ABSTRACT

BHATTACHARJEE, ABHINAB. Interface Conditions for Domain Decomposition Methods with Non-Conforming Grids. (Under the direction of Dr. Murthy N. Guddati.)

With an aim of relaxing the meshing cost of a global domain in a discrete setting, we mesh the local subdomains according to our needs after decomposing the global domain. This reduces the meshing cost but often gives rise to the problem of mismatched grids at the interfaces. To handle this incompatibility at the interface, we incorporate three important concepts of Domain Decomposition, rational approximation of the boundary conditions and a "weak" information transfer between non-conforming meshes through a common non-overlapping interface.

This thesis solves the 2D Laplace equation in two or multiple domains for static load in anti-plane shear setting using (a) Alternating Schwarz method (b) with data transfer across non conforming meshes using an orthogonal projection approach and (c) approximating the boundary stiffness using springs and Perfectly Matched Discrete Layers (PMDLs), which were originally developed for unbounded modeling of continuous wave propagation. After careful verification of the method in unidirectional split, this algorithm was tested for two dimensional decomposition.

Numerical examples are presented in 1-dimensional as well as 2-dimensional settings. The only difference in multiple domains is the use of sequential forward-backward sweeping (for 1D split) and a staggered sweeping (for 2D split) instead of alternating methodology. The results show a good convergence on the interface which ensures proper interfacial transfer of information between the mismatched grids and thus the overall accuracy of the final solution.

Interface Conditions for Domain Decomposition Methods with Non-Conforming Grids

by
Abhinab Bhattacharjee

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Civil Engineering

Raleigh, North Carolina

2020

APPROVED BY:

Dr. Gnanamanikam Mahinthakumar	Dr. Shamim Rahman

Dr. Murthy N. Guddati
Chair of Advisory Committee

DEDICATION

To my parents,family and friends.

BIOGRAPHY

Abhinab, son of Mr. Ajoy Kumar Bhattacharjee and Mrs Kabita Bhattacharjee was born in the state of Jharkand(India) and grew up in the steel city of Durgapur, West Bengal where he completed his high school. He graduated from National Institute of Technology, Durgapur in 2015 with a major in Civil Engineering. In 2017, he moved to North Carolina State University for his masters in Structural Engineering with a focus on computational mechanics. This thesis is a result of the research conducted during this time.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Guddati, for giving me an opportunity to pursue research in his group. It has been an enlightening, challenging as well as an exciting journey. I would be indebted to him for all his invaluable advises, financial support, brainstorming sessions, relevant courses and most importantly, his stoic patience with me.

I would like to thank Dr. Kumar and Dr. Rahman for being in my advising committee.

I would like to thank all my friends and colleagues in Raleigh without whom I would not have survived my masters.

I would also like to thank my teachers in both elementary and high school, as they have shaped me and my way of contemplation and perception.

Finally I would like to thank my family for their unconditional support and love.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Unprecedented growth in computing power has enabled us to model, simulate and consequently study complex physical problems. There may be cases where we need more accurate solutions around a local region in a global domain. Such cases include, but not limited to, molecular dynamics (MD), crack propagation, multi-scale and multi-physics problem to name a few. Solving the entire global domain (in a discrete setting) with the same degree of discretization may lead to high meshing cost. To reduce the meshing and subsequently the overall computational cost, we decompose the global domain and discretize the subdomains according to our needs. This often gives rise to mismatched or non-conforming meshes at the interfaces of the subdomains. Transfer of information in the case of matched meshes is trivial. In this study, with an aim to efficiently transfer information between domains which have non-conforming meshes, we bring together three concepts which are explained next. Besides reducing the meshing cost, it facilitates the use of different solvers in different subdomains, if required. This gives us more flexibility and an obvious advantage over a fully coupled simulations, which is often plagued by high

computational cost. The proposed methodology brings together existing ideas related to (a) Optimized Schwarz Method (b) rational approximation of interfacial stiffness and (c) orthogonal projection of interfacial data. These are elaborated in the remainder of the chapter.

1.1 Optimized Schwarz Method

The global domain of interest is decomposed into smaller subdomains and each subdomain is solved separately, either in parallel or sequentially, reducing the computational expense of solving the entire problem together. This is the basic idea of the Domain Decomposition Methods (DD). The interfacial data transfer is modeled with Dirichlet, Neumann or Robin data transfer. The subdomains can be completely non-overlapping or can have small overlap, which is included to help with convergence. This study focuses on non-overlapping subdomains where the interfaces do not overlap. There is vast literature on domain decomposition methods and the reader is referred to e.g. [1] for an excellent overview. The algorithm of interest to us is what is known as Schwarz Alternating Method. It was initially proposed in 1870 by a German mathematician Herman Schwarz to iteratively solve the Poison's problem in the subdomains by transferring the Dirichlet data across the interface. The domains are overlapping as shown in the Fig. 1.1

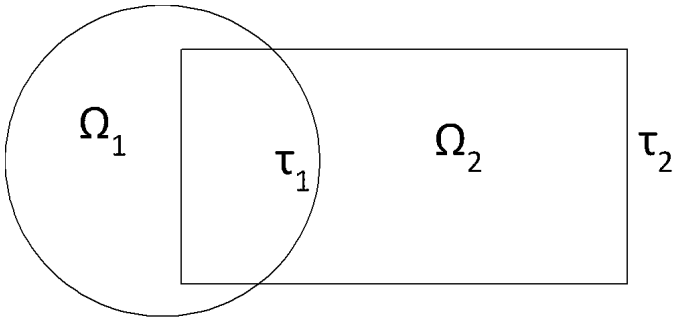


Figure 1.1 Overlapping Schwarz Method

The equation that was being solved is represented by following equations to find u : $\Omega \rightarrow$ R

$$-\Delta\left(u_1^{2n}\right) = f_1 \text{ in } \Omega_1 \qquad -\Delta\left(u_2^{2n+1}\right) = f_2 \text{ in } \Omega_2 \tag{1.1}$$

$$u_1^{2n} = 0 \text{ on } \partial\Omega_1 \cap \partial\Omega \qquad u_2^{2n+1} = 0 \text{ on } \partial\Omega_2 \cap \partial\Omega \tag{1.2}$$

$$u_1^{2n} = u_2^{2n-1} \text{ on } \partial\Omega_1 \cap \overline{\Omega_2} \qquad u_2^{2n+1} = u_1^{2n} \text{ on } \partial\Omega_2 \cap \overline{\Omega_1} \tag{1.3}$$

Equation (1.3) signifies the iterative methodology of the algorithm where u_1^{2n} is calculated from the previous step solution in Ω_2 that is u_2^{2n-1} and the next step solution u_2^{2n+1} is calculated from the current step in Ω_1 that is u_1^{2n}. The same equation signifies the information transfer across the interface, albeit only Dirichlet data.

We, however, do not use this classical Schwarz method in our work. There are a few reasons for doing so. Firstly, the original algorithm works only in the case of overlapping subdomains. It converges instantly if both the subdomains *completely* overlap, which is trivial, while the method does not converge if they are non overlapping. This is a restriction in cases where the need to decompose the geometry first before solving the subdomains are important/required. Some physical examples include fluid-structure interaction, Atomistic-to-Continuum (AtC) coupling to name a few. PL Lions in his paper [2] proposed to use a different kind of transmission conditions across the interface to get faster convergence. Instead of just transferring the Dirichlet data, he proposed to transfer

a combination of both Dirichlet and Neumann data across the interface. He showed that this works for non-overlapping cases, removing the disadvantage of the original algorithm. Besides, Robin data transfer can handle overlapping subdomains too.

The second motivation comes from the acoustic problems where the original algorithm does not converge for Helmholtz problems in frequency domain, even in the case of overlap. Research has showed that a Robin boundary condition has a better convergence. Since the long-term aim is to to apply the algorithm to solve the wave propagation problems, we resort to the Robin type of data transfer.

The other more important motivation comes from the fact that the classical Schwarz method has a relatively slow convergence rate and depends on the degree of overlap. Robin data transfer generally results in better convergence rate, especially when combined with the ideas of absorbing boundary condition (see e.g. [3]).

1.2 Rational Approximation of Interfacial Stiffness

Robin information transfer involves continuity across the interface of specially designed Robin data, which is a linear combination of traction (Neumann data) and displacement, where the displacement is multiplied by a stiffness-type term. It can be shown that the best possible multiplier is the stiffness of the remainder of the domain that is not represented by the current subdomain. The performance of a DD method would thus depend on the stiffness parameter used for interfacial data transfer. One idea is to use distributed springs as the stiffness parameter, while a more elaborate idea is to approximate the rest of the domain as unbounded and use the techniques developed for modeling unbounded domains.

To model unbounded domains, researchers have developed several methodologies. The domain of interest is known as the interior and the remaining unbounded domain is called the exterior. The information in the exterior is only needed at the interface. The exterior

is modeled using some mathematical approximation at the interface. One of the methods is to calculate the Green's function of the exterior and it leads to highly accurate results. The disadvantage is it is extremely expensive being global in space as well as in time. Other method that was made very famous by Berenger in[4] is the Perfectly Matched Layers which has been used extensively for wave propagation problems. Although theoretically accurate in preventing unwanted reflections at the interface, numerical implementation using a domain based method have discretization and truncation errors. Local absorbing boundary conditions are other alternatives which try to match the exact impedance of the exterior at the interface using rational or continued fraction approximation. A methodology that links PML and rational absorbing boundary conditions was proposed in [5]. It was shown that the intentionally made integration error exactly cancels the discretization error and hence it is called Perfectly Matched Discrete Layers (PMDL). This study was done for transient problems and the parameters of the PMDL were its length and the number of layers, which had exponential convergence. The PMDLs can be used for static problems in approximating the unbounded domain stiffness using a few layers of the same as was shown in [6]. The parameters are the number of PMDL layers and the layer thicknesses.

It is shown that PMDLs combine the accuracy of local ABCs and the broader applicability of the PMLs[5]. In this study both springs (simplest scalar approximation of the interfacial stiffness of the other subdomain) and PMDLs will be used to study the convergence in the context of mismatched grids.

1.3 Orthogonal Projection of interfacial data

The finite element discretization of the subdomains may not be conforming at the interfaces due to meshing discrepancy. Treating non-matching grids has been an active area of research in the field of fluid-structure interaction where a finer mesh is required to capture

the pressure from the fluid onto the solid surface. Readers are directed to e.g. [7],[8],[9] for an extensive review and [10] for a summary of existing literature. We choose an orthogonal projection method to transfer the interface information. It should be noted that our main focus is the efficient transmission conditions and not decreasing the discretization errors, which can be controlled through mesh refinement. Moreover, while DD algorithms can be used as a preconditioner for iterative (Krylov) methods, in this study, it is used as a solver where the subdomains are analyzed in an iterative, sequential manner to result in the converged solution.

1.4 Outline

The outline of rest of the thesis is as follows. Chapter 2 contains the preliminaries, with concise explanation of the PMDL with the the relevant equations (detailed derivations can be found in the appendices). Chapter 2 also contains the derivation of the Schwarz algorithm in 2D settings. Chapter 3 presents the proposed methodology, where all three components of the algorithm are summarized. Chapter 4 has numerical examples with both 1D and 2D decomposition, where a sequential, sweeping solution approach is incorporated. Chapter 5 contains the summary and conclusions.

CHAPTER 2

PRELIMINARIES

This chapter focuses on a more detailed explanation of the different concepts that are being brought together in the proposed algorithm.

1. Explanation of Perfectly Matched Discrete Layers (PMDL).

2. Application of the "robin" transmission conditions to transfer the information across the interface in the context of Laplace equation.

2.1 Perfectly Matched Discrete Layers

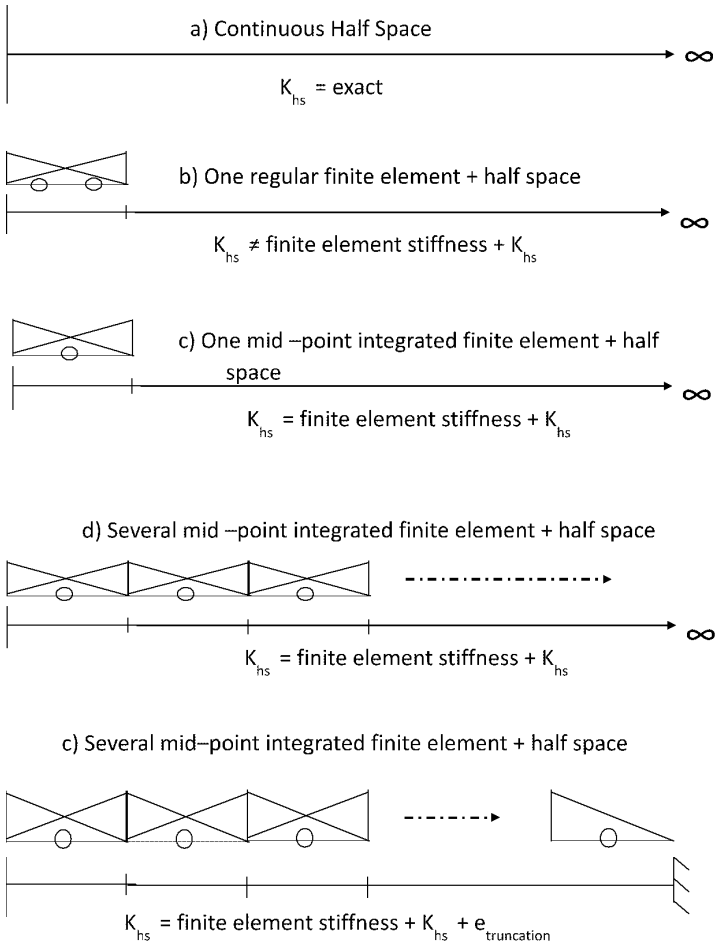


Figure 2.1 Steps in formulation of PMDL with mid-point integrated finite elements

Fig. 2.1 shows the methodology that is used for adding PMDL elements to a 1D mesh. It can be extended in a very similar manner to higher dimensions. If the original half-space is replaced by single finite element, with full integration, and another 'smaller' half-space, the exact stiffness of the original half space is lost due to discretization errors. The trick is to replace full integration with reduced integration, which gives back the exact stiffness of the original half-space [5]. The next steps is to keep adding the mid-point

integrated finite elements recursively. This will give the exact stiffness of the original half-space but will be computationally intractable due to presence of infinite number of finite elements. To make it tractable, the number of finite elements are truncated by applying a Dirichlet boundary condition on the right. This truncation introduces an error in half-space stiffness which is measured with the help of reflection coefficient, which has appropriate physical meaning for wave propagation problems, and simply a measure of error in stiffness for static problems.

$$|\mathrm{S}_n| = \prod_{j=1}^{n} \left| \frac{k - 2/L_j}{k + 2/L_j} \right|^2 \text{ in static case} \tag{2.1a}$$

$$|\mathrm{R}_n| = \prod_{j=1}^{n} \left| \frac{k - 2i/L_j}{k + 2i/L_j} \right|^2 \text{ in dynamic case} \tag{2.1b}$$

The term k, in dynamic settings, signifies the wavenumber in the direction of unboundedness. In static problem, k is generally imaginary and represents the rate of decay of displacement in the orthogonal direction. The important thing to be noted is the mid-point integration is used only in the direction perpendicular to the interface, i.e. in the direction of unboundedness. Fig. 2.2 and Fig. 2.3 show a typical example of PMDL in 2D. The error/reflection coefficient shows that for a particular length (which is real in static and complex in dynamic setting), the methodology approximates the stiffness or the so called *Dirichlet-to-Neumann* map at the interface exactly for specific values of k. For other values of k, it has been shown that a few layers of PMDL can accurately represent the unbounded domain stiffness. More detailed derivation can be found in the appendix for reference of the reader (B).

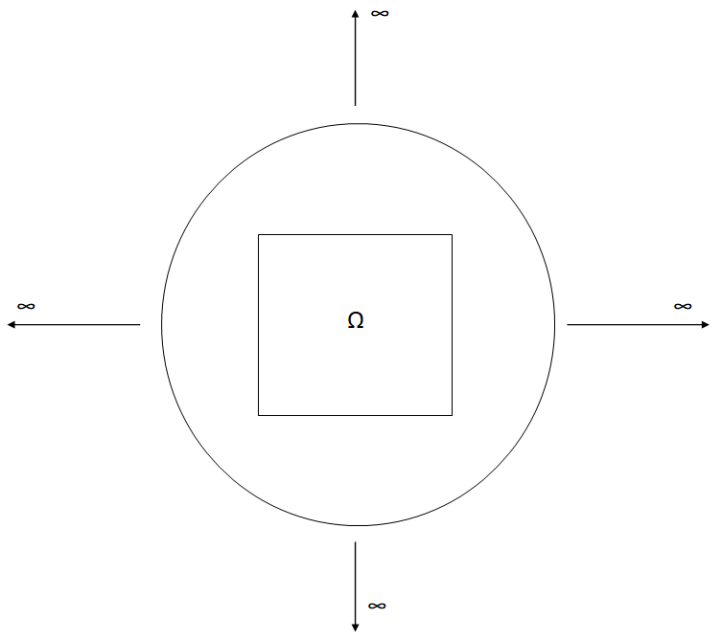


Figure 2.2 An interior subdomain unbounded in all directions

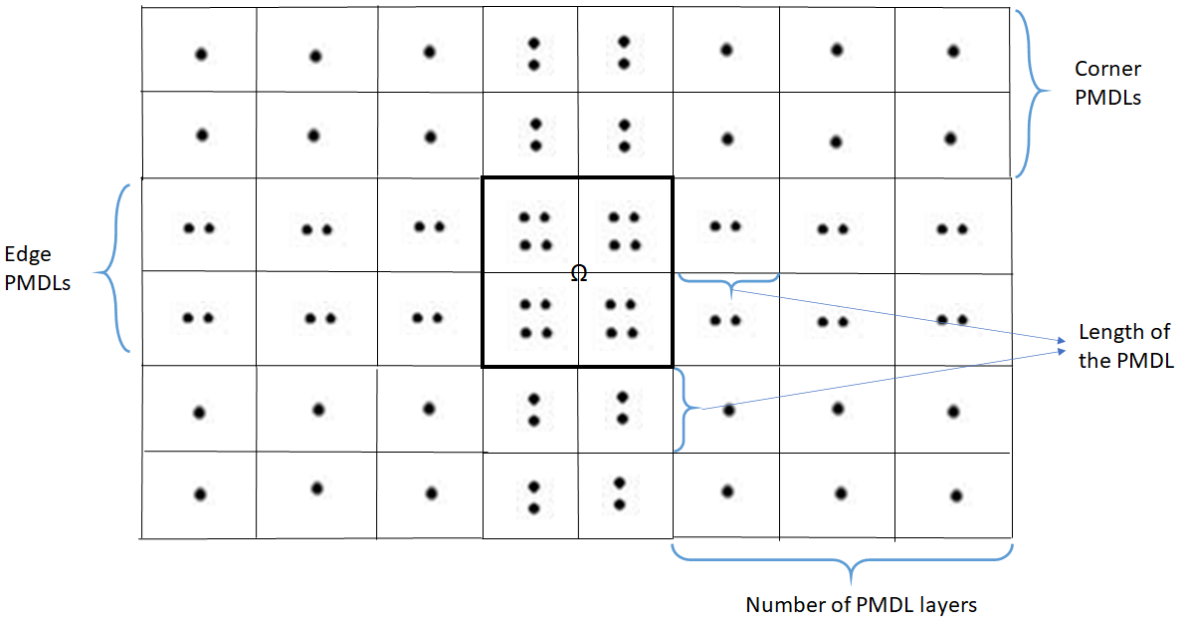


Figure 2.3 Finite Element Mesh and PMDLs with all integration points

2.2 Optimized Schwarz Method with Robin Data Transfer

Starting with the 2D scalar wave equation in time domain.

$$-\nabla\left(G\nabla u\right)+\rho\ddot{u}=f, \tag{2.2}$$

Where ρ is the density and G is the shear modulus. For simplicity in presentation, we assume G to be constant. Fourier transforming the equation in time,

$$-G\nabla^2 u+Gk^2u=f_f. \tag{2.3}$$

In (2.3), u and f_f are a function of the coordinates and frequency and $k^2=-\rho\omega^2/G$. Considering two subdomains and using subscripts 1 and 2 to denote the terms associated with each subdomains, the following set of equations are derived(Refer [11]).

$$-G\nabla^2 u_1^{2n}+Gk_1^2u_1^{2n}=f_1 \text{ in } \Omega_1 \qquad -G\nabla^2 u_2^{2n+1}+Gk_2^2u_c^{2n+1}=f_2 \text{ in } \Omega_2 \tag{2.4}$$

$$\aleph u_1^{2n}=d_1 \text{ on } \partial\Omega\cap\partial\Omega_1 \qquad \aleph u_2^{2n+1}=d_2 \text{ on } \partial\Omega\cap\partial\Omega_2 \tag{2.5}$$

$$\mathcal{B}_1u_1^{2n}=\mathcal{B}_1u_2^{2n-1} \text{ on } \tau_1 \qquad \mathcal{B}_2u_2^{2n+1}=\mathcal{B}_2u_1^{2n} \text{ on } \tau_2 \tag{2.6}$$

$$\mathcal{B}_1(\cdot)\equiv\boldsymbol{\nabla}(\cdot)\cdot\boldsymbol{n}_1+\boldsymbol{\Lambda}_1(\cdot) \qquad \mathcal{B}_2(\cdot)\equiv\boldsymbol{\nabla}(\cdot)\cdot\boldsymbol{n}_2+\boldsymbol{\Lambda}_2(\cdot) \tag{2.7}$$

In the above set of equations, $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ are the parameters of the iterative process which determines the convergence rate of the methodology. The main goal is to choose the parameters in such a way that it leads to an optimal convergence. The parameters, in this study, will be chosen to be either scalars, i.e. distributed springs (simplest way to approximate the other domain stiffness at the interface), or a rational local approximation based on PMDLs. The obvious question arises as to the choice of spring stiffness or PMDL

parameters. This can be examined using the concept of convergence factor: The rate of convergence is quite important and the convergence factor is defined as(see e.g. [11])

$$\Psi = \left| \left(\frac{\Lambda_1 - \kappa_2}{\Lambda_1 + \kappa_2} \right) \left(\frac{\Lambda_2 - \kappa_1}{\Lambda_2 + \kappa_1} \right) \right| \tag{2.8}$$

This is the factor by which the error reduces in each iteration. Here κ_1 and κ_2 are the *Dirichlet-to-Neumann* (DtN) map or stiffness of the rest of the domain at the subdomain interface. It can inferred from the convergence expression that the parameters of the methodology should be as close as possible to the DtN maps of the rest of the domain for fastest convergence. If $\Lambda_1 = \kappa_2$ or $\Lambda_2 = \kappa_1$, Ψ becomes zero and the answer converges instantly proving that the choice of parameters are optimal. However, the exact DtN map is not known for most problems, and the parameters to get a good convergence is addressed in the literature with multiple recommendations.

To see how the optimized algorithm converges, a 1D bar simulation is carried out using Robin data transfer. The geometry and subsequent decomposition is shown in Fig. 2.4 and Fig. 2.5.

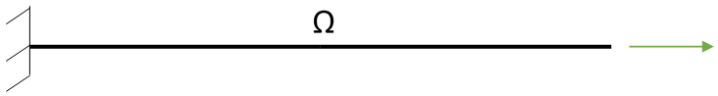


Figure 2.4 A global domain with a point load and fixed boundary condition

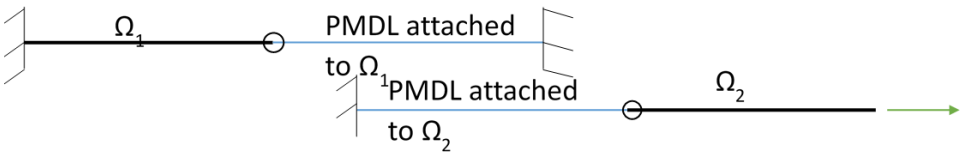


Figure 2.5 subdomains with original boundary conditions and PMDLs approximating the other domain

The governing differential equation and boundary conditions of the global problem (as shown in Fig. 2.4) are,

$$-\frac{\partial^2 u}{\partial x^2} + k^2 u = 0 \qquad x \in (0, L) \tag{2.9a}$$

$$\frac{\partial u}{\partial x} = f \qquad \text{at } x = L \tag{2.9b}$$

$$u = 0 \qquad \text{at } x = 0 \tag{2.9c}$$

The global domain is decomposed into two non-overlapping subdomains $\Omega_1 \equiv (0, L_1)$ and $\Omega_2 \equiv (L_1, L)$ as shown in the Fig. 2.5. The weak formulation is skipped here (reader can refer to [11] for more details) and the stiffness of the interior $\mathbf{K}^1$, $\mathbf{K}^2$ are known. Also $\Lambda_1 = \mathbf{P}^1$ and $\Lambda_2 = \mathbf{P}^2$, where $\mathbf{P}^1$ and $\mathbf{P}^2$ are the PMDL stiffness matrices corresponding to first and second subdomains respectively. The sub-problems that need to be solved are,

$$\begin{aligned}
&-\frac{\partial^2 u_1}{\partial x^2} + k_1^2 u_1 = 0 \quad x \in (0, L_1) && -\frac{\partial^2 u_2}{\partial x^2} + k_2^2 u_2 = 0 \quad x \in (L_1, L) \\
&u_1 = 0 \quad \text{at } x = 0 && \frac{\partial u_2}{\partial x} = f \quad \text{at } x = L \\
&\frac{\partial u_1}{\partial x} + \mathbf{\Lambda_1} u_1 = \frac{\partial u_2}{\partial x} + \mathbf{\Lambda_1} u_2 \ \text{ at } x = L_1 && -\frac{\partial u_2}{\partial x} + \mathbf{\Lambda_2} u_2 = -\frac{\partial u_1}{\partial x} + \mathbf{\Lambda_2} u_1 \ \text{ at } x = L_1
\end{aligned} \tag{2.10}$$

Equation (2.10) is discretized using standard linear finite elements of size h. After applying the boundary conditions, we obtain the following matrix equation:

$$\begin{bmatrix} \boldsymbol{K}_{ii}^1 & \boldsymbol{K}_{ib}^1 & \mathbf{0} \\ \boldsymbol{K}_{bi}^1 & K_{bb}^1 + P_{bb}^1 & \boldsymbol{P}_{be}^1 \\ \mathbf{0} & \boldsymbol{P}_{eb}^1 & \boldsymbol{P}_{ee}^1 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{1,i}^{2j} \\ u_{1,b}^{2j} \\ \boldsymbol{u}_{1,e} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{1,i} \\ 0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ f_{2,b}^{2j-1} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ P_{bb}^1 u_{2,b}^{2j-1} \\ \boldsymbol{P}_{eb}^1 u_{2,b}^{2j-1} \end{bmatrix} \tag{2.11}$$

$$\begin{bmatrix} \boldsymbol{P}^2_{ee} & \boldsymbol{P}^2_{eb} & \mathbf{0} \\ \boldsymbol{P}^2_{be} & P^2_{bb} + K^2_{bb} & \boldsymbol{K}^2_{bi} \\ \mathbf{0} & \boldsymbol{K}^2_{ib} & \boldsymbol{K}^2_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{2,e} \\ u^{2j+1}_{2,b} \\ \boldsymbol{u}^{2j+1}_{2,i} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ \boldsymbol{f}_{2,i} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ f^{2j}_{1,b} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{P}^2_{be} u^{2j}_{1,b} \\ P^2_{bb} u^{2j}_{1,b} \\ \mathbf{0} \end{bmatrix} \tag{2.12}$$

Equations (2.11) and (2.12) correspond to the first and second subdomain respectively. The subscripts on the left hand side of each equations, i.e 'i', 'b' and 'e' correspond to interior, boundary and auxiliary information. The superscript '2j', '2j-1' and '2j+1' are the iteration indices. $f_{2,b}$ is the nodal load vector corresponding to the traction at the interface on the second subdomain that is contributing to the total load acting on first subdomain. On a similar note, $f_{1,b}$ is the nodal load corresponding to the traction at the interface on the first subdomain.

The equations were implemented in MATLAB, leading to the following observations.

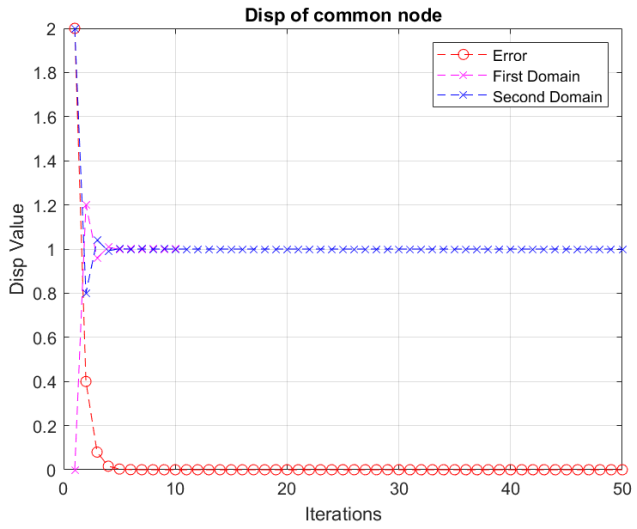


Figure 2.6 Displacement of the node at the interface

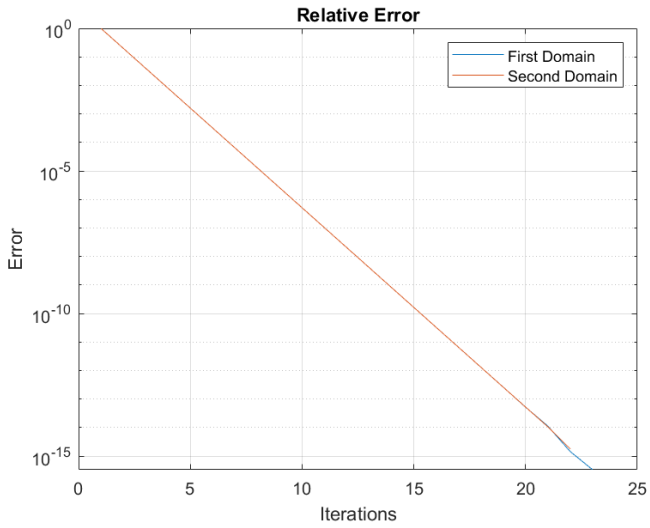


Figure 2.7 Exponential Convergence of the error at the interface

Fig. 2.6 shows the displacement of the node at the interface for both subdomains. Note that the displacement does not converge in the first step but does so iteratively to the final solution. The choice of the parameters are not optimal in this case. A slight manipulation of the parameter can lead us to the optimal choice of Λ as shown in Fig. 2.8 where displacement converges to a solution in the first two iterations. This is possible only for special cases where the exact stiffness of the other subdomain is known. Another important observation in Fig. 2.7 is that the relative error has exponential convergence for both the subdomains, which is the direct outcome of the convergence theory.

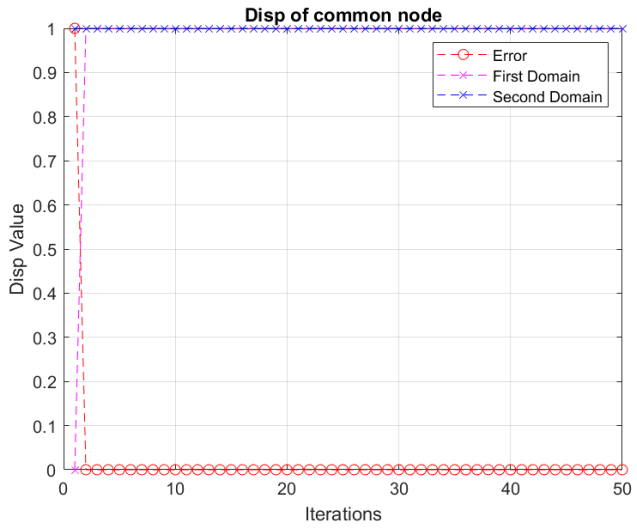


Figure 2.8 Displacement of the node at the interface for the optimal choice of parameter

2.3 Extension to Higher Dimensions: Weak Formulation

We extend the idea presented in the previous section to higher dimensions by writing the associated weak forms, to facilitate the discussion of the methodology in the next chapter. We focus on the first subdomain; the formulation for the second subdomain is similar. Multiplying (2.4) by virtual displacement δu_1 for the first subdomain and applying integration by parts results in,

$$-\int_{\tau_1} G\left(\delta u_1\right)^T \nabla u_1 d\tau_1 + \int_{\Omega_1} G\left(\nabla \delta u_1\right)^T \nabla u_1 d\Omega_1 + \int_{\Omega_1} \delta u_1^T k^2 u_1 d\Omega_1 = \int_{\Omega_1} \delta u_1^T f_1 d\Omega_1 \quad (2.13)$$

The weak form of the Robin boundary operator Φ is defined as,

$$\Phi_1 = \int_{\tau_1} \left(\delta u_1\right)^T \left(\nabla\left(\cdot\right) \cdot n_1 + \Lambda_1\left(\cdot\right)\right) d\tau_1 \quad (2.14)$$

The second term in parenthesis in the (2.14) is the stiffness (or *Dirichlet-to-Neumann*), which is applied on displacement to yield the force. This is integrated over the interface after multiplying with the virtual displacement. To conserve the work on the interface, an operator Φ_1 is defined and the displacement on both the subdomains are equated. It is important to note that the displacement coming from the second subdomain is from the previous iteration:

$$\Phi_1 u_1^n \;=\; \Phi_1 u_2^{n-1} \tag{2.15}$$

In the rest of the derivations, the superscript which denote the iteration steps would be dropped without any loss in generality. Substituting Φ_1 in the above equation,

$$\int_{\tau_1} (\delta u_1)^T (\nabla u_1 \cdot n_1)\, d\tau_1 \;+\; \int_{\tau_1} (\delta u_1)^T \Lambda_1 u_1 d\tau_1 = \int_{\tau_1} (\delta u_1)^T (\nabla u_2 \cdot n_1)\, d\tau_1 + \int_{\tau_1} (\delta u_1)^T \Lambda_1 u_2 d\tau_1 \tag{2.16}$$

Using (2.13) and (2.16) to simplify

$$G\int_{\tau_1} (\delta u_1)^T \Lambda_1 u_1 d\tau_1 \;+\; G\int_{\Omega_1} (\nabla \delta u_1)^T \nabla u_1 d\Omega \;+\; \int_{\Omega_1} (\delta u_1)^T k^2 u_1 d\Omega \;= G\int_{\tau_1} (\delta u_1)^T (\nabla u_2 \cdot n_1)\, d\tau_1 \;+\; G\int_{\tau_1} (\delta u_1)^T \Lambda_1 u_2 d\tau_1 \;+\; \int_{\Omega_1} (\delta u_1)^T f_1 d\Omega \tag{2.17}$$

We choose the standard Bubnov-Galerkin Finite element method to discretize the problem.

$$\begin{aligned} u_1 = N_1 d_1\,, \delta u_1 = N_1 \delta d_1, \\ u_2 = N_2 d_2\,, \delta u_2 = N_2 \delta d_2. \end{aligned} \tag{2.18}$$

N_1 and N_2 are the shape functions on the interfaces of the first and second subdomain respectively. In the current study, linear shape functions are used, although the extension

to the higher-order shape functions is straightforward. Substituting (2.18) in (2.17) and some manipulations results in,

$$\left[\int_{\tau_1} N_1^T G\Lambda_1 N_1 d\tau_1 \; + \int_{\Omega_1} (\nabla N_1)^T G\,(\nabla N_1)\, d\Omega \; + \int_{\Omega_1} N_1^T k^2 N_1 d\Omega\right] d_1 \; = \int_{\tau_1} N_1^T G\left\{(\nabla N_2 \cdot n_1)\, d_2\right\} d\tau_1 \; + \int_{\tau_1} N_1^T G\Lambda_1 N_2 d_2 d\tau_1 \; + \int_{\Omega_1} N_1^T f_1 d\Omega + \int_{\tau-\tau_1} N_1^T \left\{(\nabla N_1 \cdot n_1)\, d_1\right\} d\tau_1 \quad (2.19)$$

The equation (2.19) can be written in the condensed form as,

$$\boxed{K_1 \cdot D_1 = F_{1\_external} + F_{otherdomain}}. \quad (2.20)$$

where,

$$\begin{aligned} K_1 &= \text{Stiffness of the subdomain} \\ &= \int_{\tau_1} N_1^T G\Lambda_1 N_1 d\tau_1 \; + \int_{\Omega_1} (\nabla N_1)^T G\,(\nabla N_1)\, d\Omega \; + \int_{\Omega_1} N_1^T k^2 N_1 d\Omega \end{aligned} \quad (2.21)$$

$$\begin{aligned} F_{1\_external} &= \text{Body force and traction acting on the subdomain} \\ &= \int_{\Omega_1} N_1^T f_1 d\Omega \; + \int_{\tau-\tau_1} N_1^T \left\{(\nabla N_1 \cdot n_1)\, d_1\right\} d\tau_1 \end{aligned} \quad (2.22)$$

$$\begin{aligned} F_{otherdomain} &= \text{Force from the next subdomain through the mismatched interface} \\ &= \int_{\tau_1} N_1^T G\left\{(\nabla N_2 \cdot n_1)\, d_2\right\} d\tau_1 \; + \int_{\tau_1} N_1^T G\Lambda_1 N_2 d_2 d\tau_1 \end{aligned} \quad (2.23)$$

Equation (2.23) calculates the force transferred from the other subdomain corresponding to the displacements in the previous step. The first integral is the traction and the second integral is the product of the displacement from the other subdomain of the previous step and the operator (Λ) of this subdomain. This represents *Robin* data transfer in enforcing a 'weak' continuity across the subdomains.

2.4 Problem Statement

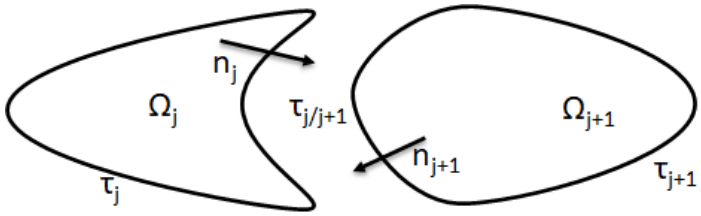


Figure 2.9 Two Non-overlapping subdomains with information transfer at the common interface

In this thesis, the 2D Laplace equation is solved iteratively in each subdomain. The global domain (Ω_{global}) is broken down into smaller subdomains (Ω_k) such that $\Omega_{global} = \cup_{k=1}^{N} \Omega_k$, where $1 \leq k \leq N$ and $\Omega_k \cap \Omega_l = \phi$, where $1 \leq k \leq l \leq N$ and $k \neq l$. The information transfer at the interface takes place through *Robin* data transfer until the solution converges on the interface as well as the interior. As mentioned, the local approximation of the stiffness of the adjacent subdomain is done with the help of springs (simplest case) and PMDLs. The integration in (2.23) is done on the interface of the first subdomain. It involves integrating two different shape functions, as the discretization on the interfaces are usually different in different subdomains. The main objective of this study is to develop an algorithm to transfer information across the interfaces by evaluating the above integral in the context of Optimized Schwarz methods with spring

and PMDL based interface conditions. Needless to say, this algorithm works in the special case of matched grids too.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Information Transfer

Since the focus of the work is interfacial information transfer, we start with the expression for the force which is coming from the other subdomain, derived in previous chapter (Equation (2.23)). Without any loss of generality, we assume $G = 1$, resulting in,

$$F_{otherdomain} = \int_{\tau_1} N_1^T \left\{ (\nabla N_2 \cdot n_1)\, d_2 \right\} d\tau_1 \; + \int_{\tau_1} N_1^T \Lambda_1 N_2 d_2 d\tau_1, \tag{3.1}$$

which forms the basis for information transfer across the subdomains. The specific approach varies slightly between springs and PMDLs, which are elaborated in the following subsections.

3.1.1 Springs

The first expression on the right in (3.1) denotes the traction transfer from the neighboring subdomain. The interface is a line in 2D settings. N_1 and N_2 are the shape functions corresponding to the first (finer/bottom) and second (coarser/top) subdomains respectively (refer to Fig. 3.1). N_1 is the linear shape function on the finer interface and N_2 is the linear shape functions in the interior elements next to the interface ($i = 1$ in Fig. 3.1). Interpolation across non-matching grids can be done in several ways, such as nearest-neighbor interpolation, orthogonal projection methods, methods based on interpolation by splines (see [10] for further details). We utilize a simple orthogonal projection using standard Gauss Quadrature in our study (see e.g. [8], [9], [12]), which is consistent with Equation (3.1). Applying Gauss quadrature results in,

$$\int_{\tau_1} N_1^T \left[\frac{\partial}{\partial \hat{n}} N_2 d_2\right] d\tau_1 \;=\; \sum_{i=1}^{n_1} \sum_{j=1}^{n_g} N_1(\xi)\left[N_2(\xi^*, \eta = 0)\right] d_2 W_1(\xi) J_1 \tag{3.2}$$

Note that a similar expression would result from integration done over the other (coarser) interface. The notation used in equation (3.2) is:

n_1 = number of elements on the interface of the 1st subdomain(fine/bottom)

n_g = number of Gauss points for each element in 1st subdomain

ξ = co-ordinate of the Gauss point in the 1st subdomain

ξ^* = Projection of the Gauss point in the second subdomain(coarse/top)

$W_1(\xi)$ = weight corresponding to the Gauss points

J_1 = Jacobian calculated in the 1st subdomain

Fig. 3.1 represents (3.2) pictorially. An explanation of its conservation properties can be found in Appendix A.

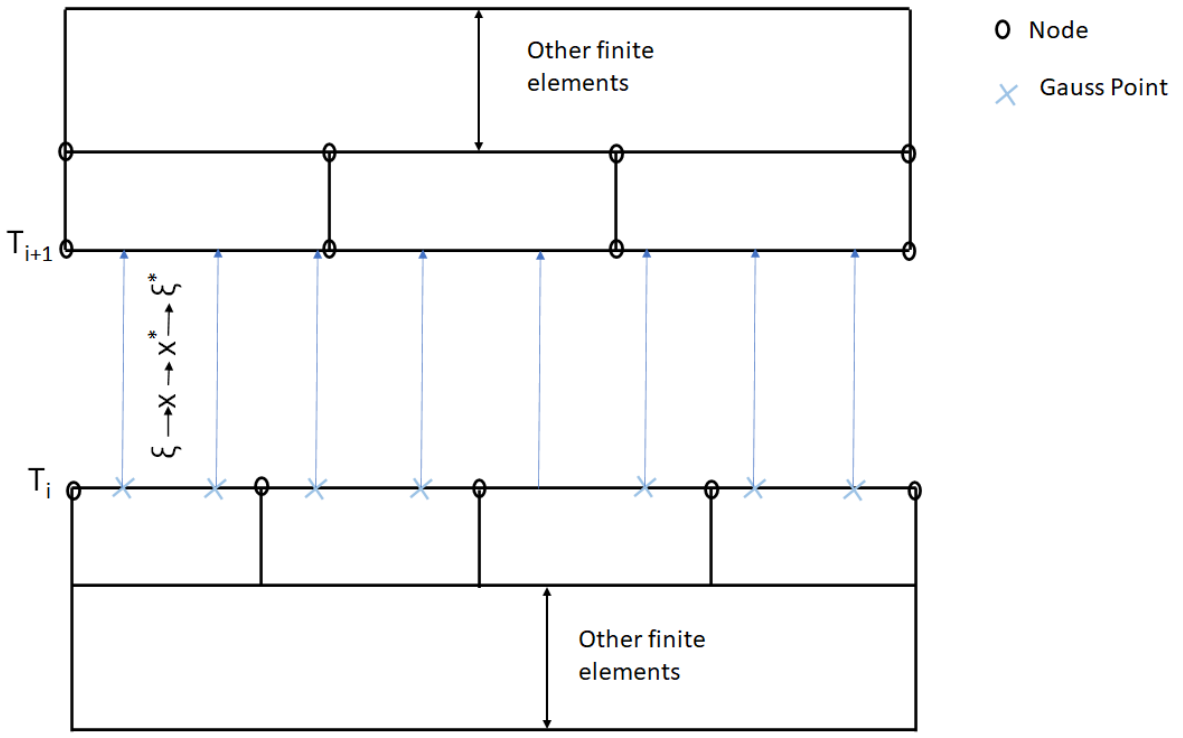


Figure 3.1 Schematic representation of transfer of traction in-between subdomains

The second expression on the right in (3.1) refers to the transfer of Dirichlet data. Considering Λ as constant, it can be taken out of the integration and the following is derived,

$$\Lambda_1 \int_{\tau_1} \left[N_1^T N_2 d\tau_1 \right] d_2$$

where

$$\Lambda_1 = \begin{bmatrix} \Lambda_1^1 & & \\ & \ddots & \\ & & \Lambda_1^{n_1+1} \end{bmatrix}$$

N_1 and N_2 are both interfacial (linear) shape functions corresponding to fine and coarse grids respectively. A similar orthogonal approach using Gauss Quadrature is used to calculate the resulting matrix, which is often called a *coupling matrix*.

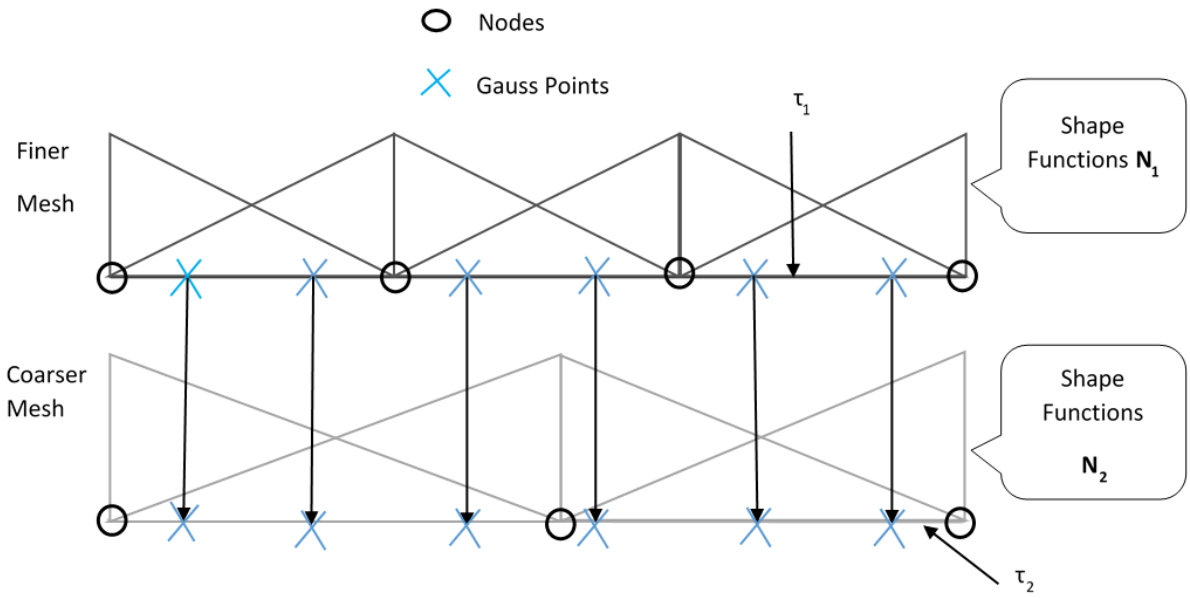


Figure 3.2 Schematic representation of coupling between two different shape functions using standard Gauss Quadrature

The shape functions on the interfaces are generally the trace of the shape functions in the interior. In this case, the integration is done on the finer interface with the help of standard Gauss quadrature with full integration (i.e. two points for each element). Interpolation is done for each Gauss point on the finer mesh from the coarser mesh, to build the contribution matrix for each element. The element contributions are assembled accordingly to derive the coupling matrix. The process for the reverse mapping, that is from the coarse to fine mesh, is done in a very similar fashion. Note that integrating over the finer interface yields an *extension* operator whereas integrating over the coarser interface yields *projection* operator.Figure Fig. 3.2 shows the methodology of calculating the coupling matrix.

$$\int_{\tau_1} N_1^T N_2 d\tau_1 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_g} N_1(\xi) N_2(\xi^*) W_1(\xi) J_1 \tag{3.3}$$

where,

n_1 = number of elements on the interface of the first subdomain(fine)

n_g = number of Gauss points for each element in first subdomain

ξ = coordinate of the Gauss point in the first subdomain

ξ^* = Projection of the Gauss point in the second subdomain

$W_1(\xi)$ = Weight corresponding to the Gauss points

J_1 = Jacobian calculated in the first subdomain

Let the assembly operator be denoted by A which assembles the contribution from each element to the global coupling matrix.

$$\int_{\tau_1} N_1^T N_2 d\tau_1 \;=\; \Big[\mathrm{A}\,[\kappa]\Big] \;=\; \Big[\mathrm{C}\Big]_{n_1 \times n_2}. \tag{3.4}$$

in (3.3), κ refers to the element contribution which is then assembled according to the connectivity. The coupling matrix C has a few properties which should be noted:

1. The coupling matrix has a size of $n_1 \times n_2$. n_1 is the total number of degrees of freedom on the 1st (finer) subdomain interface on which we are integrating for the coupling matrices. n_2 is the total degrees of freedom for the interface of the second subdomain. Needless to say, $n_1 \neq n_2$ as the meshes are not matching on the interface which makes the extension and the projection operators non-square matrices.

2. It is sparse non-symmetric with a tridiagonal structure, for linear finite elements,

i.e.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & 0 & 0 \\ a_{21} & a_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{mn-1} & a_{mn} \end{bmatrix}$$

3.1.2 PMDLs

The transfer of interior traction in the case of PMDL is exactly the same as the spring, which is explained in detail above and shown mathematically in (3.2) and pictorially in Fig. 3.1.

The transfer of the displacement from the adjoining subdomain is slightly different in this case. The Λ is not a constant or a diagonal matrix, as it is in the case of the springs. It is equal to the PMDL stiffness which is a coupled matrix. We incorporate a piecewise linear interpolation technique to map the displacement from the one subdomain to the other and vice verse. The interpolated displacement from the adjoining subdomain is multiplied with the PMDL stiffness (associated with the subdomain to which it is being interpolated), to get the contributions from the displacement part of the Robin data transfer.

3.2 Subdomain Decomposition

A model problem is taken as shown in the Fig. 3.2

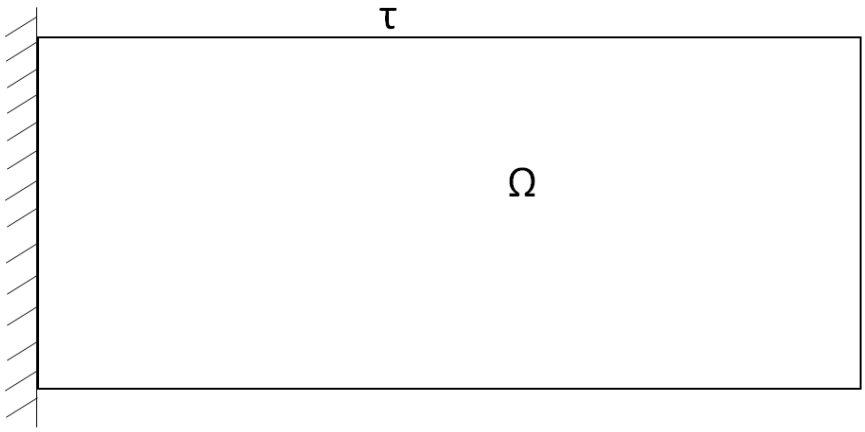
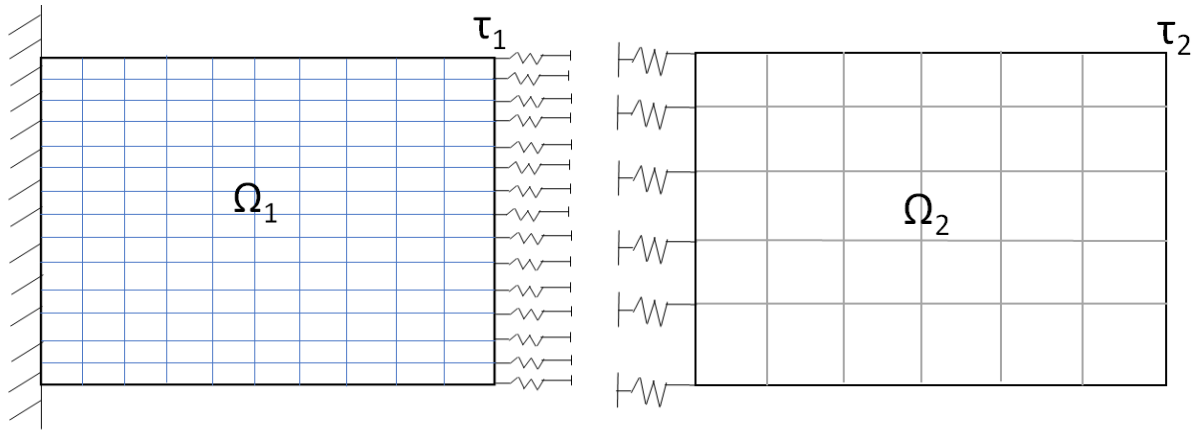


Figure 3.3 A Global domain with Dirichlet boundary condition on the left

3.2.1 1D Decomposition

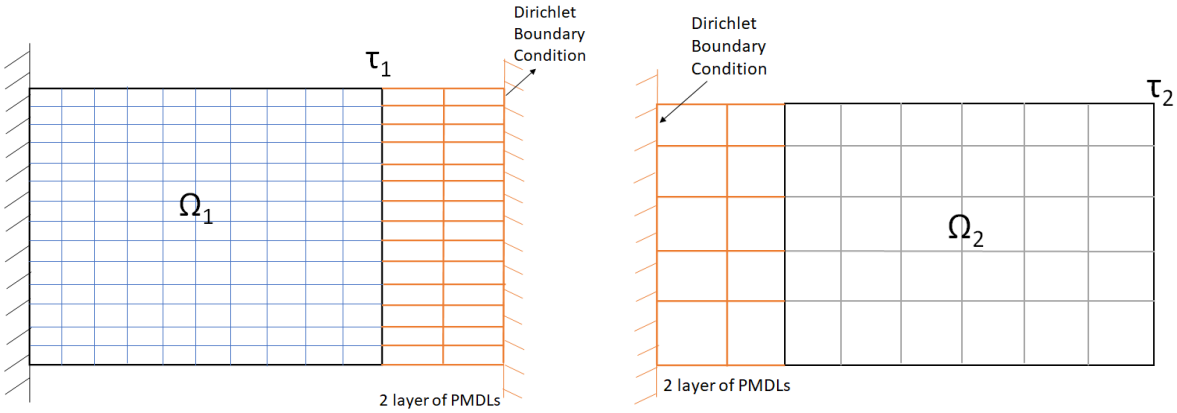
The global domain is decomposed into 2 subdomains and discretized separately and differently. As mentioned, there are two DtN maps considered in this study, springs and PMDLs, as shown in the Fig. 3.4 and Fig. 3.5. Either of them are attached at the interface to approximate the other domain. The number of (distributed) springs attached to each node is determined by the number of degree of freedom of the nodes (in this case, it is one, while e.g. it would be two for plane strain problems). The material properties are the same as the interior subdomains. The spring stiffness and the PMDL length and/or numbers are the parameters of the method and influences convergence as will be discussed in the next chapter on numerical examples.



(a) First subdomain with finer mesh

(b) Second subdomain with coarser mesh

Figure 3.4 Schematic representation of DtN maps (springs) at the interface



(a) First subdomain with finer mesh

(b) Second subdomain with coarser mesh

Figure 3.5 Schematic representation of DtN maps (PMDLs) at the interface

3.2.2 2D Decomposition

The same global domain is split in a two directions with varying discretization in each of them as shown in the (Fig. 3.6)

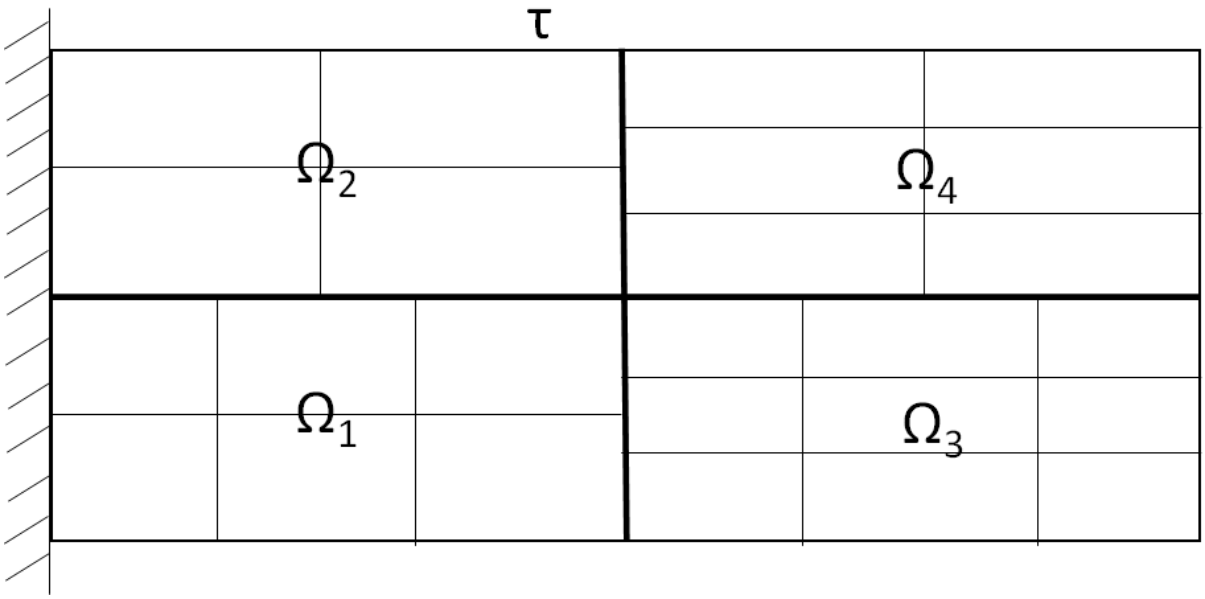


Figure 3.6 2D Split resulting in 4 subdomains

Fig. 3.7 and Fig. 3.8 shows the decomposition of the global domain and the subsequent individual local finite element discretization.

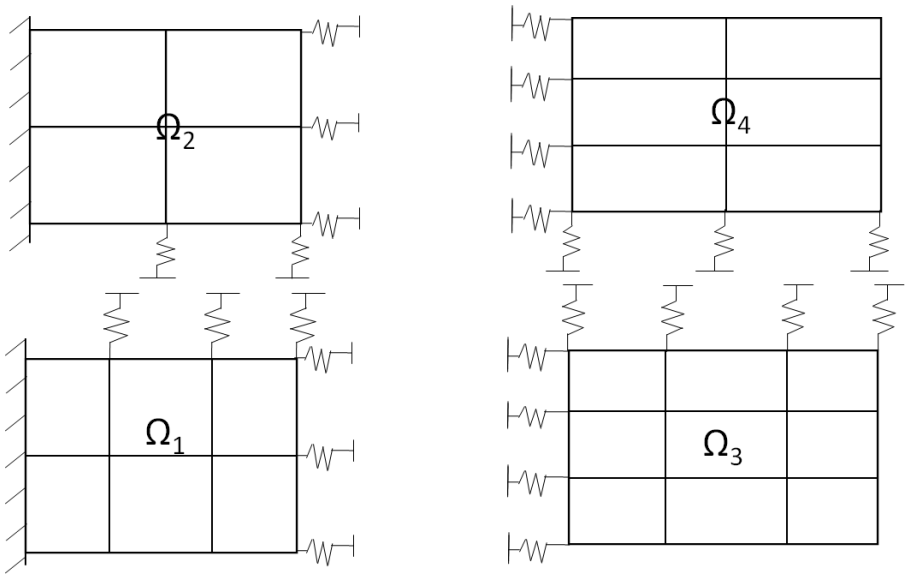


Figure 3.7 2D Split with springs as DtN map

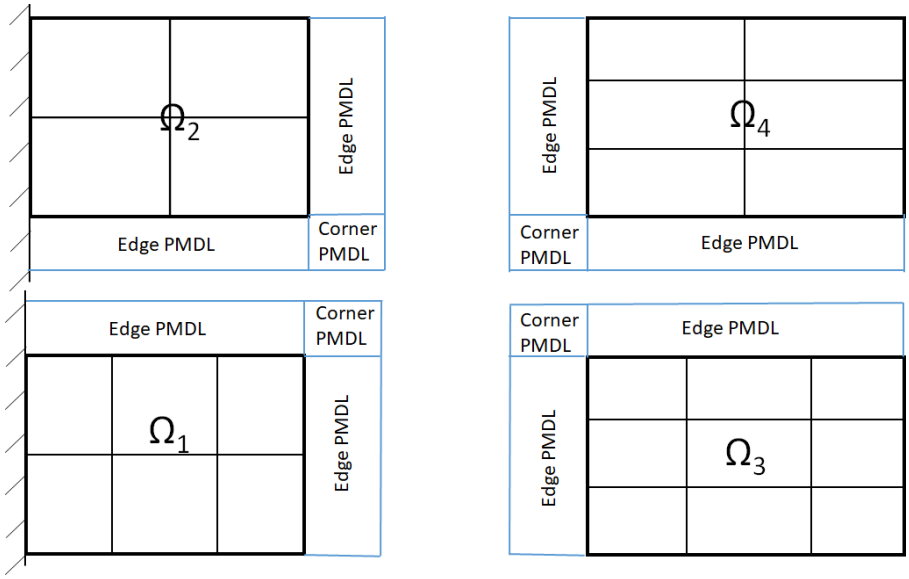


Figure 3.8 2D Split with PMDLs as DtN map

The main difference between a 1D and 2D split is the introduction of the corner elements. The PMDL stiffness, or the parameter Λ, incorporates the contribution from the corner elements along with the edge PMDLs.

3.3 Solution Procedure

The first example contains two non-overlapping subdomains. The transfer of information happens at the interface as shown in Fig. 3.9

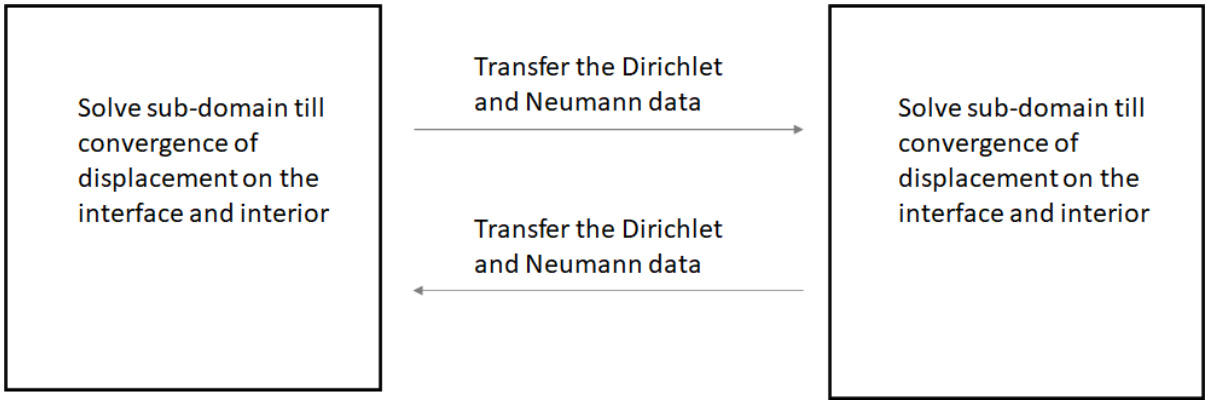


Figure 3.9 Two subdomains with transfer of information

For this two-subdomain case, an alternating methodology is incorporated, which is shown in Algorithm 1

Algorithm 1 Iterative Alternating Method for 2 subdomains

1: **function** SOLVER
2: Define Tolerance, global "iteration" number equals 1
3: Calculate the "*mass*" matrix on each interfaces, if required, to calculate external traction
4: Calculate the *coupling* matrices for each pair of interfaces
5: Repeat until convergence $\left[\frac{u_n - u_\infty}{u_\infty} < tolerance\right]$ for Ω_1
6: **if** iteration == 1 **then**
7: Guess traction and displacement from $\Omega_2 = 0$ on τ_{12}
8: Solve Ω_1
9: "Iteration" increased by 1
10: Reset the contribution from other other domain (Ω_2) to zero, if not already
11: **go to** 19
12: **else**
13: Extract the interior traction from Ω_2 on τ_{12} and map the values onto Ω_1 (f_{mapped})
14: Map the displacement from Ω_2 to Ω_1 on τ_{12} by linear interpolation/ mapping using the coupling matrix (d_{mapped})
15: Update RHS by adding $f_{mapped} + \Lambda \cdot d_{mapped}$ from Ω_2
16: **go to** 8
17: **end if**
18: Repeat until convergence $\left[\frac{u_n - u_\infty}{u_\infty} < tolerance\right]$ for Ω_2
19: **if** iteration == 1 **then**
20: Guess traction and displacement from $\Omega_1 = 0$ on τ_{21}
21: Solve Ω_2
22: "Iteration" increased by 1
23: Reset the contribution from the other domain (Ω_1) to zero, if not already
24: **go to** 6
25: **else**
26: Extract the interior traction from Ω_1 on τ_{21} and map the values onto Ω_2 (f_{mapped})
27: Map the displacement from Ω_1 to Ω_2 on τ_{21} by linear interpolation/mapping using the coupling matrix (d_{mapped})
28: Update RHS by adding $f_{mapped} + \Lambda \cdot d_{mapped}$ from Ω_1
29: **go to** 21
30: **end if**
31: **end function**

There can be cases where the need to decompose the global domain into more than two subdomains arises, and the algorithm has to be modified a bit. Instead of alternating approach, a sequential forward and backward sweeping across the subdomains is adopted. The sweeping is continued till satisfactory convergence on the interfaces and interior are achieved in all subdomains. In each step, information needs to be transferred from the adjoining subdomains through the mismatched interfaces. The extension to n ($n > 3$) number of subdomain is straightforward. The algorithm is also tested for a more complex problem of multi-dimensional decomposition (2D in this study). We incorporate a staggered forward and backward sweep to solve the subdomains iteratively. Both cases are represented pictorially in Fig. 3.10 and Fig. 3.11.

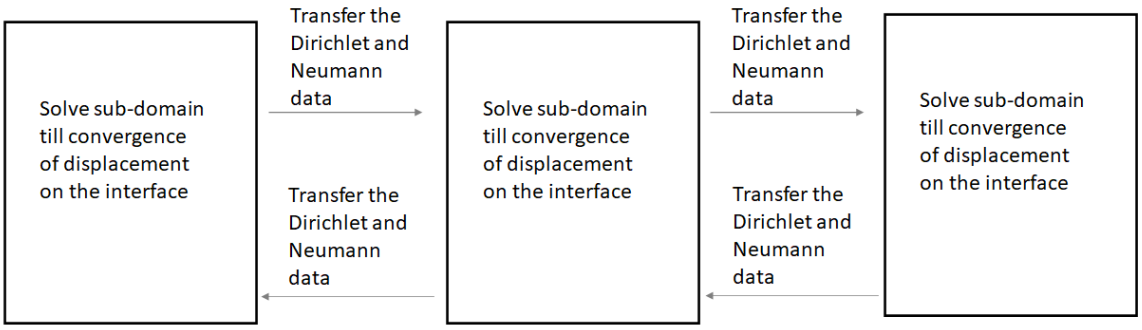


Figure 3.10 Three subdomains with exchange of information

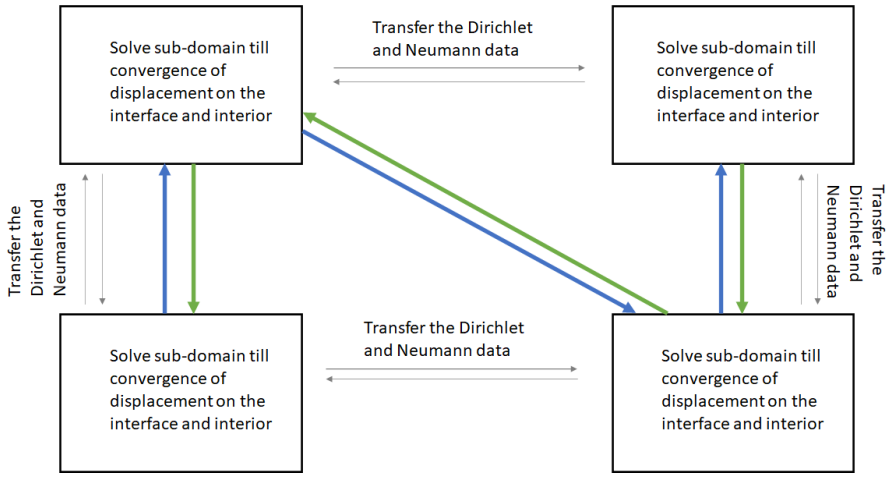


Figure 3.11 Two dimensional decomposition with staggered sweep

Algorithm 2 Iterative Method for 3 or more subdomains

function SOLVER
Define Tolerance, global "iteration" number equals 1
Calculate the "*mass*" matrix on each interfaces, if required, to calculate the external traction
Calculate the "*Coupling*" matrices for each pair of interfaces
    for $k \leftarrow 1$ to N **do**
        Repeat until convergence $= \frac{u_m - u_\infty}{u_\infty} < tolerance$ for Ω_k, where $m =$ iteration number
        if iteration == 1 **then**
            Guess traction and displacement from all adjoining subdomains = 0 through the common interfaces τ_{kn}, $n =$ adjoining subdomain so that $\Omega_k \cap \Omega_n : \tau_{kn}$
            Solve Ω_k
            "Iteration" number increased by 1
            Reset the contributions from the adjoining subdomains to zero, if not already
        else
            Extract the interior traction from all adjoining subdomains and map the values orthogonally to Ω_k (f_{mapped})
            Map the displacements from all adjoining subdomains by linear interpolation/mapping using coupling matrix (d_{mapped})
            Update RHS by adding $f_{mapped} + \Lambda \cdot d_{mapped}$ from all the adjoining subdomains
            go to 9
        end if
    end for
end function

CHAPTER 4

NUMERICAL EXAMPLES

The algorithm is tested for several cases of one dimensional and two dimensional splitting in the context of anti-plane shear. The loading conditions considered in this study are (a) Uniform (b) Trapezoidal and (c) Sine-square pulse, in the context of both interface conditions i.e. springs and PMDLs. The sine-square pulse is defined as $\sin^2 \alpha x$ where $0 \leq x \leq \pi/\alpha$ The relative error in the interior and interface is checked for satisfactory convergence. The implementation is performed in FEP++, a C++ based finite element code consisting of several user defined subroutines for pre-processing , solver and post-processing[13]. The meshing is done using a third party software called GMSH[14].To verify the finite element code, a comparison is done initially with ANSYS thermal model (steady state) and the FEP++ model without decomposition. The justification for using a thermal model is its similarity with FEP++ model (both are governed by the Laplace equation). The ANSYS and FEP++ models have the same grids. Only rectangular geometry is considered in this study. Further, for the sake of simplicity, only uniform grids are used; the extension to non-uniform meshing is straightforward.

4.1 Verification of FEP++ for Undecomposed Model

The model geometry and loading are shown in Fig. 4.1. Note that there is no decomposition in FEP++ model. Both the models have a square domain and a finite element grid of 20X20. The field variable (displacement/temperature) is calculated at each node and the relative error is calculated in terms of the L_2 norm of the difference, i.e.,

$$e_f = \left\| \frac{\mathbf{f}_1 - \mathbf{f}_2}{\mathbf{f}_1} \right\|^2 \tag{4.1}$$

where $\|\mathbf{f}\|^2 = \sqrt{\mathbf{f}^T\mathbf{f}}$

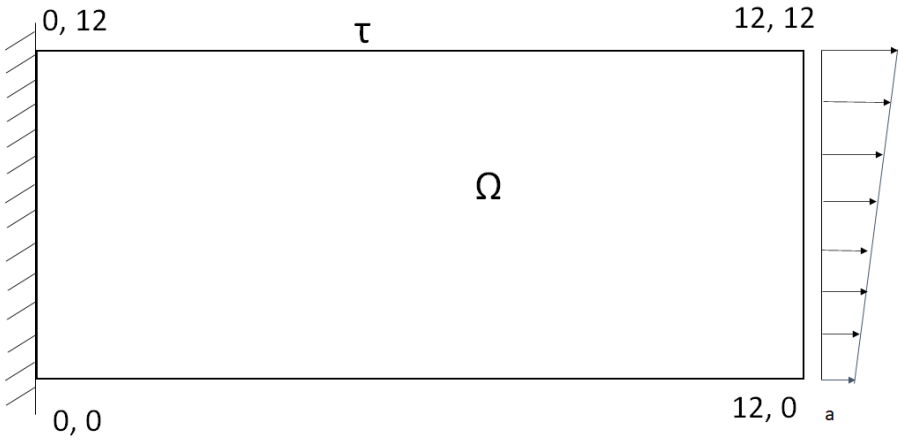
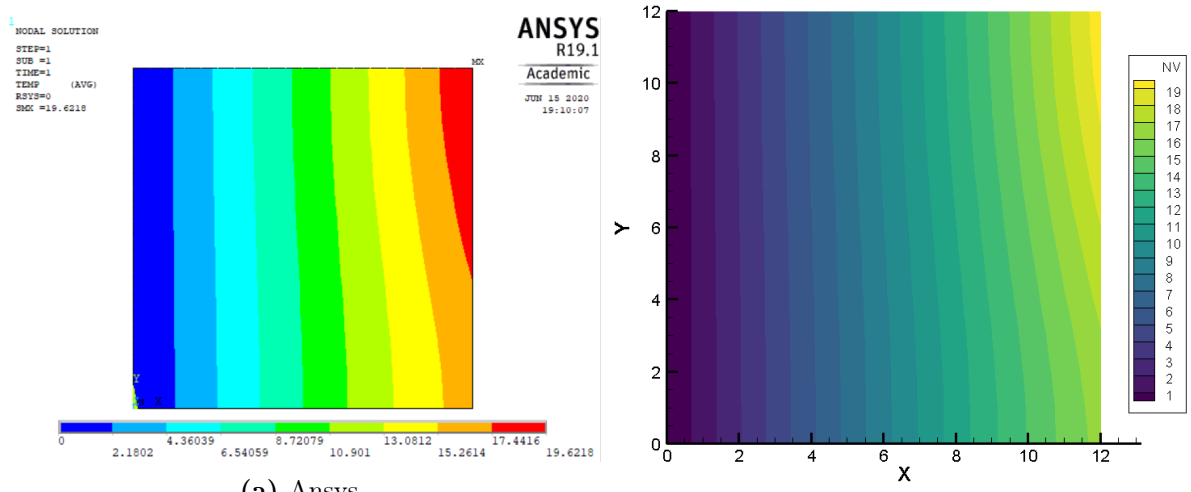


Figure 4.1 Domain without decomposition



(a) Ansys

(b) FEP++

Figure 4.2 Contour Plot of Nodal Value

Fig. 4.2 represents the displacement contours for the two analyses which looks very similar. A relative error was calculated using (4.1) and found to be of the order $1.0126X10^{-12}$ indicating satisfactory verification of the FEP++ program. Given this, hereafter, the reference solutions for various problems will be computed with undecomposed FEP++ model. It will be compared to the decomposed model solutions for satisfactory convergence to the correct solution.

4.2 Two subdomain Decomposition

The following Fig. 4.3 refers to the example that is being solved to verify the methodology in the anti-plane shear setting. The constitutive matrix D is:

$$D = G \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

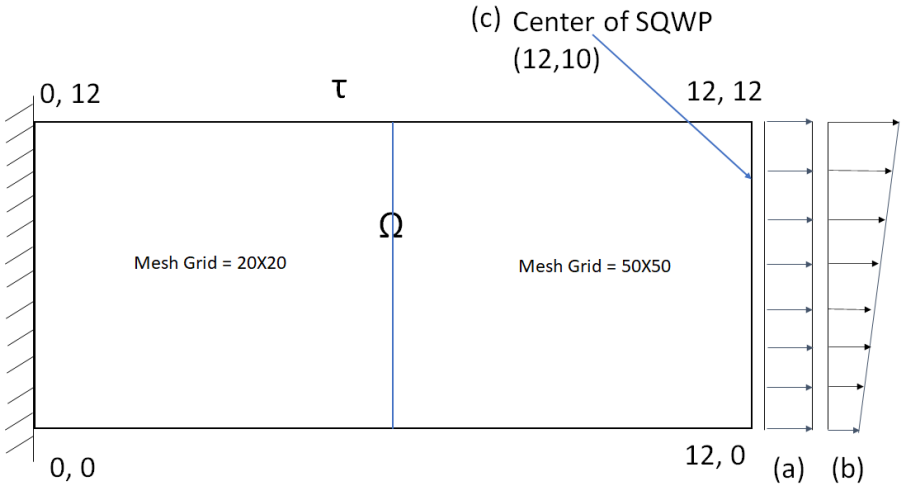


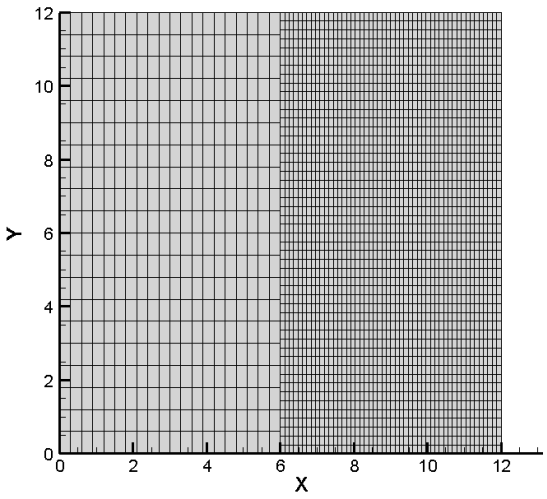
Figure 4.3 Two subdomains with different loading as indicated

To facilitate non-matching meshes, the first subdomain is meshed with a 20X20 grid and the second subdomain with a 50X50 mesh as shown in Fig. 4.3. Unless mentioned otherwise, meshing is kept the same for all kinds of loading explored in this study. Optimal choice for the parameters of the method vary with each local approximations, as mentioned in earlier chapters and is a proper research problem in itself. There are vast literature regarding a good and optimal choice of parameter[15, 16, 17, 18]. Optimal parameters depends on mesh grid on the interface, geometry, loading and boundary

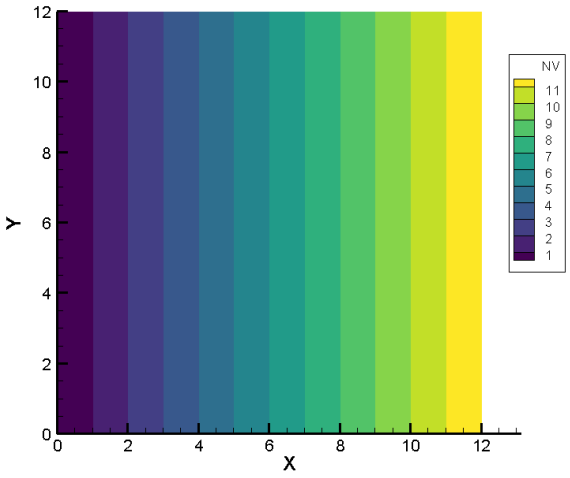
conditions among others. However, one way of choosing the optimal value of scalar Λ is by a *brute-force* approach for the spring case[19]. The boundary conditions are also taken into consideration when choosing the parameters for this example. The optimum value for the first subdomain is considered to be one-tenth of the value for the second subdomain for the given Fig. 4.3. A one-way and a two-way optimized value of Λ are taken from the literature [15] for comparison with the PMDL cases. In a practical application, we do not use the *brute-force* approach and solve the actual problem several times to find the best Λ. However, owing to the simplicity of the problem, and to understand the convergence behavior, we utilize both *brute-force* as well as the approaches in [15]. In the case of PMDLs the total length of the PMDL layers is chosen equal to the subdomain it is approximating. The convergence study is presented in Fig. 4.5. In Fig. 4.5(a), the algorithm is considered converged when the relative L_2 difference in consecutive iterations reaches a tolerance of 10^{-10}.

The convergence study shows that approximation by 2-layer PMDLs has satisfactory convergence in a few iterations. 3 layers are better and single layer gives a worse approximation and consequently bad convergence (worse than the "best" spring case). It is also to be noted that the PMDL gives better convergence than the spring stiffness taken from the literature. The convergence for different loading conditions are presented in Fig. 4.5b - Fig. 4.5d. Note that one way optimized case is referred to as "Optimized" and two way case is referred to as "2 way Optimized" in the plots. For uniform loading condition, PMDLs immediately converges to the exact solution in a single iteration given the exactness of PMDL in capturing the effect of the other domain for the specific (uniform) loading condition. Not surprisingly, this is not the case for other loading conditions, but PMDLs nevertheless lead to relatively quick convergence.

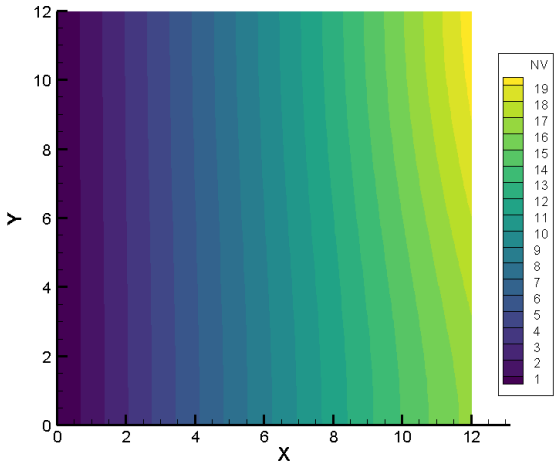
A comparison with the undecomposed model showed a relative norm of $1.0149X10^{-12}$ (uniform), $2.1067X10^{-4}$ (Trapezoidal) and $1.3314X10^{-3}$ (Sine Square Pulse), which is attributed to the discretization error.



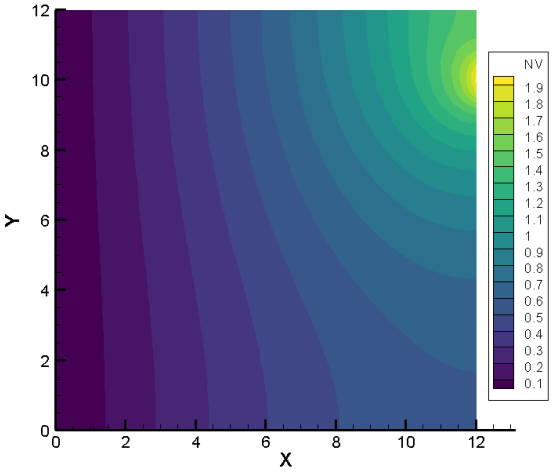
(a) Non-Matching Mesh Grid



(b) Contour for Uniform Load

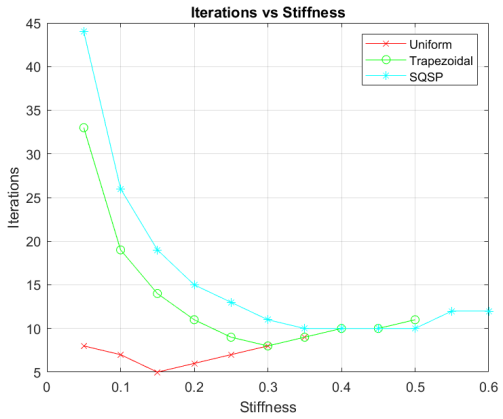


(c) Contour for Trapezoidal Load

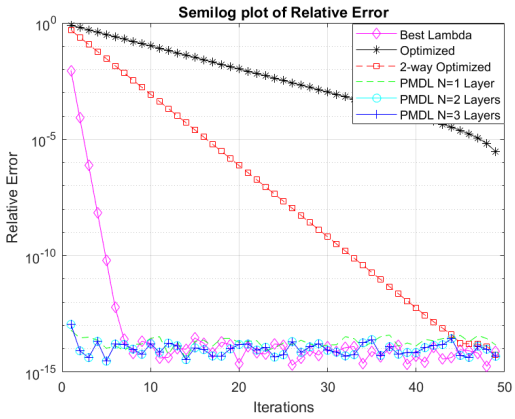


(d) Contour for SQSP

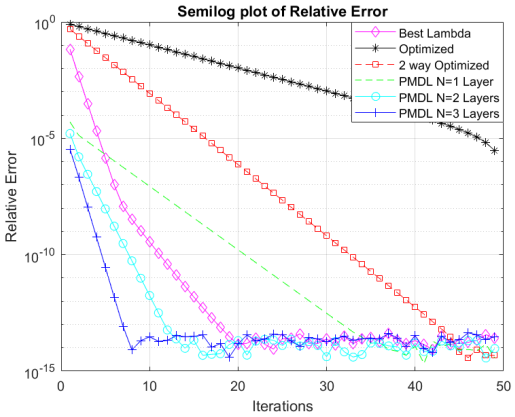
Figure 4.4 1D split with 2 subdomains



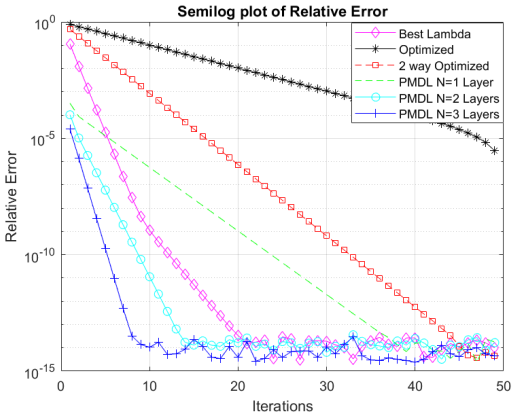
(a) "Best" Spring Stiffness vs Iterations



(b) Convergence for Uniform Load



(c) Convergence for Trapezoidal Load



(d) Convergence for SQSP

Figure 4.5 1D split with 2 subdomains

4.3 Two-Dimensional Split

The next case considered in this study is the two-dimensional decomposition. The geometry with the loading conditions are shown in Fig. 4.6. The stiffness of the springs (λ) are chosen to be same for all the interfaces for the case where we solve the problem several times to get the "best" parameter possible. However as mentioned in section 4.2, we never solve the problem several times but use a value to get the desired solution. Similar to the section 4.2, the "optimal" value is taken from the literature[15] (both one-way and two-way optimized).

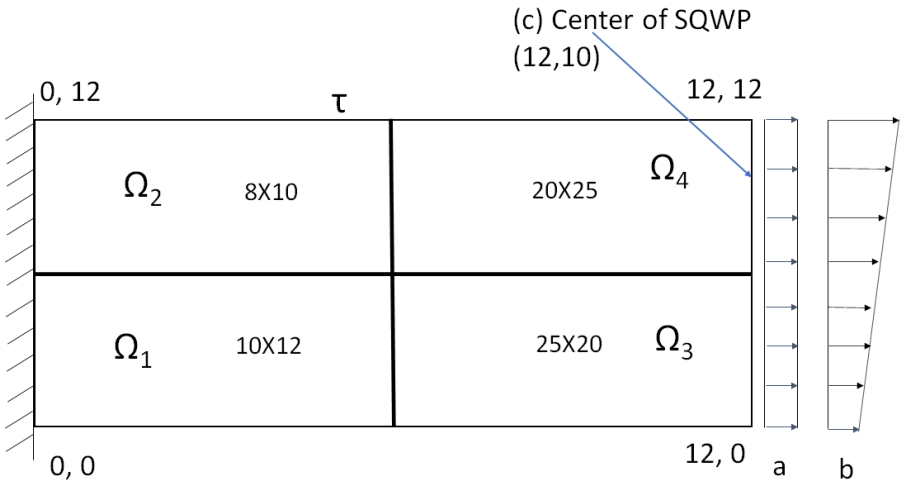
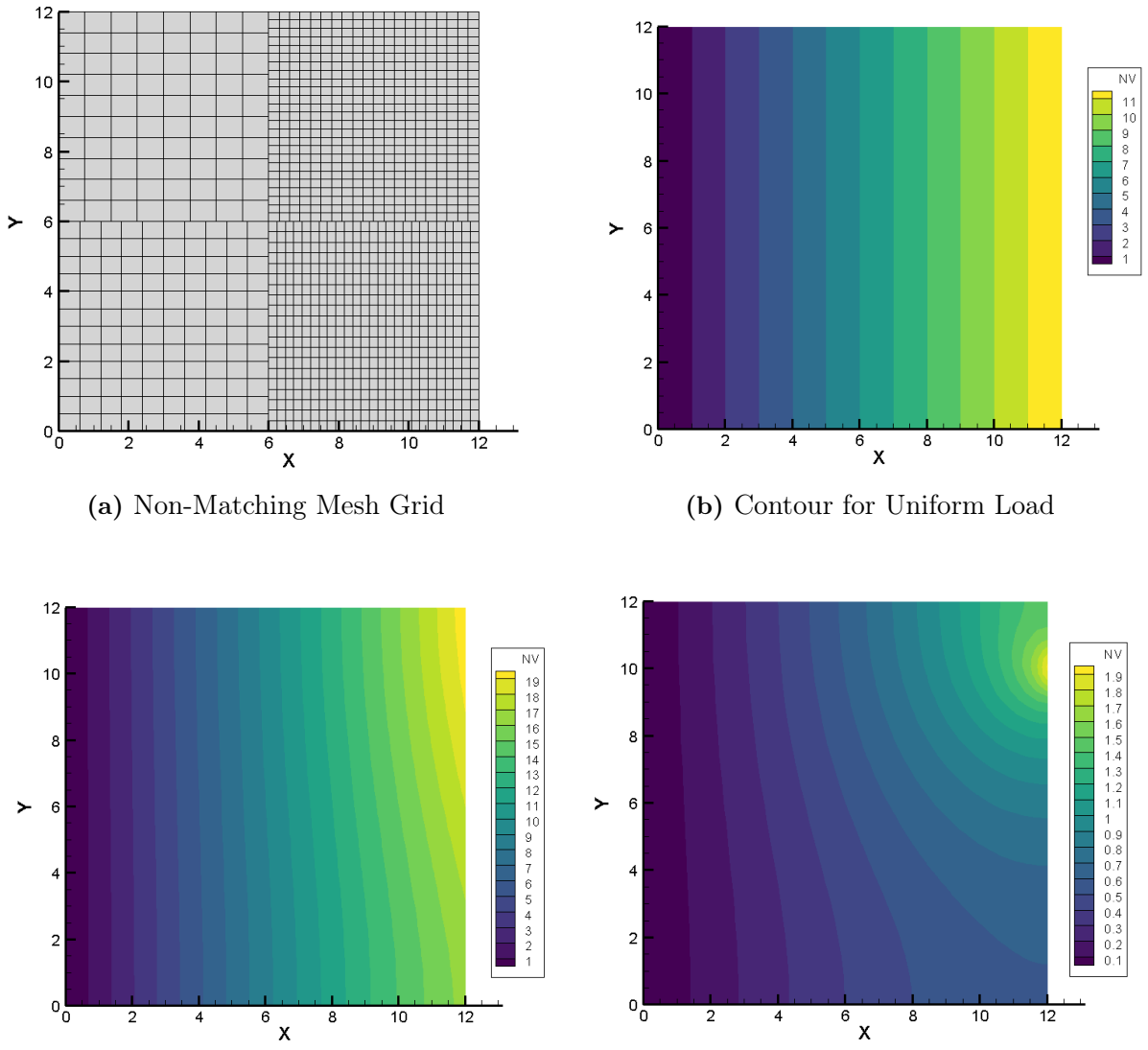


Figure 4.6 Two subdomains with different loading as indicated

Fig. 4.7b - Fig. 4.7d and Fig. 4.8b - Fig. 4.8d show the displacement contours and the convergence plots respectively. It is to be noted that PMDLs do not give any discernible advantage over the springs (the best case), although it being slightly better in approximation. However, it is better than the parameter taken from the literature, which is a more practical comparison. The result converges in all the cases. An interesting modification would include changes in the PMDLs to incorporate the effect of geometry

and decomposition in its algorithm to have better convergence than regular springs. This is out of scope of this study and will be possibly dealt in the future extensions. More PMDL layers would give better convergence, which is not surprising.

A comparison with the undecomposed model showed a relative norm of $1.5926X10^{-10}$ (uniform), $5.3784X10^{-4}$ (Trapezoidal) and $1.2748X10^{-2}$ (Sine Square Pulse)



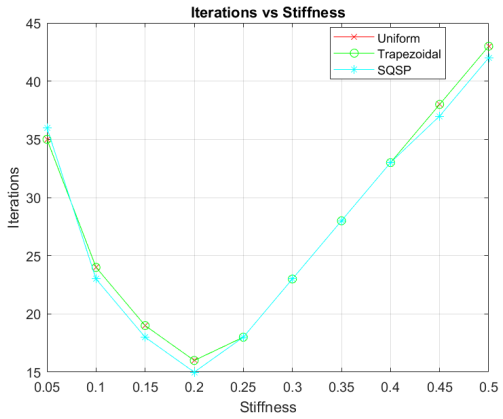
(a) Non-Matching Mesh Grid

(b) Contour for Uniform Load

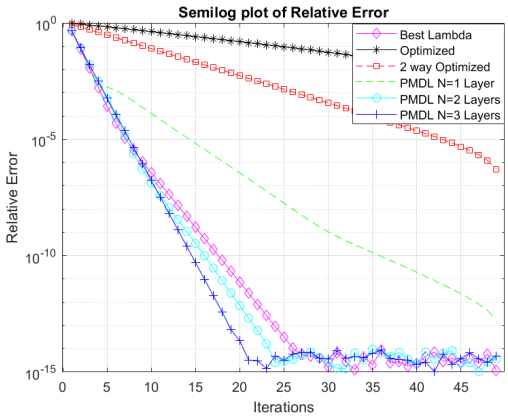
(c) Contour for Trapezoidal Load

(d) Contour for SQSP

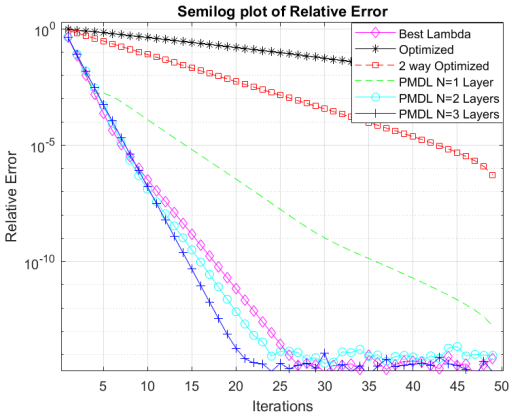
Figure 4.7 2D split with 4 subdomains



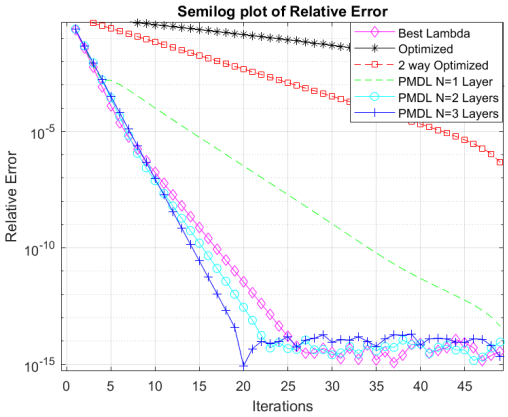
(a) "Optimum" Spring Stiffness vs Iterations



(b) Convergence for Uniform Load



(c) Convergence for Trapezoidal Load



(d) Convergence for SQSP

Figure 4.8 2D split with 4 subdomains

4.4 Sand-Box Problem

Having gained confidence in the algorithm developed so far with some synthetic examples, a more realistic *sand-box* problem is considered. The geometry and an example discretization are as shown in the Fig. 4.9. The top center subdomain, where the loading is applied, is discretized with densest grid with gradual decrease in the neighboring subdomains. The idea of finding the "best" parameter is dropped in this realistic problem and we resort to the "optimal" parameter from the literature[15]. The reason for that, as stated already in 4.2, is we do not iterate and solve the problem several times to find the "best" possible parameter but use a parameter (from existing literature maybe) to get the solution.

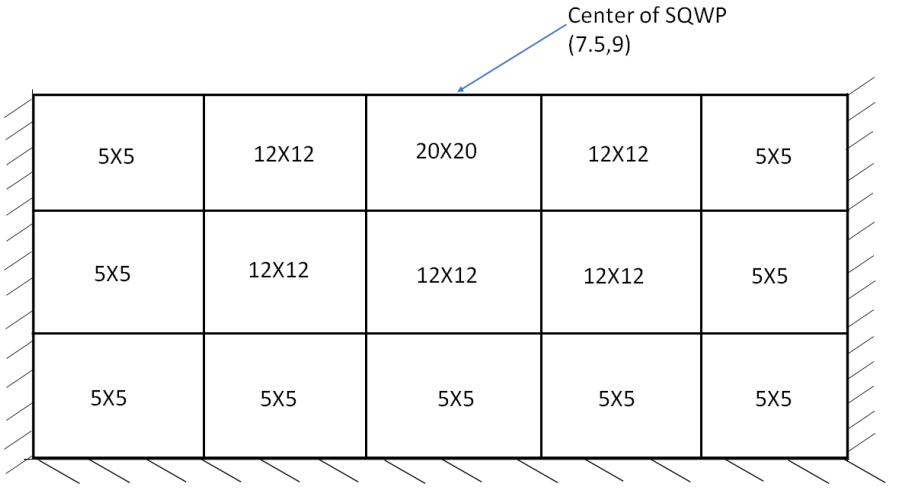
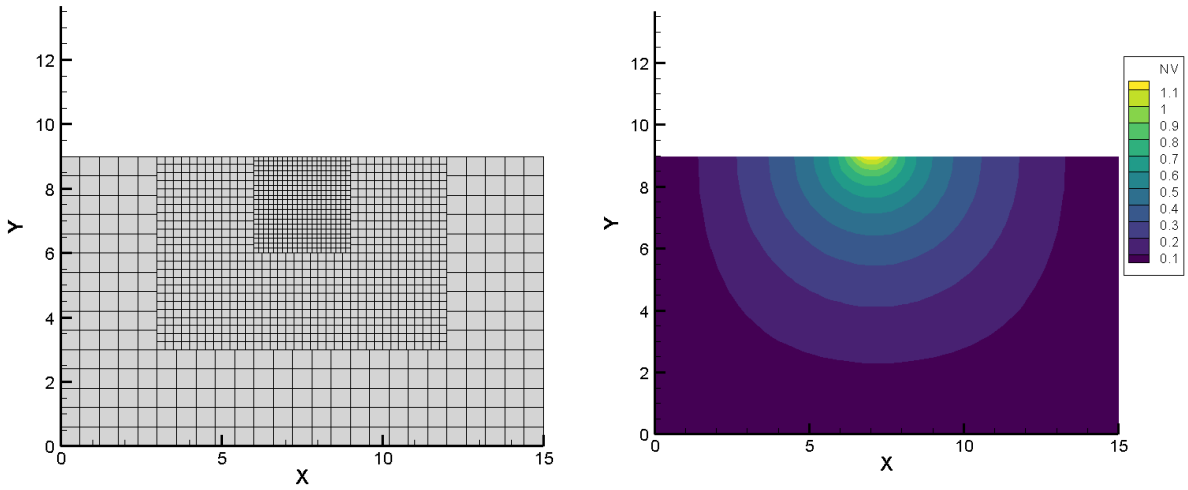


Figure 4.9 Sand-Box Model



(a) Mesh **(b)** Contour of Displacement

Figure 4.10 SandBox Problem

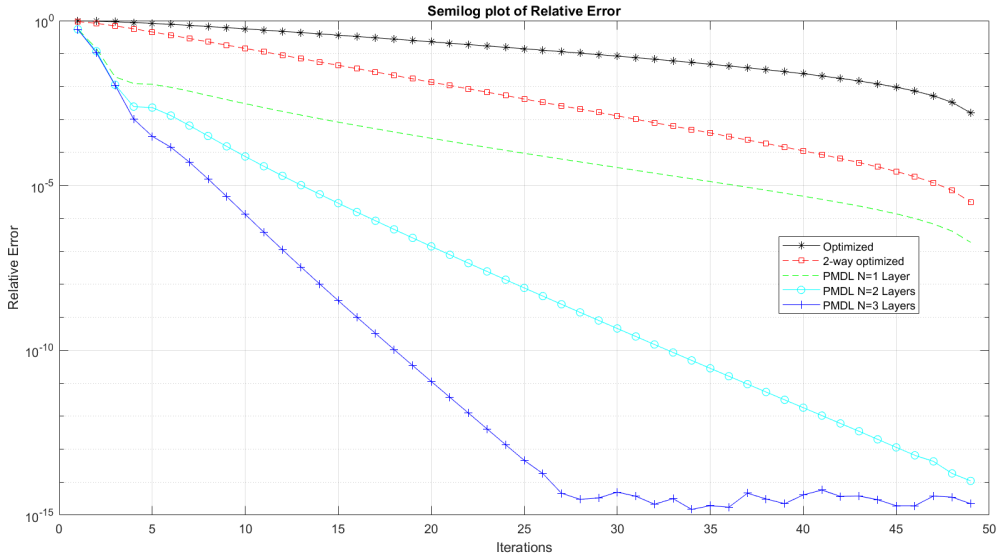


Figure 4.11 Convergence plots of Displacement for different parameters

The algorithm works properly in a complicated geometry as shown in Fig. 4.9 and the solution converges. The PMDL parameter (i.e. length) is adjusted according to the number of layers that are attached to the interface so that the total length is equal to the

length of the subdomains, the stiffness of which it is trying to approximate. As seen in Fig. 4.11, the PMDL approximation is more efficient than the "optimal" Λ found in the literature[15]. As expected, the convergence gets better with increase in the number of PMDL layers with a very slight increase in computational cost. Evidently, PMDL seems to be a good choice.

A comparison with the undecomposed model showed a relative norm of $2.9564X10^{-2}$, which again is attributed to the discretization error.

CHAPTER 5

SUMMARY and CONCLUSIONS

With the motivation of reducing the meshing cost, a domain-decomposition solver is developed where the domain is decomposed, and each subdomain is independently discretized. The resulting mesh discrepancy is handled through orthogonal projection, while the information transfer is handled through Robin data transfer built on the concept of perfectly matched discrete layers (PMDL). The resulting interface conditions result in fast convergence and thus computational efficiency. Numerical examples are presented to illustrate the efficacy of the method.

The main focus is on the transmission conditions across mismatched grids and implementation of corner elements in a two-dimensional decomposition. The convergence of the solution signifies proper transmission of information across an interface which are not matched or conforming. A relative comparison between the decomposed and undecomposed models indicates satisfactory accuracy. Rectangular subdomains with uniform mesh is considered this study, but the ideas should extend to more general subdomains. Several loading conditions along with a combination of relatively simple and complicated

geometries are examined. In general, PMDLs prove to be a better choice compared to distributed springs that are prevalent in the literature. Some of the possible future points of explorations, based on this study, are,

1. Extending the methodology to more complicated geometries and decomposition.
2. Exploring the applicability to frequency ($\omega \neq 0$) and time domain analyses.
3. Using the proposed DD method as a preconditioner to iterative linear solvers.

BIBLIOGRAPHY

[1] V. Dolean et al. "An Introduction to Domain Decomposition Methods: algorithms, theory and parallel implementation". Master. Lecture. France, Jan. 2015.

[2] P. L. Lions. "On the Schwarz Alternating Method III: A Variant for Nonoverlapping Subdomains". In: *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*. Ed. by T. F. Chan et al. Society for Industrial and Applied Mathematics, 1990. Chap. 11, pp. 202–223.

[3] B. Engquist and H.-K. Zhao. "Absorbing boundary conditions for domain decomposition". In: *Applied Numerical Mathematics* 27.4 (1998). Special Issue on Absorbing Boundary Conditions, pp. 341 –365.

[4] J.-P. Berenger. "A perfectly matched layer for the absorption of electromagnetic waves". In: *Journal of Computational Physics* 114.2 (1994), pp. 185 –200.

[5] M. N. Guddati and K.-W. Lim. "Continued fraction absorbing boundary conditions for convex polygonal domains". In: *International Journal for Numerical Methods in Engineering* 66.6 (2006), pp. 949–977. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1574`.

[6] S. Savadatti and M. Guddati. "A finite element alternative to infinite elements". In: *Computer Methods in Applied Mechanics and Engineering* 199 (July 2010), pp. 2204–2223.

[7] C. Farhat et al. "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity". In: *Computer Methods in Applied Mechanics and Engineering* 157.1 (1998), pp. 95 –114.

[8] J. R. Cebral and R. Lohner. "Conservative Load Projection and Tracking for Fluid-Structure Problems". In: *AIAA Journal* 35.4 (1997), pp. 687–692. eprint: https://doi.org/10.2514/2.158.

[9] R. K. Jaiman et al. "Assessment of conservative load transfer for fluid–solid interface with non-matching meshes". In: *International Journal for Numerical Methods in Engineering* 64.15 (2005), pp. 2014–2038. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1434.

[10] A. de Boer et al. "Review of coupling methods for non-matching meshes". In: *Computer Methods in Applied Mechanics and Engineering* 196.8 (2007). Domain Decomposition Methods: recent advances and new challenges in engineering, pp. 1515 –1525.

[11] S. Thirunavukkarasu and M. Guddati. "A domain decomposition method for concurrent coupling of multiscale models". In: *International Journal for Numerical Methods in Engineering* 92 (Dec. 2012), pp. 918–939.

[12] D. Dureisseix and H. Bavestrello. "Information transfer between incompatible finite element meshes: Application to coupled thermo-viscoelasticity". In: *Computer Methods in Applied Mechanics and Engineering* 195.44 (2006), pp. 6523 –6541.

[13] M. Guddati et al. "FEP++: An Object Oriented Finite Element Program Linear and Nonlinear Transient Analysis". In: *unpublished technical note* (Jan. 2003).

[14] 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. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2579.

[15] M. J. Gander. "Optimized Schwarz Methods". In: *SIAM J. Numer. Anal.* 44.2 (Feb. 2006), 699–731.

[16] M. Gander and F. Kwok. “Best Robin Parameters for Optimized Schwarz Methods at Cross Points”. In: *SIAM Journal on Scientific Computing* 34 (Jan. 2012).

[17] M. GANDER. “Schwarz methods over the course of time”. In: *ETNA. Electronic Transactions on Numerical Analysis [electronic only]* 31 (Jan. 2008).

[18] M. J. Gander et al. “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. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1433`.

[19] M. J. Gander et al. “A New Cement to Glue Nonconforming Grids with Robin Interface Conditions: The Finite Element Case”. In: *Domain Decomposition Methods in Science and Engineering*. Ed. by T. J. Barth et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 259–266.

[20] M. Guddati. “Arbitrarily wide-angle wave equations for complex media”. In: *Computer Methods in Applied Mechanics and Engineering* 195 (Jan. 2006), pp. 65–93.

APPENDICES

APPENDIX A

Orthogonal Projection is conservative

The domain based method that has been incorporated in this study is the standard finite elements with Bi-linear shape functions in the interior and linear shape functions on the interfaces. We use the summation property of the shape functions that is $\sum Ni = 1$ to show that any physical quantity can be conserved across the interface. The following equation has already been defined in Chapter 3

$$\left[\int_{\tau_1} N_1^T N_1 d\tau_1\right] \zeta_1 \;=\; \int_{\tau_2} N_1^T N_2 \zeta_2 d\tau_2 \tag{A.1}$$

We try to prove that the physical quantity ζ are conserved on each of the interfaces

$$\int_{\tau_1} \zeta_1 d\tau_1 \;=\; \int_{\tau_2} \zeta_2 d\tau_2 \tag{A.2}$$

Here ζ are any continuous physical quantity such as displacement or force. The discretized form will be referred to as $\hat{\zeta}$. Starting from the left hand side of equation, considering

(A.2) and considering $\zeta_1 = N_1\hat{\zeta_1}$ and $\sum N_2 = 1$, $\sum N_1 = 1$

$$\begin{aligned}\int_{\tau_1} N_1\hat{\zeta_1}d\tau_1 &= \int \sum N_1^i N_1^j \hat{\zeta_1^j} d\tau_1 = \sum \int_{\tau_1} N_1^i N_1^j \hat{\zeta_1^j} d\tau_1 \\ &= \sum \int_{\tau_2} N_1^i N_2^j \hat{\zeta_2^j} d\tau_2 \; = \; \int_{\tau_2} \sum N_1^i N_2^j \hat{\zeta_2^j} d\tau_2 \\ &= \int_{\tau_2} N_2^j \hat{\zeta_2^j} d\tau_2 = \int_{\tau_2} \zeta_2 d\tau_2\end{aligned} \tag{A.3}$$

APPENDIX B

Perfectly Matched Discrete Layers

PMDLs were originally developed for modeling unbounded domains in the context of transient scalar wave propagation problems [5]. Further studies have shown that it can be applied to modeling unbounded domains where the governing partial differential equation is second order in space [20]. This algorithm can be extended to static and time-harmonic analysis too. As this report concerns with a static analysis, we turn to the literature by Savaddati and Murthy [6] for references. The generalized static anti-shear plane governing equation is of the form (without body force),

$$\mu \frac{\partial^2 u}{\partial x^2} + \mu \frac{\partial^2 u}{\partial z^2} = 0 \tag{B.1}$$

Referring back to Fig. 2.1 and considering $x_0 \leq x < \infty$, the objective is to replace the half-space by PMDL/finite element layers. Fourier transforming the (B.1), the following

linear equation in one variable results,

$$\mu\left(\frac{\partial^2 u}{\partial x^2} - k_z^2 u\right) = 0 \tag{B.2}$$

k_z represents the spatial frequency in the vertical direction and is related to the orthogonal spatial frequency k_x through the dispersion relationship,

$$k_x^2 + k_z^2 = 0 \tag{B.3}$$

Further, the traction boundary condition at $x = x_0$ can be related to the exact half-space stiffness through the equation (B.4). Solving (B.2), applying the boundary condition at $x = \infty$ and using (B.4), the exact stiffness of the half-space can be related to the spatial frequency, in a static setting, as shown in (B.5). The interesting part in (B.5) is the independence of the location of the right half-space. It means the right half space will have the same dynamic stiffness at the left boundary irrespective of where the boundary lies along x-direction.

$$\mu\frac{\partial u}{\partial x}\bigg|_{x=x_0} + K_{exact}u_0 = 0 \tag{B.4}$$

$$K_{exact} = \mu k \text{ where } k = |k_z| \geq 0 \tag{B.5}$$

To numerically simulate the stiffness of the right half-space, the need to replace the half-space by finite elements is imminent. If the right half-space (x_0, ∞) is replaced by one finite element (with linear shape functions and x_0, x_1) and a 'smaller' half space (x_1, ∞), the total stiffness of the composite half-space is not expected to be equal to K_{exact}. Needless to say, this happens because of the discretization error of the finite element. The 'smaller' half space still has stiffness equal to K_{exact} at the left boundary (x_1). If the

composite stiffness at the left boundary (x_0) is denoted by K_0 and the displacement by u_0, the following equation results,

$$\begin{Bmatrix} K_0 u_0 \\ 0 \end{Bmatrix} = \left(\begin{bmatrix} a_1 & a_2 \\ a_2 & a_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{exact} \end{bmatrix} \right) \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} \tag{B.6}$$

Condensing out u_1 from (B.6) and simplifying it,

$$K_0 = a_1 - \frac{a_2^2}{a_1 + K_{exact}} \tag{B.7}$$

Making $K_0 = K_{exact} = \mu k$, substituting in (B.7) and simplifying,

$$a_1^2 - a_2^2 = (\mu k)^2 \tag{B.8}$$

Equation (B.8) refers to an important property called "fixed point", which implies if a finite element having the above property is attached to an existing half-space, it churns out an exact half-space (stiffness) again. In other words the finite element does not have any discretization errors in calculating the dynamic stiffness of the half-space. Discretizing equation (B.2) with standard Bubnov Galerkin finite elements, a discrete form is evaluated. Gauss Quadrature with full integration evaluates the values of a_1 and a_2 as shown in (B.9) and, with mid-point integration, as shown in (B.10). It is clear from (B.9) and (B.10) that a mid-point integrated finite element satisfy the "fixed point property" and is used in constructing the matrix for calculating the half-space stiffness.

$$a_1 = \mu \left(\frac{1}{L} + \frac{k^2 L}{3} \right) \text{ and } a_2 = \mu \left(-\frac{1}{L} + \frac{k^2 L}{6} \right) \tag{B.9}$$

$$a_1 = \mu \left(\frac{1}{L} + \frac{k^2 L}{4} \right) \text{ and } a_2 = \mu \left(-\frac{1}{L} + \frac{k^2 L}{4} \right) \tag{B.10}$$

The absence of any discretization error in a finite element does not seem intuitive at first as any finite element, no matter how fine, should have some errors. This accuracy is achieved at a loss of information in the exterior (for example displacement in this case), whose stiffness is being calculated. However, the stiffness of the exterior is needed only at the interface and there is no inherent interest in information far away from the interface inside the exterior. In general, the PMDL stiffness (in frequency regime)is $k - m\omega^2$ for both edge and corner elements. The derivation, although being done in frequency regime, can be extended to time domain analysis without the need of any inverse Fourier transform of the former (which might lead to complicated global operators). In time domain analysis, the regular stiffness $\left(\int B^T B d\Omega\right)$ is calculated and attached to the interior domain of interest.