Final Project Report

Domain Decomposition

Muhammad Moeeze Hassan
José Andrés Pérez

Submitted to
Pierre Alain Guidault

Université Paris Saclay
04 March 2022

Contents

List of Figures

Chapter 1

Preliminary Data Setting Work

1.1 Analytical Solution

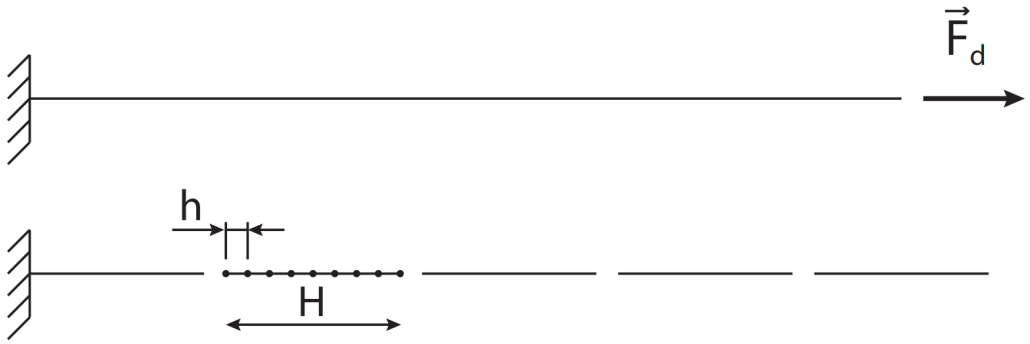


Figure 1.1: Analyzed Geometry

The objective of this project is to analyze the behavior of different domain decomposition methods, which are based on primal quantities (displacement), dual quantities (stress), and a combination of both (mixed approaches). Special interest will be taken to analyze the numerical scalability of the methods. All of the domain decomposition methods will be applied to the geometry shown on Figure 1.1, which is a simple case of a beam under tension divided into subdomains with length H, which are later subdivided into elements of length h. This will allow us to develop the techniques and not incur on numerical difficulties not proper to the course.
For reference, the analytical solution of a beam under tension with a fixed end is the following:

$$\frac{F * X}{S * E} = \delta X \tag{1.1}$$

This shows that the expected response is linear Figure 1.2 and it depends on the Force applied to the beam (F), the surface area orthogonal to the force (S), and the the Young Modulus (E). For the following parameters, the linear response obtained is this one:

- F $= 1 * 10^6[N]$
- E $= 2.1 * 10^{11}[Pa]$
- S $= 1 * 10^{-5}[m^2]$

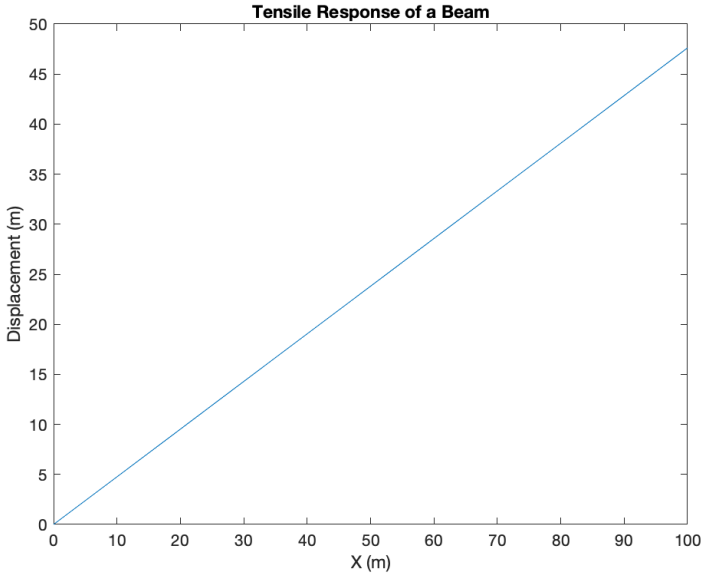


Figure 1.2: Analytical Response

These parameters where chosen to have observable values of deformation.

1.2 Finite Element Solution

The aim of this section is to develop a Finite element code to solve the beam system subjected to the following development constrains.

- Possibility to have arbitrary number of elements of size h
- Uniform mesh (constant h)
- Calculation of the stiffness matrix by assembly of the elementary stiffness matrix (exact integration)
- Take into account the boundary conditions

For this case, the bilinear form, which corresponds to the stiffness matrix is the following:

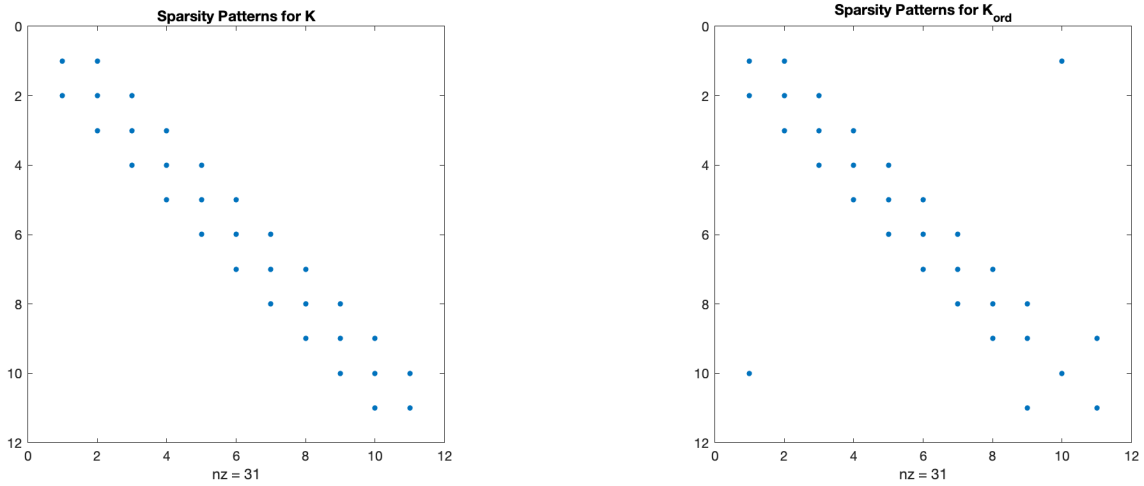
$$E * S * \int \frac{du_h}{dx} * \frac{dv_h}{dx} \tag{1.2}$$

Where u_h and v_h are the discretized form of the solution and the test functions. Piece-wise linear shape functions of degree one are chosen. Which result on the following stiffness matrix after integration:

$$\frac{E * S}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{1.3}$$

This unitary stiffness matrix is later assembled to obtain the global stiffness matrix. From a coding perspective this requires a map based on the connectivity of the elements, however since the analyzed system is a 1D beam, they are all classically connected one after the other, which results in a matrix with nonzero elements on the main diagonal, and a diagonal above and below. This mapping latter changes when there is a need to group the solution array in a different way (Internal Elements followed by boundary elements). The followings Figures show the nonzero values on the stiffness matrices, which exemplify the position taken by the terms of the elementary matrix once they are assembled

(the first figure showing the traditional assembly, and the second one showing a different assembly with internal elements u_{ii} followed by boundary elements u_{bb}).



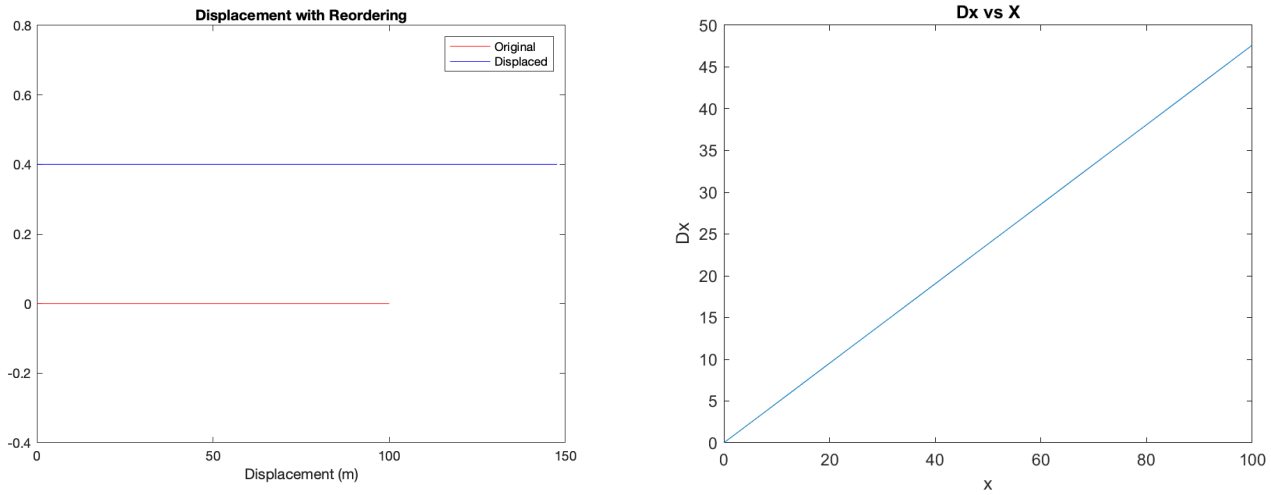
(a) Spy of Stiffness matrix

(b) Spy of Ordered stiffness matrix

Figure 1.3: Spy of K and ordered K

$$K * U = F \longrightarrow \begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & Kbb \end{bmatrix} \begin{bmatrix} u_{ii} \\ u_{bb} \end{bmatrix} = \begin{bmatrix} f_i i \\ f_{bb} \end{bmatrix} \tag{1.4}$$

The solution of this 1D FEM simulation is presented in two graphs, one that shows the non-deformed and the deformed length of the beam, and a second one that displays the δX at each point of the beam (it can be observed that the same linear response was observed to the one obtained using the analytical solution.



(a) Deformed vs original beam lengths

(b) FEM Displacement vs x

Figure 1.4: Deformation Results from FEM

1.3 Domain Decomposition Operators

First, to implement the Domain decomposition techniques, it was necessary to divide the beam into the different subdomains. These domains all have the same size as can be seen of Figure 1.1 (length

H). And they are usefull since most of the operators that will be later used are defined per subdomain, and relate quantities on the subdomain and their interaction with the neighboring subdomains to ensure displacement and stress field continuity.
From the ordered expression:

$$\begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & Kbb \end{bmatrix} \begin{bmatrix} u_{ii} \\ u_{bb} \end{bmatrix} = \begin{bmatrix} f_i i \\ f_{bb} \end{bmatrix} \tag{1.5}$$

If we elaborate on the first line, we obtain the following result:

$$u_i = K_{ii}^{-1}(f_i - K_{ib}u_b) \tag{1.6}$$

Which corresponds to a Dirichlet problem with prescribed displacement. The second line is latter elaborated and the following linear system is obtained:

$$(K_{bb} - K_{bi} * K_{ii}^{-1} * K_{ib}) * u_b = f_b - K_{bi} * K_{ii}^{-1} * f_i \tag{1.7}$$

This is called a Primal Schur approach, and it eliminates internal degrees of freedom. The operators per subdomain are then defined as:

1. The Primal Schur complement per subdomain S_p: (which morphs boundary displacement onto boundary forces), it is a local, to subdomain, Dirichlet operator.Which is calculated the following way :
$$S_p = K_{bb} - K_{bi} * K_{ii}^{-1} * K_{ib} \tag{1.8}$$

2. Primal right hand side
$$b_p = f_b - K_{bi} * K_{ii}^{-1} * f_i \tag{1.9}$$
. So the final primal Schur problem becomes:
$$S_p * u_b = b_p \tag{1.10}$$

3. Primal assembly operator A, a boolean operator which lists the number of primal relations of the interface skeleton.

And their corresponding Concatenated quantities are:

1. The concatenated Primal Schur complement:
$$\begin{bmatrix} S_p^1 & \dots & 0 \\ \vdots & S_p^{n-1} & \vdots \\ 0 & \dots & S_p^n \end{bmatrix} \tag{1.11}$$

2. The vertically concatenated primal right hand side:
$$\begin{bmatrix} b_p^1 \\ \vdots \\ b_p^n \end{bmatrix} \tag{1.12}$$

3. The horizontally concatenated primal assembly operators (A).
$$[A^1 ... A^n] \tag{1.13}$$

However, a dual approach can be analyzed as well. In which the dual quantity of stress is mainly analyzed. The new operations per subdomain that need to be calculated to solve this problem are now are:

1. The Kernel of S_p, Since S_p is not invertible it has a kernel which corresponds to the rigid body modes (on almost all cases, on this simple 1D element the fist element corresponding to the fixed end had a null kernel).

2. The Dual Schur complement per subdomain S_d: (which maps interface forces with interface displacements), since S_p is not invertible, it is calculated per subdomain using the pseudoinverse of S_p.
$$S_d = S_p^+ \tag{1.14}$$

3. Dual assembly operator $\underline{\mathrm{A}}$, which contrary to A, is a signed boolean operator, and it lists the number of dual relations for the interface skeleton (which for this 1D case equals the number of primal relations).

As it was done with the subdomain quantities on the primal approach, the quantities for the dual approach are concatenated in the following way:

1. The concatenated Dual Schur complement:
$$\begin{bmatrix} S_d^1 & \dots & 0 \\ \vdots & S_d^{n-1} & \vdots \\ 0 & \dots & S_d^n \end{bmatrix} \tag{1.15}$$

2. The concatenated dual right hand side:
$$b_d^{concatenated} = S_d^{concatenated} * b_p^{concatenated} \tag{1.16}$$

3. The horizontally concatenated dual assembly operators ($\underline{\mathrm{A}}$).
$$[\underline{A}^1 ... \underline{A}^n] \tag{1.17}$$

4. The diagonally concatenated Rigid Body Modes

These operators are latter used to defined new quantities for both, the primal and dual methods, which will be described in detail in the following sections.

Chapter 2

Scalability Study

This section will dive into the different methods to solve the system through domain decomposition using primal, dual, and mixed approaches. For both primal and dual approaches, both direct and iterative methods will be analyzed to solve the final linear systems, while the Latin method will be analyzed for the the mixed approach. After each method, the numerical scalability of each one will be studied.

2.1 Primal Approach

2.1.1 Solution by Direct Method

As introduced on the previous chapter, through the primal Schur method we obtain a linear system in which the internal degrees of freedom are eliminated. For each subdomain we obtain the following: $S_p * u_b = b_p$, and as mentioned each quantity can be concatenated. It is clear to see that there are many duplicated quantities, since at every interface there exists a duplicity of a nodal quantity (one from each subdomain which has the same value). To eliminate this we will introduce the concept of assembled quantities:

- $$S_p^{Assembled} = A^{concatenated} * S_p^{concatenated} * (A^{concatenated})^T \tag{2.1}$$
- $$b_p^{Assembled} = A^{concatenated} * b_p^{concatenated} \tag{2.2}$$

Which finally result in the following linear system:

$$S_p^{Assembled} * u_b = b_p^{Assembled} \tag{2.3}$$

u_b can latter be used on each subdomain to calculate the internal degrees of freedom u_i through the equation 1.6.

The solution of both boundary and internal degrees of freedom can be shown in Figure 2.1. Where the u_b terms can be seen on red and the u_i terms on green. It can be seen that there is a displacement continuity condition enforced between the subdomains, and that this linear solution matches with both the classical FEM and the analytical approach. Fors this computation **20 Subdomains** and **10 Elements** per subdomain were used.

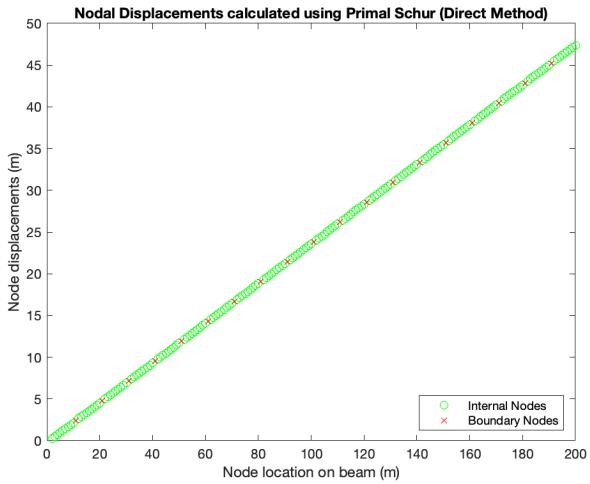
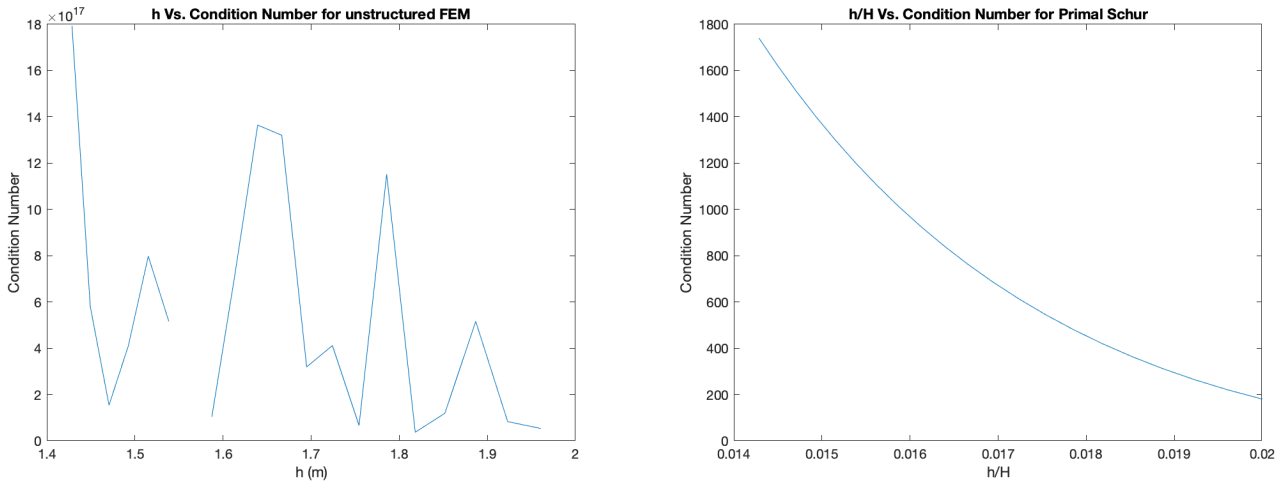


Figure 2.1: Boundary and Internal Displacement solutions using the Priam Schur method

2.1.2 Conditioning of Direct Method

The condition number of a matrix is a measure of how sensitive the solution of a linear system involving that matrix is to errors in the input data. It is defined as the ratio of the largest singular value of that matrix to the smallest singular value. If a matrix has a high condition number it is said to be ill conditioned and the solution will tend to be more unstable. The evolution of the condition number can be seen in the following figure. This figure has two plots, the first one shows how the condition number of the original stiffness matrix changes with respect to the element size Figure 2.2b, and the second one shows how the conditioning of the S_p matrix changes with respect to the h/H relationship (Figure 2.2a). It can be seen that both tend to increase the closer you get to zero. However, compared to the conditioning of the unstructured FEM stiffness matrix, the increase is smoother and more predictable.



(a) h vs. K

(b) h/H vs. S_p

Figure 2.2: Scaling parameters vs. Condition numbers κ for FEM (a) and Direct Primal Schur (b)

2.1.3 Solution by Iterative Method: Conjugate Gradient

Since it is not practical to always solve the linear system by a direct approach, iterative methods are considered to solve it instead. This conjugate gradient method belongs to a Krylov method, which

get their name due to the iterative construction of Krylov subspaces κ. The principle of these solvers for a linear system Ax = b at each iteration is the following:

$$x_{(i)} \in x_0 + \kappa(A, r_0) \tag{2.4}$$

$$r_{(i)} \perp \kappa(A, r_0) \tag{2.5}$$

The nodal displacement results using this method can be observed on Figure 2.3. It can be observed that it matches the expected linear response of the system.

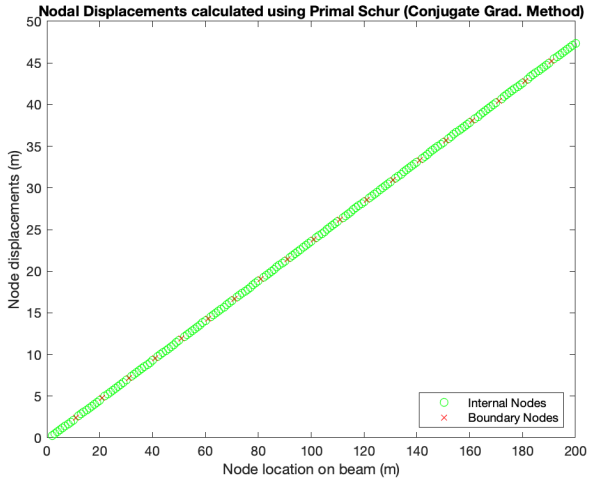
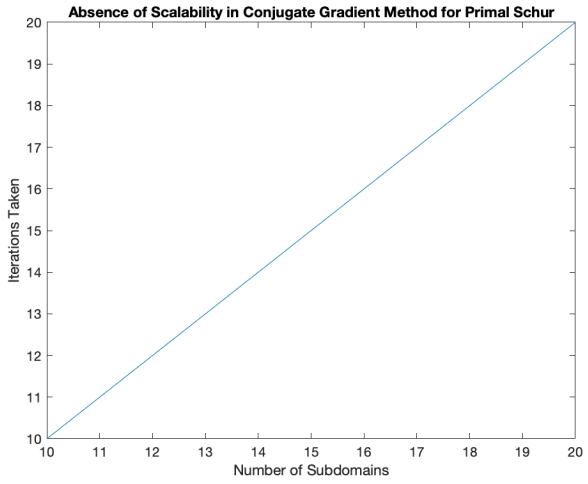


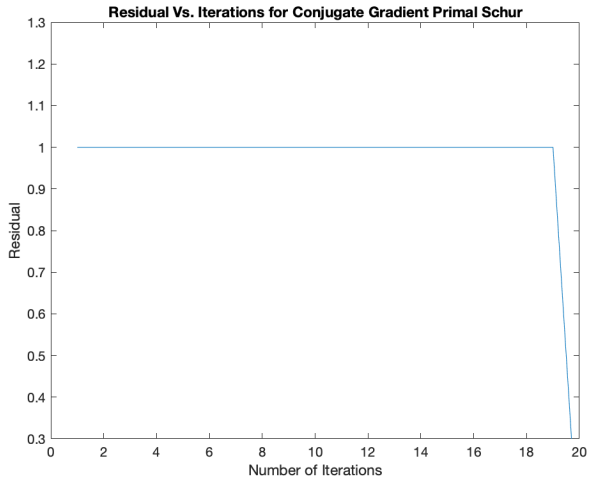
Figure 2.3: Primal Conjugate Gradient Solutions

2.1.4 Scalability of the solution

The solution using a conjugate gradient method is not numerically scalable since the number of iterations are linearly dependent on the number of subdomains as seen in Figure 2.4a. The numerical unscalablitiy is also observed from the residual, which remains constant (equal to the normalized residual at the first iteration), right up until the last iterations (Figure 2.4b



(a) No. of sub-Domains vs iterations



(b) Iterations vs. residual

Figure 2.4: Scalability of conjugate gradient method for Primal Schur

2.1.5 Solution by Iterative Method: Preconditioned Conjugate Gradient

To cope with the non-scalability of the of the conjugate gradient method, a preconditioned will be added. Usually this consists on solving the following system instead:

$$M^{-1} * A * x = M^{-1} * b \tag{2.6}$$

Where M is generally a symmetric, positive definite matrix that approximates A such that the conditioning of $M^{-1} * A$ is better than A. In this case a Neumann preconditioner will be employed, definied in the following way:

$$\tilde{S_p}^{-1} = \tilde{A} * S_d^{concatenated} * \tilde{A}^T \tag{2.7}$$

$$\tilde{A} = (A^{concatenated} * (A^{concatenated})^T)^{-1} * A^{concatenated} \tag{2.8}$$

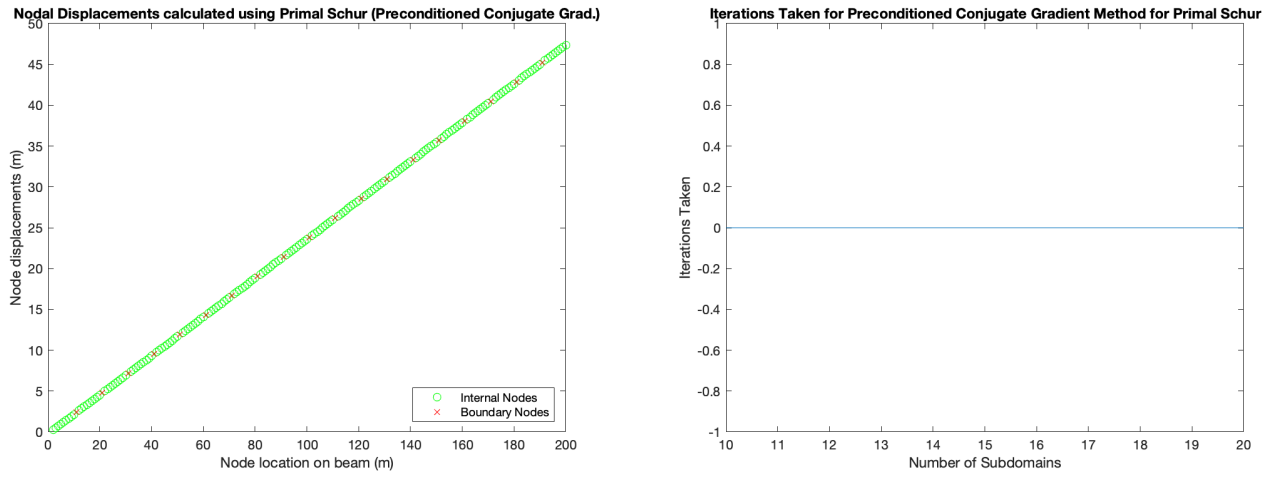
However instead of simply applying the conjugate gradient algorithm to this system, the following projected problem will be introduced instead:

$$(P^T * S_p * P) * u_b^* = P_T * (b_p - S_p * u_b^0) = P^T * b_p \tag{2.9}$$

Where:

- $u_b = u_b^0 + P * u_b^*$
- $u_b^0 = \tilde{G} * (\tilde{G}^T * S_p * \tilde{G})^{-1} * \tilde{G}^T * b_p$
- $P = I - \tilde{G} * (\tilde{G}^T * S_p * \tilde{G})^{-1} * \tilde{G}^T * S_p$
- $\tilde{G} = \tilde{A} * R^{concatenated}$

The solution of the system using this method can be observed on Figure 2.5a, and it can be noted that it matches the expected linear response.



(a) Displacement Solutions

(b) No. of Sub-Domains vs. Iterations

Figure 2.5: Preconditoned Primal Solution (a) and Iterations taken (b)

Using the preconditioner as $\tilde{S_p^{-1}}$, which is the Neumann preconditioner, the conjugate gradient becomes parallel scalable (the number of iterations are independent from the number of sub-domains). It is worth noting that with a tolerance of $1 * 10^{-4}$ for the 2-norm of the residual, the initial guess (u_b^0) reaches convergence.

2.1.6 Conditioning of the preconditioned system

We were expecting the conditioning of the multiplication $\tilde{S_p^{-1}} * S_p$ to be lower to the original conditioning of S_p, however the observed results are not as expected.

- conditioning of $S_p = 760$
- conditioning of $\tilde{S_p^{-1}} * S_p = 5.0872\text{e+}05$

2.2 Dual Approach

As previously discussed, the discretized system can be solved through domain decomposition using a dual approach. With the stress being the dual quantity of interest. On this approach the constraint of displacement continuity is taken into account through the use of a Lagrange Multiplier. The problem can thus be written as the stationarity of the following Lagrangian:

$$J_u(u_E) = \int_\Omega \epsilon : K : \epsilon d\Omega - \int_\Omega f_d * u_E d\Omega - \int_{\delta\Omega} F_d * u_E d\Gamma \tag{2.10}$$

$$J_{u,F}(e_E, W_E) = \sum_E J_u - \sum_{(E,E')} \int_{\Gamma_{(E,E')}} \lambda_{(E,E')} * (W_{(E,E')} - W_{(E',E)}) d\Gamma \tag{2.11}$$

Where equation 2.10 corresponds to the classical potential energy potential, and equation 2.11 includes the added Lagrange multiplier. The solution, after discretization has the following form.

$$\begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & Kbb \end{bmatrix} \begin{bmatrix} u_{ii} \\ u_{bb} \end{bmatrix} = \begin{bmatrix} f_i i \\ f_{bb} \end{bmatrix} + \begin{bmatrix} \lambda_i (= 0) \\ \lambda_b \end{bmatrix} \tag{2.12}$$

$$R^T(f + \lambda) = 0 \tag{2.13}$$

Where R on equation 2.13 contains the discretized rigid body modes of each subdomain. Now, similar to the procedure done on the primal approach, the internal degrees of freedom are eliminated, which results in the following system:

$$S_p^{concatenated} * u_b^{concatenated} = b_p^{concatenated} + \lambda_b^{concatenated} \tag{2.14}$$

$$(R^{concatenated})^T (b_p^{concatenated} + \lambda_b^{concatenated}) = 0 \tag{2.15}$$

$$\lambda_b^{concatenated} = (\underline{A}^{concatenated})^T * \lambda_b \tag{2.16}$$

To calculate u_b from the system, one must calculate the inverse of S_p, however, as it was mentioned, the inverse of S_p is not defined and one must calculate the pseudo-inverse "S_p^+" instead. Nevertheless, the solution will be known up to the rigid body modes.

$$u_b = S_p^+ * (b_p + \lambda_b) + (R_b * \alpha_b)_{rigid-body-motion} \tag{2.17}$$

The Dual Schur component S_d is then defined as S_p^+. And it maps interface forces into interface displacements. This former equation is defined per subdomain, and must be concatenated, and latter assembled to obtain the following linear system:

$$\begin{bmatrix} S_d & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \lambda_b \\ \alpha_b \end{bmatrix} = \begin{bmatrix} -b_d \\ -e \end{bmatrix} \tag{2.18}$$

The Assembled quantities used are the following:

- $S_d = \underline{A}^{concatenated} * S_d^{concatenated} * (\underline{A}^{concatenated})^T$
- $b_d = S_p^{concatenated} * b_p^{concatenated}$
- $G = \underline{A}^{concatenated} * R^{concatenated}$
- e $= (R^{concatenated})^T * b_p^{concatenated}$

2.2.1 Solution by Direct Method

The steps to obtain the solution are the following:

- Directly solve the linear system to obtain the parameters α and λ.
- Solve for u_b using equation 2.17.
- Solve for u_i using equation 1.6.

The final solutions for u_b and u_i are plotted in Figure 2.6. And it can be observed that the obtained solution matches the previously obtained linear response.

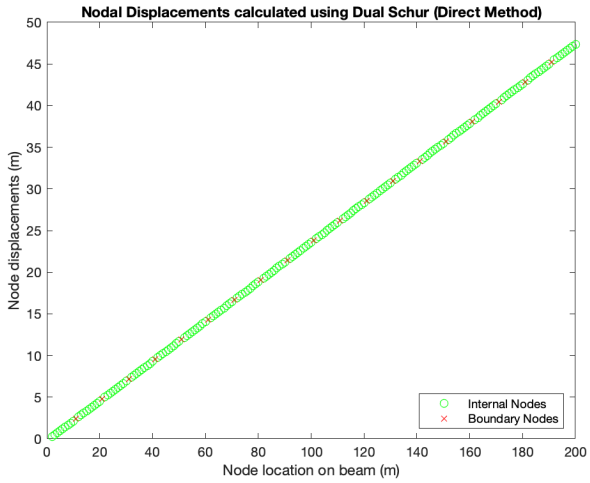


Figure 2.6: Dual Direct Solution

2.2.2 Solution by Iterative Method: FETI method

However, the dual linear system is not solved directly in practice. This time, the problem is solved by a method of distributed projected conjugate gradient to take into account the condition of edge forces balanced on each substructure. This leads to the Finite Element Tearing and Interconnecting method, "FETI" for short. This method makes use of a Dirichlet preconditioner, which is defined as a low cost approximation of the inverse of S_d:

$$\tilde{S_d}^{-1} = \underline{\tilde{A}} * S_p^{concatenated} * \underline{\tilde{A}}^T \tag{2.19}$$

$$\underline{\tilde{A}} = (\underline{A}^{concatenated} * (\underline{A}^{concatenated})^T)^+ * \underline{A}^{concatenated} \tag{2.20}$$

This preconditioner is latter applied to the following projected system:

$$(P^T * S_d * P) * \lambda_b^* = P^T * (-b_d - S_d * \lambda_b^o) \tag{2.21}$$

Where (for homogeneous structures):

- $\lambda_b = \lambda_b^0 + P * \lambda_b^*$
- λ_b^0 = -G*$(G^T * G)^{-1}$*$\mathbf{e}$
- $\mathbf{P} = \mathbf{I}$ - G*$(G^T * G)^{-1} * G^T$

Where λ_0 is the initial guess to the solution and this projector P can be understood as the search of rigid body modes connections on each subdomain that minimize the displacement jump at the interfaces. The solution of this approach can be seen on Figure 2.7, and it can be seen that the results match with the previously obtained linear response. An interesting observation is that for this method, we obtain instant convergence using the proposed initial guess for λ_b^0 (0 iterations just like Figure 2.5a) and a tolerance of $1 * 10^{-4}$ for the 2-norm of the residual.
To calculate the pseudoinverse the "pinv" function in matlab was used, and to calculate the rigid body modes per subdomain, the "null" function was used to obtain the kernel of each S_p. It is worth mentioning that the pseudoinverse is not unique, and this the results from S_d might vary.

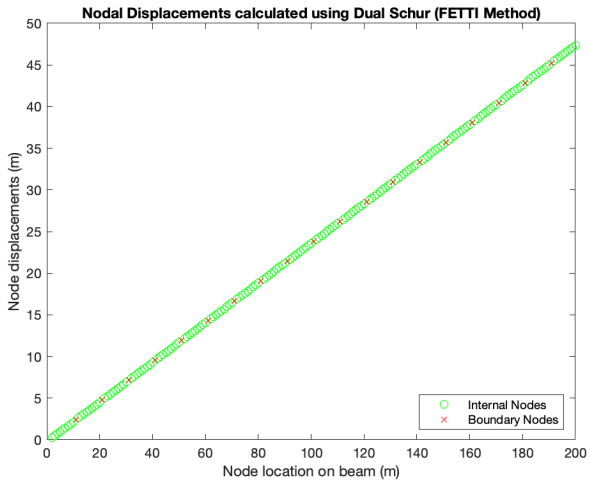


Figure 2.7: Dual Direct Solution using FETI Method

2.2.3 Rigid Body Modes

To obtain the rigid body modes of the substructures at convergence one must look at the equation 2.17. The Rigid body mode part is labeled as the multiplication $R_b * \alpha_b$, in which R_b is the kernel of S_p of each subdomain, and α_b is obtained as a post-processing stage of the FETI method using the following equation:

$$\alpha_b^{concatenated} = (G^T * G)^T * (-b_d - S_d * \lambda_{(m)} \tag{2.22}$$

The final result is shown in Figure 2.8, and we can observe that the internal subdomains have 2 rigid body modes (since they 2 degrees of freedom), the first one has no rigid body modes (that's why the first displacement is equal to 0), and the last element has only one rigid body mode.

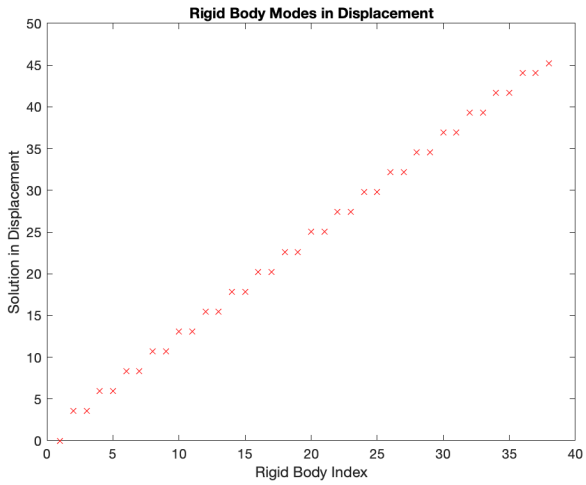


Figure 2.8: Rigid body solutions in displacement

2.3 Mixed Approach

The previous two approaches, primal and dual have one type of variable as main unknown. The Primal approach has boundary/interface displacements, and the the dual approach has Lagrange multipliers. The mixed approach proposes more unknown variables at the interfaces which could be seen as a drawback, however since we consider force and displacement at the interface, more complex behaviors can be analyzed. However, for our simple 1D case, the benefits are obviously not exploited. There are various methods to solve this mixed approach, but we will focus on the monoscale LaTin approach.

2.3.1 Mixed Monoscale LaTin approach

This method consists in separation of the equations into two manifolds:

- A linear (and possibly global) set of equations
- A local (and possibly nonlinear) set of equations.

The second stage of the LaTin method is an iterative scheme separated into two stages: a local and a global stage. An interesting characteristic of the LaTin method is that each iteration is performed on the whole domain, which implies that data storage requirements are high (depending on the complexity of the model) if no model reduction techniques are employed.
The obtained solutions from the Latin Method can be observed on Figure 2.9 and it can be observed that it has the linear response proper to this system.

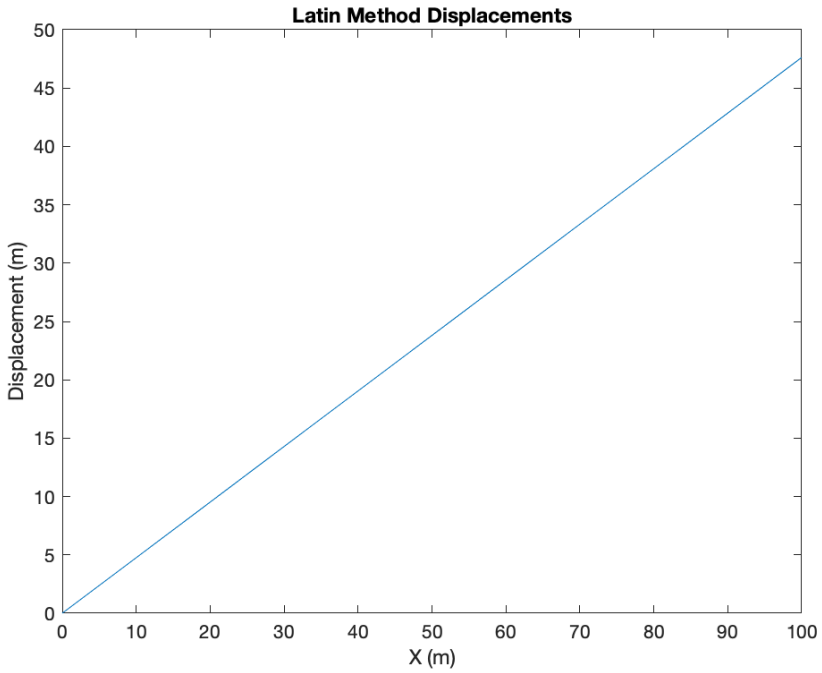
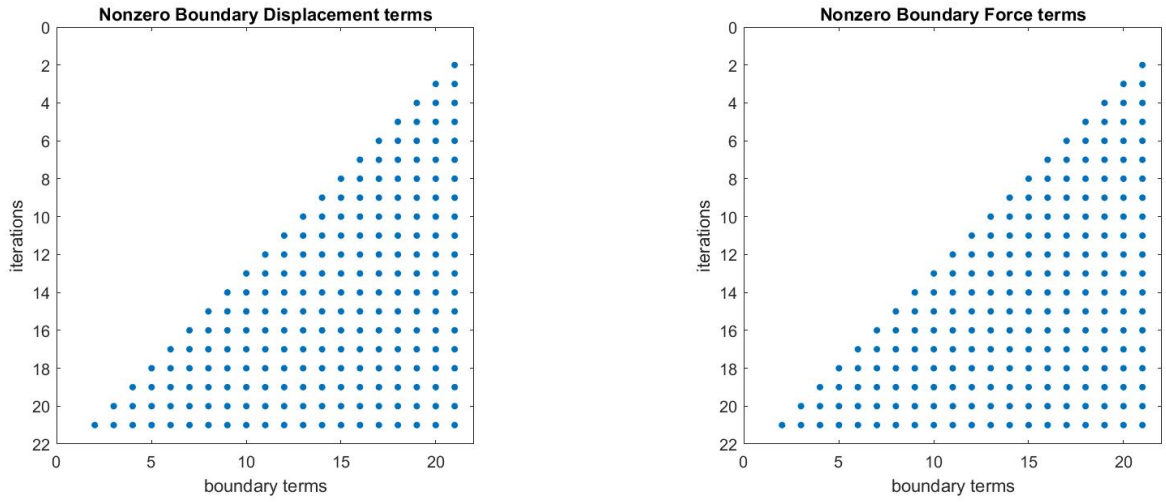


Figure 2.9: Nodal Displacements from Latin Method

We can also analyze the scalability of this method. The monoscale LaTin method is not numerically scalable, since the iterations needed to converge equals the number of subdomains, similar to the conjugate gradient method seen for the primal approach. On Figure 2.10 each row shows the nonzero terms obtained at each iteration for both diaplacement and force at the interfaces. We can observe how the information truly is "propagating" from subdomain to subdomain on each iteration starting from the subdomain where the force is imposed.



(a) Propagation of Interface Displacement Information (b) Propagation of Interface Force Information

Figure 2.10: Propagation of Interface Information Latin Method

Chapter 3

Conclusions and Comments

3.1 Conclusions

- The classical finite element approach results in a linear system of the form Au=b. However, this can become a big system depending on the complexity and refinement of the simulation. Because of this it is ideal to find ways to optimize the solution, and domain decomposition based on the physics of the model is an interesting approach.
- Domain decomposition of an original FEM model can be based on primal, dual, or mixed quantities. Which result in smaller local linear systems.
- The direct solution of any of the previously mentioned methods is undesirable, and because of this, iterative solutions are studied to solve linear systems.
- From the iterative solutions, the conjugate gradient applied on the primal problem is not numerically scalable. So an interesting alternative is to introduce a preconditioner, this results in a system that reaches convergence on the first iterations. For our case on iteration 0.
- The FETI method also involves a preconditioner and the observed behavior matched the one observed for the primal method with a preconditioner.
- The monoscale LaTin method is not numerically scalable, due to this coarse scale problem resolutions used to enhance the method.