

RESEARCH ARTICLE

WILEY

Development of a semi-implicit contact methodology for finite volume stress solvers

Alessandro Scolaro¹  | Carlo Fiorina¹ | Ivor Clifford² | Andreas Pautz^{1,2}

¹Laboratory for Reactor Physics and Systems Behaviour, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, 1015, Switzerland

²Nuclear Energy and Safety, Paul Scherrer Institute (PSI), Villigen, 5232, Switzerland

Correspondence

Alessandro Scolaro, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Email: alessandro.scolaro@epfl.ch

Abstract

The past decades have seen numerous efforts to apply the finite volume methodology to solid mechanics problems. However, only limited work has been done by the finite volume community toward the simulation of mechanical contact. In this article, we present a novel semi-implicit methodology for the solution of static force-loading contact problems with cell-centered finite volume codes. Starting from the similarities with multi-material problems, we derive an implicit discretization scheme for the normal contact stress with a straightforward inclusion of frictional forces and correction vectors for non-orthogonal boundaries. With the introduction of a sigmoid blending function interpolating between contact stresses and gap pressure, the proposed approach is extended to cases with partially closed gap. The contact procedure is designed around an arbitrary mesh mapping algorithm to allow for non-conformal meshes at the contact interface between the two bodies. Finally, we verify the contact methodology against five benchmarks cases.

KEYWORDS

contact, finite volume methods, solid mechanics

1 | INTRODUCTION

The study of the mechanical contact between two bodies is a fundamental aspect of computational solid mechanics (CSM) and it plays a crucial role in several engineering applications,¹ from metal forming to the deformation of structures' foundations, vehicle crash simulations, biomechanics, and countless more. The numerical solution to contact problems can be challenging, as the mathematical treatment of contact mechanics introduces a severely discontinuous boundary nonlinearity, even for the case of simple linear elasticity.²

Numerous researchers have investigated this topic over the years and most of the efforts have been carried out employing the finite element (FE) methodology, which as of today is the most popular numerical scheme in CSM. Several methods have been developed to enforce the contact constraints in the FE formulation with the most common being the Lagrange multiplier method, the augmented Lagrange multiplier methods, the penalty method and the mortar method.³ A comprehensive review of the FE computational treatment of contact problems can be found in the works from Laursen,⁴ Wriggers,¹ Zavarise and Wriggers,⁵ and De Lorenzis et al.² An overview of the more recent contributions in the field can be found in the work from Popp and Wriggers.⁶

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2021 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons Ltd.

In the last decades, the finite volume (FV) methodology has become a viable alternative to FE modeling for the solution of solid mechanics problems. Limiting the scope to the cell-centered FV approach also followed in this article, one should certainly mention the works of Demirdžić et al.,^{7,8} Jasak and Weller,⁹ Tuković and Jasak,¹⁰ Maneeratana,¹¹ and Cardiff et al.¹² Parallel research directions have investigated the use of other FV variants based on a different grid arrangement, such as the vertex-centered,¹³ the staggered grid,¹⁴ the meshless,¹⁵ or the face-centered¹⁶ approaches. A more general overview of the efforts carried out over the last 30 years to employ the FV methodology in CSM applications can be found in the recent review from Cardiff and Demirdžić.¹⁷

A first cell-centered FV contact stress solver was developed by Jasak and Weller¹⁸ in 2000. They complemented their basic linear elastic solver already discussed in Reference 9 with an additional contact boundary condition, capable of detecting the overlapping portion of the two facing surfaces. Their contact methodology followed a Dirichlet–Neumann coupling approach commonly used in partitioned fluid–structure interaction procedures: they prescribed the normal component of the displacement on one surface while transferring the force on the other surface. The contact algorithm discussed in Reference 18 was tested on a frictionless case but was not extended to deal with nonzero friction.

Cardiff et al.¹⁹ pointed out that the solver developed by Jasak and Weller, was effective in two-dimensions (2D), but it produced unrealistic stress peaks close to the contact region in three-dimensions (3D). Moving from this limitation, Cardiff developed an alternative frictionless contact boundary condition based on the penalty-method. The implementation of the penalty method was explicit, that is, the boundary contact stress is treated as a source term, and it is calculated based on the displacements obtained at the previous iteration. This boundary condition was later extended to include a Coulomb friction component²⁰ and it was implemented in a FV toolbox for solid mechanics and fluid–solid interaction named solids4Foam,²¹ valid both for small strain and finite strain simulations. The contact methodology developed by Cardiff was tested on several benchmark cases, and always compared well against the available analytical solutions or reference numerical solutions obtained with commercial FE codes. Recent efforts to improve the standard penalty method approach or create a valid alternative can be found in the works of Škurić et al.²² or Batistić et al.²³ A few examples of other important contributions to the treatment of contact mechanics with the FV (outside the cell-center approach considered in this text) are the works from Taylor et al.,²⁴ Bessonov et al.,²⁵ and Suliman et al.²⁶

The present work was motivated by the simulation of the thermo-mechanical contact taking place in the fuel rods of commercial nuclear power plants.^{27–29} Treating the contact between the ceramic oxide fuel pellets and the slender metallic cladding is a complex task. One must solve for a *force-loading* scenario (as opposed to a *displacement-loading* scenario where all boundary displacements are fixed), that involves multiple free bodies in contact deforming under the influence of thermal expansion and plastic strains, and a gap size that evolves over time, with possible gap reopenings. Also, the time step typically considered in nuclear fuel rod simulation are quite large (they can vary from ~ 1 to $\sim 10^5$ s) and the inertial forces that might otherwise help stabilizing the solution are neglected. For a similar scenario, the penalty method as developed by Cardiff results in an ill-conditioned problem, difficult if not impossible to solve. In some cases, convergence can be reached but only if the solution of the linear system resulting from the discretization of the governing equations is heavily under-relaxed. In the experience of the authors, this under-relaxation technique, combined with the level of mesh refinement typically necessary to obtain an accurate solution with the penalty method, can make the simulation extremely slow.

The objective of this article is to present an alternative methodology to solve static force loading contact problems within a cell-centered FV framework. Partly building on some of the ideas from the original solver from Jasak and Weller, we added the constraint that when two bodies are in contact, they share at the interface not just the traction but also the normal component of the displacement. As in the treatment of multi-material bodies developed by Tuković et al.,³⁰ we derive an implicit discretization of the contact stresses, expressed in terms of the current value of the displacement field. In this work, we introduce the methodology for the simple case of a small-strain stress solver, implemented, and tested within the OpenFOAM framework.³¹ Despite being born from the needs of the nuclear fuel modeling community, the methodology here presented can be of general interest for treating complex force-loading contact situations using finite volume discretization techniques.

This article is structured as follows. First, the fundamental aspects of stress analysis within cell-centered FV codes are recalled in Section 2. Then, the novel contact methodology proposed in this article is described in detail in Section 3. Finally, the methodology is validated against five benchmark cases with known solution in Section 4.

2 | FUNDAMENTAL ASPECTS OF STRESS ANALYSIS WITHIN FINITE VOLUME CODES

The contact methodology presented in this article operates on a finite volume framework for solid mechanics developed in the previously mentioned work by Demirdžić et al. and Tuković and Cardiff. A complete description of the domain and equation discretization, numerical schemes and solution procedures can be found in their respective publications.^{18–21,30} Nevertheless, the fundamentals of stress analysis with cell-centered FV codes are recalled in this section to ease the understanding of the proposed contact methodology.

2.1 | Mathematical model

For a solid with volume V bounded by a surface S with outward normal \mathbf{n} , the strong integral form of the momentum conservation equation is given by:

$$0 = \int_V \nabla \cdot \boldsymbol{\sigma} dV = \oint_S \mathbf{n} \cdot \boldsymbol{\sigma} dS, \quad (1)$$

where the inertia and body forces have been neglected for clarity of exposition. The Cauchy stress tensor $\boldsymbol{\sigma}$ for an isothermal linear-elastic material is given by the Hooke's law constitutive relation:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \lambda\text{tr}(\boldsymbol{\varepsilon})\mathbf{I}, \quad (2)$$

where μ and λ are the Lamé coefficients. The small-strain tensor $\boldsymbol{\varepsilon}$ is given by:

$$\boldsymbol{\varepsilon} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T], \quad (3)$$

where \mathbf{u} is the total displacement. Using the relations above, the surface traction vector \mathbf{t} is defined as:

$$\mathbf{t} = \mathbf{n} \cdot \boldsymbol{\sigma} = \mu \mathbf{n} \cdot \nabla \mathbf{u} + \mu \mathbf{n} \cdot (\nabla \mathbf{u})^T + \lambda \text{tr}(\nabla \mathbf{u}) \mathbf{n}. \quad (4)$$

The definition of the mathematical problem is completed by specifying appropriate initial and boundary conditions, usually displacement, traction, or symmetry.

2.2 | Domain discretization

In FV codes, the domain is discretized into convex polyhedral control volumes (CVs) that do not overlap and fill the domain completely, and the mathematical problem represented by Equation (1) is solved by applying the governing equation to each CV. The typical FV discretization considers a generic cell of volume V_P centered around the point P and separated from the neighboring cell centered in N by the distance vector $\boldsymbol{\delta}_f$. The face S_f shared between the two cells is centered around the point f and has outward normal \mathbf{n}_f . Further details on the discretization of the spatial domain are given in References 9 and 20.

The CV bounding surface S_P can be expressed as a combination of internal faces (f), boundary faces (b) where the traction, the displacement, or a combination of both are fixed, and faces that can potentially contact a separate solid body (c):

$$S_P = \sum_f S_f + \sum_b S_b + \sum_c S_c. \quad (5)$$

Depending on the CV location in the domain, one or more of the boundary face subgroups might be empty and if the cell is internal to the domain there are no boundary or contact faces for the considered CV.

2.3 | Numerical method and solution procedure

The surface integrals in Equation (1) applied to V_P are substituted with the sum of products over the cell faces calculated at the face center with the mid-point rule:

$$0 = \oint_{S_P} \mathbf{n} \cdot \boldsymbol{\sigma} dS = \oint_{S_P} \mathbf{t} dS = \sum_f \mathbf{t}_f S_f + \sum_b \mathbf{t}_b S_b + \sum_c \mathbf{t}_c S_c. \quad (6)$$

The discretization of each traction above will be discussed in more details in the sections that follow. In general, the discretization of the traction produces various terms designated as:

- *implicit*, if they depend on the cell-center values at the current iteration,
- *explicit*, if they depend on the cell-center values at the previous iteration, or
- *fixed sources*.

Collecting all these terms, Equation (6) is discretized into the following algebraic form:

$$a_P \mathbf{u}_P + \sum_N a_N \mathbf{u}_N + \sum_{P_{(m \text{ or } s)}} a_{P_{(m \text{ or } s)}} \mathbf{u}_{P_{(m \text{ or } s)}} = \mathbf{b}_P, \quad (7)$$

where the neighboring cells from the same body are centered in N , while those from a separate body in contact are centered in P_m or P_s . The subscripts m and s refer to the *master* and *slave* side of the gap and will be described more in detail in Section 3. The coefficients a_P , a_N , and $a_{P_{(m \text{ or } s)}}$, as well as the source term \mathbf{b}_P are discussed in the following sections for each respective traction. Fixed terms and explicit contributions deriving from the boundary conditions are also lumped in \mathbf{b}_P .

The system of equations resulting from the complete discretization for each CV might be written in a more compact way as:

$$[A][\mathbf{u}] = [\mathbf{b}], \quad (8)$$

where $[A]$ is the coefficient matrix collecting the diagonal coefficients a_P and off-diagonal coefficients a_N and $a_{P_{(m \text{ or } s)}}$, and $[\mathbf{b}]$ is the vector collecting all the source terms \mathbf{b}_P .

Although efforts are being made in the OpenFOAM community toward the development of block-coupled solvers for solid mechanics applications in which the three components of displacement are solved as a single linear system,^{12,32} the most common solution approach and the one adopted in this article is the segregated solution scheme. The system is decomposed into its three components (one per Cartesian direction) which are solved separately and sequentially, and the dependence on other components is added to the explicit terms.

Because of the presence of explicit terms in the system in Equation (8), it is necessary to iterate the solution until a user pre-defined convergence criterion is met. In this work, the system is solved using the geometric agglomerated algebraic multigrid (GAMG) linear solver.^{33,34} The solution is iterated until the normalized L1-norm of the residual vector, or more simply the residuals r , falls below a user defined threshold, which is typically between 10^{-6} and 10^{-9} . The residuals are defined as:

$$r = \frac{1}{n} \sum ||[\mathbf{b}] - [A][\mathbf{u}]||, \quad (9)$$

where n is a normalization factor.

2.3.1 | Discretization of traction over internal faces

In the cell-centered FV approach for solid mechanics originally deriving from the work of Demirdžić et al.⁷ and then further developed by Jasak and Weller,⁹ the surface traction on internal faces is split into an implicit part that directly contributes to the coefficient matrix $[A]$ and an explicit part which contributes to the source of the linear system $[\mathbf{b}]$.

To improve the convergence of the discretization procedure, Equation (4) applied to the internal faces is manipulated as follows:

$$\mathbf{t}_f = \overbrace{(2\mu + \lambda)_f \mathbf{n}_f \cdot (\nabla \mathbf{u})_f}^{\text{implicit}} + \overbrace{\mu_f \mathbf{n}_f \cdot (\nabla \mathbf{u})_f^T + \lambda_f \text{tr}(\nabla \mathbf{u})_f \mathbf{n}_f - (\mu + \lambda)_f \mathbf{n}_f \cdot (\nabla \mathbf{u})_f}^{\text{explicit}}. \quad (10)$$

As done by Cardiff et al.,²¹ the equation above can be rewritten in the following equivalent form:

$$\mathbf{t}_f = \overbrace{(2\mu + \lambda)_f \mathbf{n}_f \cdot (\nabla \mathbf{u})_f}^{\text{implicit}} + \overbrace{\mathbf{n}_f \cdot \boldsymbol{\sigma}_f - (2\mu + \lambda)_f \mathbf{n}_f \cdot (\nabla \mathbf{u})_f}^{\text{explicit}}. \quad (11)$$

The main advantage of Equation (11) with respect to Equation (10) is that the expression above is not limited to isothermal linear-elastic materials. By appropriately defining the stress and strain tensors, Equation (11) can be used also in the presence of additional strain components, such as thermal or plastic strains.

The surface normal gradient in the implicit term is discretized into an implicit orthogonal and an explicit non-orthogonal component, as follows:

$$\mathbf{n}_f \cdot (\nabla \mathbf{u})_f = \overbrace{|\Delta_f| \frac{\mathbf{u}_N - \mathbf{u}_P}{|\delta_f|}}^{\text{orthogonal contribution}} + \overbrace{\mathbf{k}_f \cdot (\nabla \mathbf{u})_f}^{\text{non-orthogonal correction}}, \quad (12)$$

where $\Delta_f = \frac{\delta_f}{\mathbf{n}_f \cdot \delta_f}$ is the orthogonal contribution vector, δ_f is the distance vector between the cell-center P and the center N of the neighbor cell, and is \mathbf{k}_f the non-orthogonal correction vector as shown in Figure 1.

The cell-center gradient of the displacement field, which is needed to evaluate the stress field according to Equations (2) and (3) are calculated using a least squares approach.³⁵ This makes the gradient second order in space irrespective of the mesh and it is particularly relevant to avoid large errors in the stress field in the presence of skewed meshes. However, the stresses and the gradients in the explicit terms of Equation (11) are thereafter interpolated to the faces using linear interpolation as this has a smaller effect on the solution accuracy.

Based on Equations (11) and (12), the following diagonal coefficient, off-diagonal coefficient, and source term are obtained from the discretization of the traction over the internal faces:

$$a_P = -|\Delta_f| \frac{(2\mu + \lambda)_f}{|\delta_f|} S_f, \quad (13)$$

$$a_N = |\Delta_f| \frac{(2\mu + \lambda)_f}{|\delta_f|} S_f, \quad (14)$$

$$\mathbf{b}_P = S_f \mathbf{n}_f \cdot \boldsymbol{\sigma}_f - (2\mu + \lambda)_f S_f \mathbf{n}_f \cdot \nabla \mathbf{u}_f + (\mathbf{n}_f - \Delta_f) \cdot (\nabla \mathbf{u})_f. \quad (15)$$

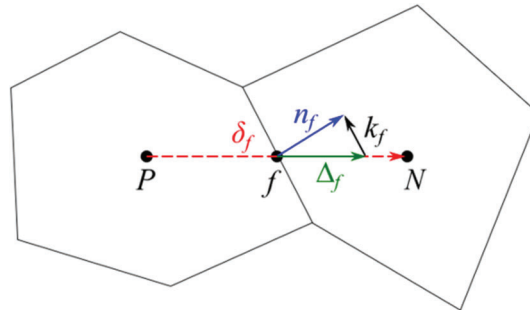


FIGURE 1 Non-orthogonal correction vectors

The material properties are assumed continuous and smooth, therefore we apply linear interpolation to obtain the value $(2\mu + \lambda)_f$ for the internal faces.

2.3.2 | Discretization of traction over boundary faces

The discretization of the traction on boundary faces τ_b depends on the specific boundary condition used. For fixed displacement (Dirichlet) boundaries, the traction is decomposed as done in Equations (11) and (12) for the internal faces:

$$\mathbf{t}_b = (2\mu + \lambda)_b \frac{\mathbf{u}_b - \mathbf{u}_p}{|\delta_b|} + \mathbf{n}_b \cdot \boldsymbol{\sigma}_b - (2\mu + \lambda)_b \mathbf{n}_b \cdot \nabla \mathbf{u}_b, \quad (16)$$

where δ_b is the distance vector between cell-center and boundary face-center, and \mathbf{n}_b is the outward normal of the boundary face with area S_b . The value \mathbf{u}_b at the boundary is known and the discretization leads to the following diagonal coefficient and source term:

$$a_p = -\frac{(2\mu + \lambda)_b}{|\delta_b|} S_b, \quad (17)$$

$$\mathbf{b}_p = (2\mu + \lambda)_b S_b \frac{\mathbf{u}_b}{|\delta_b|} + S_b \mathbf{n}_b \cdot \boldsymbol{\sigma}_b - (2\mu + \lambda)_b S_b \mathbf{n}_b \cdot \nabla \mathbf{u}_b. \quad (18)$$

For fixed-traction boundaries, the traction τ_b is known. Therefore, the diagonal and off-diagonal coefficients are null, while the source term is equal to:

$$\mathbf{b}_p = \mathbf{t}_b S_b. \quad (19)$$

The definition of the traction is still used to calculate the updated boundary displacement \mathbf{u}_b . From Equation (11), with minor manipulations, the normal gradient at the current iteration i is obtained as:

$$(\mathbf{n}_b \cdot \nabla \mathbf{u}_b)^i = \frac{(\mathbf{t}_b - \mathbf{n}_b \cdot \boldsymbol{\sigma}_b^{i-1})}{(2\mu + \lambda)_b} + (\mathbf{n}_b \cdot \nabla \mathbf{u}_b)^{i-1} \quad (20)$$

In turn, the normal gradient is used to calculate the updated value of the boundary displacement:

$$\mathbf{u}_b^i = \mathbf{u}_p^{i-1} + |\delta_{pb}| (\mathbf{n}_b \cdot \nabla \mathbf{u}_b^i). \quad (21)$$

3 | PROPOSED METHODOLOGY FOR CONTACT BOUNDARY CONDITIONS

We consider two separate solids that can potentially enter in contact through the section S_c of their respective boundaries. We assume that the two bodies might be submerged in a fluid with a known pressure p_g . The two opposing surfaces delimit a region of space called the *gap*, which is either *closed*, as in case of contact, or *open*. The two boundaries are typically designated as *master* (subscript m) and *slave* (subscript s) and must obey the contact constraint conditions. In the remaining of this article, quantities are designated with subscripts cm or cs if they belong to the master or slave surface, respectively.

Traditionally, contact problems in FV codes are solved employing the penalty method.^{19,20,36} Once a slave face-center is detected to penetrate a master face, a local interfacial pressure p_i is calculated as:

$$p_i = \left(\gamma \frac{A}{V} K \right) |g|, \quad (22)$$

where A and V are the master cell surface area and volume, K is the master bulk modulus and $\gamma \geq 0$ is the so-called *penalty scale factor* which is introduced to prevent numerical instability at the expense of a larger penetration. The local gap width g (negative in case of penetration) is calculated as:

$$g = [(\mathbf{f}_{cm} + \mathbf{u}_{cm}) - (\mathbf{f}_{cs} + \mathbf{u}_{cs})] \cdot \mathbf{n}_{cm}, \quad (23)$$

where \mathbf{f}_{cm} and \mathbf{f}_{cs} are the position vectors associated with the master and slave face-center points. Neglecting friction, the local contact traction is equal to:

$$\mathbf{t}_c = \begin{cases} -p_i \mathbf{n}_c & \text{if } g < 0, \\ -p_g \mathbf{n}_c & \text{if } g \geq 0, \end{cases} \quad (24)$$

where the same known gap/fluid pressure p_g is applied to the two opposing surfaces when the gap is open. If no fluid is considered, the gap pressure is set to zero so that the surface traction is zero when the surfaces are not in contact. The penalty method is explicit: the traction is added directly to the source term as done in Equation (19) for the fixed-traction boundary type, while the corresponding normal gradient is obtained once again from Equation (20).

We developed an alternative *implicit contact methodology* moving from the consideration that two bodies in contact share strong similarities with bi-material solids. Indeed, when the two opposing surfaces are in contact, they share not just the same traction but also the same position vector, that is, the total displacement summed to the initial position vector, while the gradient of the displacement normally remains discontinuous across the contact interface due to possibly different thermo-mechanical properties. The main differences with bi-material cases are that the contact interface cannot sustain positive traction and that the tangential component of the contact traction might be transmitted only partially, depending on the friction characteristics of the two surfaces.

Imposing the use of conformal meshes would be too restrictive, in particular for complex geometries. For this purpose, the new contact methodology is designed using the arbitrary mesh interpolation (AMI)³⁷ method as implemented in OpenFOAM,³⁸ although the extension to the general grid interface (GGI)³⁹ mapping method as implemented in the foam-extend fork of OpenFOAM⁴⁰ would be straightforward.

Figure 2 shows the contact between two identical boundaries, with the surface designated as slave having twice the number of faces as the surface designated as master. When the traction is discretized on one side of the contact interface, the AMI mapping algorithm reconstructs a single opposing cell with its face-center, cell-center and related quantities. As an example, on the right of Figure 2 the slave cells centered in P_s and with volume V_{P_s} are reconstructed from the point of view of the master boundary faces, attached to cells centered in P_m and with volume V_{P_m} . The specific interpolation algorithm used in this article is the *face-area-weighting* technique, which works as follows: For each face on either side of the gap, the AMI calculates the normalized area fractions covered by the projection of the opposing faces; any quantity on the slave side can then be interpolated to the master side and vice versa with a simple weighted average, using the previously calculated area fractions as weights.

In the sections that follow, we derive the contact force discretization always from the point of view of the master surface, that is, when dealing with non-conformal meshes it is always implied that the slave cell in P_s is the reconstructed

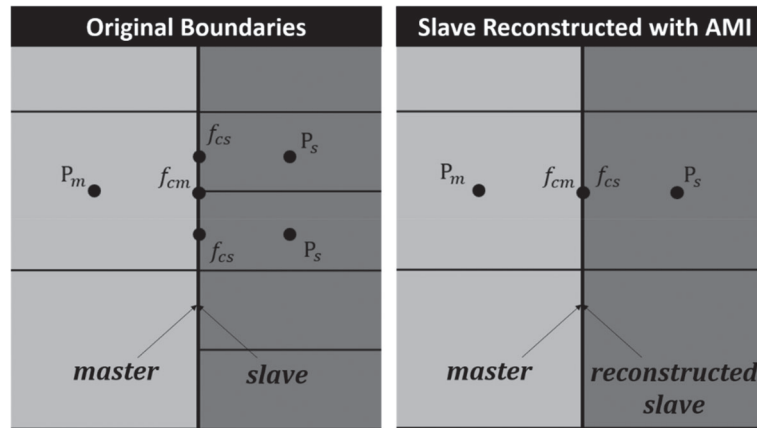


FIGURE 2 Non-conformal master and slave surface in contact (left). The mapping algorithm reconstructs a virtual opposing slave cell for each face on the master side (right), and vice versa

opposing cell as determined by the AMI method. The derivation would be identical if obtained from the point of view of the slave surface, and it is not repeated to avoid making the text unnecessarily cumbersome to read.

We describe the new contact methodology in the next six sub-sections, gradually increasing the complexity of the contact problem considered. First, in Section 3.1, we derive the contact traction for the simple case of fully closed gap for two identical boundaries with orthogonal meshes. Then, in Section 3.2, we add the correction term for non-orthogonal boundaries, which are necessary in the case of non-conformal boundaries. With the introduction of a blending function that interpolates between contact stresses and gap pressure, we extend the contact methodology to scenarios with fully or partially open gap in Section 3.3, and we describe the characteristics of the specific blending function adopted in this work in Section 3.4. Finally, in Section 3.5, we show how to extend the methodology to the more general case of boundaries with different geometry and orientation.

3.1 | Implicit contact with closed gap, identical boundaries and orthogonal meshes

As shown in Figure 3, we start by considering the two cells in contact V_{P_m} and V_{P_s} in the case of conformal meshes. We assume also that the two outward normal vectors lie in the same direction, that is, $\mathbf{n}_{cm} = -\mathbf{n}_{cs}$, and are parallel to δ_{cm} and δ_{cs} , defined as the cell-center to face-center distance vectors on the master and slave side, respectively.

Following the derivation described by Tuković et al.³⁰ for internal interfaces (see also Appendix A in Reference 12 for the complementary derivation), the traction vector at the contact interface \mathbf{t}_c can be decomposed into normal and tangent components:

$$(\mathbf{t}_n)_c = (2\mu + \lambda)_c \mathbf{n}_c \cdot (\nabla \mathbf{u}_n)_c + \lambda_c \text{tr}(\nabla_t \mathbf{u}_t)_c \mathbf{n}_c, \quad (25)$$

$$(\mathbf{t}_t)_c = \mu_c \mathbf{n}_c \cdot (\nabla \mathbf{u}_t)_c + \mu_c (\nabla_t \mathbf{u}_n)_c, \quad (26)$$

where the tangent gradient operator $\nabla_t = (\mathbf{I} - \mathbf{nn}) \cdot \nabla$ is introduced, and the subscripts n and t refer to the normal and tangent components of traction and displacement.

The normal component of the traction on the two opposite sides of the contact interface S_c can be discretized as follows:

$$(\mathbf{t}_n)_{cm} = \frac{(\mathbf{u}_n)_{cm} - (\mathbf{u}_n)_{P_m}}{R_{nm}} + \mathbf{Q}_{nm}, \quad (27)$$

$$(\mathbf{t}_n)_{cs} = \frac{(\mathbf{u}_n)_{cs} - (\mathbf{u}_n)_{P_s}}{R_{ns}} + \mathbf{Q}_{ns}, \quad (28)$$

Borrowing terminology from circuit analysis, we have introduced the *normal resistances*:

$$R_{nm} = \frac{(\delta_n)_{cm}}{(2\mu + \lambda)_{cm}}, \quad (29)$$

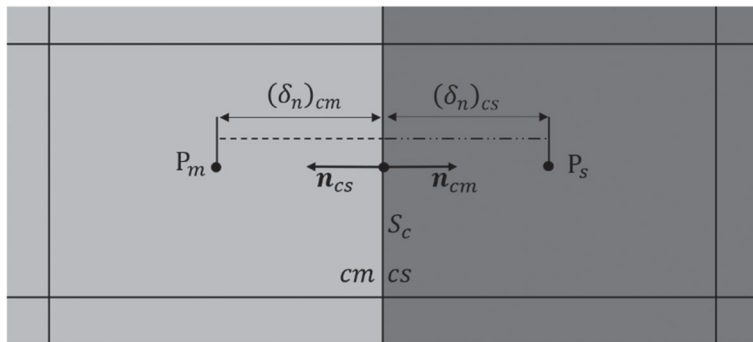


FIGURE 3 Example of contact between master cell V_{P_m} and slave cell V_{P_s} . The meshes are conformal and orthogonal

$$R_{ns} = \frac{(\delta_n)_{cs}}{(2\mu + \lambda)_{cs}}, \quad (30)$$

where the normal cell-center to face-center distances are defined as $(\delta_n)_{cm} = \mathbf{n}_{cm} \cdot \boldsymbol{\delta}_{cm}$ and $(\delta_n)_{cs} = \mathbf{n}_{cs} \cdot \boldsymbol{\delta}_{cs}$. For orthogonal boundaries they coincide with the norms $|\boldsymbol{\delta}_{cm}|$ and $|\boldsymbol{\delta}_{cs}|$.

In Equations (27) and (28), we have also introduced the substitutions:

$$\mathbf{Q}_{nm} = \lambda_{cm} \text{tr}(\nabla_t \mathbf{u}_t)_{cm} \mathbf{n}_{cm} = (\mathbf{n}_{cm} \mathbf{n}_{cm}) \cdot (\mathbf{n}_{cm} \cdot \boldsymbol{\sigma}_{cm}) - (2\mu + \lambda)_{cm} \mathbf{n}_{cm} \cdot (\nabla \mathbf{u}_n)_{cm}, \quad (31)$$

$$\mathbf{Q}_{ns} = \lambda_{cs} \text{tr}(\nabla_t \mathbf{u}_t)_{cs} \mathbf{n}_{cs} = (\mathbf{n}_{cs} \mathbf{n}_{cs}) \cdot (\mathbf{n}_{cs} \cdot \boldsymbol{\sigma}_{cs}) - (2\mu + \lambda)_{cs} \mathbf{n}_{cs} \cdot (\nabla \mathbf{u}_n)_{cs}. \quad (32)$$

These substitutions are not just for clarity. Indeed, by simply adjusting the definition of strain and stress tensors, the expressions for the traction in Equations (27) and (28) are not limited to linear-elastic materials but can be used also in the presence of additional strain components.

When master and slave faces are in contact and the gap width is zero, it follows from Equation (23) that:

$$(\mathbf{u}_n)_{cm} + [(\mathbf{f}_n)_{cm} - (\mathbf{f}_n)_{cs}] = (\mathbf{u}_n)_{cs}, \quad (33)$$

that is, the normal component of the displacement field is continuous at the contact interface only if corrected for initial face-center position. Making use of the continuity of the traction across the contact interface and considering that the outward normal lies in opposite direction, the traction in Equation (27) is equal to the opposite of the traction in Equation (28), that is, $(\boldsymbol{\tau}_n)_{cm} = -(\boldsymbol{\tau}_n)_{cs}$. Extracting and substituting $(\mathbf{u}_n)_{cs}$ from Equation (33), we derive the following expression for the normal component of the displacement at the boundary:

$$(\mathbf{u}_n)_{cm} = w_n (\mathbf{u}_n)_{P_m} + (1 - w_n) (\mathbf{u}_n)_{P_s} - \frac{R_{nm} R_{ns}}{R_{nm} + R_{ns}} (\mathbf{Q}_{ns} + \mathbf{Q}_{nm}) - (1 - w_n) [(\mathbf{f}_n)_{cm} - (\mathbf{f}_n)_{cs}], \quad (34)$$

where we have introduced the *normal interpolation weight*:

$$w_n = \frac{R_{ns}}{R_{nm} + R_{ns}}. \quad (35)$$

The displacement $(\mathbf{u}_n)_{cs}$ on the slave side can be obtained simply introducing Equation (34) in Equation (33). The boundary displacement in Equation (34) is a weighted linear interpolation between the cell-center values in the two opposing cells, plus an additional explicit term. Finally, substituting Equation (34) in Equation (27), we derive the following relation for the normal component of the contact traction on the master face:

$$(\mathbf{t}_n)_{cm} = \overline{(2\mu + \lambda)_c} \frac{(\mathbf{u}_n)_{P_s} - (\mathbf{u}_n)_{P_m}}{\delta_{ms}} - w_n \mathbf{Q}_{ns} + (1 - w_n) \mathbf{Q}_{nm} - \overline{(2\mu + \lambda)_c} \frac{(\mathbf{f}_n)_{cm} - (\mathbf{f}_n)_{cs}}{\delta_{ms}}, \quad (36)$$

where $\delta_{ms} = (\delta_n)_{cm} + (\delta_n)_{cs}$, and $\overline{(2\mu + \lambda)_c}$ is calculated with harmonic interpolation.

$$\overline{(2\mu + \lambda)_c} = \frac{(2\mu + \lambda)_{cm} (2\mu + \lambda)_{cs}}{\frac{(\delta_n)_{cs}}{(\delta_n)_{cm} + (\delta_n)_{cs}} (2\mu + \lambda)_{cm} + \frac{(\delta_n)_{cm}}{(\delta_n)_{cm} + (\delta_n)_{cs}} (2\mu + \lambda)_{cs}}. \quad (37)$$

The normal traction on the slave face is simply the opposite of Equation (36).

3.1.1 | Frictional component

Typically, depending on the surface characteristics of the two materials, only a fraction of the tangent or frictional force is transmitted across the contact surface, usually up to a certain fraction of the normal traction. Thus, we introduce the friction coefficient $\alpha \in [0, \infty)$ so that:

$$(\mathbf{t}_t)_{cm} = \min(|(\mathbf{t}_t)_{cm,\infty}|, \alpha |(\mathbf{t}_n)_{cm}|) \frac{(\mathbf{t}_t)_{cm,\infty}}{|(\mathbf{t}_t)_{cm,\infty}|}. \quad (38)$$

The traction $(\mathbf{t}_t)_{cm,\infty}$ is the tangent traction on the master face in case of infinite friction coefficient, that is, when the two boundaries in contact cannot slide relatively to each other. Applying the same procedure presented in the previous section, we derive the following relation for the tangent traction on the master face in case of infinite friction coefficient:

$$(\mathbf{t}_t)_{cm,\infty} = \overline{\mu}_c \frac{(\mathbf{u}_t)_{P_s} - (\mathbf{u}_t)_{P_m}}{\delta_{ms}} - w_t \mathbf{Q}_{ts} + (1 - w_t) \mathbf{Q}_{tm}, \quad (39)$$

where $\overline{(\mu)}_c$ is obtained with harmonic interpolation. We have also introduced the substitutions:

$$\mathbf{Q}_{tm} = \mu_{cm} \nabla_t u_{nm} = (\mathbf{I} - \mathbf{n}_{cm} \mathbf{n}_{cm}) \cdot (\mathbf{n}_{cm} \cdot \boldsymbol{\sigma}_{cm}) - \mu_{cm} \mathbf{n}_{cm} \cdot \nabla \mathbf{u}_{tm}, \quad (40)$$

$$\mathbf{Q}_{ts} = \mu_{cs} \nabla_t u_{ns} = (\mathbf{I} - \mathbf{n}_{cs} \mathbf{n}_{cs}) \cdot (\mathbf{n}_{cs} \cdot \boldsymbol{\sigma}_{cs}) - \mu_{cs} \mathbf{n}_{cs} \cdot \nabla \mathbf{u}_{ts}, \quad (41)$$

and we have defined the *tangent resistances*:

$$R_{tm} = \frac{(\delta_n)_{cm}}{\mu_{cm}}, \quad (42)$$

$$R_{ts} = \frac{(\delta_n)_{cs}}{\mu_{cs}}, \quad (43)$$

and the *tangent interpolation weight*:

$$w_t = \frac{R_{ts}}{R_{tm} + R_{ts}}. \quad (44)$$

The slave traction is simply the opposite of Equation (38). Given that the tangential traction is found explicitly as a function of the normal component, it is added directly to the source term as done in Equation (19) for the fixed-traction boundary type. The corresponding normal gradient is obtained once again from Equation (20) and it is used to calculate the tangential component of the boundary displacement $(\mathbf{u}_n)_{cm}$ and $(\mathbf{u}_n)_{cs}$. For this reason, the implicit contact methodology proposed in this article is more precisely *semi-implicit*.

3.2 | Correction for non-orthogonal boundaries and non-conformal meshes

In the previous sections, we have assumed that the cell-center to face-center vector δ_c is parallel to the outward normal vector and the meshes on both side of the contact interface are identical. Similar to Reference 36, we add in this section an explicit non-orthogonal correction term to accurately discretize the contact traction on non-orthogonal meshes.

As shown in Figure 4, we can define the correction vectors¹:

$$\mathbf{k}_m^{corr} = \delta_{cm} - (\delta_n)_{cm} \mathbf{n}_{cm}, \quad (45)$$

$$\mathbf{k}_s^{corr} = \delta_{cs} - (\delta_n)_{cs} \mathbf{n}_{cs} \quad (46)$$

By extracting the outward normal vector from the equations above and by substituting in Equation (25), we can rewrite the normal traction on the master and slave side (previously defined in Equations 27 and 28) as:

$$(\mathbf{t}_n)_{cm} = \underbrace{\frac{(\mathbf{u}_n)_{cm} - (\mathbf{u}_n)_{P_m}}{R_{nm}}}_{\text{orthogonal contribution}} - \underbrace{\frac{\mathbf{k}_m^{corr} \cdot (\nabla \mathbf{u}_n)_{cm}}{R_{nm}}}_{\text{non-orthogonal correction}} + \mathbf{Q}_{nm}, \quad (47)$$

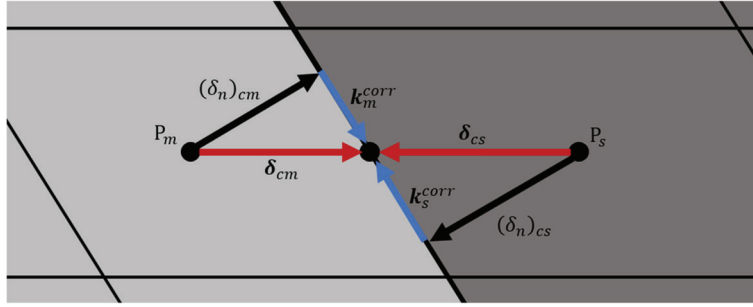


FIGURE 4 Non-orthogonal correction vectors for master cell V_{P_m} and slave cell V_{P_s} . The meshes are supposed to be conformal

$$(\mathbf{t}_n)_{cs} = \underbrace{\frac{(\mathbf{u}_n)_{c_s} - (\mathbf{u}_n)_{P_s}}{R_{ns}}}_{\text{orthogonal contribution}} - \underbrace{\frac{\mathbf{k}_s^{corr} \cdot (\nabla \mathbf{u}_n)_{cs}}{R_{ns}}}_{\text{non-orthogonal correction}} + Q_{ns} \quad (48)$$

Following the same derivation presented in Section 3.1, we obtain the following normal non-orthogonal correction traction. Once again, we show only the relation for the master face as the traction on the slave side is simply the opposite:

$$(\mathbf{t}_n)_{cm}^{corr} = \frac{1}{(2\mu + \lambda)_c} \frac{\mathbf{k}_s^{corr} \cdot (\nabla \mathbf{u}_n)_{cs} - \mathbf{k}_m^{corr} \cdot (\nabla \mathbf{u}_n)_{cm}}{\delta_{ms}}. \quad (49)$$

Applying the same procedure to the tangent component of the traction, we derive the tangent non-orthogonal correction traction for the infinite friction case:

$$(\mathbf{t}_t)_{cm,\infty}^{corr} = \frac{1}{\mu_c} \frac{\mathbf{k}_s^{corr} \cdot (\nabla \mathbf{u}_t)_{cs} - \mathbf{k}_m^{corr} \cdot (\nabla \mathbf{u}_t)_{cm}}{\delta_{ms}}. \quad (50)$$

A non-orthogonal correction term is always necessary in the presence of non-conformal meshes, even when the original boundaries are orthogonal. Indeed, the AMI can reconstruct the single opposing slave cell, but the reconstructed cell-center and its related fields do not necessarily lie in front of the master face-center. In the example on the left of Figure 5, the master cell V_{P_m} is in contact only with the slave cell V_{P_s} in the original geometry. The face-area-weighting technique detects a single interpolation weight equal to one even if the slave cell extends beyond the area in contact with V_{P_m} . When the traction is discretized on the master face, the AMI reconstructs the opposing slave cell as on the right of Figure 5 but the weighted average results in the values of the original slave cell V_{P_s} (this is valid for any quantity that needs to be reconstructed such as displacement or stress at the boundary or at the cell-center, face-center, cell-center location, etc.). Thus, to calculate the slave traction at the master face-center f_{cm} (see the example in Figure 5) it is necessary to add a non-orthogonal correction term as the one in Equations (47) and (48). For this purpose, it is sufficient to redefine the slave δ_{cs} vector (see the example in Figure 5) as the distance vector between the master face-center and the reconstructed slave cell-center.

As the final discretization of the contact traction is derived from the expression of the traction at the face center from the two sides of the contact interface, the approach described in this and in the previous sections corrects also for face skewness (even in the presence of non-conformal meshes), unlike the standard non-orthogonal correction used in cell-centered FV. In the future, the treatment here proposed for the contact boundaries could be extended to the internal faces. Besides reducing the skewness error, this could be relevant for cases (like those typical of nuclear fuel simulations) where a material presents strong gradients in the thermo-mechanical properties due to a temperature profile.

3.3 | Transition between open and closed gap

The traction \mathbf{t}_c must reduce to the pressure of the fluid in the gap region $-p_g \mathbf{n}_c$ (or to zero if no fluid is present) when the surfaces are not in contact, while the relations derived in the previous sections are valid only if the gap is closed. The sudden change in surface traction, which translates in the transition from an implicit discretization of the contact forces

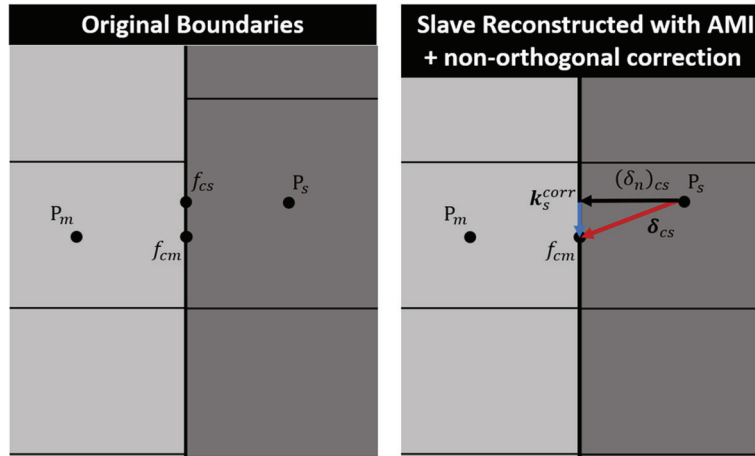


FIGURE 5 Non-orthogonal correction for non-conformal boundaries. The AMI method introduces non-orthogonality even in the presence of orthogonal original boundaries

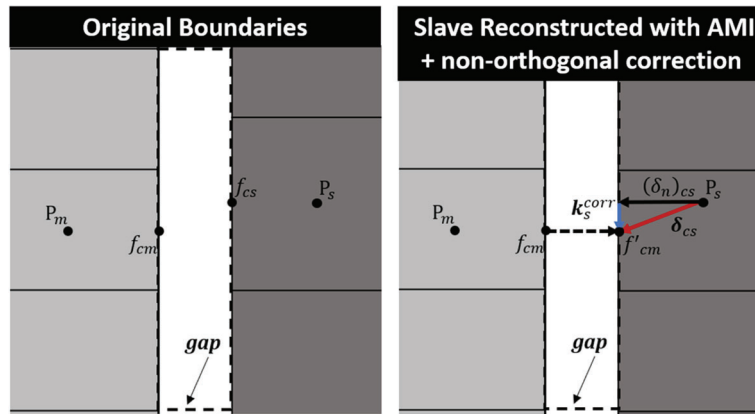


FIGURE 6 Non-orthogonal correction with open-gap. The master face-center must be projected on the slave face to calculate the slave traction

to the explicit fixed-traction scheme, can be treated numerically with the introduction of a blending function interpolating between the gap pressure (or traction-free condition) and the contact traction.

Different choices of blending function are possible and more details about the one proposed in this article are given in the next section. More generally, the blending coefficient $\beta(g)$ is a function of the gap width g , already defined in Equation (33), and must have the following characteristics:

- $\beta(g) \in [0, 1]$.
- $\lim_{g \rightarrow \infty} \beta(g) = 0$.

It is also desirable that $\beta(g)$ is sufficiently small for $g > 0$, at least starting from some fraction of the initial gap width. This helps to avoid or limit *spurious attractive forces* that would be introduced if the two surfaces were considered partly attached, although they are still separated by an open gap. The meaning of “sufficiently small” cannot be defined a-priori and it depends on various factors, such as case setup, boundary conditions and mesh resolution.

To properly account for the presence of an open gap, it is necessary to modify the definition of the slave cell-center to face-center distance vector. As shown in Figure 6, the master face-center is projected across the gap along the outward normal direction, to find the point f'_{cm} on the slave face. The slave vector δ_{cs} is then calculated as the distance between f'_{cm} and the reconstructed slave cell-center.

With this minor modification to the definition of δ_c and with the addition of the blending function, the contact methodology can be finally extended to open gap scenarios. After adding (implicitly) and subtracting (explicitly) the tangential component of the normal gradient, the total contact traction on the master face can be rewritten as:

$$\begin{aligned}
 \mathbf{t}_{cm} = & \overbrace{\beta(2\mu + \lambda)_c \frac{(\mathbf{u})_{P_s} - (\mathbf{u})_{P_m}}{\delta_{ms}}}^{\text{implicit}} - \overbrace{\beta(2\mu + \lambda)_c \frac{(\mathbf{u}_t)_{P_s} - (\mathbf{u}_t)_P}{\delta_{ms}}}^{\text{explicit}} \\
 & + \beta \left[\overbrace{-w_n \mathbf{Q}_{ns} + (1 - w_n) \mathbf{Q}_{nm} - \frac{(\mathbf{f}_n)_{cm} - (\mathbf{f}_n)_{cs}}{\delta_{ms}} + (\boldsymbol{\tau}_t)_{cm}}^{\text{explicit}} \right] + \beta \left[\overbrace{\frac{(\mathbf{k}_s^{corr} \cdot (\nabla \mathbf{u}_n)_{cs} - \mathbf{k}_m^{corr} \cdot (\nabla \mathbf{u}_n)_{cm})}{(2\mu + \lambda)_c \delta_{ms}}}}^{\text{explicit}} \right] \\
 & + \overbrace{(1 - \beta)(-p_g \mathbf{n}_{cm})}^{\text{explicit}}.
 \end{aligned} \tag{51}$$

It is important to note that, as for the penalty factor in the penalty method, the introduction of the blending coefficient in the implicit contact methodology can be interpreted as a softening of the contact interface. This inevitably changes the physics of the simulation with larger penetration and wider contact area to preserve the global force balance, but it is necessary for convergence in many situations. Different from the penalty method where the contact pressure is applied only when there is penetration, the blending can be made continuous across the gap, that is, one can select an appropriate blending function to avoid a discontinuous jump from the gap pressure (usually very weak) to the contact stresses (usually very high). Additionally, because the contact stresses are taken care of in a semi-implicit manner, the performances (stability and convergence properties) of the implicit contact methodology are expected to be superior to those of the explicit FV penalty method implementation derived from Cardiff.

3.4 | The sigmoid blending function

A curve that respects the characteristics described in the previous section and at the same time allows for a smooth and gradual change in blending coefficient is the *sigmoid function*. The specific function proposed in this article is given as:

$$\beta(g) = \frac{1}{1 + \exp\left(\ln(10^6 - 1) \frac{(g+o)}{h}\right)}. \tag{52}$$

The sigmoid above is completely defined by the two parameters shown in Figure 7: the *offset* o , that is the position of the inflection point along the g axis, and its width. In our case, we decided to define the latter as the half-width h of a symmetric interval around inflection point so that $\beta(h - o) = 10^{-6}$, although any value $\in (0, 0.5)$ could have been chosen. The user can provide the width and the offset as an *absolute* value (in meters) or in *relative* terms as a fraction of the cell-center to face-center distance (designated in this case as o_{rel} and h_{rel}).

Figure 8 shows the effect of changing the offset and the width of the blending curve. For the same width, a larger offset causes a higher degree of penetration but reduces the spurious attractive forces when the gap is still open. In certain cases, a larger offset might also obstruct convergence, given that the smaller is the blending coefficient, the smaller is the implicit contribution of the traction in Equation (51). Alternatively, for the same offset, a larger half-width allows a smoother change in blending coefficient but increases the risk of incurring in spurious attractive forces when the two bodies are not in contact.

To avoid any spurious positive traction the sigmoid can be *truncated*, that is, once the user activates the corresponding option the blending function is forced to be equal to 0 for $g \geq 0$. The main drawback is the discontinuity at the interface which might worsen the convergence of some simulations. This issue might be partially solved with a more significant relaxation of the displacement field or with smaller time-steps if the problem is time-dependent.

Independently from the truncated option, we found it is useful to cut the sigmoid function using a threshold of 10^{-6} : below this value the blending coefficient is directly set to 0. This avoids oscillations and limits the spurious attractive forces.

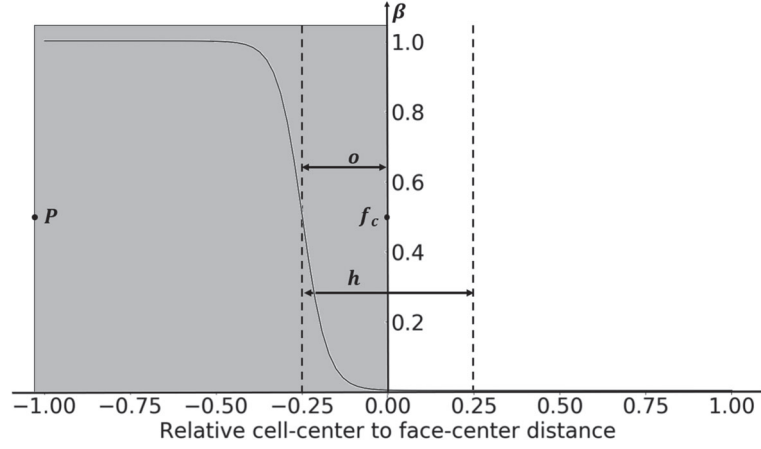


FIGURE 7 Sigmoid blending function. The offset o and the half-width h are shown

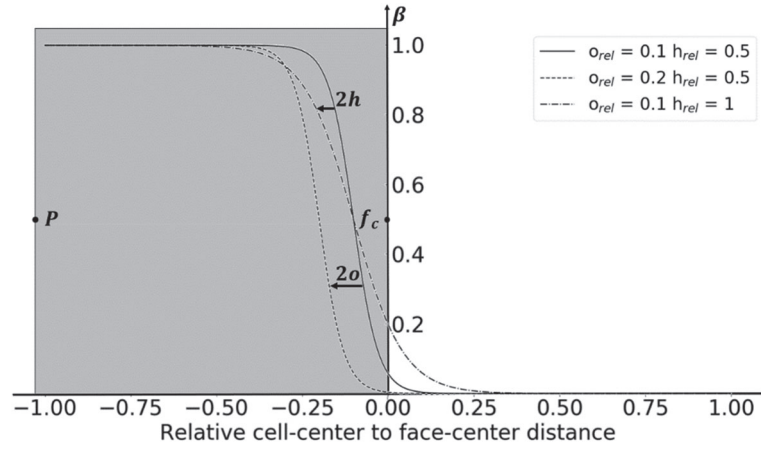


FIGURE 8 Effect of changing offset and half-width on the blending function curve

3.5 | Implicit contact for boundaries with different geometry and orientation

All the relations derived in the previous sections assume that the two outward normal vectors \mathbf{n}_{cm} and \mathbf{n}_{cs} lie in the same direction. However, with few simple modifications, we can extend the implicit contact methodology to the more general case of boundaries with different orientation.

First, we introduce the *common normal* vectors \mathbf{n}_{cm}^* and \mathbf{n}_{cs}^* given by²:

$$\mathbf{n}_{cm}^* = \frac{\mathbf{n}_{cm} - \mathbf{n}_{cs}}{|\mathbf{n}_{cm} - \mathbf{n}_{cs}|} = -\mathbf{n}_{cs}^*. \quad (53)$$

Then, as done in Section 3.3, we project the master face-center across the gap on the slave face, only this time the point is projected along the direction determined by \mathbf{n}_{cm}^* . The corresponding point is used to calculate the slave vector δ_{cs} as explained in Section 3.2 and as shown in Figure 9.

Following the same derivation presented in Section 3.1, we decompose the contact traction into a normal component along \mathbf{n}_{cm}^* (or \mathbf{n}_{cs}^*) and tangent component perpendicular to \mathbf{n}_{cm}^* (or \mathbf{n}_{cs}^*). It follows that the explicit terms in the definition of the traction must be changed into:

$$\mathbf{Q}_{nm^*} = \lambda_{cm} tr(\nabla_t^* \mathbf{u}_{t^*})_{cm} \mathbf{n}_{cm}^* = (\mathbf{n}_{cm}^* \mathbf{n}_{cm}^*) \cdot (\mathbf{n}_{cm} \cdot \boldsymbol{\sigma}_{cm}) - (2\mu + \lambda)_{cm} \mathbf{n}_{cm}^* \cdot (\nabla \mathbf{u}_n)_{cm}, \quad (54)$$

$$\mathbf{Q}_{ns^*} = \lambda_{cs} tr(\nabla_t^* \mathbf{u}_{t^*})_{cs} \mathbf{n}_{cs}^* = (\mathbf{n}_{cs}^* \mathbf{n}_{cs}^*) \cdot (\mathbf{n}_{cs} \cdot \boldsymbol{\sigma}_{cs}) - (2\mu + \lambda)_{cs} \mathbf{n}_{cs}^* \cdot (\nabla \mathbf{u}_n)_{cs}. \quad (55)$$

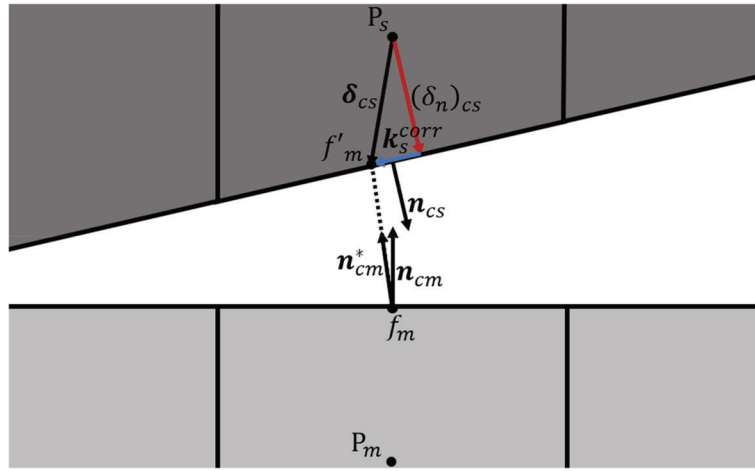


FIGURE 9 Boundaries with different orientation. The master face-center is projected along the common normal \mathbf{n}_c^*

And:

$$\mathbf{Q}_{tm^*} = \mu_{cm} \nabla_{t^*} \mathbf{u}_{nm^*} = (\mathbf{I} - \mathbf{n}_{cm}^* \mathbf{n}_{cm}^*) \cdot (\mathbf{n}_{cm} \cdot \boldsymbol{\sigma}_{cm}) - \mu_{cm} \mathbf{n}_{cm}^* \cdot \nabla \mathbf{u}_{tm^*}, \quad (56)$$

$$\mathbf{Q}_{ts^*} = \mu_{cs} \nabla_{t^*} \mathbf{u}_{ns^*} = (\mathbf{I} - \mathbf{n}_{cs}^* \mathbf{n}_{cs}^*) \cdot (\mathbf{n}_{cs} \cdot \boldsymbol{\sigma}_{cs}) - \mu_{cs} \mathbf{n}_{cs}^* \cdot \nabla \mathbf{u}_{ts^*}. \quad (57)$$

With these simple modifications, we derive the final relation for the total contact traction on the master side is:

$$\begin{aligned} \mathbf{t}_{cm} = & \overbrace{\frac{(\mathbf{u})_{P_s} - (\mathbf{u})_{P_m}}{\delta_{ms}}}^{\text{implicit}} - \overbrace{\frac{(\mathbf{u}_{t^*})_{P_s} - (\mathbf{u}_{t^*})_{P_m}}{\delta_{ms}}}^{\text{explicit}} \\ & + \beta \left[\overbrace{-w_n \mathbf{Q}_{ns^*} + (1 - w_n) \mathbf{Q}_{nm^*}}^{\text{explicit}} - \frac{(\mathbf{f}_{n^*})_{cm} - (\mathbf{f}_{n^*})_{cs}}{\delta_{ms}} + (\boldsymbol{\tau}_{t^*})_{cm} \right] \\ & + \beta \left[\overbrace{\frac{\mathbf{k}_s^{\text{corr}} \cdot (\nabla \mathbf{u}_{n^*})_{cs} - \mathbf{k}_m^{\text{corr}} \cdot (\nabla \mathbf{u}_{n^*})_{cm}}{\delta_{ms}}}^{\text{explicit}} \right] + \overbrace{(1 - \beta) (-p_g \mathbf{n}_{cm})}^{\text{explicit}}. \end{aligned} \quad (58)$$

The discretization of the contact traction on the master face leads to the following diagonal and off-diagonal coefficients, and source term:

$$a_{P_m} = -\beta \frac{\overline{(2\mu + \lambda)_c}}{\delta_{ms}} S_c, \quad (59)$$

$$a_{P_s} = a_{N_c} = \beta \frac{\overline{(2\mu + \lambda)_c}}{\delta_{ms}} S_c, \quad (60)$$

$$\begin{aligned} \mathbf{b}_{P_m} = & S_c \beta \left[-\frac{(\mathbf{u}_{t^*})_{P_s} - (\mathbf{u}_{t^*})_{P_m}}{\delta_{ms}} - w_n \mathbf{Q}_{ns^*} + (1 - w_n) \mathbf{Q}_{nm^*} \right] + S_c \beta \left[-\frac{(\mathbf{f}_{n^*})_{cm} - (\mathbf{f}_{n^*})_{cs}}{\delta_{ms}} + (\boldsymbol{\tau}_{t^*})_{cm} \right] \\ & + S_c \beta \left[\frac{\mathbf{k}_s^{\text{corr}} \cdot (\nabla \mathbf{u}_{n^*})_{cs} - \mathbf{k}_m^{\text{corr}} \cdot (\nabla \mathbf{u}_{n^*})_{cm}}{\delta_{ms}} \right] + S_c (1 - \beta) (-p_g \mathbf{n}_{cm}). \end{aligned} \quad (61)$$

The discretization of the traction on the slave faces leads to the following coefficients and source term:

$$a_{P_s} = -\beta \frac{\overline{(2\mu + \lambda)_c}}{\delta_{ms}} S_c, \quad (62)$$

$$a_{P_m} = a_{N_c} = \beta \frac{(2\mu + \lambda)_c}{\delta_{ms}} S_c, \quad (63)$$

$$\begin{aligned} \mathbf{b}_{P_s} = & \left[-(2\mu + \lambda)_c \frac{(\mathbf{u}_{t^*})_{P_m} - (\mathbf{u}_{t^*})_{P_s}}{\delta_{ms}} + w_n \mathbf{Q}_{ns^*} - (1 - w_n) \mathbf{Q}_{nm^*} \right] + S_c \beta \left[+(2\mu + \lambda)_c \frac{(\mathbf{f}_{n^*})_{cm} - (\mathbf{f}_{n^*})_{cs}}{\delta_{ms}} + (\boldsymbol{\tau}_{t^*})_{cs} \right] \\ & + S_c \beta \left[(2\mu + \lambda)_c \frac{\mathbf{k}_m^{corr} \cdot (\nabla \mathbf{u}_{n^*})_{cm} - \mathbf{k}_s^{corr} \cdot (\nabla \mathbf{u}_{n^*})_{cs}}{\delta_{ms}} \right] + S_c (1 - \beta) (-p_g \mathbf{n}_{cs}). \end{aligned} \quad (64)$$

4 | BENCHMARK CASES

The implicit contact methodology has been implemented as a custom boundary condition in OFFBEAT,^{41–45} a stress solver developed for nuclear engineering applications and based on the OpenFOAM C++ library.^{31,46} In this section, we verify the methodology against five test cases with known solution. The first two cases show that the contact approach introduced in this work reduces to the treatment of multi-material interfaces developed by Tuković when considering two materials in contact with infinite friction coefficient and closed gap. The third case is a classic contact patch test that studies whether the new contact methodology correctly transfer a constant stress distribution across interfaces with dissimilar meshes. The fourth and fifth cases are traditional punch tests used already for the verification of other contact procedures both with the FVM and the FEM. The main aim is to assess if the new methodology can correctly capture the contact stresses for static force-loading scenarios in the presence of curved contact surfaces and partially open gap.

4.1 | Cases 1 and 2: Bi-material cylinder with internal pressure or outer tangential traction

For two bodies in contact with infinite friction coefficient and without the formation of an open gap, the implicit contact procedure should be equivalent to the treatment for multi-material interfaces developed by Tuković.

To validate the use of the implicit contact methodology for similar scenarios, we examine two cases both consisting of a bi-material cylinder in plane stress with the dimensions shown in Figure 10. In Case 1, the inner cylinder is pressurized with a constant pressure p_i , while the outer surface is modeled as stress-free; in Case 2, the inner surface is fixed and a constant tangential load τ is applied to the outer cylinder.

Both test cases were analyzed in the original publication from Tuković. The main difference is that in this article the bi-material cylinder is reproduced with two separate concentric cylinders. The two materials, indicated with different color in Figure 10, have a Poisson's ratio of $\nu_1 = 0.35$ and $\nu_2 = 0.3$, while the Young's modulus ratio E_2/E_1 ranges from 0.2 to 10. The computational model has a total of 50 cells in the radial direction. The cells in the azimuthal direction are 480 for the inner cylinder and 920 for the outer cylinder. This difference in azimuthal discretization is introduced to test the implicit contact methodology on non-conformal meshes.

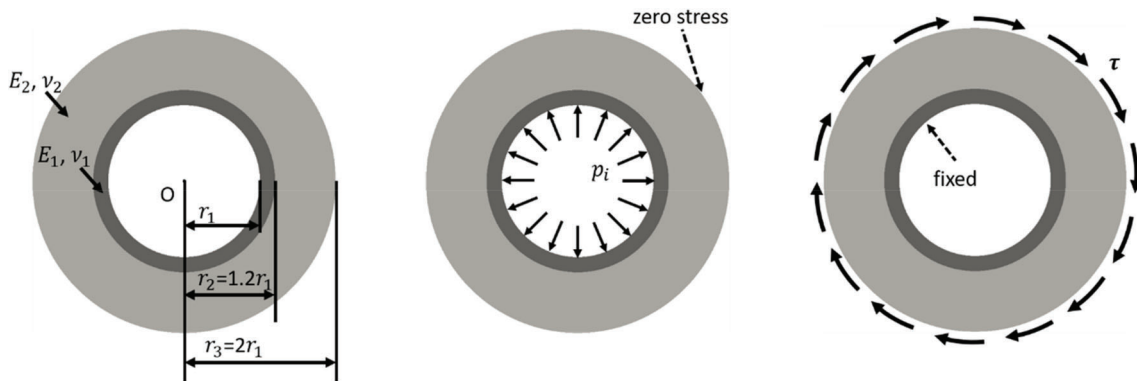


FIGURE 10 Bi-material cylinder, dimensions (left), load for case with uniform internal pressure (center), and load for case with outer tangential traction (right)

As there is no gap opening, it is not necessary to truncate the sigmoid blending function and the relative offset o_{rel} can be set to zero. Also, the relative half-width h_{rel} can be set to the small value of 0.1 without incurring in the issue of spurious attractive forces and without affecting the convergence properties of the simulations. With this choice of blending parameters, the final interpenetration (relative to the radial size of the cells on the contact interface) is about 10^{-4} for Case 1 and it is certainly negligible.

All simulations are in steady state and are performed on the same Intel Xeon CPU E5-1660 v4 using one of the eight 3.2 GHz cores. Relaxation factors are not necessary to reach convergence, and the simulation time varies from ~ 10 to ~ 35 s depending on the boundary conditions (Case 1 or Case 2) and on the E_2/E_1 ratio.

Case 1 was already analyzed by Phan and Mukherjee,⁴⁷ where the authors have derived the analytical solutions for the radial and hoop stresses:

$$\sigma_r(r) = \begin{cases} \frac{r_1^2 p_i - r_2^2 p_{12} + (p_{12} - p_i) \left(\frac{r_1 r_2}{r} \right)^2}{r_2^2 - r_1^2}, & r_1 \leq r \leq r_2, \\ \frac{r_2^2 p_{12} - p_{12} \left(\frac{r_2 r_3}{r} \right)^2}{r_3^2 - r_2^2}, & r_2 < r \leq r_3, \end{cases} \quad (65)$$

$$\sigma_\theta(r) = \begin{cases} \frac{r_1^2 p_i - r_2^2 p_{12} - (p_{12} - p_i) \left(\frac{r_1 r_2}{r} \right)^2}{r_2^2 - r_1^2}, & r_1 \leq r \leq r_2, \\ \frac{r_2^2 p_{12} + p_{12} \left(\frac{r_2 r_3}{r} \right)^2}{r_3^2 - r_2^2}, & r_2 < r \leq r_3, \end{cases} \quad (66)$$

where p_{12} is the contact pressure at the interface:

$$p_{12}(r) = \frac{\frac{2r_1^2 p_i}{E_1(r_2^2 - r_1^2)}}{\frac{1}{E_2} \left(\frac{r_3^2 + r_2^2}{r_3^2 - r_2^2} + \nu_2 \right) + \frac{1}{E_1} \left(\frac{r_2^2 + r_1^2}{r_2^2 - r_1^2} - \nu_1 \right)}. \quad (67)$$

Similarly, Case 2 was already analyzed by Nie and Batra,⁴⁸ where they derived the analytic solution for the shear stress:

$$\sigma_{r\theta}(r) = \tau \frac{r_3^2}{r^2}. \quad (68)$$

The solution for the tangential displacement, instead, was derived by Tuković:

$$\sigma_\theta(r) = \begin{cases} \frac{r_3^2 \tau}{2\mu_1} \left(\frac{r}{r_1} - \frac{1}{r} \right), & r_1 \leq r \leq r_2, \\ \frac{r_3^2 \tau}{2} \left[\left(\frac{1}{\mu_1 r_1^2} - \frac{1}{\mu_1 r_2^2} + \frac{1}{\mu_2 r_2^2} \right) r - \frac{1}{\mu_2 r} \right], & r_2 < r \leq r_3. \end{cases} \quad (69)$$

The radial and hoop stress for Case 1 are shown in Figures 11 and 12, while the shear stress and tangential displacement for Case 2 are shown in Figures 13 and 14. Overall, the profiles compare well with the analytical solutions, but the errors are in some cases slightly larger than those reported by Tuković et al. in their original publication where they mention errors below 1%. In our simulations we obtained errors of: $\sim 0.5\%$ for the radial stress and just above 1% for the hoop stress of Case 1; $\sim 0.1\%$ for the shear stress and $\sim 3.5\%$ for the tangential displacement of Case 2.

Two main sources of error are identified. One derives from the non-conformal discretization of the two contact interfaces as the AMI algorithm detects a small gap between the two surfaces, despite them being only different discretization of the same circle. This error seems difficult to avoid in the current implementation of the implicit contact methodology, but it is expected to decrease and approach zero with progressive mesh refinement. The second source of error is simply the result of the introduction of a blending function, given that any value of the blending coefficient lower than one causes some readjustment of the displacement to satisfy the global force balance and might affect the solution. Tests were performed to verify that by doubling the azimuthal mesh resolution for both cylinders or by fixing a blending coefficient higher than 0.99 (as it can be done for example by using a negative offset or by defining a different blending function) one can bring the average errors below 0.1%.

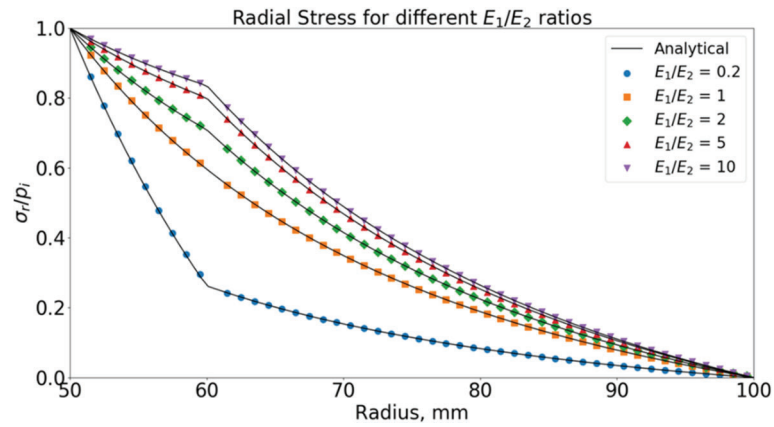


FIGURE 11 Radial stress in bi-material cylinder with inner pressure p_i for various E_1/E_2 ratios

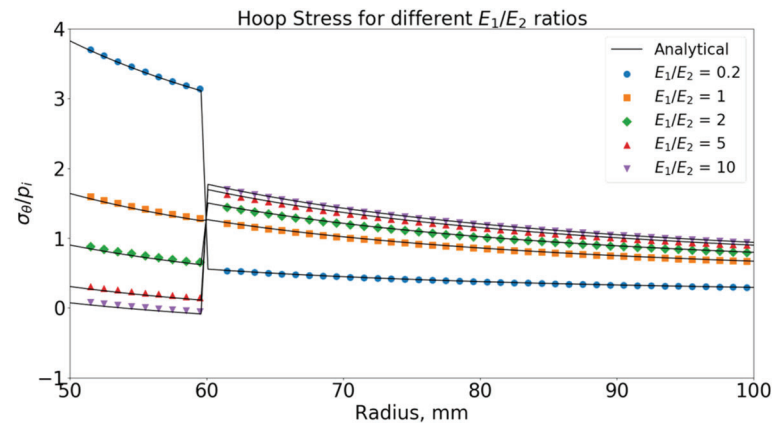


FIGURE 12 Hoop stress in bi-material cylinder with inner pressure p_i for various E_1/E_2 ratios

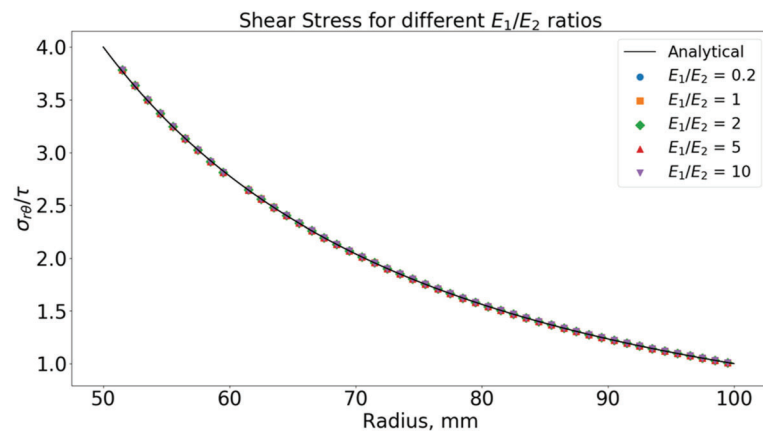


FIGURE 13 Shear stress in bi-material cylinder with tangential load τ for various E_1/E_2 ratios

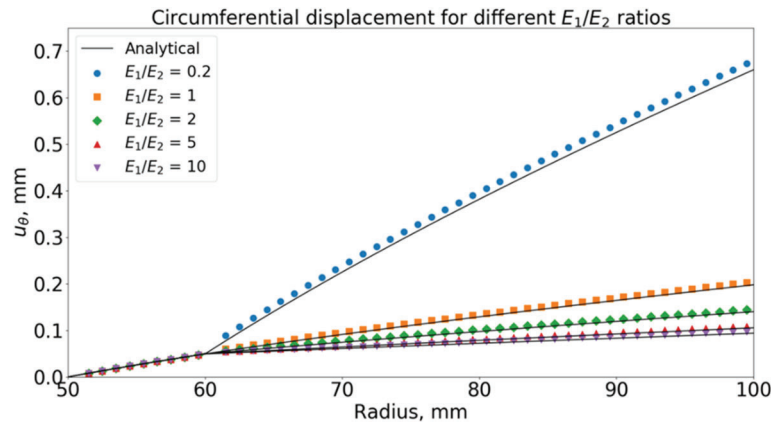


FIGURE 14 Circumferential displacement in bi-material cylinder with tangential load τ for various E_1/E_2 ratios

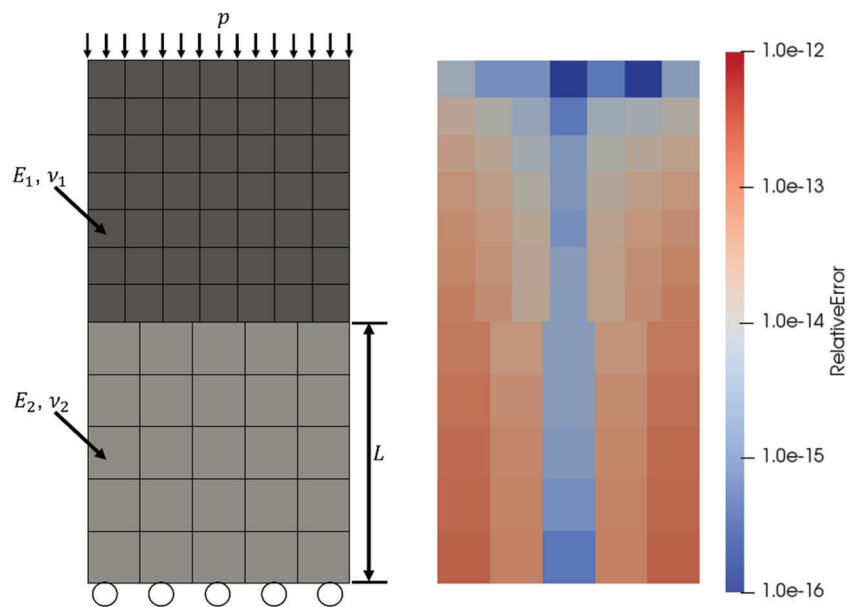


FIGURE 15 Contact patch test. The model with the mesh and boundary conditions is shown on the left. A color map of the relative error for the vertical or normal component of the stress field (with respect to a uniform distribution equal to the top pressure p) is shown on the right

4.2 | Case 3: Contact patch test

The third verification case is a 2D contact patch test along the classical definition of Taylor and Papadopoulos.⁴⁹ Similar tests have been extensively used in the FE community for the verification of contact methodologies as done for example in References 50–55 for the mortar method. The case consists of a 2D square block of side L pressed by a uniform pressure p against a second block of identical dimensions. To generalize the test for contact cases involving different material properties, the two materials, indicated with different color on the left of Figure 15, have a Poisson's ratio of $\nu_1 = 0.35$ and $\nu_2 = 0.3$, and a Young's modulus ratio E_2/E_1 equal to 0.5. A different discretization is used for the two blocks: a 5×5 grid is selected for the bottom block, a 7×7 grid for the top one.

With the regards to the parameters of the implicit contact simulation, the relative offset and half-width are set to 0 and 0.1, respectively, while the sigmoid function is not truncated since the gap is closed (i.e., there is no need to limit spurious attractive forces). For the purpose of this test, the threshold for the residuals of the displacement field is set to 10^{-14} , which represents the minimum residual that we could achieve in double precision in OpenFOAM.

The right side of Figure 15 shows a color map of the relative error for the normal stress component with respect to an expected uniform value equal to p obtained using the implicit contact algorithm. The magnitude of the reported errors is consistent with the chosen residuals, thus confirming a correct transmission of stresses through the contact interface.

4.3 | Case 4: Two-dimensional punch test

The fourth verification case is a 2D linear-elastic punch test from the NAFEMS benchmark series,⁵⁶ which we have already analyzed in a previous publication⁴² using the penalty method. The case, approximated as in plane strain, consists of a cylindrical steel punch pressed with a constant point load of 35 kN on an aluminum foundation. This setup is shown with its main dimensions in Figure 16 where the two bodies are displayed with different colors. The Young's moduli are $E_{\text{foundation}} = 70$ GPa and $E_{\text{punch}} = 210$ GPa, while both materials have the same Poisson's ratio of $\nu = 0.3$. The contact is assumed to be frictionless so that the analytical solution found in literature can be used as comparison.

The problem is symmetric and only half of the geometry is reproduced in the computational model, using symmetry boundary conditions. The case is solved with a set of four different mesh refinement levels, from Level 1 to Level 4, with a total of 2660, 8231, 27,491, and 99,469 cells, respectively. All computational models are generated using the Salome platform⁵⁷ following the same guiding principle: the 2D NETGEN algorithm with quadrilateral elements and coarse grid settings is used for most of the model, while a regular, quadrangular and more refined grid is used both close to the contact area and close to the point load on the top of the punch. Moving from a refinement level to the next, the cell width in the quadrangular zones is halved. As an example, the coarsest mesh is shown with a detail of the contact area on the center-left side of Figure 16.

The relative offset and half-width are set to 0 and 1, respectively, leading to an average relative interpenetration (with respect to the cell size) of $\sim 1 \cdot 10^{-2}$ for the finest mesh. The sigmoid function is truncated to limit spurious attractive forces. The same simulations are performed employing a boundary condition based on the penalty method (derived from the implementation of Cardiff), so that the performance and accuracy of the two contact methodologies can be compared. The penalty scale factor is set to 1.

All simulations are performed in steady state on the same Intel Xeon CPU E5-1660 v4, using one of the eight 3.2 GHz cores. For static contact cases with force loading like the one analyzed in this section, the penalty method results in an ill-conditioned problem and an unstable solution. To reach convergence, we use a relaxation factor of 0.1 for the interface pressure together with an equation relaxation factor (as opposed to field relaxation) of 0.999. Due to the partial contact between punch and foundation surfaces and due to the implicit treatment of the boundary stresses, the implicit contact simulations do not require relaxation.

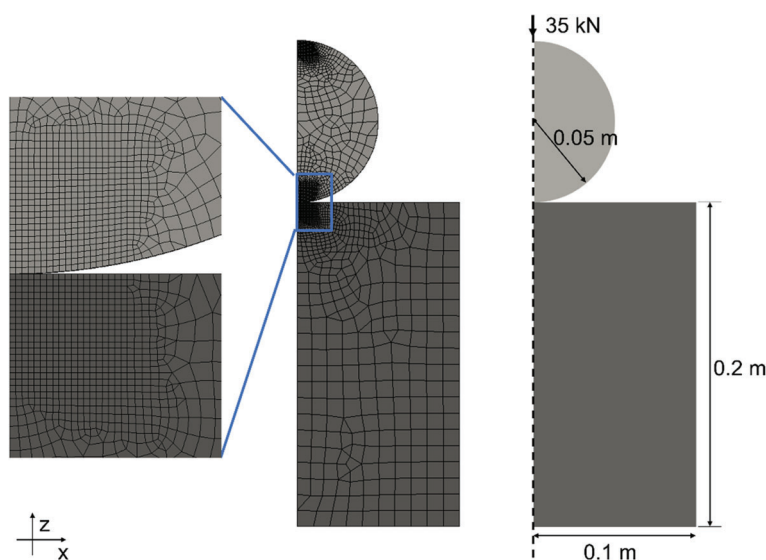


FIGURE 16 Two-dimensional frictionless punch test: Model (right) and mesh with detail of the contact region (center and left). A similar quadrangular grid is used for the area close to the point load

As done in Reference 56 (although another classic reference for this case is Reference 58), an analytical solution for the contact stresses along the rounded edge of the punch can be obtained using Hertzian contact theory which assumes the following parabolic pressure distribution (which is known to be a very good approximation):

$$p = p_{\max} \sqrt{1 - \xi^2}, \quad (70)$$

where $p_{\max} = 3.5854$ GPa is the maximum pressure, $\xi = x/l$ is the normalized coordinate along the rounded edge and $l = 6.210$ mm is the contact arc length along the punch outer surface.

In Figure 17, the calculated contact pressure is compared against the analytical solution for both the implicit and penalty methodologies. For the sake of a clearer graph, we show only the profiles obtained with the finest mesh, Level 4. Both simulations compare well against the results predicted by the Hertzian contact theory: the implicit contact calculates a maximum pressure of 3.5774 GPa and a contact arc length of 6.315 mm, with a relative difference of -0.2% and 1.7% with respect to the analytical values; while the penalty method predicts a maximum pressure of 3.5673 GPa and a contact arc length of 6.379 mm, with a relative difference of -0.5% and 2.7% .

To examine the discretization errors, the converged value of p_{\max} is approximated using Richardson's extrapolation method.⁵⁹ Figure 18 shows that the discretization errors of the implicit method for the contact pressure are below 1% even on the coarsest mesh, probably due the small penetration achieved thanks to parameters selected for the blending function. For comparison, the discretization errors for the penalty method simulations are also included in the graph: despite a similar convergence order, the errors are slightly higher. However, more accurate results could be obtained either by using more refined grids or by increasing the penalty scale factor beyond 1, probably at the cost of convergence.

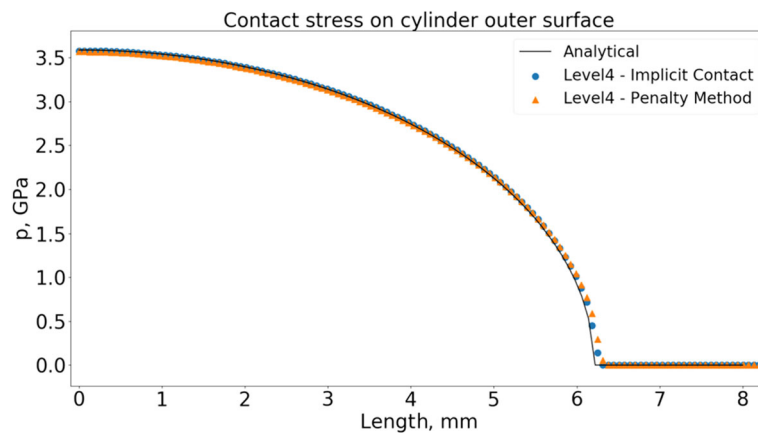


FIGURE 17 Two-dimensional punch test: Contact stresses on the cylinder outer surface

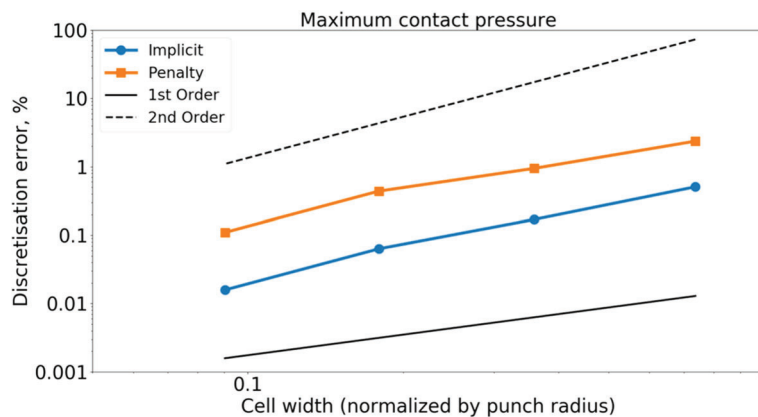


FIGURE 18 Discretization error in the maximum contact pressure on the rounded edge of the punch

The gain in performance achieved with the implicit contact methodology is difficult to assess. On the one hand, the simulations performed with the penalty-based boundary condition are significantly slower. The speedup offered by the implicit methodology ranges from a factor of $\sim 1000\times$ on the Level 1 grid to a factor of $\sim 400\times$ on the Level 4 grid (the performance boost due to the implicit treatment of the boundary stresses decreases as the ratio between the number of boundary faces and the total number of cells decreases). The superior convergence properties are apparent also from Figure 19, where the residuals of the x - and y -components of the displacement field are plotted against the number of iterations of the linear solver (the residuals are limited to the first 3000 iterations for the simulation with the penalty-based boundary condition). On the other hand, the penalty method simulation is severely affected by the almost point-like nature of the contact case considered, and by the use of the equation relaxation technique. This is known to significantly affect convergence and might not be necessary for dynamic cases with displacement loading. Separate tests were performed to verify that, when using the same computational models to simulate a short transient (e.g., $10\ \mu s$ with time steps of $1\ \mu s$) or a fixed downward displacement on the punch (as opposed to the point-load), the running times of the two contact methodologies become comparable. Nevertheless, the performance boost remains relevant for improving the analysis of static, force-loading cases with the FVM, which is the main objective of this article.

4.3.1 | The impact of the sigmoid blending function parameters

Case 4 is ideal to show the importance of selecting appropriate parameters for the sigmoid blending function. As an example, we use the most refined model, Level 4. Starting from the original simulation with $o_{rel} = 0$, $h_{rel} = 1$ and the truncated option activated, we perform 3 separate simulations where we progressively: deactivate the truncated option; change h_{rel} to 0.5; change o_{rel} to 0.5. We focus on the following three quantities: the maximum contact pressure on the punch; the contact arc length on the curved surface of the punch and the maximum positive stress on the foundation top surface relative to the maximum contact pressure. The results of the three simulations are compared against the original simulation in Table 1.

First, deactivating the truncated option causes a sudden increase in spurious attractive forces. Due to these high positive stresses (~ 0.8 GPa), the contact pressure distribution is slightly distorted to respect the global force balance: the maximum (negative) pressure and the contact arc length increase. Nevertheless, the spurious positive forces are limited to just a few cells near the edge of the contact area, and their effect on the global results of the simulation is surprisingly small (0.1% error for the maximum pressure, 2% for the contact arc length).

Decreasing the relative half-width from 1 to 0.5 makes the blending function steeper (see Section 3.4). Outside the interpenetration zone the value of the blending coefficient is lower, and the spurious attractive forces decrease (~ 0.4 GPa). The maximum pressure and the contact arc length approach more closely the values predicted by the original simulation.

Finally, increasing the relative offset from 0 to 0.5, causes the blending function to shift inward toward the cell-center (see again Section 3.4). For the same interpenetration, the value of the blending coefficient is lower, and it is probably below the 10^{-6} threshold immediately outside the interpenetration zone. The interpenetration increases, the contact area

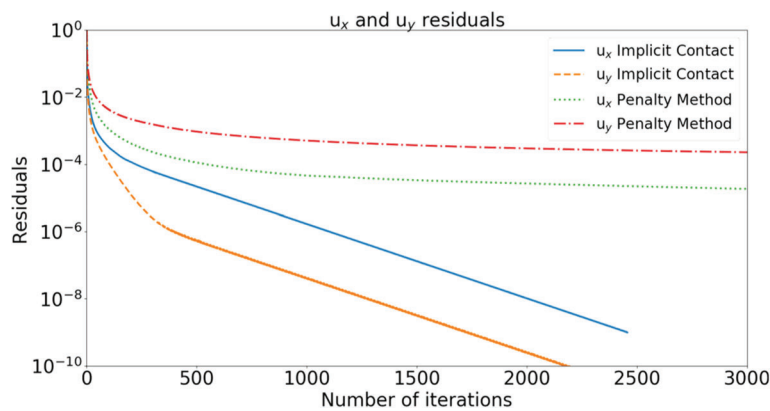
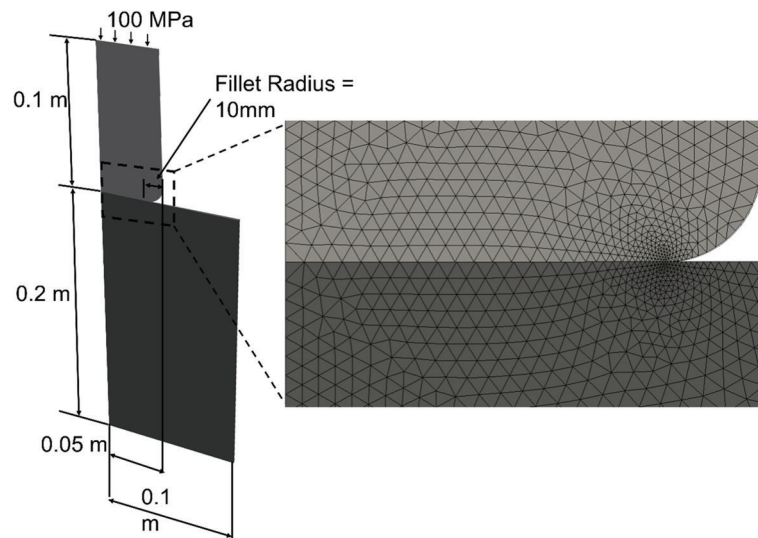


FIGURE 19 Residuals against number of iterations for Case 4 on the finest mesh. Residuals are shown both for the implicit contact and penalty-method based boundary conditions

TABLE 1 Frictionless 2D punch test: Effect of changes in the blending function parameters on the results for mesh refinement Level 4

Truncated option	h_{rel}	o_{rel}	Max contact pressure (GPa)	Contact arc length (mm)	Rel. max positive stress
On	1	0	3.5774	6.315	~0%
Off	1	0	3.5813	6.444	25%
Off	0.5	0	3.5790	6.379	11%
Off	0.5	0.5	3.5763	6.509	~0%

Note: The results considered are the maximum contact pressure on the punch, the contact arc length along the curved surface of the punch, and the maximum positive stress on the foundation top surface relative to the maximum pressure. The first line corresponds to the original simulation performed for the analysis of the benchmark.

**FIGURE 20** Punch test with rounded edge, 2D wedge model: main dimensions (left) and detail of the finest mesh (Level 3) near the contact area (right)

becomes larger, and the maximum contact pressure decreases. Despite negligible spurious positive forces, the error on the predicted contact arc length is larger than for the previous simulations (due to the increased interpenetration). In general, however, it is preferable to reduce as much as possible (or eliminate) the contribution of the positive spurious forces. A more refined grid could be used to improve the accuracy of the contact arc length.

It must be said that it is difficult to identify general patterns and that the choice of the optimal blending function parameters seems to be strongly case dependent. This is certainly one of the main drawbacks of the implicit methodology which might be alleviated in the future with the development of an automated algorithm. Nevertheless, several tests indicate that a good balance between accuracy, stability and convergence speed is reached with values from 0 to 1 for the relative offset and from 0.1 to 1 for the relative half-width.

4.4 | Case 5: Punch test with rounded edge

The last verification case is another linear-elastic punch test from the NAFEMS benchmark series. This time, the punch is a steel cylinder characterized by a rounded edge on its base and pressed with a constant pressure of 100 MPa against a cylindrical aluminum foundation. The setup and its main dimensions are shown in Figure 20, while the material properties for aluminum and steel are the same used in the previous benchmark (Case 4). This punch test has been studied in the past by several authors,^{60,61} notably by Cardiff et al.²¹ using a FV contact stress solver based on the penalty method.

As the case is axisymmetric, we consider first a 2D *wedge* geometry. The model is discretized with a triangular mesh using 3 refinement levels, from Level 1 to Level 3, for a total number of cells of 1389, 4059, and 14,731, respectively. All computational models are created with the Salome platform⁵⁷ using the NETGEN algorithm, and are progressively refined in the region closer to the punch rounded edge, as shown in the detail in Figure 20 for the finest mesh. Regarding the settings for the sigmoid blending function, the relative offset is set to zero and the half-width is set to 0.1 for all the simulations, leading to an average relative interpenetration (with respect to the cell size) of $\sim 5 \cdot 10^{-4}$ on the finest mesh. The sigmoid function is truncated to limit the emergence of spurious positive traction forces along the rounded edge.

The benchmark is reproduced twice, first as a frictionless contact case and then considering a friction coefficient α equal to 0.1. The reference solution for both conditions is obtained with Code Aster⁶² on the highest mesh refinement level or Level 3.³ The contact formulation used in the Code Aster simulations is the generalized Newton method for both normal and frictional contact stresses. It is described as a continuous formulation of the Signorini–Coulomb law in the form of augmented Lagrangian.⁶³ Only for the frictionless case, we perform an additional FV simulation for each mesh refinement level employing a boundary condition based on the penalty method and using a standard penalty scale factor of 0.1.

In the three following graphs, we show the axial displacement (Figure 21), the contact stress (Figure 22), and the axial displacement (Figure 23) calculated along the top horizontal edge of the foundation. In these plots, we also include the reference solution from Code Aster and the solution obtained with the penalty method. To avoid overcrowded graphs, we display the results for the frictionless contact case (left part of the figures) and those for the frictional contact (right part of the figures) separately, and we show only the results obtained on the finest mesh, Level 3.

The figures reveal that the accuracy of the implicit contact methodology proposed in this article compares well against the more traditional penalty method. The results are also in line with those obtained by Cardiff with his FV penalty method contact stress solver,²¹ although the convergence rate shown in this article might seem faster due to the use of local refinement at the edges of the contact region.

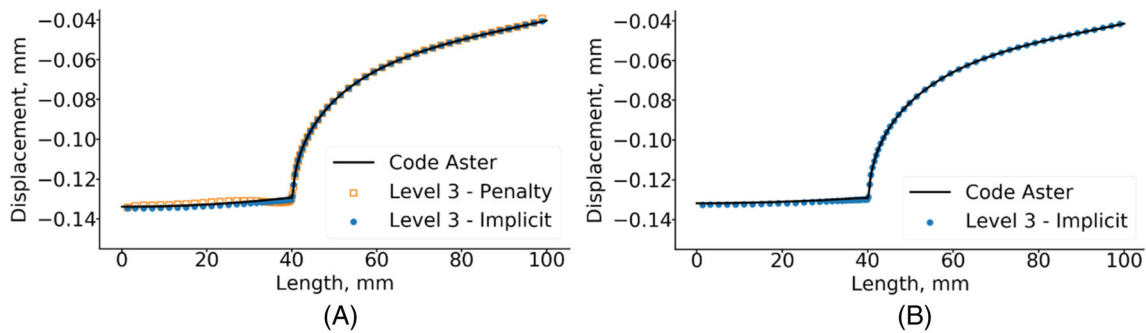


FIGURE 21 Punch test with rounded edge, axial displacement on the foundation top edge. On the left, the results obtained for the frictionless contact case. On the right, the results obtained with a friction coefficient equal to 0.1. (A) Frictionless case. (B) Frictional case

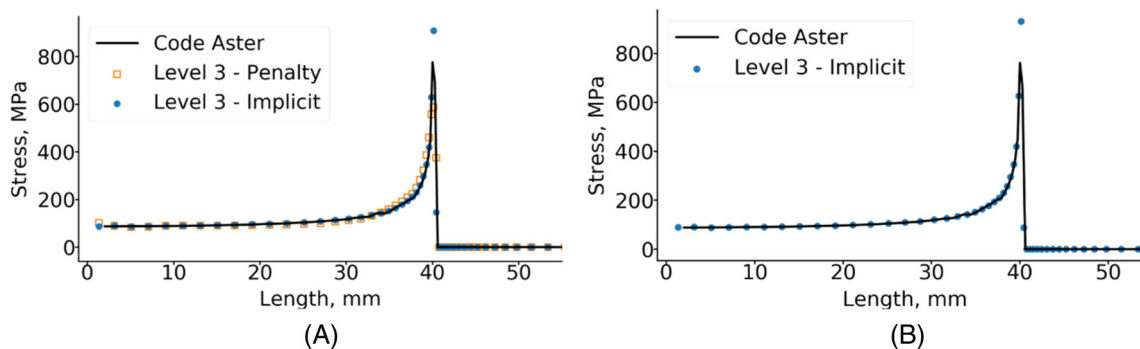


FIGURE 22 Punch test with rounded edge, contact stress on the foundation top edge. On the left, the results obtained for the frictionless contact case. On the right, the results obtained with a friction coefficient equal to 0.1. (A) Frictionless case. (B) Frictional case

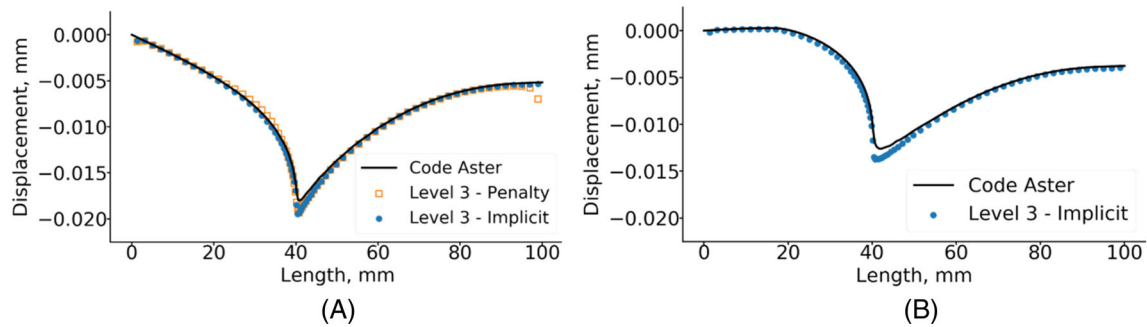


FIGURE 23 Punch test with rounded edge, radial displacement on the foundation top edge. On the left, the results obtained for the frictionless contact case. On the right, the results obtained with a friction coefficient equal to 0.1. (A) Frictionless case. (B) Frictional case

TABLE 2 Punch test with rounded edge: Running times for the frictionless case on wedge geometry

Mesh refinement	Level 1	Level 2	Level 3
Number of cells	1389	4059	14,731
Number of faces on foundation top	31	43	70
Implicit contact running time	3 s	7 s	33 s
Penalty method running time	14 s	28 s	115 s
Speedup	4.7×	4×	3.5×

Note: The results are shown for each mesh refinement level (the number of cells and the number of faces is indicated) both for the implicit and penalty method.

Some minor differences with the solution from Code Aster seem to remain, however, as it emerges from a closer look at the radial displacement curves. Indeed, the implicit contact method (in agreement with the penalty method) predicts a sharper and more peaked profile where the rounded edge touches the foundation, both for the frictionless and frictional contact cases. It is difficult to identify the main reason for these differences, as the reference solution is not analytic. It is likely, however, that this is simply due to numerical differences that will become smaller as the mesh is further refined in both codes.

Regarding the difference in performance between the penalty and implicit contact methods for the FV-based stress solver, the implicit methodology consistently proves to converge more quickly. Table 2 summarizes the running time required by the two methods on the three mesh refinement levels (the comparison can be done for the frictionless case only as the implementation at hand of the penalty method does not include frictional forces). The table also indicates the speedup provided by the implicit contact method, which varies between 4.7 for the coarsest to 3.5 for the finest mesh. These values, although considerably smaller than those obtained for the previous punch test benchmark (Case 4), are still quite significant. As mentioned in the analysis of the previous benchmark, the decrease in performance gain with increasing mesh refinement is expected, considering that the ratio of the number of boundary faces to the total number of cells ratio decreases. Each simulation was performed on an Intel Xeon CPU E5-1660 v4 with eight 3.2 GHz cores, using a single core. A relaxation factor of 0.999 for the equation solution was necessary to reach convergence when employing the penalty-based boundary condition, while a relaxation factor of 0.9 for the displacement field was used to stabilize the solution with the implicit contact methodology.

As done for the previous benchmark case, separate simulations were performed to verify that the performance boost obtained with the implicit methodology becomes less relevant or negligible when considering inertial forces or fixed displacement scenarios, all cases for which the use of an equation relaxation factor is not necessary.

To test the applicability of the methodology for 3D geometries, the frictionless punch test is performed once again using 3D computational models. This time, the three refinement levels contain a total of 13,211, 33,645, and 131,142 cells, from Level 1 to Level 3 respectively. Both the implicit contact and penalty method boundary conditions are used, so that the computation time can be compared also for a 3D simulation. To save computational resources, the model

includes only a quarter of the geometry, making use of the symmetry of the setup. The mesh, shown in Figure 24 for the highest refinement level, is composed mainly of tetrahedral cells and it is generated with the NETGEN algorithm from the SALOME platform.

The results of the 3D simulations compare well against the solutions obtained previously on the 2D wedge geometry. As an example, in Figure 25 we show the axial displacement along the horizontal top edge of the foundation calculated on the most refined 3D mesh. For comparison, we include in the graph the profiles obtained with Code Aster and with the implicit contact algorithm on the most refined 2D wedge model (i.e., the same results already shown in Figure 21).

Finally, as done for the 2D wedge models, we report the running time required for the 3D simulations in Table 3. The gain in performance guaranteed by the implicit contact method is larger than for the respective 2D cases, although it is



FIGURE 24 Punch test with rounded edge: Three-dimensional computational model for the finest mesh refinement level

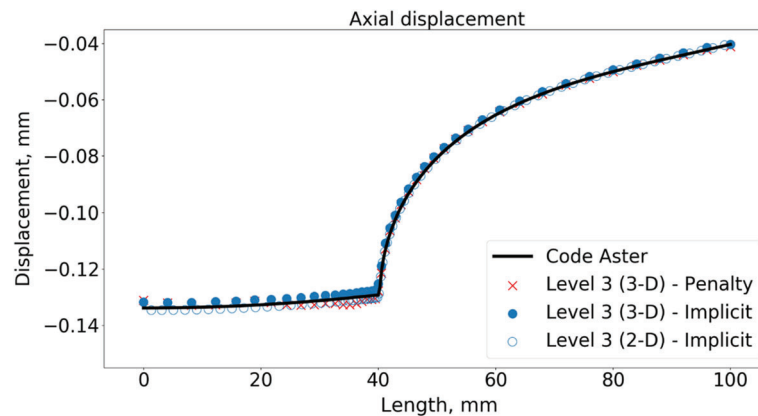


FIGURE 25 Punch test with rounded edge: Axial displacement on foundation top horizontal edge calculated with the most refined 3D model. The reference solution calculated with Code Aster on a 2D wedge model (and the respective implicit contact 2D solution) is included in the graph

TABLE 3 Punch test with rounded edge: Running times for the frictionless case on 3D geometry

Mesh refinement	Level 1, 3D	Level 2, 3D	Level 3, 3D
Number of cells	13,211	33,645	131,142
Number of faces on foundation top	862	1976	4854
Implicit contact running time	15 s	72 s	535 s
Penalty method running time	661 s	1328 s	5205 s
Speedup	44.1×	18.4×	9.7×

Note: The results are shown for each mesh refinement level (the number of cells and the number of faces is indicated) both for the implicit and penalty method.

even more sensitive to mesh refinement. This confirms that the speedup given by the implicit methodology is case and mesh dependent.

5 | CONCLUSIONS

In this article, we have presented a novel (semi-)implicit methodology for the solution of static, force-loading contact problems with finite volume stress solvers. This article joins the international efforts carried out in the last 30 years to make the finite volume method a viable alternative to the more common finite element techniques for solid mechanics problems.

The implicit algorithm presented in this article has been implemented as a custom boundary condition in a cell-centered finite volume stress solver built using the OpenFOAM C++ library. The contact methodology was inspired by the treatment of bi-material interfaces inside the domain developed by Tuković and is based on a semi-implicit discretization of the contact stresses which allows a straightforward inclusion of frictional forces. The introduction of a sigmoid blending function, that interpolates between contact stresses and gap pressure, extends the procedure to cases where the gap is partially open. Further, a treatment for non-conformal boundaries between for the two bodies in contact, which is based on OpenFOAM's arbitrary mesh interface, has been derived. Finally, the implicit algorithm is accompanied by correctors for non-orthogonal boundaries.

The new method has been validated against five benchmarks with known solution, including cases with fully closed and partially opened gap. The results show good agreement with the reference solutions, either analytic or obtained using the well-known finite element code named Code Aster. For the test cases studied in this article, the performance gain compared to the penalty method was case-dependent but always substantial, as it varied between a factor of 3.5× and a factor of 1000×.

Some problems remain and some aspects of the methodology require a more meticulous analysis. For example, the choice of the contact tolerances, that is, selecting the correct parameters of the blending function, is more complex and less intuitive than choosing a single penalty scale factor when using a penalty method boundary condition. Dedicated studies for cases of interest are needed to define best-practice values for these coefficients and the development of a semi-automated algorithm to identify the optimal blending coefficients would certainly be beneficial. Also, additional efforts should be carried out to study the frictional response of the methodology more in detail.

DATA AVAILABILITY STATEMENT

The data supporting the findings of this study are available from the corresponding author upon reasonable request. They will be made publicly available via a git repository once the OFFBEAT solver used in this study is released. The release is scheduled for early 2022.

ENDNOTES

¹ It is worth noting that the correction vectors presented in Equations (45) and (46), have different units with respect to the correction vector given in Equation (12). The traction vector defined in Equation (4) has the units of pressure (Pa). Then it follows that the correction vector in Equation (12) is dimensionless, while the vectors presented in Equations (45) and (46), have the units of length (m), as they must have the same units of the δ vectors (which are distance vectors).

²These vectors coincide with the search direction used by the face area weighting technique to calculate the weighting factors for the AMI mapping.

³The reference documents^{56,60,61} report a slightly different solution if compared to the Code Aster results presented in this article. This discrepancy is due to the much coarser mesh used in the mentioned publications. Before the preparation of this paper, by running Code Aster on the two finest mesh refinement levels (Level 2 and Level 3), we verified that the mesh used to obtain the results presented in References 56, 60, and 61 is relatively coarse and the solution therefore not fully converged.

ORCID

Alessandro Scolaro  <https://orcid.org/0000-0001-9701-529X>

REFERENCES

1. Wriggers P. *Computational Contact Mechanics*. Springer; 2006.
2. De Lorenzis L, Wriggers P, Weißenfels C. Computational contact mechanics with the finite element method. In: Stein E, Borst R, Hughes TJR, eds. *Encyclopedia of Computational Mechanics*. 2nd ed.; John Wiley and Sons; 2017:1-45.
3. Belgacem FB, Hild P, Laborde P. The mortar finite element method for contact problems. *Math Comput Model*. 1998;28(4-8):263-271. doi:10.1016/S0895‐7177(98)00121‐6
4. Laursen T. A.. *Computational Contact and Impact Mechanics*. 1st ed. Berlin, Heidelberg: Springer; 2003.
5. Zavarise G, Wriggers P, *Trends in computational Contact Mechanics*. Lecture Notes in Applied and Computational Mechanics (LNACM). Vol 58; 1st ed. Berlin Heidelberg: Springer-Verlag; 2011.
6. Popp A, Wriggers P, eds. *Contact Modeling for Solids and Particles*. Vol 585; 1st ed. Berlin Heidelberg: Springer International Publishing; 2018.
7. Demirdžić I, Martinović D, Ivanković A. Numerical simulation of thermal deformation in welded workpiece. *Zavarivanje*. 1988;31(5-6):209-219.
8. Demirdžić I, Martinović D. Finite volume method for thermo-elasto-plastic stress analysis. *Comput Methods Appl Mech Eng*. 1993;109(3-4):331-349. doi:10.1016/0045‐7825(93)90085‐C
9. Jasak H, Weller HG. *Application of the finite volume method and unstructured meshes to linear elasticity*. Int J Numer Methods Eng; 2000. doi:10.1002/(SICI)1097‐0207(20000520)48:2<267::AID‐NME884>3.0.CO;2‐Q
10. Tuković Ž, Jasak H. Updated Lagrangian finite volume solver for large deformation dynamic response of elastic body. *Trans Famena*. 2007;31(1):55-70.
11. Maneeratana K. *Development of the Finite Volume Method for Non-linear Structural Applications*. PhD thesis. London, UK: Department of Mechanical Engineering at Imperial College; 2000.
12. Cardiff P, Tuković Z, Jasak H, Ivanković A. A block-coupled finite volume methodology for linear elasticity and unstructured meshes. *Comput Struct*. 2016;175:100-122. doi:10.1016/j.compstruc.2016.07.004
13. Fryer YD, Bailey C, Cross M, Lai CH. A control volume procedure for solving the elastic stress-strain equations on an unstructured mesh. *App Math Model*. 1991;15(11-12):639-645. doi:10.1016/S0307‐904X(09)81010‐X
14. Beale SB, Jülich F. Stress distribution in a plate subject to uniaxial loading. *PHOENICS J Comput Fluid Dyn*. 1990;3:255-287. <https://www.researchgate.net/publication/349929237>
15. Ebrahimnejad M, Fallah N, Khoei AR. New approximation functions in the meshless finite volume method for 2D elasticity problems. *Eng Anal Bound Elem*. 2014;46:10-22. doi:10.1016/J.ENGANABOUND.2014.04.023
16. Sevilla R, Giacomini M, Huerta A. A face-centred finite volume method for second-order elliptic problems. *Int J Numer Methods Eng*. 2018;115(8):986-1014. doi:10.1002/NME.5833
17. Cardiff P, Demirdžić I. Thirty years of the finite volume method for solid mechanics. *Arch Comput Methods Eng*. 2021;28(5):3721-3780. doi:10.1007/S11831‐020‐09523‐0
18. Jasak H, Weller HG. Finite volume methodology for contact problems of linear elastic solids. Proceedings of the 3rd International Conference of Croatian Society of Mechanics; 2000.
19. Cardiff P, Karač A, Ivanković A. Development of a finite volume contact solver based on the penalty method. *Comput Mater Sci*. 2012;64:283-284. doi:10.1016/j.commatsci.2012.03.011
20. Cardiff P, Tuković Ž, De Jaeger P, Clancy M, Ivanković A. A Lagrangian cell-centred finite volume method for metal forming simulation. *Int J Numer Methods Eng*. 2017;109(13):1777-1803. doi:10.1002/nme.5345
21. Cardiff P, Karač A, De Jaeger P, et al. An open-source finite volume toolbox for solid mechanics and fluid-solid interaction simulations; 2018. arXiv preprint arXiv 1808.10736.
22. Škurić V, De Jaeger P, Jasak H. Lubricated elastoplastic contact model for metal forming processes in OpenFOAM. *Comput Fluids*. 2018;172:226-240. doi:10.1016/J.COMPFLUID.2018.06.011
23. Batistić I, Tuković Ž, Cardiff P. A segment-to-segment algorithm for finite volume MECHANICAL contact simulations. Proceedings of the 15th OpenFOAM Workshop; 2020. http://openfoam-extend.sourceforge.net/OpenFOAM_Workshops/OFW15_2020_Arlington/program/s34.html

24. Taylor G, Breiguine V, Bailey C, Cross M. An augmented Lagrangian contact algorithm employing a vertex-based finite volume method. Proceedings of the 8th Annual Conference of the Association for Computational Mechanics in Engineering (ACME 2000); April 16-19, 2000; University of Greenwich, London, UK.
25. Bessonov NM, Golovashchenko SF, Volpert VA. Numerical modelling of contact elastic-plastic flows. *Math Model Nat Phenom*. 2009;4(1):44-87. doi:10.1051/MMNP/20094103
26. Suliman R, Oxtoby OF, Malan AG, Kok S. An enhanced finite volume method to model 2D linear elastic structures. *App Math Model*. 2014;38(7-8):2265-2279. doi:10.1016/J.APM.2013.10.028
27. Todreas N. E., Kazimi M. *Nuclear Systems Volume I*. CRC Press; 2011.
28. Olander DR. *Fundamental Aspects of Nuclear Reactor Fuel Elements*. U.S. Department of Energy; 1976.
29. Van Uffelen P, Hales J, Li W, Rossiter G, Williamson R. A review of fuel performance modelling. *J Nucl Mater*. 2019. doi:10.1016/j.jnucmat.2018.12.037516:373-412.
30. Tuković Ž, Ivanković A, Karač A. Finite-volume stress analysis in multi-material linear elastic body. *Int J Numer Methods Eng*. 2013;93(4):400-419. doi:10.1002/nme.4390
31. Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput Phys*. 1998;12(6):620-631. doi:10.1063/1.168744
32. Das S, Mathur SR, Murthy JY. An unstructured finite-volume method for structure-electrostatics interactions in MEMS. *Numer Heat Transf Part B Fundam*. 2011;60(6):425-451. doi:10.1080/10407790.2011.628252
33. Brandt A. Multilevel adaptive computations in fluid dynamics. *AIAA J*. 1980;18(10):1165-1172. doi:10.2514/3.50867
34. Behrens T. OpenFOAM's basic solvers for linear systems of equations. *Chalmers Dep Appl Mech*. 2009. http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2008/TimBehrens/tibeh-report-fin.pdf
35. Demirdžić I, Muzaferija S. Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Comput Methods Appl Mech Eng*. 1995;125(1-4):235-255. doi:10.1016/0045‐7825(95)00800‐G
36. Cardiff P. *Development of the Finite Volume Method for Hip Joint Stress Analysis*. PhD thesis. Dublin: School of Mechanical and Materials Engineering University College; 2012.
37. Farrell PE, Maddison JR. Conservative interpolation between volume meshes by local Galerkin projection. *Comput Methods Appl Mech Eng*. 2011;200(1-4):89-100. doi:10.1016/j.cma.2010.07.015
38. OpenFOAM 2.1.0: arbitrary mesh interface | OpenFOAM. Accessed December 23, 2020. <https://openfoam.org/release/2-1-0/ami/>
39. Beaudoin M, Jasak H. Development of a generalized grid Interface for turbomachinery simulations with OpenFOAM. *Talk*. 2008;2008:1-11. [papers3://publication/uuid/81CCD00D-DF48-4595-B591-C577955CEA06](https://publication/uuid/81CCD00D-DF48-4595-B591-C577955CEA06).
40. The foam-extend project. Accessed December 23, 2020. <https://sourceforge.net/projects/foam-extend/>
41. Scolaro A, Clifford I, Fiorina C, Pautz A. The OFFBEAT multi-dimensional fuel behavior solver. *Nucl Eng Des*. 2020;358:110416. doi:10.1016/j.nucengdes.2019.110416
42. Scolaro A, Clifford I, Fiorina C & Pautz A Cladding plasticity modeling with the multidimensional fuel performance code OFFBEAT. Global/TopFuel 2019 Proceedings.
43. Scolaro A, Clifford I, Fiorina C, Pautz A. First steps towards the development of a 3D nuclear fuel behavior solver with openfoam. Proceedings of the International Conference on Nuclear Engineering, Proceedings, ICONE; Vol. 3, 2018. doi: 10.1115/ICONE26‐82381
44. Clifford I, Pecchia M, Mukin R, Cozzo C, Ferroukhi H, Gorzel A. Studies on the effects of local power peaking on heat transfer under dryout conditions in BWRs. *Ann Nucl Energy*. 2019;130:440-451. doi:10.1016/j.anucene.2019.03.017
45. Scolaro A. *Development of a Novel Finite Volume Methodology for Multi-Dimensional Fuel Performance Applications*. EPFL; 2021.
46. The OpenFOAM foundation. Accessed February 05, 2021. <https://openfoam.org/>
47. Phan AV, Mukherjee S. The multi-domain boundary contour method for interface and dissimilar material problems. *Eng Anal Bound Elem*. 2009;33(5):668-677. doi:10.1016/j.enganabound.2008.10.004
48. Nie GJ, Batra RC. Material tailoring and analysis of functionally graded isotropic and incompressible linear elastic hollow cylinders. *Compos Struct*. 2010;92(2):265-274. doi:10.1016/j.compstruct.2009.07.023
49. Taylor RL, Papadopoulos P. On a patch test for contact problems in two dimensions. *Nonlinear Comput Mech*. 1991;690-702.
50. Dittmann M, Franke M, Temizer I, Hesch C. Isogeometric analysis and thermomechanical mortar contact problems. *Comput Methods Appl Mech Eng*. 2014;274:192-212. doi:10.1016/j.cma.2014.02.012
51. Kim JY, Youn SK. Isogeometric contact analysis using mortar method. *Int J Numer Methods Eng*. 2012;89(12):1559-1581. doi:10.1002/nme.3300
52. McDevitt TW, Laursen TA. A mortar-finite element formulation for frictional contact problems. *Int J Numer Methods Eng*. 2000;48(10):1525-1547. doi:10.1002/1097‐0207(20000810)48:10<1525::AID‐NME953>3.0.CO;2‐Y
53. Popp A, Wohlmuth BI, Gee MW, Wall WA. Dual quadratic mortar finite element methods for 3D finite deformation contact. *SIAM J Sci Comput*. 2012;34(4):B421-B446. doi:10.1137/110848190
54. Puso MA, Laursen TA. A mortar segment-to-segment contact method for large deformation solid mechanics. *Comput Methods Appl Mech Eng*. 2004;193(6-8):601-629. doi:10.1016/j.cma.2003.10.010
55. Duong TX, De Lorenzis L, Sauer RA. A segmentation-free isogeometric extended mortar contact method. *Comput Mech*. 2019;63(2):383-407. doi:10.1007/s00466‐018‐1599‐0
56. Konter AWA. Advanced finite element contact benchmarks. FENET-UNOTT-DLE-09; 2005.

57. SALOME platform. Accessed December 29, 2020. <https://www.salome-platform.org/>
58. Johnson KL. *Contact Mechanics*. Cambridge University Press; 1985.
59. Roache PJ. Quantification of uncertainty in computational fluid dynamics. *Annu Rev Fluid Mech*. 2003;29:123-160. doi:10.1146/ANNUREV.FLUID.29.1.123
60. De Soza T. SSNA122-benchmark NAFEMS of validation of contact 2: punch (rounded edges); 2016. <http://www.gnu.org/copyleft/fdl.html>
61. 3D punch NAFEMS contact benchmark 2 | SimScale validation. <https://www.simscale.com/docs/validation-cases/3d-punch-rounded-edges-nafems-contact-benchmark-2/>
62. Code Aster; 2017. <http://www.code-aster.org>
63. Generalized Newton method for contact and friction—Code_Aster. Accessed September 3, 2021. <https://www.code-aster.org/spip.php?article612>.

How to cite this article: Scolaro A, Fiorina C, Clifford I, Pautz A. Development of a semi-implicit contact methodology for finite volume stress solvers. *Int J Numer Methods Eng*. 2021;1-30. doi: 10.1002/nme.6857