

EPFL

MASTER IN COMPUTATIONAL SCIENCE AND ENGINEERING

LABORATOIRE DE SIMULATION EN MÉCANIQUE DES SOLIDES

Implementation within Akantu of smooth contact mechanics with the mortar method

Author:

Davide PRADOVERA

Supervisor:

Prof. Guillaume ANCIAUX

Academic Year 2015-2016

Spring semester

Abstract

The scope of this project is the implementation of the *mortar* method for the enforcement of smooth contact constraints in an open-source finite element framework. Smooth contact constraints are rigorously defined, and two approaches to their enforcement (penalty and Lagrange multipliers) are derived rigorously. A detailed description of a complete algorithm for the application of such methods is provided, with a particular attention to the strategies used to tackle non-linearities. Then a brief presentation of the most important features of the implementation of the algorithm is given. To check the correctness of the implementation, several tests are performed: in particular the numerical results are compared with the hertzian solution for a problem of unilateral smooth contact.

Acknowledgements

The completion of this project would not have been possible without the participation and assistance of many people inside the “Laboratoire de Simulation en Mécanique des Solides” at EPFL. Their contributions are sincerely appreciated.

A particular mention goes to professor Guillaume Anciaux for his guidance and endless support.

Contents

1	Introduction	3
2	The mortar method	4
2.1	Background and notation	4
2.2	Mortar method for smooth contact	6
2.2.1	Lagrange multipliers approach	8
2.2.2	Penalty approach	9
2.3	Uniqueness and smoothness of the projection operator	10
2.4	Variation of normal gap	11
3	Algorithm	14
3.1	Identification of candidates for the active contact area	15
3.2	Coupling of the facets on the two sides	15
3.3	Computation of normals, gaps and local coordinates on quadrature points	16
3.4	Computation of extra entries of the stiffness matrix and of the RHS	18
3.4.1	Lagrange multipliers approach	19
3.4.2	Penalty approach	20
3.5	Solution of modified linear system	22
3.6	Convergence check	22
3.7	Check on active area	23
3.8	Comments on the algorithm	25
4	Implementation	26
4.1	Object-oriented portion of the code	26
4.2	Portion of the code in Python	29
5	Examples	30
5.1	Example 1: computation of the projection operator for linear finite elements in 2D	30
5.2	Example 2: application of the mortar approach to a simple 2D problem	31
5.3	Example 3: application of the mortar method to a hertzian problem	34
6	Conclusion and outlook	37
7	References	38

1 Introduction

Contact mechanics have a great importance because of their many applications in engineering and industry. In the last decades, research on this topic has spawned several strategies for the implementation of contact models: Lagrange multipliers, penalty and Nitsche methods are just some of the more known examples, which are widely applied in the field of computational solid mechanics.

In problems in which a mesh-based discretization is applied (which is of common practice, for example in *finite elements* or *finite differences* approaches), the application of any of the aforementioned methods may be complicated by non-matching meshes at the contact interface (see Figure 1). Such situations may easily occur even in the case of static analysis, because of the necessity to discretize differently two objects (or even portions of a single body), or simply as a consequence of the use of general mesh generators [1].

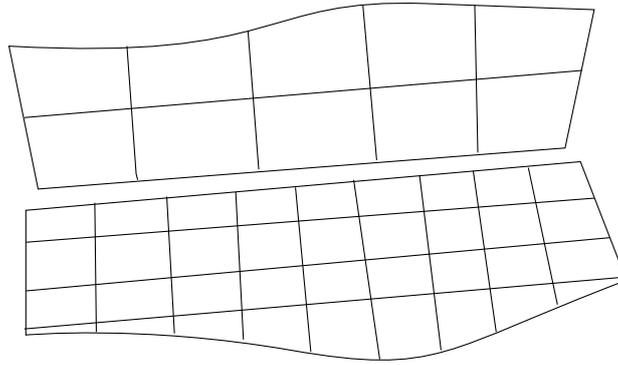


Figure 1: Non-matching discretizations at the contact interface.

Among the (many) results that recent research has lead to, this paper focuses on the *mortar* method as a way to apply contact conditions between non-matching meshes, and describes in detail the implementation of said method within the framework of *Akantu* [7], a free software which is being developed by the *Laboratoire de Simulation en Mécanique des Solides* (LSMS) at EPFL.

2 The mortar method

In this section, a brief overview of the mathematical foundations of the mortar method will be provided, along with the basic notation needed for its application to isoparametric finite elements. Most of the dissertation will be based on [1].

2.1 Background and notation

Given a smooth domain $\Omega \subseteq \mathbb{R}^d$ (with $d \in \{1, 2, 3\}$) partitioned in the (possibly empty) subsets Γ_D and Γ_N and a suitable functional space V (e.g. $V = (H^1(\Omega))^d$), the general formulation of the (potential) energy of Ω for a given displacement $\boldsymbol{\varphi}$ is:

$$\Pi(\boldsymbol{\varphi}) = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{\varphi}) : \boldsymbol{\varepsilon}(\boldsymbol{\varphi}) \, d\mathbf{X} - \int_{\Omega} \rho \mathbf{b} \cdot \boldsymbol{\varphi} \, d\mathbf{X} - \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \boldsymbol{\varphi} \, d\mathbf{X} \quad (1)$$

where $\boldsymbol{\varphi}$ respects some boundary conditions on Γ_D :

$$\boldsymbol{\varphi} \in V_{\bar{\mathbf{u}}} = \{\boldsymbol{\psi} \in V \text{ s.t. } \boldsymbol{\psi} = \bar{\mathbf{u}} \text{ on } \Gamma_D\} \quad (2)$$

In the formulae above, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\rho \mathbf{b}$ is the volume force field, \mathbf{n} is the outer normal vector to $\partial\Omega$, $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$ are vector fields (which represent the boundary values for displacement field and stress vector field respectively).

Moreover we will suppose that linear elasticity holds:

$$\begin{cases} \boldsymbol{\sigma}(\boldsymbol{\varphi}) = \mathbb{C}[\boldsymbol{\varepsilon}(\boldsymbol{\varphi})] \\ \boldsymbol{\varepsilon}(\boldsymbol{\varphi}) = \frac{\nabla \boldsymbol{\varphi} + \nabla^T \boldsymbol{\varphi}}{2} \end{cases} \quad (3)$$

where $\boldsymbol{\varepsilon}$ is the linear strain tensor and \mathbb{C} is the (linear) elasticity tensor.

Under smoothness and compatibility assumptions on $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$, if $|\Gamma_D| > 0$, it can be proven that $\Pi(\mathbf{u})$ is strictly convex. Hence it admits a unique minimum point \mathbf{u} , which may be found by solving

$$\partial_{\boldsymbol{\eta}} \Pi(\mathbf{u}) := \lim_{h \rightarrow 0} \frac{\Pi(\mathbf{u} + h \boldsymbol{\eta}) - \Pi(\mathbf{u})}{h} = 0 \quad \forall \boldsymbol{\eta} \in V_0 \quad (4)$$

which, exploiting the linearity of \mathbb{C} , can be written as

$$\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\eta}) \, d\mathbf{X} = \int_{\Omega} \rho \mathbf{b} \cdot \boldsymbol{\eta} \, d\mathbf{X} + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\mathbf{X} \quad \forall \boldsymbol{\eta} \in V_0 \quad (5)$$

with

$$V_0 = \{\boldsymbol{\psi} \in V \text{ s.t. } \boldsymbol{\psi} = \mathbf{0} \text{ on } \Gamma_D\} \quad (6)$$

In order to obtain a numerical approximation of the solution, a computational mesh Ω^h is introduced to discretize Ω (see Figure 2). Each of the nodes is characterized by a position $\mathbf{X}_i \in \mathbb{R}^d$ and an unknown displacement $\mathbf{u}_i \in \mathbb{R}^d$.

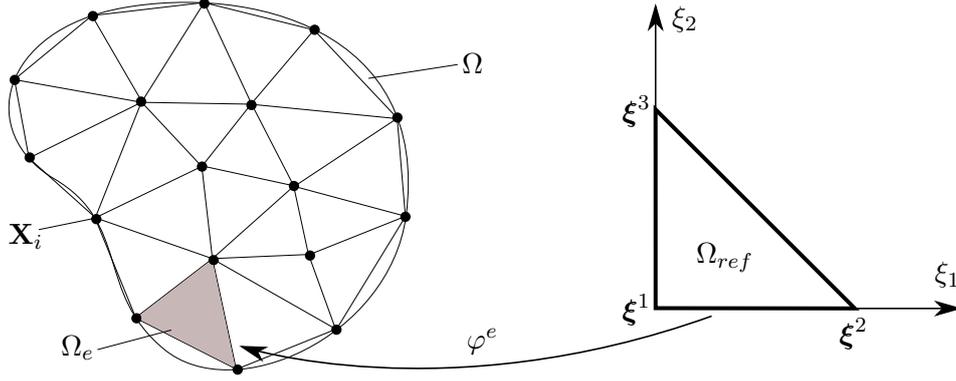


Figure 2: Domain discretization with triangular linear elements and isoparametric mapping from the reference element to the element Ω_e .

For each mesh element Ω_e , a (topologically equivalent) reference element Ω_{ref} is considered, and a bijective mapping $\varphi^e : \Omega_{ref} \rightarrow \Omega_e$ is defined (see Figure 2). By using this mapping, both geometry and field variables defined on Ω_e can be expressed on Ω_{ref} (in terms of the *natural coordinates* $\boldsymbol{\xi}$) through a composition with $(\varphi^e)^{-1}$.

The mapping φ^e is defined in terms of a lagrangian¹ *local* basis $\{N_I^e(\boldsymbol{\xi})\}_{I=1}^{N^e}$ of a given finite-dimensional vector space on the reference element (a polynomial space or a tensor product of polynomial spaces). For example, coordinates and displacements are defined as interpolations of their respective nodal values:

$$\mathbf{X}_e(\boldsymbol{\xi}) := \sum_{I=1}^{N^e} N_I^e(\boldsymbol{\xi}) \mathbf{X}_I^e \quad \text{and} \quad \mathbf{u}_e(\boldsymbol{\xi}) := \sum_{I=1}^{N^e} N_I^e(\boldsymbol{\xi}) \mathbf{u}_I^e \quad (7)$$

and the current positions can be obtained as

$$\mathbf{x}_e(\boldsymbol{\xi}) = \mathbf{X}_e(\boldsymbol{\xi}) + \mathbf{u}_e(\boldsymbol{\xi}) = \sum_{I=1}^{N^e} N_I^e(\boldsymbol{\xi}) (\mathbf{X}_I^e + \mathbf{u}_I^e) = \sum_{I=1}^{N^e} N_I^e(\boldsymbol{\xi}) \mathbf{x}_I^e \quad (8)$$

In general φ^e is non-linear in $\boldsymbol{\xi}$, but any interpolated field is always linear in its nodal values.

The Galerkin method can now be applied to problem (5) by introducing a finite-dimensional space $V^h \subseteq V$ to approximate V . The definition of V^h is based on the mesh Ω^h : the elements of V^h are usually chosen as globally continuous functions which are piecewise defined on Ω^h .

Then one has to define² the affine space

$$V_{\bar{\mathbf{u}}}^h = \{\boldsymbol{\psi} \in V^h \text{ s.t. } \boldsymbol{\psi} = \bar{\mathbf{u}} \text{ on } \Gamma_D\} \quad (9)$$

¹If $\{\boldsymbol{\xi}^J\}_{J=1}^{N^e}$ are the natural coordinates of the nodes, a lagrangian basis $\{N_I^e(\boldsymbol{\xi})\}_{I=1}^{N^e}$ has the property that

$$N_I^e(\boldsymbol{\xi}^J) = \delta_{IJ} \quad \text{for } I, J = 1, \dots, N^e$$

²If the domain Ω does not coincide with the mesh Ω^h (see Figure 2), in general the Galerkin method is *non-conforming*, since $V_{\bar{\mathbf{u}}}^h \not\subseteq V_{\bar{\mathbf{u}}}$ and $V_0^h \not\subseteq V_0$. In this case one should replace the Dirichlet boundary Γ_D with its approximation on the mesh in the two definitions (9) and (10).

and the subspace

$$V_0^h = \{\boldsymbol{\psi} \in V^h \text{ s.t. } \boldsymbol{\psi} = \mathbf{0} \text{ on } \Gamma_D\} \quad (10)$$

to approximate $V_{\bar{\mathbf{u}}}$ and V_0 respectively. Moreover a basis $\{\mathbf{v}_i\}_{i=1}^{\dim V_0^h}$ of V_0^h has to be chosen: because of its local support, the lagrangian basis on the nodes of the mesh is commonly used. If this is the case one can define element-wise the global basis $\{\mathbf{v}_i\}_{i=1}^{\dim V_0^h}$ in terms of the local basis $\{N_I^e\}_{I=1}^{N^e}$.

By exploiting the arbitrariness³ of virtual displacements $\boldsymbol{\eta}$ in (5), an equivalent formulation can be obtained:

$$\text{find } \mathbf{u} \in V_{\bar{\mathbf{u}}}^h \text{ s.t. } \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}_i) \, d\mathbf{X} = \int_{\Omega} \rho \mathbf{b} \cdot \mathbf{v}_i \, d\mathbf{X} + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v}_i \, d\mathbf{X} \text{ for } i = 1, \dots, \dim V_0^h \quad (11)$$

Furthermore, the integrals appearing in the weak formulation (11) can be computed element-wise and one can obtain relations of the form⁴

$$\text{find } \{\mathbf{u}_j\}_{j=1}^{\dim V^h} \subset \mathbb{R}^d \text{ s.t. } \sum_{j=1}^{\dim V^h} K_{ij} \mathbf{u}_j = \mathbf{f}_i \text{ for } i = 1, \dots, \dim V^h \quad (12)$$

where the coefficients $K_{ij} \in \mathbb{R}^{d \times d}$ and $\mathbf{f}_i \in \mathbb{R}^d$ depend on the boundary conditions, on the geometry of the problem, on the constitutive relation (3) and on the choice of the basis on each element $\{N_I^e\}_{I=1}^{N^e}$.

In the end the problem can be expressed as

$$K \hat{\mathbf{u}} = \hat{\mathbf{f}} \quad (13)$$

where $K \in \mathbb{R}^{(N_{nodes}d) \times (N_{nodes}d)}$, $\hat{\mathbf{u}}, \hat{\mathbf{f}} \in \mathbb{R}^{N_{nodes}d}$ and

$$(K)_{di+j, dk+l} = (K_{ik})_{jl}, \quad (\hat{\mathbf{u}})_{di+j} = (\mathbf{u}_i)_j \quad \text{and} \quad (\hat{\mathbf{f}})_{di+j} = (\mathbf{f}_i)_j \quad (14)$$

for $i, k = 1, \dots, N_{nodes}$ and $j, l = 1, \dots, d$.

2.2 Mortar method for smooth contact

We consider the situation where two bodies Ω^{nm} and Ω^m (the notation stands for “non-mortar” and “mortar” respectively) may come into contact (the extension to the case of self-contact is mostly trivial). The contact conditions are enforced on the two portions of the boundaries Γ^{nm} and Γ^m which are expected to come into contact. For any displacement field $\boldsymbol{\varphi}$ the *deformed* boundaries are given by

$$\gamma^{nm, \boldsymbol{\varphi}} := \{\mathbf{X} + \boldsymbol{\varphi}(\mathbf{X}) \text{ s.t. } \mathbf{X} \in \Gamma^{nm}\} \quad (15)$$

$$\gamma^{m, \boldsymbol{\varphi}} := \{\mathbf{X} + \boldsymbol{\varphi}(\mathbf{X}) \text{ s.t. } \mathbf{X} \in \Gamma^m\} \quad (16)$$

³To each nodal value $\boldsymbol{\eta}_i$ of the virtual displacements corresponds the equation indexed by i in (12).

⁴With respect to (11), one obtains $\dim V^h - \dim V_0^h$ additional equations of the form $\mathbf{u}_i = \bar{\mathbf{u}}_i$ from the Dirichlet boundary conditions, i.e. by imposing that \mathbf{u} belongs to $V_0^h \subset V^h$.

The *projection* operator

$$p^{(\varphi)} : \gamma^{nm,\varphi} \rightarrow \gamma^{m,\varphi}$$

is defined in the following way: for a given $\mathbf{x}^{nm} \in \gamma^{nm,\varphi}$, $p^{(\varphi)}(\mathbf{x}^{nm}) \in \gamma^{m,\varphi}$ is the closest point⁵ to \mathbf{x}^{nm} on the deformed mortar side $\gamma^{m,\varphi}$.

The constraint which prevents interpenetration between the bodies can be expressed as

$$(\mathbf{x} - p^{(\varphi)}(\mathbf{x})) \cdot \mathbf{n}^{m,\varphi}(p^{(\varphi)}(\mathbf{x})) \geq 0 \quad \forall \mathbf{x} \in \gamma^{nm,\varphi} \quad (17)$$

where $\mathbf{n}^{m,\varphi}(\mathbf{x})$ represents the outer normal to $\gamma^{m,\varphi}$ at point $\mathbf{x} \in \gamma^{m,\varphi}$ (see Figure 3). This condition is usually translated as

$$g_N^{(\varphi)}(\mathbf{X}) \geq 0 \quad \forall \mathbf{X} \in \Gamma^{nm} \quad \text{or} \quad g_N^{-}(\varphi)(\mathbf{X}) = 0 \quad \forall \mathbf{X} \in \Gamma^{nm} \quad (18)$$

where the *normal gap* $g_N^{(\varphi)}(\mathbf{X})$ and the *normal penetration* $g_N^{-}(\varphi)(\mathbf{X})$ are respectively defined as

$$g_N^{(\varphi)}(\mathbf{X}) = ((\mathbf{X} + \varphi(\mathbf{X})) - p^{(\varphi)}(\mathbf{X} + \varphi(\mathbf{X}))) \cdot \mathbf{n}^{m,\varphi}(p^{(\varphi)}(\mathbf{X} + \varphi(\mathbf{X}))) \quad (19)$$

and

$$g_N^{-}(\varphi)(\mathbf{X}) = \begin{cases} g_N^{(\varphi)}(\mathbf{X}) & \text{if } g_N^{(\varphi)}(\mathbf{X}) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

for $\mathbf{X} \in \Gamma^{nm}$.

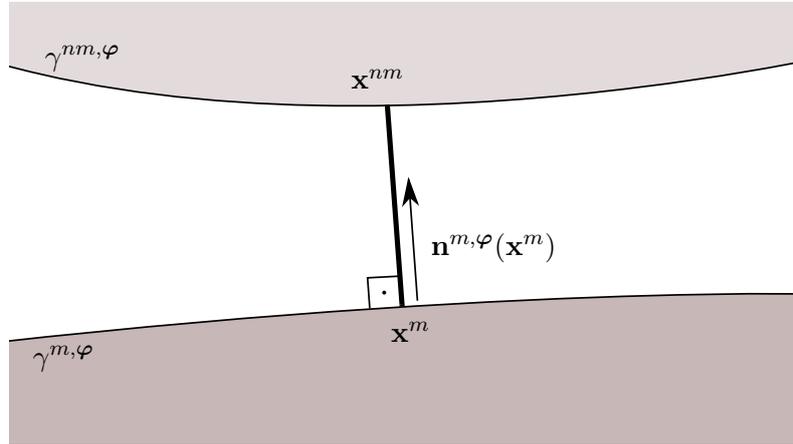


Figure 3: Surfaces which may come in contact ($\mathbf{x}^{nm} := \mathbf{X}^{nm} + \varphi(\mathbf{X}^{nm})$ and $\mathbf{x}^m := p^{(\varphi)}(\mathbf{x}^{nm})$).

In order to include this condition in the model, an additional contribution to the energy functional (1) has to be considered:

$$\Pi_{total} = \Pi + \Pi_{contact} \quad (21)$$

⁵If Γ^m is non-empty, there exists at least one point which fulfils the definition. Uniqueness issues will be discussed in Section 2.3.

2.2.1 Lagrange multipliers approach

In the Lagrange multipliers approach, the additional contribution to the energy is

$$\Pi_{contact}(\boldsymbol{\varphi}, \mu) = \Pi_{lag}(\boldsymbol{\varphi}, \mu) := \int_{\Gamma^{nm}} \mu g_N^{(\boldsymbol{\varphi})} d\mathbf{X} \quad (22)$$

where μ is a (non-negative) scalar field defined on Γ^{nm} which represents the Lagrange multiplier corresponding to the contact constraint (18).

Now, in the computation of the equilibrium configuration, one has to consider the variation with respect to both $\boldsymbol{\varphi}$ and μ :

find $\mathbf{u} \in V_{\bar{\mathbf{u}}}, \lambda \in Q, \lambda \geq 0$ s.t.

$$\begin{cases} \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\eta}) d\mathbf{X} + \int_{\Gamma^{nm}} \lambda \delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] d\mathbf{X} = \int_{\Omega} \rho \mathbf{b} \cdot \boldsymbol{\eta} d\mathbf{X} + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} d\mathbf{X} & \forall \boldsymbol{\eta} \in V_0 \\ \int_{\Gamma^{nm}} \delta \lambda g_N^{(\mathbf{u})} d\mathbf{X} = 0 & \forall \delta \lambda \in Q \end{cases} \quad (23)$$

where Q is a suitable functional space defined on Γ^{nm} (e.g. $Q = L^2(\Gamma^{nm})$) and $\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}]$ represents the variation of $g_N^{(\mathbf{u})}$ in direction $\boldsymbol{\eta}$ (see Section 2.4 for a rigorous definition).

The Galerkin approximation of (23) can be obtained similarly to the case discussed in the previous paragraph. However, one must choose an additional finite-dimensional space $Q^h \subseteq Q$ in which to approximate the Lagrange multiplier profile. Usually the elements of Q^h are piecewise polynomials⁶, whose degrees of freedom are nodal values on the boundary $\{\lambda_i\}_{i=1}^{\dim Q^h} \subset \mathbb{R}$. Lagrange multipliers can now be defined on each facet⁷ f of the non-mortar boundary in an isoparametric fashion:

$$\lambda_f(\boldsymbol{\xi}) := \sum_{I=1}^{M^f} M_I^f(\boldsymbol{\xi}) \lambda_I^f \quad (24)$$

where $\{M_I^f(\boldsymbol{\xi})\}_{I=1}^{M^f}$ is a lagrangian local basis for the Lagrange multipliers and $\{\lambda_I^f\}_{I=1}^{M^f} \subseteq \{\lambda_i\}_{i=1}^{\dim Q^h}$ are the nodal values of λ which are relevant for facet f (i.e. the values corresponding to Lagrange nodes belonging to f).

By exploiting the arbitrariness⁸ of virtual displacements $\boldsymbol{\eta}$ and lagrange multipliers variations $\delta \lambda$ in (23), one can obtain an approximate relation of the form (see Section 3.4.1)

$$\text{find } \{\mathbf{u}_k\}_{k=1}^{\dim V^h} \subset \mathbb{R}^d, \{\lambda_l\}_{l=1}^{\dim Q^h} \subset \mathbb{R}^+ \text{ s.t.} \quad \begin{cases} \sum_{k=1}^{\dim V^h} K_{ik} \mathbf{u}_k + \sum_{l=1}^{\dim Q^h} \mathbf{m}_{il} \lambda_l = \mathbf{f}_i & \text{for } i = 1, \dots, \dim V^h \\ \sum_{k=1}^{\dim V^h} \mathbf{m}_{kj} \cdot \mathbf{u}_k = g_j & \text{for } j = 1, \dots, \dim Q^h \end{cases} \quad (25)$$

⁶Sometimes the hypothesis of global continuity is dropped to gain other properties, see e.g. *dual Lagrangian basis* approaches in [1].

⁷We will call *facet* a portion of the boundary of an element e delimited by vertices (in 2D) or edges (in 3D) of e .

⁸To each nodal value $\boldsymbol{\eta}_i$ of the virtual displacements corresponds the equation indexed by i in (25), whereas to each nodal value $\delta \lambda_j$ of the Lagrange multipliers variations corresponds the equation indexed by j in (25).

where the new coefficients $\mathbf{m}_{kl} \in \mathbb{R}^d$ and $g_j \in \mathbb{R}$ depend on the active contact area Γ^{nm} , on the geometry of the problem, on the local basis on each element $\{N_I^\varepsilon\}_{I=1}^{N^e}$ and the local basis on each non-mortar facet $\{M_I^f\}_{I=1}^{M^f}$. In general the explicit expressions of these coefficients is not available, since they involve the non-linear projection function $p^{(\varphi)}$. Numerical approximations based on quadrature formulae will be provided in Section 3.4.1.

An equivalent expression for (25) is given by

$$\begin{bmatrix} K & M \\ M^T & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{g}} \end{bmatrix} \quad (26)$$

where K , $\hat{\mathbf{u}}$ and $\hat{\mathbf{f}}$ are the same as in (14). $M \in \mathbb{R}^{(N_{nodes}d) \times \dim Q^h}$ and $\hat{\boldsymbol{\lambda}}, \hat{\mathbf{g}} \in \mathbb{R}^{\dim Q^h}$ are given by

$$(M)_{di+j,k} = (\mathbf{m}_{ik})_j, \quad (\hat{\boldsymbol{\lambda}})_k = \lambda_k \quad \text{and} \quad (\hat{\mathbf{g}})_k = g_k \quad (27)$$

for $i = 1, \dots, N_{nodes}$, $j = 1, \dots, d$ and $k = 1, \dots, \dim Q^h$.

2.2.2 Penalty approach

The penalty approach provides an approximation of the contact constraint (18). The additional contribution to the energy is

$$\Pi_{contact}(\varphi) = \Pi_{pen}(\varphi) := \frac{1}{2} \int_{\Gamma^{nm}} \varepsilon_{pen} \left(g_N^-(\varphi) \right)^2 d\mathbf{X} \quad (28)$$

The number $\varepsilon_{pen} > 0$ is commonly called *penalty parameter*, and represents the stiffness of an elastic repulsion constraint due to normal penetration. For any given value ε_{pen} some residual penetration will be present in the solution so that the fictitious elastic force can balance the bulk elasticity. Larger penalty parameters lead to more accurate results: the limit solution for $\varepsilon_{pen} \rightarrow \infty$ coincides with the one obtained with the Lagrange approach [1].

In this case the weak formulation (5) becomes

$$\begin{aligned} \text{find } \mathbf{u} \in V_{\bar{\mathbf{u}}} \text{ s.t. } & \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\eta}) d\mathbf{X} + \int_{\Gamma^{nm}} \varepsilon_{pen} g_N^{-(\mathbf{u})} \delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] d\mathbf{X} = \\ & = \int_{\Omega} \rho \mathbf{b} \cdot \boldsymbol{\eta} d\mathbf{X} + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} d\mathbf{X} \quad \forall \boldsymbol{\eta} \in V_0 \end{aligned} \quad (29)$$

where $\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}]$ represents the variation⁹ of $g_N^{(\mathbf{u})}$ in direction $\boldsymbol{\eta}$ (see Section 2.4).

⁹The variation of the normal penetration $\delta g_N^{-(\mathbf{u})}[\boldsymbol{\eta}]$ has been replaced with the variation of the normal gap $\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}]$. If the gap is negative they are indeed the same (see (20)); if the gap is non-negative, they are different, but $g_N^{-(\mathbf{u})} \delta g_N^{-(\mathbf{u})}[\boldsymbol{\eta}] = g_N^{-(\mathbf{u})} \delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] = 0$ since the penetration is zero.

The Galerkin approximation of (29) can be obtained similarly to the case in which no contact is present. The final result can be approximated as (see Section 3.4.2)

$$\text{find } \{\mathbf{u}_j\}_{j=1}^{\dim V^h} \subset \mathbb{R}^d \text{ s.t. } \sum_{j=1}^{\dim V^h} (K_{ij} + \varepsilon_{pen} \delta K_{ij}) \mathbf{u}_j = (\mathbf{f}_i + \varepsilon_{pen} \delta \mathbf{f}_i) \quad \text{for } i = 1, \dots, \dim V^h \quad (30)$$

where the new coefficients $\delta K_{ij} \in \mathbb{R}^{(d \times d)}$ and $\delta \mathbf{f}_i \in \mathbb{R}^d$ depend on the active contact area Γ^{nm} , on the geometry of the problem and on the choice of the basis on each element $\{N_I^e\}_{I=1}^{N^e}$. In general the explicit expressions of these coefficients is not available, since they involve the non-linear projection function $p^{(\varphi)}$. Numerical approximations based on quadrature formulae will be provided in Section 3.4.2.

An equivalent expression for (30) is given by

$$(K + \varepsilon_{pen} \delta K) \widehat{\mathbf{u}} = \widehat{\mathbf{f}} + \varepsilon_{pen} \widehat{\delta \mathbf{f}} \quad (31)$$

where K , $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{f}}$ are the same as in (14) and $\delta K \in \mathbb{R}^{(N_{nodes}d) \times (N_{nodes}d)}$ and $\widehat{\delta \mathbf{f}} \in \mathbb{R}^{(N_{nodes}d)}$ are given by

$$(\delta K)_{di+j, dk+l} = (\delta K_{ik})_{jl} \quad \text{and} \quad (\widehat{\delta \mathbf{f}})_{di+j} = (\delta \mathbf{f}_i)_j \quad (32)$$

for $i, k = 1, \dots, N_{nodes}$ and $j, l = 1, \dots, d$.

2.3 Uniqueness and smoothness of the projection operator

Since the mortar boundary is non-empty, the existence of $p^{(\varphi)}(\mathbf{X} + \varphi(\mathbf{X}))$ is guaranteed for any displacement field φ and any point \mathbf{X} . However, in general it is neither unique nor continue, as shown in Figure 4: for \mathbf{X}_1 , two equidistant points (\mathbf{X}_2 and \mathbf{X}_3) can be chosen as possible projections.

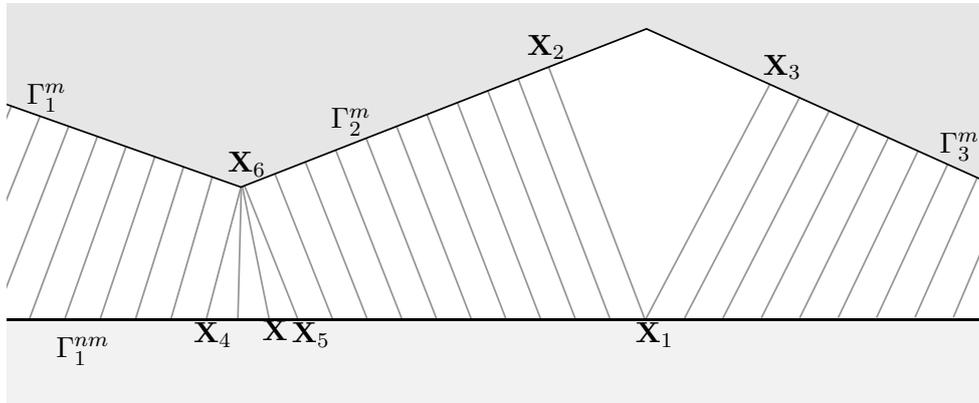


Figure 4: Computation of the projection operator for some points of Γ_1^{nm} . The projection $p^{(0)}(\mathbf{X}_1)$ may be chosen among the two values \mathbf{X}_2 and \mathbf{X}_3 . Also, $p^{(0)}(\mathbf{X}) = \mathbf{X}_6$ for all points \mathbf{X} between \mathbf{X}_4 and \mathbf{X}_5 .

Also, it is possible to observe that the projection operator is continuous (in \mathbf{X}) everywhere except in a set of points of zero measure (only \mathbf{X}_1 in the example shown in Figure 4): this can be proven rigorously, under some regularity assumptions on the mortar boundary (which hold in all mesh-based applications).

Using a similar argument, the continuity of $p^{(\varphi)}(\mathbf{X} + \varphi(\mathbf{X}))$ with respect to φ can be shown for any displacement field, except for those which cause the previously described singularities.

2.4 Variation of normal gap

In this section the variation of the normal gap will be properly defined and derived in the isoparametric case. The variation of the normal gap at point \mathbf{X} in configuration \mathbf{u} in direction $\boldsymbol{\eta}$ is given by

$$\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}](\mathbf{X}) := g_N^{(\mathbf{u} + \boldsymbol{\eta})}(\mathbf{X}) - g_N^{(\mathbf{u})}(\mathbf{X}) \quad (33)$$

The point $\mathbf{X} \in \mathbb{R}^d$ belongs to some non-mortar facet Γ_i^{nm} , hence one can write

$$\mathbf{x} := \mathbf{X} + \mathbf{u}(\mathbf{X}) = \sum_{I=1}^{N^{\Gamma_i^{nm}}} N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}) \left(\mathbf{X}_I^{\Gamma_i^{nm}} + \mathbf{u}_I^{\Gamma_i^{nm}} \right) \quad (34)$$

where $\{N_I^{\Gamma_i^{nm}}\}_{I=1}^{N^{\Gamma_i^{nm}}}$ is some basis defined on Γ_i^{nm} , and $\boldsymbol{\xi} = (\varphi^{\Gamma_i^{nm}})^{-1}(\mathbf{X})$ is some value of the natural coordinates on Γ_i^{nm} . $\mathbf{X}_I^{\Gamma_i^{nm}}$ and $\mathbf{u}_I^{\Gamma_i^{nm}}$ represent the nodal values of, respectively, the undeformed position and the displacement field on Γ_i^{nm} .

Similarly, the point $p^{(\mathbf{u})}(\mathbf{x})$ belongs to some mortar facet Γ_j^m , hence one can write

$$p^{(\mathbf{u})}(\mathbf{x}) = \sum_{I=1}^{N^{\Gamma_j^m}} N_I^{\Gamma_j^m}(\boldsymbol{\zeta}) \left(\mathbf{X}_I^{\Gamma_j^m} + \mathbf{u}_I^{\Gamma_j^m} \right) \quad (35)$$

where $\{N_I^{\Gamma_j^m}\}_{I=1}^{N^{\Gamma_j^m}}$ is some basis defined on Γ_j^m , and $\boldsymbol{\zeta} = (\varphi^{\Gamma_j^m})^{-1}(p^{(\mathbf{u})}(\mathbf{x}))$ is some value of the natural coordinates on Γ_j^m . $\mathbf{X}_I^{\Gamma_j^m}$ and $\mathbf{u}_I^{\Gamma_j^m}$ represent the nodal values of, respectively, the undeformed position and the displacement field on Γ_j^m .

Now we define

$$\tilde{\mathbf{x}} := \mathbf{X} + (\mathbf{u} + \boldsymbol{\eta})(\mathbf{X}) = \sum_{I=1}^{N^{\Gamma_i^{nm}}} N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}) \left(\mathbf{X}_I^{\Gamma_i^{nm}} + \mathbf{u}_I^{\Gamma_i^{nm}} + \boldsymbol{\eta}_I^{\Gamma_i^{nm}} \right) \quad (36)$$

For almost every \mathbf{X} (one has to exclude the zero-measure set which causes singularities in the projection operator), $p^{(\varphi)}(\mathbf{X} + \varphi(\mathbf{X}))$ is continuous in φ (see Section 2.3). Hence the projection in the perturbed configuration $p^{(\mathbf{u} + \boldsymbol{\eta})}(\tilde{\mathbf{x}})$ lies almost surely on the same mortar facet as $p^{(\mathbf{u})}(\mathbf{x})$, at least for $\boldsymbol{\eta}$ small enough.

Thus, $\tilde{\zeta}(\boldsymbol{\eta}) = \left(\varphi^{\Gamma_j^m}\right)^{-1} \left(p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}})\right)$ is well defined, and $p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}})$ can be written by using the local basis on Γ_j^m :

$$p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}}) = \sum_{I=1}^{N_j^{\Gamma_j^m}} N_I^{\Gamma_j^m} \left(\tilde{\zeta}(\boldsymbol{\eta})\right) \left(\mathbf{X}_I^{\Gamma_j^m} + \mathbf{u}_I^{\Gamma_j^m} + \boldsymbol{\eta}_I^{\Gamma_j^m}\right) \quad (37)$$

Also, another consequence of the continuity of $p^{(\varphi)}$ is:

$$\lim_{\|\boldsymbol{\eta}\| \rightarrow 0} \tilde{\zeta}(\boldsymbol{\eta}) = \zeta \quad (38)$$

since the limit projection coincides with $p^{(\mathbf{u})}(\mathbf{x})$.

Hence definition (33) yields¹⁰

$$\begin{aligned} \delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}](\mathbf{X}) &= (\tilde{\mathbf{x}} - p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}})) \cdot \mathbf{n}^{m,\mathbf{u}+\boldsymbol{\eta}} \left(p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}})\right) - (\mathbf{x} - p^{(\mathbf{u})}(\mathbf{x})) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) \simeq \\ &\simeq \left(\delta \mathbf{x}[\boldsymbol{\eta}] - \delta p^{(\mathbf{u})}[\boldsymbol{\eta}](\mathbf{x})\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) + \left(\mathbf{x} - p^{(\mathbf{u})}(\mathbf{x})\right) \cdot \delta \left(\mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right)\right) [\boldsymbol{\eta}] \end{aligned} \quad (39)$$

where higher order terms have been neglected.

We analyse separately the three terms:

- The first term involves the variation of \mathbf{x} and can be easily computed by using (34) and (36):

$$(\tilde{\mathbf{x}} - \mathbf{x}) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) = \left(\sum_{I=1}^{N_i^{\Gamma_i^{nm}}} N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}) \boldsymbol{\eta}_I^{\Gamma_i^{nm}}\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) \quad (40)$$

- The second term involves the variation of $p^{(\mathbf{u})}$: by using (35) and (37) one gets

$$\begin{aligned} &\left(p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}}) - p^{(\mathbf{u})}(\mathbf{x})\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) \simeq \\ &\simeq \sum_{I=1}^{N_j^{\Gamma_j^m}} \left(\left(N_I^{\Gamma_j^m} \left(\tilde{\zeta}(\boldsymbol{\eta})\right) - N_I^{\Gamma_j^m}(\zeta)\right) \left(\mathbf{X}_I^{\Gamma_j^m} + \mathbf{u}_I^{\Gamma_j^m}\right) + N_I^{\Gamma_j^m}(\zeta) \boldsymbol{\eta}_I^{\Gamma_j^m}\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) \simeq \\ &\simeq \sum_{I=1}^{N_j^{\Gamma_j^m}} \left(\partial_{\boldsymbol{\alpha}} N_I^{\Gamma_j^m}(\zeta) \left(\mathbf{X}_I^{\Gamma_j^m} + \mathbf{u}_I^{\Gamma_j^m}\right) + N_I^{\Gamma_j^m}(\zeta) \boldsymbol{\eta}_I^{\Gamma_j^m}\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) \end{aligned} \quad (41)$$

where $\boldsymbol{\alpha} = \tilde{\zeta}(\boldsymbol{\eta}) - \zeta$. In the last step we have used a first order Taylor expansion for $N_I^{\Gamma_j^m}$ near ζ . Also, in both steps higher order terms have been neglected. Comparing this expression with (34) yields

$$\left(\partial_{\boldsymbol{\alpha}} \mathbf{x}\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) + \left(\sum_{I=1}^{N_j^{\Gamma_j^m}} N_I^{\Gamma_j^m}(\zeta) \boldsymbol{\eta}_I^{\Gamma_j^m}\right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x})\right) \quad (42)$$

¹⁰The notation used for the term $\delta p^{(\mathbf{u})}[\boldsymbol{\eta}](\mathbf{x})$ is slightly incorrect and misleading. A rigorous definition is the one used in equation (41):

$$p^{(\mathbf{u}+\boldsymbol{\eta})}(\mathbf{X} + (\mathbf{u} + \boldsymbol{\eta})(\mathbf{X})) - p^{(\mathbf{u})}(\mathbf{X} + \mathbf{u}(\mathbf{X}))$$

Since the normal vector is orthogonal to the tangent plane (hence to any tangent vector $\partial_\alpha \mathbf{x}$), the first contribution is zero.

- The third term involves the variation of $\mathbf{n}^{m,\mathbf{u}}$:

$$(\mathbf{x} - p^{(\mathbf{u})}(\mathbf{x})) \cdot \left(\mathbf{n}^{m,\mathbf{u}+\boldsymbol{\eta}} \left(p^{(\mathbf{u}+\boldsymbol{\eta})}(\tilde{\mathbf{x}}) \right) - \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x}) \right) \right) \quad (43)$$

which is zero: indeed, $\mathbf{x} - p^{(\mathbf{u})}(\mathbf{x})$ is parallel to $\mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x}) \right)$ (because of the minimality properties of $p^{(\mathbf{u})}(\mathbf{x})$), and any unit norm vector is orthogonal to its variation¹¹.

In the end one gets

$$\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}](\mathbf{X}) = \left(\sum_{I=1}^{N_I^{\Gamma_i^{nm}}} N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}) \boldsymbol{\eta}_I^{\Gamma_i^{nm}} - \sum_{I=1}^{N_I^{\Gamma_j^m}} N_I^{\Gamma_j^m}(\boldsymbol{\zeta}) \boldsymbol{\eta}_I^{\Gamma_j^m} \right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x}) \right) \quad (44)$$

with $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ being the natural coordinates of, respectively, the starting point \mathbf{X} (or equivalently \mathbf{x} in the deformed configuration) and the projection $p^{(\mathbf{u})}(\mathbf{x})$.

It is important to note that this derivation is correct only if the mortar surface is smooth near $p^{(\mathbf{u})}(\mathbf{x})$. If the discretization of the boundary is not differentiable at $p^{(\mathbf{u})}(\mathbf{x})$, neither $\mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x}) \right)$ nor $\delta \left(\mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x}) \right) \right) [\boldsymbol{\eta}]$ are defined (see e.g. $p^{(0)}(\mathbf{x}) = \mathbf{x}_6$ in Figure 4): in this situation the normal vector is *defined* as a unit norm vector in direction $\mathbf{x} - p^{(\mathbf{u})}(\mathbf{x})$ (the sense has to be chosen so that the vector points outwards) and term (43) is *assumed* to be zero. This approximation could be equivalently obtained by replacing the sharp angle in the mesh with a differentiable surface (see Figure 5).

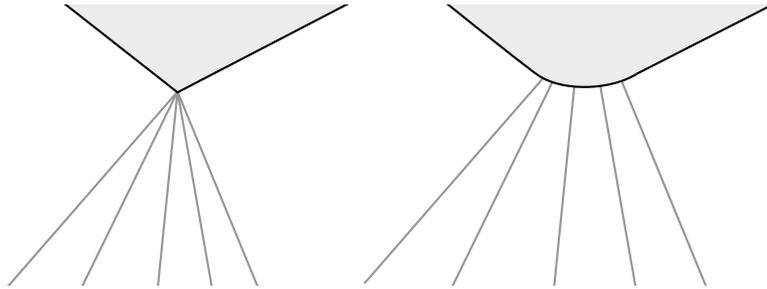


Figure 5: To define the outer normal in points where the surface is not differentiable, one may approximate sharp angles with C^1 junctions (similarly to how *mollifiers* are used in functional analysis to approximate non-smooth functions [2]).

¹¹The norm of a unit vector is constant, hence

$$0 = \delta(\mathbf{v} \cdot \mathbf{v}) = (\delta \mathbf{v}) \cdot \mathbf{v} + \mathbf{v} \cdot (\delta \mathbf{v}) = 2 \mathbf{v} \cdot (\delta \mathbf{v})$$

3 Algorithm

In order to use the mortar method, an algorithm based on multiple steps is carried out (see Figure 6). In the following paragraphs each step will be described in detail.

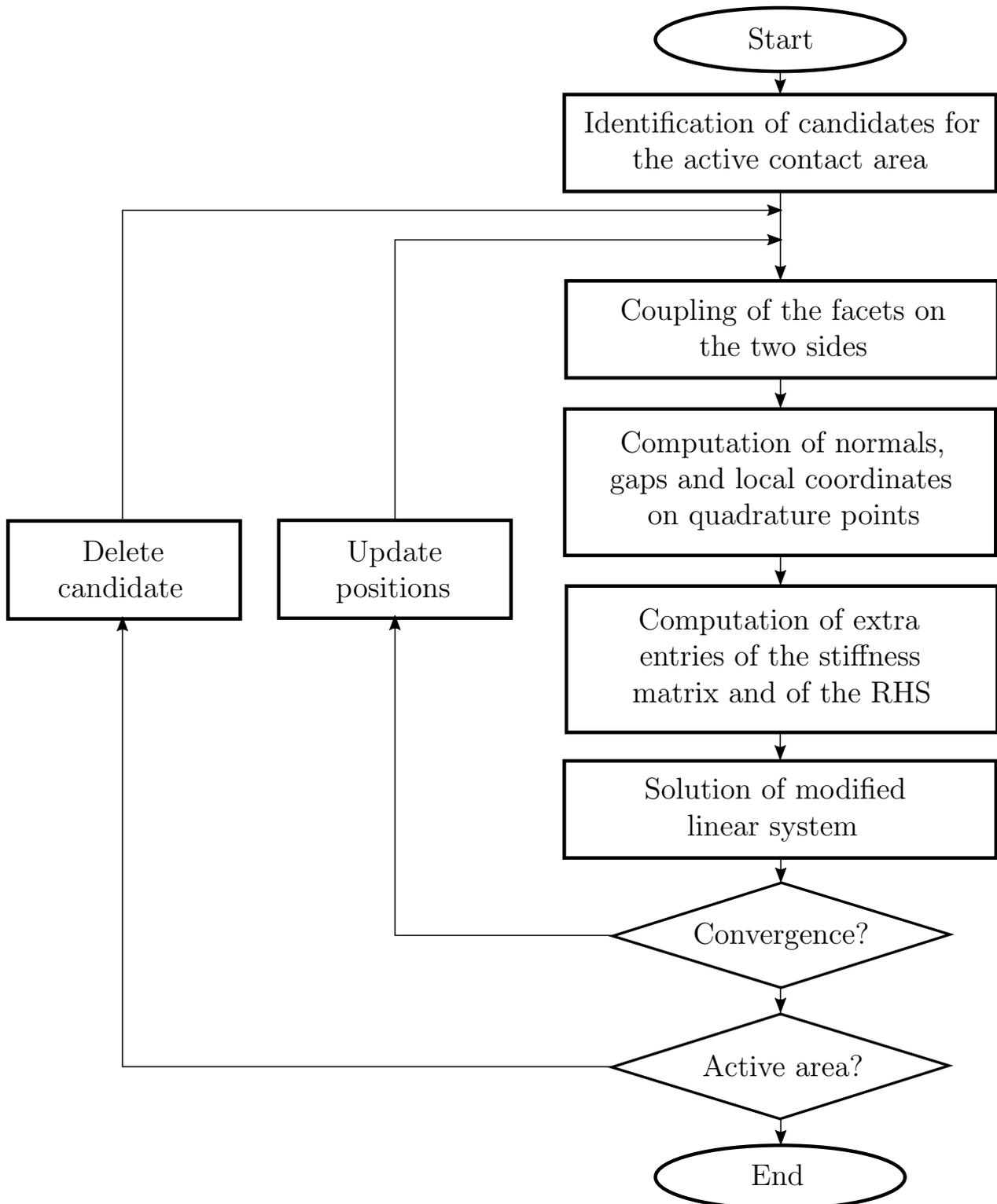


Figure 6: Diagram describing the mortar algorithm.

3.1 Identification of candidates for the active contact area

In the previous section, we have supposed the active boundaries Γ^{nm} and Γ^m to be known. In most application this is not true, and specialized algorithms have to be applied in order to find an initial guess for the active area (identifying exactly the active area at this stage is impossible if the solution is not known in advance).

Several efficient algorithms have been designed to solve this problem, but their description and implementation are outside the scope of this paper. As such we will suppose that two sets of facets $\tilde{\Gamma}^{nm} = \{\Gamma_i^{nm}\}$ and $\tilde{\Gamma}^m = \{\Gamma_i^m\}$, which are guesses for Γ^{nm} and Γ^m respectively, are provided by the user. Observe in particular that multiple separate active areas may be present (see Figure 7), hence we do not suppose the elements of $\tilde{\Gamma}^{nm}$ (and of $\tilde{\Gamma}^m$ as well) to be adjacent.

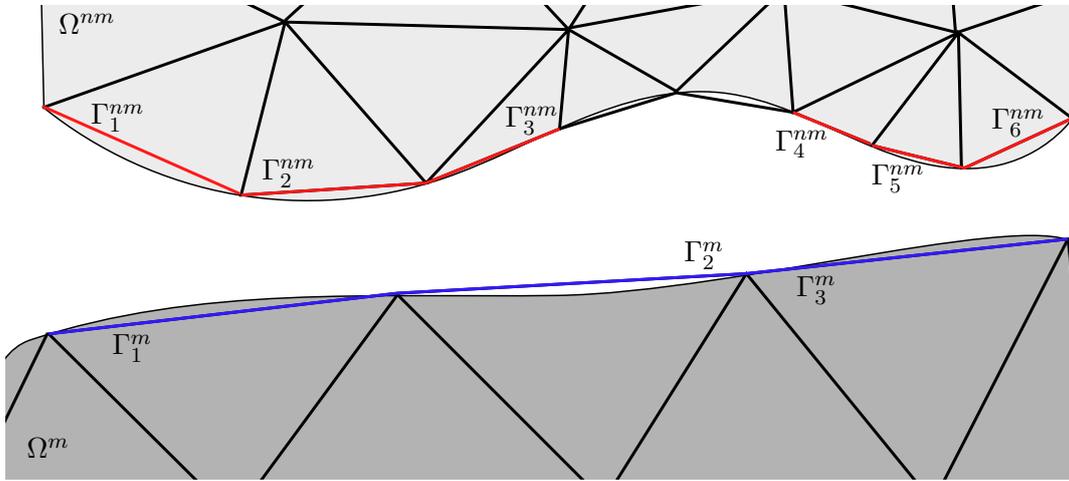


Figure 7: Identification of the active contact area. In red the elements of $\tilde{\Gamma}^{nm}$, in blue the elements of $\tilde{\Gamma}^m$.

3.2 Coupling of the facets on the two sides

One of the most crucial phases of the algorithm is the approximation¹² of the projection function $p^{(\varphi)}$. As such, it is split in two sub-steps: in the first one, the *coupling* step, the algorithm gathers informations about the neighborhood of the facets belonging to the non-mortar side.

For each non-mortar facet Γ_i^{nm} a list of mortar facets $\pi(\Gamma_i^{nm}) \subseteq \tilde{\Gamma}^m$ is considered. The elements of $\pi(\Gamma_i^{nm})$ represent the possible destinations of the projection operator from a point of Γ_i^{nm} . As such they are found (heuristically) by identifying, for each of the nodes \mathbf{X}_j^{nm} of

¹²Two types of numerical error prevent, in general, the computation of the exact value of $p^{(\varphi)}$: the first one is due to the discretization of the boundaries (e.g. $\Gamma_1^m \not\subseteq \Gamma^m$ in Figure 7), the second one is introduced by the approximation of p , which may be necessary because of its non-linearity (see Section 2.3).

Γ_i^{nm} , the mortar facet whose nodes have the smallest average distance from \mathbf{X}_j^{nm} :

$$\pi(\Gamma_i^{nm}) := \left\{ \arg \min_{\Gamma_k^m \in \tilde{\Gamma}^m} \frac{1}{N^{\Gamma_k^m}} \sum_{l=1}^{N^{\Gamma_k^m}} \|\mathbf{X}_l^{\Gamma_k^m} - \mathbf{X}_j^{\Gamma_i^{nm}}\|, \text{ for } j = 1, \dots, N^{\Gamma_i^{nm}} \right\} \quad (45)$$

where $\{\mathbf{X}_l^{\Gamma_k^m}\}_{l=1}^{N^{\Gamma_k^m}}$ represent the positions of the nodes of $\Gamma_k^m \in \tilde{\Gamma}^m$.

Problems may arise in this phase if the size of Γ_i^{nm} is too big when compared with the average size of the destination facets. Indeed, if this is the case, the projection of points $\mathbf{X} \in \Gamma_i^{nm}$ which are too far from the nodes may lay on elements not belonging to $\pi(\Gamma_i^{nm})$, as shown in Figure 8.

A possible solution to this issue involves the inclusion in $\pi(\Gamma_i^{nm})$ of all the facets which are the closest to some point $\mathbf{X} \in \Gamma_i^{nm}$. In general this is computationally expensive, since some kind of adaptive sampling on Γ_i^{nm} would be needed. An easier to implement (but less general) solution exploits the symmetry of the contact constraints (18) and is described in Section 3.8.

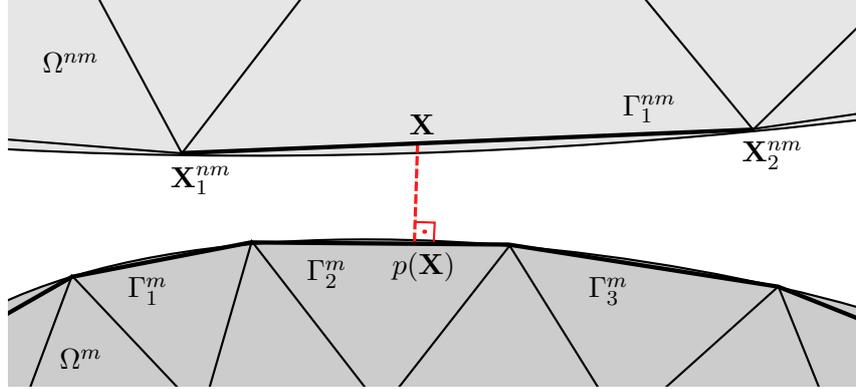


Figure 8: Γ_1^m and Γ_3^m belong to $\pi(\Gamma_1^{nm})$, since they are the closest facets to \mathbf{x}_1^{nm} and \mathbf{x}_2^{nm} respectively. However Γ_2^m is not included in $\pi(\Gamma_1^{nm})$, hence the computation of $p(\mathbf{x})$ could be inaccurate or even lead to errors.

3.3 Computation of normals, gaps and local coordinates on quadrature points

After the *coupling* step has been completed, it becomes possible to carry out the approximation of the projection function: in order to find the projection of a given point $\mathbf{x} \in \Gamma_i^{nm}$, it is projected on each of the possible destinations $\Gamma_j^m \in \pi(\Gamma_i^{nm})$; then the “best” (in some sense that will be defined) of the results is chosen as approximation of the projection.

It is important to note that undeformed boundaries are used instead of the deformed ones, which are unknown. Indeed, we suppose that the following first order approximation holds:

$$p^{(\mathbf{u})}(\mathbf{X} + \mathbf{u}(\mathbf{X})) \simeq p^{(0)}(\mathbf{X}) \quad \forall \mathbf{X} \in \Gamma^{nm} \quad (46)$$

Afterwards, an iterative procedure is applied to provide corrections to this assumption (see Section 3.6).

Given a point $\mathbf{x} \in \Gamma_i^{nm}$ on the non-mortar side and a mortar facet $\Gamma_j^m \in \pi(\Gamma_i^{nm})$, the problem of finding the projection of \mathbf{x} onto Γ_j^m can be cast in the following form:

$$p^{(0)}(\mathbf{x}; \Gamma_j^m) := \arg \min_{\mathbf{x}^m \in \Gamma_j^m} \|\mathbf{x}^m - \mathbf{x}\|^2 \quad (47)$$

which is equivalent to

$$p^{(0)}(\mathbf{x}; \Gamma_j^m) := \arg \min_{\zeta \in \Omega_{ref}^{\Gamma_j^m}} d(\zeta; \mathbf{x}, \Gamma_j^m)^2 := \arg \min_{\zeta \in \Omega_{ref}^{\Gamma_j^m}} \left\| \sum_{I=1}^{N_{\Gamma_j^m}} N_I^{\Gamma_j^m}(\zeta) \mathbf{X}_I^{\Gamma_j^m} - \mathbf{x} \right\|^2 \quad (48)$$

where $\Omega_{ref}^{\Gamma_j^m}$ is the reference element corresponding to Γ_j^m , i.e. the domain of $\varphi^{\Gamma_j^m}$:

$$\Gamma_j^m \equiv \varphi^{\Gamma_j^m} \left(\Omega_{ref}^{\Gamma_j^m} \right) \quad (49)$$

If the elements of the local basis are smooth functions, unbounded for $\|\zeta\| \rightarrow \infty$ (which is always true in the case of polynomials), problem (48) can be solved by looking for global minima first: by differentiating in a generic direction α one obtains

$$\left(\sum_{I=1}^{N_{\Gamma_j^m}} N_I^{\Gamma_j^m}(\zeta) \mathbf{X}_I^{\Gamma_j^m} - \mathbf{x} \right) \cdot \left(\sum_{I=1}^{N_{\Gamma_j^m}} \partial_{\alpha} N_I^{\Gamma_j^m}(\zeta) \mathbf{X}_I^{\Gamma_j^m} \right) = 0 \quad \forall \alpha \in \mathbb{R}^{d-1} \quad (50)$$

In general this leads to a system of $d - 1$ non-linear equations, whose solutions $\{\hat{\zeta}_k\}$ are the stationary points of $d(\cdot; \mathbf{x}, \Gamma_j^m)^2$.

Now a sub-routine is applied in order to select the best projection within the set of stationary points $\hat{Z} := \{\hat{\zeta}_k\}$:

1. The point $\hat{\zeta}_l \in \hat{Z}$ which minimizes $d(\cdot; \mathbf{x}, \Gamma_j^m)^2$ over \hat{Z} is found and removed from \hat{Z} .
2. If $\hat{\zeta}_l$ belongs to $\Omega_{ref}^{\Gamma_j^m}$, it is picked and the routine ends.
3. The projection of $\hat{\zeta}_l$ onto $\Omega_{ref}^{\Gamma_j^m}$ is found by solving (heuristically or exactly) the problem

$$\hat{\zeta}_l^{new} := \arg \min_{\zeta \in \Omega_{ref}^{\Gamma_j^m}} \|\zeta - \hat{\zeta}_l\| \quad (51)$$

4. $\hat{\zeta}_l^{new}$ is added to \hat{Z} .
5. If $\hat{\zeta}_l^{new}$ minimizes $d(\cdot; \mathbf{x}, \Gamma_j^m)^2$ over \hat{Z} , it is picked and the routine ends.
6. Otherwise $\hat{\zeta}_l^{new}$ is removed from \hat{Z} and the routine restarts from step 1 (the cardinality of \hat{Z} has decreased by 1).

This procedure (stationary point search + best projection selection) is repeated for all mortar facets in $\pi(\Gamma_i^{nm})$. Among the projections obtained $\{\mathbf{x}_k^m\}_{k=1}^{|\pi(\Gamma_i^{nm})|}$, with natural coordinates $\{\xi_k\}_{k=1}^{|\pi(\Gamma_i^{nm})|}$, the final approximation of the projection $p^{(0)}(\mathbf{x})$ is

$$\overline{p^{(0)}}(\mathbf{x}) := \arg \min_{k=1, \dots, |\pi(\Gamma_i^{nm})|} \|\mathbf{x}_k^m - \mathbf{x}\| \quad (52)$$

with corresponding natural coordinates $\bar{\xi}$. Also, in this step it is possible to identify the facet Γ_j^m on which the projection destination lies.

The normal vector can now be defined by normalizing the relative position between points¹³:

$$\overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{x})) := s \frac{\mathbf{x} - \overline{p^{(0)}}(\mathbf{x})}{\|\mathbf{x} - \overline{p^{(0)}}(\mathbf{x})\|} \quad (53)$$

with

$$s = \text{sign} \left(\left(\overline{p^{(0)}}(\mathbf{x}) - \mathbf{x} \right) \cdot \mathbf{n}^{nm,0}(\mathbf{x}) \right) \quad (54)$$

where $\mathbf{n}^{nm,0}(\mathbf{x})$ represents the outer normal vector to Γ_i^{nm} at point \mathbf{x} , which always exists (provided the elements of the basis $\{N_I^{\Gamma_i^{nm}}\}$ are smooth enough).

Moreover the normal gap in the undeformed configuration $g_N^{(0)}(\mathbf{x})$ can be easily found as

$$g_N^{(0)}(\mathbf{x}) = s \|\mathbf{x} - \overline{p^{(0)}}(\mathbf{x})\| \quad (55)$$

See Section 5.1 for a simple example of application of this step.

3.4 Computation of extra entries of the stiffness matrix and of the RHS

The previous steps provide the informations needed for the computation of the normal gap at any non-mortar point. This knowledge is used to compute numerically the coefficients introduced in (25) and (30), by applying quadrature formulae to the integrals appearing in the weak formulations (23) and (29).

Given some quadrature points $\{\xi_p\}_{p=1}^{n_p} \subseteq Z^{\Gamma_i^{nm}}$ and corresponding weights $\{w_p\}_{p=1}^{n_p} \subset \mathbb{R}$, the following approximation is considered:

$$\int_{\Gamma_i^{nm}} f(\mathbf{X}) d\mathbf{X} \simeq \sum_{p=1}^{n_p} f_{\Gamma_i^{nm}}(\xi_p) \|\mathbf{a}_p^{nm}\| w_p \quad (56)$$

\mathbf{a}_p^{nm} is the normal of the tangent vector, whose norm is equal to the determinant of the jacobian of the mapping $\varphi^{\Gamma_i^{nm}} : \Gamma_{ref} \rightarrow \Gamma_i^{nm}$ (see Section 2.1), computed on the quadrature point ξ_p . Moreover, we use the subscript Γ_i^{nm} to indicate the composition with the mapping $\varphi^{\Gamma_i^{nm}}$, e.g. $f_{\Gamma_i^{nm}} = f \circ \varphi^{\Gamma_i^{nm}}$.

¹³An alternative (simpler) definition is given by:

$$\overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{x})) := \mathbf{n}^{m,0}(\overline{p^{(0)}}(\mathbf{x}))$$

However, this expression is less general, since it cannot be applied when $\mathbf{n}^{m,0}(\overline{p^{(0)}}(\mathbf{x}))$ is undefined, i.e. if $\overline{p^{(0)}}(\mathbf{x})$ lies on the boundary of a facet (see Figure 5).

3.4.1 Lagrange multipliers approach

In the Lagrange multipliers approach, one needs to compute

$$\int_{\Gamma_i^{nm}} \lambda \delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] d\mathbf{X} \simeq \sum_{p=1}^{n_p} \lambda_{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \left(\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] \right)_{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \|\mathbf{a}_p^{nm}\| w_p \quad (57)$$

The first term has been defined in (24):

$$\lambda_{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) = \sum_{J=1}^{M_{\Gamma_i^{nm}}} M_J^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \lambda_J^{\Gamma_i^{nm}} \quad (58)$$

The second term has been computed in (44), and is approximated by $\left(\delta g_N^{(\mathbf{0})}[\boldsymbol{\eta}] \right)_{\Gamma_i^{nm}}(\boldsymbol{\xi}_p)$ (see Section 3.6):

$$\left(\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] \right)_{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \simeq \left(\sum_{I=1}^{N_{\Gamma_i^{nm}}} N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \boldsymbol{\eta}_I^{\Gamma_i^{nm}} - \sum_{I=1}^{N_{\Gamma_j^m}} N_I^{\Gamma_j^m}(\boldsymbol{\zeta}_p) \boldsymbol{\eta}_I^{\Gamma_j^m} \right) \cdot \mathbf{n}^{m,0}(p^{(0)}(\mathbf{X}_p)) \quad (59)$$

where $\mathbf{X}_p := \varphi^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p)$ and Γ_j^m is the facet on which the projection $p^{(0)}(\mathbf{X}_p)$ lies, and $\boldsymbol{\zeta}_p$ are the natural coordinates of $p^{(0)}(\mathbf{X}_p)$ on Γ_j^m .

Applying the previous step (see Section 3.3) with starting point $\mathbf{x} := \mathbf{X}_p$ yields

$$\left(\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] \right)_{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \simeq \left(\sum_{I=1}^{N_{\Gamma_i^{nm}}} N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \boldsymbol{\eta}_I^{\Gamma_i^{nm}} - \sum_{I=1}^{N_{\Gamma_{\bar{j},p}^m}} N_I^{\Gamma_{\bar{j},p}^m}(\bar{\boldsymbol{\zeta}}_p) \boldsymbol{\eta}_I^{\Gamma_{\bar{j},p}^m} \right) \cdot \overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{X}_p)) \quad (60)$$

where variables with a bar are the ones obtained in the previous step.

By comparing (58) and (60), it is possible to observe that the approximation of each term of the sum (57) is linear in both lagrange multipliers ($\lambda_J^{\Gamma_i^{nm}}$) and virtual displacements ($\boldsymbol{\eta}_I^{\Gamma_i^{nm}}$ and $\boldsymbol{\eta}_I^{\Gamma_{\bar{j},p}^m}$). This is true only because of the approximation $\mathbf{u} \simeq \mathbf{0}$ in the variation of the normal gap: if this assumption were dropped, non-linearities in the displacements \mathbf{u} would appear.

In the end, each term of the sum (57) can be written as

$$\sum_{I=1}^{N_{\Gamma_i^{nm}}} \sum_{J=1}^{M_{\Gamma_i^{nm}}} \lambda_J^{\Gamma_i^{nm}} \boldsymbol{\eta}_I^{\Gamma_i^{nm}} \cdot \boldsymbol{\mu}_{IJ}^p - \sum_{I=1}^{N_{\Gamma_{\bar{j},p}^m}} \sum_{J=1}^{M_{\Gamma_i^{nm}}} \lambda_J^{\Gamma_i^{nm}} \boldsymbol{\eta}_I^{\Gamma_{\bar{j},p}^m} \cdot \boldsymbol{\nu}_{IJ}^p \quad (61)$$

where

$$\begin{cases} \boldsymbol{\mu}_{IJ}^p := M_J^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \|\mathbf{a}_p^{nm}\| w_p \overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{X}_p)) \\ \boldsymbol{\nu}_{IJ}^p := M_J^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) N_I^{\Gamma_{\bar{j},p}^m}(\bar{\boldsymbol{\zeta}}_p) \|\mathbf{a}_p^{nm}\| w_p \overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{X}_p)) \end{cases} \quad (62)$$

A comparison with (23) and (25) shows that each $\boldsymbol{\mu}_{IJ}^p$ and $\boldsymbol{\nu}_{IJ}^p$ provides a contribution to some coefficient \mathbf{m}_{il} . The indices i and l may be identified by using the global numbering of the nodes instead of the local indexing used in the formulae above: i can be obtained from the

global index of the virtual displacement, whereas j is linked to the global index of the Lagrange multiplier.

A similar derivation for the integral

$$\int_{\Gamma_i^{\Gamma^{nm}}} \delta \lambda g_N^{(\mathbf{u})} d\mathbf{X} \simeq \sum_{p=1}^{n_p} \delta \lambda_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) \left(g_N^{(\mathbf{u})} \right)_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) \|\mathbf{a}_p^{nm}\| w_p \quad (63)$$

leads to

$$\sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} \sum_{J=1}^{M^{\Gamma_i^{\Gamma^{nm}}}} \delta \lambda_J^{\Gamma_i^{\Gamma^{nm}}} \left(\mathbf{X}_I^{\Gamma_i^{\Gamma^{nm}}} + \mathbf{u}_I^{\Gamma_i^{\Gamma^{nm}}} \right) \cdot \boldsymbol{\mu}_{IJ}^p - \sum_{I=1}^{N^{\Gamma_{j,p}^m}} \sum_{J=1}^{M^{\Gamma_i^{\Gamma^{nm}}}} \delta \lambda_J^{\Gamma_i^{\Gamma^{nm}}} \left(\mathbf{X}_I^{\Gamma_{j,p}^m} + \mathbf{u}_I^{\Gamma_{j,p}^m} \right) \cdot \boldsymbol{\nu}_{IJ}^p \quad (64)$$

where $\boldsymbol{\mu}_{IJ}^p$ and $\boldsymbol{\nu}_{IJ}^p$ are the same as before. A comparison with (23) and (25) shows that each $\boldsymbol{\mu}_{IJ}^p$ and $\boldsymbol{\nu}_{IJ}^p$ provides a contribution to some coefficient \mathbf{m}_{kj} , in a symmetric way with respect to the first integral. Hence we have checked that the coefficients \mathbf{m}_{kl} appearing in the first and second equation of (25) are the same.

Moreover, the terms which do not depend on the displacements \mathbf{u} , i.e.

$$\sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} \sum_{J=1}^{M^{\Gamma_i^{\Gamma^{nm}}}} \delta \lambda_J^{\Gamma_i^{\Gamma^{nm}}} \mathbf{X}_I^{\Gamma_i^{\Gamma^{nm}}} \cdot \boldsymbol{\mu}_{IJ}^p - \sum_{I=1}^{N^{\Gamma_{j,p}^m}} \sum_{J=1}^{M^{\Gamma_i^{\Gamma^{nm}}}} \delta \lambda_J^{\Gamma_i^{\Gamma^{nm}}} \mathbf{X}_I^{\Gamma_{j,p}^m} \cdot \boldsymbol{\nu}_{IJ}^p \quad (65)$$

provide contributions to some g_j coefficient (once the variation $\delta \lambda_J^{\Gamma_i^{\Gamma^{nm}}}$ is factored out).

3.4.2 Penalty approach

In the penalty approach, one needs to compute

$$\int_{\Gamma_i^{\Gamma^{nm}}} \varepsilon_{pen} g_N^{-(\mathbf{u})} \delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] d\mathbf{X} \simeq \sum_{p=1}^{n_p} \varepsilon_{pen} \left(g_N^{-(\mathbf{u})} \right)_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) \left(\delta g_N^{(\mathbf{u})}[\boldsymbol{\eta}] \right)_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) \|\mathbf{a}_p^{nm}\| w_p \quad (66)$$

The expression of the normal gap can be obtained by plugging (7) into (24):

$$\left(g_N^{(\mathbf{u})} \right)_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) = \left(\sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} N_I^{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) \mathbf{x}_I^{\Gamma_i^{\Gamma^{nm}}} - \sum_{I=1}^{N^{\Gamma_{j,\mathbf{u}}^m}} N_I^{\Gamma_{j,\mathbf{u}}^m}(\boldsymbol{\zeta}_p^{(\mathbf{u})}) \mathbf{x}_I^{\Gamma_{j,\mathbf{u}}^m} \right) \cdot \mathbf{n}^{m,\mathbf{u}} \left(p^{(\mathbf{u})}(\mathbf{x}_p) \right) \quad (67)$$

where

$$\mathbf{x}_I^\gamma := \mathbf{X}_I^\gamma + \mathbf{u}_I^\gamma \quad \text{for } \gamma \in \{\Gamma_i^{\Gamma^{nm}}, \Gamma_j^m\} \quad (68)$$

and $\mathbf{x}_p := \varphi^{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) + \mathbf{u}(\varphi^{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p))$. Also, $\Gamma_{j,\mathbf{u}}^m$ is the mortar facet which contains $p^{(\mathbf{u})}(\mathbf{x}_p)$, and $\boldsymbol{\zeta}_p^{(\mathbf{u})}$ represent the natural coordinates of $p^{(\mathbf{u})}(\mathbf{x}_p)$ on $\Gamma_{j,\mathbf{u}}^m$.

For all terms except the deformed positions \mathbf{x}_I^γ , the approximation $\mathbf{u} \simeq \mathbf{0}$ is applied (see

Section 3.6):

$$\begin{aligned}
(g_N^{(\mathbf{u})})_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) &\simeq \left(\sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} N_I^{\Gamma^{nm}}(\boldsymbol{\xi}_p) \mathbf{x}_I^{\Gamma^{nm}} - \sum_{I=1}^{N^{\Gamma_j^{\Gamma^m}}} N_I^{\Gamma^m}(\boldsymbol{\zeta}_p) \mathbf{x}_I^{\Gamma^m} \right) \cdot \mathbf{n}^{m,0}(p^{(0)}(\mathbf{X}_p)) = \\
&= \left(\sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} N_I^{\Gamma^{nm}}(\boldsymbol{\xi}_p) (\mathbf{X}_I^{\Gamma^{nm}} + \mathbf{u}_I^{\Gamma^{nm}}) - \sum_{I=1}^{N^{\Gamma_{\bar{j},p}^{\Gamma^m}}} N_I^{\Gamma^m}(\bar{\boldsymbol{\zeta}}_p) (\mathbf{X}_I^{\Gamma^m} + \mathbf{u}_I^{\Gamma^m}) \right) \cdot \overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{X}_p))
\end{aligned} \tag{69}$$

where the same notation as (59) and (60) has been used. The values with a bar are the ones obtained in the previous step (see Section 3.3) with starting point $\mathbf{x} := \mathbf{X}_p$.

It is important to notice that the normal penetration, and not the normal gap, appears in the weak formulation (29): hence one must make sure that no contribution to the weak form comes from the terms of the sum (66) which have a positive normal gap $g_N^{(\mathbf{u})}(\mathbf{x}_p) > 0$. Since the displacement field \mathbf{u} is not known, it is impossible to know the sign of the exact normal gap. As such, a check is performed on the gap computed for the undeformed configuration: if $g_N^{(0)}(\mathbf{X}_p)$ is positive, the whole p -th contribution is put to zero.

The second term is approximated by $(\delta g_N^{(0)}[\boldsymbol{\eta}])_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p)$, which has been computed in (60):

$$(\delta g_N^{(0)}[\boldsymbol{\eta}])_{\Gamma_i^{\Gamma^{nm}}}(\boldsymbol{\xi}_p) = \left(\sum_{J=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} N_J^{\Gamma^{nm}}(\boldsymbol{\xi}_p) \boldsymbol{\eta}_J^{\Gamma^{nm}} - \sum_{J=1}^{N^{\Gamma_{\bar{j},p}^{\Gamma^m}}} N_J^{\Gamma^m}(\bar{\boldsymbol{\zeta}}_p) \boldsymbol{\eta}_J^{\Gamma^m} \right) \cdot \overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{X}_p)) \tag{70}$$

By comparing (69) and (70), it is possible to observe that the approximation of each term of the sum (66) is linear in both displacements ($\mathbf{u}_I^{\Gamma^{nm}}$ and $\mathbf{u}_I^{\Gamma_{\bar{j},p}^{\Gamma^m}}$) and virtual displacements ($\boldsymbol{\eta}_I^{\Gamma^{nm}}$ and $\boldsymbol{\eta}_I^{\Gamma_{\bar{j},p}^{\Gamma^m}}$). Similarly to the lagrange multiplier case, this is true only because of the approximation $\mathbf{u} \simeq \mathbf{0}$ in the variation of the normal gap: if this assumption were dropped, non-linearities in both displacements \mathbf{u} and virtual displacements $\boldsymbol{\eta}$ would appear.

In the end, each term of the sum (66) can be written as

$$\begin{aligned}
\varepsilon_{pen} &\left(\sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} \sum_{J=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} (\boldsymbol{\eta}_J^{\Gamma^{nm}})^T \Xi_{IJ}^p (\mathbf{X}_I^{\Gamma^{nm}} + \mathbf{u}_I^{\Gamma^{nm}}) + \sum_{I=1}^{N^{\Gamma_{\bar{j},p}^{\Gamma^m}}} \sum_{J=1}^{N^{\Gamma_{\bar{j},p}^{\Gamma^m}}} (\boldsymbol{\eta}_J^{\Gamma^m})^T \Upsilon_{IJ}^p (\mathbf{X}_I^{\Gamma^m} + \mathbf{u}_I^{\Gamma^m}) + \right. \\
&\left. - \sum_{I=1}^{N^{\Gamma_{\bar{j},p}^{\Gamma^m}}} \sum_{J=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} (\boldsymbol{\eta}_J^{\Gamma^{nm}})^T \Phi_{IJ}^p (\mathbf{X}_I^{\Gamma_{\bar{j},p}^{\Gamma^m}} + \mathbf{u}_I^{\Gamma_{\bar{j},p}^{\Gamma^m}}) - \sum_{I=1}^{N^{\Gamma_i^{\Gamma^{nm}}}} \sum_{J=1}^{N^{\Gamma_{\bar{j},p}^{\Gamma^m}}} (\boldsymbol{\eta}_J^{\Gamma^m})^T \Psi_{IJ}^p (\mathbf{X}_I^{\Gamma^{nm}} + \mathbf{u}_I^{\Gamma^{nm}}) \right)
\end{aligned} \tag{71}$$

where

$$\begin{cases} \Xi_{IJ}^p := & N_I^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) N_J^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \|\mathbf{a}_p^{nm}\| w_p \left(\overline{\mathbf{n}^{m,0}} \left(\overline{p^{(0)}}(\mathbf{X}_p) \right) \otimes \overline{\mathbf{n}^{m,0}} \left(\overline{p^{(0)}}(\mathbf{X}_p) \right) \right) \\ \Upsilon_{IJ}^p := & N_I^{\Gamma_{j,p}^m}(\bar{\boldsymbol{\zeta}}_p) N_J^{\Gamma_{j,p}^m}(\bar{\boldsymbol{\zeta}}_p) \|\mathbf{a}_p^{nm}\| w_p \left(\overline{\mathbf{n}^{m,0}} \left(\overline{p^{(0)}}(\mathbf{X}_p) \right) \otimes \overline{\mathbf{n}^{m,0}} \left(\overline{p^{(0)}}(\mathbf{X}_p) \right) \right) \\ \Psi_{JI}^p := \Phi_{IJ}^p := & N_I^{\Gamma_{j,p}^m}(\bar{\boldsymbol{\zeta}}_p) N_J^{\Gamma_i^{nm}}(\boldsymbol{\xi}_p) \|\mathbf{a}_p^{nm}\| w_p \left(\overline{\mathbf{n}^{m,0}} \left(\overline{p^{(0)}}(\mathbf{X}_p) \right) \otimes \overline{\mathbf{n}^{m,0}} \left(\overline{p^{(0)}}(\mathbf{X}_p) \right) \right) \end{cases} \quad (72)$$

A comparison with (29) and (30) shows that each Ξ_{IJ}^p , Υ_{IJ}^p , Φ_{IJ}^p and Ψ_{JI}^p provides a contribution to some coefficient δK_{ij} . The indices i and j may be identified by using the global numbering of the nodes instead of the local indexing used in the formulae above: i can be obtained from the global index of the virtual displacement, whereas j is linked to the global index of the displacement.

Moreover, the terms which do not depend on the displacements \mathbf{u} , i.e.

$$\begin{aligned} \varepsilon_{pen} \left(\sum_{I=1}^{N^{\Gamma_i^{nm}}} \sum_{J=1}^{N^{\Gamma_i^{nm}}} \left(\boldsymbol{\eta}_J^{\Gamma_i^{nm}} \right)^T \Xi_{IJ}^p \mathbf{X}_I^{\Gamma_i^{nm}} + \sum_{I=1}^{N^{\Gamma_{j,p}^m}} \sum_{J=1}^{N^{\Gamma_{j,p}^m}} \left(\boldsymbol{\eta}_J^{\Gamma_{j,p}^m} \right)^T \Upsilon_{IJ}^p \mathbf{X}_I^{\Gamma_{j,p}^m} + \right. \\ \left. - \sum_{I=1}^{N^{\Gamma_{j,p}^m}} \sum_{J=1}^{N^{\Gamma_i^{nm}}} \left(\boldsymbol{\eta}_J^{\Gamma_i^{nm}} \right)^T \Phi_{IJ}^p \mathbf{X}_I^{\Gamma_{j,p}^m} - \sum_{I=1}^{N^{\Gamma_i^{nm}}} \sum_{J=1}^{N^{\Gamma_{j,p}^m}} \left(\boldsymbol{\eta}_J^{\Gamma_{j,p}^m} \right)^T \Psi_{IJ}^p \mathbf{X}_I^{\Gamma_i^{nm}} \right) \quad (73) \end{aligned}$$

represent contributions to some $\delta \mathbf{f}_i$ coefficient (once the virtual displacements $\boldsymbol{\eta}_J^{\Gamma_i^{nm}}$ and $\boldsymbol{\eta}_J^{\Gamma_{j,p}^m}$ are factored out).

3.5 Solution of modified linear system

When the modified linear system ((26) for the Lagrange multipliers approach and (31) for the penalty method) has been assembled, it is solved to find the nodal values of the displacement field (and of the Lagrange multipliers if present). In particular, some efficient algorithms exploiting the structure of the modified system have been designed [1].

3.6 Convergence check

In most of the previous steps, we used the assumption that normal gaps and projections behave similarly in the undeformed (\mathbf{X}) and in the deformed configuration (\mathbf{x}):

$$g_N^{(\mathbf{u})}(\mathbf{x}) \simeq g_N^{(0)}(\mathbf{X}), \quad p^{(\mathbf{u})}(\mathbf{x}) \simeq p^{(0)}(\mathbf{X}) \quad \text{and} \quad \mathbf{n}^{m,\mathbf{u}}(p^{(\mathbf{u})}(\mathbf{x})) \simeq \mathbf{n}^{m,0}(p^{(0)}(\mathbf{X})) \quad (74)$$

In general this is not the case, since deformations due to contact may change quite drastically the results of the projection operator.

To tackle this issue, once the projection operator has been approximated and the displacements have been found by solving the modified linear system, a check is performed on the

magnitude of the results: if some norm of the nodal values of \mathbf{u} (for example the L^2 -norm) is larger than a threshold, the results are assumed to be inaccurate, and an iterative procedure is carried out, otherwise the algorithm proceeds to the next step.

The iterations proceed in the following way: the approximate deformed configuration $\mathbf{X} + \mathbf{u}(\mathbf{X})$ which comes from the modified linear system solution is used to replace the undeformed configuration:

$$\Omega^{new} := \Omega + \mathbf{u}(\Omega) \quad (75)$$

Hence the non-mortar and mortar boundaries are updated as well:

$$(\Gamma^{nm})^* := \Gamma^{nm} + \mathbf{u}(\Gamma^{nm}) \quad \text{and} \quad (\Gamma^m)^* := \Gamma^m + \mathbf{u}(\Gamma^m) \quad (76)$$

Now, as shown in Figure 6, these updated values are used to restart the algorithm from the *coupling* step.

This procedure is motivated by the hypothesis of “small deformations”, and is equivalent to the application of a fixed point method to solve non-linearities. Performance and robustness can be improved by considering a Newton-Raphson approach [1].

It is important to note that this iterative approach is particularly useful in the penalty case. Indeed, if the penalty parameter is too large, issues with the condition number of the modified system may appear (see Section 3.8), hence there is an upper bound to the values among which ε_{pen} may be chosen; as discussed in Section 2.2.2, this usually leads to inaccuracy in the solution, with presence of residual interpenetration. By applying the iterative procedure and combining results from different iterations, a reduction of the interpenetration is achieved (see Section 5.2).

3.7 Check on active area

A second important check on the solution has to be performed. As discussed in Section 3.1, the active boundaries Γ^{nm} and Γ^m are not known in advance.

If corrections are not applied, the numerical solution may show non-physical features: in particular *traction* may be present in the contact area even if the material does not have cohesive properties. This phenomenon is of extreme importance in the Lagrange multipliers approach because of the way contact constraints (18) are enforced¹⁴.

By looking at the continuous model (23), the second equation is used to enforce (weakly) the contact between bodies: if, for example, $Q = L^2(\Gamma^{nm})$, the equation is equivalent to an

¹⁴The presence of this phenomenon is of less magnitude in the penalty approach, but only if the penalty parameter ε_{pen} is small enough. Indeed, the iterations described in Section 3.6 and the fact that the normal penetration is used in the weak formulation (instead of the normal gap) tend to counterbalance this effect.

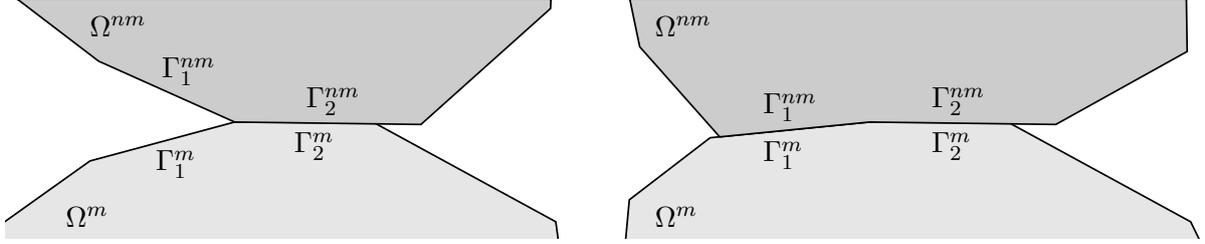


Figure 9: Solutions of a contact problem between polygonal meshes: exact (left) and numerical (right) configurations. In the numerical solution, the misclassification of Γ_1^{nm} as an active facet, if not corrected, leads to non-physical traction between the objects.

orthogonality condition, which leads to

$$g_N^{(\mathbf{u})} = 0 \quad \text{in the } L^2(\Gamma^{nm}) \text{ sense} \quad (77)$$

With the Galerkin approximation this condition is only enforced on the average value of the gap on each non-mortar facet:

$$\int_{\Gamma_i^{nm}} g_N^{(\mathbf{u})}(\mathbf{X}) \, d\mathbf{X} = 0 \quad \forall \Gamma_i^{nm} \in \tilde{\Gamma}^{nm} \quad (78)$$

which is approximated by a quadrature formula. This means that the contact constraint (17) will be *active* (in a weak sense) on all facets belonging to $\tilde{\Gamma}^{nm}$.

Now, suppose that $\Gamma_1^{nm} \in \tilde{\Gamma}^{nm}$ does not actually belong to the active area, i.e. in the exact solution the normal gap on all Γ_1^{nm} is positive (see Figure 9). If the algorithm (as described until now) is applied, then it will effectively lead to the presence of a non-physical (attractive) force which keeps Γ_1^{nm} in contact with the mortar side, as shown in Figure 9. As a consequence, one or more nodal values of the Lagrange multipliers (which represent repulsive forces between the two objects) will be negative (see Section 5.3).

To correct the numerical results, it is enough to remove Γ_1^{nm} from $\tilde{\Gamma}^{nm}$ and restart the algorithm from the original undeformed configuration¹⁵.

In general, to identify the “worst behaving” element, one can compute the nodal values of the contact force by restricting the residual

$$\mathbf{r} := K \mathbf{u} - \mathbf{f} \quad (79)$$

on the non-mortar nodes. Now a check can be performed on each node: if the minimum value of the normal component of the residual is too small, all the elements containing the corresponding node are removed from $\tilde{\Gamma}^{nm}$.

¹⁵Because of the iterative procedure described in Section 3.6, there might have been modifications in the undeformed configuration. If any element is removed from the set of active facets, it is necessary to reject any of the previously obtained displacement fields and restart from the original problem.

3.8 Comments on the algorithm

The weak enforcement of contact constraints requires the computation of integrals only on the non-mortar side. As such, in order to increase the accuracy of the method (but also its computational cost), in the choice of the non-mortar surface one should pick the boundary with the biggest number of facets. Indeed, if this is the case, the algorithm maximizes the number of points on which the projection operator is sampled. Since $p^{(0)}$ is only piecewise continuous, a globally higher number of quadrature points is the only non-adaptive procedure which can increase the accuracy of the solution without any mesh refinement.

Moreover, when solving the modified linear system (see Section 3.5), one should be careful to avoid numerical instability: in the penalty approach, a larger penalty parameter ε_{pen} yields more accurate results, but, in order to have a well-conditioned matrix, one should not choose it too big when compared with the elements of the stiffness matrix of the undeformed system; similarly, in the Lagrange multipliers approach, one has to be aware that the solution of the system has components with very diversified orders of magnitude, since the ratio between the values of the Lagrange multipliers and of the displacement field is usually (depending on the dimension of the system) comparable with the Young's modulus.

These two considerations are motivated by the fact that contact forces ($-\lambda \mathbf{n}$ or $\varepsilon_{pen} g_N^- \mathbf{n}$) compensate the internal stresses due to the deformation caused by the contact itself.

Another problem for the well-posedness of the method is the choice of the space Q^h in which to approximate the Lagrange multipliers (see Section 2.2.1). General theoretical results for mixed PDE (see e.g. [3]) show that Q^h should not be too large with respect to V^h (where the displacements are approximated): for example, in general, if Lagrange multipliers and displacements are approximated with piecewise polynomials of the same degree (on the same mesh), the algebraic problem deriving from the weak formulation is underdetermined. This has also been tested in MATLAB [9] and with the implementation described in Section 4: if linear finite elements are used for both λ and \mathbf{u} , the modified matrix appearing in (26) may be singular.

To tackle this issue, a safe solution involves the choice of different polynomial degrees for displacements and Lagrange multipliers, e.g. P^2 and P^1 respectively (see Section 5.3). Other solutions, which were not investigated by this project, require an accurate well-posedness analysis in the specific case of problem (23). Also, one may want to understand how the number of quadrature points n_p affects this issue: it has been observed empirically that lowering the number of quadrature points may (in some cases) lead from well-posed to ill-posed modified systems (26).

4 Implementation

The algorithm described in the previous section has been implemented in the framework provided by *Akantu*, an open-source object-oriented finite elements library [6]-[7].

The three central steps of the algorithm (facet coupling, computation of normals and evaluation of modified linear system) are coded in C++, using a object-oriented design which will allow an easier extension of the mortar method to other approaches (e.g. barrier and augmented Lagrange methods).

The execution of the other steps, including the two checks described in Sections 3.6 and 3.7, is carried out in Python to achieve a better efficiency in the solution of linear systems. Indeed, at the time of writing, one of the objectives of *Akantu* developers is the migration from C++ to Python for most linear algebra computations.

4.1 Object-oriented portion of the code

A general diagram representing the object-oriented portion of the code is shown in Figure 10.

The mother class `MortarSolver` contains all the features that are strictly needed for a mortar solver:

Members

- Solid mechanics model and mesh of the problem to which the mortar method is applied. Through the model, the method can obtain informations about the boundary conditions. Also, it can access functions provided by the finite element engine (FEEngine) of the model [8], such as interpolation on elements and computation of normals.
- List of non-mortar and mortar facets (see $\tilde{\Gamma}^{nm}$ and $\tilde{\Gamma}^m$ in Section 3.1). Each is stored as an `ElementGroup`, an *Akantu* type which allows elements to be indexed based on their `ElementType` (e.g. P^1 segments, P^2 triangles, etc.). These variables need to be initialized with “`initActiveArea`”.
- List of nodal positions and displacements. The former are used to store temporary changes in the undeformed configuration (see Section 3.6) without affecting the model (by using “`updatePositions`”). The latter are currently unused, but are designed to store the solution of the modified linear system.
- List of all the possible destinations for the projection operator for each non-mortar facet (see $\pi(\Gamma_i^{nm})$ in Section 3.2).

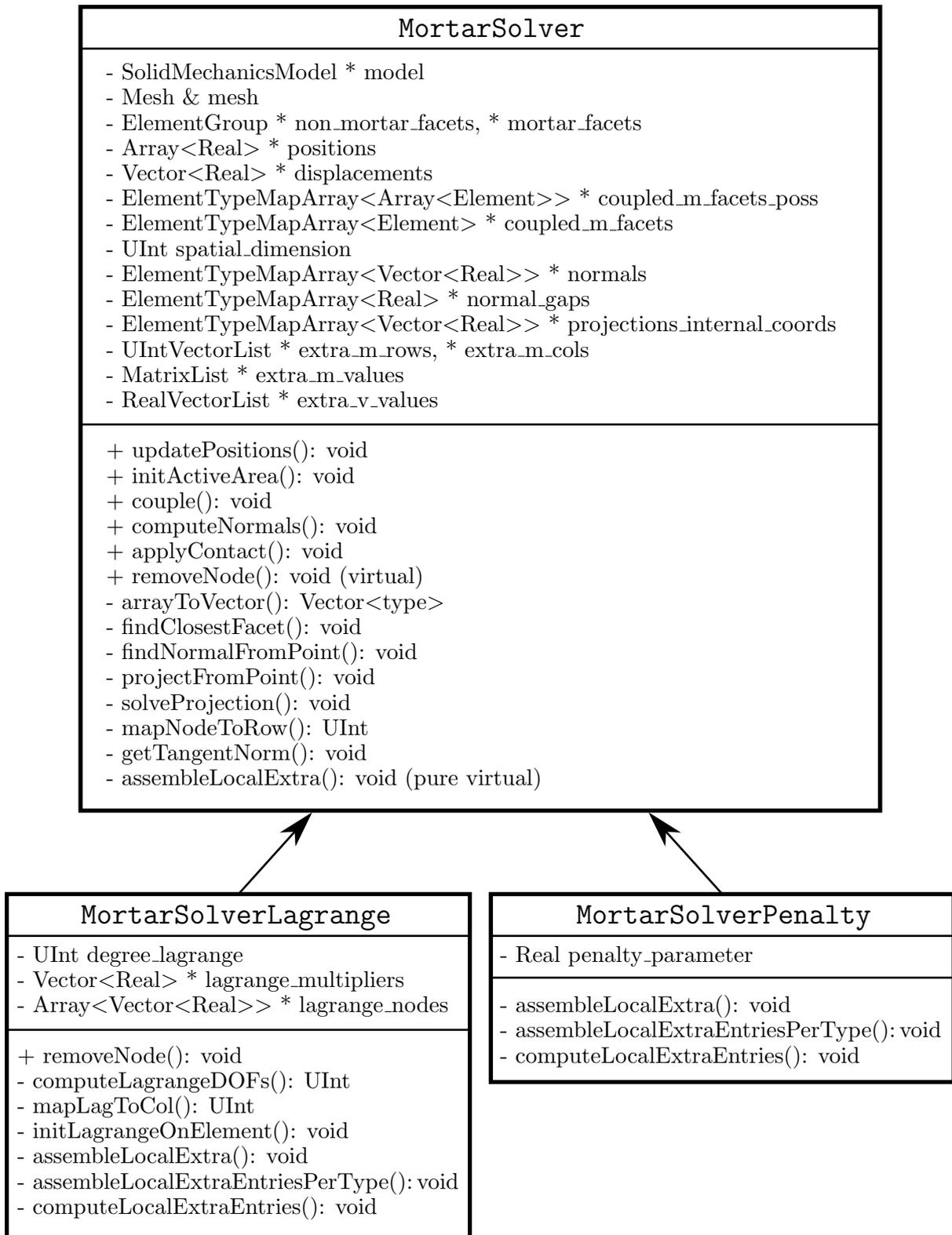


Figure 10: Class collaboration diagram for the C++ portion of the code.

- List of actual destinations for the projection operator for each quadrature point of the non-mortar facets (see $\Gamma_{\bar{j}}^m$ in Section 3.3).
- List of normals, normal gaps and natural coordinates of the projection for each quadrature point of the non-mortar facets (see $\overline{\mathbf{n}^{m,0}}(\overline{p^{(0)}}(\mathbf{X}_p))$, $g_N^{(0)}(\mathbf{X}_p)$ and $\zeta_p^{(0)}$ in Section 3.4).
- List of extra entries of the modified linear system (see Section 3.4), and corresponding position (rows and columns). The position of the entries is obtained by considering the global numbering of the nodes of each element.

Methods

- “couple” carries out the coupling step (see Section 3.2), by finding (with “findClosest-Facet”) the mortar facet which is the closest to each non-mortar node.
- “computeNormals” carries out the step described in Section 3.3: for each quadrature point on the non-mortar facet Γ_i^{nm} , it calls the function “findNormalFromPoint”, which tries to find the projection on all elements of $\pi(\Gamma_i^{nm})$ by using the sub-routine “projectFromPoint”. In particular the function “solveProjection” is applied to solve¹⁶ equation (50).
- “assembleLocalExtra” computes the extra contributions to the modified system, and must be implemented in the daughter classes. In particular “getTangentNorm” is used to compute the term $\|\mathbf{a}_p^{nm}\|$ in (56).
- “removeNode”, given a node which needs to be removed from the active area, reduces the set of non-mortar facets accordingly. It also resets the undeformed configuration (see Section 3.7). This method is overridden in the case of Lagrange multipliers, because some additional variables need to be reset.

The daughter class `MortarSolverLagrange` implements the Lagrange multipliers approach. As such it contains the additional members:

- Degree of the polynomials with which to approximate the Lagrange multipliers.
- Nodal values of the Lagrange multipliers. They are currently unused, but are designed to store a portion of the solution of the modified linear system.
- List of positions of Lagrange multipliers nodes, which may or may not coincide with nodes of the mesh. They are used to identify the columns in which extra entries of the linear system are located.

¹⁶At the moment of writing, the solver is able to find the solutions of the projection equations only in the few cases where they have a closed-form (e.g. P^1 elements in 2D and P^1 triangular elements in 3D).

The `MortarSolverPenalty` class implements the penalty approach, and has an additional member corresponding to the penalty parameter ε_{pen} .

Both daughters implement the methods which are used to compute the additional entries of the system: “assembleLocalExtra” and “assembleLocalExtraPerType” are used to cycle over all element types of the non-mortar facets; then “computeLocalExtraEntries” carries out the computations described in Section 3.4.

Additionally, the daughter implementing the Lagrange multiplier version needs some methods to manage properly the numbering of the Lagrange nodes and to compute the basis $\{M_I^f\}_{I=1}^{M^f}$ (see Section 2.2.1) on quadrature points.

4.2 Portion of the code in Python

The Python implementation has been based on examples included in the *Akantu* 2.3 distribution [7].

First, the system is initialized. In order to identify the active area, some flags (physical properties) are set while defining the geometry of the mesh [10]. In this way, portions of the boundary are given “non-mortar” and “mortar” status (see Section 3.1). Then the “createGroupsFromStringMeshData” function provided by *Akantu* [8] is used to create an `ElementGroup` for each side.

After the mortar solver has been declared and initialized, the iterative loop starts. In particular, a single while loop is used to carry out both iterative procedures:

- The convergence check (see Section 3.6) is performed by computing the L^2 -norm of the displacement vector. If convergence has not been reached yet, then a counter is incremented, the undeformed configuration is updated, and the loop is repeated.
- The active area check (see Section 3.7) is performed just after the solution of the modified linear system. The residual \mathbf{r} is computed and restricted to the non-mortar nodes. Now, if the minimum value of \mathbf{r} is negative and too big (in absolute value) when compared with the maximum residual, the counter described in the previous point is reset, along with the undeformed configuration. Also, the method “deleteNode” is called, so that the mortar solver may modify its internal list of active facets by removing the worst-behaving node (i.e. the one with the maximum traction).

5 Examples

5.1 Example 1: computation of the projection operator for linear finite elements in 2D

In order to provide a simple example of the application of the step described in Section 3.3, we consider the configuration shown in Figure 11, where linear finite elements are used to approximate the geometry of the problem. The possible destinations of the projection from \mathbf{x} have already been identified as $\pi(\Gamma_1^{nm}) = \{\Gamma_1^m, \Gamma_2^m\}$.

First we look for a projection on Γ_1^m . The distance function can be written as

$$d(\zeta; \mathbf{x}, \Gamma_1^m)^2 = \left\| \sum_{I=1}^2 N_I(\zeta) \mathbf{X}_I^m - \mathbf{x} \right\|^2 = \left\| \frac{1-\zeta}{2} \mathbf{X}_1^m + \frac{1+\zeta}{2} \mathbf{X}_2^m - \mathbf{x} \right\|^2 \quad (80)$$

where the usual lagrangian basis on linear elements has been used (the nodes coincide with the extreme points of the segment).

The stationary points are found by solving the following equation:

$$0 = \frac{\partial}{\partial \zeta} d(\zeta; \mathbf{x}, \Gamma_1^m)^2 = \left(\frac{1-\zeta}{2} \mathbf{X}_1^m + \frac{1+\zeta}{2} \mathbf{X}_2^m - \mathbf{x} \right) \cdot \frac{\mathbf{X}_2^m - \mathbf{X}_1^m}{2} \quad (81)$$

which leads to the solution

$$\hat{\zeta}_1 = \frac{2(\mathbf{X}_2^m - \mathbf{X}_1^m)}{\|\mathbf{X}_2^m - \mathbf{X}_1^m\|^2} \cdot \left(\mathbf{x} - \frac{\mathbf{X}_1^m + \mathbf{X}_2^m}{2} \right) \quad (82)$$

Since the solution is unique, we only need to check whether $\hat{\zeta}_1$ belongs to $\Omega_{ref}^{\Gamma_1^m} = [-1, 1]$. Let us suppose it does: then we take $\zeta_1 = \hat{\zeta}_1$ and

$$\mathbf{x}_1 := \sum_{I=1}^2 N_I(\zeta_1) \mathbf{X}_I^m$$

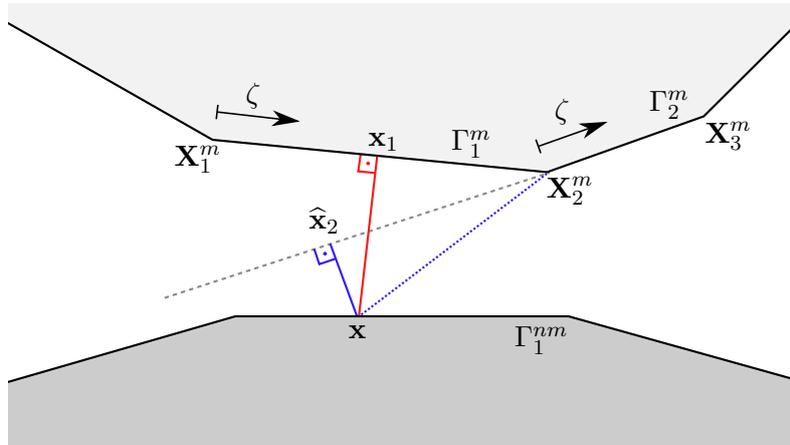


Figure 11: We want to identify the projection of \mathbf{x} on the mortar side. \mathbf{x}_1 is the projection onto Γ_1^m (which is the desired result). $\hat{\mathbf{x}}_2$ is the initial guess for the projection onto Γ_2^m , which is then replaced by the feasible $\mathbf{x}_2 = \mathbf{X}_2^m$.

as shown in Figure 11.

By proceeding similarly for Γ_2^m we obtain a stationary point

$$\hat{\zeta}_2 = \frac{2(\mathbf{X}_3^m - \mathbf{X}_2^m)}{\|\mathbf{X}_3^m - \mathbf{X}_2^m\|^2} \cdot \left(\mathbf{x} - \frac{\mathbf{X}_2^m + \mathbf{X}_3^m}{2} \right) \quad (83)$$

Now, if $\hat{\zeta}_2 < -1$, we pick $\zeta_2 = -1$ and $\mathbf{x}_2 := \mathbf{X}_2^m$ as shown in Figure 11.

The only step left is the comparison between $\|\mathbf{x}_1 - \mathbf{x}\|$ and $\|\mathbf{x}_2 - \mathbf{x}\|$, which leads to the choice of \mathbf{x}_1 as the desired projection.

5.2 Example 2: application of the mortar approach to a simple 2D problem

We consider a contact problem for two bodies in 2D. The undeformed configuration is shown in Figure 13. On Γ_D^{nm} (the right boundary of Ω^{nm}) and Γ_D^m (the left boundary of Ω^m) we enforce Dirichlet boundary conditions (respectively homogeneous and non-homogeneous), whereas on all other boundaries homogeneous Neumann conditions are applied. If no contact condition is enforced, the displacement field is uniform for each of the bodies (see Figure 14).

The penalty-based mortar method is applied to the problem, by using the implementation presented in Section 4. The initial guess for the active area is given by the left boundary of Ω^{nm} and the right boundary of Ω^m (respectively $\tilde{\Gamma}^{nm}$ and $\tilde{\Gamma}^m$ in Figure 13). The penalty parameter is $\varepsilon_{pen} = 10^7$ Pa·m, the Young's modulus of the bodies is $E = 3 \cdot 10^8$ Pa and a characteristic dimension of the system is $L = 1$ m.

After 202 iterations, the norm of the displacements becomes lower than the tolerance $\varepsilon_{tol} = 2 \cdot 10^{-6} \cdot (\text{number of nodes})$: hence the stopping condition is satisfied, and the algorithm ends (see Figure 15). Also, in order to understand how the iterations affect the solution, the evolution of the normal penetration near the bottom of the contact area is shown in Figure 12.

As a comparison, the solution obtained with the Lagrange multipliers approach is shown in Figure 16: in this case only 6 iterations were necessary.

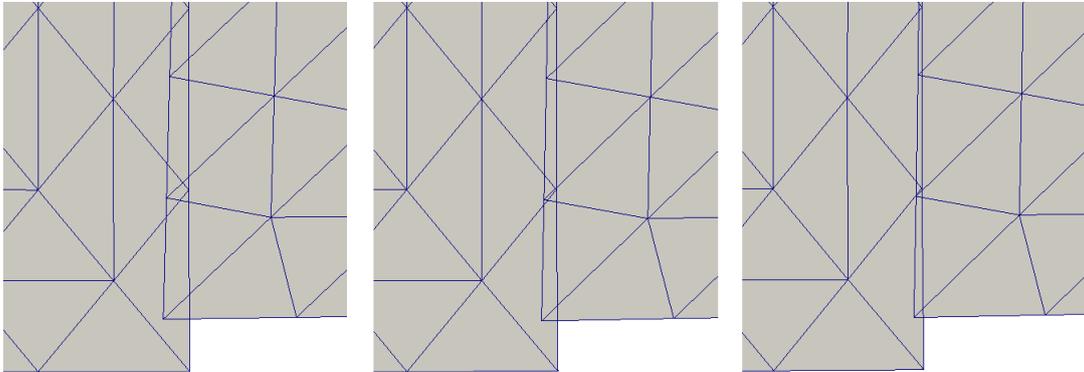


Figure 12: Zoom view of the lower part of the active area after 20 (left), 40 (middle) and 80 (right) iterations in the penalty approach.

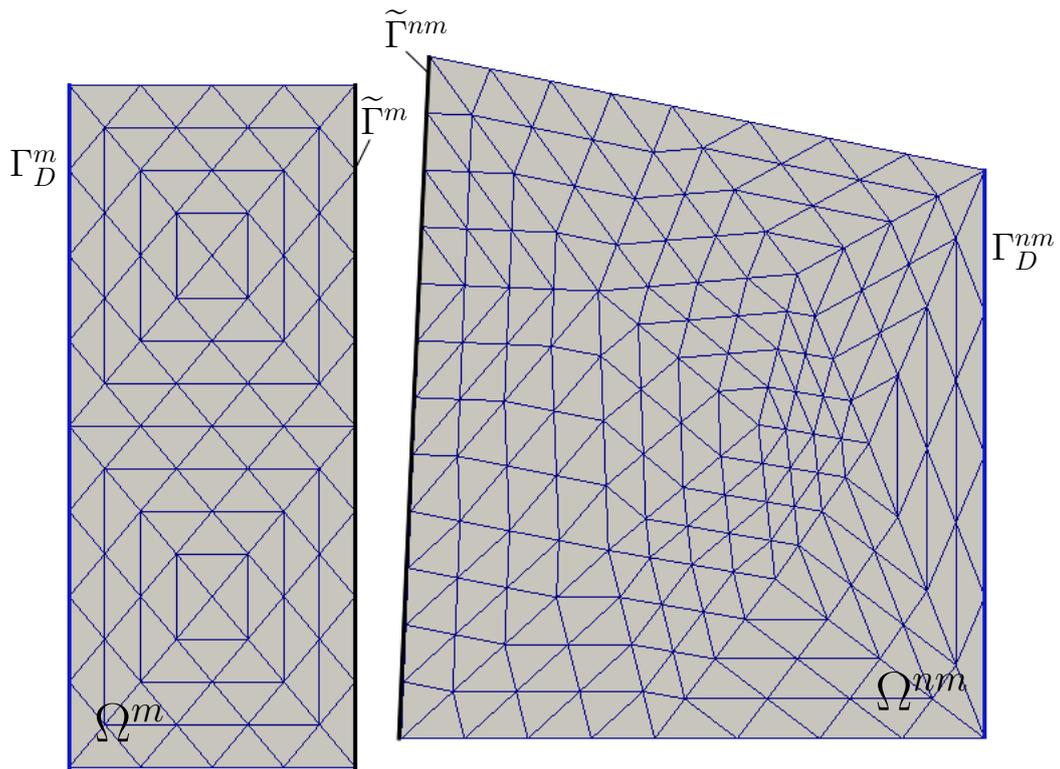


Figure 13: Undeformed configuration of the problem described in Example 2, discretized by a triangle-based linear mesh.

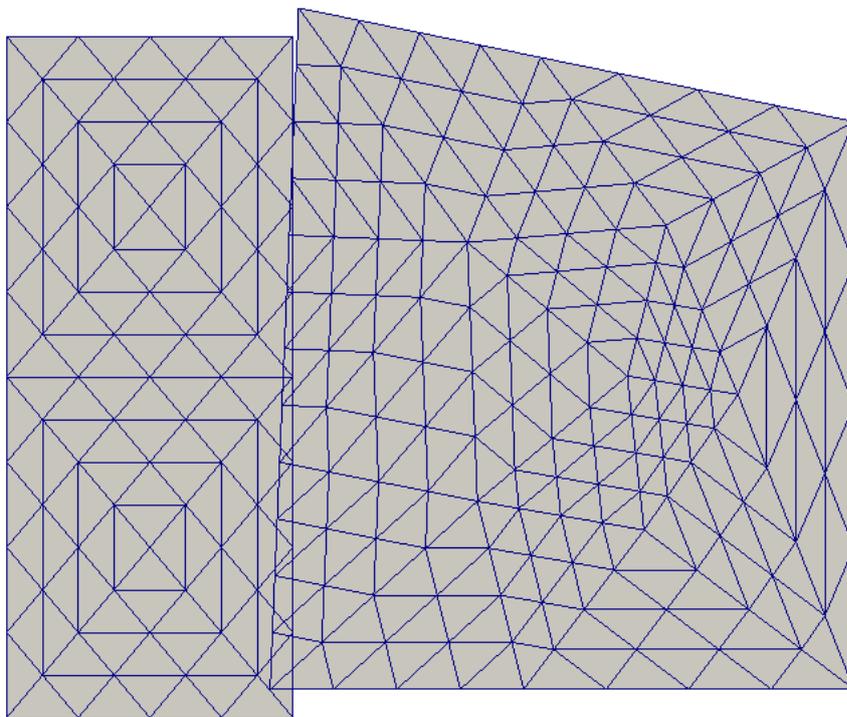


Figure 14: Solution of the problem described in Example 2, neglecting the contact constraints. Since no forces are present, the motion of each body is rigid.

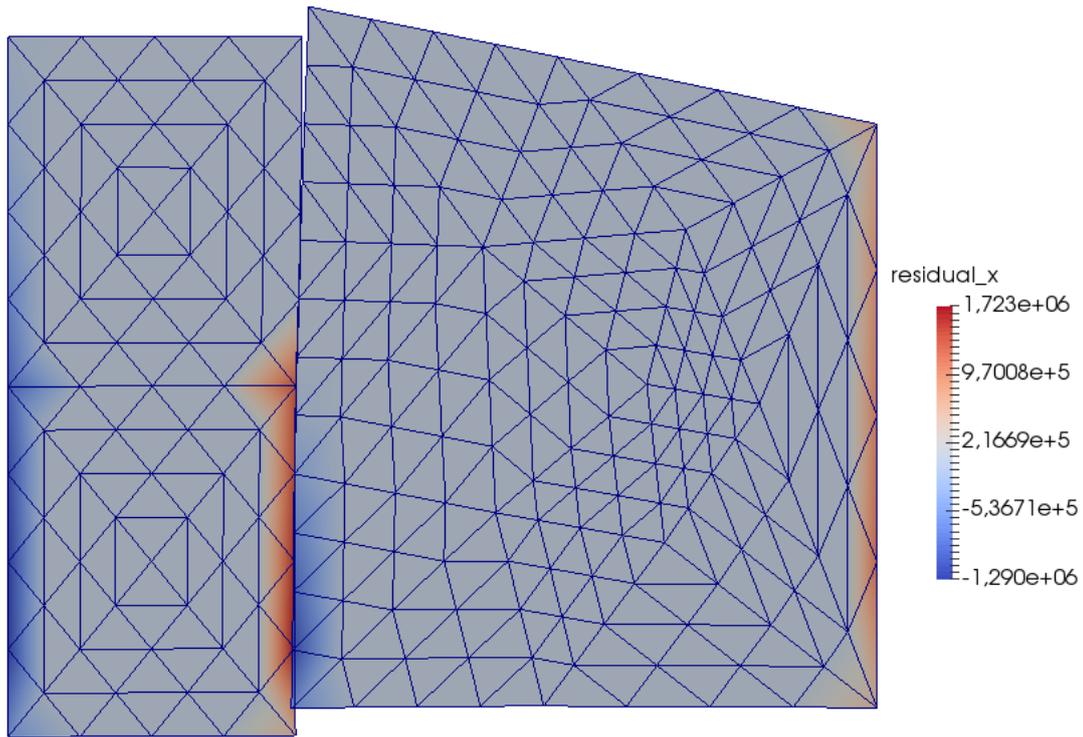


Figure 15: Solution of the problem described in Example 2, using the penalty approach to enforce contact constraints. In particular 4 non-mortar facets were removed from the active area. The horizontal component of the residuals is shown (see color scale on the side).

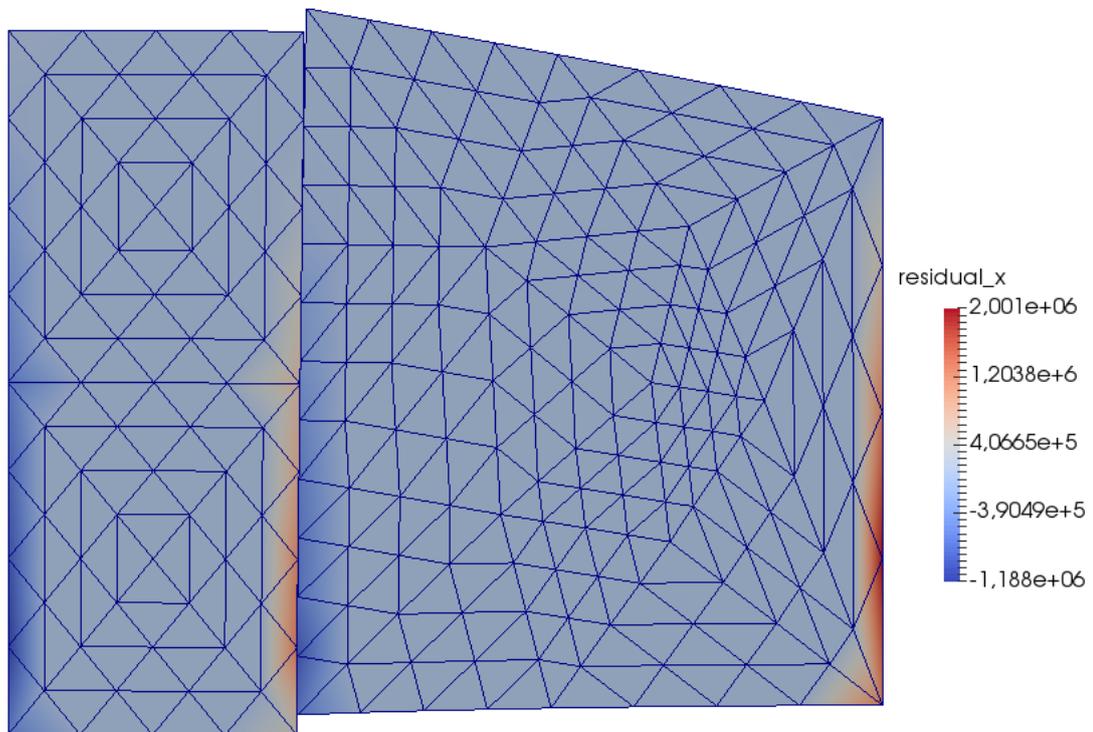


Figure 16: Solution of the problem described in Example 2, using the Lagrange multipliers approach to enforce contact constraints. In particular 3 non-mortar facets were removed from the active area. The horizontal component of the residuals is shown (see color scale on the side).

5.3 Example 3: application of the mortar method to a hertzian problem

We consider a contact problem between a deformable semicircle and a rigid plane (see Figure 17). All features of the solution (displacements, active area, contact pressure) can be approximated by applying the hertzian approach [4].

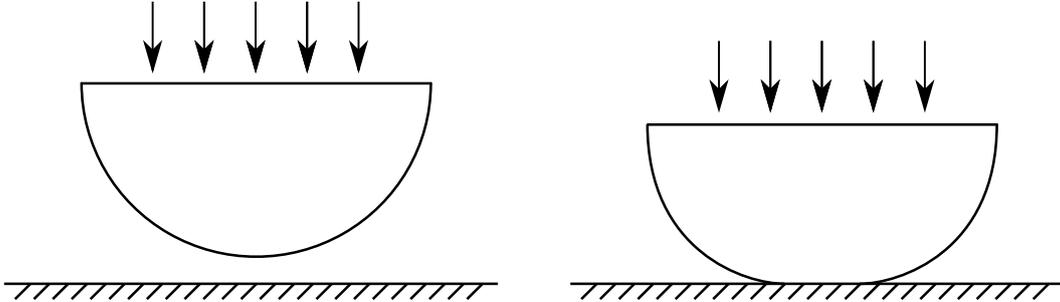


Figure 17: Undeformed (left) and deformed (right) configuration of the hertzian unilateral problem. The contact may be caused by an external force pushing down or by a fixed displacement of the top boundary.

In particular the expected profile of the contact pressure is non-differentiable at the extreme points of the contact area:

$$p(x) = p_0 \sqrt{1 - \left(\frac{x}{a}\right)^2} \quad (84)$$

where $2a$ represents the contact length and x is a local coordinate, which is taken equal to 0 on the symmetry axis of the system.

The mortar method is used to solve numerically a displacement-controlled hertzian problem. In particular a MATLAB [9] implementation of the method is applied to obtain the results presented in the following paragraphs.

We consider the triangle-based computational mesh shown in Figure 18: in particular, local refinements are used to get a higher accuracy near the contact zone. The radius of the semicircle is 1 m, and the Young's modulus of the material is $E = 3 \cdot 10^8$ Pa. Second degree (P^2) finite elements are applied to approximate geometry and displacements.

On Γ_D (the top boundary of the semicircle) non-homogeneous boundary conditions are enforced, whereas on the rest of the boundary homogeneous Neumann conditions are considered. In particular, if contact conditions are neglected, the displacement field is uniform, corresponding to pure translation (see Figure 19).

Since some interpenetration between the body and the wall appears (see Figure 19), it is necessary to apply the mortar method: in particular we choose to use the Lagrange multipliers approach. The Lagrange multipliers are approximated as a piecewise *linear* function (P^1) to

ensure the well-posedness of the numerical method (see Section 3.8).

As initial guess for the active area, we pick all facets which would overlap with the wall if no contact conditions were enforced ($\tilde{\Gamma}^{nm}$ in Figure 18).

After the first iteration, the pressure profile is the one shown in Figure 21a. Since the extreme values are too negative, one of the extreme facets of $\tilde{\Gamma}^{nm}$ is removed. After 8 iterations (and 8 removed non-mortar facets), the pressure profile has better features (see Figure 21b) and the algorithm ends. The final configuration is shown in Figure 20.

Similar results can be obtained with the penalty approach. However, from the empirical results it appears that the Lagrange multiplier method is able to provide a better approximation of the hertzian pressure profile.

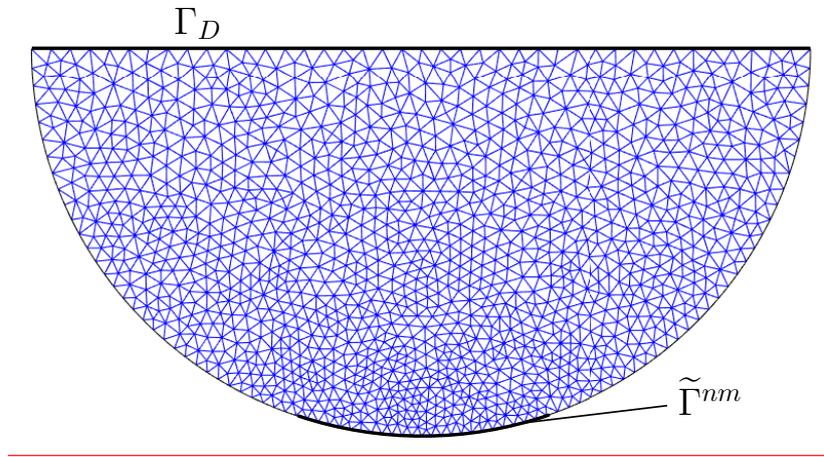


Figure 18: Undeformed configuration of the problem described in Example 3, discretized by a triangle-based quadratic mesh.

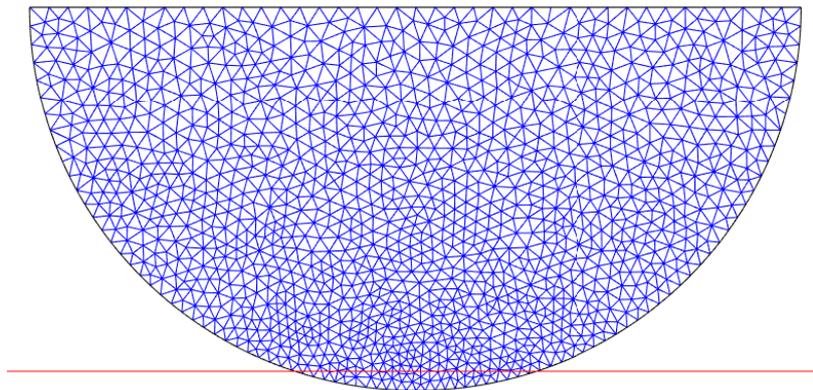


Figure 19: Solution of the problem described in Example 3, neglecting the contact constraints. Since no forces are present, the motion of the body is rigid.

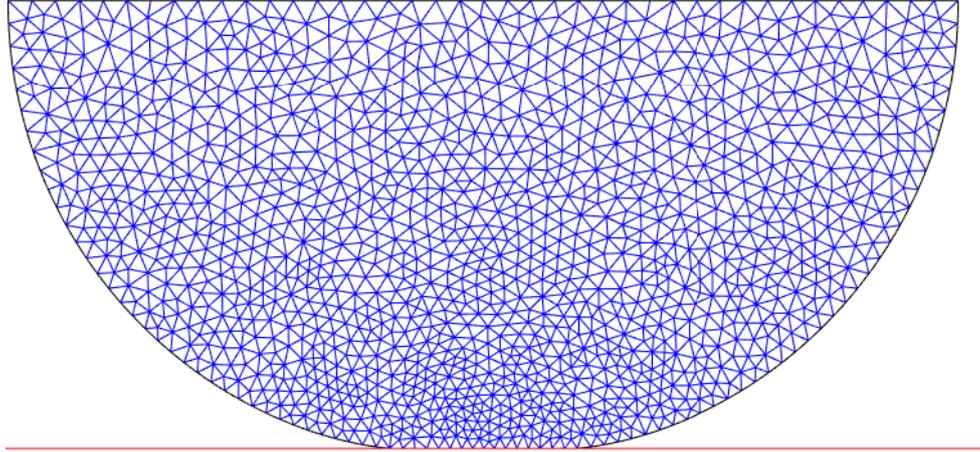


Figure 20: Solution of the problem described in Example 3, using the mortar method to enforce contact constraints.

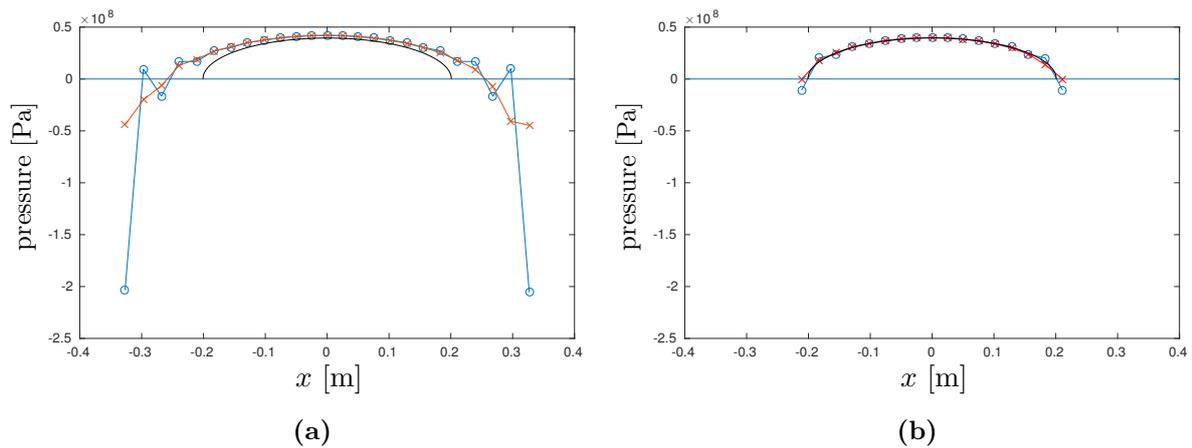


Figure 21: Contact pressure profile after the first iteration (a) and after the last iteration (b). In blue, the Lagrange multiplier profile. In red, the pressure obtained from the residual (see Section 3.7). In black the hertzian solution.

6 Conclusion and outlook

In this paper, the most important theoretical features of the mortar method have been described and rigorously derived, with a particular focus on the penalty and Lagrange multipliers approaches. A detailed description of an algorithm to apply the mortar method has been provided, as well as a brief overview of an implementation of said algorithm in C++ and Python.

The correctness of the implementation has been checked in some test cases. In particular, in the last example the numerical results appear to agree quite well with the analytical solution, in terms of contact pressure.

Some of the envisioned extensions of the code, which were not possible due to time constraints, are (from simplest to most complicated):

- Inclusion of the active area check (see Section 3.7) in the C++ portion in a object-oriented fashion, so that inheritance may be used to specialize this step in the penalty and Lagrange multipliers approaches¹⁷.
- Application of an iterative non-linear solver to find the stationary points in the computation of the projection operator (see Section 3.3) in cases in which closed-form solutions are not available.
- Implementation of *ad hoc* algorithms to solve the modified linear system by exploiting its particular structure (in particular in the Lagrange multipliers case).
- Enhancement of the robustness of the method in cases of non-smoothness of the boundaries, in particular in the normal computation step (see Section 3.3).
- Extension of the method to the case of dynamic analysis; a mortar-based predictor-corrector approach is necessary for the enforcement of contact constraints in this case.
- Extension of the method to the case of finite deformations; in particular, in this case the approximation (74) does not hold any more, and the algorithm needs to be specialized by applying an iterative Newton-Raphson approach [1].

¹⁷An estimate on the contact pressure may be found by looking at the penetration g_N (in the penalty case) or at the nodal values of the Lagrange multipliers (in the Lagrange multiplier case).

7 References

- [1] Wriggers P., “Computational Contact Mechanics”, Second edition, Springer, 2006.
- [2] Salsa S., “Partial Differential Equations in Action”, First edition, Springer, 2007.
- [3] Quarteroni A., “Numerical Models for Differential Problems”, Second edition, Springer, 2014.
- [4] Johnson K.L., “Contact mechanics”, Ninth printing, Cambridge University Press, 2003.
- [5] Pitt-Francis J., Whiteley J., “Guide to Scientific Computing in C++”, Springer, 2012.
- [6] Richart N., Molinari J.-F., “Implementation of a parallel finite-element library: Test case on a non-local continuum damage model”, *Finite Elements in Analysis and Design*, 100, p. 41-46, Elsevier Science Bv, 2015.
- [7] Laboratoire de Simulation en Mécanique des Solides, Station 18, EPFL, Lausanne, lsms.epfl.ch/akantu.
- [8] Laboratoire de Simulation en Mécanique des Solides, “Akantu. User’s Guide”, lsms.epfl.ch/files/content/sites/lsms/files/akantu_ug_v2.3.pdf, Version 2.3, EPFL, Lausanne, February 2016.
- [9] The Mathworks, Inc., Massachusset, USA, mathworks.com/products/matlab.
- [10] Geuzaine C., Remacle J.-F., “Gmsh Reference Manual”, gmsh.info/doc/texinfo/gmsh.pdf, Version 2.12, March 2016.