FETI methods for acoustic problems with porous materials

Chaitanya Sanghavi

THESE DE DOCTORAT DE

LE MANS UNIVERSITÉ

ECOLE DOCTORALE N° 602
Sciences pour l'Ingénieur
Spécialité : « *acoustique* »

Par

«Chaitanya SANGHAVI»

«FETI methods for acoustic problems with porous materials»

Thèse présentée et soutenue à « Le Mans », le « 29/09/20 »
Unité de recherche : LAUM – UMR CNRS 6613

Numéro National de Thèse (NNT): 2020LEMA1021

Rapporteurs avant soutenance :

Jean-François Deü	Professeur, CNAM Paris, France
Emmanuel Perrey-Debain	Professeur, UTC Compiègne, France

Composition du Jury :

Président :
Examinateurs :

Hadrien Bériot	Ingénieur de recherche senior, Siemens Digital Industries Software, Belgium
Olivier Dazel	Professeur, LAUM UMR CNRS 6613, France
Elke Deckers	Assistant Professeur, KU Leuven, Belgium
Jean-François Deü	Professeur, CNAM Paris, France
Gwénaël Gabard	Professeur, LAUM UMR CNRS 6613, France
Emmanuel Perrey-Debain	Professeur, UTC Compiègne, France
François-Xavier Roux	Professeur, UPMC and ONERA, France

Dir. de thèse : Olivier Dazel
Co-dir. de thèse : Gwénaël Gabard
Co-encadrant : Hadrien Bériot

Le Mans Université

Title : **FETI methods for acoustic problems with porous materials**

Keywords : FETI-2LM, FETI-H, equivalent fluids, iterative methods, two-level FETI, recycling.

Abstract :

Sound absorbing materials such as foams are widely used in many industrial and domestic applications to absorb undesirable sound. One needs to perform many calculations to get desirable properties of the treatment using optimization strategies.

The state-of-the-art computational models require prohibitively high computational time. The problematic of this PhD is to reduce the computational effort for such models to speed up design calculations.

This document is a synthesis of the work carried out in this direction. The problem is addressed using Domain Decompostion methods (DDM). It consists of splitting the original problem into small parts referred to as subdomains.

A partial solution is computed on these subdomains to match the global continuity in the domain of interest. Different DDM methods are benchmarked in terms of performance and scalability, specific for porous materials. Any DDM consists of two major costs, the factorization of the subdomains and iterative part for the global convergence. A novel factorization strategy is implemented and applied in 2D and 3D to demonstrate savings in time compared to coventional approaches. In the second part, the method is further improved to reduce the iterative costs for a series of calculations.

A final workflow is proposed to make the computational cost of these models afforable within industrial timeframes.

Titre : Méthodes FETI pour les problèmes acoustiques avec des matériaux poreux

MOTS-CLÉS: FETI-2LM, FETI-H, fluides équivalents, méthodes itératives, FETI-2LM à deux niveaux, recyclage

Résumé :
Le présent travail traite de la modélisation numérique de traitements acoustiques à grande échelle en utilisant des méthodes de décomposition de domaine (DDM). Les principaux objectifs de la thèse sont triples :

Premièrement, il est fait le choix de considérer une méthode de décomposition de domaine (DDM) appropriée pour résoudre des problèmes de Helmholtz homogènes et hétérogènes à grande échelle. Deux variantes différentes des méthodes FETI, à savoir FETI-2LM et FETI-H, sont mises en œuvre et comparées, afin de déterminer la plus appropriée.Deuxièmement, une nouvelle stratégie de résolution est appliquée au DDM choisi pour étudier de nombreux traitements acoustiques.Cette stratégie vise à réduire les coûts des calculs mis en oeuvre.

Troisièmement, des stratégies alternatives sont examinées pour réduire les coûts associés aux calculs itératifs du DDM. Une nouvelle méthode FETI-2LM à deux niveaux est formulée pour réduire les coûts des évaluations successives des revêtements acoustiques. Une autre stratégie de recyclage basée sur l'utilisation des valeurs propres extrêmes est examinée pour la méthode FETI-2LM. Au final on est en mesure de choisir la méthode la plus appropriée.

La modélisation numérique des traitements acoustiques à l'aide de méthodes efficaces nécessite des coûts de calcul importants dans la phase de conception. Ce travail fournit une contribution permettant de résoudre ce problème à l'aide de la DDM.

Contents

List of Figures

List of Tables

Acknowledgements

The present research has received funding from the European Union Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Smart Answer project (grant agreement No 722401). The research work was done in collaboration with Le Mans University and Siemens Digital Industries Software.

I would like to give a special acknowledgment to Dr. Hadrien Bériot for giving me an opportunity to be a part of this wonderful project and continuously supporting me over these three years.

I wish to show my gratitude for my supervisors Prof. Olivier Dazel and Prof. Gwénaël Gabard. During the complete project, they taught and trained me patiently and were always available for me.

I would like to thank Prof. François-Xavier Roux for giving his time and detailed insight leading to interesting discussions, specifically for the two-level domain decomposition methods. I would like to thank Prof. Jean-François Deü and Prof. Emmanuel Perrey-Debain for taking their time to read the manuscript and giving me valueable feedback. I express my gratitude towards all the jury members, for allowing me to present this work.

I am indebted to my parents and sister for supporting me throughout my life. Without their tutoring, I would not be here.

I wish to thank my old flatmate Ayoub, who helped me during the application process. I wish to thank all my colleagues at Siemens especially Ali and Yue who helped me go through challenging weeks at work. I would like to acknowledge all the Smart Answer fellows, especially Nacho with whom I spend some amazing times at workshops.

Lastly, I would like to thank all my friends from Pune, Barcelona, Stuttgart, Leuven for always been there with me especially during the tough times.

Chapter 1

Introduction

1.1 Context

The mitigation of undesirable sound for industrial and household applications is an area of active research. There are different ways of attenuating undesirable noise. Firstly, one can reduce or modify the noise at the source. Secondly, noise can be attenuated along the propagation path using sound absorbing materials. Thirdly, the noise can be mitigated at the receiver end using noise canceling devices like ear plugs, headphones, etc.
In the present work, the focus is on noise reduction along the propagation path using porous sound absorbing materials. The design and optimization of these materials for testing purposes on practical applications can be expensive and time consuming. Numerical simulations can be a quick and cost-effective approach to support the early design phase of sound absorbing materials. However, one needs to perform acoustic simulations for a large number of configurations to identify the optimal acoustic treatment for a given design problem. This can result in long calculation times. The present work aims to address the relevant computational issues associated with such large-scale models using Domain Decomposition Methods (DDM).

1.2 Objectives

The overall aim of this thesis is to solve efficiently large-scale optimization problems for a range of acoustic treatments. The specific objectives of the thesis are as follows:

1. The selection of an efficient DDM for solving large-scale acoustic problems.

2. The extension of the FETI method (a particular DDM technique) to problems combining differ-

ent absorbing materials modeled as equivalent fluids.

3. The reduction of the resolution cost involved when solving a series of models for a range of acoustic treatments.

4. The reduction in the number of additional iterations while using the FETI method through the use of recycling strategies.

Thus the present work contributes to the state-of-the-art computational methods for modeling several acoustic treatments. It proposes an improved methodology to significantly reduce the computational effort.

1.3 Overview of the thesis

In this part, a brief outline of the methodology which aims to achieve the aforementioned objectives is presented.

1.3.1 Modeling of porous materials

In Chapter 2, the theory for modeling acoustic treatments and details about the numerical discretization (specifically p-FEM) are presented. A review of different optimization techniques to identify the most suited liner configuration is also described. An axi-symmetric model of a duct with an acoustic treatment is introduced and used as a cross-verification of a more expensive 3D model. A scan of the design space for the acoustic treatment is performed to identify the optimal liner configuration for the 3D model. Several issues in these computational procedures are highlighted. The forthcoming chapters aim at addressing these issues.

1.3.2 Overview of domain decomposition methods for acoustics

Chapter 3 begins with a literature review of various domain decomposition methods applied to Helmholtz problems. Among these, two non-overlapping methods are examined in detail to solve large-scale Helmholtz problems. The first method is FETI-2LM which is based on using two Lagrange multipliers to couple the partitioned subdomains. The second method is FETI-H which relies on using only one Lagrange multiplier based on a sign convention associated with the respective subdomains. These two methods are implemented and verified for benchmark academic problems (in 2D and 3D) using the high-order finite element method p-FEM.

1.3.3 FETI models with different acoustic treatments

In Chapter 4, the continuity and transmission conditions used in FETI methods are generalized to model heterogeneous media with multiple equivalent fluids. Both aforementioned FETI-methods are implemented and verified against known analytical solutions in 2D and 3D for heterogeneous media. Scalability tests are reported to assess the performance and behavior of both methods for varying problem resolution, partitions, and frequencies. It is then explained how the FETI methods can be used efficiently when dealing with series of calculations where only a small part of the problem changes from one calculation to another. The key idea is to reuse the factorization of the sub-domains that remained unchanged between two calculations.

1.3.4 Two-level DDM

In Chapter 5, a two-level DDM is proposed. It consists of partitioning the domain into two nested levels and formulates an interface problem for each of these respective levels. A two-level FETI (a specific DDM) method is formulated and tested for 2D benchmark problems. Various ways to define tolerance at the bottom-level are considered and assessed. Besides, three different iterative schemes are compared for this two-level approach.

1.3.5 Recycling strategies

In addition to reducing the factorization costs, it is possible to accelerate domain decomposition methods by reducing the number of iterations required for convergence which represents a significant portion of the computational effort. In Chapter 6, two recycling strategies namely IKRS and GCRO-DR (abbreviations of the recycling techniques) are considered to speed up the convergence of the iterative schemes used to solve the interface problem. These techniques are applied to test cases relevant to acoustic treatments and their benefits are assessed.

In Chapter 7, a summary of all key conclusions is presented with possible future work.

1.4 Key Contributions

The main contributions from the present work are as follows:

1. Generalised transmission conditions are proposed for the FETI method to model problems with heterogeneous Helmholtz regions. In addition such regions are composed of equivalent fluids,

meaning that their densities and sound speeds can be complex valued to account for sound attenuation.

2. When dealing with a series of computations where only a small part of the model changes, it is shown how the FETI method can be used to reuse the existing factorization in the subdomains that are not modified. This significantly reduces the overall cost of scanning a design parameter space for instance.

3. A two-level FETI model is proposed. Various strategies for setting the tolerance for the bottom-level problem are assessed along with three different iterative solvers.

4. Two different recycling strategies are considered and assessed to improve the convergence of FETI method.

Chapter 2

Models for acoustic problems with porous materials

2.1 Introduction

Noise control has gained importance in the automotive and aerospace industries due to the stringent international standards to prevent environmental health hazards [93]. To fulfill these requirements, the design and development of noise mitigation techniques are of major importance. Porous materials (composed of porous skeleton saturated by a fluid) are commonly used to attenuate the undesirable sound propagation in industrial applications. The physics of conventional porous sound absorbing materials is now mastered in most of its aspects and some of these models have become commonplace for many industrial applications. Over the last two decades, several numerical modeling methods have been proposed to compute the dynamic response of vibro-acoustic systems treated with porous materials. Typically one resorts to performing a large number of calculations involving an optimization strategy to chose the most efficient configuration. Although considered mature enough, this workflow typically requires prohibitively large amounts of computational resources to get acceptable results. A bibliographic review of such a workflow will be examined and relevant drawbacks will be highlighted. The objective of this chapter is to present numerical methods for the Helmholtz problem, porous materials modeling, acoustic liners, which are a type of acoustic treatment, and then the typical configurations of interest. It is then composed of a bibliographic part and some classical calculation results which allow the reader to handle the problem of interest in a better way.

In the first part, a brief review of different numerical methods for the Helmholtz equation, specifically higher-order finite element methods are illustrated. In the second part, the state-of-the-art modeling techniques for acoustic treatments are presented. In the third part, a simplified axi-symmetric acoustic

cavity with acoustic treatment is presented with its corresponding results. Lastly, these results are cross-verified with the 3D model for large numbers of evaluations to obtain the acoustic response of the structure with the acoustic treatment. The relevant computational issues are highlighted and attempts will be made to tackle them in the forthcoming chapters.

2.2 Higher order FEM for Helmholtz equation

This section presents an overview of the numerical discretization of large-scale Helmholtz problems. The time-harmonic, linear wave propagation in a uniform, homogeneous medium within domain Ω is governed by the Helmholtz equation:

$$-(\Delta u + k^2 u) = 0 \qquad \text{in} \qquad \Omega, \tag{2.1}$$

where u is the acoustic pressure, ω is the angular frequency, $k = \omega/c$ is the wavenumber at frequency ω for a medium with celerity c. The implicit, harmonic time dependence is given by $\exp(+\mathrm{i}\omega t)$. A generic Robin boundary condition is applied on the boundary $\partial\Omega$ of Ω to complete the strong form of the problem.

$$\frac{\partial u}{\partial n} + \mathrm{i}ku = g \qquad \text{on} \qquad \partial\Omega, \tag{2.2}$$

where g is a known function and $\mathrm{i}^2 = -1$. Using the continuous Galerkin method, the weak form of the problem can be described as follows:

$$\forall q \in V, \qquad \int_\Omega \boldsymbol{\nabla} q^* \cdot \boldsymbol{\nabla} u - k^2\, q^* u \;\mathrm{d}\Omega - \int_{\partial\Omega} q^* \frac{\partial u}{\partial n} \mathrm{dS} = 0, \tag{2.3}$$

where q is the test function associated with the pressure field u, V is the space of test functions and * corresponds to complex conjugation. In the present work, * is used to represent the complex conjugate and ^{T} is used to denote the transpose. Any FEM-based discretization consists of splitting the computational domain into small non-overlapping elements. On each element Ω_e, the solution is expanded on the shape functions ϕ_n^e, generally using polynomials as follows:

$$u^h(x) = \sum_{n=1}^{N^e_{\mathrm{dof}}} \phi_n^e(\mathbf{x}) u_n^e = \Phi_e(\mathbf{x})^T \mathbf{u}_e \ , \tag{2.4}$$

where u_n^e is the contribution of the n^{th} shape function to the local solution and N^e_{dof} denotes the total number of shape functions contributing on the element, e.

During the last few decades, significant efforts were devoted tothe choice of numerical discretization to solve large-scale Helmholtz problems. Among these, the most common discretizations are based on finite difference, finite element, discontinuous Galerkin, boundary element methods. Finite-difference

numerical schemes are confined to structured meshes and require dedicated manual effort to generate high-quality meshes for realistic 3D geometries [88]. Contrarily, boundary element methods require only a surface mesh for complex 3D geometries but their extension to problems with inhomogeneous media or non-uniform background flow is still an area of research [90]. Finite element and the discontinuous Galerkin methods have become popular schemes for solving problems arising from the discretization of the time-harmonic wave propagation problems. The standard finite elements suffer largely from the dispersion error leading to a so-called pollution effect. An alternative solution is to use wave-based methods that incorporate *a priori* the knowledge of the local behavior of the solution into the numerical scheme. The rationale behind these methods is to capture the dispersion properties of the waves into the numerical approximation to address the aforementioned issue. The Ultra Weak Variational Formulation (UWVF) [48, 18, 17], the least square method (LSM) [70], the wave-based Discontinuous Galerkin Method (DGM) [38] belong to this family of methods. All these methods can be formulated in the framework of the discontinuous Galerkin method where each scheme has different flux splitting methods. A comparison of these three methods for non-uniform coefficients for a Helmholtz problem leads to the conclusion that the LSM is less accurate as compared to UWVF and DGM, whereas the performance of DGM and UWVF is similar [37]. However, the condition number of these schemes is quite high compared to classical or p-FEM based methods [61].

A possible alternative to reduce the dispersion error is to increase the interpolation order of the polynomial basis. An overview of different higher order interpolation basis functions like the Lagrange, integrated Legendre and Bernstein polynomials applied to the Helmholtz equation are compared in [12]. When solved with a Krylov subspace based method, the Bernstein polynomials provide the most efficient and superior performance [76]. In a similar work, the use of Lobatto shape functions was advocated due to their superior conditioning compared to Hermite or Bernstein shape functions [77] [Sec 4.5]. A comparison of the wave-based DGM and the higher-order polynomial, p-FEM (Lobatto shape functions) based interpolations has been performed and it was observed that in some cases p-FEM exhibits superior performance for benchmark 2D problems [61]. Owing to these reasons, the p-FEM based numerical discretization (comprising of continuous higher order approximations of the Lobatto shape functions) is used in the present work. This method is now presented.

For a 1D element, Lobatto shape functions l_i are chosen for the interpolation, i.e. the ϕ_n^e terms in (2.4). Typical to a FEM-based discretization, the physical element is mapped onto a reference element with coordinate ξ, and the shape functions are defined on a 1D reference element, $E^L = \{\xi \in \mathbb{R}; -1 < \xi < 1\}$. The first two functions (associated to order $p = 1$) corresponds to the standard

linear nodal shape functions based on the Lagrange interpolation:

$$l_0(\xi) = \frac{1-\xi}{2}\ , \quad l_1(\xi) = \frac{1+\xi}{2}. \tag{2.5}$$

For higher orders ($p > 1$), the basis is complemented by $p-1$ integrated Legendre polynomials also referred to as Lobatto shape functions:

$$l_n(\xi) = \sqrt{n-\frac{1}{2}} \int_{-1}^{\xi} L_{n-1}(\xi)\ \mathrm{d}\xi\ , \tag{2.6}$$

where, $L_n(\xi)$ is Legendre polynomial of order n. The details about these polynomials can be found in [87] [Sec 1.2.2]. The Lobatto shape functions have optimal conditioning for the stiffness matrix due to the following property:

$$\int_{-1}^{1} l'_{i-1}(\xi) l'_{j-1}(\xi)\ \mathrm{d}\xi = \delta_{ij} \quad i,j > 2, \tag{2.7}$$

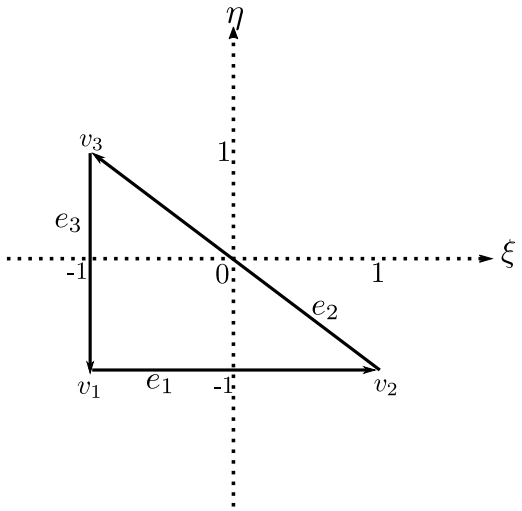


Figure 2.1: Reference triangular element with v_1, v_2, v_3 as vertices, e_1, e_2, e_3 as the edges, ξ, η are the reference coordinates. For more details refer to [87] sec[2.2.3].

For a reference 2D triangular element, as depicted in Figure 2.1, there are three different types of shape functions, namely the nodal, edge and bubble shape functions. The nodal shape functions corresponding to $p = 1$ are the Lagrange shape functions which have the value 1 at the node and 0 at other nodes, as seen in Figure 2.2 (top). The edge shape functions can be seen in Figure 2.2 (middle). Their value is null at the nodes. The bubble shape functions vanish on all edges and at all nodes. These shape functions only appear in the interior of the element as seen in Figure 2.2 (bottom) and thus have no dependence on the neighboring elements. On a given reference triangular element consisting of a fixed polynomial order p, there are three nodal shape functions, $3(p-1)$ edge functions and $(p-1)(p-2)/2$ bubble shape functions. The polynomial interpolation is referred to as

p-FEM. Typically, only linear and quadratic meshes are used to capture the geometrical discretization accurately. On a given reference triangular element Ω_e, the discrete solution of order p is expressed as follows:

$$u^h(\xi,\eta) = \sum_{r=1}^{3} \psi^{v_r}(\xi,\eta) u^{v_r} + \sum_{q=2}^{p} \sum_{r=1}^{3} \psi_q^{e_r}(\xi,\eta) u_q^{e_r} + \sum_{q_1=2}^{p} \sum_{q_2=2}^{p-1-q_1} \psi_{q_1,q_2}^{b}(\xi,\eta) u_{q_1,q_2}^{b}, \tag{2.8}$$

where $u^{v_r}, u_q^{e_r}, u_{q_1,q_2}^b$ are the degrees of freedom (DOFs) associated to the nodal, edge and bubble shape functions respectively.

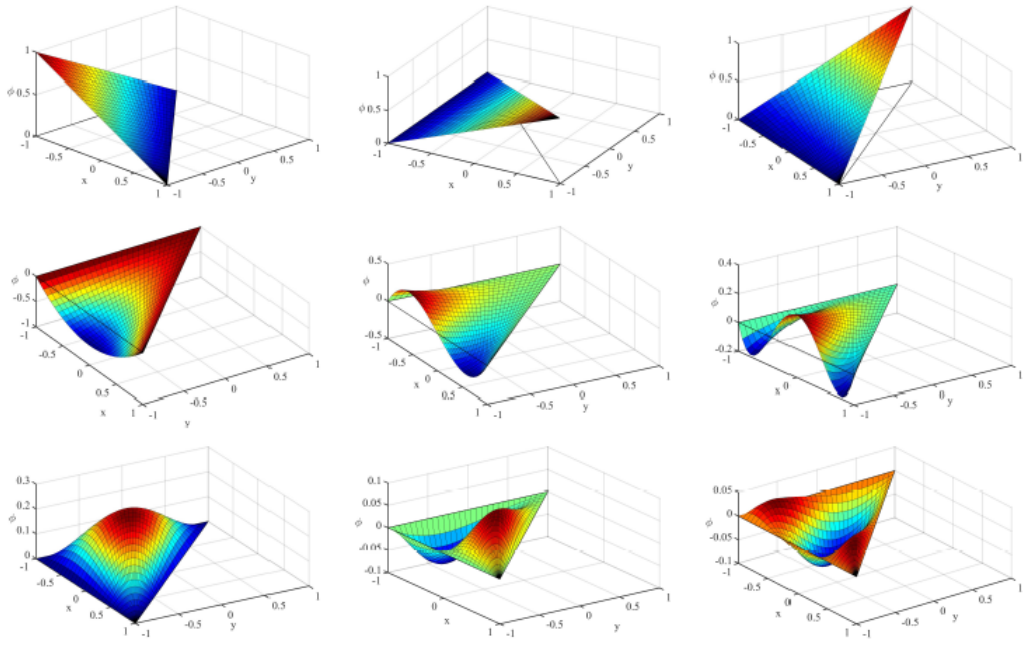


Figure 2.2: Top: Nodal shape functions associated with the vertices of the reference element, Middle: Edge shape functions for p=2, 3, 4 associated with a particular edge on the reference element, Bottom: Bubble shape functions [87].

Another important property of the Lobatto shape functions is their hierarchic nature. The shape functions for a particular order p are reused to construct the shape functions at the next order $p+1$ [87] [Sec 1.3.3]. This allows us to reduce the assembly time for the elementary matrices since only new shape functions need to be evaluated. This can help reduce the computational effort for commercial codes involving industrial applications [10].

The standard p-FEM method consists of using the same constant polynomial order, p across all the elements. The use of the constant p order throughout the domain would be non-optimal as it might be governed by the smallest element size which would result in large resolutions of the systems matrices without much improvement on the accuracy. To circumvent these issues, the adaptive p-FEM was

formulated which allows us to change the polynomial order locally in each element. This helps to reduce the cost of these models drastically in terms of CPU time as well as memory [10].

Condensation

The bubble shape functions vanish on the element boundaries. As a result, these shape functions have no interaction with the neighboring elements. These internal degrees of freedom can, therefore, be eliminated at the assembly stage which helps in improving the bandwidth and the conditioning of the global matrices. Besides, it helps to reduce the size of the global system drastically especially for higher polynomial orders, since the bubble shape functions have a quadratic dependency (in 2D) on the polynomial order, p.

Error estimates

For the classical hp-based finite element method (for the Helmholtz equation), the upper bounds for the global error in the H_1 semi-norm can be stated as below [49]:

$$E < C_1 \left(\frac{kh}{2p} \right)^p + C_2 k \left(\frac{kh}{2p} \right)^{2p}, \tag{2.9}$$

where C_1, C_2 are independent of the mesh, wavenumber and are weak functions of p. h corresponds to the mesh size. The first term corresponds to the interpolation error. It has a scaling of order p with respect to kh and can be easily controlled with a sufficiently fine resolution given by $2\pi/(kh)$. The second term corresponds to the dispersion error and the pollution effect. The dispersion error can be described as the difference between the theoretical and the numerical wavenumber which is actually observed in the model. The accumulation of this error for multiple wavelengths can result in a significant built-up of error, the so-called pollution effect [3]. Thus the dispersion error dominates the overall error in Equation (2.9) for under or just sufficiently resolved models.

As seen in Equation (2.9), increasing the polynomial order provides an exponential convergence rate. In contrast, an algebraic convergence for h refinement is observed. Moreover, when the problem resolution is high (i.e. at low values of kh), the interpolation error dominates and the H_1-norm error converges asymptotically as $\mathcal{O}(kh/2p)^p$ [6]. For the L_2-norm, the actual rate of convergence for the interpolation error increases to $\mathcal{O}(kh/2p)^{p+1}$ [6].

In general, it can be said that for high frequencies, not only the mesh resolution should be high but also the polynomial order. This typically results in a large sparse linear system of equations that can be challenging to solve.

2.3 Solvers for the Helmholtz equation

This section focuses on solving a large sparse linear system of equations arising from the numerical discretization of classical or higher-order finite element methods for the Helmholtz equation:

$$\mathbb{K}\mathbf{u}^h = \mathbf{b}, \tag{2.10}$$

where $\mathbb{K}$ is generally a large, square, sparse, complex symmetric matrix typically ranging up to millions of degrees of freedom. $\mathbf{b}$ is the forcing right-hand side and $\mathbf{u}^h$ is the unknown vector which needs to be computed.

Until recently, the two most common ways to solve any linear system of equations were direct and iterative methods. Over the years, direct methods have been preferred for their robustness, matrix-specific factorization strategies (like Crout, Cholesky), and simplicity for solving multiple right-hand-side problems. Also, if the matrix is tightly banded, the computational effort required for factorization scales linearly with the number of unknowns as $\mathcal{O}(b^2N)$ (for 2D problems), where b refers to the bandwidth and N refers to the dimension of the matrix [28]. However, even with renumbering, the factorization results in fill-in demanding additional memory requirements.

Moreover, even with the use of multi-frontal solvers, they do not scale favorably with the matrix dimensions [4]. The parallelization is not easy and in cases when this is done, the parallel scalability ranges from 50-80 % [4]. Finally, doubling the frequency typically results in an increase by factor ten on the memory requirements [36].

On the other hand, iterative solvers approach the solution in steps without having to do the expensive factorization steps. Most of the iterative schemes are based on dealing with diagonally dominant or positive definite matrices like Conjugate Gradient. However, these methods are not suitable to tackle indefinite problems such as the Helmholtz equation. To circumvent these issues, the use of Krylov subspace methods (like GMRES, ORTHODIR, GCR, etc.) are advocated [85]. These methods consist in looking for an approximate solution $\mathbf{x}_m$ from an affine subspace [8] $\mathbf{x}_0 + \mathcal{K}_m$ of m dimension by imposing Petrov-Galerkin condition as follows:

$$\mathbf{b} - \mathbb{K}\mathbf{x}_m \perp \mathcal{L}_m, \tag{2.11}$$

where $\mathcal{L}_m$ is another subspace of dimension m, $\mathbf{x}_0$ is the initial guess for the solution, the Krylov subspace $\mathcal{K}_m$ is defined as follows:

$$\mathcal{K}_m(\mathbb{K}, \mathbf{r}_0) = span\ \{\mathbf{r}_0, \mathbb{K}\mathbf{r}_0, \mathbb{K}^2\mathbf{r}_0, ...\mathbb{K}^{m-1}\mathbf{r}_0\}, \tag{2.12}$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbb{K}\mathbf{x}_0$ [84]. The choice of the constraint subspace $\mathcal{L}_m$ used to build the approximate solutions plays an important role in the iterative scheme. A detailed review of the different choice $\mathcal{L}_m$ is described in [84].

However, for multiple right-hand-side systems, the iterative process needs to be typically restarted from the initial guess. This issue is tackled using block Krylov subspace methods. This topic is discussed in detail in the later chapters of the thesis. However, even for these advanced methods, the convergence of the iterative scheme suffers when applied to indefinite problems. A standard iterative scheme was applied to Helmholtz and a Laplace problem for a point source [29]. It was observed that GMRES suffers from a sluggish convergence. Similar behaviors are also observed in advection-diffusion problems [29]. Typically these iterative methods are used with a preconditioner, to improve the conditioning of the original system. A review of the various preconditioners applied to the Helmholtz problem and their relevant issues are discussed are [44]. In general, iterative methods scale better as it typically requires limited resources however, they lack the necessary robustness of direct solvers.

2.4 Porous sound absorbing materials

Porous sound absorbing materials play a crucial role for noise mitigation in many industrial applications. These materials are composed of channels, cracks, or cavities (generally called pores) which allows the sound waves to enter and propagate in this internal network. This propagation is associated with damping mechanisms: the friction between the air molecules and pore walls. This leads to the conversion of the sound energy into heat which is later dissipated. Moreover, the acoustic waves that penetrate inside the porous result in periodic compression and release of air. This results in energy consumption [15].

2.4.1 Acoustic liners

Figure 2.3: Installation of an acoustic liner at the intake of turbofan engine [53].

Acoustic liners are commonly used for noise mitigation of turbofan engines. Figure 2.3 shows a typical example of an intake liner installed on the nacelle. The working principle of these liners involves the dissipation of the incident acoustic energy into heat, especially at the Helmholtz resonance frequency. These liners are applied on the internal walls on the engine, both in the intake and by-pass ducts.

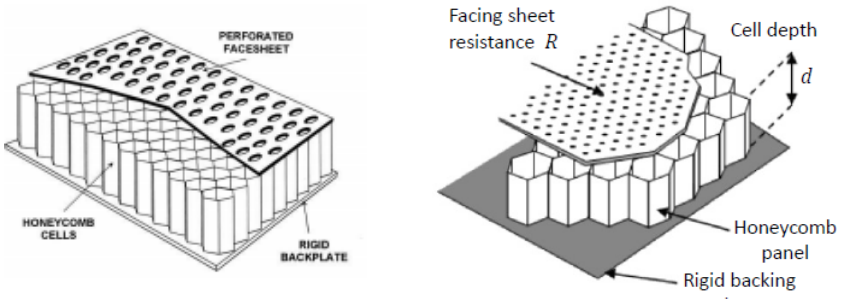


Figure 2.4: (a) A SDOF liner with a perforated face-sheet, (b) DDOF liner with a perforated or linear face and septum [53], [72].

A liner is typically a sandwich panel made by:

1. **A face-sheet**: It is a perforated or a wire mesh type structure. The different types of face-sheets can be seen in Figure 2.4. For double layer liners, different combinations of the face-sheet can be used.

2. **A honeycomb structure**: This is the structure that corresponds to the core of the sound absorbing material and it can be assimilated to a porous material. Depending on the number of honeycomb layers the liners can be classified as Single Degree Of Freedom (SDOF) or Double Degree Of Freedom (DDOF).

3. **A back-sheet**: This structure corresponds to the backing of the acoustic treatment. It is assumed to be rigid.

For some simplified models, the acoustic response of the material is dependent only on the local acoustic pressure and not on the structure of the acoustic field in the duct. Thus there is no sound propagation in the direction parallel to the surface of the acoustic treatment. These are referred to as locally reacting models [94]. Typically an impedance boundary condition is used to predict the acoustic response of these simplified models. The behavior of acoustic liners can be approximated using such impedance models. As an example, the non-dimensional impedance z of the SDOF liner is

expressed as follows [71]:

$$z = R + \mathrm{i}\chi = R + \mathrm{i}(kl - \cot(kd)), \tag{2.13}$$

where R is the non-dimensional facing sheet resistance, d is the depth of the honeycomb layer and l is the mass inertance of the facing sheet. The mass inertance is dependent on the thickness, porosity and the diameter of the holes in the facing sheet.

These simplified models are not considered in the present work. Instead, we need to have a model for the honeycomb structure which can be assimilated to a porous material or sound absorbing materials. Nevertheless, the theory and techniques developed for such sound absorbing materials could also be easily applied to impedance models specifically acoustic liners.

2.4.2 Models for sound absorbing materials

There are several prediction models used to model the physics of sound absorbing materials [22], [78]. Typically, they are grouped into two main categories. The first one corresponds to models based on the Biot theory [13] in which the wave propagates in both fluid and solid parts of the porous material. These models describe the complete vibro-acoustics behavior in porous media. The counterpart is that they are computationally demanding and require a set of parameters to be used in the solid and fluid phases which are not always easy to determine. The second class of models is the equivalent fluid models where the skeleton of the porous is assumed to be motionless. For such models, no wave propagates in the solid phase thereby simplifying the vibro-acoustic behavior. As for the structures of interest, the material is quite stiff, this second category will be considered in this manuscript.

The modeling of foams can be typically done with such kind of equivalent fluid models. Foams typically exhibit low density, high surface area and are relatively easy to manufacture which makes them a good choice for sound absorbing materials. The details and classification of different types of foams are presented in [15]. The term equivalent fluid lies on the idea that the porous material is modeled as a fluid with an equivalent density $\tilde{\rho}_{eq}$ and a complex compressibility $\tilde{K}_{eq}$. Hence, the wave number in the Helmholtz equation (2.1) is rewritten $k = \omega\sqrt{\tilde{\rho}_{eq}/\tilde{K}_{eq}}$. The parameters are complex (to take dissipation into account) and frequency dependent (to take dispersion into account). Many different prediction models have been proposed to get an estimate of these equivalent fluid properties [69],[50]. These models can be grouped into three different categories, namely empirical, analytical, semi-phenomenological depending on the complexity required (see for instance Figure 2.5).

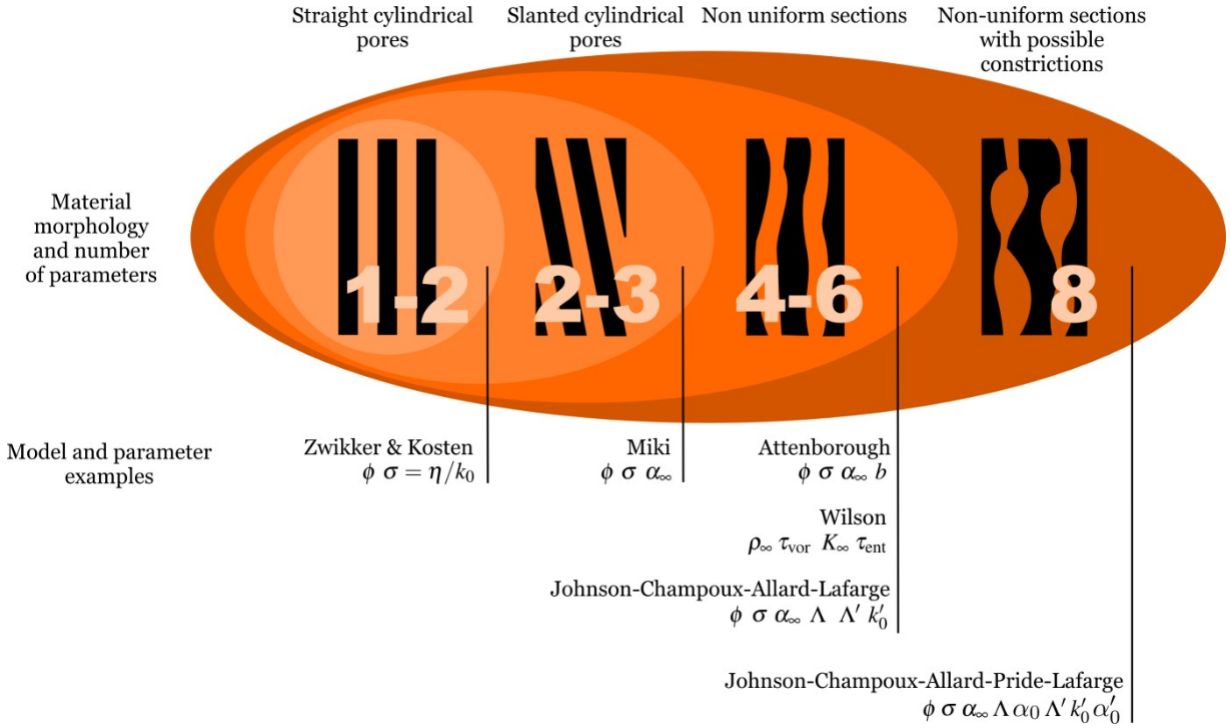


Figure 2.5: Wave propagation in motionless skeleton porous materials [15].

Delany and Bazley devised an empirical approach to predict the specific impedance and acoustic wavenumber. This empirical model was set up to match with experimental data [22] and was enhanced by Miki [69]. However, these models describe dissipation as a whole and do not rely on a modeling of the microstructural physics, thereby limiting the prediction accuracy. To solve this issue, phenomenological models are established, which take into account the viscous and thermal dissipation mechanisms. The most commonly used semi-phenomenological model is the Johnson-Champoux-Allard (JCA) model. This model is used to describe the visco-inertial and thermal dissipative effects for sound propagation in slanted cylindrical pores or generalized nonuniform cross sections [50]. This is the model that will be considered in this work.

The JCA model provides expressions for $\tilde{\rho}_{eq}$ and $\tilde{K}_{eq}$ which depend on 5 intrinsic parameters that are now presented. They are:

- Porosity ϕ: It is the ratio of the volume of the pores to the total volume in the porous frame. Its typical values are higher than 0.98 so that sound absorbing materials are quite light. This is very important for applications but its exact value is not crucial as its range of variation is small.

- Tortuosity α_∞: It is used to define the complexity that exists in the flow path in the porous

material. One way to measure tortuosity is to compare the ratio of the electric resistance of the saturated sample with an electrically conducting fluid R_s, and that of the saturating fluid, R_f. The tortuosity can then be expressed as:

$$\alpha_\infty = \phi \frac{R_s}{R_f}. \tag{2.14}$$

- Flow resistivity σ: The flow resistivity is determined by measuring the pressure drop Δp induced by a sample of porous material with a thickness of d in a steady flow of U. It is defined as:

$$\sigma = \frac{\Delta p}{Ud}. \tag{2.15}$$

The flow resistivity is the major factor for sound absorption as its value can vary a lot (between 10^3 and 5×10^5 Pa.s.m^{-2}).

- Viscous characteristic length Λ: This parameter is used to quantify the importance of viscous effects in a porous medium at high frequencies.

$$\Lambda = 2 \frac{\int v_{\text{inviscid}}^2 \, \mathrm{d}V}{\int v_{\text{s}}^2 \, \mathrm{d}S}. \tag{2.16}$$

where, v_{inviscid} is the velocity of the fluid inside the pores, v_{s} is the velocity of the fluid on the pore surface in the absence of viscosity. The integration in the numerator is performed over the volume V of the pore and the integration in the denominator is performed over the pore surfaces S in the elementary representative volume. It is also a good approximation of the size of the small pores in the network.

- Thermal characteristic length Λ': This parameter is used to describe the thermal effects in the porous medium at high frequencies. This parameter can be determined by using the following relation:

$$\Lambda' = 2 \frac{\int_V \mathrm{d}V}{\int_S \mathrm{d}S}. \tag{2.17}$$

The complex and frequency dependent estimates for the equivalent density and equivalent bulk modulus are:

$$\tilde{\rho}_{eq}(\omega) = \frac{\alpha_\infty \rho_0}{\phi} \left[1 + \frac{\sigma \phi}{\mathrm{i}\omega \rho_0 \alpha_\infty} \sqrt{1 + \mathrm{i} \frac{4 \alpha_\infty^2 \eta \rho_0 \omega}{\sigma^2 \Lambda^2 \phi^2}} \right],$$

$$\tilde{K}_{eq}(\omega) = \frac{\gamma p_0 / \phi}{\gamma - (\gamma - 1) \left[1 - \mathrm{i} \frac{8\eta}{\Lambda'^2 N_{pr} \rho_0 \omega} \sqrt{1 + \mathrm{i} \frac{\Lambda'^2 N_{pr} \rho_0 \omega}{16 \eta}} \right]^{-1}},$$

where, N_{pr} is the Prandtl number, ρ_0 is the density of air, γ is the ratio of specific heats, p_0 is the atmospheric pressure and η is the dynamic viscosity of air. Future advancements of the JCA model are the Johnson-Champoux-Allard-Lafarge and Johnson-Champoux-Allard-Pride-Lafarge model. These models require non-acoustic parameters that are more challenging to estimate [74].

2.5 Design of acoustic liners: an overview

In this section, a brief overview of the optimization methods for the design of acoustic liners is presented. Although the state-of-the-art models presented in this section solely focus on modeling liners using simple impedance models, the same modeling issues persist for acoustic treatments as the ones described in Section 2.4.2.

The objective of this part is to present the state-of-the-art optimization workflow for modeling acoustic treatments described in Section 2.4 and emphasize that such optimization workflows require the resolution of a huge number of problems that generally do not differ too much. The most critical application is the use of novel liner designs to mitigate noise propagation at the inlet of turbofan engines. The computational cost of a single configuration is prohibitively large so that it can take up to several days to identify an optimal liner configuration. A configuration corresponds to a given problem (geometry of the problem, materials ...) and the optimization process generally requires, in its calculation part, the prediction of the acoustic response of a huge number of configurations.

To model such problems, most of the time classical finite element based numerical discretization is used and it demands fine mesh resolutions at high frequencies [81, 55]. This typically results in high computational cost which is made affordable using high frequency approximations (HFA). The HFA assumes that at high frequencies, especially when a large number of modes are incident at the source, the sound field behaves like a diffused field. This allows us to perform equivalent simulations at low frequencies, whereas the desired frequency of interest is used in the impedance model [72]. These approximations however do not aim to tackle the number of resolutions that need to be evaluated, which results in undesirable computational effort.

One way to reduce these costs is to use automated optimization strategies. Different optimization strategies have been explored for liner optimization depending on the liner type, multi-modal sources, range of frequencies, complexity of the geometry, etc. The work done by Lafronza [55] uses an optimization procedure based on the design of experiments (DOE) methods to build a response surface model (RSM). This surrogate model helps to reduce the computational effort by discarding regions of poor liner designs.

Typically, for aerospace applications, the engine noise has to pass certain noise certifications that require multiple objective functions. Copiello and Ferrante [20] used multi-objective optimization for zero-splice liners in cylindrical ducts to achieve different noise certifications at the same time.

An overview and comparison of various strategies to model acoustic liners for turbofan intakes are described in the thesis of Achunche [2][Chapter 3]. It was concluded that for practical problems that

involve several modes, using an initial exploration of the design space followed by a local minimizer seems to be a robust approach. This avoids getting trapped into a local minimum of the optimization problem. The combination of using different strategies to speed up the calculation time is typically referred to as hybrid optimization strategies. In the work by P. Mustafi [72], a hybrid optimization strategy was implemented to optimize the liner resistance R and the cell depth d. It consists of combining a Genetic Algorithm (GA) to explore the design space followed by a gradient based Dynamic Hill Climbing (DHC) algorithm to identify the most ideal configuration. The results are cross verified with a GA followed by 'fmincon' using the Matlab optimization toolbox [68].

As an illustration, the contour plots for the liner resistance R and cell depth d for a range of frequencies from 0.25 to 5 kHz at approach condition can be seen in Figure 2.6. Each blue dot corresponds to one configuration and for each one of them, the complete computational model needs to be solved making the process computationally very expensive. It is reported that these optimization calculations performed over a range of frequencies took approximately 5 days and several hundreds of evaluations. [72].

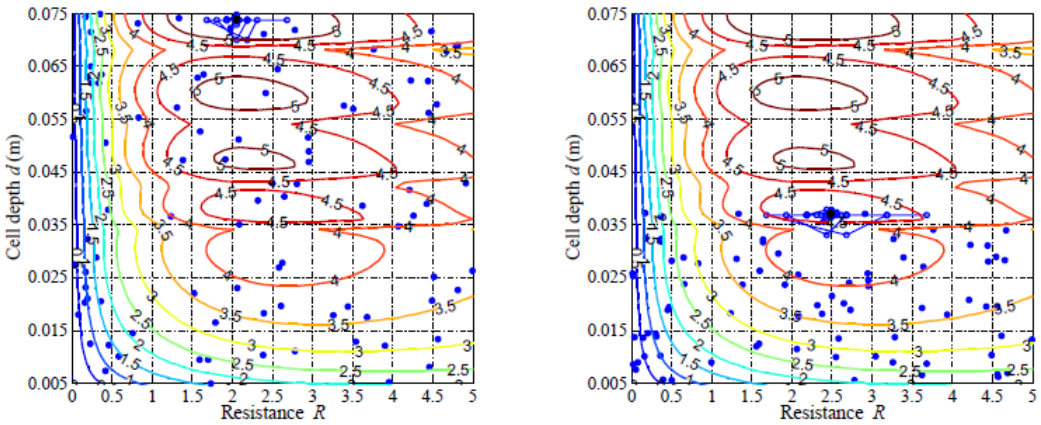


Figure 2.6: Optimization results of ΔPowerlevel ($PWL_{40^\circ \to 90^\circ}$) integrated over a range of frequencies from 0.25 - 5 kHz at approach condition, characterized by liner resistance R and cell depth d. This figure is taken from the work of P. Mustafi [72] [Chapter 5].

The typical workflow to obtain contour plots of Figure 2.6 is now summarised. The liner resistance R and the cell depth d are chosen as design parameters for optimization. The cost function or the objective function which is typically the quantity needs to be minimized. This function is evaluated for the chosen liner configuration. In the present work, the transmission loss (TL) is used as a cost function described by the following relation,

$$\mathrm{TL} = 10 \log_{10} \frac{P_{in}}{P_{out}} \ \mathrm{dB}, \tag{2.18}$$

where P_{in} and P_{out} are the acoustic powers at the inlet and the outlet respectively. The computa-

tion of the TL requires the numerical evaluation of normal pressure gradients. This leads to a slight reduction in numerical accuracy due to the derivation of shape functions. The acoustic power at the inlet can be computed analytically (for simplified problems) for a given frequency, mode, and intake geometry.

Any optimization strategy can be used to get the optimal liner properties but all of them require to compute several configurations. The method to compute a single one is now outlined as seen in Figure 2.7 .

In the first step, an acoustic envelope surrounding the turbofan engine is generated using a given CAD model. This envelope is meshed and discretized using finite element triangulation or any other numerical discretization described in the literature. Depending on the chosen frequency and the incident mode, the discretized system can consist of millions of degrees of freedom. The next step involves the computation of the LU factors of the matrix of the problem. The assembly and factorization of these large systems contribute to the majority of computational costs. The pressure field is recovered using forward-backward substitutions. As the last step, the cost function is computed. This process is repeated for each of the configurations (blue dot of Figure 2.6) as outlined in Figure 2.7.

This is the bottleneck in the optimization process since the number of configurations is extremely high [81]. In conclusion, in parallel to the research works that can be done on optimization algorithms, there is a need to find methods to solve quickly configurations especially when they do not differ too much.

In the next section, we demonstrate some typical calculations done using the aforementioned workflow.

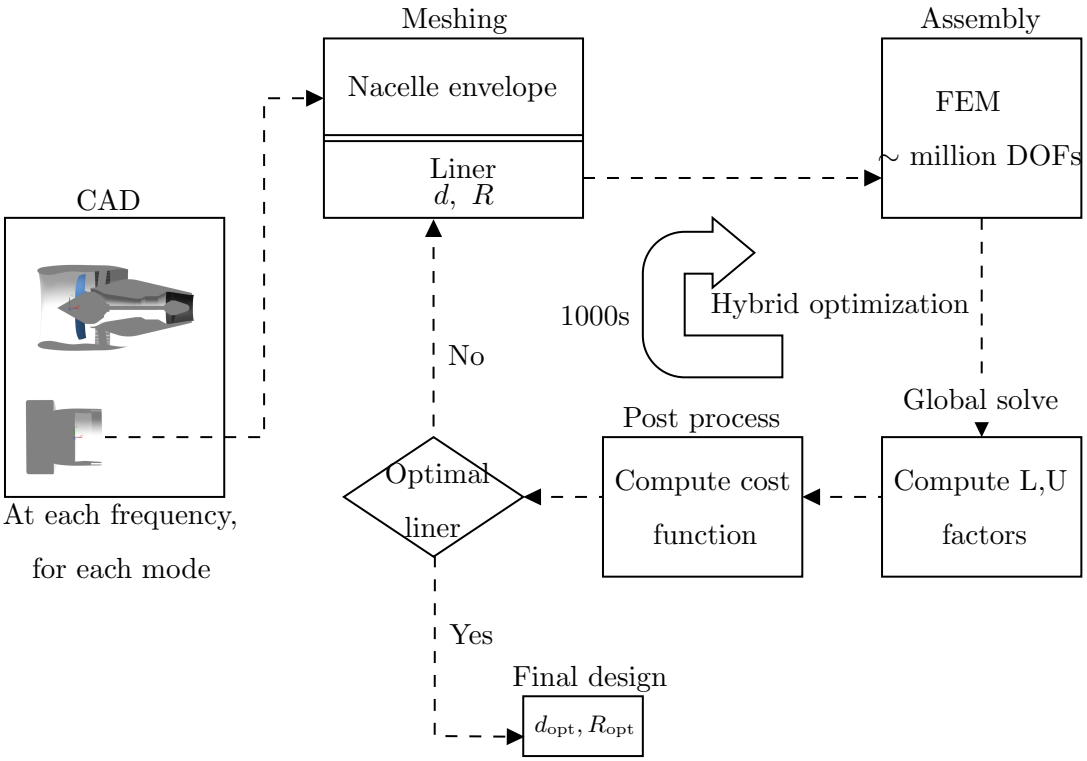


Figure 2.7: A typical optimization framework for modeling acoustic treatments in the design phase for turbofan intake which typically requires high computational resources.

2.6 Examples

The objective of this section is to present the typology of problems that are considered in this work. In a first stage, an axi-symmetric model is presented then a 3D model is implemented and cross-verified with the axi-symmetric case. Finally, contour plots for the design space are presented for the 3D model.

2.6.1 Axi-symmetric model

At first, an axisymmetric model is presented. This model is based on a 2D geometry but its physics is 3D so that it is referred to as a 2.5D model.

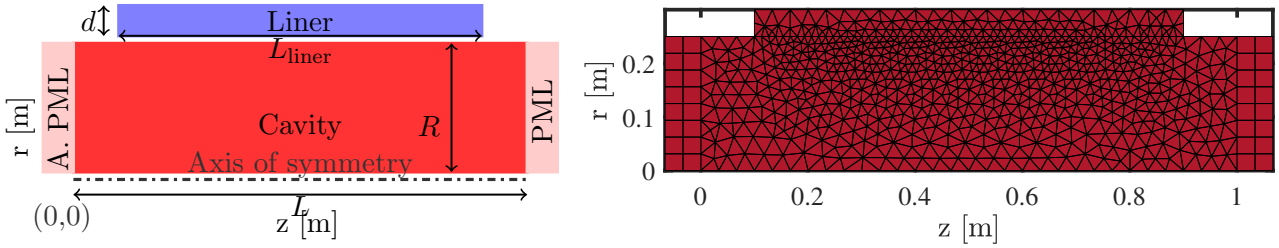


Figure 2.8: (a) Computational domain with an active PML at the inlet (left) and a PML at the outlet (right), (b) Discretization of the computational domain using p-FEM with mesh refinement between the duct and liner.

The geometry of the axi-symmetric case is shown in Figure 2.8 (a), with the main dimensions as follows:

1. Duct length, $L = 1$m
2. Duct radius, $R = 0.25$m
3. Liner length, $L_{\text{liner}} = 0.8$m
4. Liner depth, d between 5 and 10 cm and flow resistivity σ between 0.2 and $2{\times}10^4$ Nsm^{-4}

The computational domain is discretized using triangular elements (generated using Gmsh). It is shown in Figure 2.8 (b). It is composed of a cavity filled by air and a liner filled by a porous material. The JCA model is used to represent the melamine foam, details are given in Section 2.4.2. On the left and right-hand side of the computational domain, Perfectly Matched Layers (PML) are used to avoid spurious reflections from the domain [11]. The left-hand side PML is also active, in the sense that it generates a sound field entering the domain, in the form of duct modes (see for instance [36]). A choice of using 2 layers of PML suffices to be a good compromise between the computational effort and accuracy for the numerical model. The matrices are assembled using an in-house p-FEM code [10].

In all the following results, the design parameters of the liner are mentioned in pairs between braces like (d, σ). Concerning the excitation, only radial mode can be excited in this model and an excitation mode will be referred to as its index.

The numerical results are now examined in the presence and absence of acoustic treatment for higher order modes and also for plane waves at varying liner depths.

We firs demonstrate the influence of the acoustic treatment. The excitation corresponds to mode (4) at 7 kHz (6.47 kHz cut on frequency). It is injected in the presence and absence of an acoustic treatment. As seen in Figure 2.9 (b), the liner almost completely attenuates the wave propagation as

the amplitude of the pressure waves at the outlet is almost negligible. In some of these configurations, the liner provides an attenuation up to 34 dB.

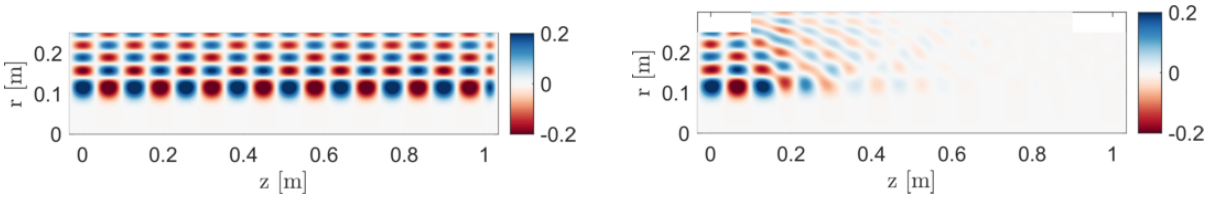


Figure 2.9: Numerical solutions of the pressure field, mode (4) at 7 kHz, (a) Hard wall duct with the cut on frequency of 6.47 kHz, (b) Lined duct (0.1, 2×10^3) providing an attenuation of $\sim$ 34 dB.

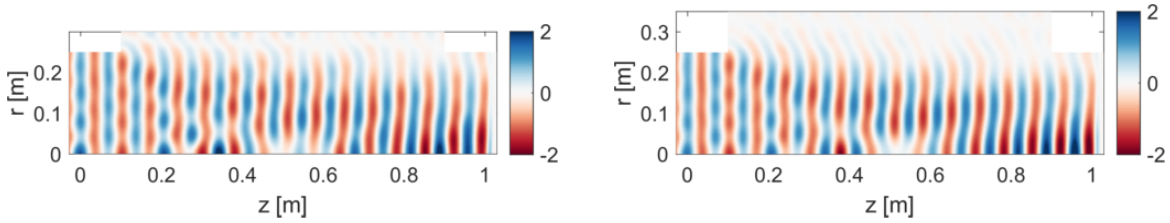


Figure 2.10: Numerical solutions of the pressure field of a plane wave at 5 kHz, (a) TL = 1.3 dB for the liner configuration (0.05, 2 $\times10^3$), (b) TL = 1.5 dB for the liner configuration (0.1, 2 $\times10^3$).

We now showcase some interpretation of the acoustic response for varying liner parameters. In general the plane wave (mode index = 0) is the most challenging to attenuate since the direction of propagation in the duct is parallel to the surface of the liner. It is therefore interesting to examine the attenuation of the plane wave, especially at high frequencies. A plane wave with unit pressure amplitude is considered at 5 kHz and different liner configurations are used to examine their performance. As seen in Figure 2.10 (a), a wave propagates in the bulk of the liner and is then scattered back by the right end of the liner. As expected, the wave enters more strongly into the liner for small values of the flow resistivity ($\phi = 2 \times 10^3$). In Figure 2.10, a higher transmission loss is observed for the configuration on the right (1.5 dB) compared to the one on the left (1.3 dB). For small liner depths, the scattered wave in the liner may reflect back into the cavity which may perhaps propagates in the downstream direction. This explains a slightly lower transmission loss for smaller liner depths. As the liner depth increases, the scattered wave in the liner has minimal reflected energy to propagate into the cavity. However, increasing the flow resistivity to very high values provides a very high resistance for the wave to enter the liner which leads to an inefficient configuration. There is therefore a balance between increasing the liner depth and the flow resistivity. In general this behavior might change depending on the angular frequency or the equivalent fluid model which is beyond the scope of this discussion.

2.6.2 Implementation of the 3D model

A full 3D model has also been implemented. It is presented in this section and cross-verified with the axi-symmetric case. The geometry in Figure 2.8 (a) is revolved around the axis of symmetry to form the 3D computational model. A conformal quadratic mesh is chosen so as to incorporate the curvature of the surfaces. As seen in Figure 2.11, the discretized mesh of the cavity with and without treatment is displayed. For this case with $p = 5$, one PML layer on each end of the duct is sufficient to avoid spurious reflections back into the computational domain.

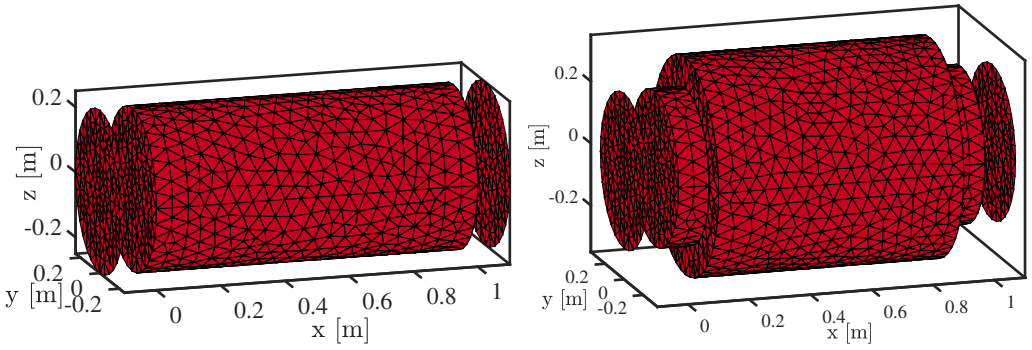


Figure 2.11: (a) The 3D model of the axi-symmetric case revolved around the axis of symmetry without acoustic treatment, (b) 3D model of the axi-symmetric case with acoustic treatment.

The numerical results are presented for two different cases: at $f = 1.5\,\text{kHz}$ for the first radial mode in Figure 2.12 (a), and $f = 2.2\,\text{kHz}$ for the second radial mode and first azimuthal mode in Figure 2.12 (b). The complete 3D model involves approximately 80k degrees of freedom, wherein the matrices are assembled using p-FEM outlined in Section 2.2. A direct solver is used to compute the pressure field, followed by the calculation of the transmission loss (TL).

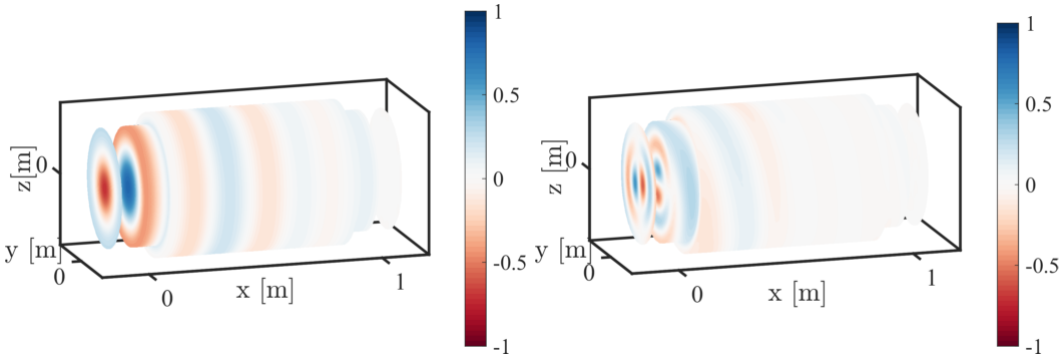


Figure 2.12: (a) $f = 1.5\,\text{kHz}$ for mode (1), (b) $f = 2.2\,\text{kHz}$ for mode (2).

The results of the transmission loss for different frequencies and duct modes are presented in Table 2.1 for the axi-symmetric and the 3D case. For increasing frequencies and duct modes, the absolute error also increases but not beyond 0.3 dB. It can thus be said that the cross-verification of

the 3D model demonstrates good accuracy compared to the axi-symmetric case.

Table 2.1: Transmission loss for $d = 0.05\,\mathrm{m}$ and $\sigma = 1.57 \times 10^4$ for varying frequencies and duct modes.

f in Hz	300	500	1500	2000	1000	1300	1500
(m, n)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)	(0,1)	(0,1)
TL (in dB): 2.5D case	4.13	9.76	4.87	2.63	9.53	8.28	7.75
TL (in dB): 3D case	4.13	9.76	4.78	2.53	9.79	8.23	7.76

Since the 3D model is cross-verified, it will be considered in the next section to scan the design space for a series of calculations.

2.6.3 Scan of the design space

In this subsection, the results for contour plots of the transmission loss calculated with the 3D model are presented for varying thickness and flow resistivity which are the design parameters of the present problem. The liner depth d is varied from $0.03\,\mathrm{m}$ to $0.08\,\mathrm{m}$. The flow resistivity σ is varied from 2×10^3 to $2 \times 10^4\,\mathrm{N\,s\,m^{-4}}$.

For each incident duct mode, the design variables d and ϕ are discretized using 25 and 5 equidistant points, respectively. Thus a scan of the design space involve 125 different liner configurations.

The results for varying mode orders at $f = 2$ kHz are presented in Figure 2.13. It can be seen that the attenuation of the plane wave is relatively small compared to the higher-order modes. In addition, increasing the liner depth for a plane wave reduces the transmission loss whereas the opposite effect is observed for higher-order modes. In general the choice of a small flow resistivity and liner depth appears to be a reasonable option for the attenuation of the plane wave. However for higher order modes, a large liner depth with relatively low flow resistivity seems to provide a maximum attenuation. Overall, the choice of a high flow resistivity does not seem to be a reasonable option for all types of modes.

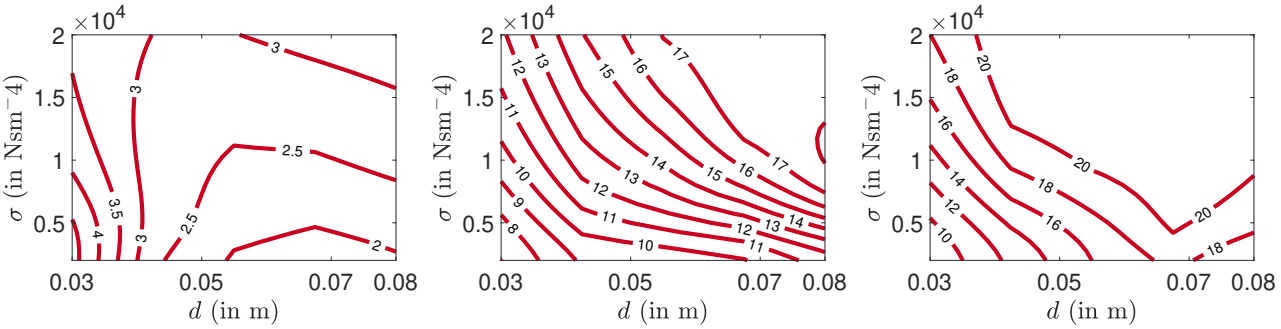


Figure 2.13: Contour plots for the transmission loss (in dB) plotted for varying liner depths and flow resistivity at 2 kHz. (a): Mode (0,0), (b): Mode (0,2), (c): Mode (1,2).

The results at $f = 2.5$ kHz for the same mode orders are presented in 2.14. An important remark is that as the frequency increases the TL reduces for the same mode orders. The conclusions drawn for the plane wave and higher order modes at 2 kHz remain the same for 2.5 kHz.

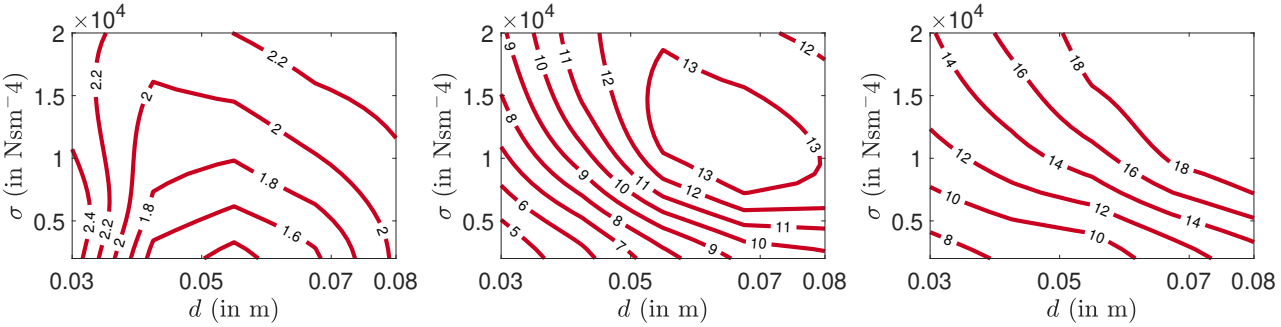


Figure 2.14: Contour plots for the transmission loss (in dB) plotted for varying liner depths and flow resistivity at 2.5 kHz. (a): Mode (0,0), (b): Mode (0,2), (c): Mode (1,2).

The critical issue of these models is the high computational cost of running a large number of evaluations. For each of the 125 calculations, the complete FE model needs to be resolved. This is not feasible in the early design phase for high resolution models. In the forthcoming Chapters, domain decomposition methods will be used to tackle these short-comings.

2.7 Conclusions

This chapter presented the models and methods that will be used in the remainder of the thesis. It also presents some first numerical results as illustration. As a conclusion, modeling of large-scale optimization problems requires multiple runs of the computational model for configurations which are quite similar. The critical issue is that the full FE models needs to be resolved for all the calculations to get optimal liner configuration. The forthcoming chapters are devoted to the presentation of original methods to compute the results in a more efficient way.

Chapter 3

Domain decomposition methods for the Helmholtz equation

The objective of this chapter is to present an overview of the performance of various Domain Decomposition Methods (DDM) for solving large-scale Helmholtz problems. This chapter is devoted to acoustic models in air so that there is no porous material involved (It will be the objective of the next chapter). The objective of DDM is to combine the advantages of direct and iterative resolution methods. Indeed, direct solvers typically do not scale well and require the factorization of large linear systems. On the other side, iterative solvers scale well, however the convergence of classical or Krylov based iterative schemes is difficult (see for instance Section 2.2). Domain Decomposition methods combine an iterative resolution of a so-called interface problem with a factorization of subdomain problems using a direct solver. In the first part of this chapter, a detailed literature review of the various DDM is undertaken. A performance analysis of two of these methods, namely FETI-2LM and FETI-H, is presented to determine the most optimal for acoustical problems.

3.1 Domain decomposition methods for wave problems

Domain decomposition methods serve as an alternative to combine the robustness of direct methods with the scalability of iterative ones. It consists of dividing the original problem into a set of smaller subdomains. In this section, a literature review of various domain decomposition methods applied to time-harmonic wave propagation problems is elaborated. Before getting into the literature review, the author would like to recall the definitions of strong and weak scalability which will be used throughout this document. An algorithm is strongly scalable when for a fixed problem size, the computational cost scales linearly with the number of processors. As an example, one should be able to solve a

problem 10 times faster using 10 processors compared to a single processor. Typically an algorithm is said to be weakly scalable when the computational cost scales linearly with increasing problem size, for the same number of processors.

Domain Decomposition Methods (DDM) for wave propagation problems have been studied extensively during the past three decades. They can be classified into three different parts namely classical Schwarz, optimized Schwarz and Schur methods.

3.1.1 Classical Schwarz methods

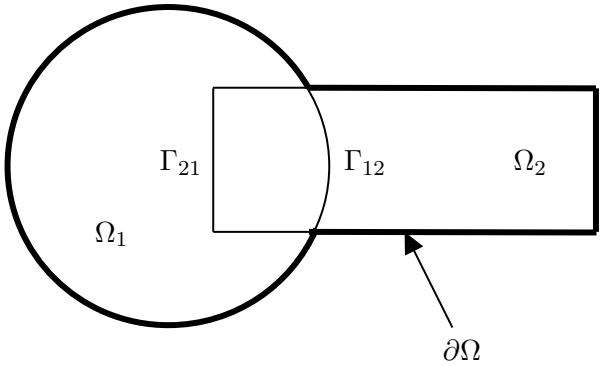


Figure 3.1: Schwarz introduced the first domain decomposition method consisting of a complicated domain, split into two simple ones, a disk and a rectangle.

These techniques are the oldest domain decomposition methods invented by Hermann Schwarz in 1869 to prove results obtained by Riemann based on a minimization principle [86]. There was a renewed interest in these methods with the advent of parallel computational tools. This resulted in subdivisions of these algorithms into two parts (at the continuous level), the alternating Schwarz [86] and the parallel Schwarz [62, 63].

The alternating Schwarz algorithm was formed to find a rigorous proof of the Dirichlet principle on arbitrary domains. The Dirichlet principle is the assumption that the minimizer of a certain energy functional is the solution to the Poisson problem which can be defined as follows:

$$\Delta u = 0 \ \text{ in } \ \Omega_i, \quad u = g \ \text{ on } \partial\Omega, \tag{3.1}$$

where Ω_1, Ω_2 are the disk and rectangular domains which have an overlap as seen in Figure 3.1. Their union is the full domain Ω. $\Gamma_{12} := \partial\Omega \cap \Omega_1, \Gamma_{21} := \partial\Omega \cap \Omega_2$ are the physical interfaces, g can be some arbitrary imposed boundary condition. The alternating Schwarz method is initiated by computing the partial solutions in each domain i.e. either the disk or rectangle with an random initial starting guess , $u_2^0 = 0$ along Γ_{12}. Once u_2^n at $n = 0$ is known, the problem in subdomain, Ω_1 is well posed and the solution in this domain can be computed. In the next step, the solution u_1^{n+1} is restricted

on the interface Γ_{21} which defines the boundary value problem in Ω_2. The solutions are computed sequentially until the desired accuracy is required. The algorithm can be described as follows:

$$\begin{aligned} \Delta u_1^{n+1} &= 0 \ \text{ in } \ \Omega_1, & \Delta u_2^{n+1} &= 0 \ \text{ in } \Omega_2, \\ u_1^{n+1} &= u_2^n \ \text{ on } \Gamma_{12}. & u_2^{n+1} &= u_1^{n+1} \ \text{ on } \Gamma_{21}. \end{aligned} \tag{3.2}$$

However the alternating Schwarz algorithm is not suitable for parallel computing which led to the new class of parallel Schwarz methods [62]. The idea is to solve the problem in each partial domain in parallel and to update the solution at the end of each solve. The resulting algorithm can be described as follows:

$$\begin{aligned} \Delta u_1^{n+1} &= 0 \ \text{ in } \Omega_1, & \Delta u_2^{n+1} &= 0 \ \text{ in } \Omega_2, \\ u_1^{n+1} &= u_2^n \ \text{ in } \Gamma_{12}. & u_2^{n+1} &= u_1^n \ \text{ in } \Gamma_{21}. \end{aligned} \tag{3.3}$$

The only difference in the algorithm arises in the fact that all the updated solutions on the interfaces are obtained from the previous iterate n and not $n+1$. However the extension of this algorithm to multiple subdomains is not straightforward especially when more than 3 subdomains have a common overlapping region. This requires sequential definition of the interfaces in order to avoid overlapping interfaces [45].

These methods were described at the discrete level in latter years. However they are not necessarily equivalent. The multiplicative Schwarz at the discrete level [45] is equivalent to the alternating Schwarz method. However the additive Schwarz at the discrete level [92] is not equivalent to the parallel Schwarz method. However the restrictive additive Schwarz [14] is equivalent to the parallel Schwarz method. The convergence of the Schwarz algorithm for elliptic problems was proved in the case of overlapping subdomains [7]. In fact the convergence rate is a function of overlap, i.e. the bigger the overlap the faster the convergence [26][Sec 1.5]. In addition, these algorithms lead to reduced memory requirements. Over time, it became crucial to extend these methods to non-overlapping subdomains. The motivation was to tackle multi-physics problems where an interface exists naturally based on the governing physics in each subdomain. However the extension of these methods to non-overlapping subdomains specifically to time-harmonic wave propagation problems lead to several issues.

Firstly, the convergence was not guaranteed as the transmission conditions only enforce the continuity of the solution and their fluxes across the interface.

Secondly, the classical overlapping Schwarz method do not converge for the Helmholtz equation as

seen below:

$$\begin{aligned} -k^2u_1^{n+1} - \Delta u_1^{n+1} = 0 \ \text{ in } \ \Omega_1, \qquad & -k^2u_1^{n+1} - \Delta u_2^{n+1} = 0 \ \text{ in } \Omega_2, \\ u_2^{n+1} = u_2^n \ \text{ on } \Gamma_{12}, \qquad & u_2^{n+1} = u_1^{n+1} \ \text{ on } \Gamma_{21}, \\ u_1 = 0 \quad \text{ on } \ \partial\Omega, \qquad & u_2 = 0 \quad \text{ on } \ \partial\Omega. \end{aligned} \tag{3.4}$$

For the two subdomain problems, if k^2 is an eigenvalue of the Laplace problem in subdomain $i = 1, 2$, its associated eigen-function v satisfies:

$$\begin{cases} -\Delta v &= k^2 v \quad \text{ in } \ \Omega_i, \\ v &= 0 \quad \text{ on } \ \partial\Omega_i. \end{cases} \tag{3.5}$$

The subdomain problems defined in Equation (3.4) are then not necessarily well-posed as any function of the type $u_i^e + \beta v$, where β is a constant, could be a possible solution [26][Sec 2.2.1]. Thus, it is possible that the local subdomains could have multiple solutions which could possibly result in convergence issues. However the global problem has a unique solution and thus the use of solely Dirichlet and Neumann type continuity conditions is not suitable for these type of problems. In addition, the extension of the alternating Schwarz method as a solver and later as a preconditioner for GMRES did not converge for an indefinite Helmholtz equation [29]. A detailed analysis was performed to see how each Fourier mode converges and it was observed that for the classical alternating Schwarz case only the high-frequency components of the error corresponding to evanescent modes converge, whereas the low-frequency components of the error corresponding to the propagative modes do not converge [40]. Thirdly, the convergence speed of the classical Schwarz method is not suitable for practical problems involving large numbers of subdomains [45]. These algorithms do not scale with increasing subdomains. In order to circumvent these issues, a new class of Schwarz methods with more general transmission conditions was required. This led to the foundation of the optimized Schwarz methods.

3.1.2 Optimized Schwarz methods

The optimized Schwarz methods (OSM) consists of formulating generalized and optimal transmission conditions [40, 39, 65]. In order to circumvent the issues associated with the classical Schwarz methods, the motivation for this work lies in the following points:

1. Prove convergence for non-overlapping subdomains
2. Improve the convergence rate
3. Extend these methods to different applications and not just Laplace problems

The focus of these methods relies on using mixed or Robin boundary conditions. Lions [63] focused on using these type of boundary conditions in the alternating Schwarz algorithm in the following way:

$$\begin{cases} \Delta u_1^{n+1} = f \quad \text{in } \Omega_1, \\ (\partial_{n_1} + \alpha_1) u_1^{n+1} = (\partial_{n_1} + \alpha_1) u_2^n \quad \text{on } \Gamma_{12}, \end{cases} \qquad \begin{cases} \Delta u_2^{n+1} = f \quad \text{in } \Omega_2, \\ (\partial_{n_2} + \alpha_2) u_2^{n+1} = (\partial_{n_2} + \alpha_2) u_1^{n+1} \quad \text{on } \Gamma_{21}, \end{cases} \tag{3.6}$$

where α_1, α_2 are constant parameters. The convergence for arbitrary non-overlapping subdomains was proved by Lions using energy estimates [64]. Després was the first to use these mixed types of boundary conditions for Helmholtz equation and prove the convergence [23, 7]. This algorithm for arbitrary subdomains can be written as follows:

$$\begin{cases} -(\Delta u_i + k^2 u_i) = 0 \quad \text{in } \Omega_i, \\ \nabla u_i \cdot \mathbf{n}_i + \mathrm{i}k u_i = \nabla u_j \cdot \mathbf{n}_i + \mathrm{i}k u_j \quad \text{on } \Gamma_{ij}, \forall j \in \mathcal{N}_i, \end{cases} \tag{3.7}$$

where $\mathcal{N}_i$ is vector of indices that share a boundary with Ω_i (such that $\partial\Omega_j \cap \partial\Omega_i \neq \emptyset$). In the case when k^2 is an eigenvalue of the Laplace operator with a Dirichlet boundary condition in a particular subdomain i, the solution is still unique and the problem is well posed. The introduction of these conditions was a key development for all different types of coupling conditions for acoustical problems. It is possible to use better transmission conditions as indicated by Lions [64] and Hagström [47]. This led to the design of optimized Schwarz methods [66, 67, 19]. The idea it to replace the Robin type boundary condition with more general boundary conditions as follows:

$$\mu_i \boldsymbol{\nabla} u_i^{n+1} \cdot \mathbf{n}_i + \mathcal{B}_{ij}(u_i^{n+1}) = \mu_i \boldsymbol{\nabla} u_j^n \cdot \mathbf{n}_i + \mathcal{B}_{ij}(u_j^n) \quad \text{on } \Gamma_{ij}, \forall j \in \mathcal{N}_i, \tag{3.8}$$

where μ_i is a real-valued constant and $\mathcal{B}_{ij}$ act as absorbing operators on the interface Γ_{ij}. The choice of these parameters can in general describe all types of Schwarz algorithms in the literature. For example, choosing $\mu_i = 0$ and $\mathcal{B}_{ij} = 1$ is the classical alternating Schwarz algorithm (without overlap). The case when $\mu_i = 1$ and $\mathcal{B}_{ij} = \mathrm{i}k$ describes the Després algorithm. The optimized Schwarz methods are based on finding the optimal choice of these parameters for a range of frequencies based on the Fourier transform [39, 25]. An alternative approach is to find the optimal DtN (Dirichlet to Neumann) map for the operator $\mathcal{B}_{ij}$,(μ_i =1). However the optimal DtN maps typically result in non-local transmission conditions which renders the method costly and hard to implement in existing frameworks. Moreover, in a generic problem, the transmission conditions may involve variable and discontinuous physical coefficients across the interface for which the exact DtN map is challenging to determine. A possible alternative is to compute local transmission conditions based on Taylor expansions or

Padé approximations. The use of second-order local transmission conditions requires the computation of the Laplace–Beltrami operator on the physical interfaces. This improves the effectiveness of the transmission conditions without affecting the sparsity pattern generated by the underlying numerical discretization [29].

Apart from improving the transmission conditions, the other issues associated with any non-overlapping sub-structuring strategy are due to cross-points. They typically arise when more than two subdomains intersect at a particular point in 2D domain. These cross-points are a numerical artifact that affects the conditioning and consequently the performance of the method. These issues can be avoided completely if a cell-centered finite-volume method or discontinuous Galerkin method is used. Since the basis is discontinuous each cross-point is an independent degree of freedom and only a weak coupling exists between these cross-points.

Different techniques are proposed to address the issue related to cross-points for a continuous FEM-based approach. In [43] two different approaches are proposed on methods to discretize the Neumann conditions at cross points, namely an auxiliary variable method and a complete communication. which improves the convergence (for arbitrary partitioning) for 2D problems. In [42], best Robin parameters are chosen to accelerate the convergence using GMRES algorithm.

Another aspect is the scalability of the method for increasing frequencies and subdomains. Typically DDM without overlap are not scalable with increasing frequency and subdomains. The use of coarse space is advocated to achieve strongly-scalable algorithm. In theory, the coarse-space corrections comes with additional computations and parallel overhead and thus a perfect strongly-scalable algorithm is very challenging to achieve. It typically consists in using a coarse grid correction at each iteration to damp the slowly varying components of the error [41, 42, 27].

3.1.3 Schur-based methods

This class of domain decomposition methods originated to model large-scale elliptic problems at the algebraic level. The idea is to introduce an additional set of variables on the interface.

$$\begin{cases} \Delta u_1 = 0 \quad \text{in } \Omega_1, \\ u_1 = g \quad \text{on } \partial\Omega_1 \cap \partial\Omega, \\ u_1 = u_2 = \lambda_{\mathrm{p}} \quad \text{on } \Gamma_{12}, \\ \nabla u_1 \cdot n_1 = \nabla u_2 \cdot n_2 \quad \text{on } \Gamma_{12}, \end{cases} \qquad \begin{cases} \Delta u_2 = 0 \quad \text{in } \Omega_2, \\ u_2 = g \quad \text{on } \partial\Omega_2 \cap \partial\Omega, \\ \nabla u_2 \cdot n_2 = \nabla u_1 \cdot n_1 = \lambda_{\mathrm{d}} \quad \text{on } \Gamma_{21}, \\ u_2 = u_1 \quad \text{on } \Gamma_{21}, \end{cases} \tag{3.9}$$

where λ_{p}, λ_{d} are the additional sets of new variables. These methods are referred to as Schur methods because they require the computation of the Schur complement of the subdomain matrices. The Schur

methods are split into two main types: the primal and the dual Schur methods. The primal Schur methods are based on using Lagrange multipliers for the primal variables. The Balanced Domain Decomposition (BDD) belong to these class of Schur methods as seen in Equation (3.9) on the left. This results in the formulation of a condensed interface problem, $\mathbf{F}_{\mathrm{p}}\lambda_{\mathrm{p}} = \mathbf{d}$, using the continuity of normal fluxes across the interfaces. On the other hand, the dual Schur methods, like Finite Element Tearing and Interconnect (FETI), consist in formulating a condensed interface problem for the dual variables, $\mathbf{F}_{\mathrm{d}}\lambda_{\mathrm{d}} = \mathbf{d}$ assuming the continuity of the pressure field across the interface. This can be seen in the algorithm described in Equation (3.9) on the right.

The resulting interface matrices are solved using a Krylov subspace-based iterative scheme to recover the solution in the complete computational domain. The additional advantage of formulating this interface problem is that it improves the conditioning of the interface matrix compared to the original problem. These methods combine both the robustness of the direct solvers and the scalability of the iterative schemes. However as explained in the previous section, if these methods are used to solve the Helmholtz problem, the associated local problems may be ill-posed.

The FETI-2LM method was the first attempt among this class of methods to be applied to solve the indefinite Helmholtz problem [54, 21]. It consists in introducing two separate sets of Lagrange multipliers on the interface to couple the subdomains to form a so-called 'second-level problem'.
There are two main components of the residual error present during the iterative process: (a) the high frequency phenomena on the interface corresponding to the evanescent waves in the subdomains, due to their local support these errors are typically dealt with using a local preconditioner. (b) The low frequency phenomena corresponding to the propagative modes over all subdomains. These residual errors have a global support and are usually dealt with using a coarse grid which impacts the parallel performance. Both these issues are addressed to propose a scalable and robust FETI-2LM algorithm [21]. An improved variant of this algorithm namely FETI-H consists in introducing only 1 Lagrange multiplier on each side of the partitioned interface [33]. This method was compared with the former variant for 2D scattering problems and it was found that the FETI-H method outperformed FETI-2LM especially when a coarse space was used [33]. This will be studied in detail in the latter part of this chapter. A further improvement of the FETI-H variant namely the FETI-DPH consists in keeping these cross points as primal variables and imposing additional constraints on these points [30]. Another way to avoid the issues with the cross-points is to introduce a discontinuous plane-wave basis for the interpolation [35]. The primal Schur-methods like the BDDC-H [57] have been applied to the Helmholtz problem and compared with FETI-DPH. It was found that both methods have similar convergence rates [57].

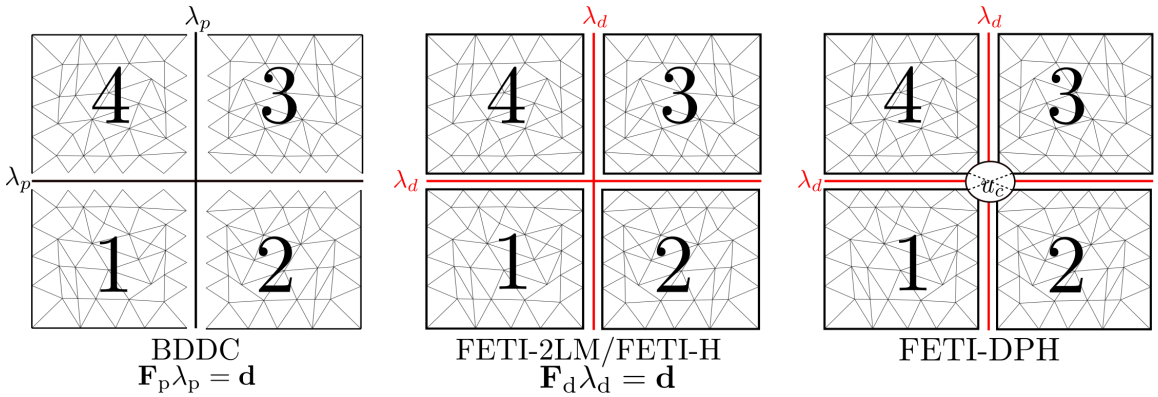


Figure 3.2: The sub-structuring for Schur type of methods, (a) Balanced domain decomposition with constraints for Helhmoltz(BDDC-H), (b) Finite element tearing and interconnect(FETI-2LM/FETI-H), (c) Combination of primal and dual type of sub-structuring FETI-DPH.

The sub-structuring strategy for primal and dual Schur based methods is illustrated in Figure 3.2. The computational domain is split into 4 non-overlapping subdomains. There is a cross-point shared by all four subdomains. In Figure 3.2 (a), it can be seen that the skeleton of the primal variables is used to form the interface problem. There are referred to as primal methods namely BDDC-H. However for the dual sub-structuring algorithms in Figure 3.2 (b), the original skeleton remains undisturbed and an additional set of variables are defined. The FETI methods like FETI-2LM and FETI-H are based on dual sub-structuring strategy. In order to deal with potential cross-points, a dual-primal sub-structuring is adopted as seen in Figure 3.2 (c). The cross-points are preserved to be primal unknowns which helps to improve the conditioning and convergence of the condensed interface problem composed of dual variables.

The dual sub-structuring strategy based on the FETI methods has many advantages. As seen in Figure 3.2 (b), dual variables are introduced which do not affect the original subdomains. As a result, this becomes easier to implement in an existing framework. In addition, the computation of normal pressure gradients on the partitioned interfaces can be avoided. Owing to these reasons, the dual substructuring algorithms are chosen in the present work.

For the dual sub-structuring Schur-based algorithms there are two variants of the FETI based method namely FETI-2LM and FETI-H. The next section describes the theory, formulation of both these FETI variants on a simple 2 subdomain case.

3.2 Presentation of FETI methods

This section describes in detail the FETI-2LM and FETI-H algorithm for solving Helmholtz problems. The same problem as described in Section 2.2 is considered. The initial domain is split into several

smaller non-overlapping domains Ω_i with $i = 1,....,N_s$ where N_s is the number of subdomains. For brevity, the formulation is presented for two subdomains 1 and 2, see for instance Figure 3.3. The process can be easily extended to arbitrary numbers of partitions.

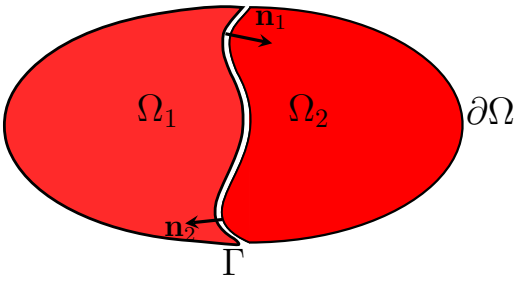


Figure 3.3: Example of a domain split into two non-overlapping subdomains.

The boundary value problem in each subdomain is recalled with a Robin type boundary condition applied on $\partial\Omega$:

$$\begin{aligned} \Delta u_i + k^2 u_i &= 0 \qquad \text{in} \qquad \Omega_i, \\ \frac{\partial u_i}{\partial n_i} + \mathrm{i}k u_i &= g_i \qquad \text{on} \qquad \partial\Omega_i \backslash \Gamma \quad i = 1, 2, \end{aligned} \tag{3.10}$$

where $\mathbf{n}_i$ is the unit outward normal on $\partial\Omega_i$. g_i is the external source which is defined and known.

In order to couple the subdomains, continuity of the pressure field and the normal pressure gradient is imposed on Γ:

$$u_1 = u_2, \qquad \frac{\partial u_1}{\partial n_1} = -\frac{\partial u_2}{\partial n_2}, \quad \text{on } \Gamma. \tag{3.11}$$

The problem described in Equation (3.10) along with the continuity constraints in Equation (3.11) is solved using FETI-2LM and its improved variant FETI-H. The theory and the formulation are now presented.

3.2.1 FETI-2LM

3.2.1.1 Introduction

The FETI-2LM method consists in using 2 Lagrange multipliers to couple the partitioned domains. This coupling is done by using the concept of characteristics method which consists in splitting the waves at the partitioned interface Γ, into incoming and outgoing parts. The method was introduced by Thompson [89] for solving time-domain wave propagation problems. It can be understood on a simple 1D example.

The acoustic wave is written as the sum of incoming and outgoing waves:

$$\begin{aligned} u(n) &= \mathcal{A} e^{-\mathrm{i}k^+ n} + \mathcal{B} e^{-\mathrm{i}k^- n}, \\ \frac{\mathrm{d}u}{\mathrm{d}n} &= -\mathcal{A}\mathrm{i}k^+ e^{-\mathrm{i}k^+ n} - \mathcal{B}\mathrm{i}k^- e^{-\mathrm{i}k^- n}, \end{aligned} \tag{3.12}$$

where $\mathcal{A}$ and $\mathcal{B}$ are the wave amplitudes corresponding to the outgoing and incoming waves respectively, and $k^{\pm} = \omega/\pm c_0$. Without loss of generality, it can be assumed that the interface is at $n = 0$. The normal derivatives on the interface are equal in magnitude on either side of the interface. The incoming wave is isolated by writing:

$$\frac{\mathrm{d}u}{\mathrm{d}n} + \mathrm{i}k^{+}u = \mathcal{B}\mathrm{i}(k^{-} + k^{+})\mathrm{e}^{-\mathrm{i}k^{-}n}. \tag{3.13}$$

This is equivalent to a generic Robin-type boundary condition which can be expressed as follows:

$$\nabla u \cdot \mathbf{n} + \mathrm{i}k^{+}u = g, \tag{3.14}$$

where g is the source term. If the solution u_{ex}, outside the domain is known, the external source can be expressed as below:

$$g = \nabla u_{ex} \cdot \mathbf{n} + \mathrm{i}k^{+}u_{ex}. \tag{3.15}$$

This technique of using the Robin type boundary conditions is exact only when the incident wave is normal to the partitioned interface. However, in the case when there is an oblique incidence better approximations of the DtN map can be used [39, 25]. This typically consists in replacing the $\mathrm{i}k^{+}$ parameter with more generic absorbing operators.

A generic way to introduce the coupling conditions on the interface can be mathematically described as follows:

$$\begin{aligned}
\frac{\partial u_1}{\partial n_1} + \mathcal{D}_1 u_1 &= \mathcal{L}_{2\to1} u_2, \\
\frac{\partial u_2}{\partial n_2} + \mathcal{D}_2 u_2 &= \mathcal{L}_{1\to2} u_1, \\
\mathcal{L}_{2\to1} u_2 &= -\frac{\partial u_2}{\partial n_2} + \mathcal{D}_2 u_2, \\
\mathcal{L}_{1\to2} u_1 &= -\frac{\partial u_1}{\partial n_1} + \mathcal{D}_1 u_1,
\end{aligned} \tag{3.16}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are the absorbing operator, and $\mathcal{L}_{2\to1}$ and $\mathcal{L}_{1\to2}$ are the transmission operators. The choice of the operators $\mathcal{D}_1$, $\mathcal{D}_2$ will be discussed later. The summation of Equation (3.16) ensuring the continuity of the normal pressure gradients results in the following:

$$\mathcal{D}_1 u_1 + \mathcal{D}_2 u_2 = \mathcal{L}_{2\to1} u_1 + \mathcal{L}_{1\to2} u_2. \tag{3.17}$$

The transmission operator involve the computation of the normal pressure gradients on the partitioned interfaces. The numerical evaluation of these gradients typically results in the loss of accuracy due to the derivation of the shape functions. This can be avoided by replacing the transmission operators by dual variables, i.e. the Lagrange multipliers as follows:

$$\begin{aligned}
\lambda_{2\to1} &= \mathcal{L}_{2\to1} u_2, \\
\lambda_{1\to2} &= \mathcal{L}_{1\to2} u_1,
\end{aligned} \tag{3.18}$$

where, $\lambda_{2\to1}$ and $\lambda_{1\to2}$ are termed as dual variables. The variables corresponding to the unknowns of the original problem i.e. the acoustic pressure variables are referred to as the primal variables.

As Lagrange multipliers are added to the unknowns of the problem, one needs to define an additional set of equations to complete the coupling of the subdomains. Imposing the continuity of the pressure field results in:

$$\begin{aligned}\mathcal{L}_{2\to1}u_1 + \mathcal{L}_{1\to2}u_2 &= (\mathcal{D}_1 + \mathcal{D}_2)u_2,\\ \mathcal{L}_{2\to1}u_1 + \mathcal{L}_{1\to2}u_2 &= (\mathcal{D}_1 + \mathcal{D}_2)u_1.\end{aligned} \tag{3.19}$$

The Equation (3.19) along with Equation (3.10) defines the strong form of the FETI-2LM problem. The absorbing operators $\mathcal{D}_1$, $\mathcal{D}_2$ are yet to defined.

One way to chose these operators is to ensure that the partitioned interface acts as a non-reflecting boundary condition. The optimal non-reflecting operator is the DtN map associated to the Helmholtz equation for the complement of the subdomains $\Omega\backslash\Omega_1$ and $\Omega\backslash\Omega_2$. Various strategies have been discussed in the literature to model these non-local operators [5]. For simplicity, the absorbing operator corresponding to the zeroth-order Taylor approximation of the DtN map is chosen as follows:

$$\mathcal{D}_1 = \mathcal{D}_2 = \mathrm{i}k. \tag{3.20}$$

3.2.1.2 Variational form

The variational formulation of FETI-2LM algorithm is now presented. Equations (3.16), (3.18) and (3.20) are introduced in the weak form (2.3), which results in two weak forms, one for each subdomain:

$$\begin{aligned}\forall q_1 \in V_1,\quad &\int_{\Omega_1} \boldsymbol{\nabla} q_1^* \cdot \boldsymbol{\nabla} u_1 - k^2 q_1^* u_1 \,\mathrm{d}\Omega + \mathrm{i}k\int_\Gamma q_1^* u_1 \mathrm{dS} - \int_{\partial\Omega_1\backslash\Gamma} q_1^* \frac{\partial u_1}{\partial n_1}\mathrm{dS} = \int_\Gamma q_1^* \lambda_{2\to1}\mathrm{dS},\\ \forall q_2 \in V_2,\quad &\int_{\Omega_2} \boldsymbol{\nabla} q_2^* \cdot \boldsymbol{\nabla} u_2 - k^2 q_2^* u_2 \,\mathrm{d}\Omega + \mathrm{i}k\int_\Gamma q_2^* u_2 \mathrm{dS} - \int_{\partial\Omega_2\backslash\Gamma} q_2^* \frac{\partial u_2}{\partial n_2}\mathrm{dS} = \int_\Gamma q_2^* \lambda_{1\to2}\mathrm{dS},\end{aligned} \tag{3.21}$$

where q_1, q_2 are the test functions associated with the pressure field u_1, u_2 respectively and V_1, V_2 are the spaces of test functions. In order to compute the dual variables described Equation (3.21), additional sets of equations are required. These sets of equations are obtained by integrating Equation (3.19) which are defined on the interface as follows:

$$\begin{aligned}\forall \psi_{2\to1}\quad &\int_\Gamma \psi_{2\to1}^*(\lambda_{2\to1} + \lambda_{1\to2} - 2\mathrm{i}k u_2)\,\mathrm{dS} = 0 \quad \text{on } \Gamma,\\ \forall \psi_{2\to1}\quad &\int_\Gamma \psi_{1\to2}^*(\lambda_{2\to1} + \lambda_{1\to2} - 2\mathrm{i}k u_1)\,\mathrm{dS} = 0 \quad \text{on } \Gamma,\end{aligned} \tag{3.22}$$

where $\psi_{2\to1}$ and $\psi_{1\to2}$ are the test functions associated with the partitioned interfaces. This interface problem is composed of only primal and dual variables without any gradient terms.

3.2.1.3 Numerical discretization

The numerical discretization consists of higher-order finite element method, p-FEM. In each element the approximation of the solution u is constructed as a linear combination of high-order shape functions, described in Section 2.2. The discretization of the variational form described in Equation (3.21) results in the following algebraic system of equations.

$$\begin{aligned} \mathbf{K}_1\mathbf{u}_1 &= \mathbf{f}_1 + \mathbf{B}_{2,1}^{\mathrm{T}}\, \mathbf{M}_{1,2}\, \boldsymbol{\lambda}_{2\to1}, \\ \mathbf{K}_2\mathbf{u}_2 &= \mathbf{f}_2 + \mathbf{B}_{1,2}^{\mathrm{T}}\, \mathbf{M}_{2,1}\, \boldsymbol{\lambda}_{1\to2}, \end{aligned} \tag{3.23}$$

where $\mathbf{K}_1$, $\mathbf{K}_2$ are the regularized subdomain matrices obtained from the discretization of the first two terms in Equation (3.21). $\mathbf{f}_1, \mathbf{f}_2$ are given forcing terms which can represent the volumic source contributions or duct mode excitations using the charateristic method on the physical boundary described in Section 3.2.1.1. $\mathbf{B}_{1,2}$, $\mathbf{B}_{2,1}$ are Boolean operators (consisting of 1 or 0) to condense the contributions of the terms on the partitioned interface. $\mathbf{M}_{1,2}$, $\mathbf{M}_{2,1}$ are mass matrices defined only on the interfaces.

In the present case, compatible meshes and interpolations are considered. This implies that the discretised matrix $\mathbf{M}_{1,2}$ associated to interface Γ is the same as $\mathbf{M}_{2,1}$. One key characteristic of all the discretised equations is that Lagrange multipliers are multiplied by $\mathbf{M}_{1,2}$. It is then judicious to consider the following substitution:

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_{2\to1} &= \mathbf{M}_{1,2}\boldsymbol{\lambda}_{2\to1}, \\ \tilde{\boldsymbol{\lambda}}_{1\to2} &= \mathbf{M}_{1,2}\boldsymbol{\lambda}_{1\to2}. \end{aligned} \tag{3.24}$$

For all the future derivations, the introduction of these Lagrange multipliers is scaled by the $\mathbf{M}_{1,2}$ matrix unless otherwise stated. The discretization of the interface problem described in Equation (3.22) results in the following set of equations:

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_{2\to1} + \tilde{\boldsymbol{\lambda}}_{1\to2} &= 2\mathrm{i}k\mathbf{M}_{1,2}\mathbf{u}_2, \\ \tilde{\boldsymbol{\lambda}}_{2\to1} + \tilde{\boldsymbol{\lambda}}_{1\to2} &= 2\mathrm{i}k\mathbf{M}_{1,2}\mathbf{u}_1. \end{aligned} \tag{3.25}$$

Introducing Equation (3.23) in the Equation (3.25) results in the following algebraic system of equations explicitly in terms of the dual variables:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} - 2ik\ \mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{B}_{2,1}^{\mathrm{T}} \\ \mathbf{I} - 2ik\ \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{B}_{1,2}^{\mathrm{T}} & \mathbf{I} \end{bmatrix}}_{\mathbf{F}_{\mathrm{I}}} \underbrace{\begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{2\to1} \\ \tilde{\boldsymbol{\lambda}}_{1\to2} \end{bmatrix}}_{\tilde{\boldsymbol{\lambda}}} = \underbrace{2ik \begin{bmatrix} \mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{f}_2 \\ \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{f}_1 \end{bmatrix}}_{\mathbf{d}}. \tag{3.26}$$

The resulting matrix, $\mathbf{F}_{\mathrm{I}}$ is complex, square, non-Hermitian and non-symmetric. The size of the interface problem is $2\mathrm{N}_{\mathrm{I}} \times 2\mathrm{N}_{\mathrm{I}}$, where N_{I} is the number of degrees of freedom on the partitioned

interface. This matrix is solved iteratively. It is important to mention that Krylov methods only requires to compute the matrix-vector products. This implies that $\mathbf{F}_\mathrm{I}$ should not be assembled. For a generic vector $\tilde{\mathbf{x}} = [\mathbf{x}_{2\to1}; \mathbf{x}_{1\to2}]^T$, this matrix vector is:

$$\mathbf{F}_\mathrm{I} \begin{bmatrix} \mathbf{x}_{2\to1} \\ \mathbf{x}_{1\to2} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{2\to1} + \mathbf{x}_{1\to2} - 2ik\, \mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{B}_{2,1}^{\mathrm{T}}\mathbf{x}_{1\to2} \\ \mathbf{x}_{1\to2} + \mathbf{x}_{2\to1} - 2ik\, \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{B}_{1,2}^{\mathrm{T}}\mathbf{x}_{2\to1} \end{bmatrix}. \tag{3.27}$$

It follows from this expression that this requires only the resolution of the respective subdomain problems, represented here formally by the inverse of the matrices $\mathbf{K}_i$. In practice these subdomain matrices are not inverted but decomposed into their LU factors. Intuitively, the right-hand side of the interface problem $\mathbf{d}$ is computed explicitly since it does not change in the iterative process. The LU factors of the matrices, $\mathbf{K}_i$ are reused to reduce these computations to forward-backward substitutions which need to be done once for each new iteration.

This algorithm is referred to as the unpreconditioned version of the FETI-2LM algorithm. The choice of the iterative scheme will be discussed later.

3.2.2 FETI-H

3.2.2.1 Introduction

The FETI-H algorithm is a modified version of FETI-2LM method. It consists in using only one Lagrange multiplier to couple the partitioned subdomains. This implies that only one coupling condition is used, contrary to the two coupling conditions described in FETI-2LM method. IT can be written as:

$$\begin{aligned} \frac{\partial u_1}{\partial n_1} + \mathcal{D}_1 u_1 &= \mathcal{L}_{2\to1} u_1, \\ \text{where,} \qquad \mathcal{L}_{2\to1} u_1 &= -\frac{\partial u_2}{\partial n_2} + \mathcal{D}_2 u_2, \end{aligned} \tag{3.28}$$

where, $\mathcal{D}_1$, $\mathcal{D}_2$ are the absorbing operators and $\mathcal{L}_{2\to1}$ is the transmission operator respectively. Subtracting Equation (3.28)(a) from Equation (3.28)(b) and imposing the normal pressure gradient continuity results in the following expression:

$$\frac{\partial u_1}{\partial n_1} + \frac{\partial u_2}{\partial n_2} = \mathcal{L}_{2\to1} u_1 - \mathcal{L}_{2\to1} u_1 + \mathcal{D}_1 u_1 - \mathcal{D}_1 u_2 = 0 \ \text{ on } \ \Gamma. \tag{3.29}$$

The same operator $\mathcal{D}_1 = \mathcal{D}_2$ has to be chosen in order to ensure the continuity of the pressure field.

$$u_1 - u_2 = 0 \ \text{ on } \ \Gamma. \tag{3.30}$$

It is interesting to note that in Equation (3.28), the normal pressure gradients have opposite signs. This means that each domain should be allocated a sign. In this simple case with two subdomains,

the attribution of a sign to each subdomain is straightforward. For the simple chequerboard pattern of partitioning, the allocation of signs to the partitions has no ambiguity. However extending this sign convention to an arbitrary type of partitioning is more involved. The work done by Farhat *et al* [33] to randomly allocate signs to different subdomains and partitioned interfaces is followed. The sign conventions can be summarized by introducing the following notations. Let ϵ_i be the sign associated with subdomain i and ϵ_j be the sign associated with all the neighbours of subdomain i, $\forall j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of indices j, such that Ω_j and Ω_i are neighbours. The first step is to randomly allocate signs to all domains by ensuring that if a particular domain has a positive sign then at least one of its neighbours has an opposite sign [33]. In the next step the signs of the subdomains are allocated to the interface. Let $\epsilon_{i,j}$ be the sign operator associated with the interfaces of the partitioned subdomains. In the case when the product $\epsilon_i \epsilon_j$ is positive then no sign is introduced on the interface. This means that no regularization is done on either side of the interface. However, if the product of ϵ_i ϵ_j is negative then the interfaces are signed following the sign of the respective subdomains.

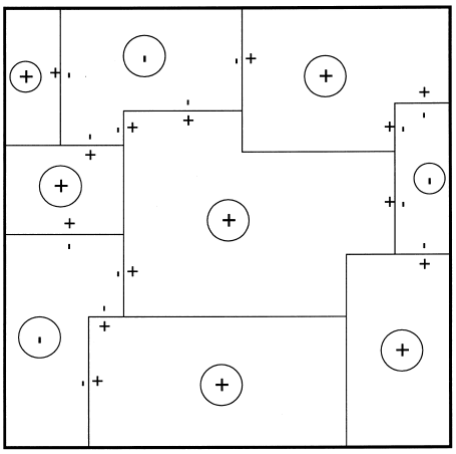


Figure 3.4: Sign conventions introduced in FETI-H [33] for arbitrary partitionings.

3.2.2.2 Variational form

The definitions of the transmission operator in Equation (3.18), absorbing operator in Equation (3.20) are used in the coupling relation inEquation (3.28). The coupling relation is introduced in the variational form in Equation (2.3). This results in the following:

$$\begin{aligned}
\forall q_1 \in V_1, \quad & \int_{\Omega_1} \boldsymbol{\nabla} q_1^* \cdot \boldsymbol{\nabla} u_1 - k^2 q_1^* u_1 \, \mathrm{d}\Omega + \epsilon_{1,2} \mathrm{i}k \int_\Gamma q_1^* u_1 \mathrm{dS} - \int_{\partial\Omega_1 \setminus \Gamma} q_1^* \frac{\partial u_1}{\partial n_1} \mathrm{dS} = \epsilon_{1,2} \int_\Gamma q_1^* \lambda_{2\to 1} \mathrm{dS}, \\
\forall q_2 \in V_2, \quad & \int_{\Omega_2} \boldsymbol{\nabla} q_2^* \cdot \boldsymbol{\nabla} u_2 - k^2 q_2^* u_2 \, \mathrm{d}\Omega + \epsilon_{2,1} \mathrm{i}k \int_\Gamma q_2^* u_2 \mathrm{dS} - \int_{\partial\Omega_2 \setminus \Gamma} q_2^* \frac{\partial u_2}{\partial n_2} \mathrm{dS} = \epsilon_{2,1} \int_\Gamma q_2^* \lambda_{2\to 1} \mathrm{dS}.
\end{aligned} \tag{3.31}$$

In order to compute the additional set of dual variables, the interface problem needs to be formulated by integrating Equation (3.30) with a test function, $\psi_{2\to1}$ which results in the following:

$$\forall\psi_{2\to1} \quad \int_\Gamma \psi^*_{2\to1}(u_1 - u_2)\mathrm{d}\Gamma = 0 \ \text{ on } \ \Gamma. \tag{3.32}$$

3.2.2.3 Numerical discretization

The FETI-H formulation is also discretized using the high-order finite-element method p-FEM. On each element the approximation of the solution u is constructed as a linear combination of high-order shape functions as described in Section 2.2. The discretization results in the following set of algebraic equations:

$$\begin{aligned}\mathbf{K}_1\mathbf{u}_1 &= \mathbf{f}_1 + \epsilon_{1,2}\mathbf{B}^{\mathrm{T}}_{2,1}\ \mathbf{M}_{1,2}\ \boldsymbol{\lambda}_{2\to1},\\ \mathbf{K}_2\mathbf{u}_2 &= \mathbf{f}_2 + \epsilon_{2,1}\mathbf{B}^{\mathrm{T}}_{1,2}\ \mathbf{M}_{1,2}\ \boldsymbol{\lambda}_{2\to1},\end{aligned} \tag{3.33}$$

where, the definitions of the operators are exactly the same as described for the FETI-2LM algorithm. In addition, the same arguments hold for the $\mathbf{M}_{1,2}$ matrices on the partitioned interface and also for the scaling introduced for the Lagrange multiplier as seen in Equation (3.24). The discretization of the interface problem described in Equation (3.32) results in the discretized continuity of the pressure on the interface, since the mass matrices can be factorized. Using the relations in Equation (3.33) the interface problem can be expressed as follows:

$$\mathbf{B}_{1,2}\mathbf{K}_1^{-1}(\mathbf{f}_1 + \epsilon_{1,2}\mathbf{B}^{\mathrm{T}}_{2,1}\ \mathbf{M}_{1,2}\ \boldsymbol{\lambda}_{2\to1}) - \mathbf{B}_{2,1}\mathbf{K}_2^{-1}(\mathbf{f}_2 + \epsilon_{2,1}\mathbf{B}^{\mathrm{T}}_{1,2}\ \mathbf{M}_{1,2}\ \boldsymbol{\lambda}_{2\to1}) = \mathbf{0}. \tag{3.34}$$

The resulting set of equations can be expressed in a concatenated way as follows:

$$\underbrace{\begin{bmatrix}\epsilon_{1,2}\mathbf{B}_{2,1}\mathbf{K}_1^{-1}\mathbf{B}^{\mathrm{T}}_{2,1}- \\ \epsilon_{2,1}\mathbf{B}_{1,2}\mathbf{K}_2^{-1}\mathbf{B}^{\mathrm{T}}_{1,2}\end{bmatrix}}_{\mathbf{F}_\mathrm{I}}\underbrace{\begin{bmatrix}\tilde{\boldsymbol{\lambda}}_{2\to1}\end{bmatrix}}_{\tilde{\boldsymbol{\lambda}}} = \underbrace{\begin{bmatrix}-\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{f}_1+ \\ \mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{f}_2\end{bmatrix}}_{\mathbf{d}}. \tag{3.35}$$

The resulting interface problem is sparse, complex symmetric but not Hermitian and contains roughly half the number of degrees of freedom compared to its FETI-2LM variant, i.e. $\mathrm{N_I} \times \mathrm{N_I}$. The matrix-vector products required in the iterative process are obtained as follows: For a generic vector $\tilde{\mathbf{x}} = [\mathbf{x}_{2\to1}; \mathbf{x}_{1\to2}]^T$, this matrix vector is:

$$\mathbf{F}_\mathrm{I}\begin{bmatrix}\mathbf{x}_{2\to1}\end{bmatrix} = \begin{bmatrix}\epsilon_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{B}^{\mathrm{T}}_{1,2}\mathbf{x}_{2\to1} - \epsilon_{2,1}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{B}^{\mathrm{T}}_{2,1}\mathbf{x}_{2\to1}\end{bmatrix}. \tag{3.36}$$

It follows from this expression that this requires only the resolution of the local problem on each sub-domain, represented here formally by the inverse of the matrices $\mathbf{K}_i$. The sign conventions are introduced once and do not change and thus the rest of the solution procedure is similar to the FETI-2LM algorithm. This algorithm is referred to as the unpreconditioned version of the FETI-H algorithm. The choice of the iterative scheme will be discussed in the next subsection.

3.2.2.4 Iterative solvers for the interface problem

This subsection focuses on solving the interface problem in Equation (3.35), Equation (3.26) using an iterative solver. The choice of the iterative scheme plays an important role to limit the cost of the calculation. It is typically recommended to use Krylov subspace methods based on full reorthogonalization for matrix problems arising from a FETI method. Iterative schemes like GMRES, ORTHODIR, ORTHOMIN, GCR [84][Sec 6.5, 6.9] are commonly used.

In GMRES, the descent direction vectors u are orthonomalized whereas in ORTHODIR the vectors arising from the projection of the interface matrix on the descent directions are orthonormalized. ORTHODIR requires the storage of not just the descent direction vectors but also these projected vectors, however it provides the approximate solution at each step (without requiring additional computations) which is crucial to examine the performance of the method. Mathematically, both GMRES and ORTHODIR are equivalent and provide the same approximate solution at each iteration.
In this chapter, ORTHODIR is used to solve both the FETI variants so as to compare these methods in terms of their performance. A deeper analysis of the differences between these algorithms will be presented in chapter 4. There is a slight variance in the cost and memory requirements of these methods, details of which can be found in literature [84] [Sec 6.9]. The details of the ORTHODIR iterative scheme is outlined in Section A.1. The interface matrix in FETI-H is complex symmetric, which is not fully exploited using ORTHODIR. Nevertheless, ORTHODIR is a good choice to compare the performance of both methods in the absence of any preconditioner.

The implementation of FETI-2LM on benchmark test cases has been verified [58][Chapter 4]. The novelty in this work is the implementation and verification of FETI-H algorithm on a high-order FEM based discretization and comparison of both these FETI variants for the same benchmark problems.

3.3 Results

In this section, numerical results of FETI-2LM and FETI-H implementation are presented and compared for varying h, p, wavenumbers and number of subdomains.

3.3.1 Scalability analysis

The results for the FETI-2LM algorithm have been presented [58][Chapter 4]. In this work, the same results are recomputed and presented (in the same implementation framework) to provide a comparison of both FETI-variants on the same problem. These results are added to provide a fair comparison between the two methods and thus not commented in greater depth. However, the results

presented for the FETI-H algorithm are completely novel and to the best of authors knowledge have not been presented before on a p-FEM based discretization for the Helmholtz equation.

3.3.1.1 Test case description

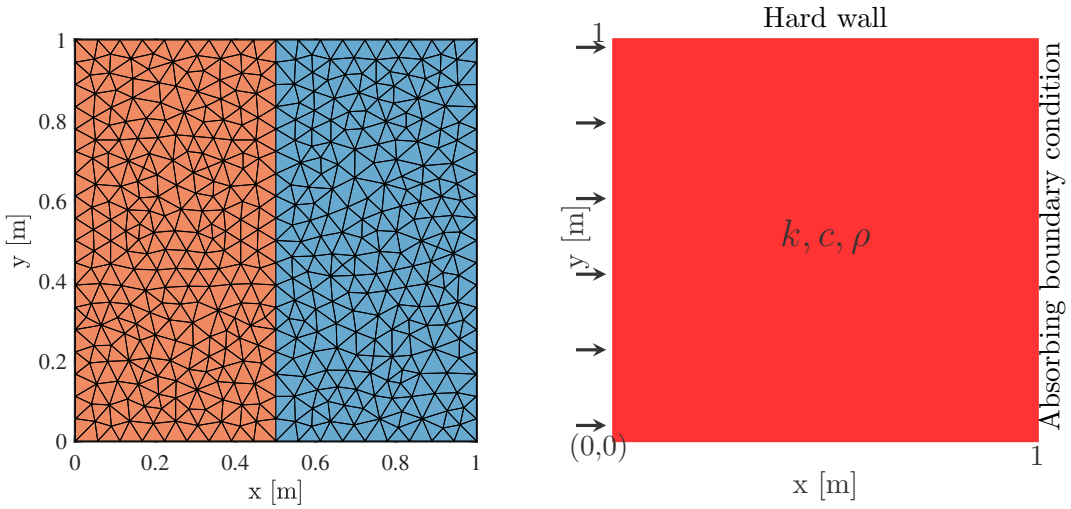


Figure 3.5: (a) Computational domain consisting of homogeneous medium split into two partitions, (b) Domain consisting of homogeneous medium with different boundary conditions.

A 2D guided wave problem is considered which has been used to analyze the performance of FETI methods in previous studies. In this work, Robin boundary condition is used to inject an incoming wave from the left boundary of the domain in Figure 3.5 (b). In addition, absorbing boundary conditions are used for truncation and avoiding spurious reflections back into the domain. The top and the bottom ends are hard walls. The computational domain is of unit dimensions with a single plane wave of the form:

$$u_{\text{ex}} = \mathrm{e}^{-\mathrm{i}kx}, \tag{3.37}$$

where x represents the propagation direction. The geometry is discretized into finite element triangulations using an open source meshing tool, Gmsh [46]. After the discretization, the geometry is split into multiple subdomains using a free graph partitioning tool, METIS [52]. METIS is a multi-level graph partitioning software package used for partitioning large meshes, and computing fill-reducing orderings of sparse matrices. METIS can be used from within Gmsh or externally as a standalone API. In this work, METIS is used as an standalone API as it provides additional functionalities for unequal load balancing. METIS ensures two important distinctions for partitioning.

1. It ensures that all the partitions have more or less equal load sharing in terms of the partitions. This would help to improve the parallel efficiency and scalability of the method.

2. The next objective is to minimize the communication between the partitions. This would reduce the exchange time between the subdomains and also help improve the parallel efficiency.

After the partitioning, the subdomain matrices are assembled using p-FEM with a constant polynomial order p throughout the whole domain. The numerical models are solved iteratively using ORTHODIR. The iterative scheme is initiated with a zero solution, where a tolerance $\epsilon = 10^{-8}$ is set for the normalized residual error of the interface problem. The normalized residual error, $\tilde{r}$ is computed as follows:

$$\tilde{r} = \frac{\|\mathbf{d} - \mathbf{F}_{\mathrm{I}}\tilde{\boldsymbol{\lambda}}\|}{\|\mathbf{d}\|} \,. \tag{3.38}$$

The iterative scheme is allowed to run up to a maximum of 1000 iterations. As the analytical solution is known, the E_{L_2}, E_{H_1} errors in % can be computed after the global solution is recovered from the subdomains. The accuracy of the numerical solution is evaluated by calculating the relative E_{L_2} error as follows:

$$E_{L_2}(\Omega) = \frac{||u^h - u_{\mathrm{ex}}||_{L_2(\Omega)}}{||u_{\mathrm{ex}}||_{L_2(\Omega)}} \times 100, \quad \text{with } ||\cdot||_{L_2(\Omega)} = \left(\int_\Omega |\cdot|^2 \mathrm{d}\Omega \right)^{1/2}. \tag{3.39}$$

Let N_s be the number of partitions, N_{DOF} be the total number of DOFs, D_λ be the number of DOFs per wavelength, n_c be the number of iterations required for convergence to the set tolerance. The number of degrees of freedom per wavelength, D_λ are computed in the following way:

$$D_\lambda = \frac{2\pi}{kL}\left(\sqrt{N_{\mathrm{DOF}}} - 1\right). \tag{3.40}$$

In the next part, the influence of various parameters is analyzed on the performance of both FETI methods. The strong scalability behavior is evaluated using the varying subdomain case for a fixed problem size. The weak scalability performance is analyzed by varying the problem parameters like the mesh and p order. In addition, the influence of increasing wavenumbers for the same resolution is also examined.

3.3.1.2 Influence of the number of subdomains

In this part, the behavior of the FETI-H method is examined for varying partitions for a fixed problem size. This corresponds to the strong scalability of a DDM. For a given relatively high Helmholtz number kL, and fixed polynomial order and mesh resolution, the number of partitions are varied from $N_s = 2, 4, 8, 16, 32$ using the METIS API as seen in Figure 3.6.

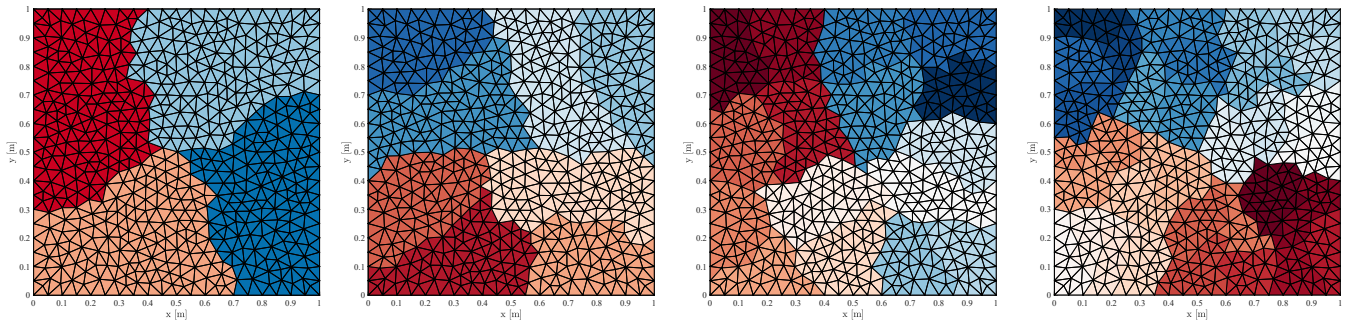


Figure 3.6: Different non-overlapping subdomains $N_s = 4, 8, 16, 32$ generated using METIS.

The results for varying partitions are presented in Table 3.1 and Figure 3.7.

Table 3.1: Effect of the number of subdomains N_s, fixed: $h/L = 1/60$, $kL = 100$, $p = 6$, $\epsilon = 10^{-8}$ for the FETI-H algorithm, the results marked in $\star$ correspond to the FETI-2LM algorithm.

N_s	N_I	N_{DOF}	D_λ	n_c	$n_c^\star$	E_{L_2} in %	E_{H_1} in %
2	433	172,118	26	223	132	5.9×10^{-5}	4×10^{-4}
4	892	172,576	26	348	188	5.9×10^{-5}	4×10^{-4}
8	1669	173,348	26	596	288	5.9×10^{-5}	4×10^{-4}
16	2564	174,232	26	982	365	1.5×10^{-4}	4×10^{-4}
32	4093	175,736	26	1000	532	87.36	87.47
						$(5.9\times10^{-5*})$	$(4\times10^{-4*})$

Influence of varying subdomains

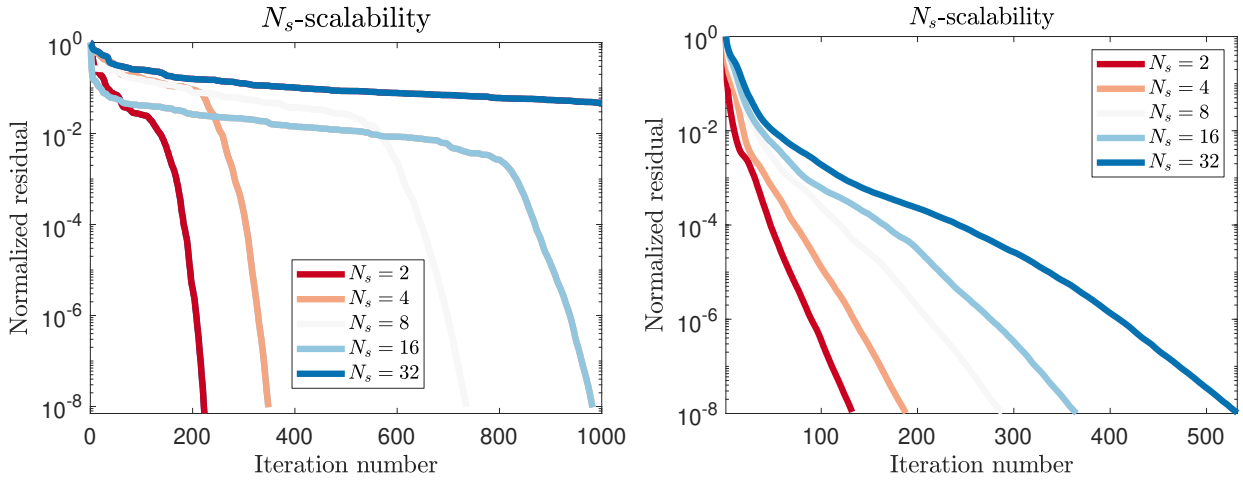


Figure 3.7: (a) FETI-H, (b) FETI-2LM algorithm, effect of the number of subdomains N_s for $p = 6$, $kL = 100$, $h/L = 1/60$, $\epsilon = 10^{-8}$.

The convergence of the normalized residual can be seen in Figure 3.7. There are two distinct

regimes which can be seen where the convergence rate is complete different namely, the pre-convergence regime and the convergence regime.

- Pre-convergence regime: This regime is characterized by slow or almost flat convergence. During this regime, the subdomains exchange information and the information propagation to the neighbours begins. As the number of subdomains increase, the iterations required to exchange this information also increases drastically. It can be seen that usually more than 70 percent of the iterations are consumed to exchange this information for the FETI-H algorithm. For the case with 32 partitions, all the 1000 iterations are consumed in this regime, which is a considerable computational effort. However, this behavior is expected for FETI-H since no additional coarse space correction has been devised to tackle this. However the design of these coarse space is extremely challenging, especially for p-FEM.
- Convergence regime: In this regime, the convergence of the normalized residual is much faster. The convergence rate in this regime only slightly depends on the number of partitions, N_s.

It can be seen that the L_2 and H_1 errors remain constant for the first 4 examples for varying partitions, except for the slight variance of the L_2 error for the case of 16 partitions. This could be attributed to the fact that a very high number of iterations are required for convergence to the set tolerance. Moreover, the last example of 32 partitions, did not converge to the set tolerance even after 1000 iterations which explains the high L_2 and H_1 errors. There is a four-fold increase in the number of iterations for an eight-fold increase in the number of partitions, which showcases the poor scalability of the unpreconditioned FETI-H algorithm. It can be seen that the FETI-2LM algorithm scales better in terms of the number of iterations required for convergence for the same accuracy compared to FETI-H. The total number of DOFs only vary slightly owing to the duplication introduced during partitioning.

3.3.1.3 Effect of mesh refinement

In this part, the mesh size is varied for a fixed polynomial order, partitioning and frequency. The normalized residual is plotted for varying iterations as seen in Figure 3.8 and Table 3.2. For the first two mesh resolutions, the complete span of the interface problem is required for the convergence to set tolerance. Since D_λ, is quite low for the coarsest mesh, the relative errors are high which decrease consistently as the mesh is refined. The number of iterations increase the mesh refinement upto a certain point i.e. till $h/L = 1/60$, followed by a decrease in the number of iterations on further refinement upto $h/L = 1/100$. In addition, the slope in the convergence regime seems to decrease

slightly with increase in mesh refinement. In general, it can thus be said the FETI-H algorithm is scalable with respect to the mesh size.

Table 3.2: Effect of mesh refinement for $N_s = 5$, $p = 6$, kL= 100, $\epsilon = 10^{-8}$ for the FETI-H algorithm.

h/L	N_I	N_{DOF}	D_λ	n_c	E_{L_2} in %	E_{H_1} in %
1/10	192	4667	4.3	190	33.8	34.5
1/15	283	11009	6.6	280	0.9	1.4
1/30	577	43163	13.1	383	0.006	0.025
1/60	1105	172,787	26.1	439	6.3×10^{-5}	4×10^{-3}
1/80	1519	304,373	34.6	428	7×10^{-6}	7.5×10^{-5}
1/100	1831	480,533	43.5	409	1.9×10^{-6}	1.8×10^{-5}

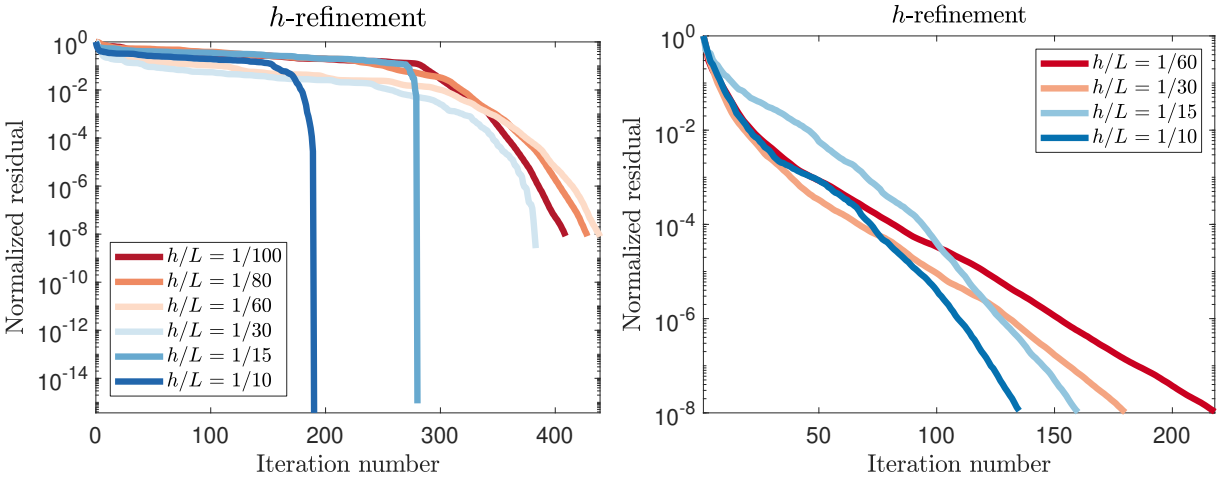


Figure 3.8: (a) FETI-H, (b) FETI-2LM algorithm, effect of mesh refinement for $N_s = 5$, $p = 6$, $kL = 100$, $\epsilon = 10^{-8}$.

The FETI-2LM algorithm seems to scale better with mesh refinement for the same set of examples presented for the FETI-H algorithm.

3.3.1.4 Effect of the interpolation order

In this part, the polynomial order, p is varied for a fixed mesh resolution, partitioning and wavenumber as seen in Table 3.3 and Figure 3.9 on the left. The first observation is that there is very small dependency of the polynomial order on the convergence regime even for high orders. Another observation is that for increase in the polynomial order from 1 to 6, there is 7 order drop in the error. The errors stagnate after this point due to the discretization error introduced in the system. The number of iterations increase only slightly compared to increase in the global size of the problem. It can thus be

said that the FETI-H algorithm scales with p-refinement. However, the FETI-2LM algorithm scales better than FETI-H for increasing polynomial orders as seen in Figure 3.9 on the right.

Table 3.3: Effect of polynomial order for $N_s = 5$, $h/L = 1/60$, $kL = 100$, $\epsilon = 10^{-8}$ for the FETI-H algorithm.

p	N_I	N_{DOF}	D_λ	n_c	E_{L_2} in %	E_{H_1} in %
1	194	5060	4.5	147	129.9	130.1
2	373	19607	8.8	224	14.7	15.5
3	556	43655	13.1	277	0.27	0.7
4	739	77201	17.4	329	0.01	0.065
5	922	120,245	21.8	387	7.4×10^{-4}	5.5×10^{-3}
6	1124	172,805	26.1	505	6.2×10^{-5}	4×10^{-4}
7	1288	234,827	30.4	496	5.4×10^{-5}	5.4×10^{-5}

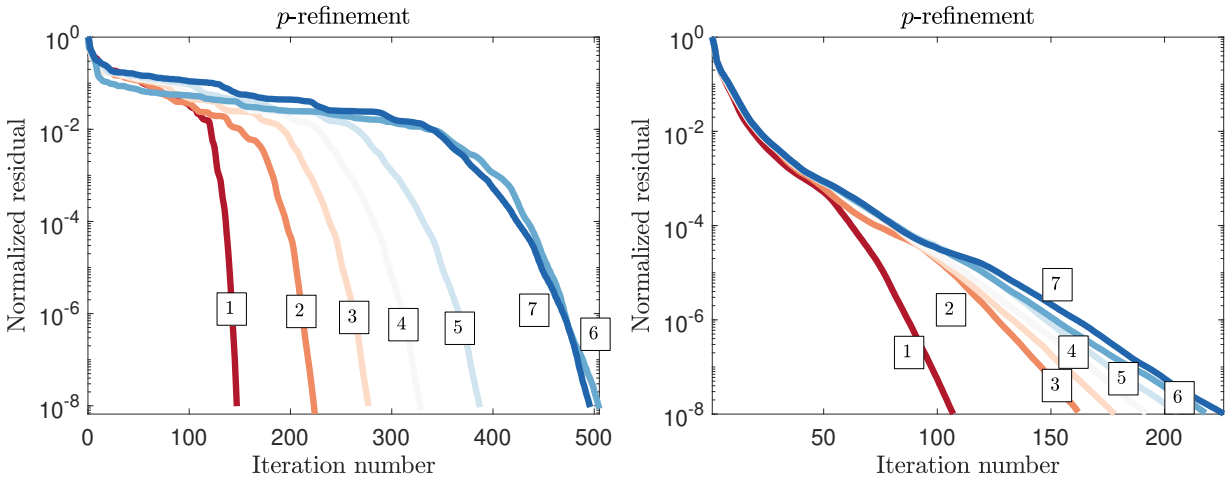


Figure 3.9: (a) FETI-H, (b) FETI-2LM algorithm, effect of polynomial order for $kL = 100$, $h/L = 1/60$, $\epsilon = 10^{-8}$ for $N_s = 5$.

3.3.1.5 Effect of the wavenumber

In this part, the mesh resolution, polynomial order and the number of partitions are kept constant for varying wavenumbers as seen in Figure 3.10 and Table 3.4. It can be seen that number of iterations increases drastically for high wavenumbers. For a 8 fold increase in the wavenumber, the number of iterations, n_c increases by a factor more than 4, thus the algorithm is not scalable with increasing wavenumbers. This is in agreement with the literature [54] and contrary to the work done on FETI-2LM for the same benchmark problem [59]. As seen in Table 3.4, the iterations for FETI-2LM only increase slightly for increasing wavenumbers for the same accuracy and number of partitions. As the wavenumber increases, the error also increases since the problem resolution is the same for all the

calculations.

Table 3.4: Effect of varying wavenumbers, fixed $h/L = 1/30, p$= 6, $\epsilon = 10^{-8}$ for arbitrary partitioning $N_s = 5$, the results marked in $\star$ correspond to the FETI-2LM algorithm.

kL	N_I	N_{DOF}	D_λ	n_c	$n_c^\star$	E_{L_2} in %	E_{H_1} in %
25	577	43163	52.2	121	179	5.2×10^{-6}	8.6×10^{-6}
50	577	43163	26.1	234	185	5×10^{-5}	4.2×10^{-4}
100	577	43163	13.1	383	193	6×10^{-3}	0.025
200	577	43163	6.5	580	209	1.65	2.0

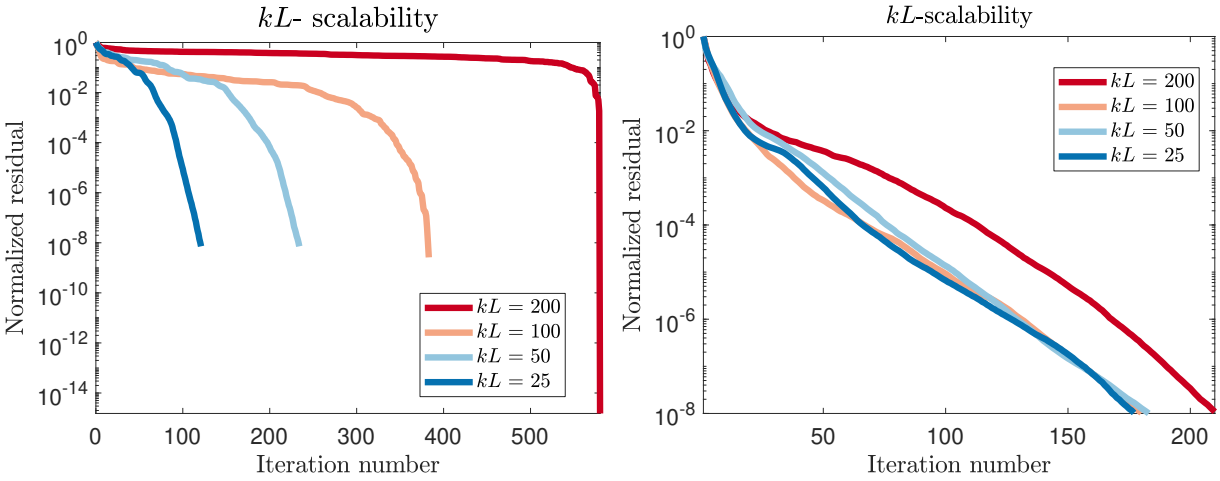


Figure 3.10: (a) FETI-H, (b) FETI-2LM algorithm, effect of varying wavenumbers, fixed $h/L = 1/30, p = 6$, $\epsilon = 10^{-8}$ for $N_s = 5$.

Contrarily, the FETI-2LM algorithm scales quite well for varying wavenumbers as seen in Figure 3.10.

To conclude, it can be said that the FETI-H algorithm scales with p and h refinement, however the scalability in terms of increasing partitions, N_s and wavenumber, k is quite poor. In the next section, the FETI-2LM and FETI-H methods are compared in terms of their performance.

3.3.2 Comparison

This subsection focuses on comparing both the FETI variants for increasing partitions and wavenumbers. The scalability results for FETI-2LM have been discussed and presented in literature [59]. It has been observed that both FETI-2LM and FETI-H exhibit scalability with increasing polynomial order and mesh refinement. As a result, both methods are only compared for N_s, k -scalability.

3.3.2.1 Increasing subdomains

In this analysis, the number of subdomains are varied for a fixed polynomial order, mesh size, wavenumber, interface tolerance. The same iterative scheme ORTHODIR for the same set tolerance is used to compare both the FETI - variants, details are described in Section 3.3.1.2. The results for varying iteration number with increasing partitions is plotted in Figure 3.11 (a). The first remark is that both methods have the exact same accuracy for exactly the same problem parameters. It can be easily seen that FETI-2LM outperforms FETI-H even for small number of partitions. Moreover, the scalability is quite poor for FETI-H compared to FETI-2LM. In general, the number of iterations required for convergence is directly related to the computational cost, so it becomes fair to say that FETI-2LM outperforms FETI-H for increasing partitions in the absence of any additional preconditioner.

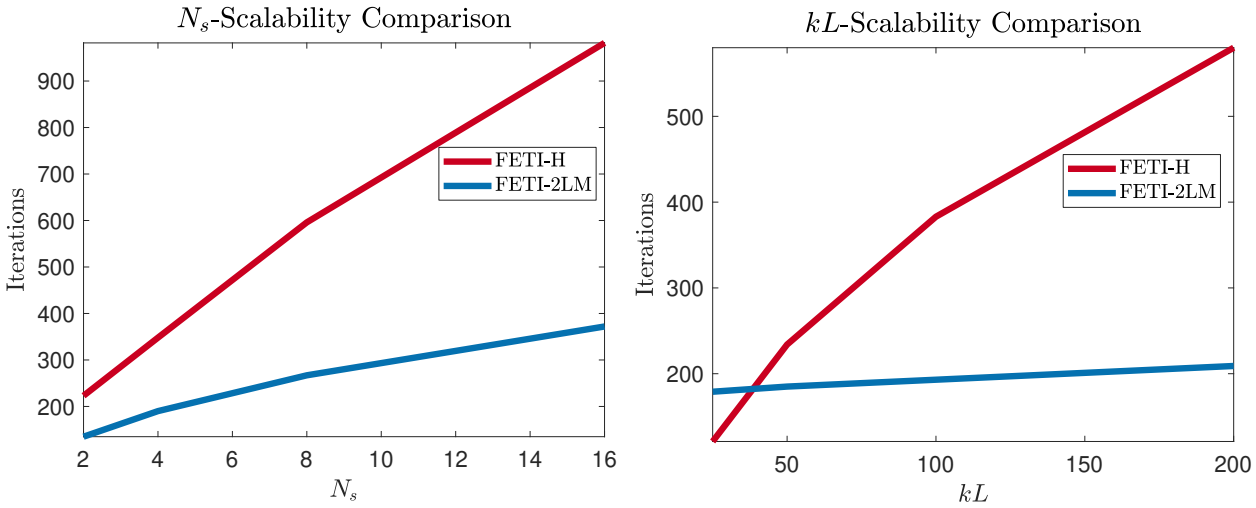


Figure 3.11: For the same resolution and accuracy, (a) comparison of FETI-H with FETI-2LM for varying subdomains, (b) comparison of FETI-H with FETI-2LM for varying wavenumbers.

3.3.2.2 Increasing wavenumber

As seen Figure 3.11 (b), both FETI variants are plotted for varying wavenumbers and fixed polynomial order, resolution and partitions. The details of this test are described in wavenumber scalability test described before in Section 3.3.1.5. Both the methods have the exact same accuracy with all other problem parameters kept the same. It can be seen that, FETI-2LM scales quite well for increasing wavenumbers. Contrarily, the scalability of FETI-H is quite poor. In general, FETI-2LM quite easily outperforms FETI-H for increasing wavenumbers. In order to improve the scalability of both FETI variants, a second level auxiliary problem is constructed by projecting the interface problem onto a suitable coarse space. At each iteration, the interface residual is made orthogonal to the chosen coarse subspace. It has been remarked that the use of this coarse subspace type of preconditioner is more

efficient on FETI-H compared to FETI-2LM.
However the design of these coarse subspaces for a numerical discretization arising from discretization of p-FEM is quite challenging. Infact, the design of optimal coarse space for indefinite wave propagation problems is still an active area of research.

3.4 Conclusions

This chapter presented an overview of the DDM applied for time harmonic wave-propagation problems. In the first part, a detailed literature review of the three different class of DDM namely the classical Schwarz, optimized Schwarz and the Schur-based for time-harmonic wave propagation problems is done. Out of these DDM, two Schur-based dual sub-structuring methods namely, FETI-2LM and FETI-H are formulated, implemented and verified.
The scalability tests have been performed to access the performance for modeling the Helmholtz equation using both the FETI variants for varying problem parameters, number of partitions and frequencies. As discussed in the previous work [58] [Chapter 4], the FETI-2LM algorithm demonstrates excellent scalability for varying wavenumbers, h, p refinement however as expected scales poorly for varying subdomains.
The comparison of both these algorithms for varying partitions and wavenumbers led to the conclusion that FETI-2LM scales quite well compared to FETI-H. However for low wavenumbers ($kL < 50$), FETI-H seems to perform slightly better. Since the objective of using DDM is to solve large-scale problems at high frequencies, this benefit is alleviated for FETI-H. In general, it can be said that FETI-2LM clearly outperforms FETI-H in the absence of a global preconditioner.
The next chapter focuses on extending these methods to domains comprising of several media.

Chapter 4

Domain decomposition methods for several fluids

4.1 Introduction

Domain decomposition methods have been examined extensively for homogeneous media arising from the discretization of Helmholtz equation [29]. The popular FETI methods like FETI-2LM, FETI-H and FETI-DPH have been studied for exterior homogeneous Helmholtz problem [34], [30] as outlined in Chapter 3. Nevertheless, problems associated to coupling of several fluids, such as the ones presented in chapter Chapter 2, are not treated in the literature. One can nevertheless cite that the optimized Schwarz type DDM methods have been used to model the Helmholtz problems with discontinuous coefficients, however only theoretical estimates for convergence have been determined [24].

In many practical applications there is a need to model problems consisting of different media. A classical example is the modeling of the Kundt's tube which can involve several materials. Moreover many industrial applications require the modeling of acoustic treatments for noise mitigation like duct silencers, mufflers, which are practical applications of models with several fluids. Chapter 2 highlighted the computational issues in the optimization methods for acoustic treatments, specifically for liners. There is a need to develop original techniques that can improve the resolution process. This chapter focuses on Domain Decomposition Methods (DDM) for time-harmonic problems with several fluids. The key objective is to extend the FETI methods to problems with several fluids by introducing continuity and transmission conditions. An additional contribution is to generalise the formulation of FETI methods to equivalent fluids.

A part of this chapter is submitted as an article in the Journal of Sound and Vibration.

In the first part, the FETI methods are extended to problems with several fluids. In the second

part, the implementation is verified for known analytical solutions on 2D benchmark problems. In the latter part, scalability tests are performed in 2D to analyze the performance of this method. The performance and accuracy of the method are verified for 3D models. In the last part, an improved factorization strategy using the chosen DDM is formulated to reduce the costs for several evaluations. This strategy is applied to the same models as presented in Chapter 2. Finally, the computational savings using the proposed DDM are discussed.

4.2 FETI methods for several fluids

This section presents the problem description of a domain composed of several fluids. Generalized coupling and continuity conditions are presented to couple several fluids using FETI-based methods. Without loss of generality and for the sake of simplicity, only the case of two different fluids is presented. The extension to an arbitrary number of fluids is straightforward.

A two-dimensional domain composed of two fluids (denoted by the indices 1 and 2 respectively) corresponding to subdomains, Ω_1 and Ω_2 is presented in Figure 4.1. $\mathbf{n}_i$ denotes the unit outward normal on the boundary $\partial\Omega_i$ of Ω_i. The two subdomains are separated by an interface, $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$.

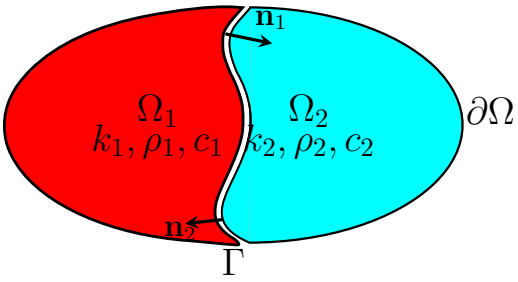


Figure 4.1: Domain with two fluids, each fluid has different physical properties. They are separated by the interface Γ.

In each subdomain, the pressure field u is governed by the Helmholtz equation:

$$\frac{1}{\rho_i}\Delta u + \frac{\omega^2}{\rho_i c_i^2} u = 0 \qquad \text{in} \qquad \Omega_i, \quad i = 1, 2, \tag{4.1}$$

where c_i and ρ_i respectively denote the speed of sound and the density of the fluid in sub-domain Ω_i. These coefficients can be of complex valued and frequency dependent to take dissipative effects into account. A remark is that Equation (4.1) is scaled by a factor of $1/\rho$ as compared to the classical form of Helmholtz equation. The reason for this is found in the continuity relations at the interface Γ for fluids with different densities as seen in Equation (4.2):

$$u_1 = u_2 \quad , \quad \frac{1}{\rho_1}\frac{\partial u_1}{\partial n_1} = -\frac{1}{\rho_2}\frac{\partial u_2}{\partial n_2} \quad \text{on} \quad \Gamma. \tag{4.2}$$

The first relation corresponds to the continuity of pressure and the second one states the continuity of normal velocity. The densities may differ and therefore this relation does not imply the continuity of the normal gradient of pressure. For boundaries of the subdomains other than the common interface Γ, external forcing terms are classically introduced through Robin-type boundary conditions:

$$\frac{1}{\rho_i}\frac{\partial u_i}{\partial n_i} + \mathrm{i}\frac{\omega}{Z_i}u_i = g_i \qquad \text{on} \qquad \partial\Omega_i \backslash \Gamma \quad i = 1, 2, \tag{4.3}$$

where g_i represents a generic external source. Z_i is the characteristic impedance of a fluid with density ρ_i and speed of sound c_i defined as follows:

$$Z_i = \rho_i c_i \ \ \forall\, i = 1, 2. \tag{4.4}$$

The variational formulations associated with Equation (4.1) for the two subdomains can be expressed as follows:

$$\forall q_i \in V_i, \qquad \frac{1}{\rho_i}\int_{\Omega_i} \boldsymbol{\nabla} q_i^* \cdot \boldsymbol{\nabla} u_i - k_i^2\, q_i^* u_i \,\mathrm{d}\Omega - \frac{1}{\rho_i}\int_{\partial\Omega_i} q_i^* \frac{\partial u_i}{\partial n_i}\mathrm{dS} = 0, \quad i = 1, 2, \tag{4.5}$$

where q_i is the test function associated with the field u_i, V_i is the space of test functions and $k_i = \omega/c_i$. Each of the two subdomains can be self-partitioned, nevertheless for the sake of simplicity, no partitioning is introduced in each subdomains.

The next subsections describe the theory, formulation of FETI-2LM and FETI-H.

4.2.1 FETI-2LM

The generalized Robin type boundary conditions are introduced similar to the formulation for homogeneous fluid as follows:

$$\begin{aligned} \frac{\partial u_1}{\partial n_1} + \mathcal{D}_1 u_1 &= \rho_1 \mathcal{L}_{2\to 1} u_1, \\ \mathcal{L}_{2\to 1} &= -\frac{1}{\rho_2}\frac{\partial u_2}{\partial n_2} + \mathcal{D}_2 u_2, \\ \frac{\partial u_2}{\partial n_2} + \mathcal{D}_2 u_2 &= \rho_2 \mathcal{L}_{1\to 2} u_2, \\ \mathcal{L}_{1\to 2} &= -\frac{1}{\rho_1}\frac{\partial u_1}{\partial n_1} + \mathcal{D}_1 u_1, \end{aligned} \tag{4.6}$$

where $\mathcal{D}_1$, $\mathcal{D}_2$ are absorbing operators, and $\mathcal{L}_{1\to 2}, \mathcal{L}_{2\to 1}$ are the transmission operators. The scaling by a factor ρ is introduced in Equation (4.6) in order to simplify the continuity relations at the interface.

We recall the Lagrange multipliers that are introduced to avoid the explicit computation of the normal pressure gradients.This improves the overall performance of the FETI-2LM method [87]. They

are expressed as follows:

$$\begin{aligned}\lambda_{2\to1} &= \mathcal{L}_{2\to1}u_2,\\ \lambda_{1\to2} &= \mathcal{L}_{1\to2}u_1.\end{aligned} \tag{4.7}$$

Adding Equation (4.6) and imposing the continuity constraints results in the following expression for the interface problem:

$$\begin{aligned}\lambda_{2\to1} + \lambda_{1\to2} &= \left(\frac{\mathcal{D}_1}{\rho_1} + \frac{\mathcal{D}_2}{\rho_2}\right)u_1 \ \text{ on } \ \Gamma,\\ \lambda_{1\to2} + \lambda_{2\to1} &= \left(\frac{\mathcal{D}_1}{\rho_1} + \frac{\mathcal{D}_2}{\rho_2}\right)u_2 \ \text{ on } \ \Gamma.\end{aligned} \tag{4.8}$$

In the present work, the zeroth order transmission conditions are used as described in Equation (4.9).

$$D_1 = \mathrm{i}k_1, \qquad D_2 = \mathrm{i}k_2. \tag{4.9}$$

Introducing the definitions of Equation (4.9) in Equation (4.8) results in the following additional system of equations for the interface problem.

$$\begin{aligned}\lambda_{2\to1} + \lambda_{1\to2} &= u_1 Y_{1,2},\\ \lambda_{2\to1} + \lambda_{1\to2} &= u_2 Y_{1,2},\\ \text{with,} \quad Y_{1,2} &= \mathrm{i}\omega\big(Z_1^{-1} + Z_2^{-1}\big),\end{aligned} \tag{4.10}$$

where Z_1, Z_2 are the characteristic impedances in the respective medium.

The variational formulation is now presented.

4.2.1.1 Variational form

Equation (4.6) with the definitions from Equation (4.7), Equation (4.9) is introduced in the weak form of the subdomain problem.

$$\forall q_1 \in V_1, \quad \frac{1}{\rho_1}\int_{\Omega_1} \boldsymbol{\nabla} q_1^* \cdot \boldsymbol{\nabla} u_1 - k_1^2 q_1^* u_1 \,\mathrm{d}\Omega + \frac{\mathrm{i}\omega}{Z_1}\int_\Gamma q_1^* p_1 \mathrm{dS} - \frac{1}{\rho_1}\int_{\partial\Omega_1\setminus\Gamma} q_1^* \frac{\partial u_1}{\partial n_1}\mathrm{dS} = \int_\Gamma q_1^* \lambda_{2\to1}\mathrm{dS}, \tag{4.11a}$$

$$\forall q_2 \in V_2, \quad \frac{1}{\rho_2}\int_{\Omega_2} \boldsymbol{\nabla} q_2^* \cdot \boldsymbol{\nabla} u_2 - k_2^2 q_2^* u_2 \,\mathrm{d}\Omega + \frac{\mathrm{i}\omega}{Z_2}\int_\Gamma q_2^* p_2 \mathrm{dS} - \frac{1}{\rho_2}\int_{\partial\Omega_2\setminus\Gamma} q_2^* \frac{\partial u_2}{\partial n_2}\mathrm{dS} = \int_\Gamma q_2^* \lambda_{1\to2}\mathrm{dS}. \tag{4.11b}$$

The additional set of equations on the interface are integrated using the test functions $\psi_{2\to1}$ and $\psi_{1\to2}$, associated to dual variables:

$$\int_\Gamma \psi_{2\to1}^* \Big[\lambda_{1\to2} + \lambda_{2\to1} - Y_{1,2}u_1\Big]\mathrm{dS} = 0, \tag{4.12a}$$

$$\int_\Gamma \psi_{1\to2}^* \Big[\lambda_{1\to2} + \lambda_{2\to1} - Y_{1,2}u_2\Big]\mathrm{dS} = 0. \tag{4.12b}$$

An interesting remark is that the variational form presented in Equation (4.11a), Equation (4.11b), Equation (4.12a), Equation (4.12b) are exactly equivalent to the original FETI-2LM formulation for a homogeneous fluid. It can thus be said that the current formulation is a generalization of the original Helmholtz problem applied to several fluids.

4.2.1.2 Numerical discretization

This section presents the numerical discretization of the variational form. The subdomain matrices are built using higher-order finite elements, abbreviated as p-FEM. The discretization of Equation (4.11) leads to a system of the form:

$$\begin{aligned} \mathbf{K}_1\mathbf{u}_1 &= \mathbf{f}_1 + \mathbf{B}_{2,1}^{\mathrm{T}}\, \mathbf{M}_{1,2}\, \boldsymbol{\lambda}_{2\to1}, \\ \mathbf{K}_2\mathbf{u}_2 &= \mathbf{f}_2 + \mathbf{B}_{1,2}^{\mathrm{T}}\, \mathbf{M}_{1,2}\, \boldsymbol{\lambda}_{1\to2}, \end{aligned} \tag{4.13}$$

where the definitions of the operators are the same as described in Chapter 3. The mass matrix on the interface remains the same owing to compatible meshes and interpolation. The Lagrange multipliers are scaled with the mass matrices described in detail in Chapter 3.

$$\tilde{\boldsymbol{\lambda}}_{2\to1} = \mathbf{M}_{1,2}\boldsymbol{\lambda}_{2\to1}, \quad \tilde{\boldsymbol{\lambda}}_{1\to2} = \mathbf{M}_{1,2}\boldsymbol{\lambda}_{1\to2}. \tag{4.14}$$

The additional system of equations to compute the dual variables can be formulated as below:

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_{2\to1} + \tilde{\boldsymbol{\lambda}}_{1\to2} &= Y_{1,2}\mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{u}_2, \\ \tilde{\boldsymbol{\lambda}}_{2\to1} + \tilde{\boldsymbol{\lambda}}_{1\to2} &= Y_{1,2}\mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{u}_1. \end{aligned} \tag{4.15}$$

Using Equation (4.13) it is possible to eliminate the pressure variables $\mathbf{u}_1$ and $\mathbf{u}_2$ to derive a so-called interface problem for the modified Lagrange multipliers:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} - Y_{1,2}\, \mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{B}_{2,1}^{\mathrm{T}} \\ \mathbf{I} - Y_{1,2}\, \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{B}_{1,2}^{\mathrm{T}} & \mathbf{I} \end{bmatrix}}_{\mathbf{F}_{\mathrm{I}}} \underbrace{\begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{2\to1} \\ \tilde{\boldsymbol{\lambda}}_{1\to2} \end{bmatrix}}_{\tilde{\boldsymbol{\lambda}}} = \underbrace{Y_{1,2} \begin{bmatrix} \mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{f}_2 \\ \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{f}_1 \end{bmatrix}}_{\mathbf{d}}. \tag{4.16}$$

Each media can be partitioned into multiple non-overlapping subdomains and the extension to multiple subdomains is straightforward. The resulting matrix $\mathbf{F}_{\mathrm{I}}$ is square, complex composed of full blocks. It is of size $2\mathrm{N}_{\mathrm{I}} \times 2\mathrm{N}_{\mathrm{I}}$, where N_{I} is the number of degrees of freedom on the interface. The resulting interface matrix is solved using ORTHODIR. Details of the algorithm and matrix-vector products are described in detail in Section 3.2.1.3. This completes the generalized FETI-2LM formulation to model several fluids.

4.2.2 FETI-H

In this subsection, the extension of FETI-H to problems with several equivalent fluids are presented. The coupling conditions used can be expressed as follows:

$$\begin{aligned} \frac{\partial u_1}{\partial n_1} + \mathcal{D}_1 u_1 &= \rho_1 \mathcal{L}_{2\to 1} u_1, \\ \text{where,} \qquad \mathcal{L}_{2\to 1} u_1 &= -\frac{\rho_1}{\rho_2}\frac{\partial u_2}{\partial n_2} + \mathcal{D}_1 u_2, \end{aligned} \tag{4.17}$$

where $\mathcal{D}_1$, $\mathcal{L}_{2\to 1}$ are the absorbing and transmission operators respectively. The choice of using the same absorbing operator on either side of the interface comes as a constraint seen in Chapter 3. Rearranging terms and imposing the continuity constraints, the interface problem reduces to the following:

$$u_1 - u_2 = 0 \ \text{ on } \ \Gamma. \tag{4.18}$$

For the extension to multiple subdomains, sign conventions are introduced as described in Section 3.2.2.

4.2.2.1 Variational form

The variational form corresponding to FETI-H algorithm with sign conventions can be described as follows:

$$\begin{aligned} \forall q_1 \in V_1, \quad & \frac{1}{\rho_1}\int_{\Omega_1} \boldsymbol{\nabla} q_1^* \cdot \boldsymbol{\nabla} u_1 - k_1^2 q_1^* u_1 \, \mathrm{d}\Omega + \frac{\mathrm{i}\omega\epsilon_{1,2}}{Z_1}\int_\Gamma q_1^* u_1 \mathrm{dS} - \epsilon_{1,2}\int_\Gamma q_1^* \lambda_{2\to 1}\mathrm{dS} = \\ & \qquad \frac{1}{\rho_1}\int_{\partial\Omega_1\setminus\Gamma} q_1^* \frac{\partial u_1}{\partial n_1}\mathrm{dS}, \\ \forall q_2 \in V_2, \quad & \frac{1}{\rho_2}\int_{\Omega_2} \boldsymbol{\nabla} q_2^* \cdot \boldsymbol{\nabla} u_2 - k_2^2 q_2^* u_2 \, \mathrm{d}\Omega + \frac{\mathrm{i}\omega\epsilon_{2,1}}{Z_2}\int_\Gamma q_2^* u_2 \mathrm{dS} - \epsilon_{2,1}\int_\Gamma q_2^* \lambda_{1\to 2}\mathrm{dS} = \\ & \qquad \frac{1}{\rho_2}\int_{\partial\Omega_2\setminus\Gamma} q_2^* \frac{\partial u_2}{\partial n_2}\mathrm{dS}. \end{aligned} \tag{4.19}$$

The additional equation for the interface problem can be obtained by integrating Equation (4.18):

$$\forall \psi_{2\to 1} \quad \int_\Gamma \psi_{2\to 1}^* (u_1 - u_2)\mathrm{d}\Gamma = 0 \ \text{ on } \ \Gamma. \tag{4.20}$$

It can be observed that this relation is the same as the FETI-H formulation for the homogeneous fluid. This is because the terms corresponding to the absorbing operator do not appear in the interface problem. In addition, the choice of the absorbing operator only influences the subdomain integrals without affecting the integral terms on the interface.

4.2.2.2 Numerical discretization

This section presents the numerical discretization of the variational form. The subdomain matrices are built using higher-order finite element, abbreviated as p-FEM. The discretization of Equation (4.19) results in the following set of equations:

$$\begin{aligned} \tilde{\mathbf{K}}_1 \mathbf{u}_1 &= \mathbf{f}_1 + \epsilon_{1,2} \mathbf{B}_{2,1}^{\mathrm{T}} \, \mathbf{M}_{1,2} \, \boldsymbol{\lambda}_{2\to1}, \\ \tilde{\mathbf{K}}_2 \mathbf{u}_2 &= \mathbf{f}_2 + \epsilon_{2,1} \mathbf{B}_{1,2}^{\mathrm{T}} \, \mathbf{M}_{1,2} \, \boldsymbol{\lambda}_{2\to1}, \end{aligned} \tag{4.21}$$

where the definitions of the matrices and vectors are the same as described in the FETI-2LM algorithm. The remaining details about the methodology remain the same as described in Section 3.2.2.

The interface problem for the two subdomain case expressed in terms of the dual variables is as follows:

$$\underbrace{\begin{bmatrix} \epsilon_{1,2}\mathbf{B}_{2,1}\mathbf{K}_1^{-1}\mathbf{B}_{2,1}^{\mathrm{T}} - \\ \epsilon_{2,1}\mathbf{B}_{1,2}\mathbf{K}_2^{-1}\mathbf{B}_{1,2}^{\mathrm{T}} \end{bmatrix}}_{\mathbf{F}_{\mathrm{I}}} \underbrace{\begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{2\to1} \end{bmatrix}}_{\tilde{\boldsymbol{\lambda}}} = \underbrace{\begin{bmatrix} -\mathbf{B}_{2,1}\mathbf{K}_1^{-1}\mathbf{f}_1 + \\ \mathbf{B}_{1,2}\mathbf{K}_2^{-1}\mathbf{f}_2 \end{bmatrix}}_{\mathbf{d}}. \tag{4.22}$$

The resulting interface matrix $\mathbf{F}_{\mathrm{I}}$ is square, complex-symmetric, non-Hermitian, sparse and of size $\mathrm{N_I} \times \mathrm{N_I}$. It is important to note that the interface matrix is independent of the effective admittance and also of the mass matrix computed on the interface. Another crucial feature of this FETI-H algorithm is that the interface problem is exactly the same as in the homogeneous case without any alterations.

4.3 Results in 2D

In this section, the results for the verification and performance of the FETI variants namely, FETI-2LM and FETI-H applied to problems with several different fluids are presented. In the first part, the numerical verification of the h-p convergence for a domain composed of two fluids on a benchmark 2D problem are presented. The second test case focuses on the comparison of the aforementioned FETI methods with scalability tests. In the last part, the implementation of one of these FETI-variants is verified for several fluids.

4.3.1 Test case description

A 2D guided wave problem with a 1D analytical solution is considered. The computational domain is a square cavity with unit dimensions, as illustrated in Figure 4.2. This physical domain consists of two fluids with a physical interface Γ located at $x = 0.5\,\mathrm{m}$. The part on the left of Γ is composed of air and the right part is a porous material. The porous material is modeled as an equivalent fluid

using the JCA model. The details of this model are detailed in Section 2.4.2. The parameters of the porous materials are presented in Table 4.1. Two different porous materials with varying properties are considered.

The computational domain is discretized with triangular finite elements using Gmsh [46]. An example is given in the left part of Figure 4.2. An essential requirement of the FETI method is that the generated meshes should be conformal on the physical interface Γ, which allows implementing the coupling conditions between the two physical domains.

A plane wave propagating in the positive x direction is injected from the left boundary.

$$\frac{\partial u_1}{\partial n_1} + \mathrm{i}k_1 u_1 = g \ , \quad \text{for } x = 0 \ ,$$

in which g is a forcing term. An absorbing boundary condition is imposed on the right boundary in the porous medium using the following Robin condition:

$$\frac{\partial u_2}{\partial n_2} + \mathrm{i}k_2 u_2 = 0 \ , \quad \text{for } x = 1 \ .$$

The upper and lower boundaries are hard walls ($\partial u/\partial n = 0$). Since the objective of this test case is to verify the interface transmission conditions, each domain is composed of a single partition. The subdomain matrices are generated using higher-order p-FEM basis as described in Section 2.2. The interface problem defined in Equation (4.16) is solved using ORTHODIR for a set tolerance $\epsilon = 10^{-8}$. The primal solution is recovered using the local subdomain operators. The Figure 4.2 on the right illustrates the numerical solution for the pressure field for $\omega = 6.91 \times 10^4$ rad/s. It can be seen that the porous material damps the incoming pressure waves. The continuity of the primal variables on the physical interface are also verified.

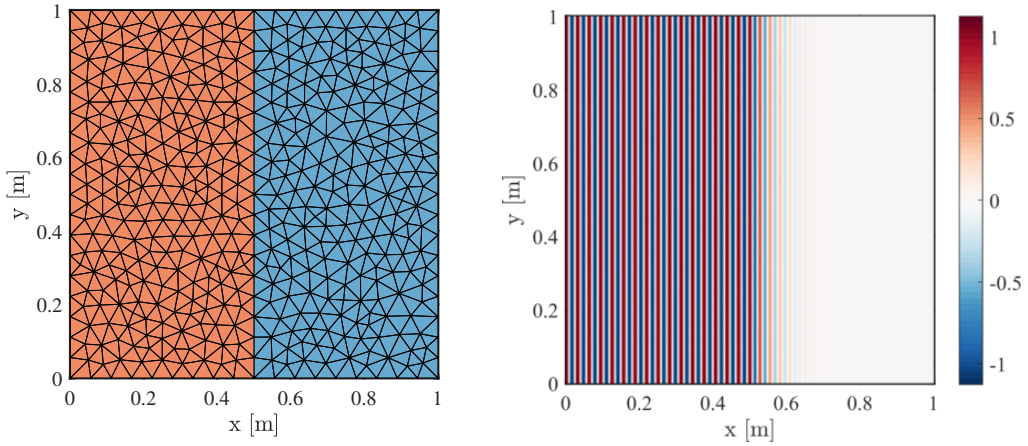


Figure 4.2: (a) One partition for each medium with a conformal interface mesh (Γ at $x = 0.5\,\mathrm{m}$), (b) Real part of the pressure in Pa plotted for a plane wave at $\omega = 6.91 \times 10^4$ rad/s.

The performance of the method is examined by computing the relative L_2 norm of the error. The number of degrees of freedom per wavelength is computed taking into account the largest wavenumber from both fluids [61].

$$D_\lambda = \frac{2\pi}{kL}\left(\sqrt{N_{\text{DOF}}} - 1\right), \qquad k = \max(|k_1|, |k_2|), \tag{4.23}$$

where $L = \sqrt{\text{area}(\Omega)}$ is a typical length scale of the computational domain and k_1 and k_2 are the wavenumbers in the air and the porous material respectively. In order to ensure that the two waves are well resolved, we consider the maximum wavenumber in this definition.

In Figure 4.3, the relative error is plotted as a function of D_λ for varying p orders, considering the two porous materials defined in Table 4.1. For each material, two cases are considered, the duct is composed of two parts (air and porous material), or the duct is completely filled with porous material. Each graph shows the convergence plots for a homogeneous and a heterogeneous medium, and for different polynomial orders. In each situation, an asymptotic convergence rate of $D_\lambda^{-(p+1)}$ for the relative L_2 norm of the error is retrieved, which is in agreement with the theoretical [6] and numerical [9] results. It can thus be said that the convergence rate in the asymptotic regime remains equivalent for a domain composed of a single or two different fluids, which verifies the proposed implementation.

Table 4.1: Physical properties for air medium, frequency dependent properties for (a) porous material 1, (b) porous material 2 used to estimate the equivalent density and equivalent speed of sound for $\omega = 1000\,\text{rad/s}$.

Properties	Air	Porous material 1	Porous material 2
Porosity ϕ (-)	-	1	0.8
Flow resistivity σ (Nsm^{-4})	-	1.06×10^4	5×10^4
Tortuosity α_∞ (-)	-	1.2	1.1
Thermal characteristic length Λ' (m)	-	240×10^{-6}	240×10^{-6}
Viscous characteristic length Λ (m)	-	490×10^{-6}	490×10^{-6}
Speed of sound c (m/s)	341.97	76.15 +74.34i	37.12 + 40.56i
Density ρ (kg/m^3)	1.21	1.56 -10.57i	1.70 - 50i

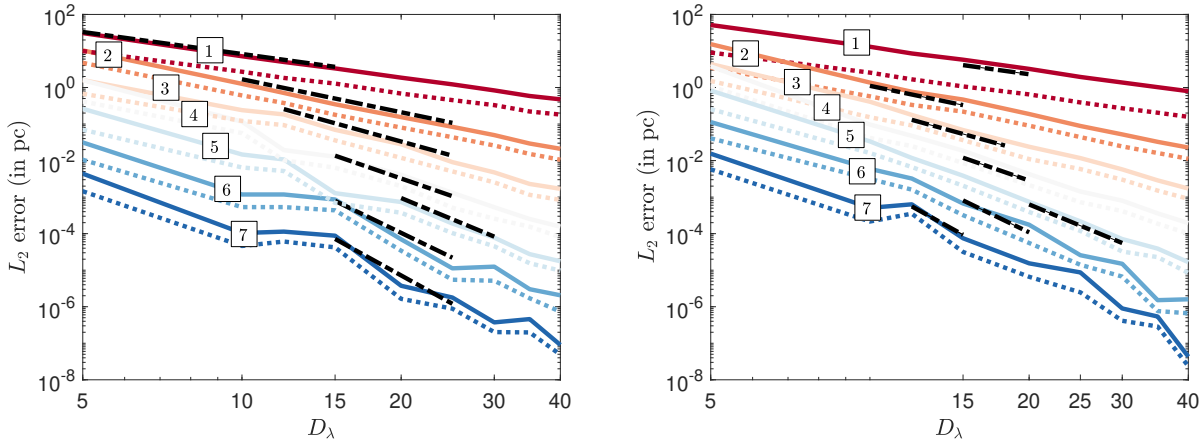


Figure 4.3: Convergence of the L_2 norm of the error for varying D_λ for porous material 1 (a) and 2 (b) respectively. —— represents the test case with air and porous material, ····· represents the homogeneous case with only the porous material, -·-·- represents the slopes of order $-(p+1)$.

4.3.2 Scalability results in 2D

In this section, the performance and behavior of the FETI methods are presented. The analysis is performed for a domain comprising of two fluids, one of them is the air and the other is a porous material modeled as an equivalent fluid. The method to analyze the scalibility is the same as the one presented in Section 3.3.1.

The notation of this section are based on the ones of the Chapter 3 with an index a or p to refer to the domain (air or porous) to which they correspond. Let N_a, N_p be the number of subdomains, k_a, k_p be the wavenumber, D_a, D_p be the degrees of freedom per wavelength, E_a, E_p be the relative L_2 norm of the error (in %) in the air and the porous media respectively computed for known analytical solutions.

4.3.2.1 Influence of the number of subdomains

In this part, the behavior of both FETI variants is examined for different numbers of partitions in both air and porous medium for a fixed mesh size, polynomial order and frequency. Each medium is partitioned using the graph partitioning tool, METIS, (see Figure 4.4 and Figure 4.5). In Figure 4.4, the partitions in the air medium, N_a are varied from 2 to 8 for a fixed number of partitions in the porous medium, N_p=2. In Figure 4.5, the partitions in the porous medium, N_p are varied from 2 to 8 for a fixed number of partitions in the air medium, N_a=2. A remark is that the number of partitions in a particular medium can be varied without affecting the discretization or partitioning of the other media.

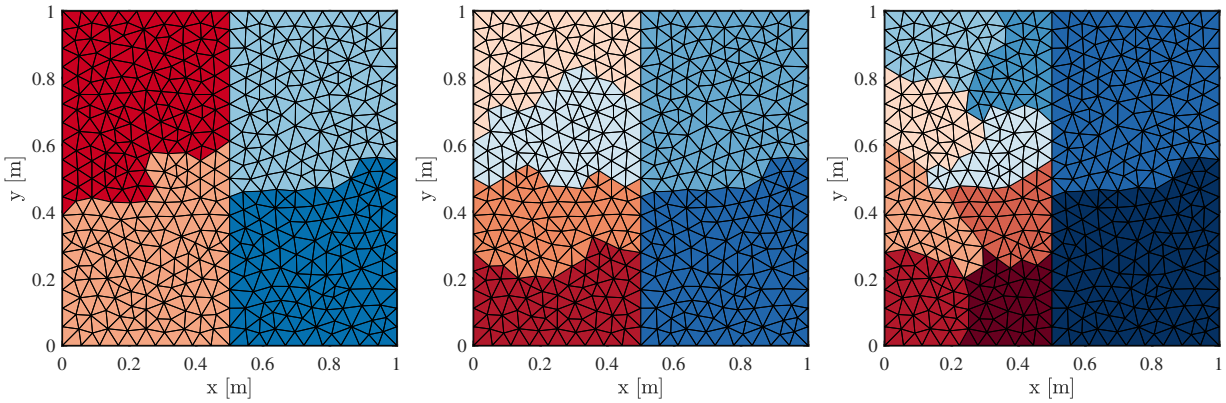


Figure 4.4: Varying partitions in the air medium while keeping the same number of partitions in the porous medium.

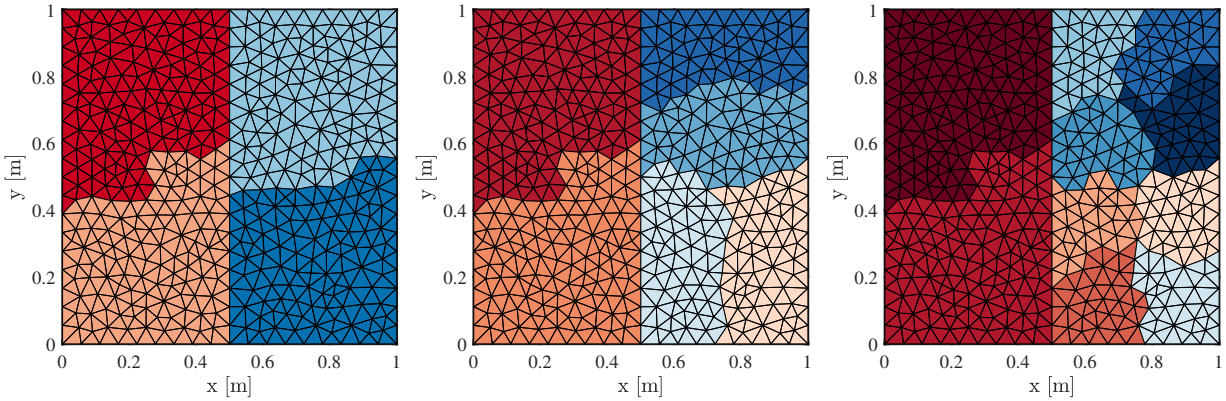


Figure 4.5: Varying partitions in the porous medium while keeping the same number of partitions in the air medium.

For the above described partitions, the results for the FETI-H algorithm are presented in Table 4.2. All examples have the same accuracy, $E_a = 5.1{\times}10^{-5}$ %, $E_p = 1.57 \times 10^{-4}$ % and same wavelength resolution, D_a=23, D_p=19. The first observation is that the number of iterations increases drastically (319 to 839) when the number of partitions in the air medium grows. However, the increase in the number of iterations is slightly less for varying partitions in the porous medium (319 to 578), compared to the air medium. In general, for increasing partitions in either medium at the same accuracy, the FETI-H algorithm is not scalable. There is a slight variance in the total number of DOFs due to the increasing number of partitions. The convergence of the normalized residual can be seen in Figure 4.6. The convergence plots has 2 different convergence regimes similar to the regimes reported for the homogeneous case results for FETI-H, described in Section 3.3.1.2. For varying partitions in the air medium, the pre-convergence regime dominates the computational effort as large numbers of iterations are consumed in this region. However, for varying partitions in the porous medium, the pre-convergence regime is quite small (compared to varying partitions in the air medium) and thus

the computational effort is slightly reduced.

Table 4.2: Results for FETI-H: (a) Varying number of partitions in air medium for $N_p = 2$, (b) Varying number of partitions in porous medium for $N_a = 2$. All cases have a fixed accuracy $E_a = 5.1\times10^{-5}$ %, $E_p = 1.57\times10^{-4}$ %, $p = 6$, h/L =1/30, and $k_aL = 100$.

N_a	N_I	N_{DOF}	n_c
2	814	171,274	319
4	1299	171,756	637
8	1717	172,168	839

N_p	N_I	N_{DOF}	n_c
2	814	171274	319
4	1269	171726	496
8	1670	172120	578

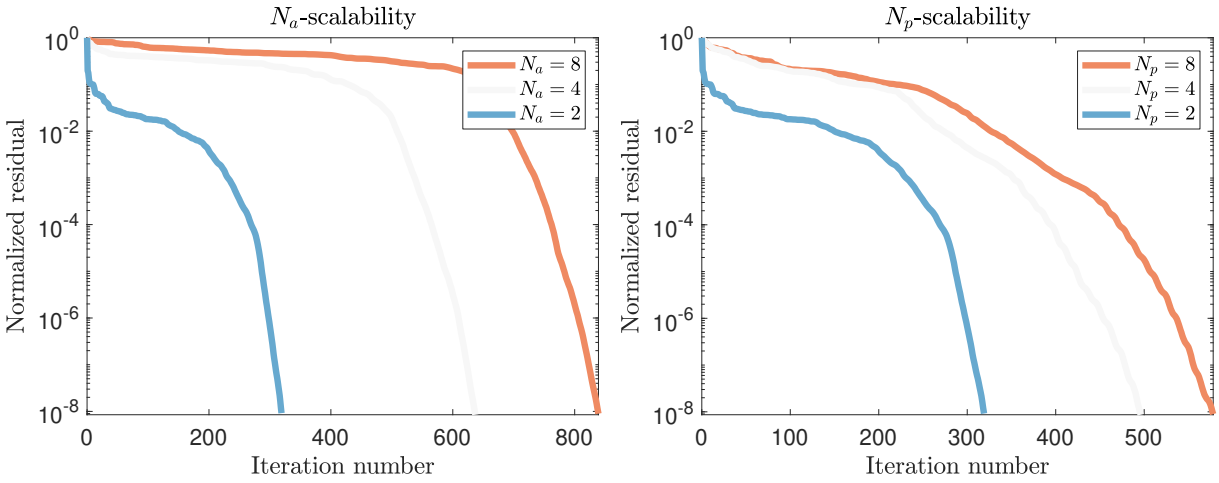


Figure 4.6: Effect of varying subdomains for FETI-H (a) N_a varying from 2 to 8, for $N_p = 2$, (b) N_p varying from 2 to 8, for $N_a = 2$. Other parameters $h/L = 1/60$, $p = 6$, $k_aL = 100$, $\epsilon = 10^{-8}$ are fixed.

In this part, the results for the FETI-2LM algorithm are presented for varying partitions, see Figure 4.4 and Figure 4.5. All the examples have the same accuracy and wavelength resolution as mentioned for the FETI-H case. The preliminary observation is that the number of iterations increases (147 to 251) with varying partitions in the air medium. Contrarily, for varying partitions in the porous medium, the FETI-2LM algorithm shows close to perfect strong scalability, i.e. the number of iterations remains almost constant for increasing partitions. It should be noted that this strong scalability is observed in the absence of any coarse-space preconditioner.

Table 4.3: Results for FETI-2LM: (a) Varying number of partitions in air medium for $N_p = 2$, (b) Varying number of partitions in the porous medium for $N_a = 2$. All cases have a fixed accuracy $E_a = 5.1\times10^{-5}$ %, $E_p = 1.57 \times10^{-4}$ %, $p = 6$, $h/L = 1/30$, and $k_aL = 100$.

N_a	N_I	N_{DOF}	n_c
2	1628	171,274	147
4	2598	171,756	208
8	3434	172,168	251

N_p	N_I	N_{DOF}	n_c
2	1628	171,274	147
4	2538	171,726	149
8	3340	172,120	150

The convergence of the normalized residual can be seen in Figure 4.7. In FETI-2LM, there is no clear distinction between the pre-convergence and the convergence regime as observed in FETI-H algorithms. For varying partitions in the air medium, the convergence rate varies only slightly in the pre-convergence regime. However, in the convergence regime, the convergence rate drops with increasing partitions. In contrast, for varying partitions in the porous medium, the convergence rate remains the same in both regimes. The perfect scalability for varying partitions in the porous medium is a behavior peculiar to equivalent fluids and in general, might not be observed for fluids with non-dissipative properties.

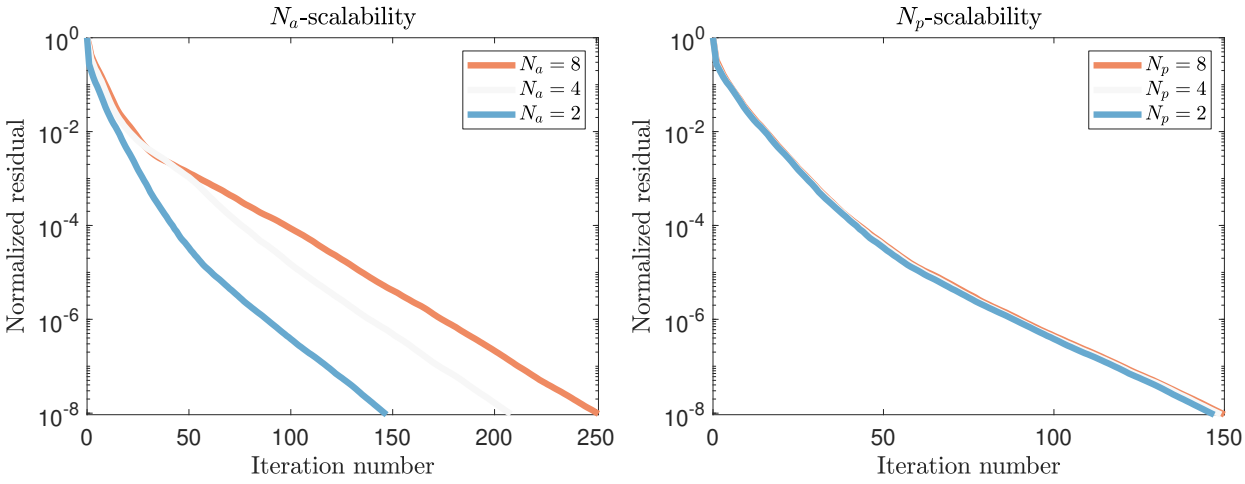


Figure 4.7: Effect of varying subdomains for FETI-2LM algorithm (a) partitions in air medium N_a are varying, for fixed number of partitions $N_p = 2$, (b) partitions in porous medium N_p are varying, for fixed number of partitions $N_a = 2$. Other parameters $h/L = 60$, $p = 6$, $k_aL = 100$, $\epsilon = 10^{-8}$ are fixed.

4.3.2.2 Effect of the mesh refinement

In this part, the mesh size is varied for a fixed polynomial order, partitioning and frequency. The normalized residual is plotted for varying iterations for FETI-2LM and FETI-H algorithms as seen in Figure 4.8, Table 4.4, and Table 4.5.

In Table 4.4, the problem is under-resolved for the first example and thus a high relative error is observed. This error drops asymptotically when the mesh is refined. It is interesting to observe that for the first two examples the numbers of iterations are very similar to the problem size. This is due to the fact that FETI-H consumes a large number of iterations in the pre-convergence regime. As the mesh is further refined, the number of iterations required for convergence also increases, however without requiring to solve the full size of the interface problem. For the latter two examples, the slope in the convergence regime decreases when increasing the mesh refinement.

Table 4.4: Effect of mesh refinement for FETI-H, for $N_a = 3$, $N_p = 2$, $p = 6$, $\epsilon = 10^{-8}$, $k_a L = 100$.

h/L	$\mathrm{N_I}$	$\mathrm{N_{DOF}}$	D_a	D_p	n_c	E_a in %	E_p in %
10	163	5033	4	3	160	26.41	43.31
15	271	11543	6	5	268	1.07	2.15
30	517	43967	11	10	446	5.2×10^{-3}	1.7×10^{-2}
60	1045	171,503	23	19	552	5.09×10^{-5}	1.57×10^{-4}

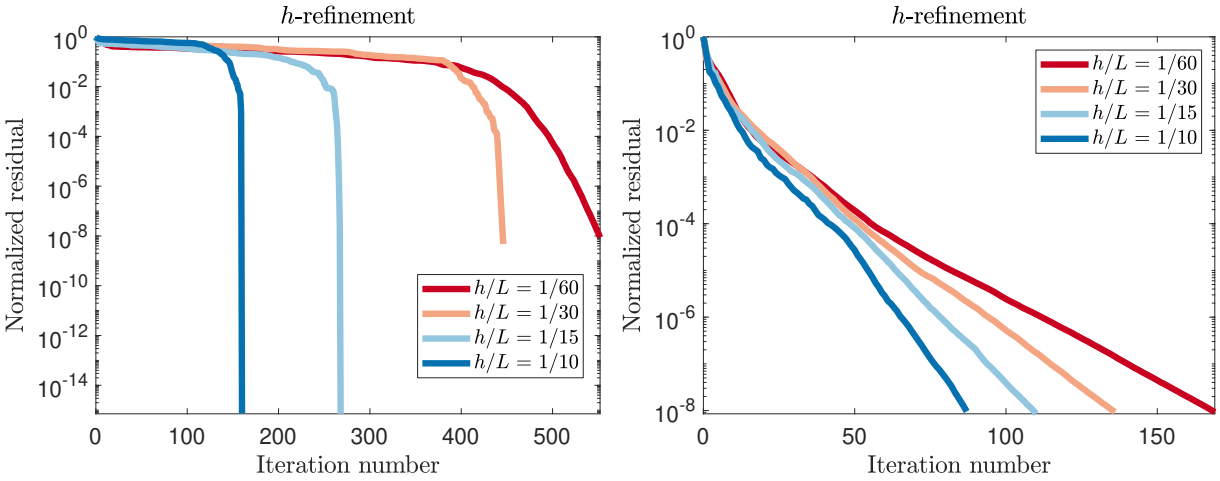


Figure 4.8: Effect of mesh refinement, left: FETI-H, right: FETI-2LM for $N_a = 3$, $N_p = 2$, $k_a L = 100$, $p = 6$, $\epsilon = 10^{-8}$.

In Table 4.5, the problem is under-resolved in both media for the first example which results in high relative error. However, in general the number of iterations increase moderately (87 to 169) as compared to FETI-H with the same mesh refinement. A general remark is that roughly twice or more iterations are required for FETI-H compared to FETI-2LM to achieve the same accuracy, for increasing mesh refinement.

The pre-convergence regime is quite similar whereas the slope in the convergence regime reduces

when increasing the mesh refinement for all examples. In general, it can be said that the FETI-2LM algorithm scales much better than FETI-H.

Table 4.5: Effect of mesh refinement for FETI-2LM, for $N_a = 3$, $N_p = 2$, $p = 6$, $\epsilon = 10^{-8}$, $k_aL = 100$.

h/L	N_I	N_{DOF}	D_a	D_p	n_c	E_a in %	E_p in %
10	326	5033	4	3	87	26.41	43.31
15	542	11543	6	5	110	1.07	2.15
30	1034	43967	11	10	136	5.2×10^{-3}	1.7×10^{-2}
60	2090	171,503	23	19	169	5.09×10^{-5}	1.57×10^{-4}

4.3.2.3 Effect of the interpolation order

In this part, the polynomial order p is varied for a fixed mesh resolution, partitioning and frequency for both FETI-2LM and FETI-H algorithms as seen in Figure 4.9, Table 4.6 and Table 4.7.
For the FETI-H algorithm, the first observation is that the error roughly drops by 7 orders of magnitude from p=2 to p=7, which is consistent with the theoretical convergence rate for classical p-FEM.
The total number of DOFs increases drastically (from p=1 to p=7), whereas the size of the interface problem grows only slightly. The number of iterations increases considerably (147 to 614) compared to the increase in the size of the interface problem, however it was observed that for the same accuracy the number of iterations for convergence do not change drastically. Thus, it is judicious to say that the FETI-H algorithm scales with p-refinement. The slope in the convergence regime seems to have a slight dependency on the polynomial order, as the order p increases, the slope in the convergence regimes reduces.

Table 4.6: Effect of p-refinement for FETI-H, for $N_a = 3$, $N_p = 2$, $\epsilon = 10^{-8}$, $h/L = 1/60$, $k_aL = 100$.

p	N_I	N_{DOF}	D_a	D_p	n_c	E_a in %	E_p in %
1	180	5013	4	3	147	147.12	183.29
2	353	19451	8	6	269	9.89	17.56
3	526	43319	11	10	342	0.21	0.4
4	699	76617	15	13	414	0.01	0.03
5	872	119,345	19	16	487	7.93×10^{-4}	2.11×10^{-3}
6	1045	171,503	23	19	552	5.09×10^{-5}	1.57×10^{-4}
7	1218	233,091	26	23	614	3.98×10^{-6}	1.08×10^{-5}

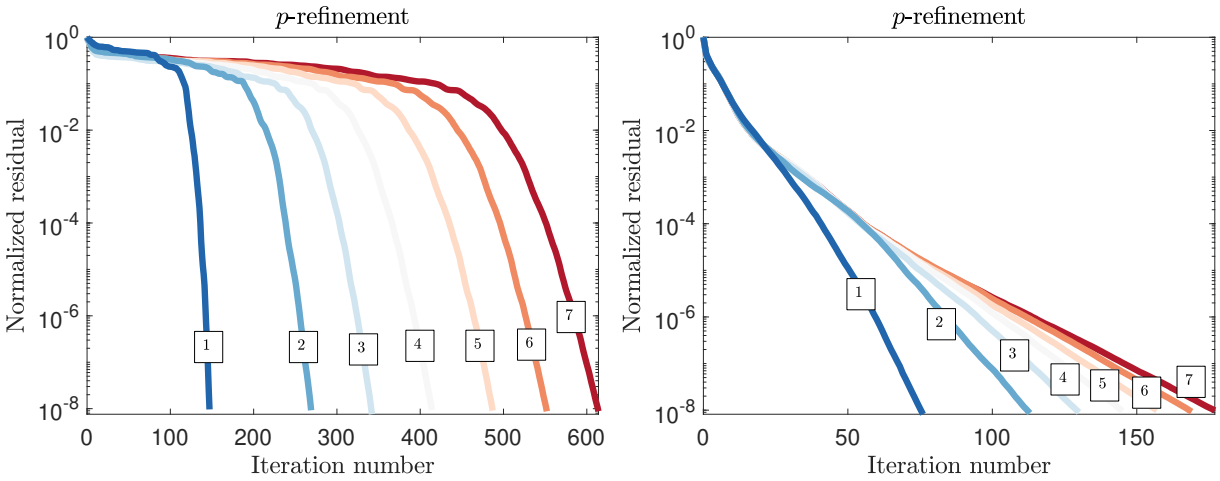


Figure 4.9: Effect of p-refinement, (a) FETI-H, (b) FETI-2LM for $N_a = 3$, $N_p = 2$, $k_aL = 100$, $\epsilon = 10^{-8}$, $h/L = 1/60$.

Table 4.7: Effect of p-refinement for FETI-2LM, for $N_a = 3$, $N_p = 2$, $\epsilon = 10^{-8}$, $h/L = 1/60$, $k_aL = 100$.

p	N_I	N_{DOF}	D_a	D_p	n_c	E_a in %	E_p in %
1	360	5013	4	3	76	147.12	183.29
2	706	19451	8	6	113	9.89	17.56
3	1052	43319	11	10	130	0.21	0.4
4	1398	76617	15	13	145	0.01	0.03
5	1744	119,345	19	16	157	7.93×10^{-4}	2.11×10^{-3}
6	2090	171,503	23	19	169	5.10×10^{-5}	1.57×10^{-4}
7	2436	233,091	26	23	177	6.14×10^{-6}	1.11×10^{-5}

The FETI-2LM algorithm scales much better compared to FETI-H in terms of number of iterations (76 to 177) as seen in Table 4.7. In the pre-convergence regime, all the curves fall into one with the same convergence rate. However, the slope in the convergence regime seems to have a slight dependency on the polynomial order, as the p-order increases, the slope in the convergence regimes reduces. In general, FETI-H requires roughly twice or more iterations for convergence as compared to FETI-2LM, for the same accuracy of the primal solution.

4.3.2.4 Influence of the frequency

In this part, the angular frequency is varied for a fixed mesh resolution, polynomial order and number of partitions. The properties in the porous medium are frequency dependent modeled using the JCA model. Since the frequency remains the same in both media, it suffices to use this parameter to

examine the results presented in Figure 4.10 and Table 4.8, Table 4.9 for both FETI-2LM and FETI-H algorithms.

As seen in Table 4.8, the number of iterations required for convergence increases with the angular frequency. This is expected of the FETI-H algorithm and observed in the results of the homogeneous case as well. As the angular frequency increases, the accuracy of the algorithm reduces since the problem resolution decreases. For very high angular frequencies, almost the complete subspace of the interface problem needs to be computed, which is computationally expensive. The convergence plots of the residual error highlight the high computational effort spend in the pre-convergence regime with increasing angular frequencies.

As seen in Table 4.9, the number of iterations required for convergence for the same sized interface problem reduces with increasing angular frequency. This phenomenon is specific to the case of equivalent fluids, since for a homogeneous fluid, the iteration count increases with increasing frequency. This is in agreement with similar work using OSM [24], where the convergence rate was found to improve for heterogeneous media. It can be observed from Figure 4.10 that the slope in the convergence regime improves with increasing angular frequency, contrary to the homogeneous fluid case. This is a remarkable result since it justifies modeling large wavenumbers in the air medium without an increase in the computational cost.

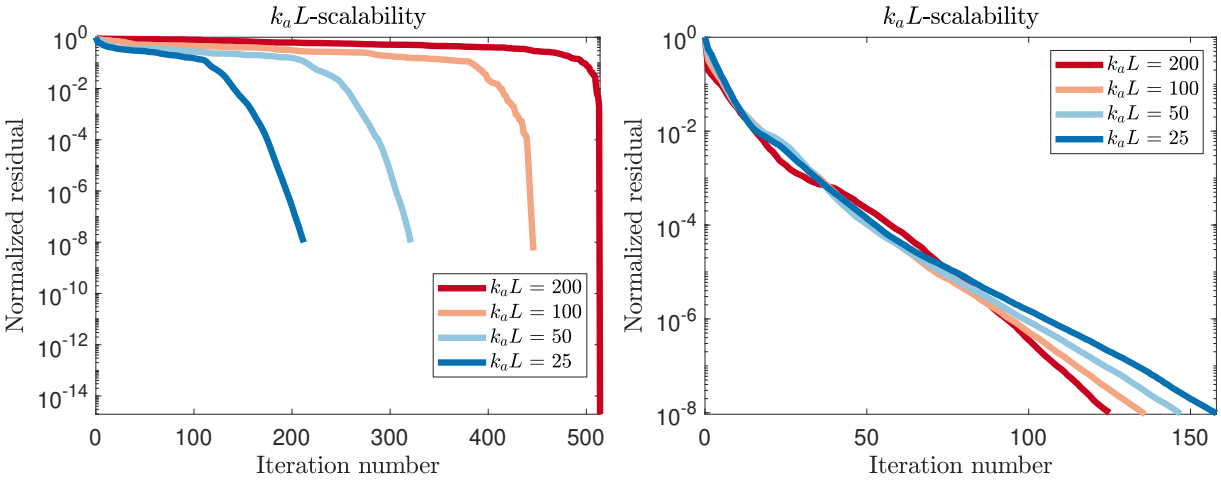


Figure 4.10: Effect of wavenumbers on the iteration count for (a) FETI-H, (b) FETI-2LM, $N_a = 3$, $N_p = 2$, $h/L = 1/30$, $p = 6$, $k_aL = 100$, $\epsilon = 10^{-8}$.

Table 4.8: Effect of varying wavenumbers for FETI-H, $N_a = 3$, $N_p = 2$, $h/L = 1/30$, $p = 6$, $\epsilon = 10^{-8}$.

ω	N_I	N_{DOF}	D_a	D_p	n_c	E_a in %	E_p in%
8549	517	43967	45	34	212	1.14×10^{-6}	2.82×10^{-6}
17099	517	43967	23	19	321	4.31×10^{-5}	1.8×10^{-4}
34197	517	43967	11	10	446	5.2310^{-3}	0.02
68395	517	43967	6	5	514	1.13	2.28

Table 4.9: Effect of varying wavenumbers for FETI-2LM, $N_a = 3$, $N_p = 2$, $h/L = 1/30$, $p = 6$, $\epsilon = 10^{-8}$.

ω	N_I	N_{DOF}	D_a	D_p	n_c	E_a in %	E_p in%
8549	1034	43967	45	34	158	4.35×10^{-6}	9.36×10^{-6}
17099	1034	43967	23	19	147	4.32×10^{-5}	1.8×10^{-4}
34197	1034	43967	11	10	136	5.29×10^{-3}	0.02
68395	1034	43967	6	5	125	1.13	2.28

A systematic analysis of FETI-2LM and FETI-H methods has been presented for varying problem parameters like the number of partitions, mesh, polynomial order and angular frequencies. For varying subdomains in the air or the porous medium, FETI-H does not scale with the iteration count, however the scalability is improved for varying partitions in the porous medium. Contrarily, FETI-2LM shows almost perfect scalability with varying number of partitions in the porous medium.

For h and p refinement, both FETI-2LM and FETI-H show scalability for the iterations count, however FETI-H requires roughly twice more iterations than FETI-2LM for the same accuracy of the primal solution. For increasing frequencies, FETI-H does not scale for the iteration count, however FETI-2LM scales remarkably well. For increasing angular frequencies, the number of iterations required for convergence decreases, contrary to results observed for the homogeneous fluid case.

It is thus judicious to choose the FETI-2LM method over FETI-H for modeling equivalent fluids. The forthcoming results are presented using only the FETI-2LM method.

4.3.3 FETI-2LM applied for problems with more than two fluids

Only problems with two fluids were presented up to now. In this section, the FETI-2LM algorithm is applied to problems with more than two fluids consisting of multi-layered porous materials. The objectives of this test case are twofold: (a) verify the implementation and accuracy of the primal solution for multiple fluids with frequency dependent properties; (b) verify the implementation for arbitrary numbers of partitions introduced in each fluid.

An air cavity with a multi-layered porous material is considered, see Figure 4.11. Both porous materials are modeled using the JCA model with the properties of each of the layers described in Table 4.10. Two physical interfaces are present, the first at x=0.5 m. between air and porous material 1 and the second at x=0.7 m. between the two layers of porous materials. The same boundary conditions as described in the Section 4.3.1 are introduced on the outer boundaries of the computational domain. Each part of the computational domain corresponding to a particular fluid is split into non-overlapping subdomains using the automatic graph partitioning tool METIS [52]. As seen in the Figure 4.11, respectively 7, 2 and 4 partitions are introduced in each of the three computational regions.

Table 4.10: Physical properties for air medium, frequency dependent properties for the porous material 1 and porous material 2 used to estimate the equivalent density and equivalent speed of sound for $\omega = 18840$ rad/s .

Properties	Air	Porous material 1	Porous material 2
Porosity (-)	-	0.5	0.8
Flow resistivity (Ns/m^4)	-	1.06 $\times 10^4$	5 $\times 10^4$
Tortuosity (-)	-	1.2	1.1
Thermal characteristic length (m)	-	240$\times 10^{-6}$	240$\times 10^{-6}$
Viscous characteristic length (m)	-	490 $\times 10^{-6}$	490$\times 10^{-6}$
Speed of sound (m/s)	341.97	284.38 + 33.26i	202.48+ 115.11i
Density (kg/m^3)	1.21	3.27 - 0.66i	1.70 - 2.65i

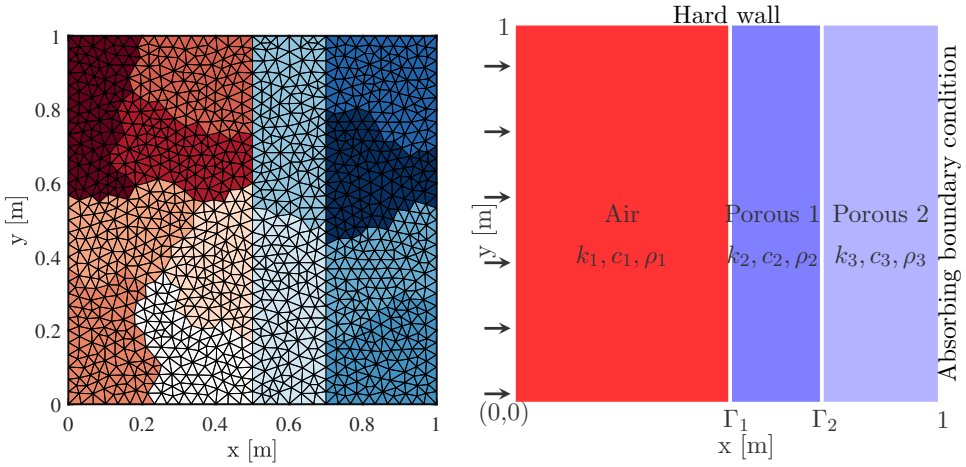


Figure 4.11: (a) 7,2,4 partitions of each media respectively done using METIS, (b) Domain with three fluids, Air - Porous 1 - Porous 2, The Porous 1 and Porous 2 are modeled using the JCA model.

The interface problem is solved using ORTHODIR with a set tolerance of $\epsilon = 10^{-8}$ for three

different types of partitioning as outlined in Table 4.11. The preliminary observation is that increasing the number of partitions has no effect on the L_2 norm of the error, in fact the relative error in % for the latter cases is reasonably small for a tolerance of 10^{-8}.

The second observation is that increasing the number of partitions typically results in an increase of the iteration count. For a five-fold rise in the number of partitions (3 to 15), the number of iterations approximately grows by a factor 4.5.

The primal solution for the pressure field is plotted in Figure 4.12. Since the first porous material has low porosity and flow resistivity, the attenuation of the pressure wave is not significant. However, as the porosity and flow resistivity increase, the pressure wave is almost completely absorbed in the second porous material. Qualitatively, it can also be seen that the pressure field is continuous across the physical interfaces between the fluids.

Table 4.11: Convergence of the iterative procedure for varying numbers of partitions in different regions for a set interface tolerance $\epsilon = 10^{-8}$.

N_s	N_I	N_{DOF}	n_c	E_{L_2} in %	Relative Error in %
1-1-1	604	30553	39	0.016	-
3-2-2	1172	30832	148	0.016	2.24×10^{-4}
7-2-4	1858	31168	181	0.016	7.64×10^{-4}

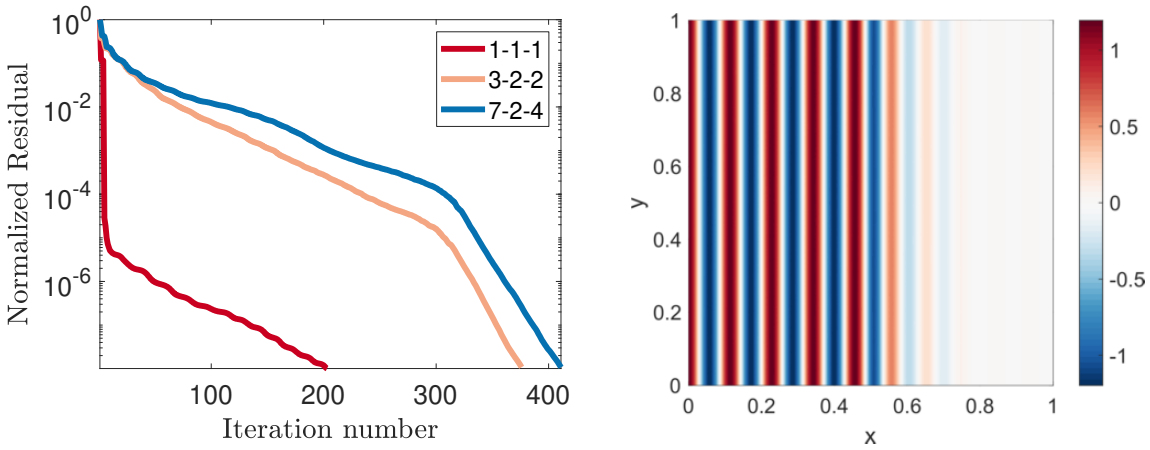


Figure 4.12: (a) Convergence of the iterative procedure for varying partitions in different fluids, (b) Numerical solution of the pressure field in Pa. for $\omega = 18840\,\mathrm{rad/s}$.

This section presented results for FETI-2LM and FETI-H for benchmark 2D problems. The scalability analysis for two fluids demonstrated how FETI-2LM outperforms FETI-H for different problem parameters. Moreover, the extension to several fluids does not affect the performance of the former FETI variant.

4.4 Results in 3D

In this section, the FETI-2LM method is applied for 3D problems with known analytical solution. The objective of this test case is to demonstrate the accuracy of the numerical implementation of the FETI-2LM method for increasing partitions for a 3D case with a known analytical solution. A cube with unit dimensions is used to model the acoustic cavity. At x=1, there is a physical interface between the duct and the acoustic treatment. The acoustic treatment consists of a melamine foam modeled using the JCA model. The numerical results for 1 kHz frequency are presented in Figure 4.13. It can be observed that the relative L_2 error (which is 1%) for a known analytical solution remains the same for increasing partitions up to 32. The number of iterations for a fixed set tolerance of $\epsilon = 10^{-8}$ increases by a modest factor of 1.6 for a 16 fold increase in the number of partitions as seen in Figure 4.14.

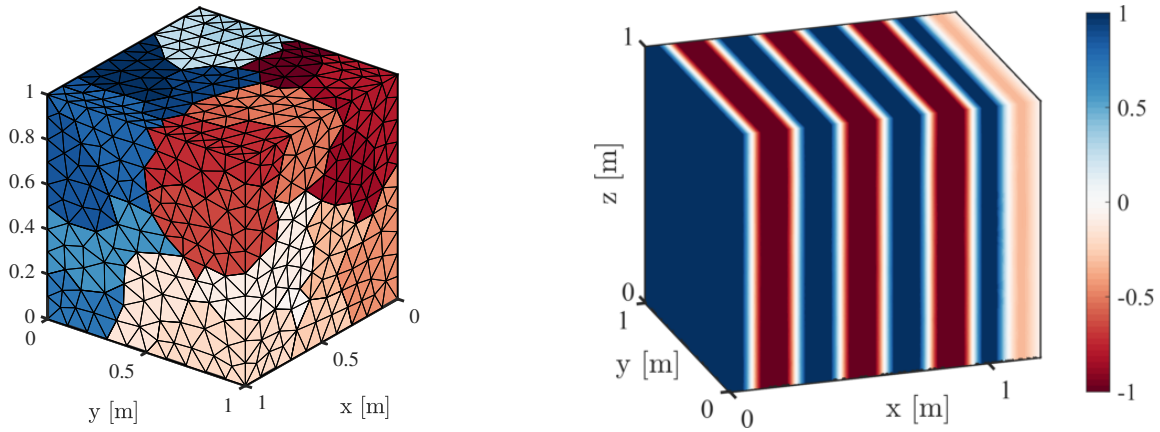


Figure 4.13: (a) Partitioning the tube cavity into 32 non-overlapping partitions using METIS, (b) Numerical solution for a plane wave at $f = 1$ kHz with 1 % accuracy.

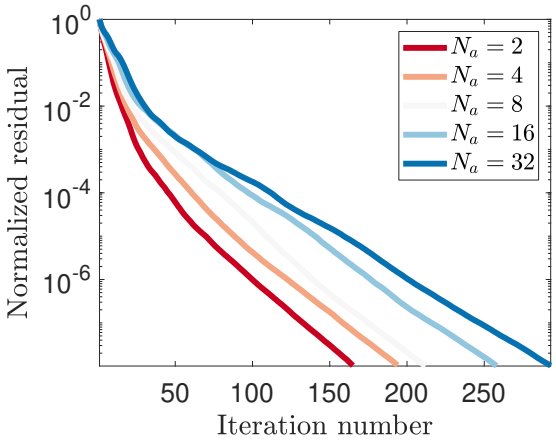


Figure 4.14: Convergence of the ORTHODIR iterative scheme for varying partitions up to 32 for $f = 1$ kHz.

4.5 FETI methods and optimization strategies

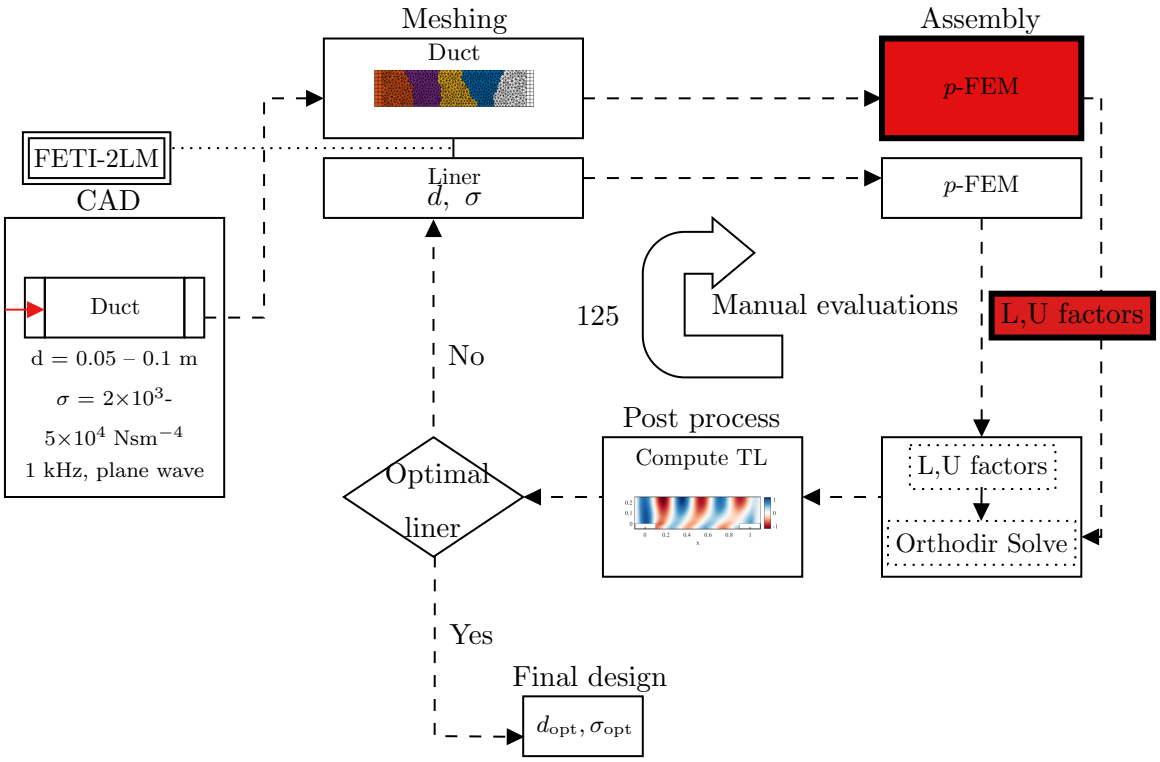


Figure 4.15: A FETI-2LM based substructuring strategy to save factorization costs for successive evaluations in the optimization loop.

4.5.1 Overview

In Chapter 2, the general framework of optimization methods of acoustic liners was presented. In this section, a modified strategy is proposed to reduce the computational costs involving a large number of configurations.

With a conventional approach, for each new liner configuration, the complete computational model needs to be assembled and requires the calculation of the LU factors. However, the liner which represents only a small part of the computational domain can be modeled as one subdomain. For each subsequent evaluation, the subdomains associated with the acoustic cavity remain the same. This significantly reduces the cost at each evaluation of a new liner since the cost of the LU factorization in the liner subdomain is small. The major benefit is the cost savings in remeshing, assembly and factorization of the cavity for each successive liner configuration.

The new workflow can be described in the following way:

1. The duct is discretized and partitioned into non-overlapping subdomains using METIS.

2. The subdomain matrices are assembled using a p-FEM based numerical discretization only once

for a given frequency and a prescribed duct mode.

3. The acoustic treatment is modeled as an equivalent fluid and is discretized ensuring a conformal surface mesh. It is then assembled using a p-FEM based numerical discretization.

4. The interface problem is solved iteratively using FETI-2LM as described in Chapter 3 and in the latter step the pressure field is recovered using forward-backward substitutions.

5. For successive evaluations, only the steps (3) and (4) need to redone since the factorizations associated with the subdomain matrices in the duct remain unchanged.

4.5.2 Verification of the model using FETI-2LM

In this part, FETI-2LM method is verified to produce the same results for the problem described in Section 2.6.

The 3D computational domain is discretized using two separate meshes of tetrahedron elements (generated using Gmsh). The first mesh is linked to the duct which is the invariant part of the domain, whereas the second mesh is associated with the liner which changes between calculations.
When changing the liner depth or flow resistivity, a new mesh for the liner is computed, while the mesh for the duct remains unchanged and is reused. Indeed, a conformal mesh needs to be enforced between the duct and liner to satisfy the continuity constraints in the FETI-2LM method. However, this requires using the same surface mesh from the duct for all liner configurations. As a result, the liner is discretized with a predefined surface mesh obtained from the duct.
The new version of Gmsh [46] (4.4.1 or later) can generate a 3D discretization for the liner, with a predefined input surface mesh. The implementation of this workflow is verified for flat and curved geometries. We demonstrate this workflow on a cylindrical duct as seen in Figure 4.16.
In Figure 4.16 (a), a quadratic duct mesh is partitioned into four non-overlapping subdomains using METIS. The liner is meshed using the aforementioned workflow in Gmsh. In Figure 4.16(b), the surface mesh for the discretized liner is superimposed on the duct. It can be clearly seen that the node mapping is conserved. Figure 4.16(c) displays the discretized duct and liner for a specific liner configuration.

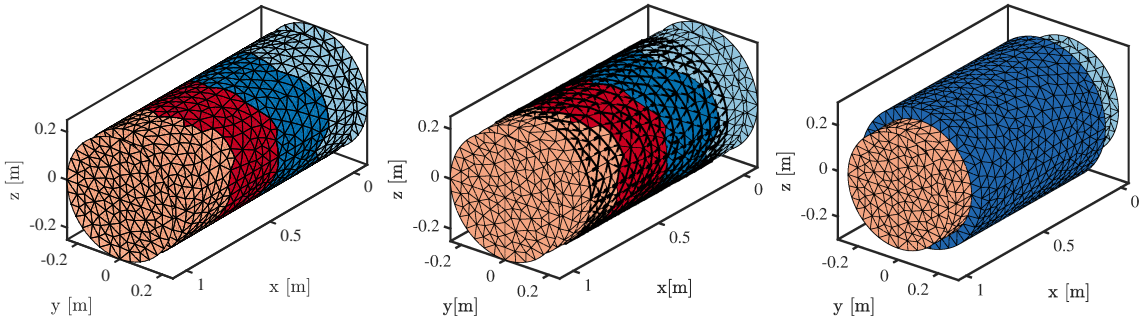


Figure 4.16: (a)Partitioning of duct into 4 non-overlapping partitions using METIS, (b) the interface mesh for the liner is superimposed on the duct to check conformity, (c) the complete mesh of the computational domain for a specific liner configuration.

The total number of DOFs is around 90,000. The interface problem, composed of dual variables has roughly 11,000 DOFs. It is solved using ORTHODIR for a set tolerance of 10^{-8}. The primal solution is recovered using forward-backward substitutions with the LU factors in each subdomain. As the last step, the transmission loss is evaluated for each calculation using the relation. The same setup is run for several liner configurations, and by reusing the previously computed LU factors for the duct subdomains. The same results are reproduced for the transmission loss as presented in Section 2.6.3. As an illustration, we showcase some specific results as described in Table 4.12.

Table 4.12: Transmission loss for $d = 0.05\,\text{m}$ and $\sigma = 1.57 \times 10^4$ for varying frequencies and duct modes.

f in Hz	1500	1500	2200
(m, n)	(0,0)	(0,1)	(1,2)
TL (in dB) :p-FEM	4.78	7.76	17.98
TL (in dB): FETI-2LM	4.78	7.76	17.98

4.6 Computational savings

This section illustrates the savings in the computational time compared to the conventional strategies presented in Chapter 2. The same problem described in Section 2.6.2 is modeled as described in Section 4.5.2.

At first, the assembly and factorization costs are illustrated using the conventional approach in Chapter 2. The matrices are assembled using p-FEM, which are sparse, complex symmetric [10]. The bandwidth of sparse matrices can be improved using the sparse reverse Cuthill-McKee algorithm. MATLAB has a built-in function abbreviated as "symrcm" which reorders the matrix entries [1] (see pg. 1-11872) based on this algorithm. It was found that using this built-in function drastically reduces

memory requirements. In the present work, all the matrices are reordered using this algorithm. Once the bandwidth is optimized, a direct solver is used to compute the primal solution. It requires the computation of the LU factors which compose of a major part of the computational cost.

Table 4.13: Direct solver cost for an 80k DOF problem.

	Assembly	Factorization	Memory (in Gb)
For 1 calculation	1min 10 s.	$\sim$ 39 mins.	10.24 Gb
For 125 calculations	2hrs 25mins.	3 days 10 hrs.	10.24 Gb

The results are computed in MATLAB on a single core. The system has an i7-processor @3.10 GHz, 32 Gb RAM with cache memory of 9 Mb. Results presented in Table 4.13 are computed on problem with 80,000 DOFs. It can be seen that the factorization costs are considerably high for each calculation. Merely, the projected factorization time for 125 calculations requires more than 3 days. Moreover, the algorithm demands more than 10 Gb memory to store the LU factors.

We now present the savings in computational time and memory using the factorization strategy for FETI-2LM method. The duct is partitioned into 4 non-overlapping subdomains, each part comprising of roughly 18,000 DOFS. The other details of the problem are the same as described in Section 4.5.2. The sequential times of the computations done on one core are reported in Table 4.14. As the matrices have smaller dimensions, the factorization costs are drastically reduced from 39 to 6 mins. Moreover, the memory footprint for all subdomains is reduced to 4.7 Gb.

Since the duct subdomains remain invariant for all calculations, these calculations are reported only once. The computational time for all liner configurations is reported in Table 4.14 and the total time is computed. It can be easily seen that the computational costs specifically linked to factorization are drastically reduced (from over 3 days to 3 hrs) which makes these models affordable in the early design phase.

Since the duct subdomains remain invariant for all calculations, the assembly and factorization time are minimal.

Table 4.14: Cost savings using a factorization strategy for the FETI-2LM solver.

125 calculations	Assembly	Factorization	Memory (in Gb)
Duct subdomains -4	56 s.	6 mins.	3.76
Liner subdomain- 1	$\sim$ 29 mins.	3hrs 7mins.	0.95
Total time	30 mins.	3 hrs 13mins.	4.71

The presented results have no direct co-relation with the total computational time for following

reasons. Firstly, the calculations for any DDM-based solver are typically done in parallel using multiple cores to speed up computations.
Secondly, only the factorization and assembly costs are presented in this part and the iterative costs associated with the FETI-2LM method are not presented. This will be discussed in the forthcoming chapters.
The reported computational times in Table 4.14 and Table 4.13 are software specific, however they demonstrate the significant computational savings using the FETI-2LM method.

4.7 Conclusions

The modeling of several fluids that have complex and frequency dependent properties using two different FETI variants namely FETI-2LM and FETI-H has been presented. Generalized transmission and continuity conditions have been devised and verified to couple of several regions with different fluids.

The first key observation was that the convergence rate for the L_2 norm of the error was preserved when compared to a p-FEM based numerical scheme without any partitions. Specifically, the different parameters used to model the equivalent fluid do not have any impact on the performance of the FETI algorithms in terms of accuracy.

Scalability tests have been performed to assess the performance for modeling several fluids using the two FETI variants for varying problem parameters, partitions, and frequencies. In general, similar trends have been observed as for the standard FETI-2LM approach for modeling classical Helmholtz problems using p-FEM based discretization.

The performance of FETI-2LM is superior compared to FETI-H for varying problem parameters, partitions, and frequencies. An interesting remark is that the number of iterations required for convergence decrease for increasing frequencies, contrary to the homogeneous problem. Also, FETI-2LM algorithm requires the same number of iterations for convergence for increasing partitions in the equivalent fluid medium. However, it does not scale with increasing partitions in the air medium which was also observed in the homogeneous problem. For increasing problem resolutions using h or p refinement only slightly increases the iteration count resulting in much improved accuracy for the dual as well as the primal solutions. Typically, for a fixed accuracy, varying the problem resolution parameters, p or h, results in the same number of iterations for convergence.

The p and h refinements for fixed accuracy typically also result in the same number of iterations as for the FETI-2LM. However, the iterations count for all examples is always found to be higher for

FETI-H. Contrary to FETI-2LM, FETI-H does not scale with varying partitions in either medium or frequencies. In general, for all test examples comparing FETI-2LM and FETI-H for the same accuracy, FETI-2LM always outperforms FETI-H. Thus a clear choice of using FETI-2LM for the remaining part of this document is done.

It can thus be said that at least in the unpreconditioned version, FETI-2LM is the clear choice for modeling several fluids governed by the Helmholtz equation. The implementation of the FETI-2LM is verified for the case of several fluids with varying partitions in each of the fluids. The performance and accuracy were also verified for 3D models.

A novel factorization strategy was proposed to solve problems involving several liner calculations. This strategy demonstrates considerable cost reductions compared to classical methods described in Chapter 2. The cost savings for small 3D models using the factorization strategy for FETI-2LM method are presented. However, there is more potential for cost savings in the iterative process for the FETI-based methods. The forthcoming chapters will focus on reducing the costs linked to the iterative part of the FETI methods for a series of calculations.

Chapter 5

Two-level FETI-2LM

5.1 Introduction

The previous chapter demonstrates the computational savings for the resolution of large problems where only a small part of the subdomain changes. Specifically, a FETI-2LM method is used to reuse the LU factors of the invariant subdomains in the case of several calculations. However, this only leads to modest computational gains. In fact, the FETI-2LM method involves iterative part which comprises of significant computational costs. Typically, each iteration in a specific calculation involves a matrix vector (MV) product, forcing a local solve of all subdomains. This involves the exchange of dual variables between the subdomains hampering the parallel performance of these methods. The key objective of this chapter is to investigate novel FETI-methods to reduce the computational effort associated with the iterative costs.

In this chapter, a novel two-level FETI-2LM formulation is outlined to reduce the costs linked to the iterative part. In order to avoid misunderstandings, it is important to clarify the terminology "two-level FETI", which depends on the context. The FETI method described in the original paper by Roux [82] consists in formulating the interface problem for the dual variables and in the latter phase introducing a coarse-grid correction at each iteration of this interface problem [31]. Also, a multi-level FETI-DP formulation was proposed to address the issues relating to the increasing size of the coarse grid for very large-scale discrete problems [91]. Hence in these papers, the terminology "two-level" mainly refers to multi-grid techniques. In the method presented in this chapter the FETI method is used to find a solution to the subdomain problem, we then have a FETI method encapsulated in another one. Hence the vocable "two-level" has a different meaning. To the best of authors knowledge, a novel FETI-methods dedicated for a series of calculations for acoustic applications has

not been done before.

In the first part of the chapter, the theory and the formulation of the two-level FETI-2LM method is presented. In the second part, the implementation of the two-level FETI-2LM method is verified. The influence of using different tolerance criteria for the iterative schemes like ORTHODIR, GMRES and ORTHOMIN is discussed. In the last part, suitable conclusions are drawn regarding the choice of the iterative method for the two-level FETI-2LM method.

5.2 Two-level FETI

This section proposes a two-level FETI method. The configuration of interest is the same as the one examined in Chapter 3. This problem is first recalled, the method is then presented.

5.2.1 Formulation

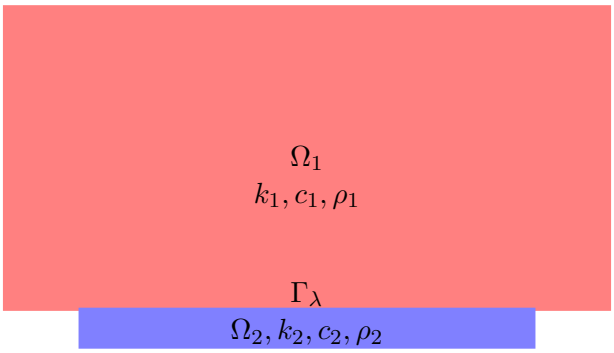


Figure 5.1: Problem of interest consisting of two domains 1 and 2.

A problem composed of two fluids (respectively denoted by 1 and 2) corresponding to subdomains, Ω_1 and Ω_2 is presented in Figure 5.1. Their common boundary is denoted by Γ_λ. In each domain, the pressure field u is governed by the Helmholtz equation similar to Section 4.2 in Chapter 4.

$$\frac{1}{\rho_i}\Delta u + \frac{k_i^2}{\rho_i}u = 0 \qquad \text{in} \qquad \Omega_i, \quad i = 1, 2. \tag{5.1}$$

The continuity of the pressure and normal velocity fields is applied on the interface Γ_λ and a generic Robin condition is on the vertical boundaries of Ω_1:

$$\frac{\partial u_1}{\partial n_1} + \mathrm{i}k_1 u_1 = g_1 \qquad \text{on} \qquad \partial\Omega_1 \backslash \Gamma_\lambda,$$

where g_1 is a known function. The remaining boundaries of Ω_1 and Ω_2 are modeled as hard walls.

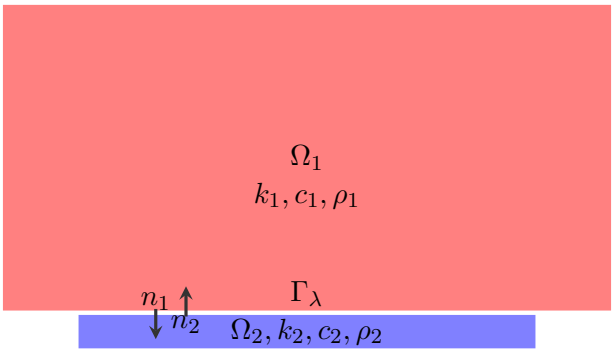


Figure 5.2: top-level splitting of the domain with the interface Γ_λ.

Subdomain 1 represents the acoustic cavity and will constitute a major part of the total computational domain. Subdomain 2 corresponds to the acoustic liner which is a small part of the complete domain. We are considering a series of configurations where the problem associated with Ω_1 remains the same while the problem in Ω_2 changes.

In this chapter, we will present a two-level FETI method which consists in using twice the FETI method:

- The two subdomains Ω_1 and Ω_2 will lead to a so-called top-level FETI (TL) with an interface problem on Γ_λ. However, subdomain 1 can be further split into smaller subdomains. It then requires the resolution of local problems linked to these smaller subdomains.
- To solve this local problem, a so-called bottom-level FETI (BL) problem is formulated, which determines the pressure fields in the subdomains.

In this presentation, we will only solve the bottom-level FETI method on Ω_1 as it is the subdomain on which the direct resolution is the most computationally demanding and the one which is invariant in an optimization problem. The resolution of the local problem on Ω_2 will be done with a direct solver but note that this does not limit the generality of the approach.

5.2.2 Top-level FETI

The top-level consists of splitting the domain into two parts Ω_1 and Ω_2. The same sets of equations described in Section 4.2.1 are used to solve this top-level FETI problem. The FETI-2LM continuity and transmission conditions are imposed on the interface Γ_λ. Dual variables, or Lagrange multipliers, $\tilde{\boldsymbol{\lambda}}_{2\to1}$, $\tilde{\boldsymbol{\lambda}}_{1\to2}$ are defined on Γ_λ and the interface problem (see for instance Equation (4.16)) with the notations from Chapter 4 is recalled: The resulting interface problem can be expressed in terms of the

dual variables:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} - Y_{1,2}\,\mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{B}_{2,1}^{\mathrm{T}} \\ \mathbf{I} - Y_{1,2}\,\mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{B}_{1,2}^{\mathrm{T}} & \mathbf{I} \end{bmatrix}}_{\mathbf{F}_{\mathrm{I}}^{1,2}} \underbrace{\begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{2\to1} \\ \tilde{\boldsymbol{\lambda}}_{1\to2} \end{bmatrix}}_{\tilde{\boldsymbol{\lambda}}} = \underbrace{Y_{1,2} \begin{bmatrix} \mathbf{M}_{1,2}\mathbf{B}_{2,1}\mathbf{K}_2^{-1}\mathbf{f}_2 \\ \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{K}_1^{-1}\mathbf{f}_1 \end{bmatrix}}_{\mathbf{d}_\lambda} . \tag{5.2}$$

The interface matrix $\mathbf{F}_{\mathrm{I}}^{1,2}$ is a complex square matrix and contains full blocks. It is of size $\mathrm{N}_{\mathrm{I}}^{\lambda}$, where $\mathrm{N}_{\mathrm{I}}^{\lambda}$ designates the number of degrees of freedom of interface Γ_λ. The interface problem is solved by an iterative algorithm which only requires to be able to compute the matrix-vector product $\mathbf{F}_{\mathrm{I}}\tilde{\boldsymbol{\lambda}}$. Hence this matrix it then not formed explicitly.

This interface problem then requires the factorization of matrices $\mathbf{K}_1$ and $\mathbf{K}_2$. As mentioned earlier, we will focus on $\mathbf{K}_1$ whose formal inverses are presented in blue and red. They are present in both the left and right-hand sides, see the terms in red and blue in equation Equation (5.2). The term marked in red consist of the contribution of the sources which do not change in the iterative loop. However, the term marked in blue involves the dual variables $\tilde{\boldsymbol{\lambda}}_{2\to1}$ which evolves in the iterative process. It is then useful to split the solution in two parts: the fixed contribution from the sources and the variable contribution due to the dual variables:

$$\mathbf{u}_1 = \mathbf{u}^s + \mathbf{u}^\lambda, \tag{5.3}$$

where $\mathbf{u}^s$ denotes the fixed contribution from the source terms and $\mathbf{u}^\lambda$ denotes the variable contribution from the dual variables. Computing these two quantities requires solving linear systems defined by $\mathbf{K}_1$:

$$\begin{aligned} \mathbf{K}_1\mathbf{u}^s &= \mathbf{f}_1, \\ \mathbf{K}_1\mathbf{u}^\lambda &= \mathbf{B}_{1,2}^{\mathrm{T}}\tilde{\boldsymbol{\lambda}}_{2\to1}. \end{aligned} \tag{5.4}$$

These two problems are solved by the bottom-level FETI.

5.2.3 Bottom-level FETI

The key objective of the bottom-level FETI is to solve Equation (5.4) without having to compute factorization of the matrix $\mathbf{K}_1$.

5.2.3.1 Presentation of the bottom-level problem

We will now solve the problems described in Equation (5.4) to compute the pressure fields $\mathbf{u}^s$, $\mathbf{u}^\lambda$. Both problems in Equation (5.4) have the same matrix $\mathbf{K}_1$ and only the right-hand side changes.

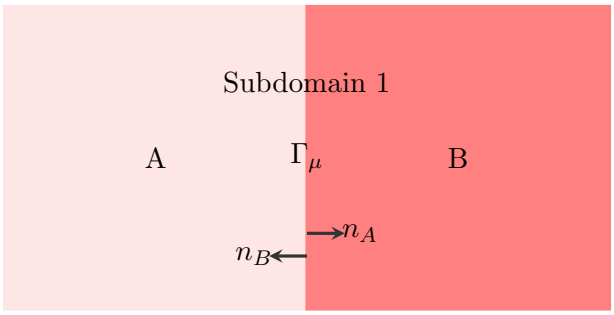


Figure 5.3: Partitioning of Ω_1 into two parts, namely subdomains A and B.

Thus, the bottom-level problem is presented in terms of the following generic problem:

$$\mathbf{K}_1\mathbf{u}_g = \hat{\mathbf{b}}_1, \text{where, } \hat{\mathbf{b}}_1 = \begin{cases} \mathbf{f}_1, \\ \mathbf{B}_{1,2}^{\mathrm{T}}\tilde{\boldsymbol{\lambda}}_{2\to1}. \end{cases} \tag{5.5}$$

where $\mathbf{u}_g$ represent a generic pressure field which could be either $\mathbf{u}^s$ or $\mathbf{u}^\lambda$.

Ω_1 is split into a number of smaller subdomains indexed by capital letters as A,B,C.... The additional dual variables introduced by this partitioning are denoted $\boldsymbol{\mu}$. They exist only on the interior interfaces within Ω_1. To simplify the description of the method, Ω_1 is partitioned into 2 non-overlapping subdomains, namely A and B, as seen in the Figure 5.3. In this case, there is only one interface within Ω_1 which is denoted Γ_μ. It is straightforward to generalise the method to an arbitrary number of partition in Ω_1.

One can solve this generic bottom-level problem using any domain decomposition method, like Balanced Domain Decomposition or Optimized Schwarz methods. However, in the present work only FETI-based methods are discussed. Besides, from the results of Chapter 4, it was observed that FETI-2LM outperformed FETI-H. Thus we will solve the bottom-level problem with FETI-2LM.

The transmission and continuity equations proposed in Chapter 4 are reused to couple the partitioned subdomains A,B which results in the following equations:

$$\begin{aligned} \frac{\partial u_A}{\partial n_A} + \mathrm{i}k_1 u_A &= \rho_1 \mu_{B\to A} \quad \text{on } \Gamma_\mu, \\ \frac{\partial u_B}{\partial n_B} + \mathrm{i}k_1 u_B &= \rho_1 \mu_{A\to B} \quad \text{on } \Gamma_\mu, \end{aligned} \tag{5.6}$$

where u_A, u_B are the unknown pressure fields. n_A, n_B are the unit outward normals. $\mu_{A\to B}$, $\mu_{B\to A}$ are the unknown dual variables for Ω_A and Ω_B, respectively.

The resulting weak form of the problem for subdomains Ω_A, Ω_B can be written as follows:

$$\begin{aligned}
\forall w_A \in V_A \quad & \frac{1}{\rho_1}\int_{\Omega_A}\{\boldsymbol{\nabla} w_A^\star \cdot \boldsymbol{\nabla} u_A - k_1^2 w_A^\star u_A\}\mathrm{d}\Omega + \frac{\mathrm{i}k_1}{\rho_1}\int_{\Gamma_\mu} w_A^\star u_A \mathrm{dS} = \\
& \frac{1}{\rho_1}\int_{\partial\Omega_A\setminus\Gamma_\mu} w_A^\star \frac{\partial u_A}{\partial n_A}\mathrm{dS} + \int_{\Gamma_\mu} w_A^\star \mu_{B\to A}\mathrm{dS}, \\
\forall w_B \in V_B \quad & \frac{1}{\rho_1}\int_{\Omega_B}\{\boldsymbol{\nabla} w_B^\star \cdot \boldsymbol{\nabla} u_B - k_1^2 w_B^\star u_B\}\mathrm{d}\Omega + \frac{\mathrm{i}k_1}{\rho_1}\int_{\Gamma_\mu} w_B^\star u_B \mathrm{dS} = \\
& \frac{1}{\rho_1}\int_{\partial\Omega_B\setminus\Gamma_\mu} w_B^\star \frac{\partial u_B}{\partial n_B}\mathrm{dS} + \int_{\Gamma_\mu} w_B^\star \mu_{A\to B}\mathrm{dS},
\end{aligned} \tag{5.7}$$

in which w_A, w_B are the test functions associated with the field u_A, u_B and V_A, V_B are the functional spaces for these test functions.

The continuity of the pressure field and the normal pressure gradient is imposed on the interface (subdomain 1 is homogeneous). This results in an additional equations to compute the dual variables.

$$\begin{aligned}
\forall \xi_{A,B}^\star \in V_A \quad & \int_{\Gamma_\mu} \xi_{A,B}^\star \Big[\mu_{A\to B} + \mu_{B\to A} - \tilde{s} u_B\Big]\mathrm{dS} = 0, \\
\forall \xi_{B,A}^\star \in V_B \quad & \int_{\Gamma_\mu} \xi_{B,A}^\star \Big[\mu_{B\to A} + \mu_{A\to B} - \tilde{s} u_A\Big]\mathrm{dS} = 0,
\end{aligned} \tag{5.8}$$

with $\tilde{s} = 2\mathrm{i}k_1/\rho_1$. This parameter $\tilde{s}$ depends only on the properties of the subdomain Ω_1 and these properties of Ω_1 does not change for several liner calculations.

The numerical discretization results in the following systems of equations:

$$\begin{aligned}
\tilde{\mathbf{K}}_A \mathbf{u}_A &= \hat{\mathbf{b}}_A + \mathbf{B}_{A,B}^{\mathrm{T}}\, \tilde{\boldsymbol{\mu}}_{B\to A}, \\
\tilde{\mathbf{K}}_B \mathbf{u}_B &= \hat{\mathbf{b}}_B + \mathbf{B}_{B,A}^{\mathrm{T}}\, \tilde{\boldsymbol{\mu}}_{A\to B}, \\
\tilde{\boldsymbol{\mu}}_{A\to B} = \mathbf{M}_{A,B}\, \boldsymbol{\mu}_{A\to B}, &\quad \tilde{\boldsymbol{\mu}}_{B\to A} = \mathbf{M}_{A,B}\, \boldsymbol{\mu}_{B\to A},
\end{aligned} \tag{5.9}$$

where $\tilde{\mathbf{K}}_A$, $\tilde{\mathbf{K}}_B$ are the regularized Helmholtz matrices. $\hat{\mathbf{b}}_A$, $\hat{\mathbf{b}}_B$ are the discretized forcing terms. $\mathbf{u}_A$, $\mathbf{u}_B$ are the primal unknown degrees of freedom arising from the discretization of the pressure field in Ω_A and Ω_B, respectively. $\tilde{\boldsymbol{\mu}}_{B\to A}$ and $\tilde{\boldsymbol{\mu}}_{A\to B}$ are the dual unknown degrees of freedom arising from the discretization of the field of the dual variables (for details see Section 3.2.1.3). $\mathbf{M}_{A,B}$ is the mass matrix on the partitioned interface Γ_μ. The Boolean operators $\mathbf{B}_{A,B}$, $\mathbf{B}_{B,A}$ are used to project the subdomain information on the interface Γ_μ. The discretization of the interface conditions in Equation (5.8) leads to additional sets of equations:

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_{A\to B} + \tilde{\boldsymbol{\mu}}_{B\to A} - \tilde{s}\, \mathbf{M}_{A,B}\mathbf{u}_B = \mathbf{0}, \\
\tilde{\boldsymbol{\mu}}_{A\to B} + \tilde{\boldsymbol{\mu}}_{B\to A} - \tilde{s}\, \mathbf{M}_{A,B}\mathbf{u}_A = \mathbf{0}.
\end{aligned} \tag{5.10}$$

Expressing Equation (5.10) only in terms of dual variables using the relation from Equation (5.9) (a), (b) results in the following system of equations:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} - \tilde{s}\,\mathbf{M}_{A,B}\mathbf{B}_{B,A}\tilde{\mathbf{K}}_B^{-1}\mathbf{B}_{B,A}^{\mathrm{T}} \\ \mathbf{I} - \tilde{s}\,\mathbf{M}_{A,B}\mathbf{B}_{A,B}\tilde{\mathbf{K}}_A^{-1}\mathbf{B}_{A,B}^{\mathrm{T}} & \mathbf{I} \end{bmatrix}}_{\mathbf{F}_{\mathrm{I}}^{A,B}} \underbrace{\begin{bmatrix} \tilde{\boldsymbol{\mu}}_{A\to B} \\ \tilde{\boldsymbol{\mu}}_{B\to A} \end{bmatrix}}_{\tilde{\boldsymbol{\mu}}} = \underbrace{\tilde{s}\begin{bmatrix} \mathbf{M}_{A,B}\big(\mathbf{B}_{B,A}\tilde{\mathbf{K}}_B^{-1}\hat{\mathbf{b}}_B\big) \\ \mathbf{M}_{A,B}\big(\mathbf{B}_{B,A}\tilde{\mathbf{K}}_A^{-1}\hat{\mathbf{b}}_A\big) \end{bmatrix}}_{\hat{\mathbf{d}}_\mu}. \tag{5.11}$$

The interface matrix $\mathbf{F}_{\mathrm{I}}^{A,B}$ is a complex square matrix and contains full blocks. It is of size $\mathrm{N}_{\mathrm{I}}^{\mu}$, where $\mathrm{N}_{\mathrm{I}}^{\mu}$ designates the number of unknowns on the interface Γ_μ. This problem can be solved using any of the iterative schemes described before, with a set tolerance to get converged results for the dual variables. The tolerance at the bottom-level is referred to as ε_B. The next steps involves recovering the local solutions in subdomains A and B:

$$\begin{aligned} \tilde{\mathbf{K}}_A\hat{\mathbf{u}}_A &= \hat{\mathbf{f}}_A + \mathbf{B}_{A,B}^{\mathrm{T}}\,\tilde{\boldsymbol{\mu}}_{B\to A}^{conv}, \\ \tilde{\mathbf{K}}_B\hat{\mathbf{u}}_B &= \hat{\mathbf{f}}_B + \mathbf{B}_{B,A}^{\mathrm{T}}\,\tilde{\boldsymbol{\mu}}_{A\to B}^{conv}, \end{aligned} \tag{5.12}$$

where $\tilde{\boldsymbol{\mu}}_{B\to A}^{conv}, \tilde{\boldsymbol{\mu}}_{A\to B}^{conv}$ are the converged dual variables linked to the generic bottom-level interface problem. In order to recover the primal variables over Ω_1, averaging at the common interfaces of the subdomains is used. In addition, when the number of partitions increase, cross points might be present. An averaging of the solution at these cross points is also necessary. Finally the solution in Ω_1 is recovered as follows:

$$\hat{\mathbf{u}}_1 = \begin{cases} \hat{\mathbf{u}}_A \quad \forall\ \hat{\mathbf{x}}_A \in \Omega_A \backslash \Gamma_\mu, \\ \hat{\mathbf{u}}_B \quad \forall\ \hat{\mathbf{x}}_B \in \Omega_B \backslash \Gamma_\mu, \\ (\hat{\mathbf{u}}_A + \hat{\mathbf{u}}_B)/2 \quad \forall\ \hat{\mathbf{x}}_A\,, \hat{\mathbf{x}}_B \in \Gamma_\mu. \end{cases} \tag{5.13}$$

In principle, it would be possible to continue this process of splitting a specific subdomain into a number of smaller subdomains. For instance, one can split the subdomain A into 2 smaller subdomains to recover the solution $\hat{\mathbf{u}}_A$. For brevity, we limit the number of levels in this particular case only to the top and the bottom-levels.

5.2.3.2 Contribution from the source terms

This section deals with the computation of the pressure field $\mathbf{u}^s$ using the generic bottom-level problem, as illustrated in Figure 5.4. This contribution due to the sources remains fixed in the top-level problem described in Equation (5.2). Moreover, these source terms do not change while performing several liner calculations. As a result this computation is performed only once throughout the optimization

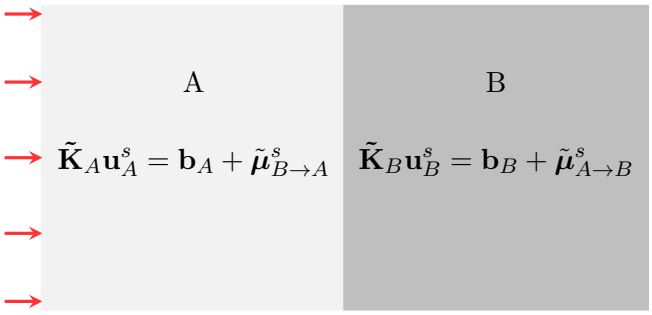


Figure 5.4: Computation of the solution $\mathbf{u}^s$ using the generic bottom-level problem.

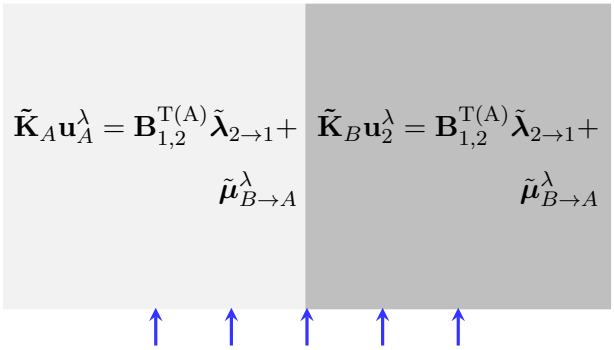


Figure 5.5: Computation of the pressure field $\mathbf{u}^\lambda$ using the bottom-level problem.

cycle.

The interface problem in Equation (5.11) is used to compute the dual variables $\tilde{\boldsymbol{\mu}}_{A\to B}^s$, $\tilde{\boldsymbol{\mu}}_{B\to A}^s$ by using the forcing term as follows:

$$\begin{aligned}\hat{\mathbf{b}}_A &= \mathbf{B}_{A,1}^{\mathrm{T}}\hat{\mathbf{f}}_1,\\ \hat{\mathbf{b}}_B &= \mathbf{B}_{B,1}^{\mathrm{T}}\hat{\mathbf{f}}_1,\end{aligned} \tag{5.14}$$

where $\mathbf{B}_{B,1}^{\mathrm{T}}, \mathbf{B}_{A,1}^{\mathrm{T}}$ are Boolean operators to project the source terms from subdomain 1 to A and B respectively. These problems are solved using the generic bottom-level problem to get the converged solution for the dual variables up to the tolerance ε_B.

Once $\tilde{\boldsymbol{\mu}}_{A\to B}^s$ and $\tilde{\boldsymbol{\mu}}_{B\to A}^s$ are computed, one can then determine $\mathbf{u}^s$ using Equation (5.12) and Equation (5.13). The right-hand side of the top-level interface problem can now be simplified as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{I} - Y_{1,2}\,\mathbf{M}_{1,2}\mathbf{B}_{2,1}\tilde{\mathbf{K}}_2^{-1}\mathbf{B}_{2,1}^{\mathrm{T}} \\ \mathbf{I} - Y_{1,2}\,\mathbf{M}_{1,2}\mathbf{B}_{1,2}\tilde{\mathbf{K}}_1^{-1}\mathbf{B}_{1,2}^{\mathrm{T}} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{1\to 2} \\ \tilde{\boldsymbol{\lambda}}_{2\to 1} \end{bmatrix} = Y_{1,2} \begin{bmatrix} \mathbf{M}_{1,2}\mathbf{B}_{2,1}\tilde{\mathbf{K}}_2^{-1}\mathbf{f}_2 \\ \mathbf{M}_{1,2}\mathbf{B}_{1,2}\mathbf{u}^s \end{bmatrix}. \tag{5.15}$$

5.2.3.3 Contribution due to dual variables

The generic bottom-level problem is used to compute $\mathbf{u}^\lambda$ at each top-level iteration. The resulting dual variables generated at the bottom-level are denoted $\tilde{\boldsymbol{\mu}}_{A\to B}^\lambda$ and $\tilde{\boldsymbol{\mu}}_{B\to A}^\lambda$ as seen in Figure 5.5. This process is continued until the top-level problem has converged to the desired tolerance to provide $\tilde{\boldsymbol{\lambda}}_{2\to 1}^{conv}$

and $\tilde{\boldsymbol{\lambda}}_{1\to2}^{conv}$. The pressure field in subdomain 1 is recovered by solving the generic bottom-level problem with the new forcing term as seen in Equation (5.5). All the computation steps remain the same as described for generic bottom-level problem. This problem can be described as below.

$$\mathbf{K}_1\mathbf{u}_f^{\lambda} = \mathbf{B}_{1,2}^{\mathrm{T}}\tilde{\boldsymbol{\lambda}}_{2\to1}^{conv}. \tag{5.16}$$

Thus we presented a methodology to obtain $\mathbf{u}^s, \mathbf{u}_f^{\lambda}$ using the bottom-level approach. The total pressure in subdomain 1 can be obtained as a sum of the contributions of the individual pressure fields due to the source and dual variables $\tilde{\boldsymbol{\lambda}}_{2\to1}^{conv}$.

$$\mathbf{u}_1 = \mathbf{u}^s + \mathbf{u}_f^{\lambda}\ . \tag{5.17}$$

The pressure in subdomain 2 can retrieved by simply using forward-backward substitutions. Equation (5.2) :

$$\tilde{\mathbf{K}}_2\mathbf{u}_2 = \mathbf{f}_2 + \mathbf{B}_{2,1}^{\mathrm{T}}\ \tilde{\boldsymbol{\lambda}}_{1\to2}^{conv}\ . \tag{5.18}$$

The pressure field at the top-level is averaged on the interface Γ_λ to recover the primal solution over the complete domain:

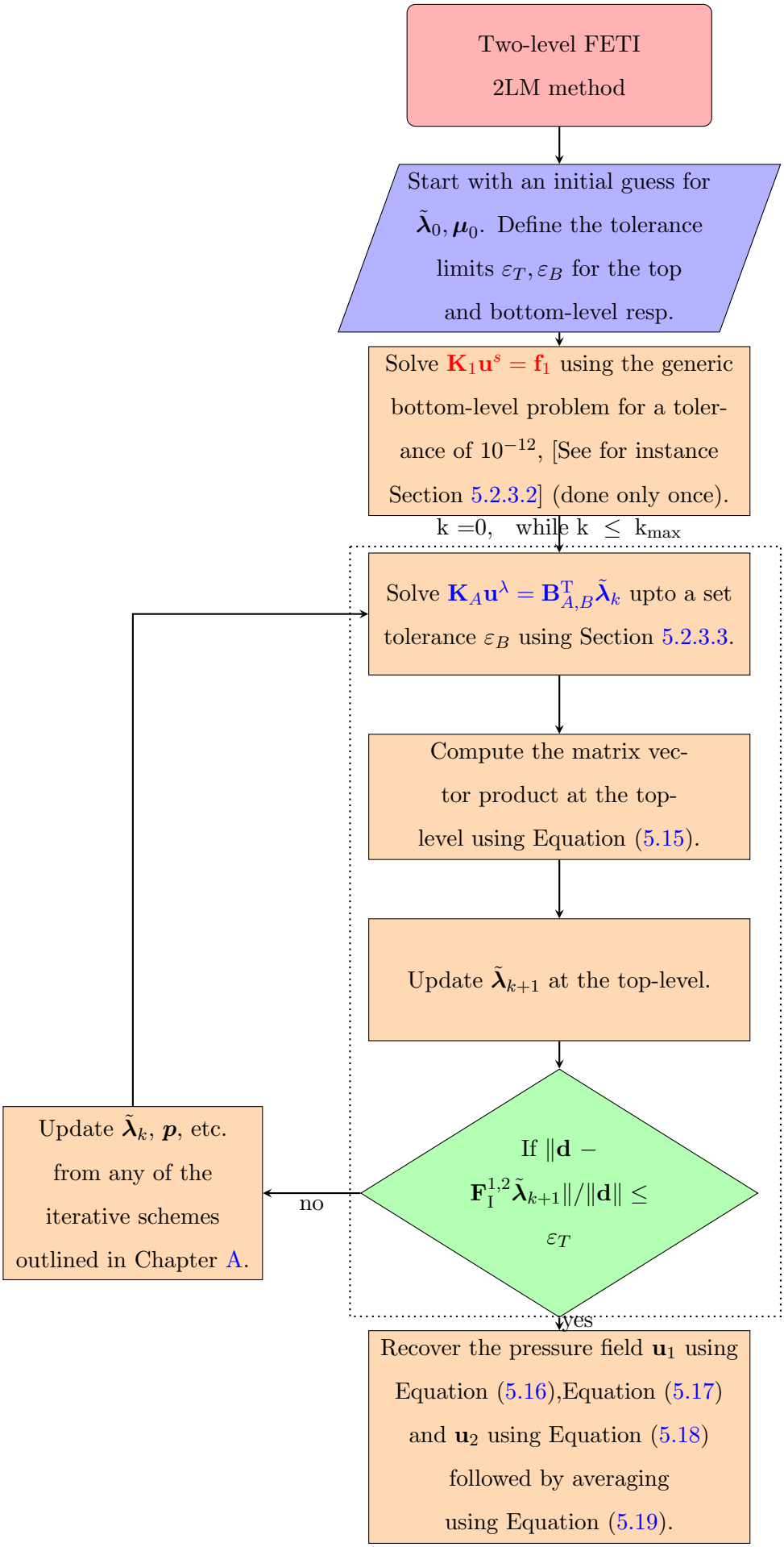
$$\mathbf{u} = \begin{cases} \mathbf{u}_1 & \forall\ \hat{\mathbf{x}}_1 \in \Omega_1\backslash\Gamma_\lambda, \\ \mathbf{u}_2 & \forall\ \hat{\mathbf{x}}_2 \in \Omega_2\backslash\Gamma_\lambda, \\ (\mathbf{u}_1 + \mathbf{u}_2)/2 & \forall\ \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2 \in \Gamma_\lambda. \end{cases} \tag{5.19}$$

Thus the bottom-level problem provides a generic way to compute the primal solution without having to do the expensive factorization linked to subdomain 1.

We now present a flow-chart of the two-level FETI-2LM method for a single calculation.

5.2.4 Flowchart

A flowchart of the two-level FETI-2LM comprising of top and bottom level FETI methods is presented. $k_{\max}$ is the maximum number of iterations defined for the chosen iterative scheme.


Two-level FETI
2LM method
Start with an initial guess for $\tilde{\boldsymbol{\lambda}}_0, \boldsymbol{\mu}_0$. Define the tolerance limits $\varepsilon_T, \varepsilon_B$ for the top and bottom-level resp.
Solve $\mathbf{K}_1\mathbf{u}^s = \mathbf{f}_1$ using the generic bottom-level problem for a tolerance of 10^{-12}, [See for instance Section 5.2.3.2] (done only once).
k =0, while k $\leq$ k_{max}
Solve $\mathbf{K}_A\mathbf{u}^\lambda = \mathbf{B}_{A,B}^T\tilde{\boldsymbol{\lambda}}_k$ upto a set tolerance ε_B using Section 5.2.3.3.
Compute the matrix vector product at the top-level using Equation (5.15).
Update $\tilde{\boldsymbol{\lambda}}_{k+1}$ at the top-level.
If $\|\mathbf{d} - \mathbf{F}_I^{1,2}\tilde{\boldsymbol{\lambda}}_{k+1}\|/\|\mathbf{d}\| \leq \varepsilon_T$
no
Update $\tilde{\boldsymbol{\lambda}}_k$, $\boldsymbol{p}$, etc. from any of the iterative schemes outlined in Chapter A.
yes
Recover the pressure field $\mathbf{u}_1$ using Equation (5.16),Equation (5.17) and $\mathbf{u}_2$ using Equation (5.18) followed by averaging using Equation (5.19).

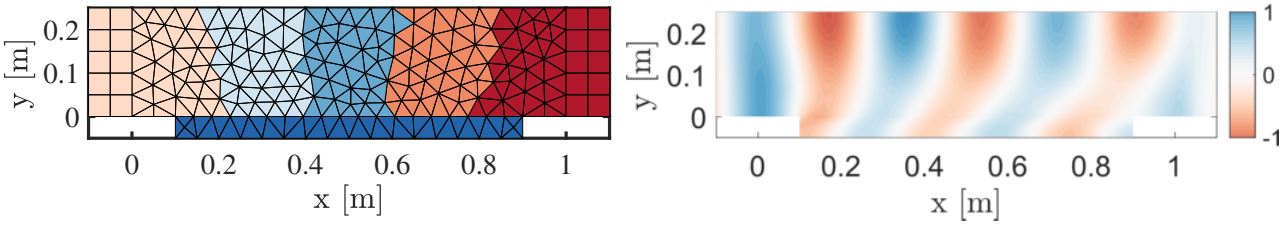


Figure 5.6: (a) Partitioning of the duct into 5 subdomains. The liner is modeled as an independent subdomain, (b) Example of pressure field for a plane wave at 1 kHz.

5.3 Verification on a simple case

This section describes the verification of the two-level FETI-2LM method. This verification is done mainly to ensure implementation has been done correctly as the two level FETI is algebraically equivalent to the resolution of the original FEM problem. The two-level approach for one calculation is indeed expensive compared to the original FEM or the one-level FETI-2LM method. However, this approach is meant to tackle a series of calculations and not specifically a particular configuration of interest.

A two-dimensional test case is used for this task, as seen in Figure 5.6. It represents a duct of length 1 m along the x-axis with a height of 0.25 m. The duct is acoustically treated with a liner modeled using the JCA model. The liner length is 0.8 m and its depth is 5 cm. A plane wave at 1 kHz is injected from the left boundary using an active PML, see [10]. A standard PML is used on the right boundary to absorb the outgoing waves. The FETI-2LM transmission conditions are applied on the interface between the duct and the acoustic treatment. All the other boundaries are hard walls.

The computational domain is discretized using triangular elements and a conformal mesh between the duct and the liner. The duct is partitioned into 5 subdomains using METIS. The dual variables introduced due to this METIS partitioning constitute the bottom-level FETI-2LM problem. A fixed tolerance of 10^{-6} is used for both the top- and the bottom-level FETI-2LM interface problems. The bottom-level problem linked with the computation of sources is solved with a much finer tolerance of 10^{-12}. This allows to capture the RHS of the top-level interface problem very accurately. Since the calculation is done only once, it involves negligible overhead in the computational cost.

The primal solution is recovered using forward-backward substitutions as described in Section 5.2.4. As a last step, the transmission loss (TL) is computed and compared with the same results obtained from the (one-level) FETI-2LM formulation described in Chapter 4 for the iterative schemes, ORTHODIR, GMRES and ORTHOMIN. For brevity, the results for only presented with the ORTHODIR algorithm for the tolerance of 10^{-6}. It can be clearly seen that the two-level FETI-2LM method shows good agreement with the FETI-2LM method and correspondingly with the original FEM problem.

Table 5.1: Comparison of the TL for the FETI-2LM and two-level FETI-2LM method for a plane wave at different frequencies.

Frequency (in Hz)	1000	2000
TL (in dB) for One-level FETI-2LM	4.6709	1.3126
TL (in dB) for Two-level FETI-2LM	4.6715	1.3117

We can consider that the method is verified. We will now study the influence of resolution parameters. They are two kinds:

- The resolution algorithm itself: ORTHODIR, GMRES or ORTHOMIN
- The method to control the iterations at top and bottom-level

In the next section, we will first consider ORTHODIR as it is the first one that has been studied and which is currently used for FETI methods.

5.4 Iterative resolution with ORTHODIR

In this section and the next ones we will use the same framework to present the results (see for example fig. 5.7). The figures will be composed of three plots:

1. Left figure or (a): This figure will corresponds to the number of bottom-level iterations at each top-level iteration.
2. Middle figure or (b): This figure shows the rate of convergence of top-level interface problem. The normalized residual is plotted on a log scale as a function of the top-level iterations. In certain cases, the tolerance at the bottom-level is adapted depending on the top-level. The tolerance at the bottom-level is plotted as a function of the top-level iterations.
3. Right figure or (c): This figure plots the evolution of the relative L_2 error on the pressure with top-level iterations. The computation of this relative error requires a reference pressure solution. For this test case, the reference solution u_{ref} is obtained by solving the top-level interface problem for a very small tolerance, almost equivalent to a direct solve. At each iteration, the pressure field is recovered and the normalized error, E_k is computed as follows:

$$\mathrm{E}_k = \frac{\|u_{ref} - u_k\|}{\|u_{ref}\|}, \tag{5.20}$$

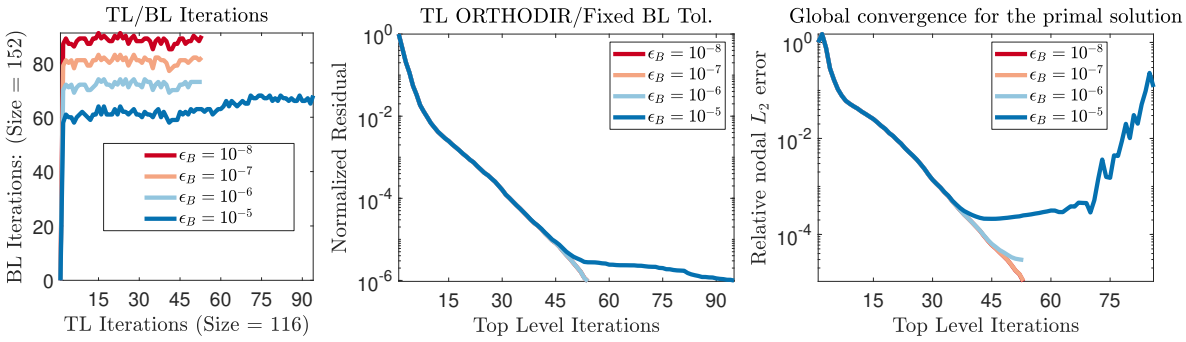


Figure 5.7: Results for ORTHODIR and a fixed tolerance at top-level $\varepsilon_T = 10^{-6}$.

where u_k is the pressure field at the k^{th} top-level iteration computed using the method described in Section 5.2.4. The interest of this figure is to check that the problem is properly solved for the primal solution as a reduction of the residual (as presented in the figure in the middle) is only a necessary condition: the method can converge toward a wrong solution.

The results algorithm are presented in two parts. First, a fixed tolerance criterion is used at the bottom-level ε_B for the same top-level tolerance ε_T. Secondly, different adaptive tolerance criteria are used at the bottom-level for the same top-level tolerance of 10^{-6}.

5.4.1 Results for a fixed tolerance

Table 5.2: ORTHODIR fixed tolerance studies for $\epsilon_T = 10^{-6}$.

ϵ_B	TL Iterations	BL Iterations
10^{-8}	53	4591
10^{-7}	53	4198
10^{-6}	53	3741
10^{-5}	94	5873

The fixed tolerance study consists in varying the tolerance ε_B at the bottom-level for a fixed tolerance at top-level ε_T . The latter is set to 10^{-6} while the range of variation of ε_B goes from 10^{-8} to 10^{-5} as seen in Table 5.2. We will define 4 cases associated with different tolerances as presented in Figure 5.7.

Figure 5.7(a) presents the evolution of the number of bottom-level iterations for each iteration at top-level. Each one of the four curves is almost constant. It means that the number of bottom-level iteration is the same and do not evolve with the top-level iterations. Of course, the number of bottom-level iteration is higher for smaller ε_B. An intermediate conclusion is that the cost of each bottom-level resolution is the same for each top-level iteration and is dependent of the level of the bottom-level

tolerance.

Figure 5.7(b) presents the evolution of the residual of the top-level problem. We now observe that during the first iterations, the four curves are superposed. As seen previously, we can deduce that it is not necessary to choose a very small bottom-level tolerance as it does not affect the evolution of the top-level residual and requires of course more computational resources. For case 4 ($\varepsilon_B = 10^{-5}$), a plateau is observed on the residual. This was expected as the tolerance chosen at the bottom-level is not sufficient to guarantee a good prediction for the pressure that are involved in the top-level resolutions.

The convergence of the primal variables (pressure) computed using the relation Equation (5.20) can be seen in Figure 5.7(c). For small bottom-level tolerances ($\varepsilon_B = 10^{-8}$ and $\varepsilon_B = 10^{-7}$) , the two curves overlap and the primal solution is recovered with a reasonable accuracy close to 10^{-5}. However for $\varepsilon_B = 10^{-6}$, there is a small loss in accuracy since the tolerance at both levels is the same. The most surprising result is for $\varepsilon_B = 10^{-5}$, the error does not reach astable plateau as the solution is completely wrong even though the top-level residual converges to the desired tolerance of 10^{-6}. This is attributed to the fact that in this case the tolerance at the bottom-level is higher than the tolerance at the top-level, which results in an incorrect top-level interface problem.

The computational costs for each of the bottom-level tolerances are shown in Table 5.8. It could be concluded that the third curve appears to be the cheapest with 3741 bottom-level iterations compared to the other curves as seen in However, the key objective is to get the most accurate solution at the cheapest cost. Since with the third curve there is a slight stagnation of accuracy for the primal solution, it could be argued that it is preferable to use the second curve with a slightly higher computational cost. From these results, a useful guideline appears to be that the bottom-level tolerance should be chosen one order lower the tolerance at top-level.

5.4.2 Adaptive tolerance

Based on the results above, it could be interesting to explore the possibility of using a higher bottom-level tolerance for the first top-level iterations and then refining the bottom-level tolerance in the latter top-level iterations. At an iteration k of the top-level problem, the tolerance on the bottom-level problem is controlled by the norm of the top-level residual $\|\mathbf{r}_k\|$ multiplied by a factor 10^b. In order to bound the value of the bottom-level tolerance, a threshold τ_B is chosen so that the bottom-level tolerance cannot exceed beyond this higher bound. This is useful especially in the first top-level iterations for which the residual of the top-level algorithm is quite high. It leads to

$$\varepsilon_B = \min(\tau_B, 10^b \|\mathbf{r}_k\|). \tag{5.21}$$

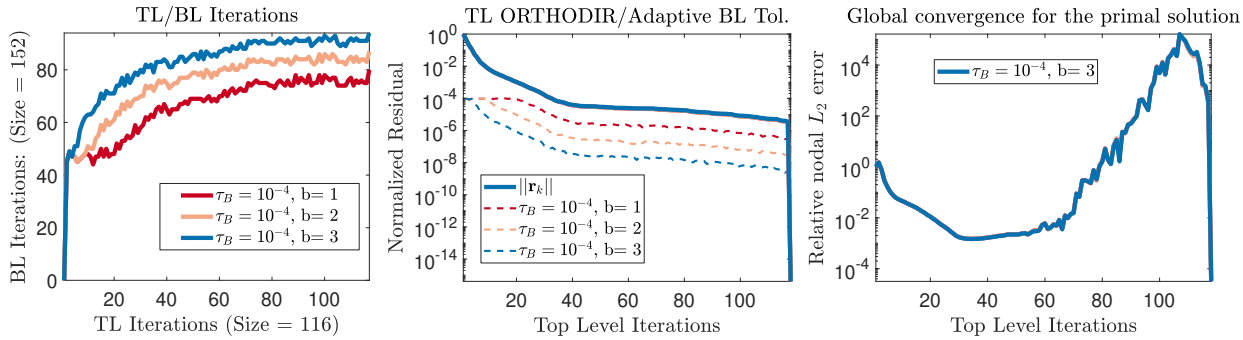


Figure 5.8: ORTHODIR with adaptive bottom-level tolerance for a fixed top-level tolerance of 10^{-6}, all top-level curves for the normalized residual fall on top of each other.

5.4.2.1 Adaptive tolerance: $\tau_B = 10^{-4}$

The first chosen threshold is $\tau_B = 10^{-4}$ which is two order of magnitude higher than the top-level tolerance $\varepsilon_T = 10^{-6}$. The evolution of the tolerance at the bottom-level is shown in dashed lines in Figure 5.8(b). The influence of the threshold is noticeable during the first iterations of the method. It can be seen that all the norm of the top-level residual are superposed (solid-blue line) which means that its evolution is independent of b. Similarly, the convergence of the primal variable is presented in Figure 5.8(c) and the three curves are superposed. As for fixed tolerance, a sharp rise is observed after 60 iterations. This phenomenon will be explained in the following of the section. Note that this instability appears despite the norm of the residual at the top-level slowly is decreasing and fluctuates until the last iteration at which the error drops to the machine precision (due to the fact that ORTHODIR computes all possible search directions, which is equivalent to a direct solve of the top-level interface problem).

Even if the number of bottom-level iteration is smaller in the first steps, which is positive, it reduces the convergence of the top-level method. By comparing Table 5.2 and Table 5.3, we see that we need 117 top-level iterations with the adaptative method while only 53 were necessary with the fixed tolerance analysis. It would be interesting to examine if this behavior is specific to the current threshold value. As a result, in the next part the threshold is decreased to examine the influence of the adaptive tolerance in greater detail.

Table 5.3: ORTHODIR adaptive tolerance studies for $\tau_B = 10^{-4}$.

ϵ_B	TL Iterations	BL Iterations
min $(\tau_B, 10^{-1}\|\mathbf{r}_k\|)$	117	7693
min $(\tau_B, 10^{-2}\|\mathbf{r}_k\|)$	117	8690
min $(\tau_B, 10^{-3}\|\mathbf{r}_k\|)$	117	9604

5.4.2.2 Adaptive tolerance: $\tau_B = 10^{-5}$

The threshold tolerance for the bottom-level is now set to 10^{-5} and the results are presented in Figure 5.9(b). Once more the normalized residuals of the top-level method are independent from parameter b.

Like in the previous example, the relative error for the primal variables stagnates then become unstable after 60 iterations, as seen in Figure 5.9(c). One observation is that, at any given top-level iteration in the computation, if the tolerance at the bottom-level is higher than the residual at the top-level, the accuracy of the primal variables is affected. Moreover, using smaller tolerances at the bottom-level in the latter iterations does not improve the accuracy of the final solution. The total number of bottom-level level iteration is once more quite high compared to the fixed tolerance study as seen in Table 5.4.

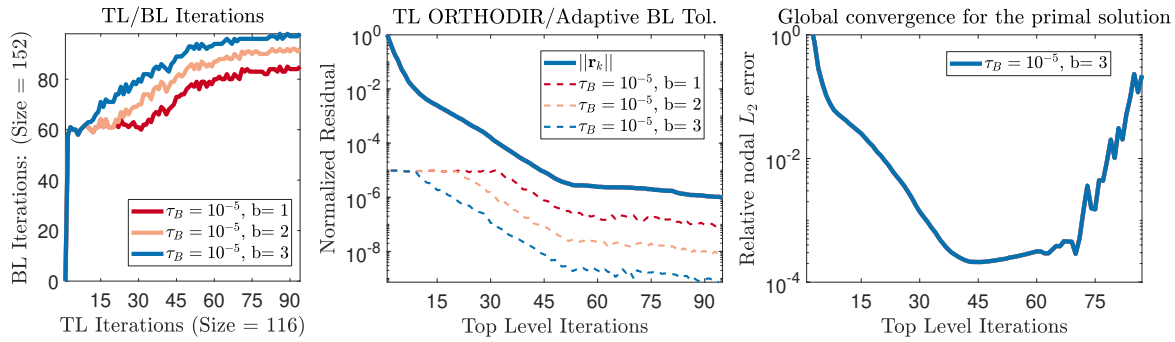


Figure 5.9: ORTHODIR with adaptive bottom-level tolerance for a fixed top-level tolerance of 10^{-6}, all top-level curves for the normalized residual fall on top of each other.

Table 5.4: ORTHODIR adaptive tolerance studies for $\tau_B = 10^{-5}$.

ϵ_B	TL Iterations	BL Iterations
min $(\tau_B, 10^{-1}\|\mathbf{r}_k\|)$	94	6734
min $(\tau_B, 10^{-2}\|\mathbf{r}_k\|)$	94	7301
min $(\tau_B, 10^{-3}\|\mathbf{r}_k\|)$	94	7902

5.4.2.3 Adaptive tolerance: $\tau_B = 10^{-6}$

Finally, we set the threshold 10^{-6} and the results are presented in Figure 5.10(b). Once more, all the curves associated to top-level convergence are superposed which means that they are independent from b. This time no plateau or unstable region is observed for the primal variables. The main conclusion is that when the bottom-level tolerance is at least as stringent as the top-level tolerance then the effect of varying the bottom-level tolerance is not visible on the accuracy of the solution.

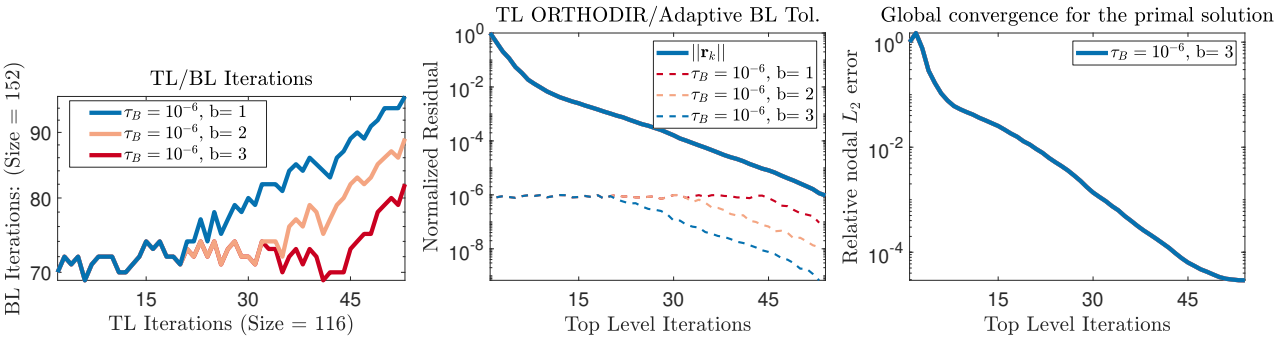


Figure 5.10: ORTHODIR with adaptive bottom-level tolerance for a fixed top-level tolerance of 10^{-6}, all top-level curves for the normalized residual fall on top of each other.

The computational costs of this for both these cases are quite similar, 3783 iterations for the adaptive tolerance case (shown with the red curve) compared to 3741 iterations for the fixed tolerance case 3. Additionally, it can be seen from Table 5.5 $b = 1$ is the most efficient in terms of computational cost since the accuracy of the primal solution is the same for all cases.

From this study, it can be said that it is not a good idea to use an adaptive tolerance for the bottom-level problem as the required threshold is too low. This results are a bit disappointing. In order to check this point a tolerance criteria from literature is examined for comparison.

Table 5.5: ORTHODIR adaptive tolerance studies for $\tau_B = 10^{-6}$.

ϵ_B	TL Iterations	BL Iterations
min $(\tau_B, 10^{-1}\|\mathbf{r}_k\|)$	53	3783
min $(\tau_B, 10^{-2}\|\mathbf{r}_k\|)$	53	3912
min $(\tau_B, 10^{-3}\|\mathbf{r}_k\|)$	53	4125

5.4.2.4 Adaptive tolerance: literature criterion

In the work by Ogino *et.al.* [73], a tolerance criterion is proposed for solving a two-level DDM problem. To reduce the number of iterations, a tolerance of 10^{-12} is used to calculate the initial residual at the bottom-level. For the latter iterations, an adaptive tolerance of $10^{-4}\|\mathbf{r}_k\|$ is used at the k^{th} iteration. The same approach is replicated with the current test case as seen in Figure 5.11(b). It is clear from the Figure 5.11(a) and Figure 5.11(c) that this criteria is not only expensive but in addition there is a loss of accuracy. A remark is that the results presented in the aforementioned work do not show the influence on the accuracy of the global solution.

In general one can say that using adaptive tolerance does does not improve the accuracy of the solution beyond the set threshold, τ_B. This is due to the fact that the top-level interface problem

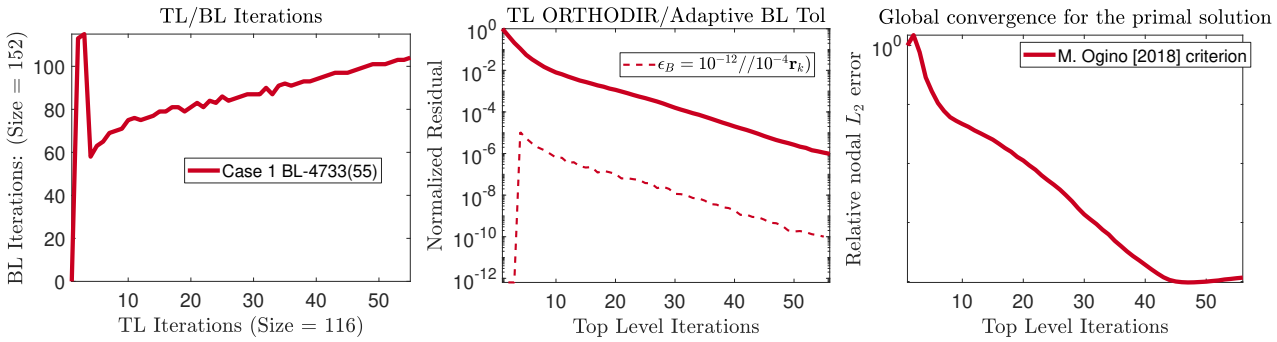


Figure 5.11: Application of the criterion proposed in [73] with a fixed top-level tolerance of 10^{-6}. The tolerance at the bottom-level is varied, for the first iteration is 10^{-12} and for latter iterations $10^{-4}r_k$.

inherits the error introduced by the bottom-level problem.

5.4.3 Conclusions for ORTHODIR

The results obtained with the ORTHODIR scheme for the two-level FETI-2LM method can be summarized as follows.

1. Between the fixed tolerance and the adaptive tolerance approaches, the former one yields the best results. The use of an adaptive tolerance at the bottom-level impacts the convergence at the top-level. In general it proves to be inefficient compared to the use of a fixed tolerance. In general, when the residual at the bottom-level is of higher magnitude than the tolerance at the top-level, the error of the primal variables tend to increase. Based on the numerical experiments reported above, the fixed tolerance at the bottom-level should be one order of magnitude smaller than the set tolerance at the top-level: $\varepsilon_B \sim \varepsilon_T/10$.

2. The accuracy of the primal variables is typically one order of magnitude less than the tolerance set for the top-level.

3. The accuracy of the primal variables exhibits a surprising behavior with an unstable solution. While the residual for the interface problem at the top-level decreases as expected, the error on the primal variables increases sharply.

This raises several question that need to be investigated further regarding the choice of the iterative scheme. Is the unstable behavior of the primal solution specific to ORTHODIR algorithm? Is it possible to control this regime of unstable numerical error at the top-level? Can adaptive tolerances scheme work better for GMRES or ORTHOMIN to reduce the computational cost? In order to address these issues, a similar analysis is carried out with the GMRES algorithm in the next section.

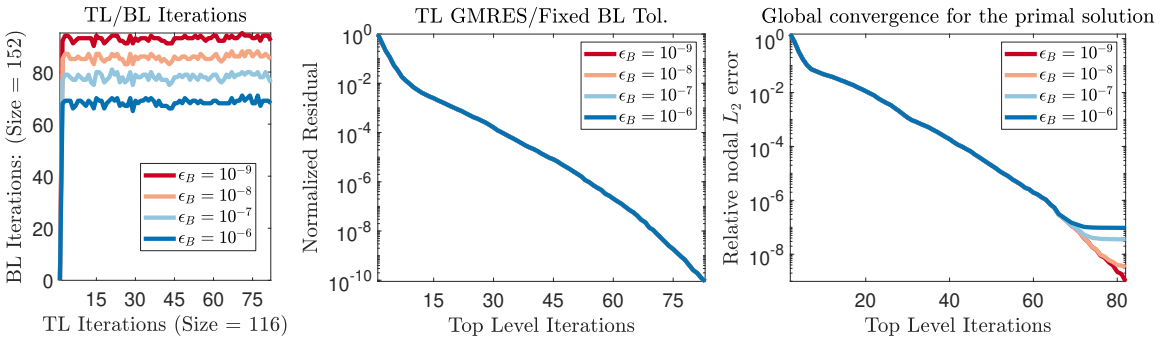


Figure 5.12: GMRES fixed tolerance studies for $\epsilon_T = 10^{-10}$.

5.5 Iterative resolution with GMRES

In this section, a similar study is done for the GMRES algorithm with either a fixed or adaptive tolerance. We remind that the difference between GMRES and ORTHODIR lies in the choice of the inner product chosen to orthonormalize the basis vector of the Krylov space, details are presented in the Chapter A.

5.5.1 Results for a fixed tolerance

Figure 5.12 shows the results obtained GMRES with a fixed bottom-level tolerance varying from 10^{-9} to 10^{-6}. The tolerance for the top-level is set to 10^{-10}. The choice of using a small top-level tolerance is made to study specifically the error at the bottom-level.

Table 5.6: GMRES fixed tolerance studies for $\epsilon_T = 10^{-10}$.

ϵ_B	TL Iterations	BL Iterations
10^{-9}	82	7524
10^{-8}	82	6937
10^{-7}	82	6326
10^{-6}	82	5641

As seen in Figure 5.12(b), the top-level convergence curves are superposed for all bottom-level tolerance. Moreover, the top-level interface problem does not suffer from a sluggish convergence, which is in contrast with what was observed with the ORTHODIR scheme: Contrary to the ORTHODIR scheme, the global convergence of the primal solution is quite stable, as seen in Figure 5.12(c). The curves flatten after the residual introduced by the bottom-level tolerance influences the accuracy of the primal solution.

In this case, the computational cost is higher than with the ORTHODIR algorithm. This is because

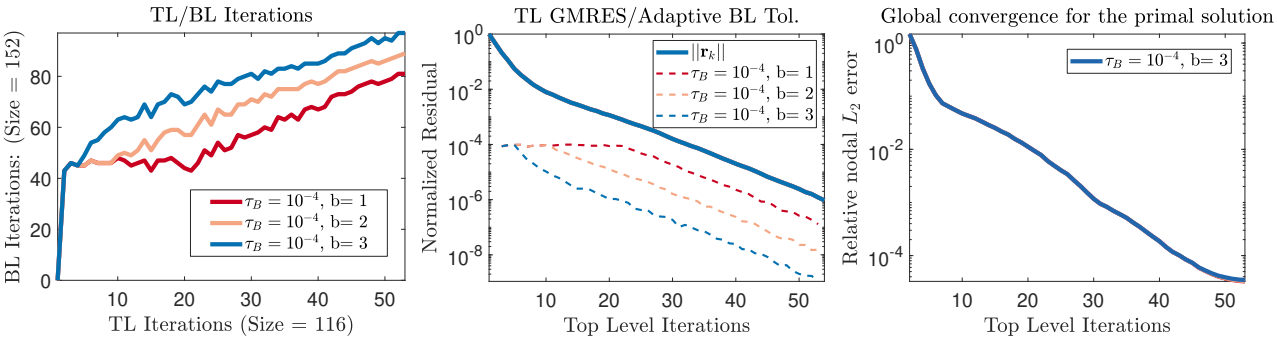


Figure 5.13: GMRES with adaptive bottom-level tolerance for a fixed top-level tolerance of 10^{-6}, all top-level curves for the normalized residual fall on top of each other.

the top-level tolerance was set to a much smaller value than in the previous case. However for the same top-level tolerance, (provided the bottom-level tolerance for ORTHODIR is the same or smaller than top-level tolerance) both iterative methods require similar computational cost.
Overall, it can be said that the GMRES algorithm is more stable and independent of the bottom-level tolerance compared to ORTHODIR.

5.5.2 Adaptive tolerance

We now examine if using an adaptive tolerance helps to improve the accuracy or in general affects the stability of the GMRES algorithm.

5.5.2.1 Adaptive tolerance: $\tau_B = 10^{-4}$

For the adaptive tolerance we follow the same approach as for ORTHODIR. The threshold is set to $\tau_B = 10^{-4}$. It can be seen in Figure 5.13 (b) that the adaptive tolerance does not have any influence on the convergence of the top-level problem. As seen in Figure 5.13 (c), the accuracy of the primal solution is slightly more than 10^{-4} for all the curves. Refining the tolerance for latter iterations does not help improve the accuracy of the solution. It can thus be said that using adaptive tolerances at the bottom-level does not help improve the accuracy of the primal solution.

Table 5.7: GMRES adaptive tolerance studies for $\tau_B = 10^{-4}$.

τ_B	ϵ_B	TL Iterations	BL Iterations
10^{-4}	min $(\tau_B, 10^{-1}\|\mathbf{r}_k\|)$	53	3009
10^{-4}	min $(\tau_B, 10^{-2}\|\mathbf{r}_k\|)$	53	3449
10^{-4}	min $(\tau_B, 10^{-3}\|\mathbf{r}_k\|)$	53	3944

The study of adaptive tolerance for GMRES algorithm at both levels results to the same conclusion

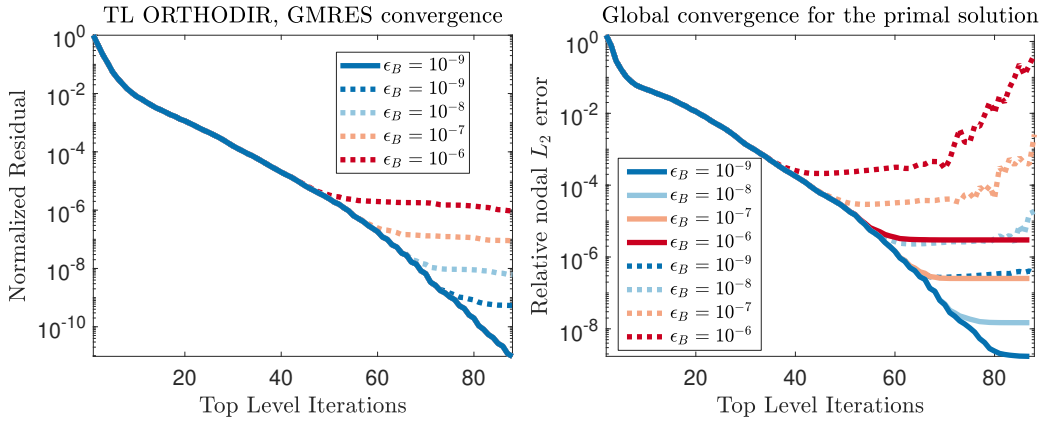


Figure 5.14: Comparison of ORTHODIR and GMRES algorithms for fixed 85 top-level iterations, the tolerance at the bottom-level is varied. ——represent GMRES results, whereas ·····denote ORTHODIR results.

as described for ORTHODIR. In general the adaptive tolerance does not seem to an efficient way to reduce the computational cost.

With the tolerance studies done so far, it can be said that the use of GMRES leads to a stable two-level FETI-2LM method. Moreover, it is clear that the unstable behavior observed in ORTHODIR is specific to the iterative scheme. In the next part, we compare the iterative behavior of GMRES and ORTHODIR for the fixed tolerance case.

5.5.3 Comparison of ORTHODIR, GMRES

The results for ORTHODIR and GMRES algorithms are now compared for the fixed tolerance criterion in Figure 5.14. The computations are done for a fixed number of top-level iterations. For the current analysis, the first 85 top-level iterations are presented. The tolerance at the bottom-level is varied from 10^{-9} to 10^{-6} (from blue to red). The dashed lines represent ORTHODIR results whereas the solid lines indicate GMRES results.

As seen in Figure 5.14 (a), the top-level convergence with ORTHODIR stagnates after the bottom-level tolerance limit is reached. However for GMRES, the top-level convergence is independent of the bottom-level tolerance. In addition, ORTHODIR solution is wrong and unstable for the cases with a bottom-level tolerance of 10^{-6} and 10^{-7}. However the GMRES solution is stable for all cases. In fact, for all cases, GMRES exhibits higher accuracy of the primal solutions compared to ORTHODIR.

We now discuss the results in Figure 5.15 in order to gain more insight into the reasons for the unstable behaviour of ORTHODIR, compared to GMRES.

We will study the evolution of the deviation of several quantities during the iterations at top-level. A

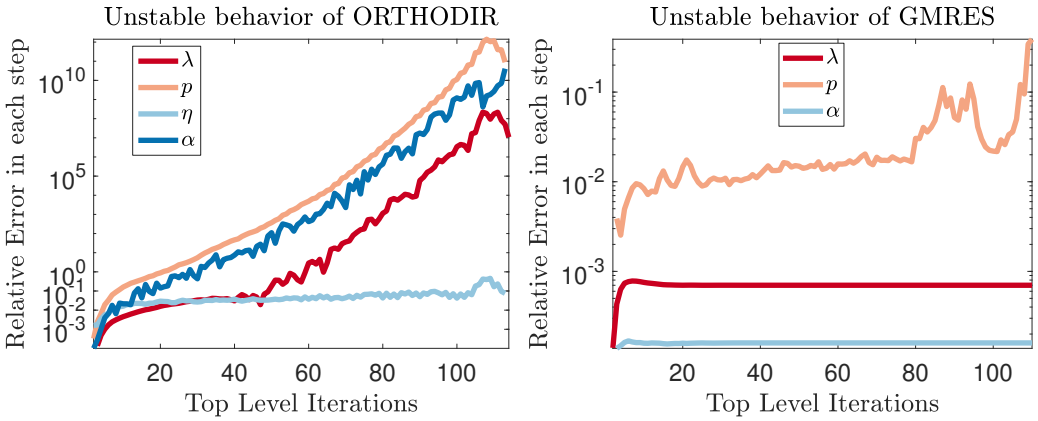


Figure 5.15: (a) Evolution of the unstable behavior in ORTHODIR, (b) evolution of the stable behavior in GMRES algorithm. The bottom-level tolerance is set to 10^{-3}.

reference solution is computed using a very small tolerance of $\varepsilon_B = \varepsilon_T = 10^{-16}$ for both the top and bottom-levels. The relative errors for the quantities $\tilde{\boldsymbol{\lambda}}$, $\boldsymbol{p}$, $\boldsymbol{\eta}$ and α are calculated against the reference values and shown in Figure 5.15. A remark is that in ORTHODIR the minimization coefficients are scalar quantities whereas in GMRES these are vector quantities.

With ORTHODIR, the search directions and the coefficients of the minimization problem blow up with increasing iteration count and reach an error of around 10 orders of magnitude. At each iteration, the solution for the dual variables is updated using the product of search directions with the coefficients of the minimization problem. Given that these two quantities diverge, the dual variables, and in turn the primal variables, will be inaccurate. Specifically, the coefficients for orthonomalization (Step 2(g)) in ORTHODIR algorithm) blow up drastically in the latter iterations. It therefore appears that the cause of the problem with ORTHODIR lies in the Gram-Schmidt orthogonalization process.

In contrast, with GMRES the errors in the search directions and the coefficients of the minimization problem remain small and do not blow up. As a result the error in the computation of the dual variables remains stable. Thus the orthogonalization process for GMRES seems be better suited for the two-level FETI-2LM algorithm. From the above analysis, it is clear that using GMRES provides a more stable two-level FETI-2LM algorithm. However using adaptive tolerance scheme does not reduce the computational effort compared to a standard (one-level) FETI approach. Another iterative solver, called ORTHOMIN, solves computes the MV product with the actual residual error at each step (see Section A.3). This motivates to examine the influence of ORTHOMIN on the two-level FETI-2LM algorithm.

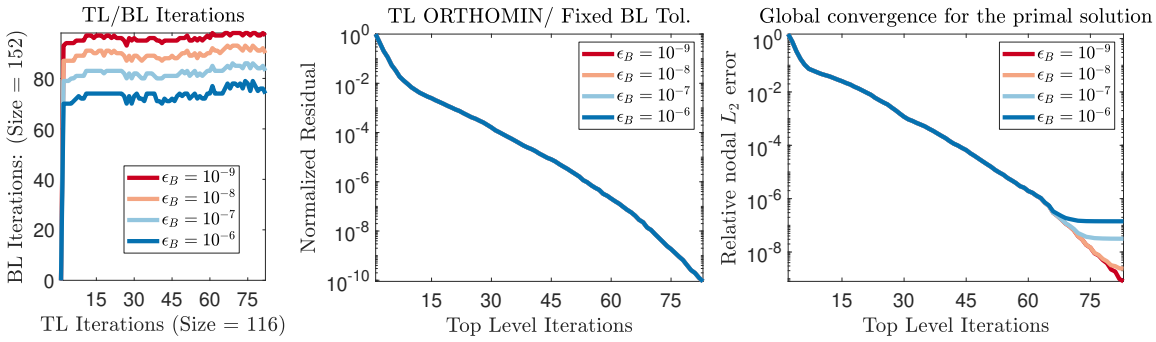


Figure 5.16: Fixed tolerance study for ORTHOMIN. Top-level tolerance: 10^{-10} for varying bottom-level tolerance.

5.6 Iterative resolution with ORTHOMIN

In this section, a third iterative solver, ORTHOMIN, is used at the top and bottom-levels. The ORTHOMIN algorithm is outlined in Section A.3.

For the same test case as before, the top-level tolerance is set to 10^{-10} with the same four bottom-level tolerance values used in Section 5.5.1. The results are shown in Figure 5.16. It can be observed that the results are very similar compared to Section 5.5.1. There is a slight variance in the computational cost as seen from Table 5.8, Table 5.6 which can be neglected. For this test case, there is no apparent difference between ORTHOMIN and GMRES. However, as explained in Section A.3, they rely on different ways to orthogonalise their vector bases.

Table 5.8: ORTHOMIN fixed tolerance studies for $\epsilon_T = 10^{-10}$.

ϵ_B	TL Iterations	BL Iterations
10^{-9}	82	7774
10^{-8}	82	7262
10^{-7}	82	6660
10^{-6}	82	5960

To have a more precise assessment of the robustness of the orthogonalization methods, it is useful to measure the orthogonality between the vectors of the different bases introduced in each iterative method. For this purpose, we compute the scalar products between the very first vector in the basis with the new vectors successively added to the basis. For ORTHODIR and ORTHOMIN, this involves the vectors $\boldsymbol{\eta}_k$, while for GMRES this involves the search directions $\boldsymbol{p}_k$. The results shown in Figure 5.17 depict the quality of orthogonalization for different iterative schemes.

It is clear that with ORTHOMIN the orthogonality constraint is enforced in a much more accurate

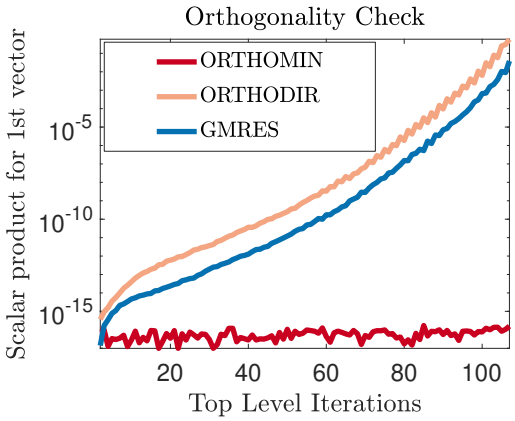


Figure 5.17: Orthogonality between the first vector and the new vector added at each iteration.

manner. In contrast, with ORTHODIR and GMRES, the orthogonality is less and less satisfied as these iterative schemes progress, at least the way the algorithms are implemented. While this difference between ORTHOMIN and GMRES didn't translate into a visible difference for the test case used in Figure 5.16, one could expect that for calculations involving many iterations the error accumulated by GMRES may become significant which might impact the convergence of the method. Finally, note also that there is a gap of at least of order of magnitude between GMRES and ORTHODIR, the former being more accurate.

This assessment of the orthogonality of the vector bases used by the different iterative schemes suggests that ORTHOMIN is the best suited when using a two-level FETI method as it is more robust than the other two schemes.

5.7 Conclusions

A two-level FETI-2LM formulation was presented. The key idea is to split the interface problem arising in the one-level FETI-2LM method into two parts. The first part where both the matrix and the right-hand side of the interface problem change (top-level) and the second part where only the right-hand side of the interface problem changes (bottom-level). The implementation of this algorithm was verified against the one-level FETI-2LM method. The stability of the proposed method was assessed in detail for three different iterative schemes, namely ORTHODIR, ORTHOMIN and GMRES.

It was found that the ORTHOMIN algorithm is most suited for the two level FETI-2LM algorithm. It provides a stable primal solution irrespective of the tolerance criteria. The performance of GMRES is very similar to ORTHOMIN for the two-level FETI-2LM method in terms of computational cost and accuracy. However, the orthogonality of the vectors in ORTHOMIN is more accurately enforced

compared to GMRES.

The most efficient tolerance criterion in terms of computational cost and accuracy is to set the tolerance at the bottom-level one order of magnitude smaller than the tolerance at the top-level. This criterion applies equally to all three iterative schemes.

The choice of adapting the tolerance at the bottom-level based on the normalized residual at the top-level does not bring significant improvement in computational cost.

Now that this two-level FETI approach has been presented and validated in details, we will consider in the next chapter the use of recycling strategies.

Chapter 6

Recycling Strategies

6.1 Introduction

In the previous chapter, a two-level FETI-2LM algorithm was proposed and verified. The top and bottom-levels are both solved using different iterative algorithms. The interface problem for the bottom-level is therefore solved repeatedly for different right-hand sides. There is an opportunity to accelerate the convergence of the bottom-level problem by using so-called recycling strategies which reuse some information from the previous resolutions of the bottom-level problem. At first we will have a closer look at these recycling strategies.

Iterative methods are based on the approximation of the solution in a subspace of relatively small dimension. A common approach is to use the Krylov subspace, which is progressively constructed by adding and orthogonalising a new vector at each iteration. When solving a single linear system, the standard approach is to build this Krylov subspace starting from zero. However, when solving a succession of linear systems, it is possible to reuse all, or part of, the Krylov subspace generated during the resolution of the previous system. These techniques are called subspace recycling. They lead to a reduction of the number of iterations and therefore a reduction of the cost of the solution procedure.

Krylov subspace recycling for successive resolutions has been applied in many engineering applications like large-scale optimization problems, time-dependent dynamic problems, etc. to reduce the computational cost [51]. These techniques can be typically split into two main categories (a) only the right-hand side changes between successive evaluations, (b) the matrix and the right-hand side both change (but the size of the matrix does not change).

The problem of solving the same system with multiple right-hand sides has been tackled in [32] by reusing the information from the conjugate gradient algorithm to efficiently reuse the previously gener-

ated Krylov subspaces. Several relevant work have been reported considering selective or total reuse of the Krylov subspaces for tackling structural dynamics and non-linear elasticity problems using FETI methods [82, 56]. However, the discrete systems for these problems are symmetric, positive-definite, which is not the case in the present work. FETI-2LM methods have also been proposed for Maxwell equations with multiple right-hand-side systems using an initial Krylov recycling strategy (IKRS) and block Krylov recycling strategy (BKRS) [83]. The BKRS offers several computational advantages. Firstly, the matrix-vector products in each iteration are replaced by matrix-matrix products which reduces the computational overhead since more data is exchanged for a reduced number of times. Secondly, a considerable reduction in the number of iterations is observed, see for instance Figure 6 in [83]. A detailed overview of the block Krylov methods is presented in [84][sec 6.12]. However, with the two-level FETI-2LM method proposed in the present work, the multiple right-hand sides are not available *a priori*. In this case, the Initial Krylov Recycling Strategy (IKRS) strategy can be applied, instead of the BKRS, to the sequence of linear systems arising from the bottom-level interface problem In order to deal with problems where both the matrix and the RHS change, several recycling approaches are proposed [79, 80, 75, 16]. GCRO-DR (Generalized conjugate residual with inner orthgonalization and Deflated Restart) is a Krylov subspace method where part of the Krylov subspace is retained at the restart of the next calculation. The idea is to deflate (i.e. reduce) the size of the Krylov subspace by focusing only on a subset of the eigenvalues of the system. However, this approach only works when the resolution of the problem to be solved does not change between successive calculations. In this work, a GCRO-DR method is applied to solve the interface problem arising from FETI-methods presented in Chapter 4.

This chapter does not adapt or formulate any new recycling strategies. The FETI-methods are adapted to be suitable for efficient recycling and applied for the two different recycling strategies. It is organized as follows. In the first part, a specific recycling approach, the Initial Krylov recycling strategy (IKRS) is implemented and applied to the two-level FETI-2LM method. In the second part, the GCRO-DR recycling strategy is applied for the interface problem of the one-level FETI-2LM method.

The originality of this chapter lies in the application of the standard and matured recycling strategies for novel FETI-methods. We judge the performance of these recycling strategies solely on the effective reduction in the new iterations for each successive calculation.

6.2 Full recycling

6.2.1 Description of the method

The IKRS can be used to solve a sequence of linear systems of equations where only the right-hand side changes. The main idea is to reuse the full subspace constructed during the previous resolutions to calculate an improved initial condition to start the next iterative resolution.
We recall the generic bottom-level interface problem which needs to be solved for each top level iteration (see for instance Section 5.2.3).

$$\mathbf{F}_{\mathrm{I}}^{A,B}\tilde{\boldsymbol{\mu}} = \mathbf{d}_{\mu}, \tag{6.1}$$

where only $\mathbf{d}_{\mu}$ changes for all calculations. In Chapter 5, it was concluded that ORTHOMIN is a good choice to solve the two-level FETI-2LM model, when compared to ORTHODIR and GMRES iterative schemes. Thus, in this section, the IKRS is presented in conjunction with ORTHOMIN. However the IKRS can also be applied with ORTHODIR or GMRES using a similar approach.
As outlined in the Section A.3, the ORTHOMIN algorithm explicitly computes the search directions $\mathbf{v}_k$ and the projection $\boldsymbol{\eta}_k$ of these search directions on the interface matrix $\mathbf{F}_{\mathrm{I}}^{A,B}$. Let's assume that we have already solved the linear system with a given right-hand side. In the process we have performed k iterations of ORTHOMIN and we have therefore accumulated k search directions $\mathbf{v}_k$ and k projection vectors $\boldsymbol{\eta}_k$. For convenience, these are stored in two matrices as follows:

$$\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k] \ , \quad \mathbf{N}_k = [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, ..., \boldsymbol{\eta}_k] \ .$$

We now have to solve the linear system again, but with a different right-hand side. The initial guess for the iterative procedure is denoted $\boldsymbol{\mu}_{\text{initial}}$. The corresponding residual is $\mathbf{r}_{\text{initial}} = \mathbf{d} - \mathbf{F}_{\mathrm{I}}\boldsymbol{\mu}_{\text{initial}}$. The idea of IKRS is to improve these initial values by using the vectors $\mathbf{v}_k$ and $\boldsymbol{\eta}_k$ which are already computed. This is done as follows:

$$\tilde{\boldsymbol{\mu}}_{\text{initial}} = \boldsymbol{\mu}_{\text{initial}} + \mathbf{V}_k\boldsymbol{\alpha}_k \ , \quad \tilde{\mathbf{r}}_{\text{initial}} = \mathbf{r}_{\text{initial}} + \mathbf{N}_k\boldsymbol{\alpha}_k \ ,$$

where $\boldsymbol{\alpha}_k = \mathbf{N}_k\mathbf{r}_{\text{initial}}$.

These improved starting points are then used with the ORTHOMIN algorithm. But instead of starting with empty sets of search directions $\mathbf{v}$ and projection vectors $\boldsymbol{\eta}$, the algorithm starts off with the complete sets of vectors computed for the previous linear system, i.e. with the matrices $\mathbf{V}_k$ and $\mathbf{N}_k$. Therefore, when an additional search direction $\mathbf{v}_{k+1}$ is created, it is orthogonalised with the k previous vectors (which represents a slight increase in the computational cost compared to the classical

ORTHOMIN algorithm). If n iterations are needed to achieve the prescribed tolerance for this new right-hand side, it means that we will have accumulated $k+n$ search directions and projection vectors. They will then be used for the subsequent resolution of the linear system with yet another right-hand side.

The details of the complete algorithm are given in the Section A.3. It has been implemented and verified to solve successive bottom-level interface problems resulting from the two-level FETI-2LM model. An important remark is that with IRKS one has to store all the search directions $\mathbf{v}_k$ and also the corresponding projection vectors $\boldsymbol{\eta}_k$, which can be memory demanding. To mitigate this, one can try to discard part of the subspace, with the consequence of requiring more iterations for convergence. In the present work, the full subspace is reused and results are presented in the next part.

6.2.2 Numerical results

The IKRS algorithm is now applied to the two-level FETI-2LM model to solve the bottom-level interface problem for a given liner configuration.

To assess the benefits of IKRS, we use the same test case as described in Section 5.3. The mesh resolution and the number of partitions can be seen in Figure 6.1. 7 partitions are used so as to have a large bottom-level interface problem. The overall model involves approximately 100,000 DOFs, and the bottom-level interface problem has roughly 2200 DOFs. ORTHOMIN is used for both the top and bottom levels. The top-level tolerance is set to 10^{-6} and, following the guidelines identified in the Chapter 5, the bottom-level tolerance is set to 10^{-7}.

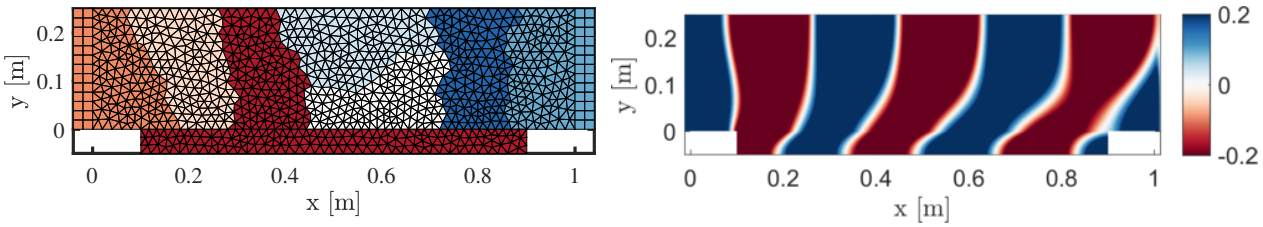


Figure 6.1: (a) Mesh with 7 partitions in the duct for the same test case as described in Section 5.3, (b) Example of numerical solution recovered using the two-level FETI-2LM method.

The first resolution of the bottom-level problem uses the standard ORTHOMIN algorithm since no information about the search directions is available. This step requires 185 iterations for convergence. The 185 search directions generated during this first resolution are reused for the second resolution to improve the initial guess used by ORTHOMIN, as explained above. Figure 6.2 shows the convergence of the ORTHOMIN algorithm for the second, third and fourth resolutions. Each graph plots the normalized residual error against the number of iterations. The red curve represents the reduction in

residual achieved by reusing the previously stored search directions to improve the initial solution of the ORTHOMIN algorithm. It consists of only scalar products whose computational cost is relatively small. The blue curve represents the reduction in residual during the new iterations required for the current resolution. These iterations represent a large part of the overall computational effort since at each new iteration, a matrix-vector product needs to be computed.

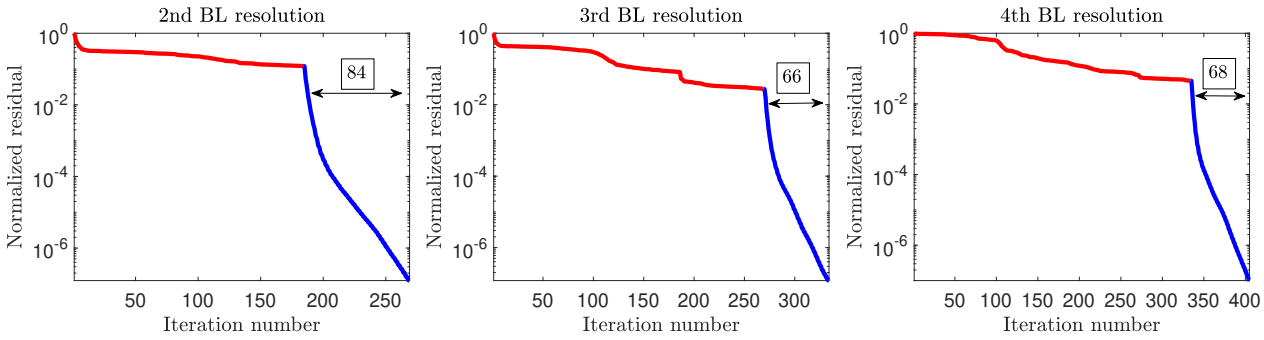


Figure 6.2: Normalized residual at the bottom level plotted against the number of iterations. Red curves represent the reduction in residual obtained from the previously stored search directions. Blue curves represent the new iterations required to converge to the desired tolerance.

As seen from Figure 6.2, the use of the previously computed search directions to improve the starting point of ORTHOMIN at each resolution does not reduce the residual by a large amount. For instance for the second resolution, the 185 existing search directions help to reduce the residual from 1 to only 0.2. Looking at the red curves in Figure 6.2, it is apparent that the existing search directions, taken individually, reduce the residual only by a small amount (one exception being for the third resolution). However, we can see from the three graphs in Figure 6.2 that the number of iterations needed for each successive resolution tend to decrease significantly.

This is confirmed over the whole resolution of the top-level problem. The top-level problem converges in 96 iterations, so one requires to solve the bottom-level problem 96 times. Figure 6.3 shows the number of iterations required for the successive resolution of the bottom-level problem. We can see that the number of iterations decreases significantly. In fact, after 20 resolutions of the bottom-level problem, the number of iterations required for convergence is very small. This rapid increase in the convergence rate of the resolution of the bottom-level problem is thanks to the recycling of previous search directions implemented with the IKRS. However, one should note that the cumulative total number of bottom-level resolutions required for this calculation is roughly 1500 iterations. This represents approximately 70% of the full size of the bottom-level interface problem. Storing all these search directions is however costly.

A similar test was performed using ORTHODIR instead of ORTHOMIN at both levels with the

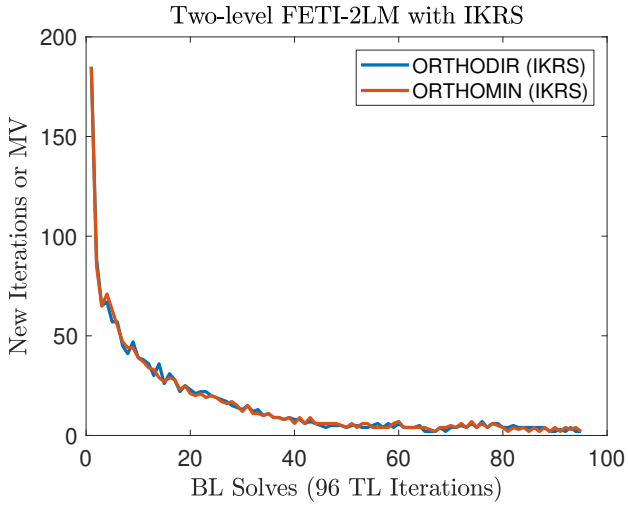


Figure 6.3: Number of iterations for each successive resolution of the bottom-level problem.

same test case. This leads to similar results as for the ORTHOMIN algorithm, as shown in Figure 6.3.

6.2.3 Conclusions

The two-level FETI-2LM algorithm has been examined for small 2D problems using the iterative schemes ORTHOMIN and ORTHODIR. The main observations are as follows.

Using the IKRS to accelerate the resolution of the two-level FETI model is beneficial but the cost of solving this model remains high. The reduction in the number of iterations provided by the IKRS is offset by the need to store a large set of search directions.

It should be noted that the dimensionality of the current test case is relatively small to provide general conclusions. However it is fair to say that, for the method to be computationally affordable, the number of bottom-level resolutions for all calculations should be sufficiently small compared to size of the interface problem.

In [60] the FETI-2LM method is applied for a 3D case to model realistic turbofan intakes, where the interface problem has a size of roughly 36,000 DOFs. In general, it can be concluded that this two-level FETI-2LM approach for relatively small sized problems is rather inefficient. For relatively high dimensional problems, this approach might show high potential savings. For instance, in [83] the interface problem has roughly 5 million DOFs, and thus this strategy provides significant gains.

6.3 Selective recycling

The previous section presented a method to use recycling for the two-level FETI method. The full subspace of the previous resolution was reused. In this section, we will only use a part of the subspace.

The question is therefore to what extent the previous search directions could be relevant, and efficient, for resolving the next configuration. The GCRO-DR [75] proposes a method to answer to this question. It is described in more details to solve several interface problems as described below:

$$\mathbf{F}_{\mathrm{I}}^{i}\,\tilde{\boldsymbol{\lambda}}_i = \mathbf{d}^i, \tag{6.2}$$

where the index i refers to the configuration under consideration.

One of the ways to select a part of the subspace is based on cheap approximations of the eigenspectrum of the relevant matrix. As an example, let's consider the eigenspectrum of the interface matrix of the one-level FETI-2LM method outlined in Chapter 4. As in the original paper, this method is applied to GMRES.

A simple lined duct test case is chosen as the configuration of interest with 5 different liner configurations for a liner depth $d = 0.05\,\mathrm{m}$ with varying flow resistivity from 2000 to 4000 at 1000 Hz. The eigenvalues for the associated interface problems are plotted in Figure 6.4 (a). It can be seen that the spectrum of $\mathbf{F}_{\mathrm{I}}$ (of fixed size 222 DOFs) does not change completely between these 5 configurations. In particular the smallest and largest eigenvalues only change a little. One can exploit this fact for faster convergence of the iterative methods in successive calculations.

GMRES leads to the following relation:

$$\mathbf{V}_{m_r}^{*}\mathbf{F}_{\mathrm{I}}\mathbf{V}_{m_r} = \mathbf{H}_{m_r}, \tag{6.3}$$

where $\mathbf{V}_{m_r}$ is the orthonormal Krylov basis generated at the previous iteration, $\mathbf{H}_{m_r}$ is a matrix composed of orthogonalization coefficients and m_r is the maximum number of iterations required for convergence of a specific calculation.

The previous relation can be interpreted as the projection of the interface matrix in the Krylov space and as it is of reduced size and Hessenberg, it is easy to compute it eigenvalues. In order to improve the numerical determination of its eigenvalues, the literature [75] suggests to consider the following solve the following eigenvalue problem:

$$(\mathbf{H}_{m_r} + h_{m_r+1,m}^{2}(\mathbf{H}_{m_r}^{-1})^{*})\mathbf{z} = \theta\mathbf{z}. \tag{6.4}$$

We now illustrate that using this combination well approximates the extreme eigenvalues of the interface matrix $\mathbf{F}_{\mathrm{I}}$. The first liner configuration corresponding to $\sigma = 2000$ is solved using FETI-2LM method which converges in 73 GMRES iterations. In this process it computes, $\mathbf{H}_{73}$ matrix. The eigenvalue problem described in Equation (6.4) is solved. In Figure 6.4 (b), all the θ eigenvalues are plotted. It can be clearly seen that all extreme eigenvalues are well captured. Thus the combination of $\mathbf{H}_{m_r}$ as described in Equation (6.4) gives a good approximation of the extreme eigenvalues of $\mathbf{F}_{\mathrm{I}}$.

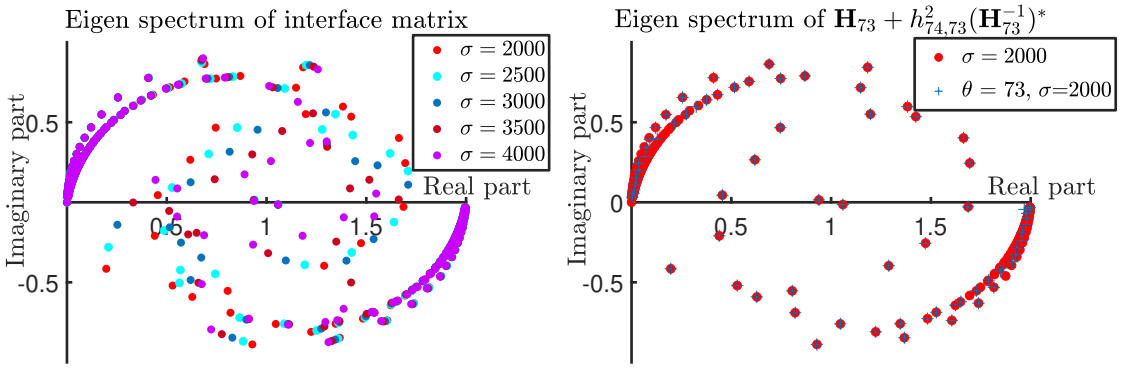


Figure 6.4: (a) Spectrum of the interface matrices for 5 different liner configurations at 1000 Hz, (b) The extreme eigenvalues are well approximated with a particular combination of the $\mathbf{H}_{m_r}$.

6.3.1 Description of GCRO-DR method

In the GCRO-DR algorithm, the calculation associated to the first configuration is a GMRES resolution. Let's say that it requires m_r iterations for convergence to the desired tolerance. A small part of this subspace k is computed based on the eigenvalues produced using Equation (6.4). These k eigenvectors are used to reduce the residual at the restart of next calculation and we will abbreviate the method as GCRO-DR (m_r, k). The details of this algorithm are outlined in the Section B.2 (see also the Appendix in [75]).

The first stage of the method consists in selecting the first k eigenvalues that will be preserved. Then, Additional iterative steps are performed to increase the basis. GCRO-DR controls these additional iterations limiting the size of the subspace m_r. Thus in that step, $m_r - k$ GMRES iterations are performed. The reduced subspace is generally not sufficient to guarantee the converge for the stored m_r iterations. If the solution has not converged, one needs to update the stored recycled subspace based on new eigenvalues.

However, the GCRO-DR algorithm works efficiently only if the matrix does not change significantly between calculations, which is a reasonable assumption in the current framework as seen in Figure 6.4 (a). In the GCRO-DR algorithm, there are three key aspects that need to be examined in detail, namely: the type of vectors to be chosen, the number of these vectors to be retained and the frequently with which the recycling subspace is deflated.

The same test case as described in Figure 6.1 is used to assess this aspect. It involves the single-level FETI-2LM model (as described in Chapter 4) with 8 subdomains (in total) and an interface problem of size 2600 approximately. Different configurations of the liner are considered by scanning two parameters:

1. Flow resistivity σ is varied from $2 \times 10^3 - 5 \times 10^4$ Nsm^{-4} using 5 equidistant points.

2. The liner depth d is varied from $0.05 - 0.1$m using 5 equidistant points.

6.3.2 Selection strategy

Concerning the selection of the kept vectors, one has the options to keep the smallest, largest or a mix of smallest and largest eigenvalues of Equation (6.4). In order to identify the most efficient choice of eigenvectors, three options are considered:

- Case 1: 150 vectors corresponding to smallest eigenvalues are retained between two configurations
- Case 2: 150 vectors corresponding to largest eigenvalues are retained between two configurations
- Case 3 : 150 vectors corresponding to 75 smallest and 75 largest eigenvalues are retained between two configurations

As a reference, a large span of the subspace ($m_r = 500$) is generated but only 150 eigenvectors are reused between two configurations. Table 6.1 (a) shows the number of iterations required by GMRES with $m_r = 500$ and by GCRO-DR(500,150) over all the 25 configurations. The choice of targeting the smallest eigenvalues appears to be the most efficient option. This is in agreement with the literature [75]. Roughly 2000 iterations are required for all configurations with option 1 compared to around 2400 iterations with option 3. Targeting the largest eigenvalues is the most inefficient option with more than 2700 iterations required.

Table 6.1: (a)Number of new iterations for the three different ways of deflating the subspace, (b) Number of new iterations depending on the number of eigenvectors retained between two configurations (150, 100 and 50 vectors).

	Smallest	Largest	Average
GMRES (500)	4375	4375	4375
GCRO-DR (500,150)	2041	2742	2404

	k=150	k=100	k=50
GMRES (500)	4375	4375	4375
GCRO-DR (500,k)	2041	2437	3125

6.3.3 Number of eigenvectors

The second aspect to study is the number of eigenvectors that are retained between configurations. Three different numbers of eigenvectors are considered: 150, 100 and 50. For these three options, Table 6.1 (b) shows the total number of iterations required to solve the 25 configurations. It is clear that choosing a large number of eigenvectors is more efficient to reduce the overall number of iteration. It should be noted that there is a cost, in terms of memory and computations, associated with reusing a large number of eigenvectors. Thus there is trade-off to find between maximising the

number of eigenvectors that are reused and minimising the amount of eigenvectors to store between two configurations.

6.3.4 Frequency of subspace deflation

In the previous part, suitable conclusions are drawn about the eigenvectors related to the recycling subspace k. However, one needs to refresh the recycling subspace if the stored vectors provide a poor approximation of eigenvalues linked to the interface matrix in the current calculation. The maximum span of the subspace m_r is used to dictate the frequency of refreshing the subspace. For instance if the current calculation requires more than $m_r - k$ iterations for convergence to the set tolerance, then the recycling subspace is refreshed.

The refreshing of the subspace comes with additional computational costs of solving an eigenvalue problem of size m_r. If the spectrum of the interface matrix changes drastically between successive evaluations, then the recycling subspace has to be refreshed often. In order to examine the behavior of varying the recycling subspace, a simple study with varying size of m_r is done. The following test cases are examined:

- Case 1: 150 smallest eigenvectors with maximum span of subspace = 500 i.e. GCRO-DR (500,150). It means that for each solve apart from the 150 stored search directions, a maximum of 350 more Arnoldi recurrences are performed in each calculation, before the recycling subspace is refreshed.
- Case 2: 150 smallest eigenvectors with maximum span of the subspace = 200 i.e. GCRO-DR (200,150). It means that for each solve apart from the 150 stored search directions, only 50 more Arnoldi recurrences are performed before the recycling subspace is refreshed. Thus a maximum of only 200 eigenvectors are stored at any given time, reducing the memory footprint of this case.

For each case, 25 different liner calculations are evaluated as mentioned in Section 6.3.3. The GMRES results are also presented for reference as seen in Table 6.2. It can be clearly seen that choosing a larger value of m_r seems to be more efficient. In addition, for case 2, one needs to solve the eigenvalue problem of size m_r involving additional operations. Indeed, case 2 limits the memory footprint at the expense of additional computations.

For the current problem, it can be said that using a large span of the recycling subspace m_r is more efficient to reduce the number of new MV products.

Table 6.2: Number of new iterations examined for two cases, $m_r = 500$ and $m_r = 200$, for both cases $k = 150$.

	m_r=500	m_r=200
GMRES (500)	4375	4375
GCRO-DR (m_r,150)	2041	3968

6.3.5 Constraint on liner partitions

In the present work, it is assumed that the liner is very small part of the computational domain, however if certain frequencies demand high problem resolution in the liner, one would require to partition the liner. The design space for the liner involves varying the liner depth. This could potentially result into unequal number of interface DOFS between successive calculations. This is not suitable for GCRO-DR algorithm as one assumes the problem size remains the same between successive calculations. Thus no partitions can be introduced in the liner medium.
The GCRO-DR recycling approach provides a systematic way to choose a part of the subspace, reducing the memory requirements for this approach. Moreover, significant reductions in the new iterations required for each calculation is observed.

6.4 Conclusions

Two different approaches have been considered in this chapter to accelerate the calculation of a series of model involving a range of liner configurations. The first approach combined the two-level FETI model presented previously with the Initial Krylov Recycling Strategy to reduce the number of iterations at the bottom-level. For this approach to yield practical benefits, the total number of top-level iterations for all calculations should be considerably small compared to bottom-level resolution. A benefit of the IKRS approach is that it provides more freedom to partition the computational domain.
The second approach is the GCRO-DR method applied to the single-level FETI model. For this approach, it can be said that it provides a significant reduction of the number of iterations required to achieved convergence. In addition this approach provides a systematic method to reduce the size of the recycling subspace, hence providing the possibility to reduce the memory footprint of the calculation. However, this approach only works when the resolution of the interface problem does not change between successive calculations.
The final workflow for computing several liner calculations using the FETI approach can be outlined

as follows

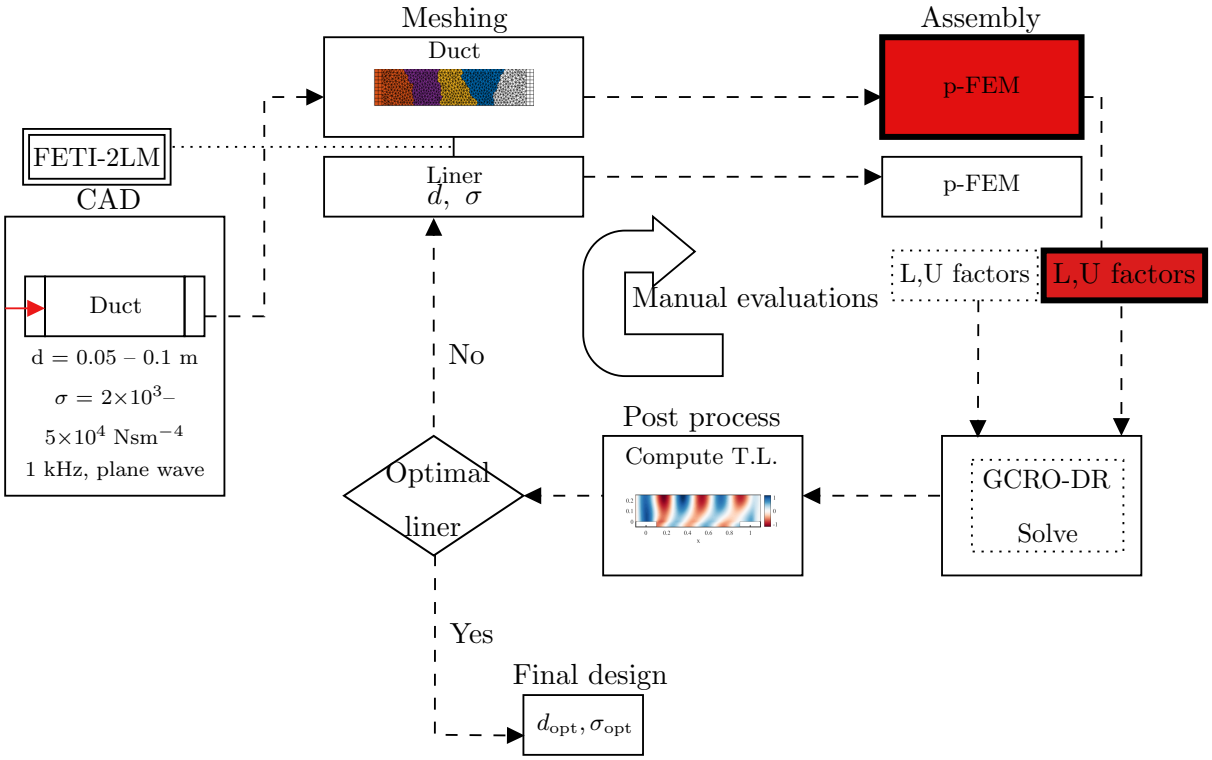


Figure 6.5: The proposed workflow with recycling for solving large-scale liner problems with acoustic treatments.

Chapter 7

Conclusions

This thesis focused on efficiently solving large-scale acoustic simulations in the frequency domain involving a range of acoustic treatments modelled as equivalent fluids. The central computational approach developed in this work was the FETI method for domain decomposition. The thesis addressed three specific areas. Firstly, the selection of an efficient FETI approach for homogeneous and heterogeneous Helmholtz problems, with a view to solve problems involving heterogeneous porous media. The second aspect addressed in the thesis was the computational issues related to the modeling of a succession of different acoustic treatments. The third aspect was to reduce the number of iterations required in each successive calculation using recycling techniques.

7.1 Summary of the main contributions

The following are the main contributions from this thesis:

1. In the first stage, the FETI-2LM and FETI-H methods were assessed when combined with high-order finite elements to solve homogeneous Helmholtz problems. Both methods were found to be scalable with the mesh size h and the polynomial order p. Contrary to literature studies, the FETI-2LM algorithm exhibited strong scalability with increasing wavenumbers but didn't scale when increasing the number of subdomains. On the other hand, FETI-H did not scale with increasing wavenumbers nor the number of subdomains. Except for small wavenumbers, the FETI-2LM method tends to outperform FETI-H. It should be noted that these observations were made for non-preconditioned versions of these two methods.

2. In the second stage, these two variants of the FETI approach were generalized to handle multiple equivalent fluids, each governed by the Helmholtz equation with complex sound speed and density

to account for the losses in a porous media. A detailed scalability study was then presented for FETI-2LM and FETI-H methods. The two methods demonstrated good scalability with varying mesh size h and polynomial order p. In addition, the FETI-2LM method showed good scalability with increasing wavenumbers. In fact, the number of iterations required to converge decreases when the wavenumber increases, which is a behavior specific to lossy propagation media. FETI-2LM also showed good scalability when increasing the number of partitions in the porous medium. However, it does not scale with increasing partitions in the air medium. In contrast, FETI-H does not scale well when increasing the wavenumber and the number of partitions in both media. In general, the performance of the FETI-H deteriorates for multiple fluids and FETI-2LM is clearly more efficient.

3. A novel factorization process was proposed to reduce the computational costs when performing a series of simulations where only the acoustic treatment changes. The FETI approach provides a way to reduce the factorization cost by reusing the LU factor already computed for the subdomain that do not change between two calculations (in this case the fluid region inside the duct). The potential computational savings in time and memory provided by this approach have been evaluated.

4. To further improve the performance of the proposed domain decomposition approach, a two-level FETI-2LM model was also devised. The rationale is to use a second FETI problem to solve the factorisation of the model in the sub-domains that do not change between two calculations. A detailed tolerance and stability analysis of this two-level approach was performed for several iterative schemes like ORTHODIR, ORTHOMIN and GMRES. Based on this study, the recommendation is to use a tolerance one order of magnitude smaller for the bottom-level interface problem compared to the top-level. It was found that using an adaptive tolerances for the bottom-level does not help to reduce the computational effort. While ORTHOMIN and GMRES demonstrate a stable behavior, ORTHODIR exhibits a unstable behaviour due to the error introduced by solving the bottom-level problem only approximately. However, the orthonormalization process of ORTHOMIN was found to be more robust compared to GMRES. ORTHOMIN is therefore recommended for solving the two-level FETI model.

5. Two recycling strategies were considered to accelerate the iterative resolution of the FETI model. The Initial Krylov Recycling Strategy (IKRS) was used to solve the two-level FETI-2LM algorithm and speed up the solution of the bottom-level problem. For relatively small problems, it requires the storage of a large number of search directions before the bottom-level problems

becomes significantly cheaper to solve. Overall, if the total number of top-level iterations for all calculations is negligible compared to the size of the bottom-level interface problem, then this approach might lead to significant benefits. The second recycling strategy consisted in applying the GCRO-DR method to the single-level FETI-2LM model. Using a maximum affordable number of eigenvectors associated with the smallest eigenvalues seems to be the most efficient approach to reduce the number of new matrix-vector products. A significant reduction in the number of new matrix-vector products can be achieved. For relatively small problems, this approach seems to be more efficient compared to the Initial Krylov Recycling Strategy. A final computational procedure was outlined to solve acoustic problems involving a large number of different liner configurations.

7.2 Perspectives

The present work contributed to accelerating numerical simulations of acoustic treatments involving a series of configurations, a situation which is very common when using optimisation algorithm for engineering design. Several directions of future research can be identified to continue this work:

1. When applying optimization methods to identify the optimal acoustic treatment in a given situation, the specifics of the optimization algorithm could play a role in choosing the best recycling method. For instance, if one resorts to using a Genetic Algorithm, then the calculation points in the design space could vary significantly from one point to the next, leading to significant changes in the eigenvalue spectrum of the interface matrix in FETI-2LM model. In such a situation, the GCRO-DR method is not expected to bring any significant benefits. However, for gradient based optimization strategies, successive calculation points in the design space are expected to be in close proximity. In that case, the GCRO-DR method should be much more efficient. On the other hand, if one decides to optimize the treatment for multiple duct modes at the same frequency, this involves solving a linear system of equations for different right-hand sides. The Block Krylov Recycling Strategy (BKRS) could prove to be very efficient for solving this type of problems for the FETI method. There is therefore a close connection between the design problem, the optimisation algorithm and the formulation of the domain decomposition method. Future work could explore this connection and identify the best combination of optimisation algorithm and recycling strategy, for instance.

2. It would be useful to consider a fully parallel implementation of the FETI model combined with a suitable recycling strategy, such as GCRO-DR. This would allow for a more precise assessment

of the benefits in terms of computational costs, in particular in the context of optimisation procedures exploring the design space of acoustic liners. The implementation of the GCRO-DR algorithm is not trivial and depends on the application and problem at hand.

3. The present thesis was focused on Helmholtz problems. It will be useful to consider other sound propagation models. For instance, in the context noise radiation from engine intakes, it would be useful to combine the equivalent fluid model to describe the porous treatments with the linearized potential equation to described sound propagation in a moving fluid. This would involve formulating new interface conditions between these two propagation models.

Scientific dissemination

Poster

1. Domain decomposition for modeling of acoustic liners. C. Sanghavi, H. Bériot, G. Gabard, and O. Dazel. *Smart Answer Workshop*, Leuven, Belgium, 2019.

Conference proceedings

1. A non-overlapping domain decomposition method-FETI-2LM applied for liner optimization in acoustic application. C. Sanghavi, H. Bériot, G. Gabard, and O. Dazel. *International Conference on Adaptive Modeling and Simulation ADMOS*, Alicante, Spain, 2019.

2. An analysis of a domain decomposition method applied to Helmholtz problem for efficient recycling. C. Sanghavi, H. Bériot, G. Gabard, and O. Dazel. 14^{th} *World Congress on Computational Mechanics (WCCM)-ECOMMAS*, Paris, France, 2020*.
   * Conference was cancelled due to COVID reasons.

Journal article

1. A domain decomposition method for modeling acoustic treatments. C. Sanghavi, H. Bériot, G. Gabard, and O. Dazel. *Journal of Sound and Vibration*, 2020. (under review)

Appendix A

Iterative methods

Overview

This Appendix details the iterative schemes - ORTHODIR, GMRES and ORTHOMIN used extensively throughout the thesis. Each of these methods consists of three key properties, namely the generation of a Krylov subspace, the orthogonalization process and the minimization problem to be solved at each iteration.

For all the iterative schemes presented, we assume to solve a linear system of equation corresponding to the interface problem of the FETI-methods. We recall the definition of this interface problem as follows:

$$\mathbf{F}_{\mathrm{I}}\tilde{\boldsymbol{\lambda}}_0 = \mathbf{d}$$

All the aforementioned iterative methods consist of iteratively building an orthonormal basis of the Krylov space $K_k = \mathrm{Span}\{\mathbf{r}_0,\ \mathbf{F}_{\mathrm{I}}\mathbf{r}_0,\ \mathbf{F}_{\mathrm{I}}^2\mathbf{r}_0,\ ...\mathbf{F}_{\mathrm{I}}^{k-1}\mathbf{r}_0\}$ with $\mathbf{r}_0 = \mathbf{d} - \mathbf{F}_{\mathrm{I}}\tilde{\boldsymbol{\lambda}}_0$.

However the key differences in these methods lies in the orthogonalization process and the way the minimization problem is solved. The Gram-Schmidt process, Householder transformations and Givens rotations are commonly used for generating an orthonormal basis. In the present work, the modified Gram-Schmidt process is used to avoid round-off errors to produce k^{th} orthgonalized vector after the k^{th} step. Let $\mathbf{V}_k$ be the set of orthonormal basis generated using the MGS (Modified Gram Schmidt) - process where each column "k" contains the orthonormal basis at the k^{th} step.

The last part consists of solving the minimization problem to get an update on the solution at each step. In the next part, each of these iterative schemes are examined in more detail.

A.1 ORTHODIR

The ORTHODIR algorithm consists of building the Krylov subspace K_k using the following constraint:

$$(\mathbf{F}_\mathrm{I}\mathbf{V}_k)^*(\mathbf{F}_\mathrm{I}\mathbf{V}_k) = \mathbf{I}_k, \tag{A.1}$$

where, $\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k]$, $\mathbf{v}_k$ represents the basis vectors.
In order to impose this constraint the projection of the basis vectors on the interface problem i.e. $\mathbf{F}_\mathrm{I}\mathbf{V}_k$ need to be stored in addition to the basis vectors $\mathbf{V}_k$, which leads to increased memory requirements. The least square minimization problem is solved to minimize the norm of the residual at the k^{th} in the computation of $\mathbf{y}_k$:

$$\mathrm{J}_{\mathbf{y}_k} = \begin{cases} \|\mathbf{d} - \mathbf{F}_\mathrm{I}\tilde{\boldsymbol{\lambda}}_k\|_2, \\ \|\mathbf{r}_0 + \mathbf{F}_\mathrm{I}\mathbf{V}_k\mathbf{y}_k\|_2. \end{cases}$$

Using the Galerkin projection to compute, $\mathbf{y}_k$ results in the following relation:

$$(\mathbf{F}_\mathrm{I}\mathbf{V}_k)^*(\mathbf{r}_0 + \mathbf{F}_\mathrm{I}\mathbf{V}_k\mathbf{y}_k) = 0. \tag{A.2}$$

Using the orthogonality relation described in Equation (A.1), simplifies the expression for the minimization problem as described below:

$$\mathbf{y}_k = -(\mathbf{F}_\mathrm{I}\mathbf{V}_k)^*\mathbf{r}_0. \tag{A.3}$$

In the latter stage, the approximate solution can be formulated using the following equation:

$$\tilde{\boldsymbol{\lambda}}_k = \tilde{\boldsymbol{\lambda}}_0 + \mathbf{V}_k\mathbf{y}_k. \tag{A.4}$$

The scalar product between two vectors a, b is denoted as (a, b). A detailed process of the ORTHODIR algorithm can be described as follows:

1. Initialization: Given an initial guess $\tilde{\boldsymbol{\lambda}}_0$, compute:
   - the residual at the initial step $\mathbf{r}_0 = \mathbf{d} - \mathbf{F}_\mathrm{I}\tilde{\boldsymbol{\lambda}}_0$.
   - the first basis vector $\mathbf{v}_0 = \mathbf{r}_0$.
   - its product by $\mathbf{F}_\mathrm{I}$: $\boldsymbol{\eta}_0 = \mathbf{F}_\mathrm{I}\mathbf{v}_0$.
   - normalise $\boldsymbol{\eta}_0$ with $\alpha_1 = 1/\sqrt{\boldsymbol{\eta}_0, \boldsymbol{\eta}_0}$ and

$$\begin{cases} \mathbf{v}_0 \longleftarrow \alpha_1\mathbf{v}_0. \\ \boldsymbol{\eta}_0 \longleftarrow \alpha_1\boldsymbol{\eta}_0. \end{cases}$$

2. Iterations (for k = 0,1,2,.... until convergence):

   (a) Compute the k+1 solution:

   $$\tilde{\boldsymbol{\lambda}}_{k+1} = \tilde{\boldsymbol{\lambda}}_k + \alpha_2 \mathbf{v}_k,$$

   where $\alpha_2 = (\mathbf{r}_k, \boldsymbol{\eta}_k)$. owing to the constraint from expression Equation (A.4).

   (b) Compute the corresponding residual:

   $$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_2 \boldsymbol{\eta}_k.$$

   (c) Check if $\mathbf{r}_{k+1} \leq \epsilon$, stop ; otherwise, continue

   (d) Compute matrix-vector product $\boldsymbol{\eta}_{k+1} = \mathbf{F}_{\mathrm{I}} \boldsymbol{\eta}_k$.

   (e) Initialize the summation: $\mathbf{v}_{k+1} = \boldsymbol{\eta}_k$.

   (f) For $l = 0, ...k$, compute :

   $$\beta_{k,l} = \frac{(\boldsymbol{\eta}_{k+1}, \boldsymbol{\eta}_l)}{(\boldsymbol{\eta}_l, \boldsymbol{\eta}_l)}.$$

   (g) Find the next basis vector:

   $$\mathbf{v}_{k+1} \longleftarrow \mathbf{v}_{k+1} + \sum_{l=0}^{k} \beta_{k,l} \mathbf{v}_l.$$

   $$\boldsymbol{\eta}_{k+1} \longleftarrow \boldsymbol{\eta}_{k+1} + \sum_{l=0}^{k} \beta_{k,l} \boldsymbol{\eta}_l.$$

   end

   (h) Normalize $\boldsymbol{\eta}_{k+1}$ with $\alpha_1 = 1/\sqrt{\boldsymbol{\eta}_{k+1}, \boldsymbol{\eta}_{k+1}}$ and

   $$\begin{cases} \mathbf{v}_{k+1} = \alpha_1 \mathbf{v}_{k+1}, \\ \boldsymbol{\eta}_{k+1} = \alpha_1 \boldsymbol{\eta}_{k+1}. \end{cases}$$

   (i) Update iteration number k $\longleftarrow$ k+1.

A.2 GMRES

The GMRES method consists of building the Krylov subspace K_k and the orthonormal basis $\mathbf{V}_k$. It imposes the following constraint in the orthognalization process

$$(\mathbf{V}_k)^*(\mathbf{V}_k) = \mathbf{I}_k \tag{A.5}$$

This method generates a set of Arnoldi vectors using the modified Gram-Schmidt process which have special physical interpretations. The Arnoldi iteration method helps to approximate the eigenvalues and eigenvectors of the interface problem $\mathbf{F}_\mathrm{I}$ as follows: If the first basis vector is chosen as the normalized residual, the following property holds true:

$$\mathbf{V}_k^* \mathbf{F}_\mathrm{I} \mathbf{V}_k = \mathbf{H}_k, \tag{A.6}$$

where $\mathbf{H}_k$ is known as the upper Hessenberg matrix formed using the coefficients generated in the MGS algorithm. The eigenvalues of the Hessenberg matrix can provide accurate approximations of specific eigenvalues of the original matrix, which can potentially be exploited for recycling strategies. The same minimization problem as described in Section A.1 is solved, however using a different projection as follows:

$$\mathbf{V}_k^* (\mathbf{r}_0 + \mathbf{F}_\mathrm{I} \mathbf{V}_k \mathbf{y}_k) = 0. \tag{A.7}$$

Using the relation in Equation (A.6) results in the simplification of the minimization problem as follows:

$$(\mathbf{V}_k^* \mathbf{r}_0 + \mathbf{H}_k \mathbf{y}_k) = 0. \tag{A.8}$$

The first basis vector is chosen as the normalized residual and the remaining vectors are orthonormal to the space of the vectors $\mathbf{V}_k$. This constraint gives the expression for the minimization problem as follows:

$$\mathbf{y}_k = \mathbf{H}_k^{-1} ||\mathbf{r}_0|| \mathbf{e}_1. \tag{A.9}$$

The approximate solution can be formulated using the following equation:

$$\tilde{\boldsymbol{\lambda}}_k = \tilde{\boldsymbol{\lambda}}_0 + \mathbf{V}_k \mathbf{y}_k. \tag{A.10}$$

A detailed process of the GMRES algorithm used in the present work is described below:

1. Initialization: Given an initial guess $\tilde{\boldsymbol{\lambda}}_0$, compute:

   - the residual at the initial step $\mathbf{r}_0 = \mathbf{d} - \mathbf{F}_\mathrm{I} \tilde{\boldsymbol{\lambda}}_0$,
   - the first basis vector $\mathbf{v}_0 = \mathbf{r}_0$,
   - normalise with $\alpha_1 = 1/\sqrt{\mathbf{v}_0, \mathbf{v}_0}$

     $$\mathbf{v}_0 \longleftarrow \alpha_1 \mathbf{v}_0,$$

2. Iterations (for k = 0,1,2,.... until convergence):

   Check if $\mathbf{r}_{k+1} \leq \epsilon$, stop ; otherwise, continue

(a) Compute matrix-vector product $\mathbf{w} = \mathbf{F}_I \mathbf{v}_k$.

(b) For $l = 0, ...k$, compute :

$$h_{k,l} = \frac{(\mathbf{w}, \mathbf{v}_l)}{\mathbf{v}_l, \mathbf{v}_l}.$$

(c) Find the next basis vector:

$$\mathbf{v}_{k+1} = \mathbf{w} + \sum_{l=0}^{k} h_{k,l} \mathbf{v}_l.$$

(d) Normalize $\mathbf{v}_{k+1}$ with $\alpha_2 = 1/\sqrt{\mathbf{v}_{k+1}, \mathbf{v}_{k+1}}$ and

$$\mathbf{v}_{k+1} \longleftarrow \alpha_2 \mathbf{v}_{k+1}.$$

(e) Solve the least square problem to compute the minimizer $\mathbf{y}_{k+1}$.

i. Initiate $\mathbf{s}_{k+1} = \mathbf{0}$.

ii. The residual at the first step is introduced, $\mathbf{s}_1 = ||\mathbf{r}_0||$.

iii. The minimizer $\mathbf{y}_{k+1} = \mathbf{H}^{-1} \mathbf{s}_{k+1}$,

where H is the upper Hessenberg matrix comprising of scalar coefficients of $h_{k+1,k}$.

(f) Compute the approximate solution :

$$\tilde{\boldsymbol{\lambda}}_{k+1} = \tilde{\boldsymbol{\lambda}}_0 + \mathbf{V}_k \mathbf{y}_k.$$

$$\mathbf{r}_{k+1} = ||\mathbf{d} - \mathbf{F}_I \tilde{\boldsymbol{\lambda}}_{k+1}||/||\mathbf{d}||.$$

As seen in step (e), the problem requires the inversion of the Hessenberg matrix at each step which can be a computationally expensive operation especially for high number of iterations. This issue is addressed in the literature typically using the planes rotation [84][sec 6.5.3]. However the objective of the present work is to examine the influence of the iterative schemes for the FETI-method and not to reduce the computational effort of the GMRES method.

A.3 ORTHOMIN

ORTHOMIN computes the matrix vector products with the true residual at each iterative step. The steps in the algorithm can be outlined as follows:

1. Initialization: Given an initial guess $\tilde{\boldsymbol{\lambda}}_0$, compute:

- the residual at the initial step $\mathbf{r}_0 = \mathbf{d} - \mathbf{F}_I \tilde{\boldsymbol{\lambda}}_0$.
- the first basis vector $\mathbf{v}_0 = \mathbf{r}_0$.
- Its product by $\mathbf{F}_I$: $\boldsymbol{\eta}_0 = \mathbf{F}_I \mathbf{v}_0$.

- normalise $\boldsymbol{\eta}_0$ with $\alpha_1 = 1/\sqrt{\boldsymbol{\eta}_0, \boldsymbol{\eta}_0}$ and

$$\begin{cases} \mathbf{v}_0 \longleftarrow \alpha_1 \mathbf{v}_0, \\ \boldsymbol{\eta}_0 \longleftarrow \alpha_1 \boldsymbol{\eta}_0. \end{cases}$$

2. Iterations (for k = 0,1,2,.... until convergence):

   (a) Compute the k+1 solution:

$$\tilde{\boldsymbol{\lambda}}_{k+1} = \tilde{\boldsymbol{\lambda}}_k + \alpha_2 \mathbf{v}_k,$$

   where $\alpha_2 = (\mathbf{r}_k, \boldsymbol{\eta}_k)$.

   (b) Compute the corresponding residual:

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_2 \boldsymbol{\eta}_k.$$

   (c) Check if $\mathbf{r}_{k+1} \leq \epsilon$, stop ; otherwise, continue

   (d) Compute matrix-vector product $\boldsymbol{\eta}_{k+1} = \mathbf{F}_{\mathrm{I}} \mathbf{r}_{k+1}$.

   (e) Initialize the summation: $\mathbf{v}_{k+1} = \mathbf{r}_{k+1}$.

   (f) For $l = 0, ...k$, compute :

$$\beta_{k,l} = \frac{(\mathbf{w}, \boldsymbol{\eta}_l)}{(\boldsymbol{\eta}_l, \boldsymbol{\eta}_l)}.$$

   (g) Find the next basis vector:

$$\mathbf{v}_{k+1} \longleftarrow \mathbf{v}_{k+1} + \sum_{l=0}^{k} \beta_{k,l} \mathbf{v}_l.$$

   (h) Store the product of the basis vector by $\mathbf{F}_{\mathrm{I}}$:

$$\boldsymbol{\eta}_{k+1} \longleftarrow \boldsymbol{\eta}_{k+1} + \sum_{l=0}^{k} \beta_{k,l} \boldsymbol{\eta}_l.$$

   (i) Normalize $\boldsymbol{\eta}_{k+1}$ with $\alpha_1 = 1/\sqrt{\boldsymbol{\eta}_{k+1}, \boldsymbol{\eta}_{k+1}}$ and

$$\begin{cases} \mathbf{v}_{k+1} \longleftarrow \alpha_1 \mathbf{v}_{k+1}, \\ \boldsymbol{\eta}_{k+1} \longleftarrow \alpha_1 \boldsymbol{\eta}_{k+1}. \end{cases}$$

   (j) Update iteration number k $\longleftarrow$ k+1.

Key differences

All of these iterative methods rely on using an orthonormal basis, MGS technique for orthogonalization and solving the minimization problem using a Galerkin projection of the Krylov subspace. In the case of infinite precision and computational effort, each of these methods would converge to the exact same solution in the same number of iterations for a given problem resolution. However, in finite precision especially for set tolerance criteria, the behavior of these methods becomes strongly dependent on the way the algorithms are implemented.

ORTHODIR and ORTHOMIN orthonormalize the $\boldsymbol{\eta}$ (or H_1 Galerkin projection) given by the constraint in Equation (A.1). However the key difference lies in the MV product in step 2(d) for each algorithm. ORTHOMIN uses the actaul residual computed in each step to update the MV product which is the true representative of the error in the system, which is not the case in ORTHODIR. This can be observed in the adaptive tolerance studies for ORTHODIR, where the computed residual in the system reduces whereas the actual error blows up.

On the other hand, GMRES orthonormalizes the search directions using L_2 Galerkin projections. Moreover, all the minimization steps are computed with respect to the initial residual of system contrary to ORTHODIR or ORTHOMIN.

Thus one can say that the presented iterative schemes vary on two key aspects, the orthogonalization process and solving the minimization problem, both steps are closely linked to each other.

Appendix B

Recycling algorithms

B.1 Initial Krylov Recycling Strategy (IKRS)

The ORTHOMIN algorithm outlined in Section A.3 consists of storing the search directions $\mathbf{V}_k$, and the span of the projected vectors $\mathbf{N}_k$. The size of these matrices is $\mathrm{N_I} \times k$, where k is the number of iterations required for convergence for the present calculation. The IKRS is applied in conjunction to ORTHOMIN in the following way.

1. Initialization: Given the same initial guess $\tilde{\boldsymbol{\lambda}}_{\text{initial}}$ from the previous evaluation and the new RHS, $\mathbf{d}_{\text{new}}$, compute:

   - the residual at the initial step $\mathbf{r}_{\text{initial}} = \mathbf{d}_{\text{new}} - \mathbf{F}_{\mathrm{I}}\tilde{\boldsymbol{\lambda}}_{\text{initial}}$.
   - Update the residual and solution from all previously stored iterations

     $$\boldsymbol{\alpha}_k = \mathbf{N}_k^* \, \mathbf{r}_{\text{initial}}.$$

   - the optimum starting solution:

     $$\tilde{\boldsymbol{\lambda}}_{\text{opt}} = \tilde{\boldsymbol{\lambda}}_{\text{initial}} + \mathbf{V}_k\boldsymbol{\alpha}_k.$$

   - the optimum residual:

     $$\mathbf{r}_k^{\text{opt}} = \mathbf{r}_{\text{initial}} - \mathbf{N_k}\boldsymbol{\alpha}_k.$$

   Once the optimal residual and guess are generated from the all the stored previous basis, the ORTHOMIN algorithm can be restarted using the residual $\mathbf{r}_k^{\text{opt}}$, $\tilde{\boldsymbol{\lambda}}_{\text{opt}}$ as outlined in Section A.3. The only difference lies in step 2 (f) where the orthogonalization has been done for all the previously stored search directions.

2. Append all the Krylov subspaces with all the stored search directions $\mathbf{V}_k$ and $\mathbf{N}_k$, to be used for next solution.

B.2 Generalized conjugate residual with inner orthgonalization and Deflated Restart - (GCRO-DR)

The GCRO-DR algorithm is presented in this part. This is taken from the Appendix in [75]. In this section, this algorithm is explained in the context of solving the interface problem of the FETI-2LM method. Let, ε_{tol} be the tolerance for the interface problem of the FETI-2LM algorithm. Let $\tilde{\boldsymbol{\lambda}}_0$ be the initial guess. For each new calculation, the interface problem, $\mathbf{F}_{\mathrm{I}}^{i}\,\tilde{\boldsymbol{\lambda}} = \mathbf{d}^{i}$, where $i = 1, 2...$ needs to be solved where both the matrix and the RHS change.

The definitions of m_r and k are recalled. Let m_r be the maximum size of the subspace that can be evaluated and k be the size of the recycling subspace. The GCRO-DR algorithm involves the following steps:

1. Compute the initial residual, $\mathbf{r}_0$ as follows:

$$\mathbf{r}_0 = \mathbf{d} - \mathbf{F}_{\mathrm{I}}\tilde{\boldsymbol{\lambda}}_0. \tag{B.1}$$

2. If the calculation has been done at least once before

   Update the residual and solution using the previously stored solution

   (a) Compute the matrix $\mathbf{F}_{\mathrm{I}}\mathbf{Y}_k$. ($\mathbf{Y}_k$ is defined in the latter part.)

   Typically for iterative schemes like ORTHODIR, GMRES or ORTHOMIN, one requires to compute the MV product for each new iteration with only one RHS. However, in the present algorithm, this step requires the computation of the MV product with 'k' number of RHS. This improves the parallel efficiency significantly since the data exchange between the subdomains is done only once for k RHS. Although the computational effort for this is considerable, the operations are done in parallel.

   (b) Compute the Q-R factors of the matrix from the above step: $\mathbf{C}_k = \mathbf{Q}$, $\mathbf{U}_k = \mathbf{Y}_k\mathbf{R}^{-1}$, where $\mathbf{C}_k$ is an orthonormal basis of size $(\mathrm{N_I} \times k)$ and $\mathbf{U}_k$ of size $(\mathrm{N_I} \times k)$. The size of the matrix $\mathbf{R}$ is $(k \times k)$.

(c) Update the residual and the initial solution

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_1 &= \tilde{\boldsymbol{\lambda}}_0 + \mathbf{U}_k \mathbf{C}_k^* \mathbf{r}_0, \\ \mathbf{r}_1 &= \mathbf{r}_0 - \mathbf{C}_k \mathbf{C}_k^* \mathbf{r}_0. \end{aligned} \tag{B.2}$$

else for the first calculation

Compute the first basis and the normalized residual error

$\mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|, \quad \mathbf{c} = \|\mathbf{r}_0\|\mathbf{e}_1.$

(d) Solve the GMRES algorithm (see for instance Section A.2) for a maximum of m_r steps or if the system converges beforehand for the set tolerance, ε_{tol}.

Store the matrix $\mathbf{H}_{m_r}$ and the orthonormalized search directions $\mathbf{V}_{m_r+1}$.

(e) Compute the updated solution and the residual

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_1 &= \tilde{\boldsymbol{\lambda}}_0 + \mathbf{V}_{m_r} \mathbf{y}_{m_r}, \\ \mathbf{r}_1 &= \mathbf{V}_{m_r+1}(\mathbf{c} - \mathbf{H}_{m_r} \mathbf{y}_{m_r}), \end{aligned} \tag{B.3}$$

where $\mathbf{y}_{m_r}$ corresponds to the coefficients of the minimization problem solved in the GMRES algorithm (see for instance Equation (A.9)).

(f) Selection of the most influencing eigenvalues is done solving the following the generalized eigenvalue problem as follows:

$$(\mathbf{H}_{m_r} + h_{m_r+1,m}^2 (\mathbf{H}_{m_r}^{-1})^*)\mathbf{z} = \theta \mathbf{z}, \tag{B.4}$$

where $h_{m_r+1,m}^2$ corresponds to (m_r+1, m) entry in the Hessenberg matrix. Compute the k eigenvectors associated with the smallest magnitude of θ. Note that this step requires the inversion of the conjugate of the Hessenberg matrix. The corresponding matrix consisting of these vectors is $\mathbf{P}_k$ of size $(m_r \times k)$.

Ultimately, one seeks to get the basis vectors in the Krylov subspace associated with the interface matrix $\mathbf{F}_{\mathrm{I}}$ which is computed using the following relation,

$$\mathbf{Y}_k = \mathbf{V}_{m_r} \mathbf{P}_k, \tag{B.5}$$

where $\mathbf{Y}_k$ is matrix of size $(\mathrm{N_I} \times k)$. However this new basis is not orthonormal.

Finally a Q-R factorization of $\mathbf{Y}_k$ is done to get an orthonormal basis .

(g) The corresponding matrices $\mathbf{C}_k$ and $\mathbf{U}_k$ can be formed using these Q,R factors. $\mathbf{C}_k = \mathbf{V}_{m_r+1}\mathbf{Q}, \quad \mathbf{U}_k = \mathbf{Y}_k \mathbf{R}^{-1}$.

end

Set counter i =1,

3. while $\|\mathbf{r}_i/\mathbf{d}\| > \varepsilon_{tol}$,

4. Perform m_r - k GMRES iterations with the linear operator $\mathbf{I}-\mathbf{C}_k\mathbf{C}_k^*\mathbf{F}_\mathrm{I}$. It involves the following steps:

   - The first vector is computed with the new residual, $\mathbf{v}_1 = \mathbf{r}_i/\|\mathbf{r}_i\|$
   - The Krylov subspace $\mathbf{V}_{m_r-k+1}$, Hessenberg matrix $\mathbf{H}_{m_r-k}$ are generated preserving the orthogonality ($\mathbf{V}_{m_r-k+1} \perp \mathbf{C}_k$) using the following relation:

$$(\mathbf{I} - \mathbf{C}_k\mathbf{C}_k^*)\, \mathbf{F}_\mathrm{I}\mathbf{V}_{m_r-k} = \mathbf{V}_{m_r-k+1}\mathbf{H}_{m_r-k} \tag{B.6}$$

   This step is very similar to the step Equation (A.6) in the GMRES algorithm (for more details see for instance [75]).

   In addition to this, one needs to ensure that the newly generated vectors are orthogonal to the previously stored vectors. The resulting matrix is called $\mathbf{B}_k$ of size ($\mathrm{N_I} \times k$), where $\mathbf{B}_k = \mathbf{C}_k^*\, \mathbf{F}_\mathrm{I}\, \mathbf{V}_{m_r-k}$.

   The complete Krylov subspace can be defined using the following relation:

$$\mathbf{F}_\mathrm{I}\tilde{\mathbf{V}}_{m_r} = \tilde{W}_{m_r+1}\, \mathbf{G}_m, \tag{B.7}$$

   where $\tilde{\mathbf{V}}_{m_r} = [\mathbf{U}_k \;\; \tilde{\mathbf{V}}_{m_r-k}]$, $\tilde{W}_{m_r+1} = [\mathbf{C}_k \;\; \tilde{\mathbf{V}}_{m_r-k+1}]$, $\mathbf{G}_{m_r} = \begin{pmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{0} & \mathbf{H}_{m_r-k} \end{pmatrix}$. Typically, the matrix $\mathbf{G}_m$ is illconditioned. The conditioning can be improved by ensuring that the columns of $\mathbf{U}_k$ have unit norm.

$$\tilde{\mathbf{U}_k} = \mathbf{U}_k\mathbf{D}_k. \tag{B.8}$$

   where $\tilde{\mathbf{V}}_{m_r} = [\tilde{\mathbf{U}_k} \;\; \tilde{\mathbf{V}}_{m_r-k}]$, $\tilde{W}_{m_r+1} = [\mathbf{C}_k \;\; \tilde{\mathbf{V}}_{m_r-k+1}]$, $\mathbf{G}_{m_r} = \begin{pmatrix} \mathbf{D}_k & \mathbf{B}_k \\ \mathbf{0} & \mathbf{H}_{m_r-k} \end{pmatrix}$.

5. The minimization problem is solved to compute $\mathbf{y}_{m_r}$ using the following relation, $\|\mathbf{W}_{m_r+1}^*\mathbf{r}_i - \mathbf{G}_{m_r}\mathbf{y}_{m_r}\|$.

6. The solution and residual are updated to check if the desired tolerance limits are reached.

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_{i+1} &= \tilde{\boldsymbol{\lambda}}_i + \tilde{\mathbf{V}}_{m_r}\mathbf{y}_{m_r}, \\ \mathbf{r}_{i+1} &= \mathbf{r}_i - \mathbf{W}_{m_r+1}\mathbf{G}_{m_r}\mathbf{y}_{m_r}. \end{aligned} \tag{B.9}$$

7. If the solution has not converged, one needs to update the stored recycled subspace with new Ritz harmonic values. This is done as follows:

   - Compute k eigenvectors for the eigenvalue problem, $\mathbf{G}_{m_r}^{*}\mathbf{G}_{m_r}\mathbf{z}_i = \theta\ \mathbf{G}_{m_r}^{*}\mathbf{W}_{m_r+1}^{*}\tilde{\mathbf{V}}_{m_r}\mathbf{z}_i$. The corresponding matrix vectors are stored in $\mathbf{P}_k$. This is very similar to the steps in Equation (B.4) $\mathbf{Y}_k = \tilde{\mathbf{V}}_{m_r}\mathbf{P}_k$.
   - The corresponding matrices $\mathbf{C}_k$ and $\mathbf{U}_k$ can be formed using the Q-R factorization $\mathbf{C}_k = \tilde{\mathbf{V}}_{m_r+1}\,\mathbf{Q}, \quad \mathbf{U}_k = \mathbf{Y}_k\mathbf{R}^{-1}$.
     end
   - Let $\mathbf{Y}_k = \mathbf{U}_k, \quad \tilde{\boldsymbol{\lambda}}_0 = \tilde{\boldsymbol{\lambda}}_i$. (for the next system)

   Update iteration counter to $i + 1$.

Bibliography

[1] " MATLAB Function Reference, R2020a" MathWorks, Technical documentation. 2020.

[2] I. M. Achunche. "Acoustic optimisation and prediction of sound propagation in turbofan engine ducts". Technical report, 2010.

[3] M. Ainsworth. "Discrete dispersion relation for hp-version finite element approximation at high wave number". In: *SIAM Journal on Numerical Analysis* 42.2 (2004), pp. 553–575.

[4] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. "A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling". In: *SIAM Journal on Matrix Analysis and Applications* 23.1 (2001), pp. 15–41.

[5] X. Antoine, H. Barucq, and A. Bendali. "Bayliss-Turkel-like Radiation Conditions on Surfaces of Arbitrary Shape". In: *Journal of Mathematical Analysis and Applications* 229.1 (1999), pp. 184–211.

[6] I. Babuska and M. Suri. "*p* and *h-p* versions of the finite element method, basic principles and properties". In: *SIAM Review* 36.4 (1994), pp. 578–632.

[7] J. D. Benamou and B. Després. "A domain decomposition method for the Helmholtz equation and related optimal control problems". In: *Journal of Computational Physics* 136.1 (Sept. 1997), pp. 68–82.

[8] M. Berger, P. Pansu, J.-P. Berry, and X. Saint-Raymond. "Affine spaces". In: *Problems in Geometry*. Springer, 1984, pp. 11–17.

[9] H. Bériot, G. Gabard, and E. Perrey-Debain. "Analysis of high-order finite elements for convected wave propagation". In: *International Journal for Numerical Methods in Engineering* 96.11 (Dec. 2013), pp. 665–688.

[10] H. Bériot, A. Prinn, and G. Gabard. "Efficient implementation of high-order finite elements for Helmholtz problems". In: *International Journal for Numerical Methods in Engineering* 106.3 (Apr. 2016), pp. 213–240.

[11] A. Bermúdez, L. Hervella-Nieto, A. Prieto, R. Rodrı, et al. "An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems". In: *Journal of Computational Physics* 223.2 (2007), pp. 469–488.

[12] P. J. Besl and N. D. McKay. "Method for registration of 3-D shapes". In: *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. International Society for Optics and Photonics. 1992, pp. 586–606.

[13] M. A. Biot. "Theory of propagation of elastic waves in a fluid-saturated porous solid. II. Higher frequency range". In: *The Journal of the Acoustical Society of America* 28.2 (1956), pp. 179–191.

[14] X.-C. Cai, C. Farhat, and M. Sarkis. "A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations". In: *Contemp. Math.* 218 (1998), pp. 479–485.

[15] L. Cao, Q. Fu, Y. Si, B. Ding, and J. Yu. "Porous materials for sound absorption". In: *Composites Communications* 10 (2018), pp. 25–35.

[16] L. M. Carvalho, S. Gratton, R. Lago, and X. Vasseur. "A flexible generalized conjugate residual method with inner orthogonalization and deflated restarting". In: *SIAM Journal on Matrix Analysis and Applications* 32.4 (2011), pp. 1212–1235.

[17] O. Cessenat and B. Despres. "Application of an ultra weak variational formulation of elliptic pdes to the two-dimensional Helmholtz problem". In: *SIAM Journal on Numerical Analysis* 35.1 (1998), pp. 255–299.

[18] O. Cessenat and B. Després. "Using plane waves as base functions for solving time harmonic equations with the Ultra Weak Variational Formulation". In: *Journal of Computational Acoustics*. Vol. 11. 2. June 2003, pp. 227–238.

[19] F. Collino, S. Ghanemi, and P. Joly. "Domain decomposition method for harmonic wave propagation: A general presentation". In: *Computer Methods in Applied Mechanics and Engineering* 184.2-4 (Apr. 2000), pp. 171–211.

[20] D. Copiello and P. Ferrante. "Multi-objective optimization of "True" zero-splice liners for aero-engine intakes". In: *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference)*. 2009.

[21] A. De La Bourdonnaye, C. Farhat, A. Macedo, and F. Magoulès. "A non Overlapping Domain Decomposition Method for the Exterior Helmholtz Problem", Technical report, 1997. INRIA.

[22] M. E. Delany and E. N. Bazley. "Acoustical properties of fibrous absorbent materials". In: *Applied Acoustics* 3.2 (Apr. 1970), pp. 105–116.

[23] B. Despres. "Domain decomposition method and the Helmholtz problems". In: *Mathematical and Numerical Aspects of Wave Propagation* (1991), pp. 44–52.

[24] V. Dolean, M. J. Gander, E. Veneros, and H. Zhang. "Optimized Schwarz methods for heterogeneous Helmholtz and Maxwell's equations". In: *Lecture Notes in Computational Science and Engineering* 116 (2017), pp. 145–152.

[25] V. Dolean, M. J. Gander, and E. Veneros. "Asymptotic analysis of optimized Schwarz methods for maxwell's equations with discontinuous coefficients". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 52.6 (2018), pp. 2457–2477.

[26] V. Dolean, P. Jolivet, and F. Nataf. "An Introduction to Domain Decomposition Methods". 2015.

[27] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld. "The optimized Schwarz method with a coarse grid correction". In: *SIAM Journal on Scientific Computing* 34.1 (2012), A421–A458.

[28] I. S. Duff, A. M. Erisman, and J. K. Reid. " Direct Methods for Sparse Matrices ". Oxford University Press, Apr. 2017.

[29] O. G. Ernst and M. J. Gander. "Why it is difficult to solve Helmholtz problems with classical iterative methods". In: *Numerical analysis of multiscale problems*. Springer, 2012, pp. 325–363.

[30] C. Farhat, P. Avery, R. Tezaur, and J. Li. "FETI-DPH: A dual-primal domain decomposition method for acoustic scattering". In: *Journal of Computational Acoustics* 13.3 (Sept. 2005), pp. 499–524.

[31] C. Farhat, P.-S. Chen, J. Mandel, and F. X. Roux. "The two-level FETI method Part II: Extension to shell problems, parallel implementation and performance results". In: *Computer methods in applied mechanics and engineering* 155.1-2 (1998), pp. 153–179.

[32] C. Farhat, L. Crivelli, and F. X. Roux. "Extending substructure based iterative solvers to multiple load and repeated analyses". In: *Computer Methods in Applied Mechanics and Engineering* 117.1-2 (July 1994), pp. 195–209.

[33] C. Farhat, A. Macedo, M. Lesoinne, F. X. Roux, F. Magoulès, and A. De La Bourdonnaie. "Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems". In: *Computer Methods in Applied Mechanics and Engineering* 184.2-4 (2000), pp. 213–239.

[34] C. Farhat, A. Macedo, and R. Tezaur. "FETI-H: A scalable domain decomposition method for high frequency exterior Helmholtz problems". In: *Eleventh International Conference on Domain Decomposition Method* (1999), pp. 231–241.

[35] C. Farhat, R. Tezaur, and J. Toivanen. "A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and Lagrange multipliers". In: *International Journal for Numerical Methods in Engineering* 78.13 (June 2009), pp. 1513–1531.

[36] G. Gabard, H. Bériot, A. G. Prinn, and K. Kucukcoskun. "Adaptive, high-order finite-element method for convected acoustics". In: *AIAA Journal*. Vol. 56. 8. American Institute of Aeronautics and Astronautics Inc., 2018, pp. 3179–3191.

[37] G. Gabard, P. Gamallo, and T. Huttunen. "A comparison of wave-based discontinuous Galerkin, ultra-weak and least-square methods for wave problems". In: *International Journal for Numerical Methods in Engineering* 85.3 (Jan. 2011), pp. 380–402.

[38] G. Gabard. "Discontinuous Galerkin methods with plane waves for time-harmonic problems". In: *Journal of Computational Physics* 225.2 (2007), pp. 1961–1984.

[39] M. J. Gander, L. Halpern, and F. Magoulès. "An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation". In: *International Journal for Numerical Methods in Fluids* 55.2 (2007), pp. 163–175.

[40] M. J. Gander, F. Magoulès, and F. Nataf. "Optimized Schwarz methods without overlap for the Helmholtz equation". In: *SIAM Journal on Scientific Computing* 24.1 (2003), pp. 38–60.

[41] M. J. Gander, L. Halpern, and K. S. Repiquet. "Discontinuous coarse spaces for DD-methods with discontinuous iterates". In: *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014, pp. 607–615.

[42] M. J. Gander and F. Kwok. "Best Robin parameters for optimized Schwarz methods at cross points". In: *SIAM Journal on Scientific Computing* 34.4 (2012), A1849–A1879.

[43] M. J. Gander and K. Santugini. "Cross-points in domain decomposition methods with a finite element discretization". In: *Electronic Transactions on Numerical Analysis* 45 (2016), pp. 219–240.

[44] M. J. Gander and H. Zhang. "A class of iterative solvers for the Helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods". In: *Siam Review* 61.1 (2019), pp. 3–76.

[45] M. J. Gander. "Schwarz methods over the course of time". In: *Electronic transactions on numerical analysis* 31 (2008), pp. 228–255.

[46] C. Geuzaine and J.-F. Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities". In: *International journal for numerical methods in engineering* 79.11 (2009), pp. 1309–1331.

[47] T. Hagstrom, R. P. Tewarson, and A. Jazcilevich. "Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems". In: *Applied Mathematics Letters* 1.3 (Jan. 1988), pp. 299–302.

[48] T. Huttunen, P. Monk, and J. P. Kaipio. "Computational aspects of the ultra-weak variational formulation". In: *Journal of Computational Physics* 182.1 (2002), pp. 27–46.

[49] F. Ihlenburg and I. Babuška. "Finite element solution of the helmholtz equation with high wave number part II: the *h-p* version of the fem". In: *SIAM Journal on Numerical Analysis* 34.1 (1997), pp. 315–358.

[50] D. L. Johnson, J. Koplik, and R. Dashen. "Theory of dynamic permeability and tortuosity in fluid-saturated porous media". In: *Journal of fluid mechanics* 176 (1987), pp. 379–402.

[51] P. Jolivet and P.-H. Tournier. "Block iterative methods and recycling for improved scalability of linear solvers". In: *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2016, pp. 190–203.

[52] G. Karypis and V. Kumar. "A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices". In: (1998).

[53] A. Kempton. "Acoustic liners for modern aero-engines ". Technical Report. 2011.

[54] A. de La Bourdonnaye, C. Farhat, A. Macedo, F. Magoules, F.-X. Roux, et al. "A Non-Overlapping Domain Decomposition Method for the Exterior Helmholtz Problem". In: *Contemporary Mathematics* 218 (1998), pp. 42–66.

[55] L. Lafronza, A. McAlpine, A. Keane, and J. Astley. "Computer-Aided Liner Optimization for BroadBand Noise". In: *10th AIAA/ CEAS Aeroacoustics Conference*. American Institute of Aeronautics and Astronautics, May 2004.

[56] M. C. Leistner, P. Gosselet, and D. J. Rixen. "Recycling of solution spaces in multipreconditioned FETI methods applied to structural dynamics". In: *International Journal for Numerical Methods in Engineering* 116.2 (2018), pp. 141–160.

[57] J. Li and X. Tu. "Convergence analysis of a balancing domain decomposition method for solving a class of indefinite linear systems". In: *Numerical Linear Algebra with Applications* 16.9 (Sept. 2009), pp. 745–773.

[58] A. Lieu. "High-accuracy methods for frequency-domain flow acoustics". PhD thesis. University of Southampton, Oct. 2019.

[59] A. Lieu, H. Bériot, G. Gabard, and F.-X. Roux. "Performance of a domain decomposition method for high-order FEM". In: *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2016*.

[60] A. Lieu, G. Gabard, and H. Bériot. "A Domain Decomposition Method with High-Order Finite Elements for Flow Acoustics". In: 0.May (2019), pp. 1–12.

[61] A. Lieu, G. Gabard, and H. Bériot. "A comparison of high-order polynomial and wave-based methods for Helmholtz problems". In: *Journal of Computational Physics* 321 (2016), pp. 105–125.

[62] P. L. Lions. "On the Schwarz alternating method. I". In: *First International Symposium on domain decomposition methods for partial differential equations* (1988), pp. 1–42.

[63] P. L. Lions. "On the Schwarz alternating method. II". In: *Second International Symposium on domain decomposition methods for partial differential equations* (1989), pp. 47–70.

[64] P. L. Lions. "On the Schwarz alternating method. III". In: *Third International Symposium on domain decomposition methods for partial differential equations* (1990), pp. 202–223.

[65] Y. Maday and F. Magoulès. "Optimized Schwarz methods without overlap for highly heterogeneous media". In: *Computer Methods in Applied Mechanics and Engineering* 196.8 (Jan. 2007), pp. 1541–1553.

[66] F. Magoulès, P. Iványi, and B. H. Topping. "Convergence analysis of Schwarz methods without overlap for the Helmholtz equation". In: *Computers and Structures* 82.22 (Sept. 2004), pp. 1835–1847.

[67] F. Magoulès, P. Iványi, and B. H. Topping. "Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation". In: *Computer Methods in Applied Mechanics and Engineering* 193.45-47 (Nov. 2004), pp. 4797–4818.

[68] " MATLAB Optimization Toolbox ". The MathWorks, Natick, MA, USA. 2018.

[69] Y. Miki. "Acoustical properties of porous materials-Modifications of Delany-Bazley models-". In: *Journal of the Acoustical Society of Japan (E)* 11.1 (1990), pp. 19–24.

[70] P. Monk and D.-Q. Wang. "A least-squares method for the Helmholtz equation". In: *Computer Methods in Applied Mechanics and Engineering* 175.1-2 (1999), pp. 121–136.

[71] R. Motsinger and R. Kraft. "Design and performance of duct acoustic treatment". In: (1991).

[72] P. Mustafi. "Improved turbofan intake liner design and optimization". PhD thesis. University of Southampton, 2013.

[73] M. Ogino, K. Yodo, R. Shioya, and H. Kawai. "Two-level extension of the hierarchical domain decomposition method". In: *Mechanical Engineering Letters* 4.0 (2018), pp. 18–00088.

[74] D. Oliva and V. Hongisto. "Sound absorption of porous materials - Accuracy of prediction methods". In: *Applied Acoustics* 74.12 (Dec. 2013), pp. 1473–1479.

[75] M. L. Parks, E. De Sturler, G. Mackey, D. D. Johnson, and S. Maiti. "Recycling Krylov subspaces for sequences of linear systems". In: *SIAM Journal on Scientific Computing* 28.5 (2006), pp. 1651–1674.

[76] S. Petersen, D. Dreyer, and O. von Estorff. "Assessment of finite and spectral element shape functions for efficient iterative simulations of interior acoustics". In: *Computer Methods in Applied Mechanics and Engineering* 195.44-47 (Sept. 2006), pp. 6463–6478.

[77] A. Prinn. "Efficient finite element methods for aircraft engine noise prediction". PhD thesis. University of Southampton, 2014.

[78] S. Ren, F. Xin, T. J. Lu, and C. Zhang. "A semi-analytical model for the influence of temperature on sound propagation in sintered metal fiber materials". In: *Materials and Design* 134 (Nov. 2017), pp. 513–522.

[79] C. Rey and F. Risler. "A Rayleigh–Ritz preconditioner for the iterative solution to large scale nonlinear problems". In: *Numerical Algorithms* 17.3-4 (1998), pp. 279–311.

[80] F. Risler and C. Rey. "Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems". In: *Numerical algorithms* 23.1 (2000), p. 1.

[81] A. RJ, R. Sugimoto, I. Achunche, M. Kewin, P. Mustafi, and E. Deane. "Reprint of: A review of CAA for fan duct propagation and radiation, with application to liner optimisation". In: vol. 1. Elsevier, 2010, pp. 143–152.

[82] F.-X. Roux. "Parallel implementation of a domain decomposition method for non-linear elasticity problems". In: *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*. SIAM, 1995, pp. 161–175.

[83] F.-X. Roux and A. Barka. "Block krylov recycling algorithms for FETI-2LM applied to 3-D electromagnetic wave scattering and radiation". In: *IEEE Transactions on Antennas and Propagation* 65.4 (2017), pp. 1886–1895.

[84] Y. Saad. "Iterative methods for sparse linear systems". 2003.

[85] Y. Saad and M. H. Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (July 1986), pp. 856–869.

[86] H. A. Schwarz. "Ueber einen Grenzübergang durch alternirendes Verfahren". In: *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zurich* 15 (1870), pp. 272–286.

[87] P. Solin, K. Segeth, and I. Dolezel. "Higher-order finite element methods". CRC Press, 2003.

[88] C. K. Tam and J. C. Webb. "Dispersion-relation-preserving finite difference schemes for computational acoustics". In: *Journal of Computational Physics* 107.2 (1993), pp. 262–281.

[89] K. W. Thompson. "Time dependent boundary conditions for hyperbolic systems". In: *Journal of Computational Physics* 68.1 (1987), pp. 1–24.

[90] L. L. Thompson. "A review of finite-element methods for time-harmonic acoustics". In: *The Journal of the Acoustical Society of America* (2006).

[91] J. Toivanen, P. Avery, and C. Farhat. "A multilevel FETI-DP method and its performance for problems with billions of degrees of freedom". In: *International Journal for Numerical Methods in Engineering* 116.10-11 (2018), pp. 661–682.

[92] O. Widlund and M. Dryja. "An additive variant of the Schwarz alternating method for the case of many subregions", Technical report, 1987. Department of Computer Science, Courant Institute.

[93] R. Wright and M. Kidner. "Vibration absorbers: a review of applications in interior noise control of propeller aircraft". In: *Journal of Vibration and Control* 10.8 (2004), pp. 1221–1237.

[94] T. Zandbergen. "Do Locally Reacting Acoustic Liners Always Behave as They Should?" In: *AIAA Journal* 18.4 (Apr. 1980), pp. 396–397.