AN ALGORITHM FOR INTERFACING
NONMATCHING FEM MESHES

ADVANCED FINITE ELEMENT METHOD
Curse: 2006

Directed by: Prof. Carlos A Felippa
Done by: Luis Rodríguez-Tembleque Solano
An algorithm for interfacing non-matching fem meshes

Contents

1 Introduction ................................................................. 3
2 Localize Lagrange Multipliers (LLM) formulation ............. 5
3 Interfacing non-matching meshes ................................. 10
4 Computation of the frame nodal location ........................ 11
  4.1 Virtual work equation set ........................................ 11
  4.2 Zero Moment Rule (ZMR) ......................................... 13
  4.3 Equivalent ZMR in 3D ............................................. 14
5 Proposed algorithm for computation of the frame nodal
   location ................................................................. 18
6 Application on 2D problems. Examples .......................... 22
7 Application on 3D problems. Examples .......................... 25
8 Conclusions and Possible Future Work .......................... 30
  References ................................................................... 31
  Appendix1 .................................................................. 32
  Appendix2 .................................................................. 35
  Appendix3 .................................................................. 37
1. Introduction

The problem of interfacing two meshes of finite elements or boundary elements through a connection frame is a problem of large applicability in several mechanical problems, even more if the solid meshes are non-matching. Examples of that are found on structure-structure or fluid-structure interaction, and in contact problems. In the first cases, the software that generates the meshes usually is not the same, so the meshes of our sub-domains will be non-matching. In contact problems, even having matching meshes, as the materials of the bodies in contact have different properties, our meshes will be non-matching after the deformation. So we can see that the problem of coupling meshes is an enormous interest problem.

The Coupling between the meshes is done through a displacement frame interposed between the interface meshes of each solid. That frame is treated with a finite element discretization and connected to the meshes through localize Lagrange multipliers, using the variational-based formulation proposed by Park and Felippa [1-3]. This methodology allow us to coupling meshes of arbitrary geometry, discretization type (e.g., FEM and BEM) and even meshes of different physics (e.g., structure and fluids). The “key” of this problem is the discretization of the interface frame. The location of the frame nodes and the displacement interpolation, have to pass the interface patch test (IPT), which means a constant stress state have to be preserved through the interface. So us we know the interacting forces on each frame face are acting, the IPT can be applied by computing the virtual work of those forces on each frame patch function and requiring that it vanish.
The idea stated above, have some difficulties when is applied on 3D problems. On 2D we can find analytically the node location using the virtual work equations and the requirement of constant stress preservation, or the zero moment rule (ZMR) developed by Park, Felippa and Rebel [4], but on 3D there is still no general analytical procedure that allows us to find the nodes location. Only when we have rectangular interfaces, meshed in a mapped way, we have an equivalent ZMR for 3D problems.

In this project, a general algorithm for interfacing non-matching 2D-3D meshes based on Park and Rebel work [5] is presented. The idea is to find the nodes location minimizing one function in which we have the contribution of each node of the frame to the virtual work, subject to some restrictions. This methodology will be applied on four simple 2D-3D cases, and the results will be compared with analytical results in most of the cases.

Figure 1.
2. **Localize Lagrange Multipliers (LLM) formulation**

Consider two sub-domains, $\Omega_1$ and $\Omega_2$, connecting by an interface or frame $\Gamma_c$ (see Fig1.). The virtual work of the total system will be the sum of the contribution of the completely free substructures $\delta \Pi_1$ and $\delta \Pi_2$, plus $\delta \Pi_c$ the virtual work over the interface $\delta \Pi_c$:

$$\delta \Pi^{total} = \delta \Pi_1 + \delta \Pi_2 + \delta \Pi_c$$  \hspace{1cm} (1)

The expression which takes the virtual work over each region could be:

$$\delta \Pi_1 = \delta u^T \left\{ K \, u - f \right\} \quad \text{or} \quad \delta \Pi_1 = \delta p^T \left\{ H \, u - G \, p - b \right\}$$  \hspace{1cm} (2)

and

$$\delta \Pi_2 = \delta \bar{u}^T \left\{ \bar{K} \, \bar{u} - \bar{f} \right\} \quad \text{or} \quad \delta \Pi_2 = \delta \bar{p}^T \left\{ \bar{H} \, \bar{u} - \bar{G} \, \bar{p} - \bar{b} \right\}$$  \hspace{1cm} (3)

Depending on what technique we use to model each sub-domain: FEM or BEM. And the expression for the virtual work over the frame would be

$$\delta \Pi_c = \int_{\Gamma_c} \delta \left\{ \lambda_i (v_i - u_i) \right\} d\Gamma + \int_{\Gamma_c} \delta \left\{ \bar{\lambda}_i (v_i - \bar{u}_i) \right\} d\Gamma$$  \hspace{1cm} (4)

Being:

- $v_i$: The displacement field over the frame.
- $\lambda_i$: The traction field over the solid 1 on $\Gamma_c$.
- $u_i$: The displacement field over the solid 1 on $\Gamma_c$.
- $\bar{\lambda}_i$: The traction field over the solid 2 on $\Gamma_c$.
- $\bar{u}_i$: The displacement field over the solid 2 on $\Gamma_c$. 

---

*Advances Finite Element method*
The equation (4) enforces the kinematical positioning of the frame in a weak sense.

![Figure 2](image)

Each field in the equation (4) is approximated in the following way:

\[ v_i = N_i(\xi) \cdot \mathbf{v}_i \]
\[ u_i = N_i(\xi) \cdot \mathbf{u}_i \]
\[ \bar{u}_i = N_i(\xi) \cdot \mathbf{u}_i \] (5)

\[ \lambda_i = \delta(\xi - \xi_p) \cdot \lambda_p \] (6)

with \( \xi = (\xi_1, \xi_2) \) and \( \xi_p \) where \( \xi \) is the frame coordinates of the sub-structure over the frame. The interface forces are point loads collocated at the boundary nodes (see Fig2.), which mathematically are localized Lagrange multipliers delta functions in one to one correspondence with the partition-node degrees of freedom.

Using Dirac's delta functions for approximating the tractions field, the integrals on expression (4) becomes
An algorithm for interfacing non-matching fem meshes

\[
C_p = \int_{\Gamma} \delta(\xi - \xi_p) N_v \, d\Gamma = N_v(\xi_p)
\]
\[
\overline{C}_p = \int_{\Gamma} \delta(\xi - \overline{\xi}_p) N_v \, d\Gamma = N_v(\overline{\xi}_p)
\] (7)

which allow us to write the equation (4) in the following matricial way:

\[
\delta \Pi_c = \delta \left\{ \lambda^T (C_b \, v - B^T \, u) \right\} + \delta \left\{ \overline{\lambda}^T (\overline{C}_b \, v - \overline{B}^T \, \overline{u}) \right\}
\] (8)

where

\[
B = \sum_{p=1}^{np} L_{up}^T B_p L_p \lambda
\] (9)

\(L[ ]_p\) is the Boolean finite element operator who extract the variable associated with the contact interface node \(p\) from the global \([\ ]\) unknowns vector. The 3x3 \(B_p\) matrix establishes the relation between the components of the field on the node \(p\) expressed using the frame system and the global system.

Now we know all the expressions on the equation (1), so carrying out the variations:

\[
\delta \Pi^{total} = \delta u^T \left\{ K \, u - f - B \lambda \right\} + \delta \overline{u}^T \left\{ \overline{K} \, u - \overline{f} - \overline{B} \overline{\lambda} \right\} + \delta v \left\{ C^T_b \, \lambda + \overline{C}^T_b \, \overline{\lambda} \right\}
\]
\[
\delta \lambda^T \left\{ C_b \, v - B^T \, u \right\} + \delta \overline{\lambda}^T \left\{ \overline{C}_b \, v - \overline{B}^T \, \overline{u} \right\}
\] (10)

The stationary point of the virtual work expression leads to the following equilibrium equation set:
An algorithm for interfacing non-matching fem meshes

Using the BEM for modeling the two sub-structures, the equation (10) take the following appearance:

\[
\begin{pmatrix}
K & 0 & B & 0 & 0 \\
0 & \overline{K} & 0 & \overline{B} & 0 \\
B^T & 0 & 0 & 0 & C_b \\
0 & \overline{B}^T & 0 & 0 & \overline{C}_b \\
0 & 0 & C_b^T & \overline{C}_b & 0
\end{pmatrix}
\begin{pmatrix}
u \\
\overline{u} \\
\lambda \\
\overline{\lambda} \\
v
\end{pmatrix}
= \begin{pmatrix}
f \\
\overline{f} \\
0 \\
0 \\
0
\end{pmatrix}
\quad \quad (11)
\]

where we have "lumped" the tractions over each interface in the coupling region using the expressions:

\[
\delta \Pi^{\text{total}} = \delta p^T \{ H \ u - G \ p - b \} + \delta \overline{p}^T \{ \overline{H} \ \overline{u} - \overline{G} \ \overline{p} - \overline{b} \} + \delta v \{ C_b \ \lambda + \overline{C}_b \ \overline{\lambda} \}
\]

\[
\delta u_c \{ M \ p - D \lambda \} + \delta \overline{u_c} \{ \overline{M} \ p - \overline{D} \overline{\lambda} \} +
\]

\[
\delta \lambda \{ C_b \ v - B^T \ u \} + \delta \overline{\lambda} \{ \overline{C}_b \ v - \overline{B}^T \ \overline{u} \}
\]

\quad \quad (12)

where we have “lumped” the tractions over each interface in the coupling region using the expressions:

\[
\delta \Pi^{\text{lump1}} = \delta u_c^T \{ M \ p - E \overline{\lambda} \}
\]

\[
\delta \Pi^{\text{lump2}} = \delta \overline{u_c}^T \{ \overline{M} \ \overline{p} - \overline{E} \overline{\lambda} \}
\]

\quad \quad (13)

\[
E = \sum_{k=1}^{np} L_{u,k}^T B_k L_{\lambda,k}
\]

and the assembled mass matrix:

\[
M = \sum_{k=1}^{np} \int_{\Gamma_k} N_i N_j d\Gamma \quad ; \quad M_k = \int_{\Gamma_k} N_i N_j d\Gamma
\quad \quad (14)
\]

Again, carrying out the stationary of the virtual work expression (12), we obtain the following equilibrium equation set:
An algorithm for interfacing non-matching fem meshes

\[
\begin{bmatrix}
H & 0 & G & 0 & 0 & 0 \\
0 & \overline{H} & 0 & \overline{G} & 0 & 0 \\
0 & 0 & M & 0 & -E & 0 \\
0 & 0 & 0 & \overline{M} & 0 & -\overline{E} \\
B^T & 0 & 0 & 0 & 0 & C_b \\
0 & \overline{B}^T & 0 & 0 & 0 & \overline{C}_b \\
0 & 0 & C_b^T & \overline{C}_b^T & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u \\
\bar{u} \\
p \\
\bar{p} \\
\lambda \\
\bar{\lambda} \\
v
\end{bmatrix} = 0
\] (15)

The localize Lagrange Multipliers can be condensed statically, in the expression (15), what takes us to the following expression:

\[
\begin{bmatrix}
H & 0 & G & 0 & 0 \\
0 & \overline{H} & 0 & \overline{G} & 0 \\
B^T & 0 & 0 & 0 & C_b \\
0 & \overline{B}^T & 0 & 0 & \overline{C}_b \\
0 & 0 & C_b^T & \overline{C}_b^T & 0
\end{bmatrix}
\begin{bmatrix}
u \\
\bar{u} \\
p \\
\bar{p} \\
v
\end{bmatrix} = 0
\] (16)

We can have also a mixed formulation FEM-BEM:

\[
\begin{bmatrix}
K & 0 & B & 0 & 0 \\
0 & \overline{K} & 0 & \overline{G} & 0 \\
B^T & 0 & 0 & 0 & C_b \\
0 & \overline{B}^T & 0 & 0 & \overline{C}_b \\
0 & 0 & C_b^T & \overline{C}_b^T & 0
\end{bmatrix}
\begin{bmatrix}
u \\
\bar{u} \\
p \\
\bar{p} \\
v
\end{bmatrix} = 0
\] (17)

The equations (11), (16) or (17) allow us to coupling two sub-domains modeled with FE or BE, using a collocation technique. But the problem is not closed. We have to satisfy one requirement that we'll see in the following sections.
3. Interfacing non-matching meshes

The formulation for coupling non-matching meshes is only possible if we satisfy the last set of equations on the expressions (11), (16) or (17), what means that the virtual work of the forces coming from each domain, over the virtual displacements of the nodes of the frame, have to be cero (18).

\[ \delta v^T \left[ C_b^+ \lambda + \overline{C_b^+} \bar{\lambda} \right] = 0 \]  

(18)

This condition gives rise to the interface patch test (IPT) which provides the frame configuration. The discretization of the frame require selecting the location of the frame nodes and the displacement interpolation so that the IPT is passed when the interacting meshes are subject to arbitrary states of constant stress. Those stress state produce interaction forces on each frame face. The IPT can be applied by computing the virtual work of those interface forces on the virtual displacements of the nodes of the frame and requiring that it vanish.
4. Computation of the frame nodal location

Let’s consider the 2D problem presented on Fig.4., in which two substructures are under a compression state. As we know the forces coming from each body we can find the nodal location frame using: the virtual work equation or the ZMR.

![Diagram showing two substructures under compression with forces applied at nodes A, B, C, D.]

4.1. Virtual work equation set

Selecting the frame nodes configuration that we have on Fig.3, the equation (18) takes the following form:

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & N_{2d}(\xi_{p_2}) & 0 \\
0 & N_{2d}(\xi_{p_2}) & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1/2 \\
1 \\
1/2
\end{bmatrix}
\begin{bmatrix}
1 & N_{5d}(\xi_{p_2}) & 0 & 0 \\
0 & N_{5d}(\xi_{p_2}) & 0 & 0 \\
0 & 0 & N_{6d}(\xi_{p_6}) & 0 \\
0 & 0 & 0 & N_{6d}(\xi_{p_6})
\end{bmatrix}
\begin{bmatrix}
1/3 \\
2/3 \\
2/3 \\
1/3
\end{bmatrix}
= 0
\]

(19)
where our unknowns are the shape frame functions evaluated on the sub-structural nodes’ projection. Adding the shape function properties (19), we have a linear dependent equation that can be solved selecting then appropriate equations.

\[
\begin{align*}
N_{2c}(\xi_{p_i}) &= 1 - N_{2b}(\xi_{p_i}) \\
N_{5b}(\xi_{p_i}) &= 1 - N_{5a}(\xi_{p_i}) \\
N_{6d}(\xi_{p_i}) &= 1 - N_{6c}(\xi_{p_i})
\end{align*}
\] (20)

One time we solve the equation set, we can built the frame using one of the following equations:

\[
\begin{align*}
N_{id}(\xi_{p_i}) &= \frac{x_B - x_{p_i}}{x_B - x_A} \\
N_{ib}(\xi_{p_i}) &= \frac{x_{p_i} - x_A}{x_B - x_A}
\end{align*}
\] (21)

being:
- \(x_A\): Coordinate of the first node of the element frame.
- \(x_B\): Coordinate of the second node of the element frame.
- \(x_{p_i}\): Coordinate of the sub-domain node projected over the frame \(\Gamma_c\).
- \(N_{id}(\xi_{p_i})\): First Frame element Shape function evaluated on \(x_{p_i}\).
- \(N_{ib}(\xi_{p_i})\): Second Frame element Shape function evaluated on \(x_{p_i}\).

What we want to know is the nodes position \(x_A\) or \(x_B\). But we only can use one of the equations (21). So we can fix one of the boundaries of the frame, and build the frame from this one.

The problem with this methodology is that require too much attention from the user, because of the linear dependence of our equation set. Also require a
initial configuration of our frame in which project the nodes of each sub-domain. So if our initial configuration is wrong, we’ll not be able to reach the solution.

4.2. Zeros Moment Rule (ZMR)

Fortunately a simple and effective rule can be invoked if the frame displacements interpolation is *piecewise linear*. First we compute, on each partition, the nodal forces that come from the constant stress stay. Then the nodal forces are transferred onto the frame as illustrate in Figure 4. In that situation, we consider the frame line as an isolated object subject to this system of forces, and draw the free-body diagram (FBD) shown in Figure 5(a).

![Figure 4. Placing frame for the benchmark problem.](image-url)
Frame points are located by the coordinate $x$ which varies from $x = 0$ to $x = L$ (interface length). The 11 normal point forces acting on the frame FBD of Figure 6(a) are called $f_{ci}^m$ and act at $x_i^m = 1$, where $i = 1...6$ for partition $m = 1$, and $i = 1...6$ for partition $m = 2$. So we define the moment function as:

$$ M(x) = \sum_{i,m} f_{ci}^m \mathcal{R}(x - x_i^m), \quad \text{where} \quad \mathcal{R}(x-a) = \begin{cases} 0 & \text{if } x < a \\ x-a & \text{if } x \geq a \end{cases} \quad (22) $$

where $\mathcal{R}(x-a)$ is McAuley’s ramp function. The zero moment rule (ZMR) states that for preservation of constant-stress states, frame node location must be located at the roots of $M(x)$.

![Diagram](image)

(a)

![Diagram](image)

(b)

Figure 5. Placing connection frame nodes for the problem of Figure 4.

### 4.3. Equivalent ZMR in 3D

Some special 3D problems in which we have rectangular interfaces meshed in a mapped way, allow us to use an equivalent ZMR for computing the frame nodal location. Also if we have matching boundary interfaces and such as they can be mapped to a rectangular domain, we can apply the equivalent ZMR.
This easy technique consists of summing and projecting the acting forces of each interface over the two main orthogonal directions. After that we have transformed our 3D problem in two 2D problems which analytical solution can be computed using the ZMR (see Figure 6). Each zero moment point we compute in those 2D problems it’s really a line which will intersect with the others lines, producing the mesh of our frame.

Using this technique, a couple of coupling problems have been solved. The first one (see Figure 7(a)) consists of two matching cubes of dimensions 2x2x2mm, made of steal (E =0.1d+5 MPa and v=0.3). Over the lower face 1 MPa of compression is applied. The higher face and two lateral faces have the perpendicular displacement disabled. In the second example (see Figure 7(b)) we have two cubes of different size, but same material. Over the upper faces
of both cubes, we apply 1MPa compression, and we constrain perpendicular displacement of the lower and two lateral faces of those cubes.

All the sub-domains have been modeled with boundary elements. So we have used the equation set (16) for coupling the two sub-structures. The interfaces and frame discretization are showed in figure 8 and 9, for each problem.

In the Appendix1, the displacements fields \((U_x, U_y, U_z)\) are presented. The figures show we have a linear distribution in the displacement field so the coupling has been developed successfully for these 3D cases.
An algorithm for interfacing non-matching fem meshes

Figure 8. Interfaces and frame meshes for the problem presented in figure 7(a).

Figure 9. Interfaces and frame meshes for the problem presented in figure 7(b).
5. Proposed algorithm for computation of the frame nodal location

The formulations presented for computing the nodal frame location on non-matching interfacing problems (solve the virtual work equation set or ZMR) have some limitations. Using the virtual work equation set for node location we don’t have a linear independent equation set, so we have to be very careful choosing the equation of our system. Furthermore we have to give an initial configuration of the frame, saying which nodes from each sub-structure belong to each element frame. So if our election is wrong we won’t be able to solve the equation set. All this things show that this methodology require too much attention from the user, its application on 3D problem is unviable.

![Domain 1](image1.png)

![Domain 2](image2.png)

Figure 10. Non-mapped non-matching interfaces.

The zero moment rule (ZMR), and its extension to some particular 3D problems is the more powerful tool we have for the moment to tackle the problem of computing the frame nodal location, but it’s extremely limited in 3D...
non-matching interfacing problems. If the interfaces are non-mapped like in figure 10, we need another tool for compute the location of the nodes of the frame. It’s in this context where appear the necessity of develop algorithms based on optimization (see Rebel, Park and Felippa [5]).

A general algorithm for compute nodal frame location in non-matching interfacing problems based on optimization is presented. We use the virtual equation and the IPT for computing the nodes location.

\[ \delta v^T \left\{ C_b^T \lambda + \overline{C_b}^T \overline{\lambda} \right\} = 0 \]  \hspace{1cm} (23)

If we analyze the expression (23), we can see that the contribution of each node of the frame to the total virtual work over the frame is:

\[ \delta v_i \cdot \text{Row}_i \left\{ C_b^T \lambda + \overline{C_b}^T \overline{\lambda} \right\} = 0 \]  \hspace{1cm} (24)

So our algorithm based on optimization has to consider the virtual work that comes from every frame node. The objective function for the algorithm finds the nodes location doing every nodal contribution to the virtual work null. The objective function of our problem takes the form:

\[ f(N_m) = \sum_i \left\{ \text{Row}_i \left\{ C_b^T \lambda + \overline{C_b}^T \overline{\lambda} \right\} \right\}^2 \]  \hspace{1cm} (25)

This function is the sum of the square of every row of the last set of our system of equations in the expression (11), or the sum of the square of every nodal virtual work. In this function, the variables are the frame shape functions evaluated on the projection of the nodes belonging to each sub-structure.
The restrictions to our problem are very simple. As the variables are the frame shape function, the restrictions are:

\[
0 \leq N_{in} \leq 1 \\
\sum_n N_{in} = 1 
\]  

(26)

In some cases, we also can add some symmetry conditions to our restriction set (26). So the problem can be summarized in:

\[
\text{Min: } \quad f (N_{in}) = \sum_i \left\{ \text{Row}\left( C_b^T \lambda + \overline{C_b}^T \overline{\lambda} \right) \right\}^2 \\
\text{Subject to: } \quad 0 \leq N_{in} \leq 1 \\
\sum_n N_{in} = 1 
\]  

(27)

After solving the optimization problem, we have frame shape function, now we have to compute the frame node location. In 2-D problems we use one of the following expressions depend on which node in the frame element is known:

\[
N_{ia}(\xi_{pi}) = \frac{x_B - x_{pi}}{x_B - x_A} \\
N_{ib}(\xi_{pi}) = \frac{x_{pi} - x_A}{x_B - x_A} 
\]

(27)

Normally on 2-D we know the position of the nodes on the boundaries. So using one of the previous expressions, we can build, step by step, the frame. Sometimes instead of know the position of one node, we know that have to satisfy some symmetry condition. So we have another extra equation to combine with the expression above.
On 3-D problems we have the same problematic than in 2-D, but instead of 2 unknown by element frame, we have four. So we have to solve the following kind of equations:

\[
\begin{align*}
x_{pi} &= \sum_{n=1}^{D} N_n (\xi_{pi}, \eta_{pi}) x_n \\
y_{pi} &= \sum_{n=1}^{D} N_n (\xi_{pi}, \eta_{pi}) y_n
\end{align*}
\]

As in 2-D cases we normally use to know the position of the boundary frame nodes or we can apply some symmetry condition, so can compute, step by step, the frame.
6. Application on 2D problems. Examples

The algorithm has been presented so we are going to solve some simple problem for checking its behavior.

◆ Ejemplo 1:

Consider the same problem as in figure 3. We distribute the nodes of the frame in an initial configuration, so we know on which frame element each structural node project (see figure 12). This point is very important because if we don’t choose the right initial configuration the solution won’t be satisfactory.

![Diagram](image)

Figure 12. Localize Lagrange multipliers over the frame.

<table>
<thead>
<tr>
<th>Node</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>force</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1/2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>$N_1(\xi_{p_1})$</td>
<td>$N_2(\xi_{p_1})$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1/2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>-1/3</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>$N_1(\xi_{p_1})$</td>
<td>$N_2(\xi_{p_1})$</td>
<td></td>
<td>-2/3</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>$N_1(\xi_{p_2})$</td>
<td>$N_2(\xi_{p_2})$</td>
<td>-2/3</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>-1/3</td>
</tr>
</tbody>
</table>

Table 1. Nodal frame contribution to virtual work.
The next step is to define the objective function. For doing that, we are going to make a table (see table 1) in which each column we have: nodes numeration, how each frame node weight the forces coming from each domain (columns: A-D), and the value of those forces. Adding the squares of multiplying A-D columns by force column we have:

\[
f(N) = \frac{1}{2} - \frac{1}{3} - \frac{2}{3} N_{fa}(\xi_{fa})^2
\]

\[
+ \left\{ N_{fa}(\xi_{fa}) - \frac{2}{3} N_{fb}(\xi_{fb}) \right\}^2
\]

\[
+ \left\{ N_{fb}(\xi_{fb}) - \frac{2}{3} N_{fc}(\xi_{fc}) \right\}^2
\]

\[
+ \left\{ N_{fc}(\xi_{fc}) - \frac{2}{3} N_{fd}(\xi_{fd}) \right\}^2
\]

(29)

The restrictions to the objective function are:

\[
N_{fa}(\xi_{fa}) = 1 - N_{fa}(\xi_{fa})
\]

\[
N_{fb}(\xi_{fb}) = 1 - N_{fa}(\xi_{fa})
\]

\[
N_{fc}(\xi_{fc}) = 1 - N_{fa}(\xi_{fa})
\]

\[
N_{fd}(\xi_{fd}) = 1 - N_{fa}(\xi_{fa})
\]

(30)

and the symmetry conditions are:

\[
N_{fa}(\xi_{fa}) - N_{fa}(\xi_{fa}) = 0
\]

\[
N_{fb}(\xi_{fb}) - N_{fa}(\xi_{fa}) = 0
\]

(31)

We only have to give the initial condition: \( N = 1 \), and use for example matlab optimization tools, for solving the problem. After doing that, we find the following solution:
We have found the correct solution. The node location is such as we satisfy the IPT.

\[ N_{2b}(\xi_{p_i}) = 0.5001 \]
\[ N_{2c}(\xi_{p_i}) = 0.4999 \]
\[ N_{5a}(\xi_{p_i}) = 0.2496 \]
\[ N_{5b}(\xi_{p_i}) = 0.7501 \]
\[ N_{6c}(\xi_{p_i}) = 0.7501 \]
\[ N_{6d}(\xi_{p_i}) = 0.2496 \]

\[ N_{IA}(\xi_{p_i}) = \frac{x_B - x_{p_i}}{x_B - x_A} \]
\[ N_{IB}(\xi_{p_i}) = \frac{x_{p_i} - x_A}{x_B - x_A} \]

\[ x_A = 0.00 \]
\[ x_B = 0.888 \]
\[ x_C = 1.111 \]
\[ x_D = 2.00 \]

\[ N_{2b}(\xi_{p_i}) = 0.5001 \]
\[ N_{2c}(\xi_{p_i}) = 0.4999 \]
\[ N_{5a}(\xi_{p_i}) = 0.2496 \]
\[ N_{5b}(\xi_{p_i}) = 0.7501 \]
\[ N_{6c}(\xi_{p_i}) = 0.7501 \]
\[ N_{6d}(\xi_{p_i}) = 0.2496 \]

\[ N_{IA}(\xi_{p_i}) = \frac{x_B - x_{p_i}}{x_B - x_A} \]
\[ N_{IB}(\xi_{p_i}) = \frac{x_{p_i} - x_A}{x_B - x_A} \]

Ejemplo 2:

Now we are going to solve the same problem, but the mesh of the upper partition has some eccentricity (e=0.2mm) (see figure 13(a)). Proceeding in the same way as in the first example, we can find the frame nodes location (see figure 13(b)).
7. Application on 3D problems. Examples

After checking successfully the algorithm in 2-D problems, we are going to apply the algorithm to two simple problems of non-matching coupling. In the first problem we have mapped meshes, but in the second we’ll have non-mapped meshes.

atee Ejemplo 3:

We are going to consider the problem presented on figure 7(a), but in this case we use a coarser mesh (see Appendix2). In the figure 14 we can see how the interfaces meshes of each sub-domain are. The frame location will be computed using the proposed algorithm, and will be compared with the frame computed using the equivalent ZMR in 3D.

The first step is to compute the nodal frame contribution to virtual work. This contribution is showed on Table2, where the frame shape function act like weight of the forces coming from each sub-structure.
Figure 14. Interfaces meshes of each sub-domain.

<table>
<thead>
<tr>
<th>Node</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td></td>
<td>1/2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/4</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td>1/2</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_3(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_4(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/4</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td>1/2</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1/4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1/3</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-2/3</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-2/3</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1/3</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td>-2/3</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td></td>
<td>-2/3</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(N_1(\xi_{p_i}, \eta_{p_i}))</td>
<td>(N_2(\xi_{p_i}, \eta_{p_i}))</td>
<td>-1/3</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>-1/3</td>
</tr>
</tbody>
</table>

Table 2. Nodal frame contribution to virtual work.
Solving our optimization problem, we find the following solution, which is the same solution we obtain using the 3-D equivalent ZMR. Applying this frame to problem we obtain the linear displacement distribution (Appendix2).

\[
\begin{align*}
N_1(\zeta_2, \eta_2) &= 0.5000 \\
N_2(\zeta_2, \eta_2) &= 0.5000 \\
N_1(\zeta_4, \eta_4) &= 0.5000 \\
N_4(\zeta_4, \eta_4) &= 0.5000 \\
N_1(\zeta_5, \eta_5) &= 0.2500 \\
N_2(\zeta_5, \eta_5) &= 0.2500 \\
N_4(\zeta_5, \eta_5) &= 0.2500 \\
N_3(\zeta_5, \eta_5) &= 0.2500 \\
N_2(\zeta_6, \eta_6) &= 0.5000 \\
N_3(\zeta_6, \eta_6) &= 0.5000 \\
N_4(\zeta_8, \eta_8) &= 0.5000 \\
N_3(\zeta_8, \eta_8) &= 0.2500 \\
N_1(\zeta_11, \eta_11) &= 0.2500 \\
N_2(\zeta_11, \eta_11) &= 0.7500 \\
N_1(\zeta_12, \eta_12) &= 0.7500 \\
N_2(\zeta_12, \eta_12) &= 0.2500 \\
N_3(\zeta_15, \eta_15) &= 0.2500 \\
N_4(\zeta_15, \eta_15) &= 0.7500 \\
N_4(\zeta_16, \eta_16) &= 0.7500 \\
N_3(\zeta_16, \eta_16) &= 0.2500
\end{align*}
\]

\[
\begin{align*}
\sum_i x_{pi} &= \sum_{n=1}^{4} N_n(\xi_{pi}, \eta_{pi}) x_n \\
\sum_i y_{pi} &= \sum_{n=1}^{4} N_n(\xi_{pi}, \eta_{pi}) y_n
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0.888</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1.111</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>0.888</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>1.111</td>
<td>2</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 15. Nodal frame configuration for the interface considered on figure 14.
Ejemplo 4:

Now we are going to solve the same problem, but one of the meshes has some eccentricity (e=0.1mm) (see figure 16). Proceeding in the same way as in the above example, we can find the frame nodes location.

\[ N_1(\xi_2, \eta_2) = 0.4787 \]
\[ N_2(\xi_2, \eta_2) = 0.5213 \]
\[ N_1(\xi_4, \eta_4) = 0.4499 \]
\[ N_4(\xi_4, \eta_4) = 0.5501 \]
\[ N_1(\xi_5, \eta_5) = 0.1864 \]
\[ N_2(\xi_5, \eta_5) = 0.2638 \]
\[ N_4(\xi_5, \eta_5) = 0.2337 \]
\[ N_3(\xi_5, \eta_5) = 0.3161 \]
\[ N_2(\xi_6, \eta_6) = 0.4498 \]
\[ N_3(\xi_6, \eta_6) = 0.5502 \]
\[ N_4(\xi_6, \eta_6) = 0.4815 \]
\[ N_3(\xi_8, \eta_8) = 0.5185 \]

\[ N_1(\xi_{11}, \eta_{11}) = 0.3250 \]
\[ N_2(\xi_{11}, \eta_{11}) = 0.6750 \]
\[ N_1(\xi_{12}, \eta_{12}) = 0.8250 \]
\[ N_2(\xi_{12}, \eta_{12}) = 0.1750 \]
\[ N_3(\xi_{15}, \eta_{15}) = 0.3250 \]
\[ N_4(\xi_{15}, \eta_{15}) = 0.6750 \]
\[ N_4(\xi_{16}, \eta_{16}) = 0.8250 \]
\[ N_3(\xi_{16}, \eta_{16}) = 0.1750 \]
Figure 17. (a) Nodal frame configuration for the interface considered on figure 16. (b) Resultant forces on the interface.

Solving our optimization problem, we find the frame configuration (figure 17a). Computing the resultant forces in the normal and tangential directions we can see that the IPT is satisfy. Also, computing the frame location using the equivalent ZMR for 3D problems, we can obtain the same frame configuration.
8. Conclusions and Possible Future Work

A general algorithm for compute nodal frame location in non-matching interfacing problems based on optimization is presented. The main advantage of this algorithm is that we don’t have to work with the linear dependent equation set showed in (19). We build the objective function (25), and we solve the optimization problem with an iterative process. The advantage of this methodology over the ZMR is that is extensible to 3-D non-matched non-mapped interfacing problems, as we could see in the examples 3 and 4. But require estimating the configuration of the frame. If we select a wrong configuration, we won’t be able to reach the correct solution, although we’ll have a frame configuration.

The possible future work will be oriented to solve, in some way, the problem of having to choose a good initial configuration of the frame. And to be able to solve frames between curved interfacing surfaces.
References:


Lagrange multipliers and its applications. *Computational Mechanics.* 24: 476-490
(2000).


Appendix 1:

Ux

Uy
An algorithm for interfacing non-matching fem meshes

Uz

Ux
An algorithm for interfacing non-matching fem meshes

Advances Finite Element method
Appendix 2:
An algorithm for interfacing non-matching fem meshes
Appendix 3:
An algorithm for interfacing non-matching fem meshes