



Continuous gap contact formulation based on the screened Poisson equation

P. Areias^{1,2} · N. Sukumar³ · J. Ambrósio^{1,2}

Received: 26 November 2022 / Accepted: 25 February 2023
© The Author(s) 2023

Abstract

We introduce a PDE-based node-to-element contact formulation as an alternative to classical, purely geometrical formulations. It is challenging to devise solutions to nonsmooth contact problem with continuous gap using finite element discretizations. We herein achieve this objective by constructing an approximate distance function (ADF) to the boundaries of solid objects, and in doing so, also obtain universal uniqueness of contact detection. Unilateral constraints are implemented using a mixed model combining the screened Poisson equation and a force element, which has the topology of a continuum element containing an additional incident node. An ADF is obtained by solving the screened Poisson equation with constant essential boundary conditions and a variable transformation. The ADF does not explicitly depend on the number of objects and a single solution of the partial differential equation for this field uniquely defines the contact conditions for all incident points in the mesh. Having an ADF field to any obstacle circumvents the multiple target surfaces and avoids the specific data structures present in traditional contact-impact algorithms. We also relax the interpretation of the Lagrange multipliers as contact forces, and the Courant–Beltrami function is used with a mixed formulation producing the required differentiable result. We demonstrate the advantages of the new approach in two- and three-dimensional problems that are solved using Newton iterations. Simultaneous constraints for each incident point are considered.

Keywords Nonsmooth contact geometries · Contact algorithm · Finite strains · Eikonal equation · Approximate distance function · Screened Poisson equation

1 Introduction

Considerable interest has recently emerged with the enforcement of essential boundary conditions [1] and contact detection algorithms [2] using approximate distance functions (ADFs). Realistic contact frameworks for finite-strain problems make use of three classes of algorithms that are often coupled (see [3]):

1. Techniques of unilateral constraint enforcement. Examples of these are direct elimination (also reduced-gradient algorithms), penalty and barrier methods, and Lagrange multiplier methods with the complementarity condition [4]. Stresses from the underlying discretization are often used to assist the normal condition with Nitsche’s method [5].
2. Frictional effects (and more generally, constitutive contact laws) and cone-complementarity forms [6,7]. Solution paradigms are augmented Lagrangian methods for friction [8], cone-projection techniques [9] and the so-called yield-limited algorithm for friction [10].
3. Discretization methods. Of concern are distance calculations, estimation of tangent velocities and general discretization arrangements. In contemporary use are (see [3]) node-to-edge/face, edge/face-to-/edge/face and mortar discretizations.

In terms of contact enforcement and friction algorithms, finite displacement contact problems are typically addressed

✉ P. Areias
pedro.areias@tecnico.ulisboa.pt

¹ DEM - Departamento de Engenharia Mecânica, Instituto Superior Técnico, Universidade de Lisboa, Building Mechanics II, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal

² IDMEC - Instituto Superior Técnico, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal

³ Department of Civil and Environmental Engineering, UC-Davis- University of California at Davis, One Shields Avenue, Davis, CA 95616, U.S.A.

with well-established contact algorithms, often derived from solutions developed by groups working on variational inequalities and nonsmooth analysis. However, in the area of contact discretization and the related area of contact kinematics [11], there are still challenges to be addressed in terms of ensuring the complete robustness of implicit codes. One of the earliest papers on contact discretization was by Chan and Tuba [12], who considered contact with a plane and used symmetry to analyze cylinder-to-cylinder contact. Francavilla and Zienkiewicz [13] later proposed an extension to node-to-node contact in small strains, allowing for direct comparison with closed-form solutions. The extension to finite strains requires further development, and the logical development was the node-to-segment approach, as described in the work of Hallquist [14]. Although node-to-segment algorithms are known to entail many deficiencies, most of the drawbacks have been addressed. Jumps and discontinuities resulting from nodes sliding between edges can be removed by smoothing and regularization [15]. Satisfaction of patch-test, which is fundamental for convergence, can be enforced by careful penalty weighing [16,17]. It is well known that single-pass versions of the node-to-segment method result in interference and double-pass can produce mesh interlocking, see [18,19]. This shortcoming has eluded any attempts of a solution and has motivated the development of surface-to-surface algorithms. One of the first surface-to-surface algorithms was introduced by Simo, Wriggers, and Taylor [11]. Zavarise and Wriggers [20] presented a complete treatment of the 2D case and further developments were supported by parallel work in nonconforming meshes, see [21]. A review article on mortar methods for contact problems [22] where stabilization is discussed and an exhaustive detailing of most geometric possibilities of contact was presented by Farah, Wall and Popp [23]. That paper revealed the significant effort that is required to obtain a robust contact algorithm. An interesting alternative approach to contact discretization has been proposed by Wriggers, Schröder, and Schwarz [24], who use an intermediate mesh with a specialized anisotropic hyperelastic law to represent contact interactions. In the context of large, explicit codes, Kane et al. [25] introduced an interference function, or gap, based on volumes of overlapping, allowing non-smooth contact to be dealt in traditional smooth-based algorithms.

In addition to these general developments, there have been specialized algorithms for rods, beams, and other structures. Litewka et al. [26] as well as Neto et al. [27,28], have produced efficient algorithms for beam contact. For large-scale analysis of beams, cables and ropes, Meier et al. [29] have shown how beam specialization can be very efficient when a large number of entities is involved.

Recently, interest has emerged on using approximate distance functions [2,30–32] as alternatives to algorithms that heavily rely on computational geometry. These algorithms

resolve the non-uniqueness of the projection, but still require geometric computations. In [30], Wolff and Bucher proposed a local construction to obtain distances inside any object, but still require geometric calculations, especially for the integration scheme along surfaces. Liu et al. [31] have combined finite elements with distance potential discrete element method (DEM) in 2D within an explicit integration framework. A geometric-based distance function is constructed and contact forces stem from this construction. In [32], the analysis of thin rods is performed using classical contact but closed-form contact detection is achieved by a signed-distance function defined on a voxel-type grid. In [2], a pre-established set of shapes is considered, and a function is defined for each particle in a DEM (discrete element method) setting with a projection that follows. In the context of computer graphics and computational geometry, Macklin et al. [33] introduced an element-wise local optimization algorithm to determine the closest-distance between the signed-distance-function isosurface and face elements. Although close to what is proposed here, no solution to a partial differential equation (PDE) is proposed and significant geometric calculations are still required.

In this paper, we adopt a different method, which aims to be more general and less geometric-based. This is possible due to the equivalence between the solution of the Eikonal equation and the distance function [34]. It is worth noting that very efficient linear algorithms are available to solve regularized Eikonal equations, as discussed by Crane et al. [35]. The work in [35] provides a viable solution for contact detection in computational mechanics. Applications extend beyond contact mechanics and they provide a solution for the long-standing issue of imposing essential boundary conditions in meshfree methods [1].

We solve a partial differential equation (PDE) that produces an ADF (approximate distance function) that converges to the distance function as a length parameter tends to zero. The relation between the screened Poisson equation (also identified as Helmholtz-like equation), which is adopted in computational damage and fracture [36,37] and the Eikonal equation is well understood [38]. Regularizations of the Eikonal equation such as the geodesics-in-heat [35] and the screened Poisson equation are discussed by Belyaev and Fayolle [39]. We take advantage of the latter for computational contact mechanics. Specifically, the proposed algorithm solves well-known shortcomings of geometric-based contact enforcement:

1. Geometric calculations are reduced to the detection of a target element for each incident node.
2. The gap function $g(\mathbf{x})$ is continuous, since the solution of the screened Poisson equation is a continuous function.
3. The contact force direction is unique and obtained as the gradient of $g(\mathbf{x})$.

4. Since the Courant–Beltrami penalty function is adopted, contact force is continuous in the normal direction.

Concerning the importance of solving the uniqueness problem, Konyukhov and Schweizerhof [40] have shown that considerable computational geometry must be in place to ensure uniqueness and existence of node-to-line projection. Another important computational effort was presented by Farah et al. [23] to geometrically address all cases in 3D with mortar interpolation. Extensions to higher dimensions require even more intricate housekeeping. When compared with the geometrical approach to the distance function [2,32], the algorithm is much simpler at the cost of a solution of an additional PDE. Distance functions can be generated by level set solutions of the transport equation [41].

The remainder of this paper is organized as follows. In Sect. 2, the approximate distance function is introduced as the solution of a regularization of the Eikonal equation. In Sect. 3, details concerning the discretization are introduced. The overall algorithm, including the important step-control, is presented in Sect. 4. Verification and validation examples are shown in Sect. 5 in both 2D and 3D. Finally, in Sect. 6, we present the advantages and shortcomings of the present algorithm, as well as suggestions for further improvements.

2 Approximate distance function (ADF)

Let $\Omega \subset \mathbb{R}^d$ be a deformed configuration of a given body in d -dimensional space and Ω_0 the corresponding undeformed configuration. The boundaries of these configurations are Γ and Γ_0 , respectively. Let us consider deformed coordinates of an arbitrary point $\mathbf{x} \in \Omega$ and a specific point, called incident, with coordinates \mathbf{x}_I . For reasons that will become clear, we also consider a *potential* function $\phi(\mathbf{x}_I)$, which is the solution of a scalar PDE. We are concerned with an approximation of the signed-distance function. The so-called *gap function* is now introduced as a differentiable function $g[\phi(\mathbf{x}_I)]$ such that [3]:

$$g[\phi(\mathbf{x}_I)] \begin{cases} < 0 & \mathbf{x}_I \in \Omega \\ = 0 & \mathbf{x}_I \in \Gamma \\ > 0 & \mathbf{x}_I \notin \Omega \cup \Gamma \end{cases} . \tag{1}$$

If a unique normal $\mathbf{n}(\mathbf{x}_I)$ exists for $\mathbf{x}_I \in \Gamma$, we can decompose the gradient of $g[\phi(\mathbf{x}_I)]$ into parallel (\parallel) and orthogonal (\perp) terms: $\nabla g[\phi(\mathbf{x}_I)] = \{\nabla g[\phi(\mathbf{x}_I)]\}_\parallel + \{\nabla g[\phi(\mathbf{x}_I)]\}_\perp$, with $\{\nabla g[\phi(\mathbf{x}_I)]\}_\perp \parallel \mathbf{n}(\mathbf{x}_I)$. Since $g[\phi(\mathbf{x}_I)] = 0$ for $\mathbf{x}_I \in \Gamma$ we have $\nabla g[\phi(\mathbf{x}_I)] \parallel \mathbf{n}(\mathbf{x}_I)$ on those points. In the frictionless case, the normal contact force component is identified as f_c and contact conditions correspond to the following complementarity conditions [4]:

$$\begin{aligned} g[\phi(\mathbf{x}_I)] f_c &= 0, \\ f_c &\geq 0, \\ g[\phi(\mathbf{x}_I)] &\geq 0, \end{aligned} \tag{2}$$

or equivalently, $\langle gg[\phi(\mathbf{x}_I)] + f_c \rangle - f_c = 0$ where $\langle x \rangle = (x+|x|)/2$. The vector form of the contact force is given by $\mathbf{f}_c = f_c \nabla g[\phi(\mathbf{x}_I)]$. Assuming that a function $g[\phi(\mathbf{x}_I)]$, and its first and second derivatives are available from the solution of a PDE, we approximately satisfy conditions (2) by defining the contact force as follows:

$$\begin{aligned} \mathbf{f}_c &= f_c \nabla g[\phi(\mathbf{x}_I)] \\ &= \kappa \min\{0, g[\phi(\mathbf{x}_I)]\}^2 \nabla g[\phi(\mathbf{x}_I)] \quad (\kappa > 0), \end{aligned} \tag{3}$$

where κ is a penalty parameter for the Courant–Beltrami function [42,43]. The Jacobian of this force is given by:

$$\begin{aligned} \mathbf{J} &= \kappa \min\{0, g[\phi(\mathbf{x}_I)]\}^2 \nabla \otimes \nabla g[\phi(\mathbf{x}_I)] \\ &\quad + 2\kappa \min\{0, g[\phi(\mathbf{x}_I)]\} \nabla g[\phi(\mathbf{x}_I)] \otimes \nabla g[\phi(\mathbf{x}_I)]. \end{aligned} \tag{4}$$

Varadhan [44] established that the solution of the screened Poisson equation:

$$c_L \nabla^2 \phi(\mathbf{x}) - \phi(\mathbf{x}) = 0 \quad \text{in } \Omega \tag{5a}$$

$$\phi(\mathbf{x}) = 1 \quad \text{on } \Gamma \tag{5b}$$

produces an ADF given by $-c_L \log[\phi(\mathbf{x})]$. This property has been recently studied by Guler et al. [38] and Belyaev et al. [34,39]. The exact distance is obtained as a limit:

$$d(\mathbf{x}) = - \lim_{c_L \rightarrow 0} c_L \log[\phi(\mathbf{x})]. \tag{6}$$

which is the solution of the Eikonal equation. Proof of this limit is provided in Varadhan [44]. We transform the ADF to a signed-ADF, by introducing the sign of outer (+) or inner (−) consisting of

$$g(\mathbf{x}) = \pm c_L \log[\phi(\mathbf{x})]. \tag{7}$$

Note that if the plus sign is adopted in (7), then inner points will result in a negative gap $g(\mathbf{x})$. The gradient of $g(\mathbf{x})$ results in

$$\nabla g(\mathbf{x}) = \pm \frac{c_L}{\phi(\mathbf{x})} \nabla \phi(\mathbf{x}) \tag{8}$$

and the Hessian of $g(\mathbf{x})$ is obtained in a straightforward manner:

$$\begin{aligned} &\nabla \otimes \nabla [g(\mathbf{x})] \\ &= \pm \frac{c_L}{\phi(\mathbf{x})} \left\{ \nabla \otimes \nabla [\phi(\mathbf{x})] - \frac{1}{\phi(\mathbf{x})} \nabla \phi(\mathbf{x}) \otimes \nabla \phi(\mathbf{x}) \right\}. \end{aligned} \tag{9}$$

Note that c_L is a square of a characteristic length, i.e. $c_L = l_c^2$, which is here taken as a solution parameter.

Using a test field $\delta\phi(\mathbf{x})$, the weak form of (5) is written as

$$\begin{aligned} &\int_{\Omega} c_L \nabla \phi(\mathbf{x}) \cdot \nabla \delta\phi(\mathbf{x}) \, dV + \int_{\Omega} \phi(\mathbf{x}) \delta\phi(\mathbf{x}) \, dV \\ &= \int_{\Gamma} c_L \delta\phi \nabla \phi(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \, dA, \end{aligned} \tag{10}$$

where dA and dV are differential area and volume elements, respectively. Since an essential boundary condition is imposed on Γ such that $\phi(\mathbf{x}) = 1$ for $x \in \Gamma$, it follows that $\delta\phi(\mathbf{x}) = 0$ on Γ and the right-hand-side of (10) vanishes.

3 Discretization

In an interference condition, each interfering node, with coordinates \mathbf{x}_N , will fall within a given continuum element. The parent-domain coordinates ξ for the incident node \mathbf{x}_I also depends on element nodal coordinates. Parent-domain coordinates are given by:

$$\xi_I = \underset{\xi}{\operatorname{argmin}} [\|\mathbf{x}(\xi) - \mathbf{x}_I\|], \tag{11a}$$

and it is straightforward to show that for a triangle,

$$\xi_{I1} = \frac{x_{I2}(x_{11} - x_{31}) + x_{12}x_{31} - x_{11}x_{32} + x_{I1}(x_{32} - x_{12})}{x_{11}(x_{22} - x_{32}) + x_{12}(x_{31} - x_{21}) + x_{21}x_{32} - x_{22}x_{31}}, \tag{11b}$$

$$\xi_{I2} = \frac{x_{I2}(x_{21} - x_{11}) + x_{12}x_{21} - x_{11}x_{22} + x_{I1}(x_{12} - x_{22})}{x_{11}(x_{22} - x_{32}) + x_{12}(x_{31} - x_{21}) + x_{21}x_{32} - x_{22}x_{31}}, \tag{11c}$$

with similar expressions for a tetrahedron [45]. The continuum element interpolation is as follows:

$$\mathbf{x}(\xi) = \sum_{K=1}^{d+1} N_K(\xi) \mathbf{x}_K = \mathbf{N}(\xi) \cdot \mathbf{x}_N, \tag{12}$$

where $N_K(\xi)$ with $K = 1, \dots, d+1$ are the shape functions of a triangular or tetrahedral element. Therefore, (11a) can be written as:

$$\xi_I = \underset{\xi}{\operatorname{argmin}} [\|\mathbf{N}(\xi) \cdot \mathbf{x}_N - \mathbf{x}_I\|] = \mathbf{a}_N(\mathbf{x}_C) \tag{13}$$

In (13), we group the continuum node coordinates and the incident node coordinates in a single array $\mathbf{x}_C =$

$\{\mathbf{x}_N \ \mathbf{x}_I\}^T$ with cardinality $\#\mathbf{x}_C = d(d+2)$. We adopt the notation \mathbf{x}_N for the coordinates of the continuum element. For triangular and tetrahedral discretizations, ξ_I is a function of \mathbf{x}_N and \mathbf{x}_I :

$$\xi_I = \mathbf{a}_N \left(\begin{Bmatrix} \mathbf{x}_N \\ \mathbf{x}_I \end{Bmatrix} \right) = \mathbf{a}_N(\mathbf{x}_C). \tag{14}$$

The first and second derivatives of \mathbf{a}_N with respect to \mathbf{x}_C make use of the following notation:

$$\mathbf{A}_N(\mathbf{x}_C) = \frac{d\mathbf{a}_N(\mathbf{x}_C)}{d\mathbf{x}_C} \quad (d \times [d(d+2)]), \tag{15}$$

$$\mathcal{A}_N(\mathbf{x}_C) = \frac{d^2\mathbf{a}_N(\mathbf{x}_C)}{d\mathbf{x}_C^2} \quad (d \times [d(d+2)]^2). \tag{16}$$

Source code for these functions is available in [Github \[45\]](#). A mixed formulation is adopted with equal-order interpolation for the displacement \mathbf{u} and the function ϕ . For a set of nodal displacements \mathbf{u}_N and nodal potential values ϕ_N :

$$\mathbf{u}(\mathbf{x}_C) = \mathbf{N}_u(\mathbf{x}_C) \cdot \mathbf{u}_N, \tag{17a}$$

$$\phi(\mathbf{x}_C) = \mathbf{N}_\phi(\mathbf{x}_C) \cdot \phi_N, \tag{17b}$$

where in three dimensions

$$\mathbf{N}_u(\mathbf{x}_C) = \begin{bmatrix} \dots & N_K[\mathbf{a}_N(\mathbf{x}_C)] & 0 & 0 & \dots \\ \dots & 0 & N_K[\mathbf{a}_N(\mathbf{x}_C)] & 0 & \dots \\ \dots & 0 & 0 & N_K[\mathbf{a}_N(\mathbf{x}_C)] & \dots \end{bmatrix}, \tag{17c}$$

$$\mathbf{N}_\phi(\mathbf{x}_C) = [\dots \ N_K[\mathbf{a}_N(\mathbf{x}_C)] \ \dots]^T, \tag{17d}$$

where $\xi_I = \mathbf{a}_N(\mathbf{x}_C)$. First and second derivatives of $N_K[\mathbf{a}_N(\mathbf{x}_C)]$ are determined from the chain rule:

$$\frac{dN_K}{d\mathbf{x}_C} = \frac{dN_K}{d\xi_I} \cdot \mathbf{A}_N(\mathbf{x}_C), \tag{18}$$

$$\frac{d^2N_K}{d\mathbf{x}_C^2} = \mathbf{A}_N^T(\mathbf{x}_C) \cdot \frac{d^2N_K}{d\xi_I^2} \cdot \mathbf{A}_N(\mathbf{x}_C) + \frac{dN_K}{d\xi_I} \cdot \mathcal{A}_N(\mathbf{x}_C). \tag{19}$$

For the test function of the incident point, we have

$$\delta\phi(\mathbf{x}_C) = N_\phi(\mathbf{x}_C) \cdot \delta\phi_N + \phi_N \cdot \frac{dN_\phi(\mathbf{x}_C)}{d\mathbf{x}_C} \cdot \delta\mathbf{x}_C. \tag{20}$$

For linear continuum elements, the second variation of $\phi(\xi)$ is given by the following rule:

$$\begin{aligned} d\delta\phi(\mathbf{x}_C) &= \delta\phi_N \cdot \frac{dN_\phi(\mathbf{x}_C)}{d\mathbf{x}_C} \cdot d\mathbf{x}_C + d\phi_N \cdot \frac{dN_\phi(\mathbf{x}_C)}{d\mathbf{x}_C} \cdot \delta\mathbf{x}_C \\ &+ \phi_N \cdot \frac{d^2N_\phi(\mathbf{x}_C)}{d\mathbf{x}_C^2} : (\delta\mathbf{x}_C \otimes d\mathbf{x}_C) \end{aligned} \tag{21}$$

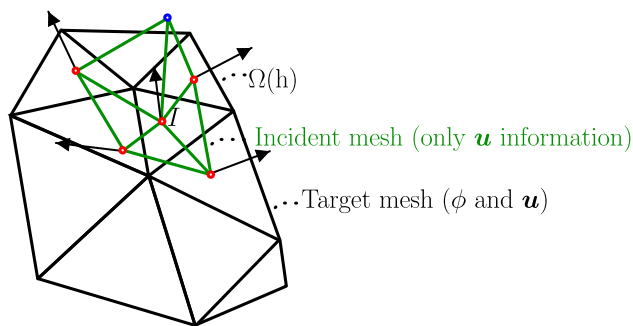


Fig. 1 Relevant quantities for the definition of a contact discretization for element e

Since the gradient of ϕ makes use of the continuum part of the formulation, we obtain:

$$\nabla\phi(\xi) = \phi_N \cdot \underbrace{\frac{dN_\phi(\xi)}{d\xi}}_{\frac{dN_\phi}{dx}} \cdot \mathbf{j}^{-1},$$

$$\nabla\delta\phi(\xi) = \delta\phi_N \cdot \frac{dN_\phi(\xi)}{d\xi} \cdot \mathbf{j}^{-1} \tag{22}$$

where \mathbf{j} is the Jacobian matrix in the deformed configuration. The element residual and stiffness are obtained from these quantities and available in Github [45]. Use is made of the Acegen [46] add-on to Mathematica [47] to obtain the source code for the final expressions.

4 Algorithm and step control

All nodes are considered candidates and all elements are targets. A simple bucket-sort strategy is adopted to reduce the computational cost. In addition,

Step control is required to avoid the change of target during Newton-Raphson iteration. The screened Poisson equation is solved for all bodies in the analyses. Figure 1 shows the simple mesh overlapping under consideration. The resulting algorithm is straightforward. Main characteristics are:

- All nodes are candidate incident nodes.
- All elements are generalized targets.

The modifications required for a classical nonlinear Finite Element software (in our case SimPlas [48]) to include this contact algorithm are modest. Algorithm 1 presents the main tasks. In this Algorithm, blue boxes denote the contact detection task, which here is limited to:

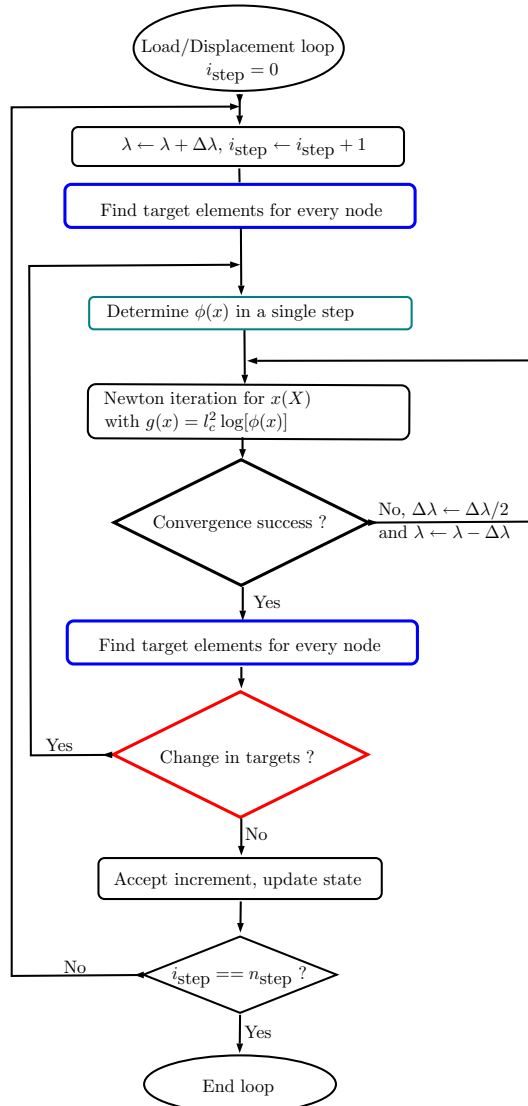
1. Detect nearest neighbor (in terms of distance) elements for each node. A bucket ordering is adopted.
2. Find if the node is interior to any of the neighbor elements by use of the shape functions for triangles and tetrahedra. This is performed in closed form.
3. If changes occurred in the targets, update the connectivity table and the sparse matrix addressing. Gustavson’s algorithm [49] is adopted to perform the updating in the assembling process.

In terms of detection, the following algorithm is adopted:

1. Find all exterior faces, as faces sharing only one tetrahedron.
2. Find all exterior nodes, as nodes belonging to exterior faces.
3. Insert all continuum elements and all exterior nodes in two bucket lists. Deformed coordinates of nodes and deformed coordinates of element centroids are considered.
4. Cycle through all exterior nodes
 - (a) Determine the element bucket from the node coordinates
 - (b) Cycle through all elements (e) in the $3^3 = 27$ buckets surrounding the node
 - i. If the distance from the node to the element centroid is greater than twice the edge size, go to the next element
 - ii. Calculates the projection on the element (ξ_I) and the corresponding shape functions $N(\xi_I)$.
 - iii. If $0 \leq N_K(\xi_I) \leq 1$ then e is the target element. If the target element has changed, then flag the solver for connectivity update.

Since the algorithm assumes a fixed connectivity table during Newton iterations, a verification is required after each converged iteration to check if targets have changed since last determined. If this occurs, a new iteration sequence with revised targets is performed.

Algorithm 1 Staggered contact algorithm based on the approximate distance function. λ is the load/displacement factor.



The only modification required to a classical FEM code is the solution of the screened-Poisson equation, the green box in Algorithm 1. The cost of this solution is negligible when compared with the nonlinear solution process since the equation is linear and scalar. It is equivalent to the cost of a steady-state heat conduction solution. Note that this corresponds to a staggered algorithm.

5 Numerical tests

Numerical examples are solved with our in-house software, SimPlas [48], using the new node-to-element contact ele-

ment. Only triangles and tetrahedra are assessed at this time, which provide an exact solution for ξ_j . Mathematica [47] with the add-on Acegen [46] is employed to obtain the specific source code. All runs are quasi-static and make use of a Neo-Hookean model. If \mathbf{C} is the right Cauchy-Green tensor, then

$$\mathbf{S} = 2^d \psi(\mathbf{C}) / d\mathbf{C}$$

where $\psi(\mathbf{C}) = \frac{\mu}{2} (\mathbf{C} : \mathbf{I} - 3) - \mu \log(\sqrt{\det \mathbf{C}}) + \frac{\chi}{2} \log^2(\sqrt{\det \mathbf{C}})$ with $\mu = \frac{E}{2(1+\nu)}$ and $\chi = \frac{E\nu}{(1+\nu)(1-2\nu)}$ being constitutive properties.

Patch test satisfaction

We employ a corrected penalty so that the contact patch test is satisfied in most cases. This is an important subject for convergence of computational contact solutions and has been addressed here with a similar solution to the one discussed by Zavarise and co-workers [16,17].

We remark that this is not a general solution and in some cases, our formulation may fail to pass the patch test. Figure 2 shows the effect of using a penalty weighted by the edge projection, see [16]. However, this is not an universal solution.

5.1 Two-dimensional compression

We begin with a quasi-static two-dimensional test, as shown in Fig. 3, using three-node triangles. This test consists of a compression of four polygons (identified as part 3) in Fig. 3 by a deformable rectangular punch (part 1 in the same Figure). The ‘U’ (part 2) is considered rigid but still participates in the approximate distance function (ADF) calculation. To avoid an underdetermined system, a small damping term is used, specifically 40 units with $L^{-2}MT^{-1}$ ISQ dimensions. Algorithm 1 is adopted with a pseudo-time increment of $\Delta t = 0.003$ for $t \in [0, 1]$.

For $h = 0.020$, $h = 0.015$ and $h = 0.010$, Fig. 4 shows a sequence of deformed meshes and the contour plot of $\phi(\mathbf{x})$. The robustness of the algorithm is excellent, at the cost of some interference for coarse meshes. To further inspect the interference, the contour lines for $g(\mathbf{x})$ are shown in Fig. 5. We note that coarser meshes produce smoothed-out vertex representation, which causes the interference displayed in Fig. 4. Note that $g(\mathbf{x})$ is determined from $\phi(\mathbf{x})$.

Using the gradient of $\phi(\mathbf{x})$, the contact direction is obtained for $h = 0.02$ as shown in Fig. 6. We can observe that for the star-shaped figure, vertices are poorly represented, since small gradients are present due to uniformity of $\phi(\mathbf{x})$ in these regions. The effect of mesh refinement is clear, with finer meshes producing a sharper growth of reaction when

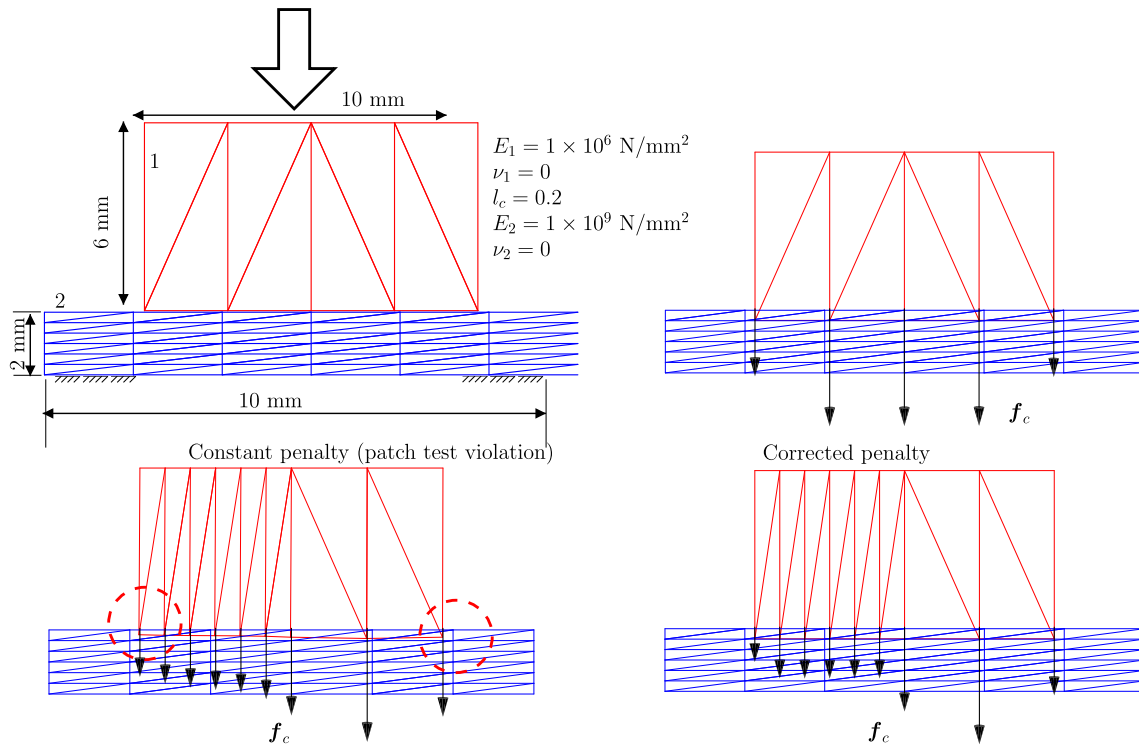
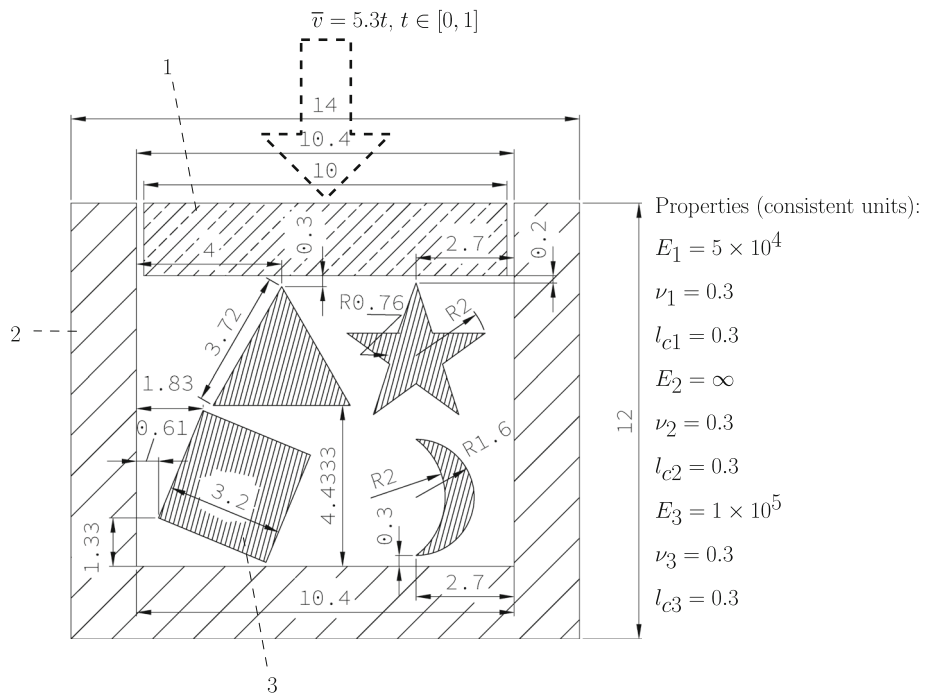


Fig. 2 Patch test satisfaction by using a corrected penalty

Fig. 3 Two-dimensional verification problem. Consistent units are used



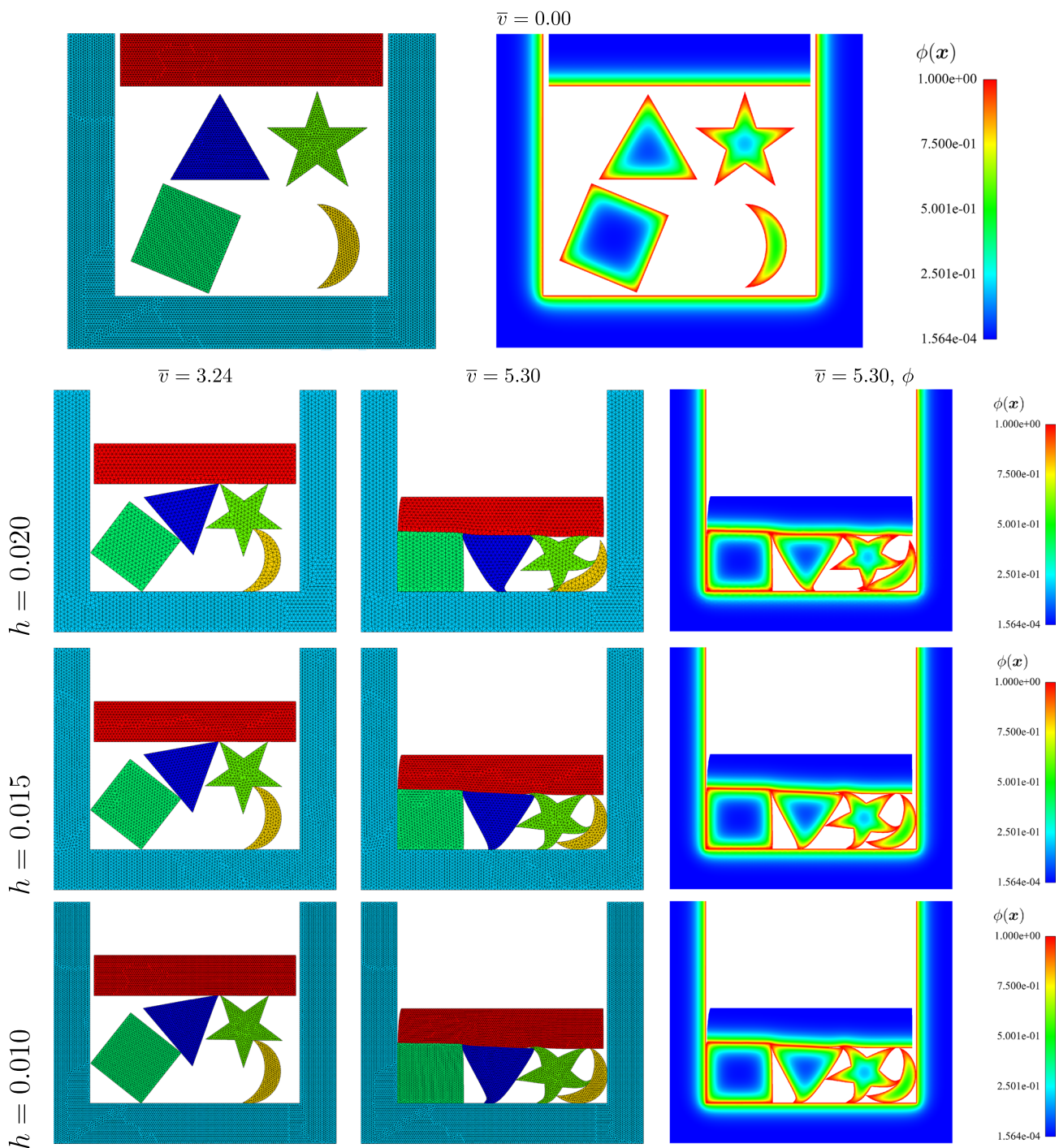


Fig. 4 Two-dimensional compression: sequence of deformed meshes and contour plot of $\phi(\mathbf{x})$

all four objects are in contact with each other. In contrast, the effect of the characteristic length l_c is not noticeable.

In terms of the effect of l_c on the fields $\phi(\mathbf{x})$ and $g(\mathbf{x})$, Fig. 7 shows that information, although $\phi(\mathbf{x})$ is strongly affected by the length parameter, $g(\mathbf{x})$ shows very similar spatial distributions although different peaks. Effects of h and l_c in the displacement/reaction behavior is shown in Fig. 8.

The mesh size h has a marked effect on the results up to $h = 0.0125$, the effect of l_c is much weaker.

5.2 Three-dimensional compression

In three dimensions, the algorithm is in essence the same. Compared with geometric-based algorithms, it significantly

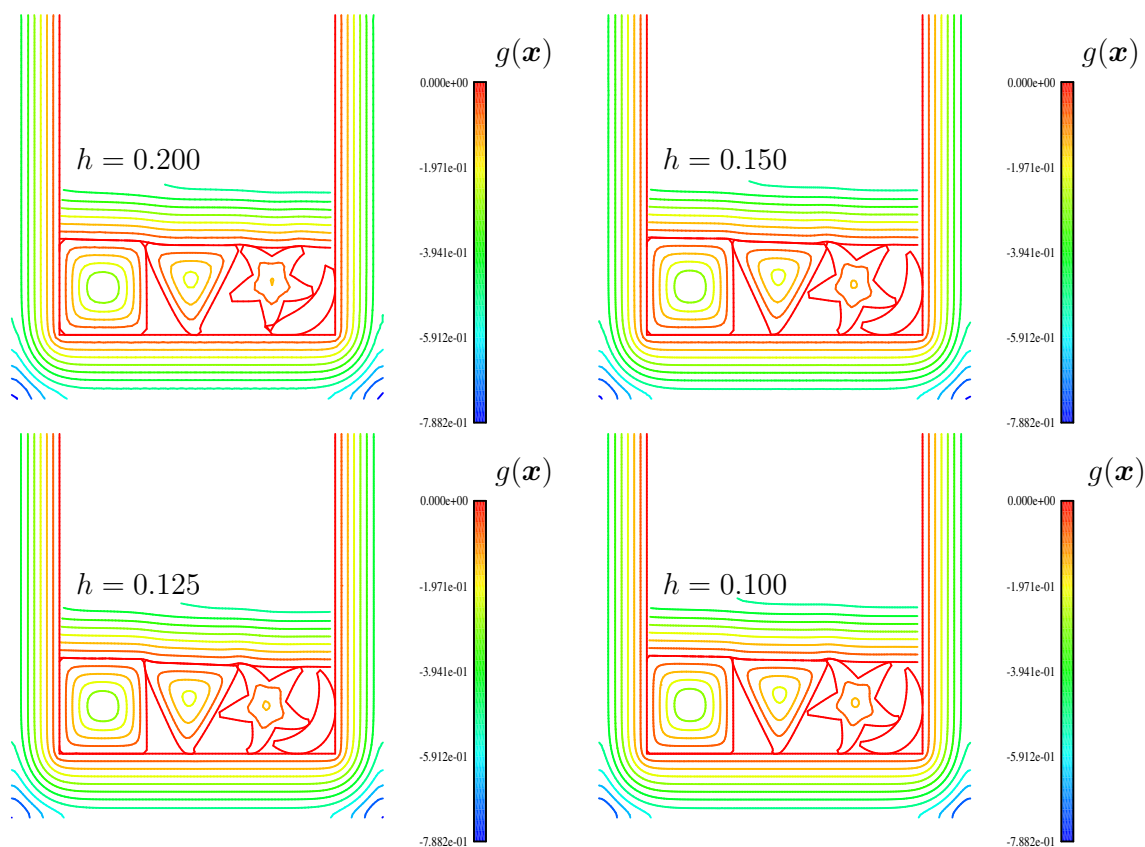


Fig. 5 Gap ($g(\mathbf{x})$) contour lines for the four meshes using $l_c = 0.3$ consistent units

reduces the coding for the treatment of particular cases (node-to-vertex, node-to-edge in both convex and non-convex arrangements). The determination of coordinates for each incident node is now performed on each tetrahedra, but the remaining tasks remain unaltered. We test the geometry shown in Fig. 9 with the following objectives:

- Assess the extension to the 3D case. Geometrical nonsmoothness is introduced with a cone and a wedge.
- Quantify interference as a function of l_c and κ as well as the strain energy evolution.

Deformed configurations and contour plots of $\phi(\mathbf{x})$ for this problem are presented in Fig. 9, and the corresponding CAD files are available on Github [45]. A cylinder, a cone and a wedge are compressed between two blocks. Dimensions of the upper and lower blocks are $10 \times 12 \times 2$ consistent units (the upper block is deformable whereas the lower block is rigid) and initial distance between blocks is 8 consistent units. Length and diameter of the cylinder are 7.15 and 2.86 (consistent units), respectively. The cone has a height of 3.27 consistent units and a radius of 1.87. Finally, the wedge has a width of 3.2, a radius of 3.2 and a swept angle of 30 degrees.

A compressible Neo-Hookean law is adopted with the following properties:

- Blocks: $E = 5 \times 10^4$ and $\nu = 0.3$.
- Cylinder, cone and wedge: $E = 1 \times 10^5$ and $\nu = 0.3$.

The analysis of gap violation, $v_{\max} = \sup_{\mathbf{x} \in \Omega} [-\min(0, g)]$ as a function of pseudotime $t \in [0, 1]$ is especially important for assessing the robustness of the algorithm with respect to parameters l_c and κ . For the interval $l_c \in [0.05, 0.4]$, the effect of l_c is not significant, as can be observed in Fig. 10. Some spikes are noticeable around $t = 0.275$ for $l_c = 0.100$ when the wedge penetrates the cone. Since κ is constant, all objects are compressed towards end of the simulation, which the gap violation. In terms of κ , effects are the same as in classical geometric-based contact. In terms of strain energy, higher values of l_c result in lower values of strain energy. This is to be expected, since smaller gradient values are obtained and the contact force herein is proportional to the product of the gradient and the penalty parameter. Convergence for the strain energy as a function of h is presented in Fig. 11. It is noticeable that l_c has a marked effect near the end of the compression, since it affects the contact force.

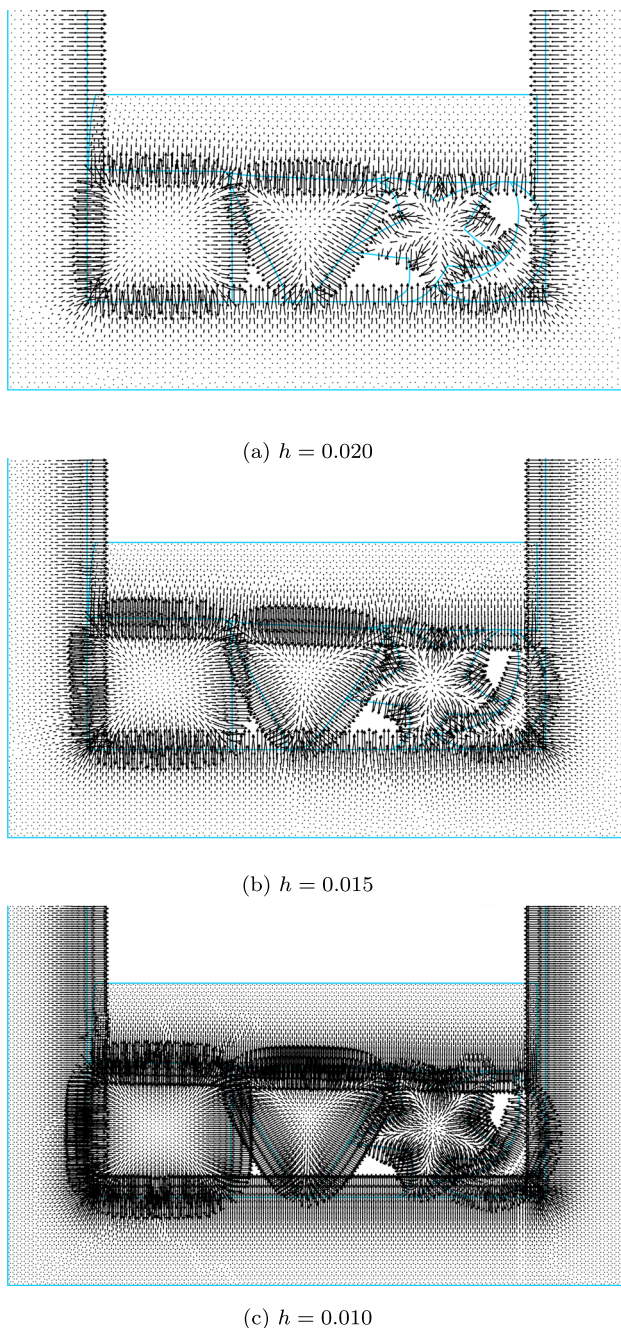


Fig. 6 Directions obtained as $\nabla\phi(\xi)$ for $h = 0.020, 0.015$ and 0.010

5.3 Two-dimensional ironing benchmark

This problem was proposed by Yang et al. [50] in the context of surface-to-surface mortar discretizations. Figure 12 shows the relevant geometric and constitutive data, according to [50] and [51]. We compare the present approach with the results of these two studies in Fig. 13. Differences exist in the magnitude of forces, and we infer that this is due to the continuum finite element technology. We use finite strain triangular elements with a compressible neo-Hookean law

[52]. The effect of l_c is observed in Fig. 13. As before, only a slight effect is noted in the reaction forces. We use the one-pass version of our algorithm, where the square indenter has the master nodes and targets are all elements in the rectangle. Note that, since the cited work includes friction, we use here a simple model based on regularized tangential law with a friction coefficient, $\mu_f = 0.3$.

5.4 Three-dimensional ironing benchmark

We now perform a test of a cubic indenter on a soft block. This problem was proposed by Puso and Laursen [18,19] to assess a mortar formulation based on averaged normals. The frictionless version is adopted [19], but we choose the most demanding case: $\nu = 0.499$ and the cubic indenter. Relevant data is presented in Fig. 14. The rigid $1 \times 1 \times 1$ block is located at 1 unit from the edge and is first moved down 1.4 units. After this, it moves longitudinally 4 units inside the soft block. The soft block is analyzed with two distinct meshes: $4 \times 6 \times 20$ divisions and $8 \times 12 \times 40$ divisions. Use is made of one plane of symmetry. A comparison with the vertical force in [19] is performed (see also [18] for a clarification concerning the force components). We allow some interference to avoid locking with tetrahedra. In [19], Puso and Laursen employed mixed hexahedra, which are more flexible than the crossed-tetrahedra we adopt here. Figure 15 shows the comparison between the proposed approach and the mortar method by Puso and Laursen [19]. Oscillations are caused by the normal jumps in gradient of $\phi(\mathbf{x})$ due to the classical C^0 finite-element discretization (between elements). Although the oscillations can be observed, the present approach is simpler than the one in Puso and Laursen.

6 Conclusions

We introduced a discretization and gap definition for a contact algorithm based on the solution of the screened Poisson equation. After a log-transformation, this is equivalent to the solution of a regularized Eikonal equation and therefore provides a distance to any obstacle or set of obstacles. This approximate distance function is smooth and is differentiated to obtain the contact force. This is combined with a Courant–Beltrami penalty to ensure a differentiable force along the normal direction. These two features are combined with a step-control algorithm that ensures a stable target-element identification. The algorithm avoids most of the geometrical calculations and housekeeping, and is able to solve problems with nonsmooth geometry. Very robust behavior is observed and two difficult ironing benchmarks (2D and 3D) are solved with success. Concerning the selection of the length-scale parameter l_c , which produces an exact solution for $l_c = 0$, we found that it should be the smallest value that is compatible

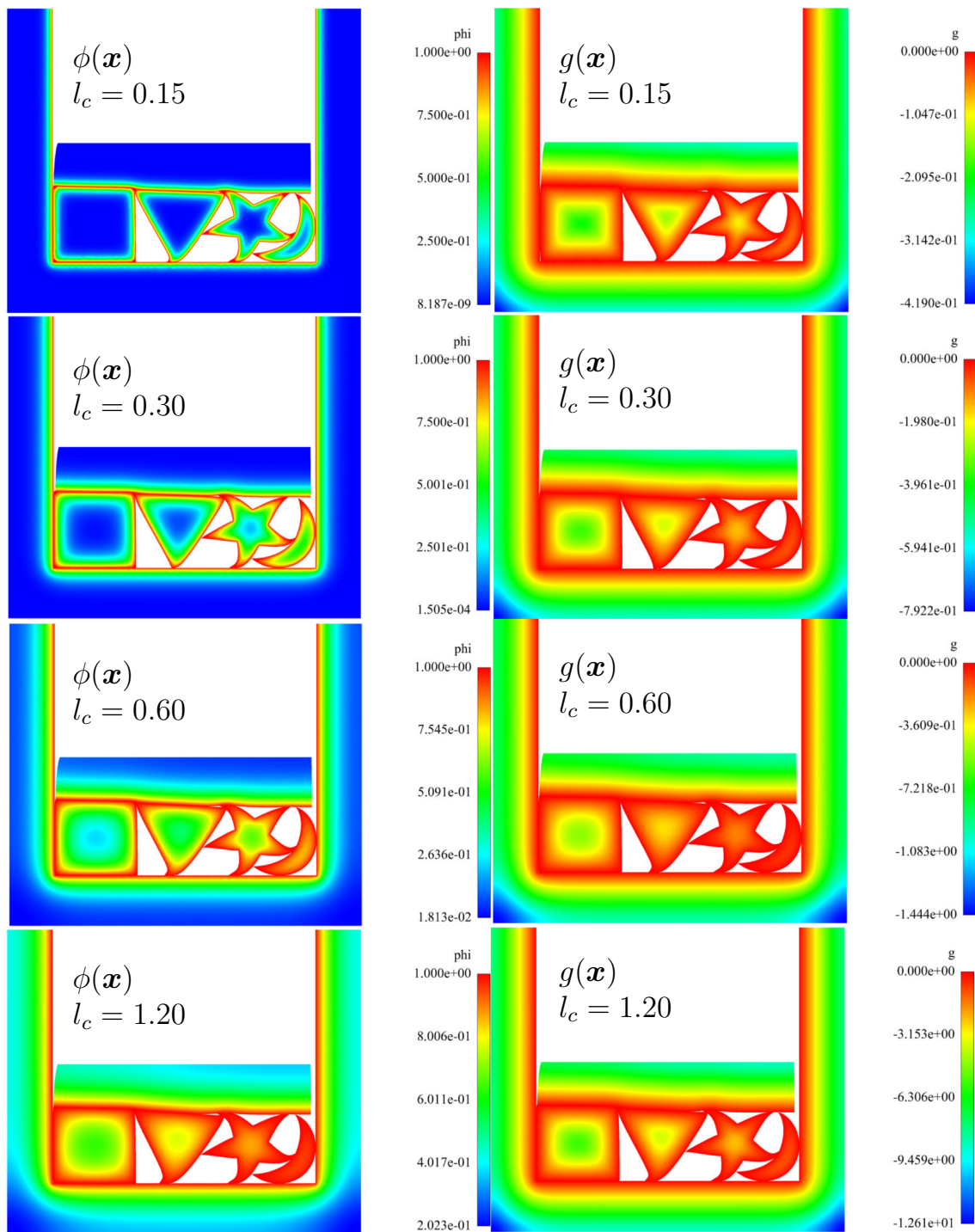


Fig. 7 Effect of l_c in the form of $\phi(\mathbf{x})$ and $g(\mathbf{x})$

with the solution of the screened Poisson equation. Too small of a l_c will produce poor results for $\phi(\mathbf{x})$. Newton–Raphson convergence was found to be stable, as well as nearly independent of l_c . In terms of further developments, a C^2 meshless

discretization is important to reduce the oscillations caused by normal jumps and we plan to adopt the cone projection method developed in [7] for frictional problems.

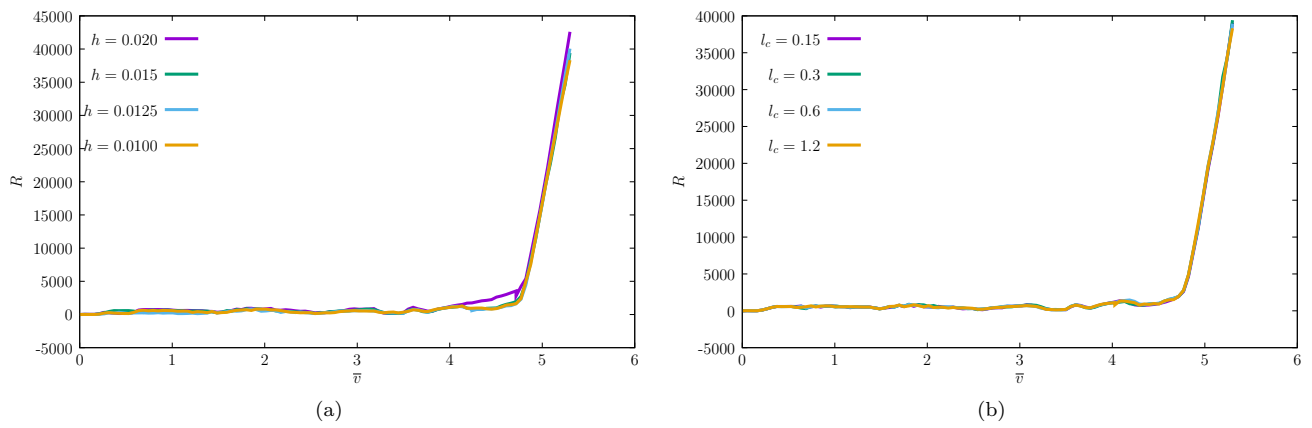
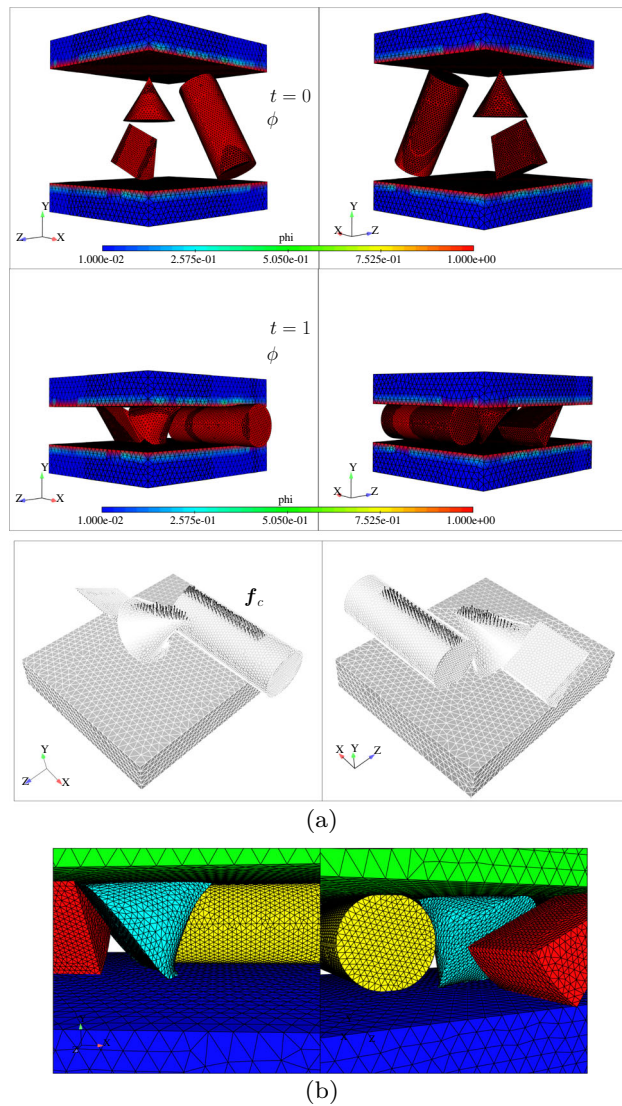


Fig. 8 Effect of h in (a) and effect of l_c in (b) in the displacement/reaction behavior

Fig. 9 Data for the three-dimensional compression test. **a** Undeformed and deformed configurations ($h = 0.025$). For geometric files, see [45]. **b** Detail of contact of cone with lower block and with the wedge. Each object is identified by a different color



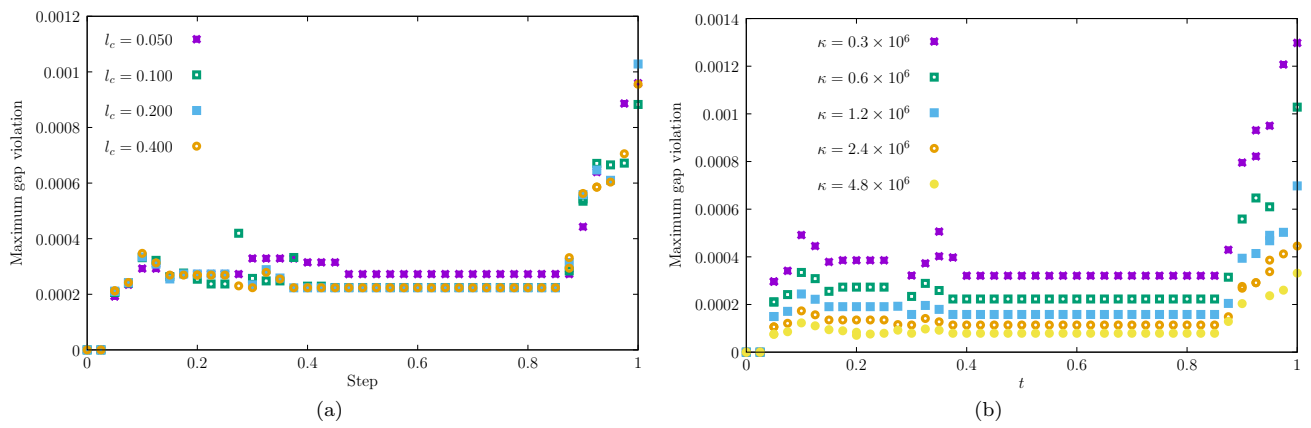


Fig. 10 **a** Effect of l_c on the maximum gap evolution over pseudotime ($\kappa = 0.6 \times 10^6$, $h = 0.030$) and **b** Effect of κ on the maximum gap evolution over pseudotime ($l_c = 0.2$)

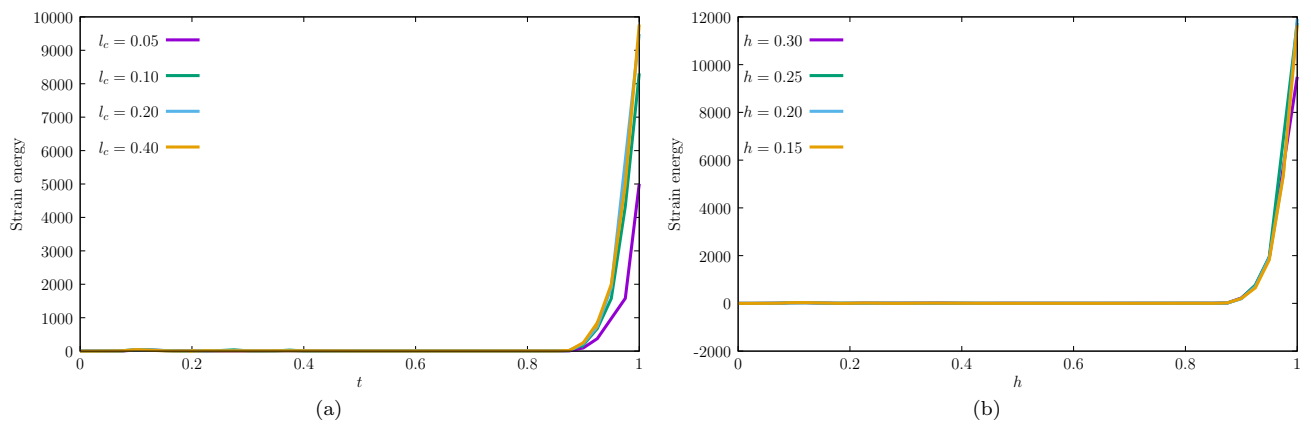
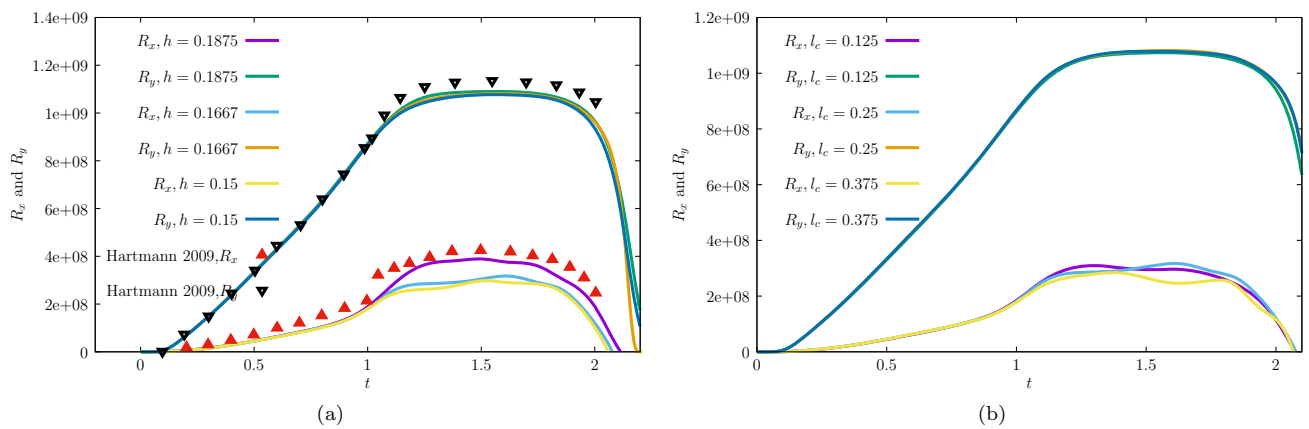
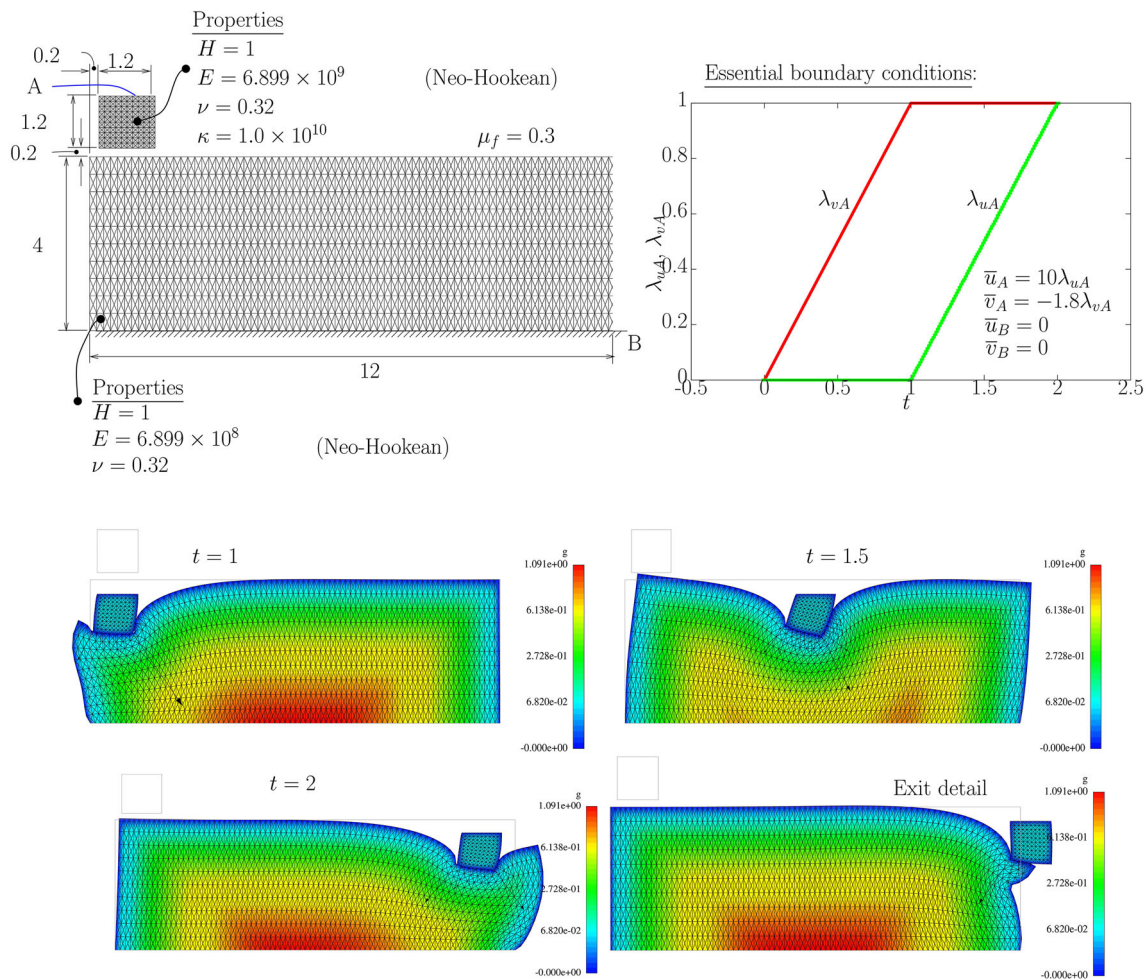


Fig. 11 **a** Effect of l_c on the strain energy ($h = 0.3$) and **b** Effect of h on the strain energy ($l_c = 0.2$). $\kappa = 0.6 \times 10^6$ is used



tion of the horizontal (R_x) and vertical (R_y) reactions and **b** Effect of l_c in the evolution of the horizontal (R_x) and vertical (R_y) reactions for $h = 0.1667$

Fig. 14 3D ironing: cube over soft block. Relevant data and results

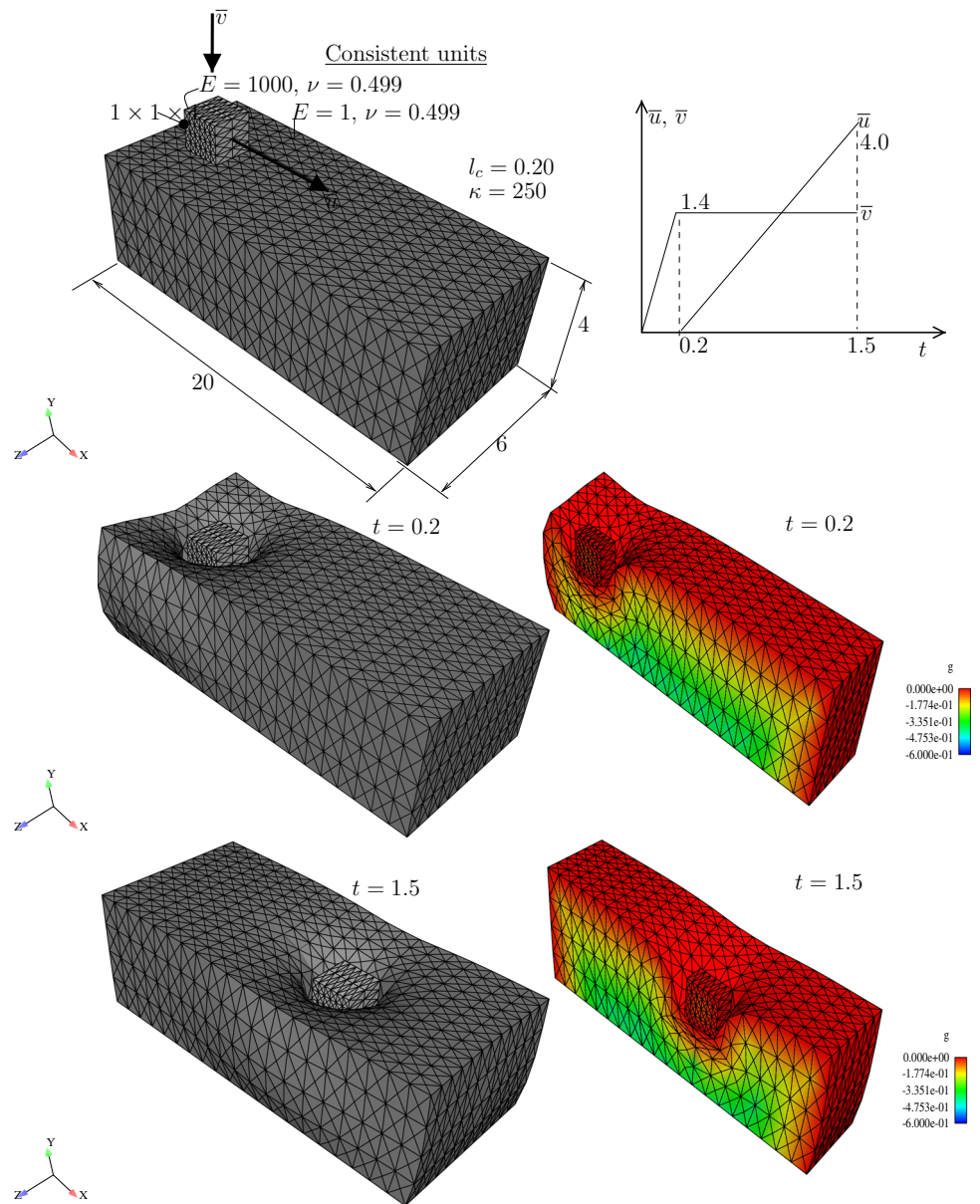
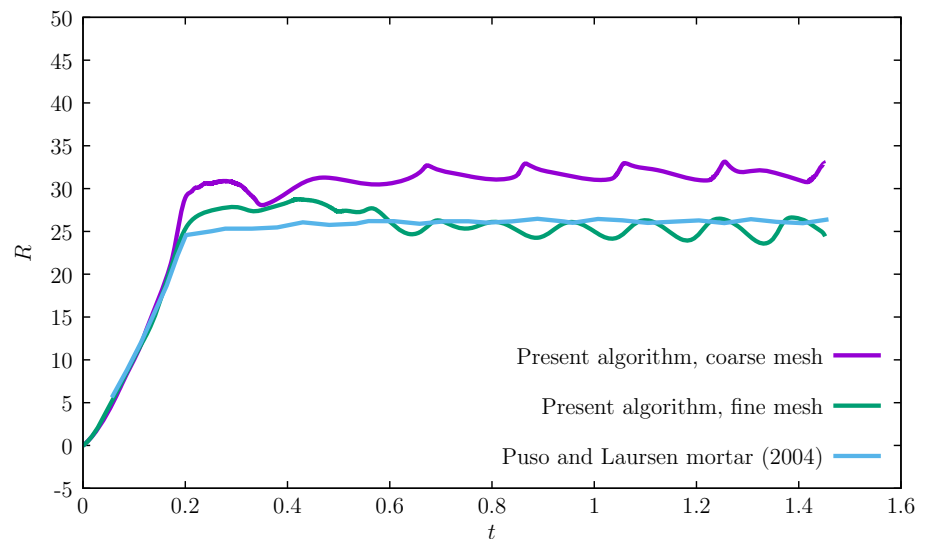


Fig. 15 3D ironing: cube over soft block. Vertical reactions compared with results in Puso and Laursen [19]



Funding Open access funding provided by FCT/IFCCN (b-on).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Sukumar N, Srivastava A (2022) Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Comput Method Appl Mech Eng* 389:114333
- Lai Z, Zhao S, Zhao J, Huang L (2022) Signed distance field framework for unified DEM modeling of granular media with arbitrary particle shapes. *Comput Mech* 70:763–783
- Wriggers P (2002) *Computational Contact Mechanics*. John Wiley and Sons, New York
- Simo J, Laursen T (1992) An augmented Lagrangian treatment of contact problems involving friction. *Comput Struct* 42:97–116
- Nitsche J (1970) Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen in der Mathematik an der Universität Hamburg* 36:9–15
- Kanno Y, Martins JAC, Pinto da Costa A (2006) Three-dimensional quasi-static frictional contact by using second-order cone linear complementarity problem. *Int J Numer Meth Eng* 65:62–83
- Areias P, Rabczuk T, de Melo FJM (2015) Coulomb frictional contact by explicit projection in the cone for finite displacement quasi-static problems. *Comput Mech* 55:57–72
- Laursen TA, Simo JC (1993) Algorithmic symmetrization of coulomb frictional problems using augmented lagrangians. *Comput Method Appl Mech Eng* 108:133–146
- De Saxcé G, Feng Z-Q (1998) The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Mathl Comput Model* 28(4–8):225–245
- Jones RE, Papadopoulos P (2000) A yield-limited Lagrange multiplier formulation for frictional contact. *Int J Numer Meth Eng* 48:1127–1149
- Simo JC, Wriggers P, Taylor RL (1985) A perturbed Lagrangian formulation for the finite element solution of contact problems. *Comput Method Appl Mech Eng* 50:163–180
- Chan SK, Tuba IS (1971) A finite element method for contact problems of solid bodies-Part I. Theory and validation. *Int J Mech Sci* 13:615–625
- Francavilla A, Zienkiewicz OC (1975) A note on numerical computation of elastic contact problems. *Int J Numer Meth Eng* 9:913–924
- Hallquist JO, Goudreau GL, Benson DJ (1985) Sliding interfaces with contact-impact in large-scale Lagrangian computations. *Comput Method Appl Mech Eng* 51:107–137
- Neto DM, Oliveira MC, Menezes LF, Alves JL (2014) Applying Nagata patches to smooth discretized surfaces used in 3D frictional contact problems. *Comput Method Appl Mech Eng* 271:296–320
- Zavarise G, De Lorenzis L (2009) A modified node-to-segment algorithm passing the contact patch test. *Int J Numer Meth Eng* 79:379–416
- Zavarise G, De Lorenzis L (2009) The node-to-segment algorithm for 2D frictionless contact: classical formulation and special cases. *Comput Method Appl Mech Eng* 198:3428–3451
- Puso MA, Laursen TA (2004) A mortar segment-to-segment frictional contact method for large deformations. *Comput Method Appl Mech Eng* 193:4891–4913
- Puso MA, Laursen TA (2004) A mortar segment-to-segment contact method for large deformation solid mechanics. *Comput Method Appl Mech Eng* 193:601–629
- Zavarise G, Wriggers P (1998) A segment-to-segment contact strategy. *Mathl Comput Model* 28(4–8):497–515
- Kim C, Lazarov RD, Pasciak JE, Vassilevski PS (2001) Multiplier spaces for the mortar finite element method in three dimensions. *SIAM J Numer Anal* 39(2):519–538
- Laursen TA, Puso MA, Sanders J (2012) Mortar contact formulations for deformable-deformable contact: past contributions and new extensions for enriched and embedded interface formulations. *Comput Method Appl Mech Eng* 205–208:3–15

23. Farah P, Wall A, Popp A (2018) A mortar finite element approach for point, line and surface contact. *Int J Numer Meth Eng* 114:255–291
24. Wriggers P, Schröder J, Schwarz A (2013) A finite element method for contact using a third medium. *Comput Mech* 52:837–847
25. Kane C, Repetto EA, Ortiz M, Marsden JE (1999) Finite element analysis of nonsmooth contact. *Comput Method Appl Mech Eng* 180:1–26
26. Litewka P (2013) Enhanced multiple-point beam-to-beam frictionless contact finite element. *Comput Mech* 52:1365–1380
27. Neto AG, Pimenta PM, Wriggers P (2016) A master-surface to master-surface formulation for beam to beam contact. Part I: frictionless interaction. *Comput Method Appl Mech Eng* 303:400–429
28. Neto AG, Pimenta PM, Wriggers P (2017) A master-surface to master-surface formulation for beam to beam contact. Part II: Frictional interaction. *Comput Method Appl Mech Eng* 319:146–174
29. Meier C, Wall WA, Popp A (2017) A unified approach for beam-to-beam contact. *Comput Method Appl Mech Eng* 315:972–1010
30. Wolff S, Bucher C (2013) Distance fields on unstructured grids: stable interpolation, assumed gradients, collision detection and gap function. *Comput Method Appl Mech Eng* 259:77–92
31. Liu X, Mao J, Zhao L, Shao L, Li T (2020) The distance potential function-based finite-discrete element method. *Comput Mech* 66:1477–1495
32. Aguirre M, Avril S (2020) An implicit 3D corotational formulation for frictional contact dynamics of beams against rigid surfaces using discrete signed distance fields. *Comput Method Appl Mech Eng* 371:113275
33. Macklin M, Erleben K, Müller M, Chentanez N, Jeschke S, Corse Z (2020) Local optimization for robust signed distance field collision. *Proc ACM Comput Graph Interact Tech* 3(1):1–9
34. Belyaev AG, Fayolle P-A (2015) On variational and PDE-based distance function approximations. *Comput Graph Forum* 34(8):104–118
35. Crane K, Weischedel C, Wardetzky M (2013) Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Trans Graph* 32(5):1
36. Peerlings RHJ, de Borst R, Brekelmans WAM, de Vree JHP (1996) Gradient enhanced damage for quasi-brittle materials. *Int J Numer Method Eng* 39:3391–3403
37. Peerlings RHJ, Brekelmans WAM, de Borst R, Geers MGD (2001) Mathematical and numerical aspects of an elasticity-based local approach to fracture. *Rev Eur Elem Finis* 10:209–226
38. Guler RA, Tari S, Unal G (2014) Screened Poisson hyperfields for shape coding. *SIAM J Imaging Sci* 7(4):2558–2590
39. Belyaev A, Fayolle P-A (2020) An ADMM-based scheme for distance function approximation. *Numer Algorithms* 84:983–996
40. Konyukhov A, Schweizerhof K (2008) On the solvability of closest point projection procedures in contact analysis: analysis and solution strategy for surfaces of arbitrary geometry. *Comput Method Appl Mech Eng* 197(33–40):3045–3056
41. Russo G, Smereka P (2000) A remark on computing distance functions. *J Comput Phys* 163:51–67
42. Courant R (1943) Variational methods for the solution of problems of equilibrium and vibrations. *Bull Am Math Soc* 49:1–23
43. Beltrami EJ (1969) A constructive proof of the kuhn-tucker multiplier rule. *J Math Anal Appl* 26:297–306
44. Varadhan SRS (1967) On the behavior of the fundamental solution of the heat equation with variable coefficients. *Commun Pure Appl Math* 20:431–455
45. Areias P (2022) 3D contact files. <https://github.com/PedroAreiasIST/contact3d>
46. Korelc J (2002) Multi-language and multi-environment generation of nonlinear finite element codes. *Eng Comput* 18(4):312–327
47. Wolfram Research Inc. (2007) Mathematica
48. Areias P. Simplas. <http://www.simplassoftware.com>. Portuguese Software Association (ASSOFT) Registry number 2281/D/17
49. Gustavson FG (1978) Two fast algorithms for sparse matrices: multiplication and permuted transposition. *Trans Math Soft-ACM* 4(3):250–269
50. Yang B, Laursen TA, Meng X (2005) Two dimensional mortar contact methods for large deformation frictional sliding. *Int J Numer Meth Eng* 62:1183–1225
51. Hartmann S, Oliver J, Weyler R, Cante JC, Hernández JA (2009) A contact domain method for large deformation frictional contact problems. Part 2: numerical aspects. *Comput Method Appl Mech Eng* 198:2607–2631
52. Bonet J, Wood RD (2008) *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, Second edition

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.