



Impact on highly compressible media in explicit dynamics using the X-FEM

Céline Dubois, Steven Le Corre, Malek Zarroug, Patrick Rozycki, Nicolas Moes

► To cite this version:

Céline Dubois, Steven Le Corre, Malek Zarroug, Patrick Rozycki, Nicolas Moes. Impact on highly compressible media in explicit dynamics using the X-FEM. *Computational Mechanics*, 2010, 46 (2), pp.329-348. 10.1007/s00466-010-0497-x . hal-01006954

HAL Id: hal-01006954

<https://hal.science/hal-01006954>

Submitted on 12 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Impact on highly compressible media in explicit dynamics using the X-FEM

Céline Dubois · Steven Le Corre · Malek Zarroug ·
Patrick Rozycki · Nicolas Moës

Abstract Finite element simulations of impact problems on highly compressible media often lead to poor accuracy due to mesh distortion. In explicit dynamics, poorly shaped elements also reduce the stable time step. In order to have satisfactory results and an acceptable computational time, the structure has to be remeshed regularly. A remeshing process can be a burdensome task, especially for 3D problems with complex geometries. In explicit methods, remeshing can also be time consuming compared to the time required for the computation. In this article, we propose to use the extended finite element method (X-FEM) to simplify the remeshing work. This simplification relies on the fact that the X-FEM allows to remesh with meshes that do not match the shape of the deformed structure. A unique simple structured mesh can be used whenever remeshing is needed. A specific algorithm is designed in order to ensure data transfer between successive meshes in the X-FEM context. Several examples demonstrate the efficiency of the proposed method. The final part of the article is dedicated to the treatment of impact problems. It is shown that the use of the penalty method with X-FEM in explicit dynamics leads to a decrease of the stable time step. We propose a specific mass scaling strategy to overcome this issue.

Keywords X-FEM · Explicit dynamics · Nonlinear elasticity · Remeshing technique · Numerical methods

C. Dubois · S. Le Corre · P. Rozycki · N. Moës Institut de recherche en Génie Civil et Mécanique, CNRS, Ecole Centrale de Nantes, 1 rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France
e-mail: Steven.le-corre@ec-nantes.fr

M. Zarroug
PSA-Peugeot Citroën DRIA/SARA/STEO/MAST,
Route de Gisy, 78943 Vélizy Villacoublay, France

1 Introduction

Many problems involving very large deformation cannot be efficiently computed by the same Lagrangian mesh throughout the simulation. As the material undergoes severe strains, elements aspect ratios deteriorate [4]. This leads to a significant loss of accuracy in the approximation. The mesh is no longer able to represent a good approximation: it fails to satisfy the completeness condition. In worst cases, several elements may invert, the Jacobian determinants become negative, which may lead to the loss of positive definiteness of the stiffness matrix and thus important local errors that may abort the calculation. Another drawback of the distortion in explicit dynamics is that the stable time step is considerably reduced [2].

A classical solution to avoid these problems is to use arbitrary Lagrangian Eulerian methods (ALE). Contrary to purely Lagrangian methods, the mesh motion is independent of the material motion and can be chosen in a way to eliminate distortion. These methods seem very attractive since they take advantage of both Lagrangian and Eulerian descriptions. Nevertheless their practical use is not straightforward. The choice of a judicious mesh motion can be a quite difficult task, particularly due to the fact that boundary nodes or material interfaces nodes in the case of multi material problems can only move in tangential direction [2]. The extended finite element method (X-FEM) is known to be an efficient solution to deal with arbitrary discontinuities like cracks, voids and inhomogeneities [9, 17, 18]. Within this method, the geometry of the discontinuity is independent of the mesh and is represented by a level-set function [20]. Several authors proposed to couple X-FEM with ALE to solve bi-material problems [11, 23]. The results are satisfactory. In ALE methods, in addition to the mesh motion two other essential points require special care. The first one is that the topology of

the mesh does not change during the simulation, the number of nodes remains the same so it has to be chosen carefully from the beginning of the calculation. The second one is that ALE equations involve convective terms to take into account transport of material through the mesh. The computation of these terms need the use of advection algorithms which can be computationally expensive.

Another class of methods, referred by Benson [4] as the “rezoning method”, can be seen as a good alternative to ALE methods. The basic idea is to run a calculation with a Lagrangian mesh until it becomes excessively distorted and then generate a new undistorted mesh. The solution fields are mapped from the old distorted mesh to this new mesh, which is used as a Lagrangian mesh to continue the calculation. We can notice that the remeshing work has to be done on the deformed configuration in order to ensure a good quality for the new mesh [12]. The new mesh can be built by adapting the old one. Some places can be refined by dividing elements but then some transition elements have to be introduced for the mesh to be conform. This process can really become heavy in three dimensions [8] so that it is often decided to replace the old mesh by a completely new one. This last method is also a better way to get a new mesh with good quality shape.

However, the difficulty arises from the fact that the new mesh has to approximate the boundary of the deformed body as good as possible. Within this context, X-FEM can be seen as a relevant solution since the mesh does not have to conform to the external interfaces. The new mesh can simply be a regular and structured mesh independent of the shape of the deformed body. The only condition to satisfy is that the deformed body lies inside the structured new mesh. This idea has been recently used within the Eulerian Finite Cover Method (FCM) to treat quasi-static equilibrium problems with large deformation [22]. The authors propose to remesh with a regular and structured mesh after each Newton–Raphson solving step. The regular mesh is the same during the entire simulation, that is why it is called the “Eulerian mesh”.

In this work, we use the possibility offered by X-FEM to remesh on structured meshes in the context of dynamics simulations. The remeshing process is not done at each time step but only when the current mesh is no longer able to handle the deformation. The new structured mesh does not have to remain the same during the whole simulation, it can be changed for a finer one for example. We present a specific algorithm to map the quantities from the old mesh to the new one. The main point of our work is to take advantage of the regularity of the new mesh to save computational time. Another great advantage of this method is that the remeshing operation can be automatized. Contrary to most rezoning methods available now, it does not require any user intervention.

2 X-FEM in non-linear explicit dynamics

2.1 Governing equations

Let us consider a body in its initial configuration Ω_0 . The vector \mathbf{X} denotes the position of a material point in this reference configuration. The motion of the body is described by $\mathbf{x} = \boldsymbol{\phi}(\mathbf{X}, t)$ where \mathbf{x} is the position of the material point \mathbf{X} at time t . The displacement of a material point can be written as $\mathbf{u}(\mathbf{X}, t) = \mathbf{x} - \mathbf{X} = \boldsymbol{\phi}(\mathbf{X}, t) - \mathbf{X}$. The deformation gradient is defined by

$$\mathbf{F} = \nabla_0 \boldsymbol{\phi}(\mathbf{X}, t) = \partial \mathbf{x} / \partial \mathbf{X}, \quad (1)$$

where ∇_0 indicates the gradient operator with respect to material coordinate \mathbf{X} . The Jacobian of the transformation is $J = \det \mathbf{F}$. The body is subjected to a force per unit mass \mathbf{b} . The strong form of the balance of momentum can be expressed as follows:

$$\nabla_0 \cdot \mathbf{P} + \rho_0 \mathbf{b} = \rho_0 \ddot{\mathbf{u}} \quad \text{in } \Omega_0, \quad (2)$$

where \mathbf{P} is the nominal stress, ρ_0 the density in the reference configuration and $\ddot{\mathbf{u}}$ the acceleration of a material point. The displacement is prescribed as $\mathbf{u} = \bar{\mathbf{u}}$ on $\Gamma_{0,u}$ and the nominal traction vector as $\mathbf{n}_0 \cdot \mathbf{P} = \bar{\mathbf{t}}_0$ on $\Gamma_{0,t}$. Initial conditions are also imposed: $\mathbf{u}(\mathbf{X}, 0) = \mathbf{u}_0(\mathbf{X})$ and $\dot{\mathbf{u}}(\mathbf{X}, 0) = \dot{\mathbf{u}}_0(\mathbf{X})$. The weak form of the momentum equations including natural (Neumann type) boundary conditions is therefore given by:

$$\begin{aligned} \int_{\Omega_0} \delta \mathbf{u} \cdot \rho_0 \ddot{\mathbf{u}} d\Omega_0 + \int_{\Omega_0} \delta \mathbf{F}^T : \mathbf{P} d\Omega_0 \\ = \int_{\Omega_0} \rho_0 \delta \mathbf{u} \cdot \mathbf{b} d\Omega_0 + \int_{\Gamma_{0,t}} \delta \mathbf{u} \cdot \bar{\mathbf{t}}_0 d\Gamma_0, \end{aligned} \quad (3)$$

where test functions $\delta \mathbf{u}$ belong to a space of C^0 functions which vanishes on the boundary of imposed displacements $\Gamma_{0,u}^0$. The first term in the left-hand side of the above equation represents the virtual inertial work, the second term is the virtual internal work. The whole right-hand side term is the virtual external work.

The mechanical behavior of the material is supposed to follow an hyperelastic law. As this kind of constitutive model is path-independent, the second Piola–Kirchhoff stress \mathbf{S} can be expressed as the derivative of a stored energy function ψ :

$$\mathbf{S} = 2\partial\psi/\partial\mathbf{C} \quad \text{where} \quad \mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}, \quad (4)$$

is the right Cauchy–green strain tensor. For numerical examples, we use the Neo-Hookean constitutive model in which the stored energy is defined by

$$\psi(\mathbf{C}) = \lambda(\ln J)^2/2 - \mu \ln J + \mu(\text{trace}(\mathbf{C}) - 3)/2, \quad (5)$$

where λ and μ are the Lamé constants. The second Piola–Kirchhoff stress is then given by:

$$\mathbf{S} = \mu(\mathbf{I} - \mathbf{C}^{-1}) + \lambda(\ln J)\mathbf{C}^{-1}, \quad (6)$$

and \mathbf{P} can be obtained by $\mathbf{P} = \mathbf{S} \cdot \mathbf{F}^T$.

2.2 Spatial discretization with non-matching meshes on interface

The X-FEM relies on the idea of enriching the finite-element approximation by additional functions through the concept of partition of unity [14]. The enrichment functions can be used to model discontinuities inside the elements like cracks, voids or inhomogeneities and also to improve accuracy in problems where the behavior of the solution field is known analytically, like stress around crack tip for example.

We consider a domain Ω^0 divided into finite elements. Let $N = \{n_1, n_2, \dots, n_m\}$ be a set of m nodes in the mesh and let $\Omega_g^0 \subset \Omega^0$ be the region where some enrichment is needed. The general form of the X-FEM displacement approximation can be expressed as follow:

$$\mathbf{u}_h(\mathbf{X}) = \sum_{n_I \in N} \phi_I(\mathbf{X}) \mathbf{u}_I + \sum_{n_J \in N_g} \phi_J(\mathbf{X}) \psi(\mathbf{X}) \mathbf{a}_J, \quad (7)$$

where the nodal set N_g is defined as $N_g = \{n_J/n_J \in N, \omega_J \cap \Omega_g^0 \neq \emptyset\}$ where ω_J is the support of the nodal shape function $\phi_J(\mathbf{X})$, i.e. the union of all elements containing n_J as a node. The additional degrees of freedom are denoted as \mathbf{a}_J , they usually do not have a physical significance but they allow one to deal with discontinuities. The function $\psi(\mathbf{X})$ is called the enrichment function.

Modeling holes is a somewhat special since there is no need for additional degrees of freedom. The classical shape functions for the displacement are simply multiplied by an enrichment function $V(\mathbf{X})$ whose value is 1 if the node lies outside the void and 0 if it is inside [21]. The displacement field can be written as

$$\mathbf{u}_h(\mathbf{X}) = \sum_{n_I \in N} \phi_I(\mathbf{X}) V(\mathbf{X}) \mathbf{u}_I = \sum_{n_I \in N} \hat{\phi}_I(\mathbf{X}) \mathbf{u}_I. \quad (8)$$

Nodes whose support is entirely inside the void are removed from the calculation (see Fig. 1). The interface between the material and the void is implicitly described as the zero level of a level-set function $\varphi(\mathbf{X})$. This function gives the signed distance of any point \mathbf{X} to the interface: $\varphi(\mathbf{X})$ is positive if \mathbf{X} lies in the void, negative in the material. As the other fields, φ is interpolated on the mesh by:

$$\varphi(\mathbf{X}) = \sum_{n_I \in N} \phi_I(\mathbf{X}) \varphi_I \quad (9)$$

We can notice for example that for linear tetrahedral elements, the level-set function is linear within each element.

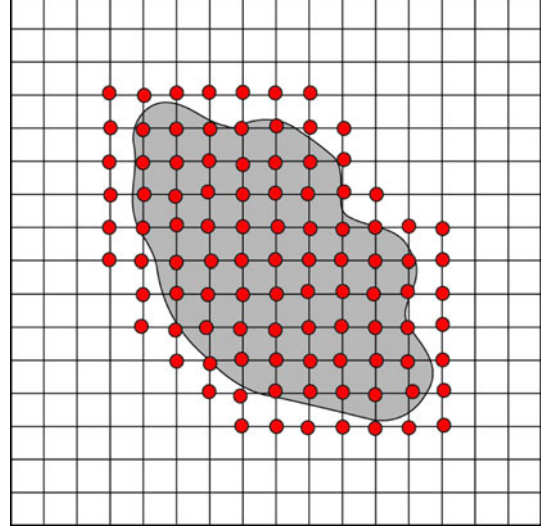


Fig. 1 Nodes of the grid used in the X-FEM approach

Substituting the trial and test functions from their approximation (7) in the weak form (3) and using the arbitrariness of test functions, the following semi-discrete system of equations is obtained:

$$\mathbf{M} \cdot \ddot{\mathbf{u}} + \mathbf{f}^{\text{int}} = \mathbf{f}^{\text{ext}} \quad (10)$$

Mass matrix terms M_{IJ} are given by:

$$M_{IJ} = \int_{\Omega_0} \rho \hat{\phi}_I \hat{\phi}_J d\Omega_0 \quad (11)$$

The internal nodal force vector $\mathbf{f}_I^{\text{int}}$ is expressed as:

$$\mathbf{f}_I^{\text{int}} = \int_{\Omega_0} \frac{\partial \hat{\phi}_I}{\partial \mathbf{X}} \cdot \mathbf{P} d\Omega_0. \quad (12)$$

The external nodal force vector $\mathbf{f}_I^{\text{ext}}$ is expressed as:

$$\mathbf{f}_I^{\text{ext}} = \int_{\Omega_0} \hat{\phi}_I \rho_0 \mathbf{b} d\Omega_0 + \int_{\Gamma_0} \hat{\phi}_I \mathbf{t}^0 d\Gamma_0 \quad (13)$$

In practice, instead of using $\hat{\phi}_I(\mathbf{X})$, elements that intersect the voids are cut along the iso-zero of the level-set and then partitioned in sub-elements. The weak form is only integrated in the sub-elements and elements that belong to the matter part [21] (see Fig. 2).

2.3 Time discretization

The numerical time integration is performed through the well known central differences explicit scheme. The mass matrix is usually diagonalized by a lumping technique so that the discrete momentum equations do not require the solution of any system of equations. The method is stable if the time step

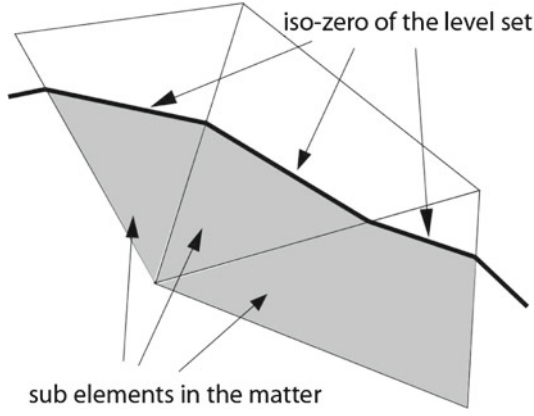


Fig. 2 For elements crossed by the *iso-zero of the level-set*, integration is done on sub-elements inside matter

remains lower than a critical value Δt_{crit} . In a linear finite element analysis without damping, one can demonstrate that

$$\Delta t_{\text{crit}} = \frac{2}{\omega_{\text{max}}} \leq \min \left(\frac{l_e}{c_e} \right), \quad (14)$$

where ω_{max} is the maximum frequency of the system, l_e is a characteristic length of element e in deformed configuration and c_e is the current wavespeed in element e . For non-linear problems, no stability theorem can be found. The critical time step is calculated for the linearized system and then a reduction factor is applied.

Rozyczki et al. [19] showed that the critical time step in the X-FEM could be the same as the one in classical FEM provided the use of a specific lumping technique. For a complete review of the mass lumping strategies for X-FEM, one can also refer to Menouillard et al. [15].

We use the central difference algorithm as detailed in Belytschko et al. [2] where the time step can be variable. As pointed out by the authors, this is more suitable for finite strains since the stable time step changes during the simulation as the mesh deforms and as the deformation of the material makes the wave speed change. The time increment between the iteration n and $n + 1$ is denoted by $\Delta t = t^{n+1} - t^n$. Each time increment can be decomposed as follows:

1. Update of nodal displacements:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \dot{\mathbf{u}}^n + \frac{\Delta t^2}{2} \ddot{\mathbf{u}}^n, \quad (15)$$

2. First partial update of nodal velocities:

$$\dot{\mathbf{u}}^{n+\frac{1}{2}} = \dot{\mathbf{u}}^n + \frac{\Delta t}{2} \ddot{\mathbf{u}}^n, \quad (16)$$

3. Computation of nodal accelerations:

$$\ddot{\mathbf{u}}^{n+1} = \mathbf{M}^{-1} \cdot (-\mathbf{f}^{\text{int},n+1} + \mathbf{f}^{\text{ext},n+1}), \quad (17)$$

4. Second partial update of nodal velocities:

$$\dot{\mathbf{u}}^{n+1} = \dot{\mathbf{u}}^{n+\frac{1}{2}} + \frac{\Delta t}{2} \ddot{\mathbf{u}}^{n+1}. \quad (18)$$

3 Remeshing on a regular grid

The main issue in remeshing is to handle data transfer between two meshes. We have to define a way to transmit the solution fields. Two kinds of data are distinguished in the literature: variables stored at nodes and variables stored at integration points. For nodal variables, collocation methods are widely used. They can be decomposed in three steps:

1. Identification: for each new node, find in which old element it is located;
2. Inversion: find the local coordinate of the new node in the old element;
3. Interpolation: use the interpolation on the old mesh to find the respective nodal value of the new node.

Other methods based on averaging procedure can be used to transfer nodal fields between incompatible meshes. For example, the least-square method allows to minimize the square of the difference between two interpolations of the same field. This method is also known as the L_2 projection and is widely used in dynamics to transfer initial conditions on a mesh [2]. Another method based on Lagrange multipliers and called the Mortar method is often used to transfer fields between interfaces in domain decomposition problems or contact problems. In the case of incompatible meshes, this method is very close to the least-square method. Opposite to the collocation method, the least-square and the Mortar methods do not imply any loss of information when transferring from a fine mesh to a coarse one. Moreover they allow to keep the duality between two fields on the new mesh. However, these averaging methods require the inversion of a global mass matrix and the construction of integral cross products of shape functions defined on different meshes. Despite their advantages in terms of accuracy, these methods can make the data transfer very time consuming in an explicit context. Furthermore, in our case, the boundaries of the new and the old mesh will not match exactly so that such projection methods do not seem easily applicable. That is why we choose to use the collocation method.

For internal variables, usually known at Gauss points, the task is much more difficult because they have to stay compatible with other variables after the projection. This has to

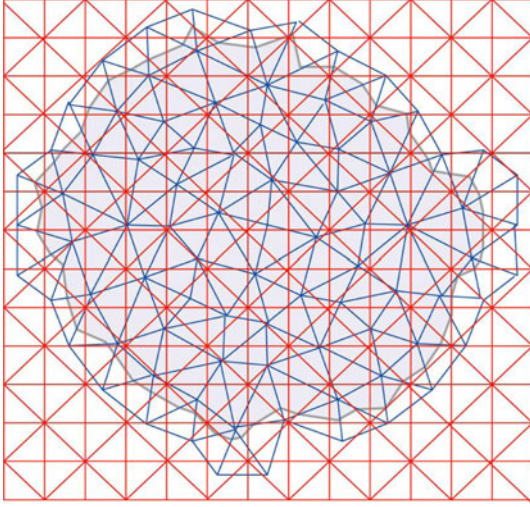


Fig. 3 X-FEM allows to remesh the deformed configuration with a non-conforming regular mesh

be taken into account in elasto-plastic models for example. Several authors [5, 13] proposed methods to ensure the equilibrium during the data transfer. We will not further detail these methods since our study focuses on hyperelastic bodies.

For nodal fields, a similar problem can arise. In explicit dynamics if we transfer all kinematics nodal fields: displacements, velocities and accelerations, we cannot ensure that the conservation of linear momentum (3) will be satisfied on the new mesh. To avoid these problems, we choose to transfer only displacements and velocities. The acceleration field is then deduced from the above by means of the linear momentum equation (17). The procedure can therefore be summarized as:

1. Projection of \mathbf{u}^{n+1} () and $\dot{\mathbf{u}}^{n+\frac{1}{2}}$ () from mesh 1 to mesh 2;
2. Computation of $\ddot{\mathbf{u}}^{n+1}$ on mesh 2 by (17).

3.1 Scan conversion technique

The use of the X-FEM allows to remesh the deformed configuration with regular and non-conforming grids (cf. Fig. 3). We can take advantage of the regularity of the new mesh to deal efficiently with the identification step required for the collocation.

Scan conversion (or rasterization) is a standard technique in computer graphics for displaying filled polygons on a grid of pixels [6]. The goal of the method is to determine efficiently pixels which lie inside polygons. Let us consider a convex polygon on a regular 2D grid in the plane (x, y) . We begin by classifying the edges of the polygon according to the highest y coordinate of their extremities. For each grid row that intersects the polygon, we determine which edges

are intersected and find the left and right intersection points x_L and x_R (Fig. 4). We can then conclude that each grid point between x_L and x_R is inside the polygon. As pointed in [6] the computational complexity of the scan conversion algorithm is $\mathcal{O}(e + r + n)$ where e the number of edges of the polygon, r the number of rows that intersects the polygon and n the number of grid points inside. If the sides of the polygon are not smaller than the grid spacing, r and e become negligible compared to n and the complexity reduces to $\mathcal{O}(n)$.

```
function ScanPolygonConvex(Polygone P)
  sort the edges of P by their maximum y coordinate;
  for each scanline y (from top to bottom) do
    select the two edges that intersect the scanline;
    compute the intersecting points  $x_L$  et  $x_R$ ;
    add all nodes with a x coordinate between  $x_L$  and  $x_R$  to the
    set of nodes scanned by P;
  end
```

Algorithm 1: Scan conversion algorithm for a two-dimensional convex polygon.

For the 3D Case, the former 2D scan conversion technique is adapted and limited to the case of tetrahedra in a uniform regular grid. We first define the intersection of the considered tetrahedron with a horizontal plane at height z , which therefore forms a 2D polygon (triangle or quadrangle) in the considered plane. This polygon is then scanned by the 2D technique described in the previous section.

Figure 5 represents the slicing of a tetrahedron to form polygons. The complexity of the algorithm is also linear in the number of grid points inside the polyhedron.

```
function ScanTetrahedron(Tetrahedron T)
  sort the faces of T by their maximum z coordinate;
  for each scanplan z (from top to bottom) do
    select the faces that intersect the scanplan;
    build the polygon P resulting from the intersection
    between the scanplan and the selected faces;
    apply the 2D function ScanPolygonConvex to P;
    add all nodes scanned by P to the set of nodes
    scanned by T;
  end
```

Algorithm 2: Scan conversion algorithm for a tetrahedron.

In our problems, we scan every deformed old elements and sub elements to find which nodes of the new regular mesh are inside. The complexity of the algorithm is linear in the number of nodes of the new mesh recovered by the old mesh.

3.2 Building of the level-set on the new mesh

In order to continue the computation after a remeshing step, we also need to locate the position of the level-set on the new mesh.

Fig. 4 Principle of the scan conversion of a polygon in 2D. **a** Case where edge1 and edge2 are activated. **b** Case where edge2 and edge3 are activated

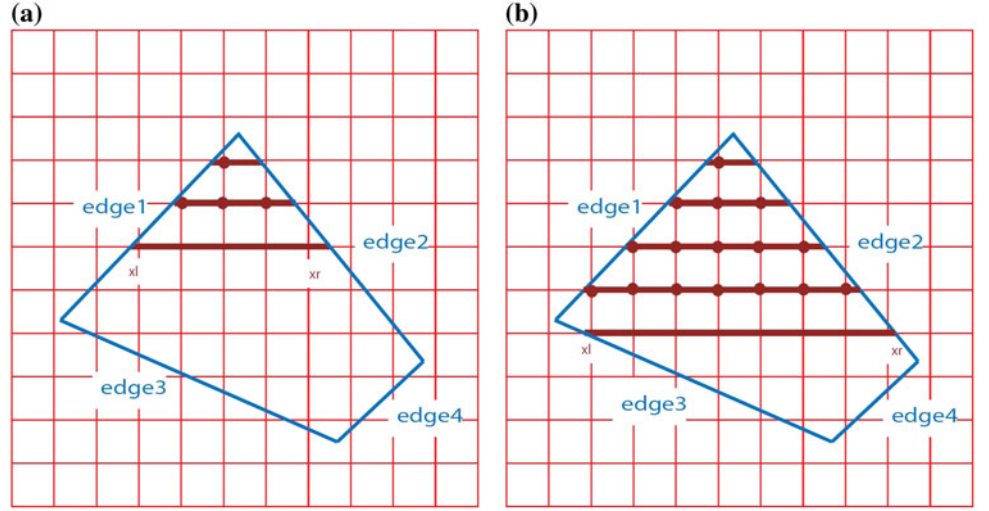
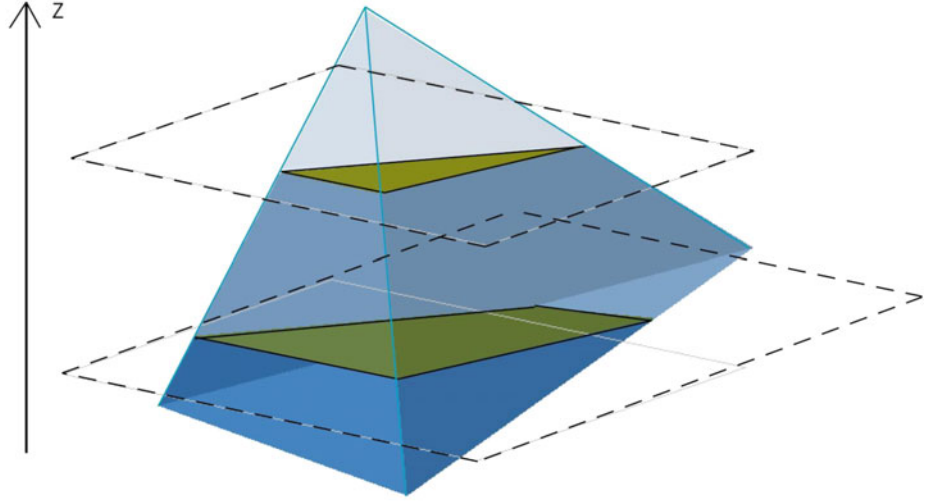


Fig. 5 Principle of the scan conversion of a tetrahedron



Opposite to the displacement, the level-set is not a Lagrangian field but an Eulerian one. The level-set is known for a given position and not for a given material point. When the mesh moves, the level-set stored at a node still represents the distance between this node and the interface in the initial configuration but not in the deformed one. When the initial mesh is deformed, the level-set field is then no longer a distance function. If this field is transferred directly by collocation, it will lead to a wrong positioning of the interface for the new mesh. This can be shown by a simple example in one dimension in Fig. 6.

One could imagine to reinitialize the level-set to a distance function on the first mesh in its deformed configuration before doing the transfer to the second mesh. The reinitialisation or redistancing of a level-set is usually done by finding the level-set ϕ satisfying the above transport equation:

$$\frac{\partial \phi}{\partial t} + \text{sign}(\phi_o)(|\nabla \phi| - 1) = 0 \quad (19)$$

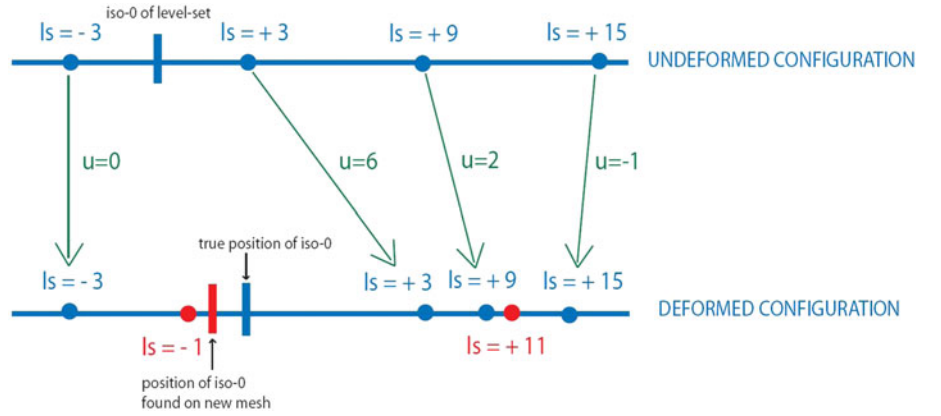
where $\phi(\mathbf{x}, t = 0) = \phi_o$ is the initial value of the level-set. In order to alleviate oscillations phenomenon, this equation is solved by means of specific algorithm (ENO, WENO) which can be computationally expensive. More efficient scheme like the Fast Marching Methods can also be used to accelerate the process, but anyway it will require additional iterations which are not suitable for explicit dynamics.

Another way to obtain a distance function is to discretize the interface Γ and to simply evaluate the distance using its definition:

$$D(\mathbf{x}) = \min_{\mathbf{y} \in \Gamma} |\mathbf{x} - \mathbf{y}| \text{sign}(\phi_o) \quad (20)$$

This method can be directly applied on the new mesh; there is no projection to do. That is the reason why we choose this solution. Moreover, we propose here a more efficient strategy using the information obtained during the scan to rebuild the level-set on the new mesh.

Fig. 6 The level-set field is no longer a distance function in the deformed configuration. If no reinitialisation step is done, collocation leads to a wrong positioning of the iso-zero



First of all, we know if a node has been scanned by a element (or sub-element) in the matter or in the void. We can conclude on the sign of the level-set for this node: negative in the matter and positive in the void. We then have to evaluate the distance. During the integration, the initial mesh has been cut along the iso-zero of the level-set, so we have access to a mesh of the deformed contour. This mesh is called the boundary mesh and is made of triangular facets. For each node of the new mesh, we can loop over boundary facets, calculate the distance to each facet and then select the minimum distance and the closest facet. The cost of this simple algorithm is $\mathcal{O}(N * F)$ where N is the number of new nodes and F the number of facets of the boundary mesh. In 3D, this can be expensive and useless because we only need to know the level-set near its iso-zero. We propose here a fast algorithm which takes advantage of the informations obtained during the scan conversion phase.

The first step of this algorithm is to select a group of nodes on the new mesh where the level-set function has to be known. This group will be called the “new narrow band nodes”. A simple way to define these nodes is to consider that they are the nodes scanned by old elements close to the level-set. The first task is therefore to select a group of elements close to the level-set in the old mesh which will be called the “old narrow band elements”, as illustrated in Fig. 7. This set is made of old elements cut by the iso-zero of the level-set function that represents the boundary of the structure. In order to be sure to have a sufficiently large band, we also include the neighbors of the cut old elements which lie inside the matter.

Then the “new narrow band nodes” can be constructed by taking the nodes scanned in this band of old elements (Fig. 8a). Because of robustness issue that will be described later, the neighboring nodes of these nodes are also included to the new narrow band. This is illustrated on Fig. 8b.

Doing so, we have limited the number of nodes where the level-set will be evaluated. To further improve this procedure, one can also reduce the number of facets to be taken into account for each node. Each facet has been created by the cutting operation of an element of the old mesh so a link

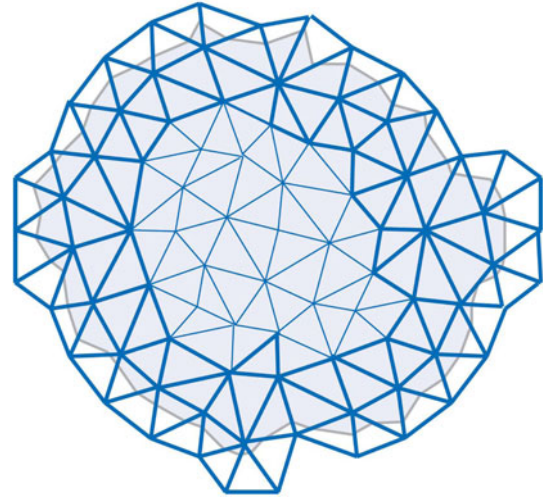


Fig. 7 “Old narrow band” elements

can be established between the facets and the elements of the “old narrow band elements”. A facet is set to be potentially close to the element that it crosses and to the neighbor of this element (see Fig. 9). Thus, an element of the “old narrow band elements” is associated to a group of eventually close facets. Defining those local groups, any node scanned in the narrow band can be related to a small amount of facets which makes the cost of the calculation of the level-set proportional to the number of nodes in the narrow band.

3.3 Algorithm robustness

An important issue is to ensure the reliability of the proposed algorithm. First of all, we can easily check that the algorithm proposed allows to obtain exactly the same results when choosing a second mesh identical to the first one.

A second important point is to ensure that no node is forgotten during scan operation. Such an error can come from the fact that our algorithm employs floating point values. Rounding operations during the scan conversion can lead to

Fig. 8 Construction of the “new narrow band” nodes.
a Phase 1. b Phase 2

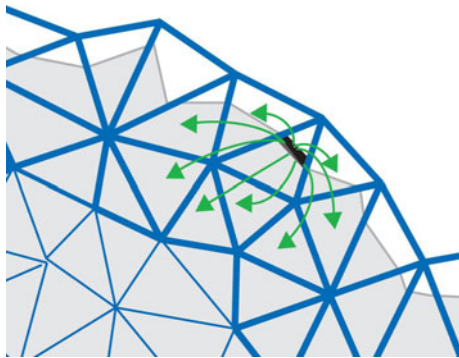
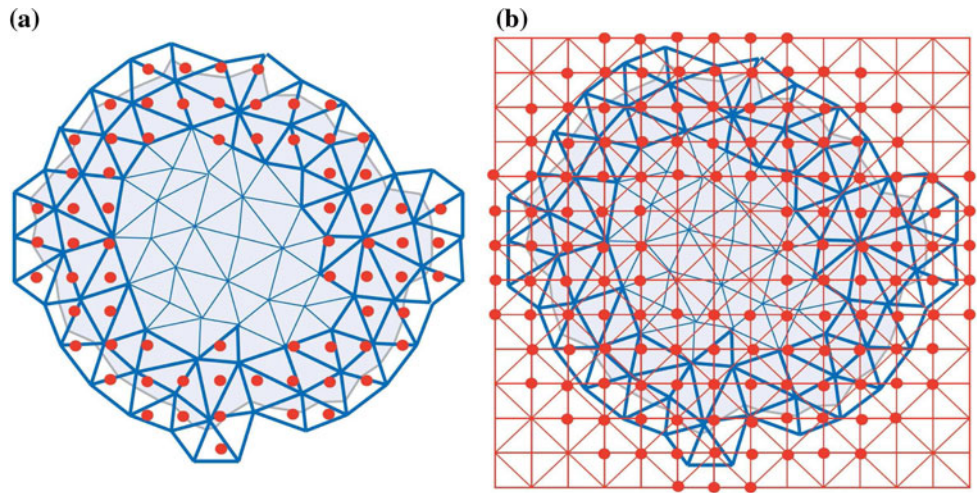


Fig. 9 A facet is potentially close to its “parents” elements and to his neighbors

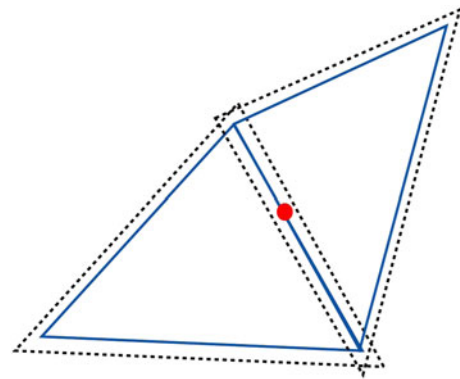


Fig. 10 Introduction of a tolerance in order to ensure that no nodes are forgotten

miss a node very close to the edge of an element. We have to introduce a tolerance ϵ when we test whether a node lies into an old element. In practice, it can be seen as a small increase in the element size (Fig. 10). Thus, for example when a new node is exactly on one edge common to two old elements, it is scanned by both. In the future, a good improvement would be to design an algorithm using only integers as it is usually done in computer graphics.

In the distance calculation, we have to take care to get all facets possibly close to a node. This is the reason why we take the neighbors of the old elements to construct the narrow band of a facet. This can avoid the critical case illustrated in Fig. 11, when the closest facet to a node does not cross the element by which it was scanned.

Another critical situation that could happen in the X-FEM context is that new nodes having to be considered on the new mesh can be unrecovered by the old mesh. If they are not scanned, no displacement or velocity fields can be transferred to them (see Fig. 12). This case is avoided by the fact that we include the neighbors of the nodes scanned in the “old narrow band elements” to construct the “new narrow band nodes”. If new nodes are not scanned, we use their closest

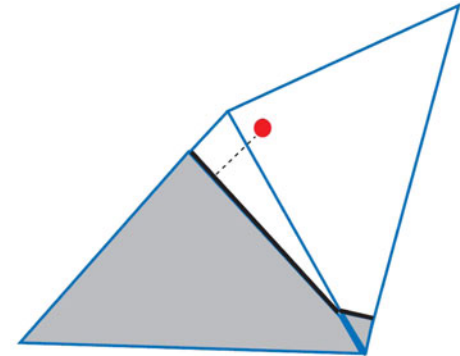


Fig. 11 The closest facet to a node does not cross the element by which it was scanned

facet to find which is their closest deformed element and then we extrapolate the solution fields (see Fig. 13).

3.4 Numerical examples

In this section, we study several examples in order to test the validity of the algorithm described above. The goal of the

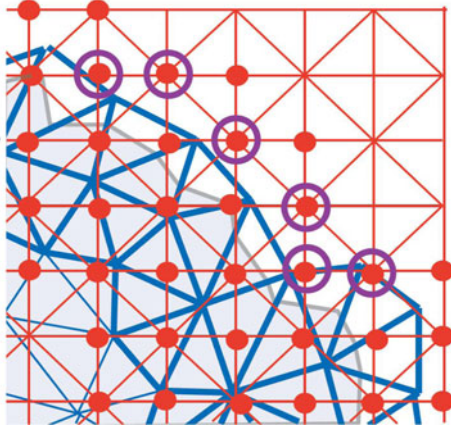


Fig. 12 Some non-scanned nodes have to be taken into account for the computation on the new mesh

two first examples is to check two important features of the algorithm: the level-set evaluation on the new mesh and the accuracy of data transfer. The third example illustrates the main advantages of the method: the reduction of the distortion rate and the increase of the critical time step.

Fig. 13 Management of data transfer for unscanned nodes:
(a) finding the closest,
(b) extrapolation of the fields values

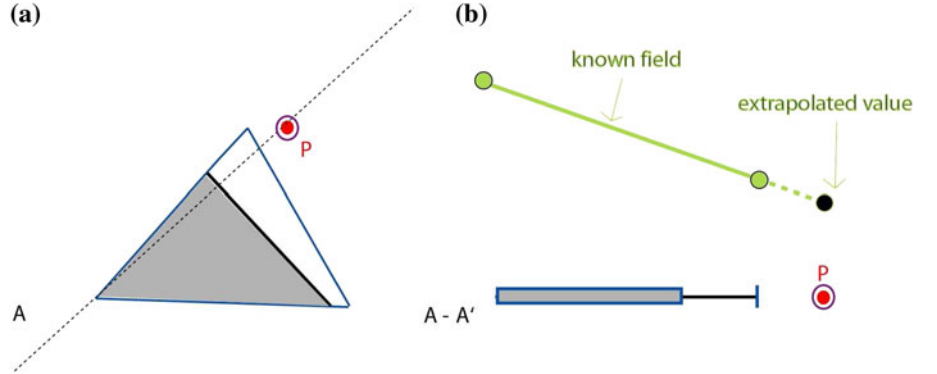
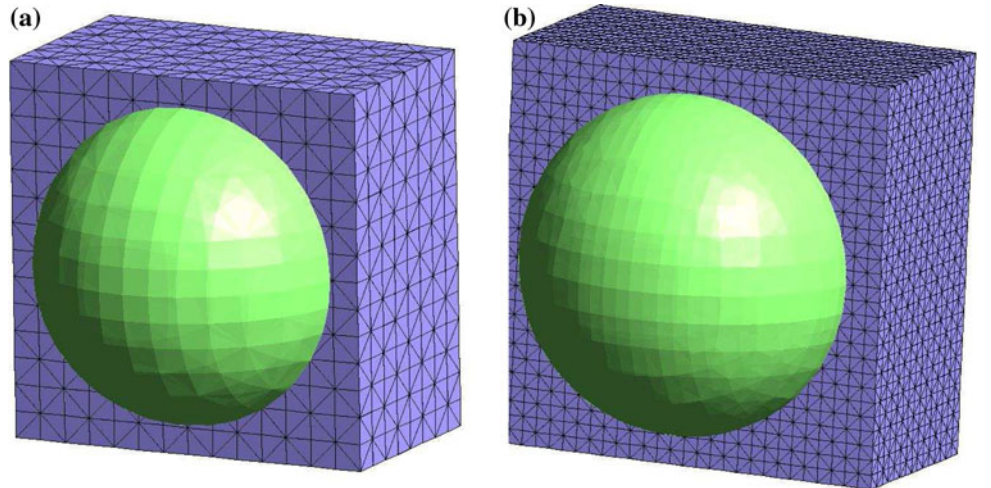


Fig. 14 Two meshes are used during the computation



3.4.1 Example 1: free fall of a sphere

In order to test the efficiency of our algorithm, we begin with a very basic problem: the free fall of a sphere. The value of the sphere radius is 2.4 m. The problem is first discretized with a regular tetrahedral mesh whose characteristic length is 0.4 m. This mesh was built from a hexahedral one and each hexahedron was cut into six tetrahedra. Two different cutting directions were used for two consecutive hexahedra in order to obtain an isotropic mesh. During the explicit computation, the initial mesh is replaced by a new one built in a similar way but with a characteristic length of 0.2 m (see Fig. 14).

The sphere undergoes a rigid body motion. Displacement and velocity fields are uniform. Data transfer is therefore straightforward. We see in Fig. 15 that those fields perfectly match the analytical solution before and after remeshing.

The first step of the level-set computation is to select the “old narrow band elements” on the initial mesh. This band is made of the elements crossed by the level-set iso-zero and their neighbors (see Fig. 16). This group of elements cover nodes on the new mesh. These nodes and their neighbors make up the “new narrow band nodes” (see Fig. 17). The new iso-zero of the level-set is computed only on this

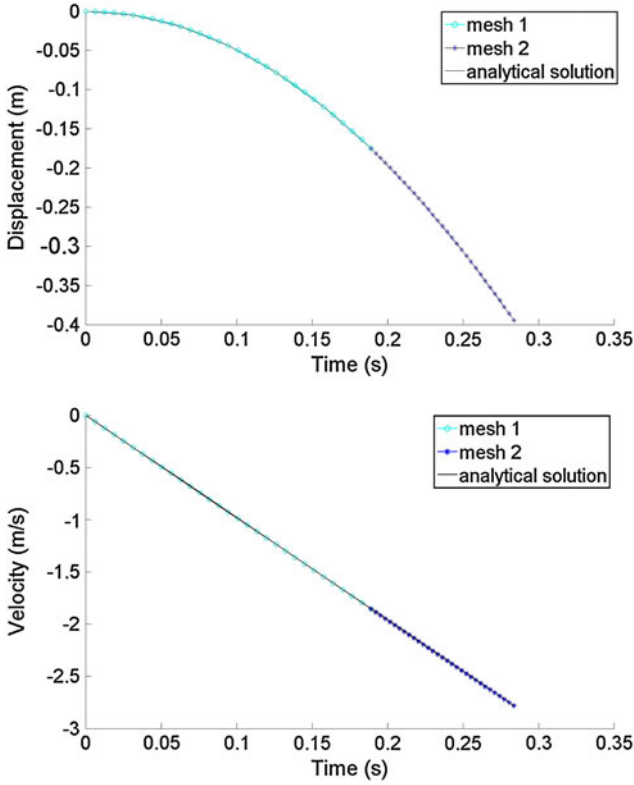


Fig. 15 Evolution of kinematic fields. **a** Displacement. **b** Velocity

small amount of nodes. In this example, there are 12,980 nodes in the “new narrow band” whereas the initial mesh contained 24,389 nodes. The computational cost to get the level-set is divided by two compared to a straight forward approach.

The second simplification of our algorithm then consists in reducing the number of facets tested to obtain the closest one. The iso-zero of the level-set initially contains 3,456 facets. The number of facets tested for each node of the “new narrow band” varies from one node to another between 20

and 200 facets and depends on the curvature of the iso-zero. An average number of about 100 facets are tested per node.

Thanks to these two simplification processes, the total number of nodes times facets investigated for the level-set computation is about 1,298,000. With the straight forward approach, the distance computation would have been done at 24,389 nodes with 3,456 facets, which leads to 84,288,384 computations. Hence, the presented method enables to reduce the computational effort by a factor 65. This first example is very promising. For industrial problems, we can expect that the number of nodes near the level-set is very small compared to the total number of nodes on the new mesh so that the benefit of our algorithms will be much more significant.

3.4.2 Example 2: dynamic traction test

In order to show the relevance of the data transfer approach, we propose in this section a simulation on a nearly cubic part subjected to dynamic loading. The dimensions of the part are $1 \text{ m} \times 1 \text{ m} \times 1.05 \text{ m}$. The boundary conditions are a complete clamping of the lower face and a sudden traction of 45,000 N on the upper face. The lateral faces can only move in the traction direction. This case can be compared to the solution of a rod under a state of uniaxial strain. For the material, we choose the Neo-Hookean model described in Sect. 2.1, whose parameters are typical of a polymeric foam: $\rho = 22 \text{ kg m}^{-3}$, $\lambda = 66,428 \text{ Pa}$ and $\mu = 16,607 \text{ Pa}$.

The part is discretized with a mesh of characteristic length 0.1 m. The loaded face is not meshed, but implicitly represented by the iso-zero of a level-set. We compare the results obtained without remeshing and with five successive remeshing steps. At each remeshing step, the new mesh used to discretize the deformed configuration is chosen equal to the initial mesh. Displacements of a point on the loaded face obtained with and without remeshing are compared in Fig. 18. Results for both simulations are nearly identical. A small gap appears between the two displacements from

Fig. 16 Old narrow band elements on mesh 1. **a** 3D view. **b** Cut view

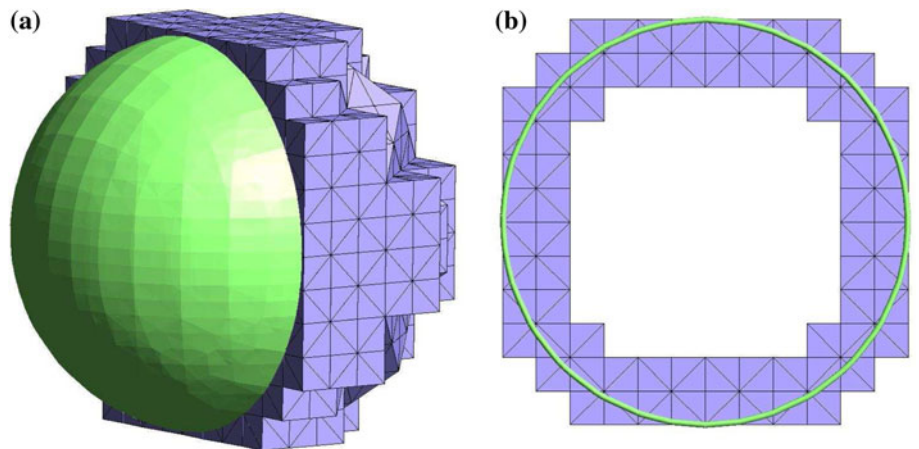


Fig. 17 New narrow band nodes on mesh 2

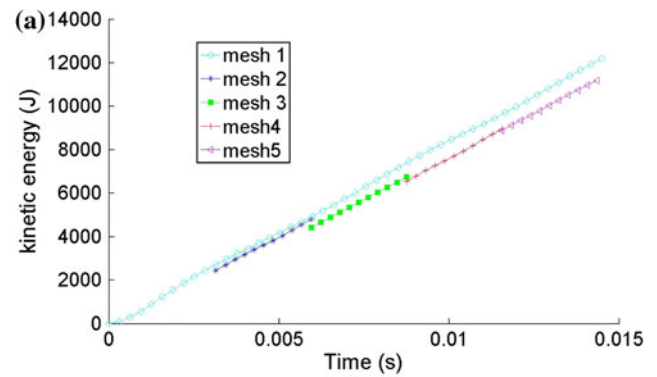
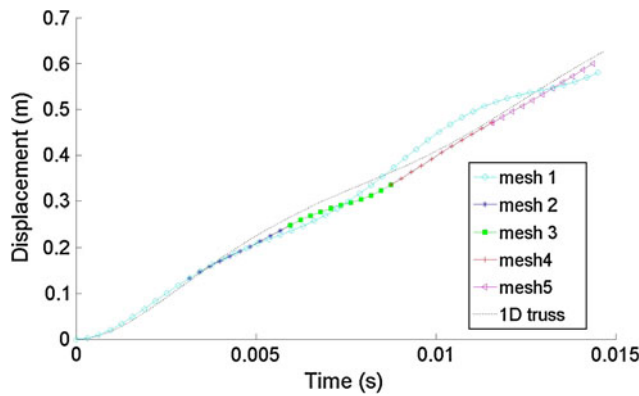
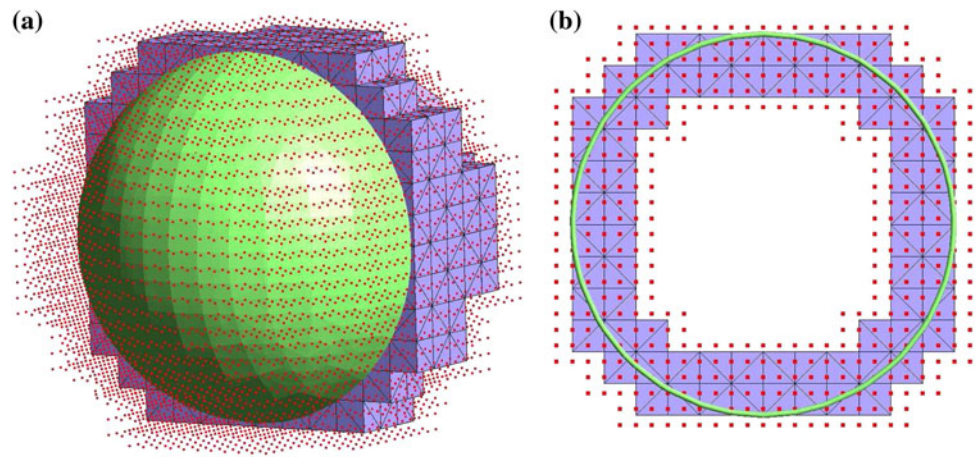


Fig. 18 Dynamic traction test: displacement of the loaded boundary

$t = 0.008$ s. After, this gap tends to reduce and becomes close to zero at $t = 0.013$ s.

We also compare global variables obtained from both simulations. Figure 19 shows the evolution of kinetic and internal energies. The results are also quite similar. Small gaps appear for both energies at different instants. The projection sometimes increases the gap and sometimes decreases it but always remains in an acceptable range.

In order to further analyze the influence of the number of projections, we compute the error of kinetic energy, internal energy (see Fig. 20) and the error on the part volume (see Fig. 21) for different numbers of remeshing steps. We can see that the effect of the remeshing is rather random. The remeshing does not lead to an automatic decrease or increase of energy and volume. Errors induced from a remeshing can be erased by the next one. For some particular numbers of projections (especially 5 and 7), one can notice errors of about 10% which could be worrying. Nevertheless, this is due to the particular design of this test where the part upper boundary is straight and raises progressively on a regular mesh. The projection error is therefore amplified because several elements are cut in the same way. For a more complex geometry, this phenomena is expected to be perfectly random so

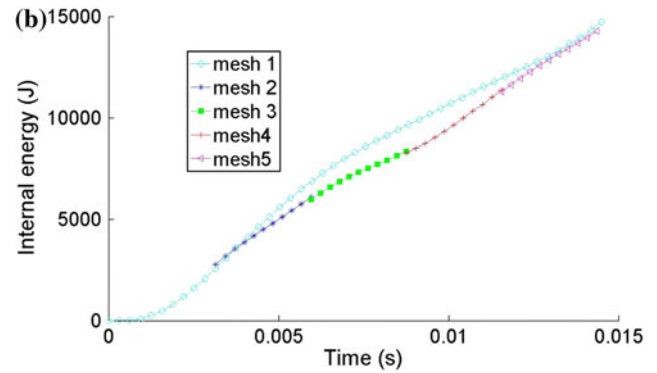


Fig. 19 Dynamic traction test: evolution in time of kinetic (a) and internal energies (b)

that the error will not be so exaggerated. The approximation induced by the projection technique is therefore acceptable.

3.4.3 Example 3: dynamic compression test

We consider here the same part than in the previous example but we change the load to a compression of 90,000 N on the upper face. The mesh is chosen identical to the previous case. Displacements obtained without remeshing can be seen in Fig. 22 where a compression ratio of 60% was reached.

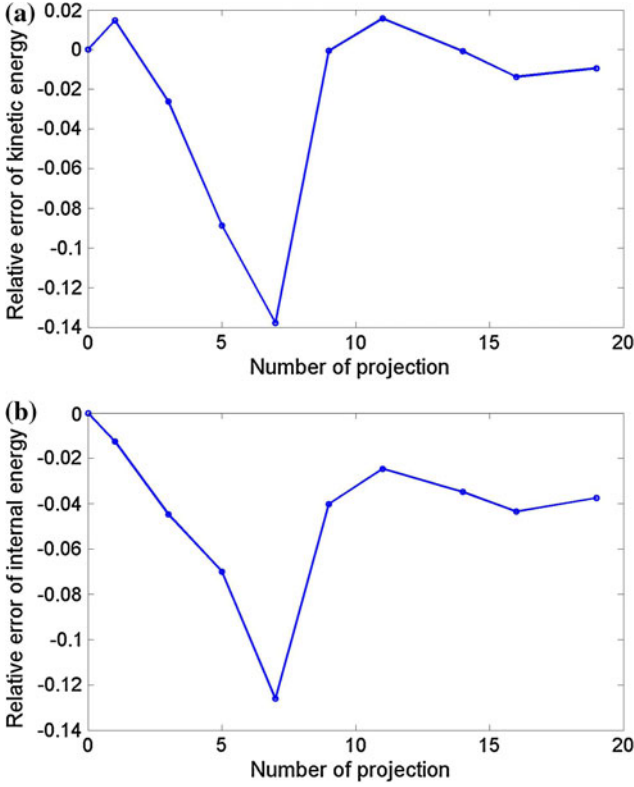


Fig. 20 Dynamic traction test: evolution of the relative error of the kinetic (a) and the internal energy (b) with the number of remeshing steps

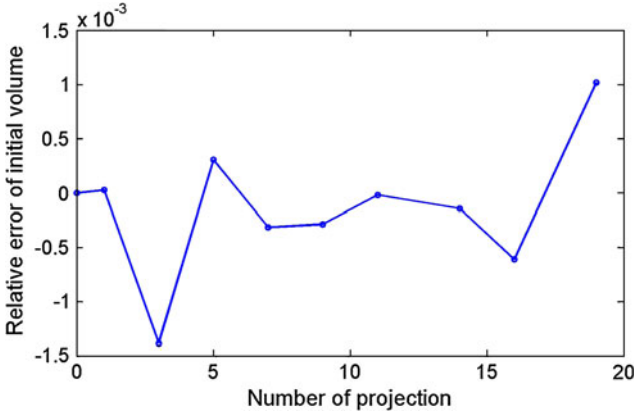


Fig. 21 Dynamic traction test: evolution of the relative error of the initial volume with the number of remeshing steps

In order to estimate the distortion of the mesh, we use a simple aspect ratio quality factor defined by:

$$Q = \alpha \frac{h_{\max}}{\rho} = \alpha \frac{h_{\max} S}{3V} \quad (21)$$

where $\alpha = \frac{\sqrt{6}}{12}$ is a normalization coefficient allowing to have a quality equal to 1 for an equilateral tetrahedron, h_{\max} is the length of the longest edge, ρ is the radius of the inscribed sphere, S is the sum of face surfaces and V is the volume.

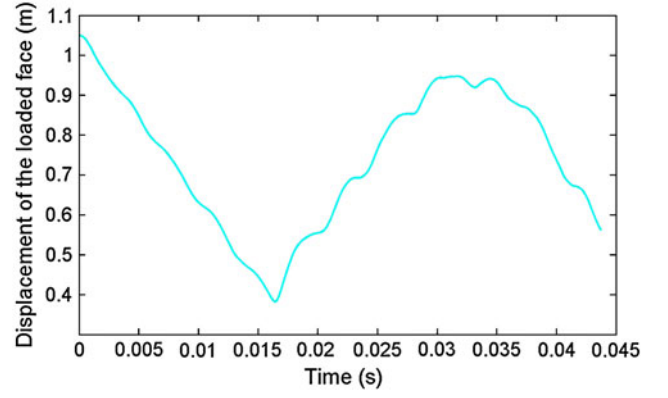


Fig. 22 Dynamic compression test: displacement of one point of the loaded face

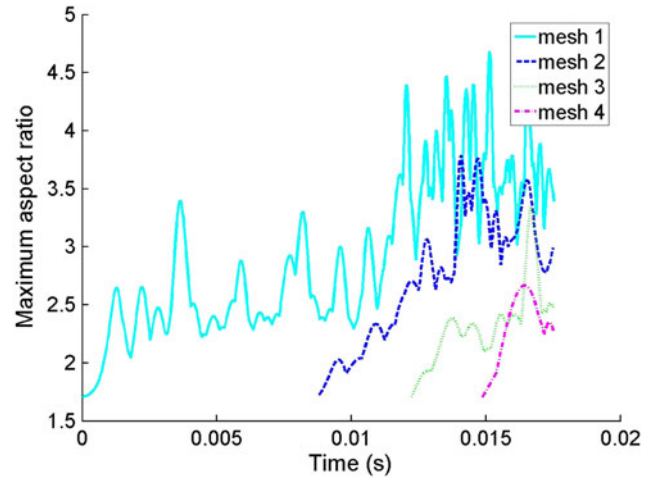


Fig. 23 Dynamic compression test: evolution in time of the maximum aspect ratio of elements with different remeshing frequencies

Our mesh is made of tetrahedra with an initial aspect ratio of 1.71. During the compression stage, from the beginning to $t = 0.016$ s, the maximum aspect ratio increases and gets close to 5 so that the accuracy of results for these elements is no longer guaranteed. Furthermore, the critical time step, which is directly linked to the elements shape, decreases dramatically, starting from 3.5×10^{-4} to 3.87×10^{-5} s. This can be excessively time consuming in an industrial context.

To highlight the way our remeshing strategy enables to overcome those two problems, different situations of remeshing were tested: 2, 3 and 4 successive remeshing steps during the compression stage. For each test, the projection mesh used to remesh the deformed part is identical to the initial one. As shown in Fig. 23, the distortion goes back to its initial value at each remeshing, which is expected because of the perfect regularity of the new mesh. Furthermore, the value of the critical time step is maintained at higher values thanks to the remeshing (Fig. 24). Even if the time step does not go back to its initial value (the wave velocity is not the same in

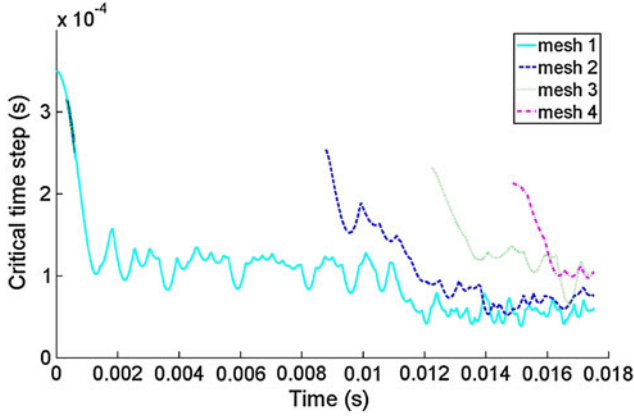


Fig. 24 Dynamic compression test: evolution in time of the critical time step for different remeshing strategies

the deformed configuration), the four remeshing steps allow to multiply the time step obtained without remeshing by a factor four approximately.

This last test finally validates the principle of the proposed technique and demonstrates its efficiency in the context of dynamic simulations implying large deformations. Nevertheless, the considered situations are still geometrically rather simple and the next section will focus on more complex deformations induced by impacts on compressible bodies.

4 Impact problems

Impact situations are present in many industrial problems, from manufacturing processes to the crash of automobiles. It seems important to include the modeling of these phenomena in our method.

During an impact, one or several bodies stay in contact until the vanishing of waves results in a release of contact forces. We can distinguish two main approaches to treat contact constraints: the Lagrange multiplier method and the penalty method. These two methods have been used as the basis of most contact algorithms [24].

The main principle of Lagrange methods is to ensure the equality of the displacements at the contact interface between two bodies by means of Lagrange multipliers, which can be seen as normal tensions. The advantage of this approach is the exact respect of the non-penetration condition but its drawback is that it requires the introduction of new degrees of freedom. Furthermore, in explicit dynamics, the Lagrange multiplier method implies the solution of an implicit system at each time step [7], which can be time consuming if contact interfaces are large. With the usual penalty method, the normal tensions are approximated by forces which are proportional to the penetration. The major drawback of this method is that some interpenetration of the contacting bodies

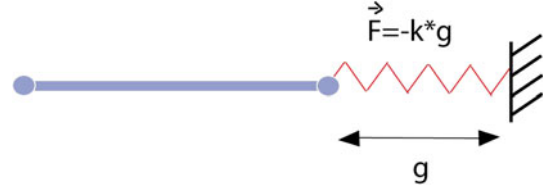


Fig. 25 One dimensional approach of the penalty method

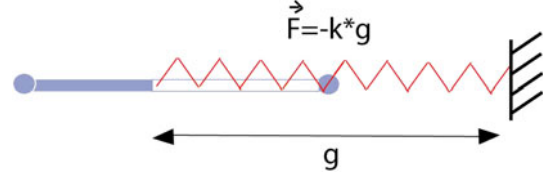


Fig. 26 One dimensional approach of the penalty method with the X-FEM

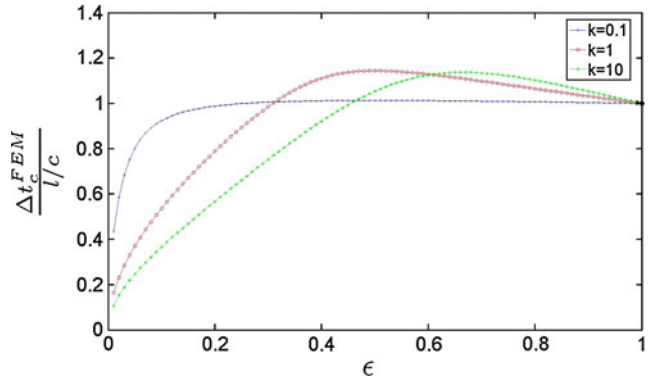


Fig. 27 One dimensional penalty test case: evolution of the critical time step with the fraction of matter for different values of the penalty function k

is allowed. Nevertheless, in the case of large deformations we consider here, this approximation can be considered as precise enough. Despite this little drawback, this method is easy to implement and computationally efficient because it does not involve any additional degrees of freedom. The major problem induced by this method is that in contrast to Lagrange multipliers method, the penalty method always decreases the critical time step [2]. The choice between the two methods depends on the accuracy needed and on the time of computation wanted. In our case, the use of the X-FEM has also to be considered to make a choice.

In the X-FEM context, contact interfaces are not meshed. This particularity has to be taken into account in the contact algorithms. The use of the X-FEM with the Lagrange multipliers method requires special care to satisfy the inf-sup condition needed for mixed formulations. Geniaut et al. [10] and Béchet et al. [1, 16] have proposed specific algorithms for statics problems to define the appropriate Lagrange multipliers discretization space. For large deformation dynamics problems, the search of this discretization space has to be

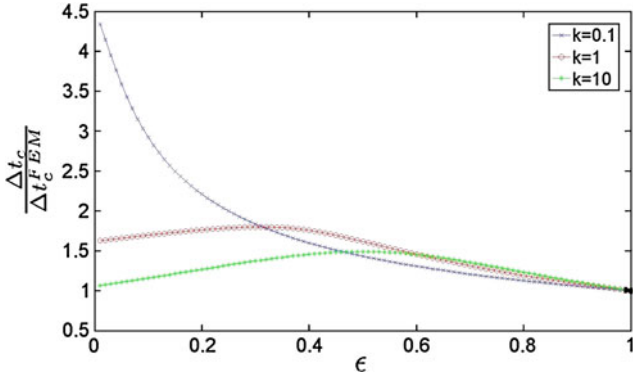


Fig. 28 One dimensional penalty test case: evolution of the critical time step according to the fraction of matter with added mass for different values of the penalty factor k

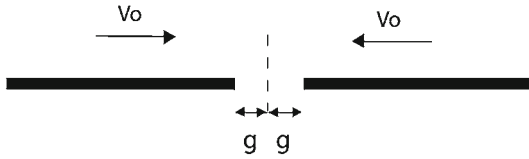


Fig. 29 One dimensional impact test

done at each time step since the contact interfaces are moving so that the computational cost of this operation can become very important in an explicit framework. Furthermore, when remeshing is needed, we cannot ensure that the transfer of Lagrange multipliers leads to the conformance with the inf-sup condition on the new mesh. In this context, the penalty method seems to be a more suitable choice.

In the following section, we will describe the use of the penalty method in explicit dynamics in classical finite elements. We will then emphasize the special treatments due to the introduction of the X-FEM. Finally, we will discuss on its reliability through some numerical examples.

4.1 Penalty method

The penalty method originates in optimization theory. Its application in contact problems consists in adding the above term to the weak form of the linear momentum:

$$\delta \mathcal{G}_P = \int_{\Gamma_c} \frac{k'}{2} \delta(g_N^2) \mathcal{H}(g_N) d\Gamma \quad (22)$$

where g_N is the normal gap function between two bodies, Γ_c is the contact interface, k' is the penalty parameter and $\mathcal{H}(g_N)$ is the Heaviside function:

$$\mathcal{H}(g_N) = \begin{cases} 1 & \text{if } g_N > 0 \\ 0 & \text{if } g_N < 0 \end{cases}$$

The non-penetration condition is supposed to be better satisfied as the penalty parameter k' is increased. The

Table 1 One dimensional impact test: numerical values

Young modulus	$E = 2.1E11$ Pa
Density	$\rho = 7,800$ kg m ⁻³
Cross section	$S = 6.4516 \times 10^{-4}$ m ²
Length	$l = 0.254$ m
Half initial gap	$g = 2.54 \times 10^{-4}$ m
Velocity	$v_o = 5.1359$ m s ⁻¹
Penalty parameter	$k = 0.1$

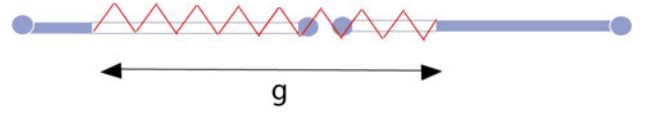


Fig. 30 One dimensional impact test: representation of the X-FEM mesh

penalty method (Fig. 25) can be seen as the addition of a spring between two bodies in order to keep them separated. It increases the stiffness of the elements at the boundary in contact. In the context of explicit methods, this leads to a decrease of the critical time step [3]. This feature can be easily demonstrated in 1D considering the impact of a truss on a rigid wall. The stability is governed by the element in contact associated with the penalty spring.

The mass and the linearized stiffness matrices are respectively given by:

$$M = \frac{\rho S l}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad K = \frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1+k \end{bmatrix}$$

where $k = \frac{l}{ES} k'$.

By applying the eigenvalue element inequality, we obtain the FEM critical time step Δt_c^{FEM} :

$$\Delta t_c^{\text{FEM}} = \frac{l}{c} \frac{2}{\sqrt{2+k} + \sqrt{4+k^2}} \quad (23)$$

The critical time step decreases as the inverse of penalty parameter k . For example, if k is chosen equal to 10 (i.e. the penalty spring is ten time more stiff than the material), the time step is decreased by 58%.

4.2 Use of the penalty method in explicit dynamics with X-FEM

4.2.1 Stability

Within the X-FEM, the penalty traction has to be applied on a surface which is not meshed. For a one dimensional case this can be modeled by attaching the penalty spring at the boundary of the material part of the X-FEM element.

As clear from Fig. 26, the boundary element is crossed by the interface at a distance ϵl of its length l . For such a case,

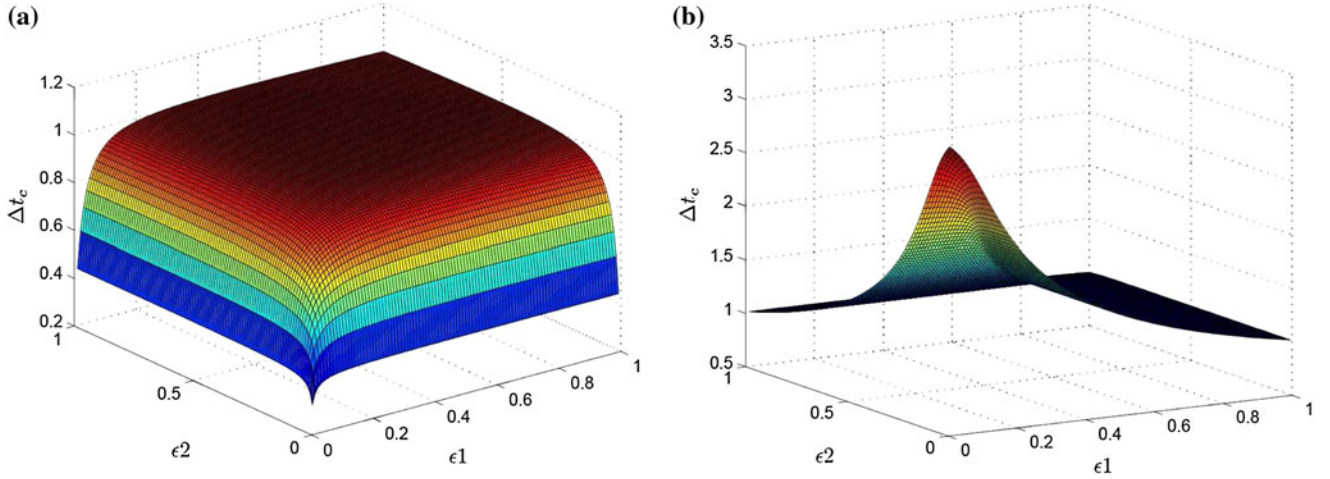


Fig. 31 One dimensional impact test: evolution of the X-FEM critical time step with (a) and without (b) mass scaling

the mass and the linearized stiffness matrices are then given by:

$$M = \frac{1}{2} \rho S l \begin{bmatrix} \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$$

$$K = \frac{ES}{l} \begin{bmatrix} \epsilon + (1 - \epsilon)^2 k & -\epsilon + \epsilon(1 - \epsilon)k \\ -\epsilon + \epsilon(1 - \epsilon)k & \epsilon + \epsilon^2 k \end{bmatrix}$$

where ϵ represents the fraction of matter in the X-FEM element.

Applying again the eigenvalues criterion, the critical time step is obtained by:

$$\Delta t_c = \frac{l}{c} \frac{2\sqrt{\epsilon}}{\sqrt{2\epsilon + k - 2k\epsilon + 2\epsilon^2 k + \sqrt{4\epsilon^2 - 8\epsilon^2 k + 8\epsilon^3 k + k^2 - 4k^2\epsilon + 8k^2\epsilon^2 - 8k^2\epsilon^3 + 4\epsilon^4 k^2}}}$$

As illustrated on Fig. 27, if the material fraction ϵ tend to zero, then the critical time step tend to zero.

The meshing procedure with X-FEM is independent of the position of the boundary, so that the parameter ϵ is not controlled and the values of the critical time step can be extremely low. This is not acceptable for industrial applications. To overcome this drawback, we propose to add mass on the X-FEM element. The added mass corresponds to the complementary mass of the “void part” of the X-FEM element. The mass matrix is therefore identical to the one obtained in finite element whereas the stiffness remains the one of the X-FEM element:

$$M = \frac{1}{2} \rho S l \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad K = \frac{ES}{l} \begin{bmatrix} \epsilon + (1 - \epsilon)^2 k & -\epsilon + \epsilon(1 - \epsilon)k \\ -\epsilon + \epsilon(1 - \epsilon)k & \epsilon + \epsilon^2 k \end{bmatrix}$$

The critical time step is then given by:

$$\Delta t_c = \frac{l}{c} \frac{2}{\sqrt{2\epsilon + k - 2k\epsilon + 2\epsilon^2 k + \sqrt{4\epsilon^2 - 8\epsilon^2 k + 8\epsilon^3 k + k^2 - 4k^2\epsilon + 8k^2\epsilon^2 - 8k^2\epsilon^3 + 4\epsilon^4 k^2}}}$$

When ϵ tend to zero, the critical time step tend to a bounded limit equal to $\frac{l}{c} \sqrt{\frac{2}{k}}$. In Fig. 28, we plot the X-FEM critical time step normalized by the one obtained in finite element with penalty Δt_c^{FEM} . We can observe that this ratio is always greater than one. Thanks to mass addition, the X-FEM time step stays greater than the one in finite elements.

This technique is a mass scaling method which has the particularity to add a naturally bounded quantity of mass. This additive mass is low compared to the mass of the

complete structure so it is expected not to affect significantly the results.

4.2.2 One dimensional impact example

In order to check the reliability of the proposed method, we consider the 1D case of two linear elastic rods moving with equal speed in opposite directions (see Fig. 29). We call g the half distance initially separating the two rods.

The numerical values are given in Table 1.

We compare the results obtained by classical FEM and by X-FEM. For both cases, each rod is discretized with 20 elements. In the X-FEM case, the impacting boundaries of the rods are not meshed: the fraction of matter for the impacting element of the rod on the left-hand side is denoted as ϵ_1 ,

whereas it is denoted as ϵ_2 for the rod at the right-hand side (see Fig. 30).

The stability of the explicit scheme is governed by the two impacting elements properties associated with the penalty spring. The time steps will differ a bit from the previous study since here the two impacted bodies are deformables.

For the FEM case (meshed boundaries), the computation of the critical time step is based on the eigenvalues of the above matrices:

$$M = \frac{\rho S l}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad K = \frac{E S}{l} \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+k & -k & 0 \\ 0 & -k & 1+k & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

which lead to the following FEM critical time step:

$$\Delta t_c^{\text{FEM}} = \frac{l}{c} \sqrt{\frac{2}{1+k+\sqrt{1+k^2}}} \quad (24)$$

For the X-FEM case without mass scaling, the element matrices depend on ϵ_1 and ϵ_2 :

$$M = \frac{\rho S l}{2} \begin{bmatrix} \epsilon_1 & 0 & 0 & 0 \\ 0 & \epsilon_1 & 0 & 0 \\ 0 & 0 & \epsilon_2 & 0 \\ 0 & 0 & 0 & \epsilon_2 \end{bmatrix}$$

$$K = \frac{E S}{l} \begin{bmatrix} \epsilon_1 + k(1-\epsilon_1)^2 & -\epsilon_1 + k\epsilon_1(1-\epsilon_1) & -k\epsilon_2(1-\epsilon_1) & -k(1-\epsilon_1)(1-\epsilon_2) \\ -\epsilon_1 + k\epsilon_1(1-\epsilon_1) & \epsilon_1 + k\epsilon_1^2 & -k\epsilon_1\epsilon_2 & -k\epsilon_1(1-\epsilon_2) \\ -k\epsilon_2(1-\epsilon_1) & -k\epsilon_1\epsilon_2 & \epsilon_2 + k\epsilon_2^2 & -\epsilon_2 + k\epsilon_2(1-\epsilon_2) \\ -k(1-\epsilon_1)(1-\epsilon_2) & -k\epsilon_1(1-\epsilon_2) & -\epsilon_2 + k\epsilon_2(1-\epsilon_2) & \epsilon_2 + k(1-\epsilon_2)^2 \end{bmatrix}$$

An analytical formula of the critical time step for the X-FEM would be difficult to analyse. We just represent in Fig. 31 the evolution of the time step for different values of ϵ_1 and ϵ_2 with and without mass scaling. This time step is normalized by the time step obtained with classical finite element Δt_c^{FEM} (see Eq. 24). As in the previous study, the X-FEM time step tends to zero when ϵ_1 or ϵ_2 tend to zero. This drawback disappears when using the mass scaling technique: the X-FEM time step even becomes greater than the FEM one.

For a case where $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.9$, we compare the results obtained by FEM and X-FEM. The mass scaling for X-FEM leads to a mass addition of 2.61% of the total mass. The evolution of displacements, velocities, accelerations and contact forces at the two contacting boundaries for FEM and X-FEM are represented in Figs. 32 and 33.

We can see that displacements, velocities and contact forces obtained with X-FEM are very close to the classical FEM solutions. However, some differences between the results of the two methods can be observed for the acceleration. Analytically, the acceleration is always zero except at the impact and at the release instants where it tends to

infinity. Obviously, such an evolution cannot be represented numerically: one only gets maximum values at impact and release times as well as numerical oscillations between these two states. These oscillations are minimized in the X-FEM thanks to the mass addition.

In Fig. 34, we can also observe the differences between X-FEM and FEM in the evolution of kinetic energy. As expected, the addition of mass with the X-FEM leads to a small increase of kinetic energy at the impact time. However, this increase does not involve significant differences between X-FEM and FEM since the mass addition is limited to 2.61% of the total mass.

This example shows that the X-FEM with mass addition allows to obtain accurate results while keeping a reasonable time step.

4.3 Remeshing during impact

We have demonstrated in the previous paragraph the possibility to treat impact problem within X-FEM. We know

focus on the effects of the remeshing process on the impact treatment, therefore combining the remeshing technique of Sect. 3 and the present penalty technique.

4.3.1 Procedure

As mentioned in Sect. 3, for a remeshing occurring at time t_{n+1} , we only transfer the predictors of displacements and velocities: \mathbf{u}^{n+1} and $\dot{\mathbf{u}}^{n+\frac{1}{2}}$. The acceleration $\ddot{\mathbf{u}}^{n+1}$ is evaluated by resolution of the equilibrium equation on the second mesh. The velocity $\dot{\mathbf{u}}^{n+1}$ is deduced from this acceleration.

The contact condition will be taken into account in the computation of $\ddot{\mathbf{u}}^{n+1}$. In explicit methods, the acceleration at time t_{n+1} does not affect the displacement at the same time. That is why Carpenter et al. [7] pointed that the contact condition has to be treated for the forward increment. This means that the computation of $\ddot{\mathbf{u}}^{n+1}$ will use contact forces depending on the penetration predicted for time t_{n+2} . We can conclude that there is no contact force to transfer between the old and the new meshes in explicit dynamics.

Due to the change of discretization between both meshes, the penetration on the second mesh at time t_{n+1} will differ

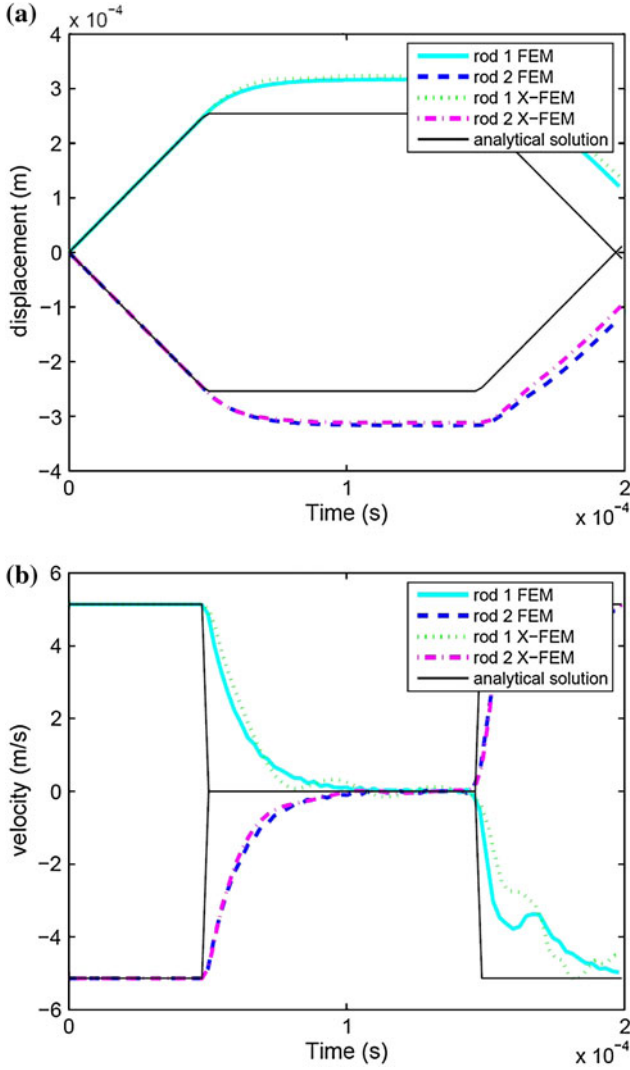


Fig. 32 One dimensional impact test: evolution of displacements (a) and velocities (b) at contacting boundaries, comparison between the FEM and X-FEM solutions for $\epsilon_1 = 0.1$, $\epsilon_2 = 0.9$ and $k = 0.1$.

from the penetration at the same time on the first mesh. In a Lagrange multiplier method, the impenetrability condition is ensured on the first mesh whereas interpenetration appears in the second one due to projection. The computation of Lagrange multipliers on the second mesh would be necessary to overcome this issue. In the penalty method, some interpenetration is allowed, so there is no reason to try to eliminate it on the second mesh. Thus, the accuracy of the results will not be affected by the remeshing procedure with the penalty method.

4.3.2 Numerical example

We now consider a 3D example involving remeshing during impact. We aim to simulate the impact of a sphere against a rigid plane. The sphere, of initial radius 0.3 m has an initial

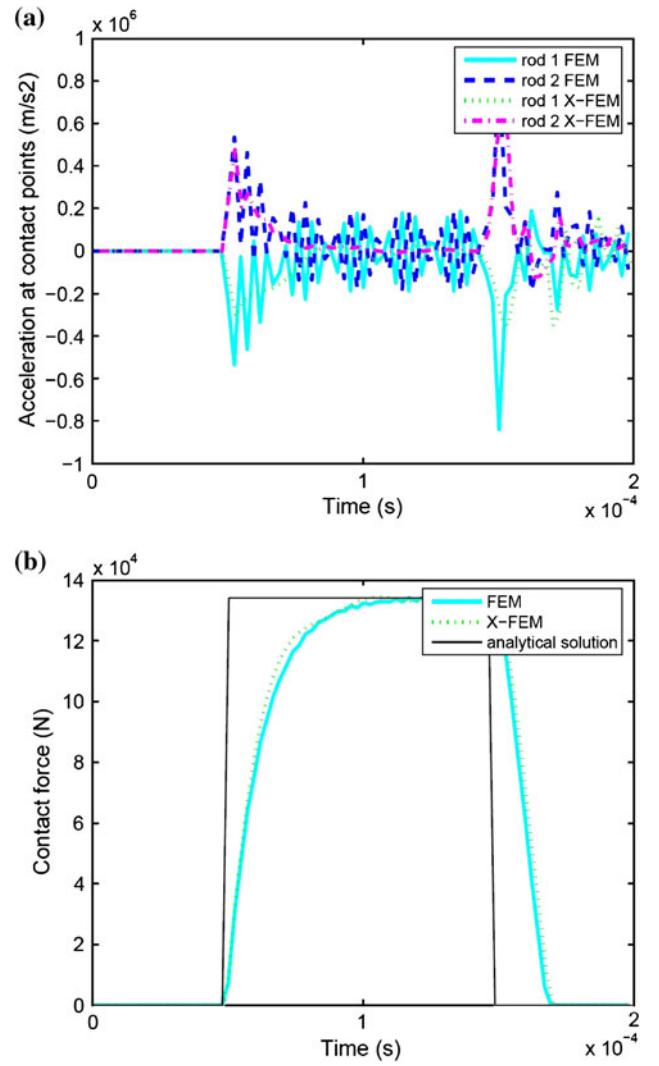


Fig. 33 One dimensional impact test: evolution of acceleration (a) and contact force (b) at contacting boundaries, comparison between the FEM and X-FEM solutions for $\epsilon_1 = 0.1$, $\epsilon_2 = 0.9$ and $k = 0.1$.

velocity of 10 m s^{-1} . We use a regular isotropic mesh, of characteristic length 0.05 m, made of tetrahedra built according to the process described in Sect. 3.4.1. The geometry of the sphere is described by a level-set function and the plane is not discretized but described by an analytical function. The initial gap between the sphere and the plane is set to 0.005 m so that contact occurs at the beginning of the computation. During the impact, the deformed sphere is remeshed by our projection method on a mesh identical to the initial one as described in Fig. 35.

The wave propagation due to impact is shown in Figs. 36 and 37, where one can see the distribution of the displacements in a cutting plane of the sphere at different time steps.

In Fig. 38, we compare the evolution of kinetic and internal energies with and without remeshing. These energies are also compared to the one obtained with ABAQUS code.

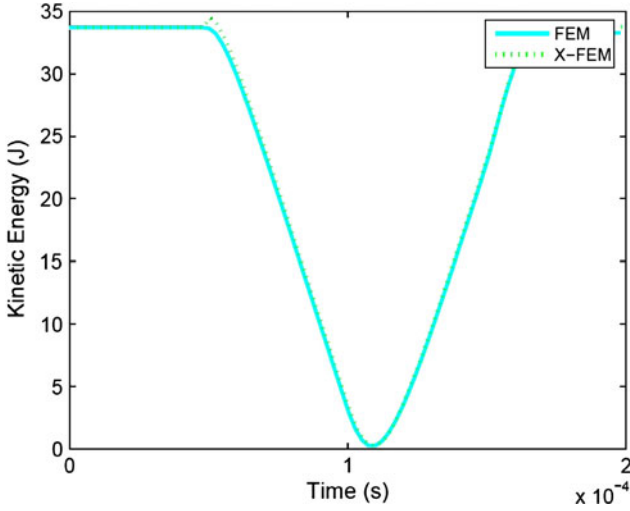


Fig. 34 One dimensional impact test: evolution of kinetic energy, comparison between the FEM and X-FEM solutions for $\epsilon_1 = 0.1$, $\epsilon_2 = 0.9$ and $k = 0.1$

We can observe a fairly good agreement between the three simulations. The remeshing leads to a small increase of the kinetic energy and a small decrease of the internal energy. These gaps seem reasonable compared to the amount of energy involved in this problem and to the other discretization errors.

Fig. 35 Impact of a sphere: state of the sphere at the beginning of the simulation on mesh 1 (a) and at the beginning of the simulation on mesh 2 (b)

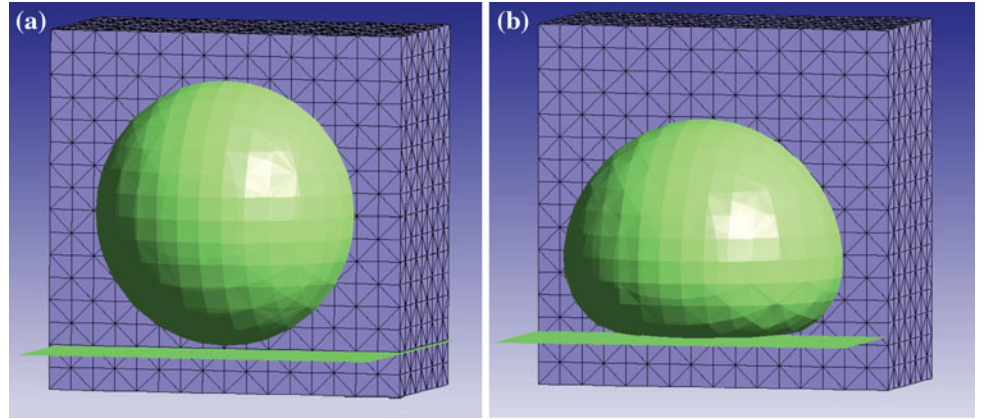
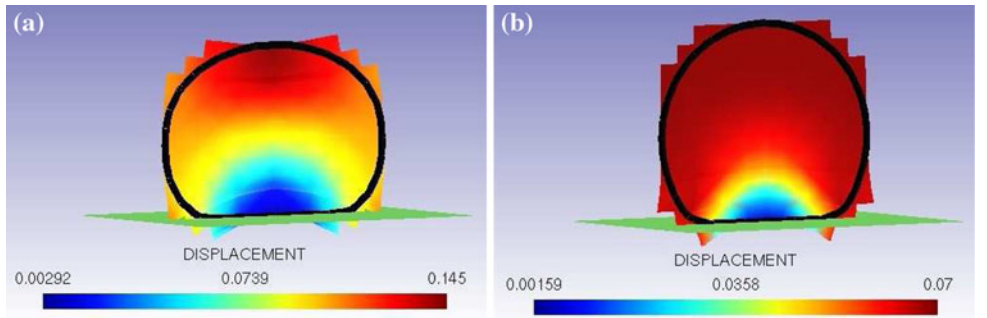


Fig. 36 Impact of a sphere: displacement field at iteration 100 (a) and 300 (b)



The penetration is almost the same from one mesh to another after the remeshing. This is illustrated in Fig. 39 where the displacement of an impacting point was plotted.

Finally, we can observe the benefit coming from the remeshing on the distortion rate (Fig. 40) and on the critical time step (Fig. 41). Without remeshing, the aspect ratio increases to values close to 4 whereas it is kept under 2.8 by the remeshing operation. Furthermore, as visible from Fig. 41, the remeshing also allows to keep a bigger time step while the material undergoes severe deformations. On this example, at the most deformed state ($t = 0.019$ s), the time step is multiplied by 4.

5 Conclusion

In this paper, we propose a new strategy to deal with impact problems on highly compressible media in the framework of explicit dynamics. The basic idea is to use the X-FEM to simplify the remeshing work needed for this kind of problems. This simplification relies on the fact that the X-FEM allows to remesh with meshes that do not match the shape of the deformed structure. A simple structured mesh can be used each time whenever a remeshing is needed.

The data transfer between successive meshes is done by a collocation method. An important step in that process is to locate the new nodes inside the old mesh. The fact of

Fig. 37 Impact of a sphere:
displacement field at iteration
500 (a) and 680 (b)

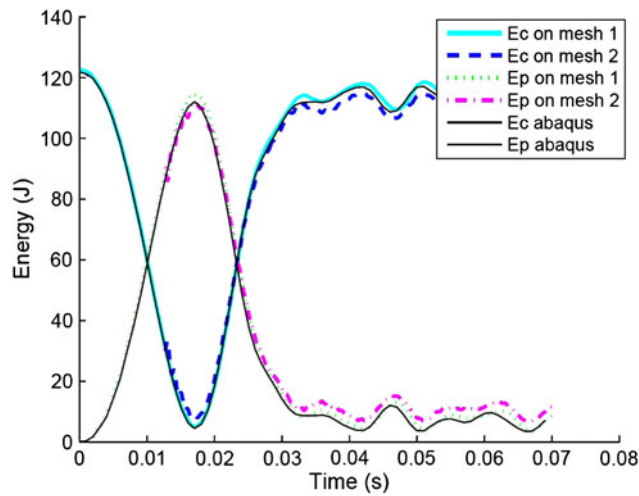
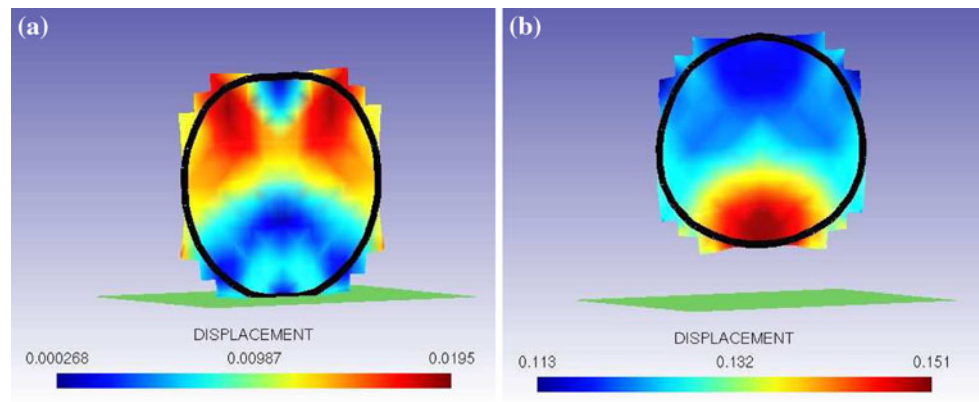


Fig. 38 Impact of a sphere: evolution of kinetic and internal energies

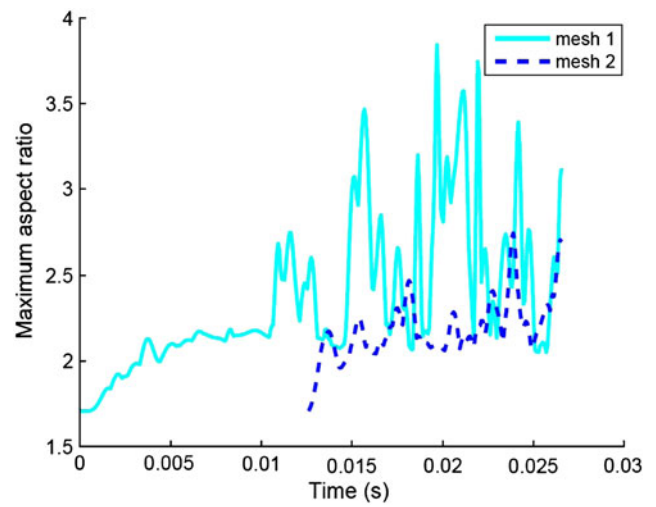


Fig. 40 Evolution of maximum aspect ratio

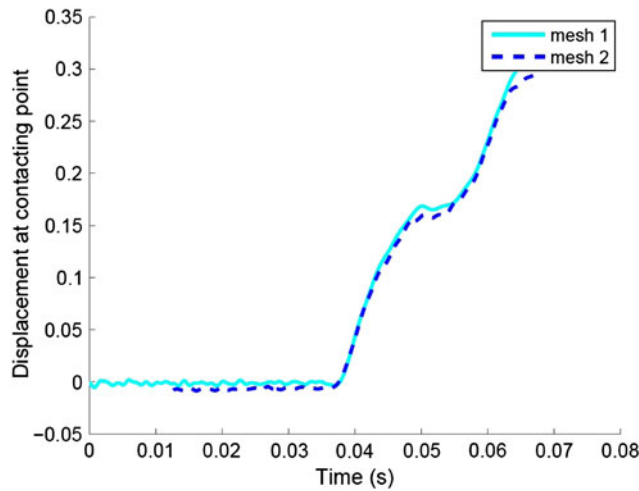


Fig. 39 Impact of a sphere: evolution of the displacement at a contact-
ing point

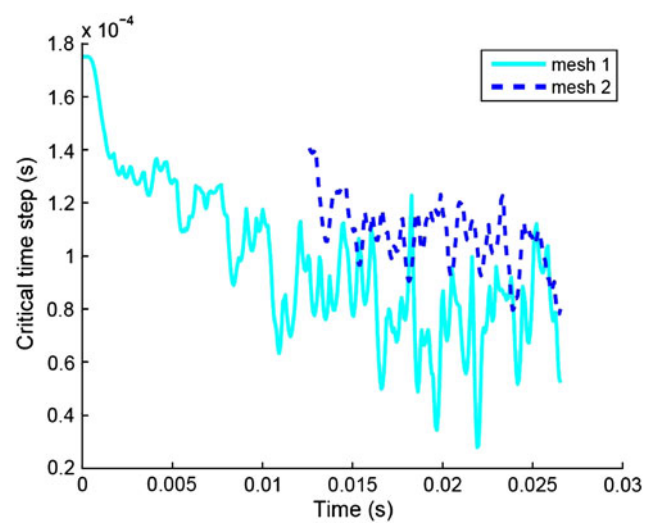


Fig. 41 Evolution of the critical time step

using regular and structured meshes makes this step considerably easier: we can use a specific search algorithm based on computational graphics methods to save computational

time. Special care is devoted to the building of the iso zero of the level-set representing the deformed piece on the new mesh.

Efficiency and accuracy of the proposed algorithms are tested on several examples. We particularly demonstrate that the remeshing strategy allows to limit the distortion and to keep a reasonable time step when the material is submitted to severe strains. The method therefore brings all the satisfying improvements one expects from a remeshing process. Naturally, compared to other remeshing techniques, our use of regular grids can be considered as non-optimal and yet somewhat academic. The extension to a more adapted mesh, which will be the subject of further works, leads to heavier calculations.

In order to be able to deal with more realistic physical situations, the treatment of an impacting rigid body was also implemented, the contact problem being treated through a penalty method. As the use of such a method with X-FEM in explicit dynamics led to a decrease of the critical time step, we proposed a specific mass lumping strategy for the X-FEM elements to overcome this issue. Excellent agreement with reference solution results are obtained for 1D and 3D examples.

Acknowledgments We would like to thank PSA Peugeot-Citron for their financial support and more particularly M. Laurent Rota for his scientific advice.

References

1. Béchet E, Moës N, Wohlmuth B (2009) A stable lagrange multiplier space for the stiff interface conditions within the extended finite element method. *Int J Numer Methods Eng* 78:931–954
2. Belytschko T, Liu WK, Moran B (2000) *Nonlinear finite elements for continua and structures*. Wiley, New York
3. Belytschko T, Neal MO (1991) Contact-impact by the pinball algorithm with penalty and lagrangian methods. *Int J Numer Methods Eng* 31:547–572
4. Benson DJ (1992) Computational methods in lagrangian and eulerian hydrocodes. *Comput Methods Appl Mech Eng* 99:235–394
5. Boroomand B, Zienkiewicz OC (1999) Recovery procedure in error estimation and adaptivity. Part II: adaptivity in nonlinear problems of elasto-plasticity behaviour. *Comput Methods Appl Mech Eng* 176:127–146
6. Breen D, Mauch S, Whitaker R (2000) 3D scan conversion of csg models into distance, closest-point and color volumes. In: *Proceedings of volume graphics* (2000), pp 135–158
7. Carpenter NJ, Taylor RL, Katona MG (1991) Lagrange constraints for transient finite element surface contact. *Int J Numer Methods Eng* 32:103–128
8. Cugnon F (2000) *Automatisation des calculs éléments finis dans le cadre de la méthode -p*. PhD thesis, Université de Liège, Mars
9. Daux C, Moës N, Dolbow J, Sukumar N, Belytschko T (2000) Arbitrary branched and intersecting cracks with the extended finite element method. *Int J Numer Methods Eng* 48:1741–1760
10. Geniaut S, Massin P, Moës N (2007) A stable 3D contact formulation for cracks using x-fem. *Revue Européenne de Mécanique Numérique* 16:259–275
11. Khoei AR, Anahid M, Shahim K (2008) An extended arbitrary Lagrangian-Eulerian finite element method for large deformation of solids mechanics. *Finite Elem Anal Des* 44(6–7):401–416
12. Lee NS, Bathe KJ (1994) Errors indicators and adaptive remeshing in large deformation finite element analysis. *Finite Elem Anal Des* 16:99–139
13. Mediavilla J, Peerlings RHJ, Geers MGD (2006) A robust and consistent remeshing-transfer operator for ductile fracture simulations. *Comput Struct* 84:604–623
14. Melenk JM, Babuska I (1996) The partition of unity finite element method: basic theory and applications. *Comput Methods Appl Mech Eng* 139:289–314
15. Menouillard T, Rethore J, Moës N, Combescure A, Bung H (2008) Mass lumping strategies for x-fem explicit dynamics: application to crack propagation. *Int J Numer Methods Eng* 74(3):447–474
16. Moës N, Béchet E, Tourbier M (2006) Imposing Dirichlet boundary conditions in the extended finite element method. *Int J Numer Methods Eng* 67(12):1641–1669
17. Moës N, Cloirec M, Cartraud P, Remacle JF (2003) A computational approach to handle complex microstructure geometries. *Comput Methods Appl Mech Eng* 192:3163–3177
18. Moës N, Dolbow JE, Belytschko T (1999) A finite element method for crack growth without remeshing. *Int J Numer Methods Eng* 46:131–150
19. Rozycki P, Moës N, Béchet E, Dubois C (2008) X-fem explicit dynamics for constant strain elements to alleviate mesh constraints on internal or external boundaries. *Comput Methods Appl Mech Eng* 197(5):349–363
20. Sethian JA (1999) *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and material science*. Cambridge University Press
21. Sukumar N, Chopp DL, Moës N, Belytschko T (2001) Modeling holes and inclusions by level sets in the extended finite element method. *Comput Methods Appl Mech Eng* 190:6183–6200
22. Terada K, Maruyama A, Kurumatani M (2007) Eulerian finite cover method for quasi-static equilibrium of hyperelastic bodies. *Commun Numer Methods Eng* 23:1081–1094
23. Vitali E, Benson DJ (2006) An extended finite element formulation for contact in multi-material arbitrary Lagrangian-Eulerian calculations. *Int J Numer Methods Eng* 67(10):1420–1444
24. Wriggers P (2006) *Computational contact mechanics*. Wiley, New York