbol{\lambda} = \boldsymbol{e}$ no longer exists, the condition $\boldsymbol{G}_I^T \boldsymbol{w}^k = \boldsymbol{0}$ cannot be enforced by a simple projection method as in a PCPG scheme. One must therefore use the technique presented in Section 11.6 and consisting in building an auxiliary coarse grid problem. In the coarse grid problem expression (11.62) we set $\boldsymbol{P} = \boldsymbol{0}$ and $\boldsymbol{C} = \boldsymbol{G}_I$. We then obtain the auxiliary coarse grid problem

$$\Delta\boldsymbol{\lambda}^k = \boldsymbol{p}^k + \boldsymbol{G}_I \gamma^k \tag{18.13a}$$

$$\boldsymbol{G}_I^T \overline{\boldsymbol{F}}_I \boldsymbol{G}_I \boldsymbol{\gamma}^k = -\boldsymbol{G}_I^T \overline{\boldsymbol{F}}_I \boldsymbol{p}^k \tag{18.13b}$$

Combining (16.13a) and (16.13b) we see that the coarse grid problem is equivalent to define a projector $\overline{\boldsymbol{P}}$ such that

$$\Delta\boldsymbol{\lambda}^k = \overline{\boldsymbol{P}}\boldsymbol{p}^k \tag{18.14a}$$

$$\overline{\boldsymbol{P}} = \boldsymbol{I} - \boldsymbol{G}_I \left(\boldsymbol{G}_I^T \overline{\boldsymbol{F}}_I \boldsymbol{G}_I \right)^{-1} \boldsymbol{G}_I^T \overline{\boldsymbol{F}}_I \tag{18.14b}$$

Two basic objections can be raised about this procedure. First to obtain the local rigid body modes an extra factorization of the local stiffness matrices is required which implies a computational overcost. Secondly, the relevance of using the null space of $\boldsymbol{K}^{(s)}$ to form an auxiliary coarse grid is questionable since in practice the time step is small and as explained earlier the local operators are dominated by the mass matrices. Hence the local rigid body modes may not always be appropriate to represent the high end spectrum of the dynamic dual interface problem.

In order to overcome the extra-cost of factorizing the local stiffness matrices, it has been proposed in [73] to use as local modes for projecting the FETI iterations the rigid body modes obtained by the cheap geometric procedure described in [61] and the efficiency of this procedure has been demonstrated on practical examples in [73].

The second objection remains though, namely that the local rigid body modes may not define the best subspace with respect to which the FETI iterations should be orthogonalized. Indeed, from the definition (16.10) of $\overline{\boldsymbol{F}}_i$ it appears that the coarse grid should be formed by the boundary restrictions of the lower eigensolutions of the local time-stepping matrices $\boldsymbol{D}^{(s)}$. Noting then the the lower eigensolutions can be obtained by performing local inverse iterations on $\boldsymbol{D}^{(s)}$, and recalling that the factored form of the local operators $\boldsymbol{D}^{(s)}$ is readily available from the interface dual problem, we propose to perform a small number of inverse iterations for computing a rough approximate of the lower local eigensolutions $\boldsymbol{U} = [\boldsymbol{u}_n^{(s)}]$, $n = 1, \ldots n_u^{(s)}$. The lower eigensolutions are obtained by an inverse iteration that basically implies

$$\boldsymbol{D}^{(s)} \boldsymbol{u}_n^{i+1} = \boldsymbol{u}_n^i \qquad\qquad i = 0, \ldots N_{it} \tag{18.15}$$

This procedure is fully parallel and fairly cheap if the number $n_u^{(s)}$ of directions per subdomain is small and if the number of local inverse iteration N_{it} is also small.

To illustrate the effect of an auxiliary coarse grid for computing a transient dynamic response, we consider the plane stress problem of a square domain excited by a load with a step time function (Fig. 16.1). The structure is discretized by 80×80 elements with four nodes and we analyze 3

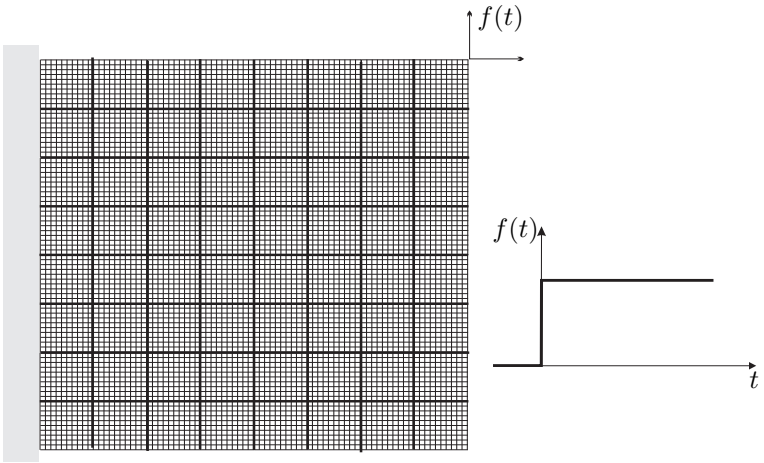


Figure 18.1: Dynamic plane stress problem

different domain decompositions, namely 2×2, 4×4 and 8×8. In Table 16.1 we give the number of FETI iterations for the first three time steps with a lumped preconditioner and when an auxiliary coarse grid is built either on the exact rigid-body modes or on local modes computed by inverse iteration. In the latter case, we compute 3 modes per subdomain by applying 5 inverse iterations per mode. The time step is taken as $h = T_1/20$ where T_1 is the period of the first eigenfrequency of the system.

decomp.	No aux. coarse			Aux. coarse $\boldsymbol{C} = [\boldsymbol{B}^{(s)}\boldsymbol{R}^{(s)}]$			Aux. coarse $\boldsymbol{C} = [\boldsymbol{B}^{(s)}\boldsymbol{U}^{(s)}]$ $n_u^{(s)} = 3, N_{it} = 5$		
	t_1	t_2	t_3	t_1	t_2	t_3	t_1	t_2	t_3
2×2	8	6	6	8	5	6	7	5	5
4×4	17	15	14	10	8	8	8	8	8
8×8	31	28	27	13	8	8	7	7	7

Table 18.1: Number of iterations for solving the 3 first time steps of a transient response analysis by the FETI method (lumped preconditioner, projection-orthogonalization and auxiliary coarse grid)

First we note that the number of FETI iterations is small compared to an equivalent static problem and that the lumped preconditioner is effective for this dynamic response analysis. This is a consequence of the fact that for practical time steps, the dual dynamic interface operator $\overline{\boldsymbol{F}}_I$ is diagonally dominant. The results in Table 16.1 show that when no auxiliary grid problem is used the number of iterations grows with the number of subdomains, meaning that the FETI method is not numerically scalable. If an auxiliary coarse grid is introduced, the number of iterations remains nearly unchanged when the number of subdomains is increased. When the coarse grid is built by using local modes computed by inverse iteration, the efficiency of the coarse grid appears to be slightly more efficient: the strategy consisting in building an auxiliary coarse grid based on approximate local modes appears to be effective.

Chapter 19

Dynamic response computation

Chapter 20

New opportunities in non-linear structural analysis

It is not our intention to present a full discussion on distributed computation for non-linear systems. Nevertheless it is important that we give some indications on how the FETI method applies to non-linear systems and what possible developments can be expected.

20.1 FETI method for non-linear systems

Non-linear structural systems are generally solved by Newton-Raphson iterations. Let us assume that the non-linear problem can be put in the general form[1]

$$\boldsymbol{g}(\boldsymbol{u}) = \boldsymbol{0} \tag{20.1}$$

where $\boldsymbol{g}$ is a non-linear function of the d.o.f. of the assembled system. The Newton-Raphson strategy then consists in linearizing the system and solving at iteration n for the displacement corrections

$$\boldsymbol{K}^t \Delta \boldsymbol{u}^n = -\boldsymbol{g}(\boldsymbol{u}^n) \tag{20.2a}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta \boldsymbol{u}^n \tag{20.2b}$$

The operator $\boldsymbol{K}^t$ is the tangent operator which for the genuine Newton-Raphson method is defined as the Jacobian matrix of the residue $\boldsymbol{g}$ computed at $\boldsymbol{u}^n$:

$$\boldsymbol{K}^t = \left[\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right]_{\boldsymbol{u}^n} \tag{20.3}$$

This method is expensive since it requires to evaluate a new stiffness matrix at every Newton-Raphson iteration and therefore it is sometimes modified by using the same tangent matrix for every iteration:

$$\boldsymbol{K}^t = \left[\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right]_{\boldsymbol{u}^0} \tag{20.4}$$

This strategy is known as the *modified Newton-Raphson* method. Note that only the exact Newton-Raphson scheme is guaranteed to have a quadratic convergence, but for smoothly non-linear systems the modified scheme is more cost effective since in that case, the convergence is only slightly affected by the fact that the tangent matrix is not exact. Obviously mixed strategies can be defined where the tangent matrix is updated according to special heuristics.

[1] Our discussion is equally valid for static problems and for dynamic problems if we assume that a time-discretization scheme has been applied

Another method consists in estimating the tangent matrix at $\boldsymbol{u}^n$ by updating the initial tangent matrix according to the tangent directions estimated by considering the successively computed residues $\boldsymbol{g}(\boldsymbol{u}^n)$ [18, 119]. This method is known as the *quasi-Newton* method. It has a better convergence than the modified Newton method, and it avoids explicit computation of new Jacobian matrices as required for the exact Newton-Raphson scheme.

From our discussion here above it is clear that solving non-linear problems comes down to solving successive linear systems. Applying a domain decomposition method to the linearized equation (17.2) yields

$$\begin{bmatrix} \boldsymbol{K}_d^t & \boldsymbol{B}^{(s)^T} \\ \boldsymbol{B}^{(s)} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_d^n \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_d(\boldsymbol{u}^n) \\ \boldsymbol{0} \end{bmatrix} \tag{20.5}$$

Hence, the FETI method can be applied as domain decomposition method exactly as for linear systems and the interface dual operator $\boldsymbol{F}_I^t$ is computed by assembling the local tangent flexibilities $\boldsymbol{K}^{(s)t^{-1}}$ or $\boldsymbol{K}^{(s)t^{+}}$. In particular when a modified Newton scheme is applied, the tangent operator remains constant and a projection-orthogonalization strategy can be used for solving the successive linearized problems. If however an exact Newton-Raphson scheme is chosen, the successive linearized problems have different tangent operators and the projection-orthogonalization is no longer valid. Nevertheless, in that case the projection-orthogonalization method can be transformed into a new preconditioning strategy [147, 158] as will be shortly explained now.

First let us re-consider the orthogonalization step in the projection-orthogonalization strategy. Let us assume for now that the operators $\boldsymbol{K}^{(s)t}$ remain unchanged and that from previous Newton-Raphson iterations we have the orthonormalized conjugate directions stored in $\boldsymbol{P}_l$ and $\boldsymbol{F}_l = \boldsymbol{F}_I^t \boldsymbol{P}_l$. The preconditioning and the orthogonalization steps in Algorithm 11.3 can be combined so that the direction of descent is formed by:

$$\boldsymbol{p}^{k+l} = \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k - \boldsymbol{P}_l (\boldsymbol{F}_l^T \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k) \tag{20.6}$$

showing that the orthogonalization step can be considered as defining an equivalent preconditioner $\tilde{\tilde{\boldsymbol{F}}}_I^{-1} = (\boldsymbol{I} - \boldsymbol{P}_l \boldsymbol{F}_l^T) \tilde{\boldsymbol{F}}_I^{-1}$ which is such that

$$\boldsymbol{P}_l^T \boldsymbol{F}_I^t \tilde{\tilde{\boldsymbol{F}}}_I^{-1} = \boldsymbol{F}_l^T \tilde{\tilde{\boldsymbol{F}}}_I^{-1} = \boldsymbol{0} \tag{20.7}$$

Eq. (17.7) indicates that the preconditioner with orthogonalization defines an exact inverse for $\boldsymbol{F}_I^t$ in the subspace of $\boldsymbol{P}_l$. Making use of (17.6) and (17.7), the residues recursively satisfy

$$\boldsymbol{P}_l^T \boldsymbol{w}^{k+1} = \boldsymbol{P}_l^T (\boldsymbol{w}^k + \eta^k \boldsymbol{F}_I^t \boldsymbol{p}^{k+l}) = \boldsymbol{0} \tag{20.8}$$

This equation shows that thanks to the orthogonalization with respect to previous directions, the residue is zero in the subspace of $\boldsymbol{P}_l$.

When the tangent operator changes from one Newton-Raphson iteration to another, the idea is then to use the conjugate directions computed with respect to previous tangent operators in order to define a good approximation of the inverse of the current tangent operator. Therefore assuming now that $\boldsymbol{P}_l$ and $\boldsymbol{F}_l$ store the normalized directions of the previous Newton-Raphson iterations, we write

$$\boldsymbol{p}^k = \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k - \boldsymbol{P}_l \boldsymbol{\nu}^k \tag{20.9}$$

where the coefficients $\boldsymbol{\nu}^k$ are such that $\boldsymbol{p}^k$ is the best approximation to the exact correction, namely

$$\boldsymbol{P}_l^T \boldsymbol{w}^{k+1} \simeq \boldsymbol{P}_l^T \boldsymbol{w}^k - \boldsymbol{F}_l^T \boldsymbol{p}^k \tag{20.10}$$

$$= \boldsymbol{0} \tag{20.11}$$

and thus we find

$$\boldsymbol{\nu}^k = -\boldsymbol{P}_l^T \boldsymbol{w}^k + \boldsymbol{F}_l^T \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k \tag{20.12}$$

Finally the preconditioning step (17.9) writes

$$\boldsymbol{p}^k = (\boldsymbol{I} - \boldsymbol{P}_l \boldsymbol{F}_l^T) \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k + \boldsymbol{P}_l \boldsymbol{P}_l^T \boldsymbol{w}^k \tag{20.13}$$

Note that if the tangent operators remain unchanged we have $\boldsymbol{P}_l^T \boldsymbol{w}^k = \boldsymbol{0}$ and we find again the expression (17.6). The method is summarized in Algorithm 17.1.

Algorithm 20.1 PCPG method for multiple right-hand sides and changing local operators

$\boldsymbol{P}_l, \quad \boldsymbol{F}_l = \boldsymbol{F}_{In-1}^t \boldsymbol{P}_l \ : \ \boldsymbol{F}_l^T \boldsymbol{P}_l = \boldsymbol{I}$
$k = 0, \boldsymbol{\lambda}^{00} = \boldsymbol{Q}\boldsymbol{G}_I \left(\boldsymbol{G}_I^T \boldsymbol{Q} \boldsymbol{G}_I \right)^{-1} \boldsymbol{e}_n$
$\boldsymbol{r}^{00} = \boldsymbol{d} - \boldsymbol{F}_{In} \lambda^0, \boldsymbol{w}^{00} = \boldsymbol{P}^T \boldsymbol{r}^{00}$
$\boldsymbol{\eta}^0 = \boldsymbol{P}_l^T \boldsymbol{r}^0$
$\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}^{00} + \boldsymbol{P}_l \boldsymbol{\eta}^0$
$\boldsymbol{r}^0 = \boldsymbol{r}^{00} - \boldsymbol{F}_{In} \boldsymbol{\lambda}^0, \boldsymbol{w}^0 = \boldsymbol{P}^T \boldsymbol{r}^0$
while $\|\boldsymbol{w}^k\| > \epsilon \|\boldsymbol{w}^{00}\|$ **do**
  $\boldsymbol{p}^k = (\boldsymbol{P} - \boldsymbol{P}_l \boldsymbol{F}_l^T) \tilde{\boldsymbol{F}}_I^{-1} \boldsymbol{w}^k + \boldsymbol{P}_l \boldsymbol{P}_l^T \boldsymbol{w}^k$
  -orthogonalization of directions-
    for $i = 0, \ldots k-1$ **do**
      $\beta_i^k = \frac{\boldsymbol{p}^{iT} \boldsymbol{F}_{In} \boldsymbol{p}^k}{\boldsymbol{p}^{iT} \boldsymbol{F}_{In} \boldsymbol{p}^i}, \ \boldsymbol{p}^k := \boldsymbol{p}^k - \beta_i^k \boldsymbol{p}^i$
    end for

  $\eta^k = \frac{\boldsymbol{p}^{kT} \boldsymbol{w}^k}{\boldsymbol{p}^{kT} \boldsymbol{F}_{In} \boldsymbol{p}^k}$
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k \boldsymbol{p}^k$
  $\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \eta^k \boldsymbol{F}_{In} \boldsymbol{p}^k$
  $\boldsymbol{w}^{k+1} = \boldsymbol{P}^T \boldsymbol{r}^{k+1}$
  $k = k + 1$
end while

From Algorithm 17.1 the following observations are noteworthy:

- The starting procedure to find $\boldsymbol{\lambda}^0$ is nearly equivalent to the projection step in the projection-orthogonalization strategy except that the initial residue $\boldsymbol{r}^0$ must be explicitly computed and cannot be deduced from $\boldsymbol{F}_l$.

- The same strategy for using previously computed directions in the preconditioner can be applied to any Conjugate Gradient method outside the context of FETI. For instance it can be adapted to iterations algorithms applied to the global system (17.2).

- This algorithm can also be applied when a perturbed system is re-analyzed. In that case the conjugate directions of the non-perturbed system are used to precondition the perturbed problem. The efficiency of this approach has been shown for computing eigenvalues of large perturbed systems [40]. It would also be interesting to investigate the application of this preconditioning method in the context of mesh refinement: conjugate directions of the previous coarser grids could be mapped onto the refined mesh and exploited in the preconditioning step.

- The efficiency of Algorithm 17.1 depends on how close the successive Newton-Raphson operators are. In [158], the high efficiency of the scheme was shown on structural problems with smooth material non-linearities. Another important case when Algorithm 17.1 should be well suited is when non-linearities are localized in certain subdomains. Unfortunately, if the non-linearities are severe, this scheme may not significantly improve the FETI convergence.

The method described here above and proposed in [147, 158] illustrates the high versatility of iteration methods where any information on the solution can be exploited in the preconditioner. Iterative solution schemes thus offer very promising perspectives for solving non-linear systems if compared to direct elimination methods. In particular, some research effort should be put into investigating the specific use of FETI in conjunction with quasi-Newton iterations.

Remarks

- Yet another interesting aspect of using iteration methods in non-linear computation is that the tolerance on the iterative solution of the linearized systems can be adapted to the accuracy of the Newton-Raphson iteration. Indeed, at the beginning of the Newton-Raphson iterations, the residue $\boldsymbol{g}(\boldsymbol{u}^n)$ is generally not small and it is therefore not necessary to compute the solution of the tangent system with a high level of accuracy. As the Newton-Raphson residue $\boldsymbol{g}(\boldsymbol{u}^n)$ decreases, the tolerance ϵ on the solution of the linearized system should then be reduced. All the difficulty of this strategy resides in being able to predict which accuracy is required for the successive linear systems so not to jeopardize the convergence of Newton-Raphson iterations.

- The overall method outlined here for solving iteratively non-linear systems consists in an outer Newton-Raphson iteration and an inner Conjugate Gradient loop. Another approach consists in merging these loops, namely in iterating with a Conjugate Gradient-like scheme on the non-linear system. This amounts to minimizing a non-quadratic function and therefore the kernel operator changes at every iteration. The drawback of this method is thus that unlike the classical Conjugate Gradient method, the iterations are not guaranteed to converge in a finite number of iterations [167].

20.2 Non-linear systems with non-linear constraints

In the previous section we have shown how the FETI method can be applied as a domain decomposition method for non-linear systems. The decomposed system (17.5) is totally equivalent to the hybrid system for a linear problem and in particular the constraint matrices $\boldsymbol{B}^{(s)}$ define simple linear compatibility constraints on the interfaces. We now consider the specific problem when non-linear constraints are defined between d.o.f. Non-linear constraints between d.o.f. of a structure originate mainly from two different sources. In contact problems, the contact constraints are non-linear due to the fact that they are active only when contact actually occurs and moreover some contact formulations may imply non-linear relations for the friction law for instance. Another case when non-linear constraints are present in the structure happens to be in multibody analysis where the bodies constituting the structure are linked together by joints which imply non-linear constraining of linking d.o.f. [28].

When non-linear constraints are present in a system, the system of equations can be put in the general form

$$\boldsymbol{g}(\boldsymbol{u}) = \boldsymbol{0} \tag{20.14a}$$

$$\boldsymbol{\Phi}(\boldsymbol{u}) = \boldsymbol{0} \tag{20.14b}$$

where $\boldsymbol{\Phi}(\boldsymbol{u})$ is a set of non-linear constraints acting on the d.o.f. $\boldsymbol{u}$. The classical solution scheme consists in applying Newton-Raphson iterations and in adding a penalization term associated with the non-linear constraints. The variational principle considered is thus the augmented Lagrangian method (Section 2.3.3) and the linearized system to be solved at every Newton-Raphson iteration

can be written in the form[2] [28]

$$\begin{bmatrix} (\boldsymbol{K}^t + p\boldsymbol{C}^T\boldsymbol{C}) & \boldsymbol{C}^T \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}^n \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g}(\boldsymbol{u}^n) \\ -\boldsymbol{\Phi}(\boldsymbol{u}^n) \end{bmatrix} \tag{20.15}$$

where $\boldsymbol{\mu}$ are the Lagrange multipliers corresponding to the non-linear constraints and where $\boldsymbol{C}$ is the linearized constraint matrix acting on the d.o.f. $\boldsymbol{u}$

$$\boldsymbol{C} = \left[\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{u}}\right]_{\boldsymbol{u}^n} \tag{20.16}$$

The effect of the penalization term is two-fold. First, as for linear constraints, it improves the definite positiveness of the operator (Section 2.3.3) and one can define an effective Usawa iteration scheme such as (2.62). But more importantly, it ensures that the non-linear constraints are not too strongly violated at every Newton-Raphson iteration so that the non-linear iteration is not to much affected by the non-compatibility of the iterates. Indeed, substituting the second set of equations of (17.15) into the first set of equation of the same relation, we can write

$$\boldsymbol{K}^t\Delta\boldsymbol{u}^n + \boldsymbol{C}^T\boldsymbol{\mu} = -\boldsymbol{g}(\boldsymbol{u}^n) + p\boldsymbol{C}^T\boldsymbol{\Phi}(\boldsymbol{u}^n) \tag{20.17}$$

showing that a force proportional to the constraint error at the previous Newton-Raphson iteration is added to the residual non-linear forces. Clearly the role of the penalization term is thus to ensure that the constraints are not too much violated during the Newton-Raphson iterations, although they cannot be exactly satisfied because of their non-linearity.[3]

A FETI method cannot be applied in a straightforward manner to the augmented linearized system (17.15) since the constraints $\boldsymbol{\Phi}$ not always define a well decomposed system and further more, as explained in Section 2.3.3, the penalization term may introduce explicit coupling between the subdomains. In the remaining of this chapter, we present different manners to treat the augmented linearized system (17.15) for parallel computation by domain decomposition.

20.2.1 Non-linear constraints with natural domain decomposition

Let us first consider the case when the constraints $\boldsymbol{\Phi}$ are defined between independent subdomains suitable for applying parallel computation (meaning that the domains can ensure a reasonable processor load balancing and that their aspect ratio is acceptable). Such a case is schematically described in Fig. 17.1 and can be encountered for instance in contact problems or in multibody analysis. The constraints then define a natural domain decomposition and in order to obtain a classical domain decomposed system, we move the penalization term to the right-hand side of the equations as in (17.17):

$$\begin{bmatrix} \boldsymbol{K}_d^t & \boldsymbol{C}^T \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}_d^n \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g}_d(\boldsymbol{u}^n) + p\boldsymbol{C}^T\boldsymbol{\Phi}(\boldsymbol{u}^n) \\ -\boldsymbol{\Phi}(\boldsymbol{u}^n) \end{bmatrix} \tag{20.18}$$

where the subscript d refers to the block diagonal character of the tangent matrix.

Note that the form (17.18) is totally equivalent to (17.15) and therefore the Newton-Raphson iterations are not at all affected by writing the linearized system under the form (17.18). What is different is the way the linearized system is solved: it can be treated exactly as if it derived from a domain decomposition method. In that case, we can apply the FETI method as described in the previous section 17.1 and the Lagrange multipliers $\boldsymbol{\mu}$ corresponding to the non-linear constraints play the role of FETI Lagrange multipliers. It is noteworthy to observe that when the subdomains

[2] We neglect the second order derivatives of $\boldsymbol{\Phi}(\boldsymbol{u})$ for the linearized system in the Newton-Raphson iterations as usually done in practice [28]. Note that this does not affect the actual non-linear problem, but it modifies the way to converge towards the non-linear solution.

[3] All the discussions of this section remain valid if the constraints are linear. In that case the constraint matrix $\boldsymbol{C}$ is constant and the compatibility residual $\boldsymbol{\Phi}(\boldsymbol{u}^n)$ is null after every Newton-Raphson iteration. As a consequence, the penalization term in (17.15) does not affect the Newton-Raphson iterations and can be removed.

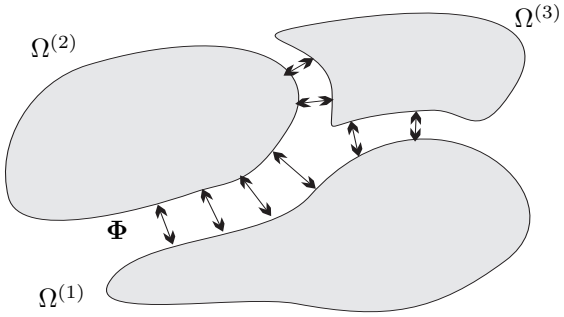


Figure 20.1: Natural domain decomposition

are associated with a locally linear model, the matrix $\boldsymbol{K}_d^t$ contains the linear operators of the subdomains and the non-linearities are completely localized in the interface constraints. Applying the FETI method is then very efficient because the dual interface operator writes

$$\boldsymbol{F}_I^t = \boldsymbol{C}(\boldsymbol{u}^n)\boldsymbol{K}_d^+\boldsymbol{C}(\boldsymbol{u}^n)^T \tag{20.19}$$

showing that it requires that the local operators be factored only once for all Newton-Raphson iterations.

20.2.2 Constructing a domain decomposition by defining additional constraints

When the non-linear constraints $\mathbf{\Phi}$ do not define a proper domain decomposition, additional constraints must be added to the system. In that case, assuming that one defines a domain decomposition by introducing additional compatibility constraints $\boldsymbol{B}\Delta\boldsymbol{u}_d^n = \boldsymbol{0}$, the augmented system of linearized equations (17.18) becomes

$$\begin{bmatrix} \boldsymbol{K}_d^t & \boldsymbol{C}^T & \boldsymbol{B}^T \\ \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}_d^n \\ \boldsymbol{\mu} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g}(\boldsymbol{u}^n) + p\boldsymbol{C}^T\mathbf{\Phi}(\boldsymbol{u}^n) \\ -\mathbf{\Phi}(\boldsymbol{u}^n) \\ \boldsymbol{0} \end{bmatrix} \tag{20.20}$$

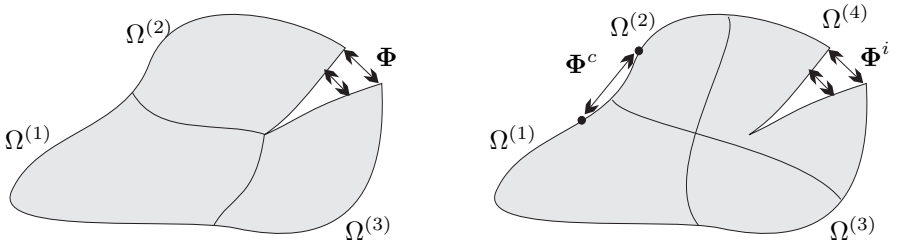
The FETI method can be applied to (17.20) exactly as for (17.18) if we now consider that the generalized compatibility constraints include the constraints $\mathbf{\Phi}$ so that the set of Lagrange multipliers $[\boldsymbol{\mu}^T\ \boldsymbol{\lambda}^T]$ play the role of FETI Lagrange multipliers.

Remark

- As a particular case, part of the interface compatibility can be defined by the constraint $\mathbf{\Phi}$ (Fig. 17.2(a)). This situation occurs for instance when analyzing fractured structures: a unilateral contact constraint must be defined on crack edges. Assuming then that $\boldsymbol{b}$ defines additional cuts to separate the domains on the d.o.f. $\boldsymbol{u}_b^{(s)}$, and calling $\boldsymbol{c}$ the non-zero submatrix of $\boldsymbol{C}$, we can formally write

$$\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{b} & \boldsymbol{0} \end{bmatrix} \tag{20.21}$$

  This relation shows that only a small number of additional compatibility constraints must be defined and therefore the iteration space for the FETI method applied to (17.20) is small. In general however it is not straightforward to impose that the interface between subdomains should include a structural interface and therefore a usual domain decomposer applied to the structure would produce a decomposition such as schematically represented in Fig 17.2(b).



(a) Minimum additional Lagrange multipliers

(b) General case

Figure 20.2: Domain decomposition by adding compatibility constraints

The strategy consisting in applying FETI to (17.20) is general and relatively easy to implement. Nevertheless, it does not take advantage of the penalization term which enhances the definite positiveness of the operator for the d.o.f. affected by the constraints $\boldsymbol{\Phi}$. In other words, the Lagrange multipliers $\boldsymbol{\mu}$ should be associated with an Usawa iteration scheme which we know to converge vary fast (see Section 2.3.3).

To this purpose, let us define the submatrix $\boldsymbol{C}^i$ corresponding to the subset of constraints $\boldsymbol{\Phi}$ relating d.o.f. pertaining to a same subdomain and $\boldsymbol{C}^c$ associated with the subset of constraints relating d.o.f. belonging to different subdomains. We have

$$\boldsymbol{C} = \left[\begin{array}{c} \boldsymbol{C}^i \\ \boldsymbol{C}^c \end{array} \right] \tag{20.22}$$

and

$$\boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{C}^{i^T}\boldsymbol{C}^i + \boldsymbol{C}^{c^T}\boldsymbol{C}^c \tag{20.23}$$

where the first term is block diagonal, i.e.

$$\boldsymbol{C}^{i^T}\boldsymbol{C}^i = \left[\begin{array}{ccc} \boldsymbol{C}^{(1)i^T}\boldsymbol{C}^{(1)i} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{C}^{(N_s)i^T}\boldsymbol{C}^{(N_s)i} \end{array} \right] \tag{20.24}$$

$\boldsymbol{C}^{(s)i}$ are the subdomain restrictions of $\boldsymbol{C}^i$. As a consequence, the augmented system (17.15) can be written as

$$\left[\begin{array}{cccc} (\boldsymbol{K}_d^t + p\boldsymbol{C}^{i^T}\boldsymbol{C}^i) & \boldsymbol{C}^{i^T} & \boldsymbol{C}^{c^T} & \boldsymbol{B}^T \\ \boldsymbol{C}^i & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}^c & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{array} \right] \left[\begin{array}{c} \Delta\boldsymbol{u}_d^n \\ \boldsymbol{\mu}^i \\ \boldsymbol{\mu}^c \\ \boldsymbol{\lambda} \end{array} \right] = \left[\begin{array}{c} -\boldsymbol{g}(\boldsymbol{u}^n) + p\boldsymbol{C}^{c^T}\boldsymbol{\Phi}^c(\boldsymbol{u}^n) \\ -\boldsymbol{\Phi}^i(\boldsymbol{u}^n) \\ -\boldsymbol{\Phi}^c(\boldsymbol{u}^n) \\ \boldsymbol{0} \end{array} \right] \tag{20.25}$$

where $\boldsymbol{\mu}^i$ and $\boldsymbol{\mu}^c$ are subsets of $\boldsymbol{\mu}$ corresponding respectively to the constraint subsets $\boldsymbol{\Phi}^i$ and $\boldsymbol{\Phi}^c$. In the latter expression, the augmented operator is block diagonal so that no inter-subdomain coupling exists due to penalization of the constraints $\boldsymbol{\Phi}$. Hence, one can apply a FETI iteration scheme by considering as FETI variables the set of multipliers $\boldsymbol{\mu}^c$ and $\boldsymbol{\lambda}$ [34], the local problems at the FETI iteration k being defined by

$$\left[\begin{array}{cc} (\boldsymbol{K}^{(s)t} + p\boldsymbol{C}^{(s)i^T}\boldsymbol{C}^{(s)i}) & \boldsymbol{C}^{(s)i^T} \\ \boldsymbol{C}^{(s)i} & \boldsymbol{0} \end{array} \right] \left[\begin{array}{c} \Delta\boldsymbol{u}^{(s)^n} \\ \boldsymbol{\mu}^i \end{array} \right] + \left[\begin{array}{cc} \boldsymbol{C}^{(s)c^T} & \boldsymbol{B}^{(s)^T} \\ \boldsymbol{0} & \boldsymbol{0} \end{array} \right] \left[\begin{array}{c} \boldsymbol{\mu}^c \\ \boldsymbol{\lambda} \end{array} \right]^k \tag{20.26}$$
$$= \left[\begin{array}{c} -\boldsymbol{g}(\boldsymbol{u}^n) + p\boldsymbol{C}^{c^T}\boldsymbol{\Phi}^c(\boldsymbol{u}^n) \\ -\boldsymbol{\Phi}^i(\boldsymbol{u}^n) \end{array} \right]$$

These local Neumann problems are solved by local Usawa iterations. The method thus proceeds as follows: in an outer loop, we solve for the Lagrange multipliers $\boldsymbol{\mu}^c$ and $\boldsymbol{\lambda}$ by a FETI iteration (Conjugate gradient on the interface problem related to the constraints $\boldsymbol{C}^c$ and $\boldsymbol{B}$), and each FETI iteration comprises a inner iteration loop consisting in local Usawa iterations.

We can also rearrange Eqs. (17.25) as follows

$$\left[\begin{array}{cc} \left[\begin{array}{ccc} (\boldsymbol{K}_d^t + p\boldsymbol{C}^{i^T}\boldsymbol{C}^i) & \boldsymbol{C}^{c^T} & \boldsymbol{B}^T \\ \boldsymbol{C}^c & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}^i & \boldsymbol{0} & \boldsymbol{0} \end{array}\right] & \left[\begin{array}{c} \boldsymbol{C}^{i^T} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{array}\right] \end{array}\right] \left[\begin{array}{c} \Delta\boldsymbol{u}_d^n \\ \boldsymbol{\mu}^c \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu}^i \end{array}\right] = \left[\begin{array}{c} -\boldsymbol{g}(\boldsymbol{u}^n) + p\boldsymbol{C}^{c^T}\boldsymbol{\Phi}^c(\boldsymbol{u}^n) \\ -\boldsymbol{\Phi}^c(\boldsymbol{u}^n) \\ \boldsymbol{0} \\ -\boldsymbol{\Phi}^i(\boldsymbol{u}^n) \end{array}\right] \tag{20.27}$$

This last form suggest that the solution can also be obtained by permuting the FETI and the Usawa iteration loops: in the outer loop one constructs Usawa iterations for the constraints $\boldsymbol{C}^i$ and at every Usawa iteration one solves by inner FETI iterations the augmented system of equations. This method was applied in [35] for multibody analysis in the particular case when only intra-subdomain constraints are defined, namely $\boldsymbol{C}^c = \boldsymbol{0}$.

To conclude this section, let us note the similarity between the FETI-Usawa or the Usawa-FETI iterations discussed here above and the two-level FETI obtained by constructing an auxiliary coarse grid problem (Section 11.6). Indeed, recognizing that Usawa iterations are nearly equivalent to Conjugate Gradient iterations for the FETI algorithm applied to an augmented system, the FETI-Usawa strategy described by (17.25) corresponds to a FETI iteration on the augmented linearized system with a second-level FETI for a coarse grid problem ensuring that the constraints $\boldsymbol{C}^i\Delta\boldsymbol{u}^n$ be strongly enforced. The Usawa-FETI iterations can be interpreted as a FETI iteration on the augmented linearized system where an auxiliary grid is defined to ensure exact compatibility between subdomains.

Closure

The work presented aims at enhancing the potentiality of substructuring methods for assembling conforming and non-conforming substructures meshes, for defining accurate and efficient reduction techniques and for designing domain decomposition methods for parallel computing in structural analysis.

In part I, different ways have been reviewed to define compatibility between subdomains and in particular we have been concerned with the assembly of non-conforming grids. We have also discussed the mathematical tools and numerical techniques for enforcing the compatibility constraints. The Lagrange multiplier method has appeared to provide the most appropriate and efficient formulation in the sense that it is based on sound variational principles that are equally applicable to conforming and non-conforming substructures and which lead in a natural way to an effective two-field hybrid formulation.

We have discussed in Chapter 4 some existing modal reduction techniques for the local subdomain systems (Craig-Bampton) and for the primal interface problem. Spectral reduction of the primal interface problem appeared to be expensive since it requires computing the lower eigenmodes of the assembled Schur complement. We have thus proposed a reduction technique for the interface problem expressed in the dual form. We showed that the modal reduction of the dual interface problem arising from the two-field hybrid formulation is indeed a highly attractive idea since the physically relevant eigenmodes to include in the spectral approximation of the interface pertain to the high end of the eigenspectrum of the dual problem and can therefore be computed by cheap non-inverse iterations. The spectral reduction of the dual operator opens new prospects for defining efficient modal reduction schemes.

We then discussed the reduction of the interface problem by introducing a small number of interface stress distribution functions. In particular, we have illustrated the relevance of such reduction methods for homogeneous and smooth problems but we also stressed the bad efficiency of those strategies for heterogeneous structures. A novel smoothing method has thus been designed for handling highly heterogeneous substructured problems. Our smoothing method is based on a Rayleigh-Ritz procedure for defining an accurate, strongly compatible solution based on the weakly compatible result obtained from the reduced dual interface problem. Combining a reduced two-field hybrid formulation with the interface smoothing technique was shown to produce a highly effective approach for solving substructured problems with an arbitrary number of subdomains and with conforming and non-conforming interfaces.

The smoothing procedure has been successfully applied to simple plane stress problems and to complex structures such as a wing-box model. However for structures with highly irregular interface stress distribution such as truss frames, reducing the interface problem by defining a

small number of simple distribution functions appeared to be unsatisfactory since the reduction scheme fails to find an accurate solution for such structures. In the case when no suitable stress distribution functions can be defined, we have proposed to iteratively approximated the interface stresses by considering the connection stresses inherent to the smoothing procedure. We thereby defined an iteration method that was found to be very effective for assembling conforming as well as non-conforming irregular substructures.

The iteration method proposed at the end of part I turns out to be the generalization of the FETI (Finite Element Tearing and Interconnecting) domain decomposition method which operates on conforming substructures in order to solve in parallel structural analysis problems. In part II, we have exposed the FETI method in the context of parallel computing.

After a short description of parallel computers and after having discussed the parallel algorithms in general (Chapter 10), the Conjugate Gradient algorithm which is the backbone of FETI iterations has been described. The most important issues specific to the application of the Conjugate Gradient algorithm to the dual interface problem have been tackled in Chapter 11. In particular we have shown how to treat a structural problem with global rigid body modes: a general Conjugate Gradient strategy has been defined for semi-definite systems and we have shown that for the FETI iterations it results in a simple and low cost modification of the iterations on the floating modes coarse grid.

We have then presented in Chapter 12 the classical preconditioners for the FETI method, namely the Dirichlet and lumped preconditioners. These basic preconditioners have been slightly modified by introducing scaling with respect to the interface multiplicities so that preconditioning is then mechanically consistent. The efficiency of the Dirichlet and lumped preconditioners have been discussed by analyzing the eigenspectrum of the preconditioned dual operator and by describing their mechanical effect. We came to the conclusion that the lumped preconditioner is well suited for simple regular problems, but the Dirichlet preconditioner should be considered when the coupling between interface degrees of freedom through the subdomain interior is significant.

In Chapter 13, the adverse effect of subdomain aspect ratio was explained while the finite element aspect ratio was found not to influence the convergence of the FETI iterations. We observed that the Dirichlet preconditioner renders the FETI iterations more robust for dealing with bad subdomain aspect ratios. The need for a Dirichlet-like preconditioner in several practical problems has led us to design an approximate Dirichlet preconditioner (Chapter 14) where the Dirichlet problems are solved only approximately in order to avoid the cost of factorizing the interior operator.

We then illustrated the poor convergence obtained when applying FETI iterations with the basic preconditioners to heterogeneous problems. In Chapter 15, a simple heterogeneous scaling is proposed for the particular case of uniform subdomains, and more preconditioners have been developed to handle general heterogeneities. Based on the mechanical interpretation of the preconditioners, we have first explained how the smoothing method exposed in Part I for post-processing weakly compatible solutions can be exploited in order to define a general framework for adapting the preconditioners to heterogeneous problems. The resulting smoothed Dirichlet preconditioner was found to be very efficient, but its high computational cost led us to define more economical alternatives. We therefore developed the smoothed and lumped preconditioner, and several non-consistently smoothed ones. In particular, we have proposed a super-lumped smoothing strategy that implies very minor computational costs and which has been shown to be very effective especially for handling shell problems. In the same chapter, a new iteration has been proposed for the case of two-subdomains to fully exploit the solution of the local Dirichlet problems solved during the preconditioning step. We have also explained how from the smoothed and averaged solutions computed within the preconditioners a primal residue can be evaluated.

In several sections of different chapters, we have studied the problem of preconditioning the floating modes coarse grid. In practice, the floating modes coarse grid is built by omitting to include the preconditioner. However, we have shown in Section 11.3.3 that a good preconditioner

can be fully effective only if the floating modes coarse grid is also preconditioned. This point was again highlighted in Section 12.5 where we have explained that including the preconditioner in the coarse grid incurs only a small cost. In Section 15.5, we have analyzed the particular case of a highly heterogeneous checkerboard layout and we have demonstrated that for this case the preconditioning of the coarse grid is primordial.

The application of the FETI method to dynamic analyses of structures has been discussed in Chapter 16 where it was found that the FETI method is well adapted for solving eigenvalue problems and for computing harmonic and transient dynamic responses. Finally, in Chapter 17, we have indicated how to treat non-linear problems with the FETI method and we have discussed different strategies for handling problems with non-linear constraints.

This work illustrates the high versatility of iterative solutions schemes in general and of the FETI method in particular. Indeed, whereas direct elimination schemes are not flexible in that they can solve systems only with a full accuracy and starting from scratch, iterative methods can be used to find approximations, and any information about the solution or about the solution space can be exploited. This versatility has been illustrated for instance in the design of heterogeneous preconditioners or for developing new strategies adapted to non-linear systems.

Finally, let us conclude by highlighting that domain decomposition methods such as FETI are becoming mature enough to replace common direct solution schemes for instance in parallel computing. Due to the mechanical aspects underlying these methods, domain decomposition should remain an inspiration source for many engineers to come.

Bibliography

[1] Y. Achdou, J-C Hontand, and O. Pironneau. A Mortar element method for fluids. In Glowinsky, Periaux, Shi, and Widlund, editors, *Domain Decomposition Methods in Sciences and Engineering*, chapter 4. John Whiley & Sons, 1995.

[2] Yves Achdou and Yu.A. Kuznetsov. Algorithms for a non conforming domain decomposition method. CMAP tech. report, 1996.

[3] Yves Achdou, Yvon Maday, and Olof Widlund. Méthode itérative de sous-structuration pour les éléments avec joints. Technical Report T. 322, série I, C. R. Acad. Sci. Paris, 1996. Analyse Numérique, p. 185-190.

[4] M. A. Ajiz and A. Jennings. A robust incomplete choleski-conjugate gradient algorithm. *International J. Numer. Methods Engineering*, 20:949–966, 1984.

[5] K. Alvin, H.M. de la Fuente, B. Haugen, and C. Felippa. Membrane triangles with corner drilling freedoms - I. the EFF element. *Fin. El. in Analysis and Design*, 12:163–187, 1992.

[6] G. Amdahl. The validity of the single processor approach to achieving large scale computing capabilities. In *AFIPS Computing Conference*, volume 30, pages 483–485, 1967.

[7] M.A. Aminpour, J.B. Ransom, and S.L. McCleary. Coupled analysis of independently modeled finite element subdomains. In *Structural Dynamics and Material Conference*, Dallas USA, April 13–15 1992. 33rd AIAA/ASME/AHS/ASC Structures.

[8] M.A. Aminpour, J.B. Ransom, and S.L. McCleary. A coupled analysis method for structures with independently modelled finite element subdomains. *International J. Numer. Methods Engineering*, 38:3695–3718, 1995.

[9] E. Anderson and al. *Lapack Users'Guide*. SIAM, 1992.

[10] Ansys powersolver. USACM-Net Digest, July 27 1995.

[11] I. Babuška and R. Narasimhan. The Babuška-Brezzi condition and the patch test: an example. *Comput. Meth. Appl. Mech. Engin.*, 140:183–199, 1997.

[12] Ivo Babuška. The finite element method with Lagrangian multipliers. *Num. Math*, 20:179–192, 1973.

[13] R. Barret, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. Templates for the solution of linear systems: Building blocks for iterative methods. from templates@cs.utk.edu.

[14] J.-L. Batoz, K.-J. Bathe, and L.-W. Ho. A study of three-node triangular plate bending elements. *International J. Numer. Methods Engineering*, 15:1771–1812, 1980.

[15] A. Beguelin, J. Dongarra, A. Geist, R. Manchek, and V. Sunderam. A users'guide to PVM parallel virtual machine. Technical report, Oak Ridge National Laboratory, 1991.

[16] T. Belytshko and E.J. Plaskacz. SIMD implementation of a non-linear transient shell program with partially structured meshes. *International J. Numer. Methods Engineering*, 33:997–1026, 1992.

[17] 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.

[18] Dimitri Bertsekas. *Constrained Optimization and Lagrange Multipliers methods*. Computer Science and Scientific Computing Series. Academic Press, 1982.

[19] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall International Editions, 1989.

[20] P. E. Bjorstad and O. B. Widlund. Iterative methods for solving elliptic problems on regions partitioned into substructures. *SIAM Journal on Numerical Analysis*, 23:1097–1120, 1986.

[21] E.F.F. Botta and F.W. Wubs. The convergence behaviour of iterative methods on severely streched grids. *International J. Numer. Methods Engineering*, 36:3333–3350, 1993.

[22] F. Bourquin. Component mode synthesis and eigenvalues of second order operators: Discretization and algorithm. *Mathematical Modelling and Numerical Analysis*, 26(3):385–423, 1992.

[23] F. Bourquin. A domain decomposition method for the eigenvalue problem in elastic multistructures. In Ciarlet, Trabucho, and Viano Eds., editors, *Asymptotic Methods for Elastic Structures*, pages 16–29. Walter de Gruyter & Co., 1995.

[24] F. Bourquin. Error analysis for modal synthesis. Technical Report R/95/09/1, LCPC, Laboratoire des Matériaux et des Structures du Génie Civil, Champ sur Marne, France, 1995.

[25] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103–134, 1986.

[26] S. Brawer. *Introduction to parallel programming*. Academic Press, 1989.

[27] A. Cardona and M. Géradin. Time integration of the equations of motion in mechanism analysis. *Comp. & Struct.*, 33:801–820, 1989.

[28] Alberto Cardona. *An Integrated Approach to Mechanism Analysis*. PhD thesis, Faculté des Sciences Appliquées, University of Liège, Belgium, 1989.

[29] S. Cescotto and R. Charlier. Frictional contact finite elements based on mixec variational principles. *International J. Numer. Methods Engineering*, 36:1681–1701, 1993.

[30] J.C. Chiou, K.C. Park, and C. Farhat. A natural partitioning scheme for parallel simulation of multibody systems. *International J. Numer. Methods Engineering*, 36:945–967, 1993.

[31] Jin-Chern Chiou. *Constraint Treatment Techniques and Parallel Algorithms for Multibody Dynamics Analysis*. PhD thesis, College of Engineering, University of Colorado, 1990.

[32] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *ASME J. Appl. Mech.*, 60:371–375, 1993.

[33] D. Coulon. Algorithme de lanczos par blocs. Technical Report VF-103, LTAS-University of Liège, Belgium, December 1993.

[34] D. Coulon. Private communication, 1996.

[35] D. Coulon, M. Géradin, and C. Farhat. Adaptation of a finite element solver for the analysis of flexible mechanisms to parallel processing systems. In Civil-Comp Press, editor, *Second International Symposium on Computational Structures Technology, Athens*. B.H.V. Topping and M. Papadrakakis, August 1994.

[36] D. Coulon, G. Piret, and M. Géradin. Parallel design of linear equations solvers for finite element systems. In *1st European Conference on numerical Methods in Engineering*. ECCOMAS, September 7–11 1992. SA–161.

[37] Lawrence Cowsar, Jan Mandel, and Mary F. Wheeler. Balancing domain decomposition for mixed problems in oil reservoir simulation. Presented at the 6th International Symposium on Domain Decomposition Methods, Como, Italy, 1992, 1992.

[38] L. Crivelli and C. Farhat. Implicit transient finite element structural computations on MIMD systems: FETI vs. direct solvers. In *Structures, Structural Dynamics and Material Conference*, La Jolla, USA, April 19–22 1993. 34rd AIAA/ASME/ASCE/AHS/ASC.

[39] Jean-Michel Cros and Françoise Léné. Méthodes de sous-structuration dynamique sur calculateur parallèles MIMD. In Hermès [104], pages 667–672.

[40] Jean-Michel Cros and Françoise Léné. Parallel iterative methods to solve large-scale eigenvalues problems in structural dynamics. In *Proceedings of the ninth conference on Domain Decomposition, Bergen*, June 1996. in press.

[41] O. Débordes and S. Quilici. Influence de l'organisation des communications et des solveurs directs et itératifs pour la parallélisation MIMD de problèmes eléments finis non-lińeaires. In Hermès [103], pages 928–944.

[42] G. Dhatt and G. Touzot. *Une présentation de la méthode des éléments finis*. Maloine, 1984.

[43] J.K Dickinson and P.A. Forsyth. Preconditioned conjugate gradient methods for three-dimensional linear elasticity. *International J. Numer. Methods Engineering*, 37:2211–2234, 1994.

[44] Q. V. Dinh and T. Fanion. Applications of dual schur complement preconditioning to problems in computational fluid dynamics and computational electromagnetics. In *Proceedings of the ninth conference on Domain Decomposition, Bergen*, June 1996. in press.

[45] J.J. Dongarra. Performance of various computers using standard linear software. Technical Report CS–89–85, Oak Ridge National Laboratory, December 1991.

[46] J.J. Dongarra, I.S. Duff, D.C. Sorensen, and H.A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, 1991.

[47] M. R. Dorr. On the discretization of interdomain coupling in elliptic boundary-value problems. In T. Chan, R. Glowinski, J. Periaux, and O. Widlund, editors, *Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 17–37. SIAM, 1989.

[48] Z. Dostal. Domain decomposition for semicoercive contact problems. In *Proceedings of the tenth conference on Domain Decomposition methods, Boulder, Colorado*, August 1997.

[49] M.C. Dracopoulos and M.A. Crisfield. Coarse/fine mesh preconditioners for the iterative solution of finite element problems. *International J. Numer. Methods Engineering*, 38:3297–3313, 1995.

[50] M.C. Dracopoulos and M.A. Crisfield. A partially sequential preconditioner for a parallel and efficient finite element solution. *Computing Systems in Engineering*, 6(6):549–561, December 1995.

[51] L.A. Drummond, I.S. Duff, and Daniel Ruiz. A parallel distributed implementation of the block conjugate gradient algorithm. Technical Report TR/PA/93/02, CERFACS, May 1993.

[52] Europort-1. Survey and results in fluid dynamics and structural mechanics. European Commission, Directorate General III (Industry), GMD-SCAI St. Augustin, Germany, 1996. Esprit III initiative.

[53] C. Farhat. Fast structural design and analysis via hybrid domain decomposition on massively parallel processors. *Computing Systems in Engineering*, 4(4–6):453–472, 1993.

[54] C. Farhat. Optimizing substructuring methods for repeated right hand sides, scalable parallel coarse solvers, and global/local analysis. *SIAM J. Sci. Stat. Comput.*, 1994.

[55] C. Farhat, Po-Shu Chen, and Paul Stern. Towards the ultimate iterative substructuring method: Combined numerical and parallel scalability, and multiple load cases. *Internat. J. Numer. Meths. Engrg.*, 37(11):1945–1975, 1994.

[56] C. Farhat, P.S. Chen, and J. Mandel. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *Internat. J. Numer. Meths. Engrg.*, 38(22):3831–3853, November 1995.

[57] C. Farhat and L. Crivelli. Large scale FE parallel nonlinear computations using a homotopy method. in Parallel Processing for Scientific Computing, 1988. Chap. 41.

[58] C. Farhat, L. Crivelli, and F.X. Roux. Extending substructures based iteratives solvers to multiple load and repeated analyses. *Comput. Meth. Appl. Mech. Engin.*, 117:195–209, 1994.

[59] C. Farhat, L. Crivelli, and F.X. Roux. A transient FETI methodology for large-scale parallel implicit computations in structural mechanics. *Internat. J. Numer. Meths. Engrg.*, 37(11):1945–1975, 1994.

[60] C. Farhat and M. Géradin. On a component mode synthesis method and its application to incompatible substructures. *Computers & Structures*, 51:459–473, 1994.

[61] C. Farhat and M. Géradin. On the computation of the null space and generalized inverse of a large matrix, and the zero energy modes of a structure. *Internat. J. Num. Meth. Engin.*, 1997. in press.

[62] C. Farhat and M. Lesoinne. Mesh partitioning algorithms for the parallel solution of partial differential equations. *Applied Numerical Mathematics*, 12:443–457, 1993.

[63] C. Farhat and M. Lesoinne. Top-domdec manual. CAS, Univerity of Colorado at Boulder, 1995.

[64] C. Farhat, N. Maman, and G. Brown. Mesh partitioning for implicit computations via iterative domain decomposition: impact and optimization of the subdomain aspect ratio. *Internat. J. Numer. Meths. Engrg.*, 38:989–1000, 1995.

[65] C. Farhat, J. Mandel, and F.X. Roux. Optimal convergence properties of the FETI domain decomposition method. In D. Keyes, Y. Saad, and D.G. Truhlar, editors, *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*. SIAM, 1994.

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

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

[68] Ch. Farhat. An introduction to parallel scientific computations. Technical report, University of Liège, Belgium, February 1991. Postgraduate studies in supercomputing.

[69] Ch. Farhat. Multiprocessors in computational mechanics. Technical report, University of Liège, Belgium, 1993. Agard-sponsored lectures delivered at LTAS.

[70] Charbel Farhat. Which parallel finite element algorithm for which architecture and which problem. *Engineering Computations*, 7(3):185–195, September 1990.

[71] Charbel Farhat. A lagrange multiplier based divide and conquer finite element algorithm. *J. Comput. Sys. Engrg.*, 2:149–156, 1991.

[72] Charbel Farhat, Po-Shu Chen, and François Xavier Roux. The two-level FETI method - part II: Extension to shell problems. parallel implementation and performance results. *Comput. Meth. Appl. Mech. Engin.*, 1996. to appear.

[73] Charbel Farhat, P.S. Chen, and F. Risler. A simple and unified framework for accelerating the convergence of iterative substructuring methods with lagrange multipliers: Application to the design of new feti coarse problems. Technical Report CU-CAS-96-26, Center for Aerospace Sructures, CU, November 1996.

[74] Charbel Farhat and François Hemez. Improving the convergence rate of a transient substructuring iterative method using the rigid body modes of its static equivalent. In *Structures, Structural Dynamics and Material Conference*, New Orleans, USA, April 10–12 1995. 36rd AIAA/ASME/ASCE/AHS/ASC. AIAA-95-1271.

[75] Charbel Farhat, L. Crivelli, and Michel Géradin. On the spectral stability of time integration algorithms for a class of constrained dynamics problems. In *Structures, Structural Dynamics and Material Conference*, La Jolla, USA, April 19–22 1993. 34rd AIAA/ASME/ASCE/AHS/ASC.

[76] Charbel Farhat and Michel Géradin. A hybrid formulation of a component mode synthesis method. In *Structural Dynamics and Material Conference*, Dallas USA, April 13–15 1992. 33rd AIAA/ASME/AHS/ASC Structures.

[77] Charbel Farhat and Michel Géradin. Using a reduced number of Lagrange multipliers for assembling a parallel incomplete field finite element approximation. *Comput. Meth. Appl. Mech. Engin.*, 97:333–354, 1992.

[78] Charbel Farhat and Jan Mandel. The two-level FETI method for static and dynamic plate problems - part I: An optimal iterative solver for biharmonic systems. *Comput. Meth. Appl. Mech. Engin.*, 1996. to appear.

[79] Charbel Farhat and Franqis-Xavier Roux. An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. *SIAM J. Sci. Stat. Comput.*, 13(1):379–396, 1992.

[80] Charbel Farhat, Eddy Pramono, and Carlos Felippa. Towards parallel I/O in finite element simulations. *International J. Numer. Methods Engineering*, 28:2541–2553, 1989.

[81] Charbel Farhat and Daniel Rixen. A new coarsening operator for the optimal preconditioning of the dual and primal domain decomposition methods: application to problems with severe coefficient jumps. In *Proceedings of the Conference on Multigrid Methods*, April 3-7 Copper Mountain 1995.

[82] Charbel Farhat and F.-X. Roux. A method of finite tearing and interconnecting and its parallel solution algorithm. *Comput. Meth. Appl. Mech. Engin.*, 32:1205–1227, 1991.

[83] Charbel Farhat, N. Sobh, and K.C. Park. Dynamic finite element simulations on the connection machine. *Int. Jrnl. of High Speed Computing*, 1(2):289–302, 1989.

[84] Charbel Farhat and Nahil SOBH. A consistency analysis of a class of concurrent transient implicit/explicit algorithms. *Comput. Meth. Appl. Mech. Engin.*, 84:147–162, 1990.

[85] C.A. Felippa and K.C. Park. A direct flexibility method. Technical Report CU-CAS-97-01, Center for Aerospace Structures, CU, January 1997.

[86] J. Fish, V. Belsky, and M. Pandheeradi. Iterative and direct solvers for interface problems with Lagrange multipliers. *Computing Systems in Engineering*, 6(3):261–273, June 1995.

[87] G. Fox, M. Johnson, G. Lysenga, S. Otto, J. Salmon, and D. Walker. *Solving problems on concurrent processors*, volume 1. Prentice Hall, 1988.

[88] R.W. Freund, O. Axelson, F.-X. Roux, R. Keunings, K. Vuik, and H. van der Vorst. Recent advances in iterative methods for solving algebraic systems and eigenvalue problems. Lecture notes, March 1994. PAI Course, KUL.

[89] M. Géradin, D. Coulon, and J.-P. Delsemme. Parallelization of the SAMCEF finite element software through domain decomposition and FETI algorithm. *Internat. J. Supercomp. Appl.*, 11:286:298, 1997.

[90] M. Géradin and D. Rixen. *Théorie des Vibrations. Application à la dynamique des structures.* Physique Fondamentale et Appliquée. Masson, Paris, 2d edition, 1996.

[91] M. Géradin and D. Rixen. *Mechanical Vibrations. Theory and Application to Structural Dynamics.* Wiley & Sons, Chichester, 2d edition, 1997.

[92] L. Giraud and R. S. Tuminaro. Time dependent solvers on distributed memory computers. Technical Report TR/PA/95/03, CERFACS, 1995.

[93] Luc Giraud. Shared and distributed implementation of block preconditioned conjugate gradient using domain decomposition on a distributed virtual shared memory computer. Technical Report TR/PA/92/91, CERFACS, Toulouse, 1992.

[94] Roland Glowinsky and Patrick Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Non-Linear Mechanics.* Studies in Applied Mathematics. SIAM Philadelphia, 1989.

[95] G. Golub and R. Underwood. The block lanczos method for computing eigenvalues. *Math. Soft.*, III:364–377, 1977.

[96] G.H. Golub and C.F. Van Loan. *Matrix computations.* Johns Hopkins University Press, 1991.

[97] Keith D. Gremban, Gary L. Miller, and Marco Zagha. Performance evaluation of a new parallel preconditioner. Technical Report CMU-CS-94-205, School of Computer Science, Carnegie Mellon University, 1994.

[98] Liang Guo-ping and He Jiang-heng. The non-conforming domain decomposition method for elliptic problems with Lagrangian multipliers. *Chinese J. Num. Math. & Appl.*, 1(15):8–19, 1993.

[99] J.L. Gustafson, G.R. Montry, and R.E. Benner. Development of parallel methods for a 1024-processor hypercube. *SIAM J. Sci. Statist. Comput.*, 9(4):609–638, 1988.

[100] J.F. Hajjar and J.F. Abel. On the accuracy of some domain-by-domain algorithms for parallel processing of transient structural dynamics. *Internat. J. Numer. Meths. Engrg.*, 28:1855–1874, 1989.

[101] R.S. Harichandran and Binshan Ye. A method of deriving parallel algorithms for direct integration in structural dynamics. *Computing Systems in Engineering*, 4(4–6):415–420, 1993.

[102] G.L. Hennigan, S. Castillo, and E. Hensel. Using domain decomposition to solve symmetric, positive-definite systems on the hypercube computer. *Internat. J. Numer. Meths. Engrg.*, 33:1941–1954, 1992.

[103] Hermès, editor. *Colloque National en Calcul des Structures, Giens*, May 11-14 1993.

[104] Hermès, editor. *Deuxième Colloque National en Calcul des Structures, Giens*, May 16-19 1995.

[105] M. R. Hestnes and E. Stiefel. Method of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, 49:409–436, 1952.

[106] S.H. Hsieh and J.F. Abel. Use of networked workstations for parallel nonlinear structural dynamic simulations of rotating bladed-disk assemblies. *Computing Systems in Engineering*, 4(4–6):521–530, 1993.

[107] T. J. R. Hughes, R. M. Ferencz, and J. O. Hallquist. Large-scale vectorized implicit calculations in solid mechanics on a cray x-mp/48 utilizing ebe preconditioned conjugate gradients. *Comput. Meth. Appl. Mech. Engin.*, 61:215–248, 1987.

[108] Thomas R. Hughes and Robert M. Ferencz. Fully vectorized EBE preconditioners for nonlinear solid mechanics: Applications to large-scale three-dimensional continuum, shell and contact/impact problems. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, USA, 1988. SIAM.

[109] G. M. Hulbert and J. Chung. The unimportance of the spurious root of time integration algorithms for structural dynamics. *Comm. Num. Meth. Eng.*, 10(8):591–597, 1994.

[110] G. M. Hulbert and T.J.R. Hughes. An error analysis of truncated starting conditions in step-by-step time integration: Consequences for structural dynamics. *Earthqu. Eng. Str. Dyn.*, 15:901–910, 1987.

[111] T. Hwang and I.D. Parsons. Multigrid solution procedures for structural dynamics eigenvalue problems. *Computational Mechanics*, 10:247–262, 1992.

[112] Y. Iliash, Y. Kuznetsov, and Y. Vassilevsky. Efficent parallel solving the potential flow problems on non-matching grids. *Numerical Methods in Engineering '96*, pages 469–475, 1996. Eccomas.

[113] J.F. Imbert. *Analyse des structures par éléments finis.* Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Cepadues Edition, 1991.

[114] S.L. Johnsson and K.K. Mathur. Experience with the conjugate gradient method for stress analysis on a data parallel supercomputer. *International J. Numer. Methods Engineering*, 27:523–546, 1989.

[115] Y.A. Kuznetsov. Overlapping domain decomposition with non-matching grids. In *Proceedings of the Ninth Conference on Domain Decomposition*, June Bergen 1996. in press.

[116] C. Lacour. Non conforming domain decomposition method for plate problems. In *Proceedings of the tenth conference on Domain Decomposition methods, Boulder, Colorado*, August 1997.

[117] C. Lacour and Y. Maday. Domain decomposition: the Mortar element method and a preconditioner for a non conforming method. In *Proceedings of the ninth conference on Domain Decomposition, Bergen*, June 1996. in press.

[118] P. Ladevèze and Ph. Lorong. Formulation et stratégies parallès pour l'analyse non linéaire des structures. In Hermès [103], pages 911–919.

[119] T.A. Laursen and B.N. Maker. An augmented Lagrangian quasi-Newton solver for constrained non-linear finite elements applications. *Internat. J. Numer. Meths. Engrg.*, 38:3571–3590, 1995.

[120] A. Lerusse and M. Géradin. A first approach to pvm. Technical Report IA-2, LTAS, June 1992.

[121] A. Lerusse and M. Géradin. Lu decomposition by precutted blocks. Technical Report IA–3, LTAS, 1993.

[122] J.-L. Lerusse. Domain decomposition techniques in tire finite element analysis. Travail de fin d'études, Faculté des Sciences Appliquées, Université de Liège, Belgium, 1996.

[123] N. Maman and C. Farhat. Matching fluid and structure meshes for aeroelastic computations: a parallel approach. *Computers & Structures*, 54:779–785, 1995.

[124] J. Mandel, R. Tezaur, and C. Farhat. An optimal Lagrange multiplier based domain decomposition method for plate bending problems. Technical Report CU-CAS-95-15, Center for Aerospace Structures, CU, 1995. to appear in Int. J. Numer. Meth. Engrg.

[125] Jan Mandel. Balancing domain decomposition. *Comm. Num. Meth. Engrg.*, 9:233–241, 1993.

[126] Jan Mandel and G. Scott Lett. Domain decomposition preconditioning for p-version finite elements with high aspect ratios. *Applied Numerical Analysis*, 8:411–425, 1991.

[127] T. A. Manteuffel. Shifted incomplete cholesky factorization. In I. S. Duff and eds. G. W. Stewart, editors, *Sparse Matrix Proceedings*, pages 41–61. SIAM, 1978.

[128] Bertrand Mercier. La place du calcul à la Direction des Etudes et Techniques Automobiles de PSA. In Hermès [104], pages 37–42.

[129] C. Militello and C. Felippa. The first ANDES elements: 9-dof plate bending triangles. *Comput. Meth. Appl. Mech. Engin.*, 93:217–246, 1991.

[130] Jean-Baptiste Mouillet and Patrick Laborde. Méthodes de condensation adaptées au calcul explicite de la réponse dynamique des structures. In Hermès [104], pages 403–408.

[131] Raju Namburu and David Turner. An effective data parallel self-starting explicit methodology for computational structural dynamics on the connection machine CM-5. *International J. Numer. Methods Engineering*, 38:3211–3226, 1995.

[132] Raju Namburu and David Turner. Non-conforming mesh gluing in the finite elements method. *International J. Numer. Methods Engineering*, 38:2283–2292, 1995.

[133] M. Ortiz and B. Nour-Omid. Unconditionally stable concurrent procedures for transient finite element analysis. *Comput. Meth. Appl. Mech. Engin.*, 58:151–174, 1986.

[134] M. Papadrakakis. *Solving Large Scale Problems in Mechanics. The Development and Application of Computational Solution Methods.* John Wiley and Sons, Ltd., New York, London, Sydney, 1993.

[135] M. Papadrakakis, S. Bitzarakis, and A. Kotsopulos. An improved dual domain decomposition technique for computational structural mechanics. In *Numerical Methods in Engineering '96*, pages 483–490. ECCOMAS, 1996.

[136] M. Papadrakakis and C.J. Gantes. Preconditioned conjugate and secant-Newton methods for non-linear problems. *International J. Numer. Methods Engineering*, 28:1299–1316, 1989.

[137] M. Papadrakakis and S. Smerou. A new implementation of the Lanczos method in linear problems. *International J. Numer. Methods Engineering*, 29:141–159, 1990.

[138] M. Papadrakakis and Y. Tsompanakis. Domain decomposition methods for parallel solution of sensitivity analysis problems. Technical Report 96-1, Institute of Structural Analysis and Seismic Research, NTUA, Athens, Greece, 1996.

[139] K.C. Park. Partitioned solution of reduced-integrated finite element equations. Technical Report CU-CAS-97-04, Center for Aerospace Structures, CU, August 1997.

[140] K.C. Park and C.A. Felippa. A variational framework for solution method develpments in structural mechanics. Technical Report CU-CAS-96-21, Center for Aerospace Structures, CU, May 1997. to appear in Journal of Applied Mechanics.

[141] K.C. Park, M.R. Justino, and C.A. Felippa. An algebraically partitioned FETI method for parallel structural analysis: Algorithm description. *International J. Numer. Methods Engineering*, 40:2717–2737, 1997.

[142] Theodore H. Pian. Finite element formulation by variational principles with relaxed continuity requirements. In A.K.Aziz, editor, *The Mathematical Foundation of the Finite Element Method with Applications to Partial Differential Equations, Part II*, pages 671–687. Academic Press, London, 1972.

[143] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems - part I: Model problem, theory and two-dimensional application. *Comput. Meth. Appl. Mech. Engin.*, 124(1-2):79–112, 1995.

[144] Makan Pourzandi and Bernard Tourancheau. Parallel performance of a Jacobi eigenvalue solution. *Computing Systems in Engineering*, 6(4/5):377–383, 1995. special issue:*International Meeting on Vector and Parallel Processing, VECPAR 3.*

[145] Luis Quiroz. *Connexion des Maillages Hétérogènes dans la Méthodes des Eléments Finis.* PhD thesis, Faculté des Sciences Appliquées, University of Liège, Belgium, 1993.

[146] C. Rey and F. Lene. Generalized krylov correction of the conjugate radient for large-scale non-linear elasticity problems. In *Proceedings of the ninth conference on Domain Decomposition, Bergen*, June 1996. in press.

[147] Ch. Rey, F. Devries, and F. Lene. Parallelism in non linear computation of heterogen structures. *Calculateurs Parallèles*, 7(3), 1995.

[148] Daniel Rixen. *Substructuring and Dual Methods in Structural Analysis.* PhD thesis, Université de Liège, Belgium, Collection des Publications de la Faculté des Sciences appliquées, n° 175, 1997.

[149] Daniel Rixen and Charbel Farhat. Highly accurate and stable algorithms for the static and dynamic analyses of independently modeled substructures. In *Structural Dynamics and Material Conference*, Salt Lake City, April 18–19 1996. 37rd AIAA/ASME/ASCE/AHS/ASC Structures.

[150] Daniel Rixen and Charbel Farhat. Preconditioning the FETI and balancing domain decomposition methods for problems with intra- and inter-subdomain coefficient jumps. In *Proceedings of the ninth conference on Domain Decomposition*, June 1996. in press.

[151] Daniel Rixen, Charbel Farhat, and M. Géradin. A Ritz based smoothing procedure for improving the static and dynamic assembly via Lagrange multipliers of independent substructure solutions. *Comput. Meth. Appl. Mech. Engin.*, 1997. accepted for publication.

[152] Daniel Rixen, Charbel Farhat, and Michel Géradin. An interface smoothing procedure for the FETI method: Application to static and dynamic structural analyses. Technical Report LTAS VA-139 report, University of Liège, Belgium, May 1994.

[153] Daniel Rixen, Charbel Farhat, and Michel Géradin. Approximation du préconditionneur de Dirichlet pour la résolution itérative du problème d'interface de la méthode hybride FETI. In Hermès [104], pages 655–660.

[154] Daniel Rixen, Carole Thonon, and Michel Géradin. Impedance and admittance of continuous systems and comparison between continuous and discrete models. In *Advanced Mathematical Methods in the Dynamics of Flexible Bodies*, Noordwijk, The Netherlands, June 3–5 1996. ESA International Workshop.

[155] Charles Rogé and Franǫis Xavier Roux. Méthodes de calcul de structures composites par sous-domaines sur calculateurs MIMD. In Hermès [103], pages 920–927.

[156] F.-X. Roux. *Méthode de Décomposition de Domaine à l'Aide de Multiplicateurs de Lagrange et Application à la Résolution en Parallèles des Equations de l'Elasticité Linéaire.* PhD thesis, Université Paris 6, 1989.

[157] F.-X. Roux. Dual and spectral properties of schur and saddle-point domain decomposition methods. In D. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs, and R.G. Voigt, editors, *Domain Decomposition Methods for Partial Differential Equations*, 1991.

[158] F.-X. Roux. Parallel implementation of a domain decomposition method for non-linear elasticity problems. In D. Keyes, Y. Saad, and D. Truhlar, editors, *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, pages 161–175, 1994.

[159] F.-X. Roux. Private communication, 1996.

[160] 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.

[161] F.X. Roux. Programmation des super-calculateurs scientifiques vectoriels et parallèles. Technical Report FR ISSN 0078-3781, Office National d'Etudes et de Recherches Aérospatiales, 1991.

[162] J. W. Ruge and K. Stuben. Algebraic multigrid. In S. F. McCormick, editor, *Multigrid Methods*, Frontiers in Applied Mathematics, pages 73–130. SIAM, 1987.

[163] Y. Saad. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Math. Comput.*, 48:651–662, 1987.

[164] Abdulmannan Saati, Sedat Biringen, and Charbel Farhat. Solving Navier-Stokes equations on a massively parallel processor: Beyond the 1Gflop performance. *Int. J. Supercomp. Appl.*, 4(1):72–80, 1990.

[165] A. Samuelson, N-E. Wiberg, and L. Bernspång. A study of the efficiency of iterative methods for linear problems in structural mechanics. *International J. Numer. Methods Engineering*, 22:209–218, 1986.

[166] Weiping Shen, Jiahao Lin, and F.W. Williams. Parallel computing for the high precision direct integration method. *Comput. Meth. Appl. Mech. Engin.*, 126:315–331, 1995.

[167] Jonathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CS-94-125, School of Computer Science, Carnegie Mellon University, August 1994.

[168] Juan C. Simo, Peter Wriggers, and Robert L. Taylor. A perturbed Lagrangian formulation for the finite element solution of contact problems. *Comput. Meth. Appl. Mech. Engin.*, 50:163–180, 1984.

[169] Gerard L.G. Sleipjen and Henk A. van der Vorst. An overview of approaches for the stable computation of hybrid BiCG methods. *Applied Numerical Mathematics*, 19(3):235–254, December 1995. Special Issue on Iterative Methods for Linear Equations.

[170] Damien Soulat and F. Devries. Mechanical criteria for the subdomains decomposition: applications to heterogeneous structures and composite materials. In *Proceedings of the ninth conference on Domain Decomposition, Bergen*, June 1996. in press.

[171] Damien Soulat and Marina Vidrascu. Prise en compte de critères mécaniques et géométriques dans le choix de la décomposition en sous-domaines. In Hermès [104], pages 711–716.

[172] P.S. Su and R.E. Fulton. A parallel domain decomposition finite element method for massively parallel computers. *Computing Systems in Engineering*, 4(4-6):489–494, 1993.

[173] M. Suarjana and K.H. Law. Successive conjugate gradient methods for structural analysis with multiple load cases. *International J. Numer. Methods Engineering*, 37:4185–4203, 1994.

[174] S. Succi. Parallel computing in industrial environment. In *Computational Methods in Applied Sciences '96*, pages 186–189. ECCOMAS, 1996.

[175] Patrick Le Tallec. Domain-decomposition methods in computational mechanics. *Computational Mechanics Advances*, 1:121–220, 1994. North-Holland.

[176] Patrick Le Tallec, Jan Mandel, and Marina Vidrascu. A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. *j-SIAM-J-NUM-ANA*, 1997. in press.

[177] Patrick Le Tallec and A. Patra. Non-overlapping domain decomposition methods for adaptive hp approximations of the Stokes problem with discontinuous pressure fields. Technical report, Texas Institute for Computational Mathematics, August 1996.

[178] Patrick Le Tallec, Yann-Hervé De Roeck, and Marina Vidrascu. Domain-decomposition methods for large linearly elliptic three dimensional problems. *J. of Computational and Applied Mathematics*, 34, 1991. Elsevier Science Publishers, Amsterdam.

[179] Patrick Le Tallec and Marina Vidrascu. Méthodes de décomposition de domaines en calcul de structures. In Hermès [103], pages 33–49.

[180] Patrick Le Tallec and Marina Vidrascu. Generalized Neumann-Neumann preconditioners for iterative substructuring. In *Proceedings of the ninth conference on Domain Decomposition*, June Bergen 1996. to appear.

[181] Carole Thonon, Daniel Rixen, and Michel Géradin. Unification of impedance/admittance and component mode formulations for the assembling of flexible structures. In *Advanced Mathematical Methods in the Dynamics of Flexible Bodies*, Noordwijk, The Netherlands, June 3–5 1996. ESA International Workshop.

[182] Denis Vanderstraeten. *Algorithms for Parallel Finite Element Computations: Mesh Partitioning and Hybrid Solver*. PhD thesis, Université Catholique de Louvain, Faculté des Sciences Appliquées, February 1996.

[183] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. *SIAM J. Sci. Stat. Comput.*, 1997. in press.

[184] W.H. Ware. The ultimate computer. *IEEE Spectrum*, 2:273–279, 1972.

[185] J.D. Whitcomb. Iterative global/local finite element analysis. *Computers & Structures*, 40:1027–1031, 1991.

[186] D.Y. Xue and Chuh Mei. Non-linear vector eigen-solver and parallel reassembly processing for structural non-linear vibration. *Computing Systems in Engineering*, 4(4-6):513–519, 1993.

[187] O. Zone. *Algorithmes Parallèles de Résolution de Probleèmes d'élément finis*. PhD thesis, Université Catholique de Louvain, Faculté des Sciences Appliquées, June 1995.