HAL
open science

On some aspects of the CNEM implementation in 3D in order to simulate high speed machining or shearing

Lounes Illoul, Philippe Lorong

To cite this version:

Lounes Illoul, Philippe Lorong. On some aspects of the CNEM implementation in 3D in order to simulate high speed machining or shearing. Computers & Structures, 2011, 89 (11-12), pp.940-958. 10.1016/j.compstruc.2011.01.018 . hal-01188918

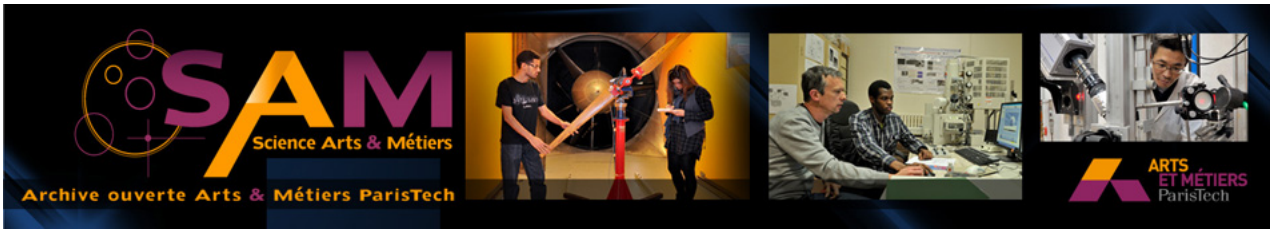
HAL Id: hal-01188918

https://hal.science/hal-01188918v1

Submitted on 31 Aug 2015

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.



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/9871

To cite this version :

Lounes ILLOUL, Philippe LORONG - On some aspects of the CNEM implementation in 3D in order to simulate high speed machining or shearing - Computers & Structures - Vol. 89, n°11-12, p.940-958 - 2011

Any correspondence concerning this service should be sent to the repository Administrator : archiveouverte@ensam.eu

On some aspects of the CNEM implementation in 3D in order to simulate high speed machining or shearing

Lounes Illoul, Philippe Lorong

Arts et Metiers ParisTech 151 boulevard de l'Hôpital, F-75013 Paris France,
amran.illoul@orange.fr - philippe.lorong@paris.ensam.fr

Abstract

This paper deals with the implementation in 3D of the Constrained Natural Element Method (CNEM) in order to simulate material forming involving large strains. The CNEM is a member of the large family of mesh-free methods, but is at the same time very close to the finite element method. The CNEM's shape function is built using the constrained Voronoï diagram (the dual of the constrained Delaunay tessellation) associated with a domain defined by a set of nodes and a description of its border.

The use of the CNEM involves three main steps. First, the constrained Voronoi diagram is built. Thus, for each node, a Voronoi cell is geometrically defined, with respect of the boundary of the domain. Then, the Sibson-type CNEM shape functions are computed. Finally, the discretization of a generic variational formulation is defined by invoking an "stabilized conforming nodal integration". In this work, we focus especially on the two last points. In order to compute the Sibson shape function, five algorithms are presented, analyzed and compared, two of them are developed. For the integration task, a discretization strategy is proposed to handle domains with strong non-convexities. These approaches are validated on some 3D benchmarks in elasticity under the hypothesis of small transformations. The obtained results are compared with analytical solutions and with finite elements results. Finally, the 3D CNEM is applied for addressing two forming processes: high speed shearing and machining.

Keywords: Meshless methods, Natural neighbour Galerkin methods, 3D constrained natural element method, Forming processes simulation

1. Introduction

The "meshless" methods are alternative approaches to the finite elements method to solve partial differential equations (PDE). As finite elements, the meshless methods allow to approximate fields with values stored on points, called nodes, distributed on the studied domain. With finite elements the approximation is defined element by element, every element being defined by all the nodes connected to it. Generally, for a same set of nodes several meshes are possible. That leads the approximation to a mesh dependence. With meshless methods the connectivity between nodes is only defined by the notions of neighborhood or domains of influence. For theses approaches, the construction of the approximation is thus not dependent any more on an any meshing but only on the relative position of the nodes. An important motivation of the meshless approaches is to avoid numerical problems resulting from the mesh, for example mesh distortion when studied domain is subjected to large strains. The meshless methods can be used advantageously for the numerical simulation of manufacturing processes like extrusion, injection, machining or blanking for example.

The finite difference method generalized for arbitrary cloud of nodes [1] was the first step toward the meshless methods. Other methods were then proposed: the "Smooth Particle Hydrodynamics" (SPH) [2] (an overview of this approach is given in [3]), the "Diffuse Element Method" (DEM) [4] [5] [6], the "Element Free Galerkin Method" (EFGM) [7], the "Reproducing Kernel Particle Method" (RKPM) [8], the "Natural Element Method" (NEM) [9], and the "Method of Finite Spheres" [10].

The NEM is based on the natural neighbor (NN) interpolant introduced by Sibson [11]. The NN interpolant uses a geometric construction: the Voronoï diagram associated with the cloud of nodes spread on the studied domain. Unlike the other meshless approaches: i) the support of NN shape functions are automatically defined, ii) the value of NN shape functions associated with internal nodes are null on the border of the domain. This last property, only valid for convex domains, is particularly interesting because it allows a direct imposition of the boundary conditions, exactly as in the finite elements framework. To extend these property to *non-convex* domains two strategies exist. The first one, based on the alpha forms [12], propose a very simple way to describe the border of the domain if this last one remains weakly non-convex. The second one, the Constrained Natural Element Method (CNEM) [13], handles the case of strongly non-convex do-

mains by building the interpolation on the Constrained Voronoï diagram, constrained by the border of the domain. The major inconvenience of NN shape functions is their significant calculation time.

This article has two main objectives. First of all it aims at clearing up certain properties inherent to the Sibson's shape functions in the $3d$ for possibly non-convex domains. Secondly it gives a presentation and a comparison, in term of performance and applicability, also for non-convex $3d$ domains, of different needed algorithms. This paper gives also the headlines of the implementation of the CNEM in a finite transformations context and shows results for two kinds of cutting manufacturing processes. The modelling of metal cutting, well known as being particularly complex [14], gives interesting applications for the proposed approach.

2. Overview of the constrained natural neighbor (CNN) interpolation

The natural neighbor interpolant is based on a geometrical construction, the Voronoï diagram. This Voronoï diagram is associated with a cloud of nodes distributed on the studied domain. The nodes are points of the domain where the degrees of freedom (DOF) of the interpolation are defined. The NN interpolation is a linear combination of NN shape functions, one for each node. The NN shape functions are local in space and interpolating (the value of the interpolation on a node is equal to the DOF of the node).

The Voronoï diagram is the dual of the Delaunay tessellation associated with the same cloud of nodes. However, unlike to shape functions whose construction are based on a tessellation, the NN shape functions only depend on the space distribution of the nodes. Figure 1 illustrates this aspect for a fixed nodal distribution and at a given point x. In this figure, the value of the shape function associated with each node is proportional to the diameter of the red disk. For the NN shape functions, the neighbors of x *(nodes of the domain for whom the associated shape functions are not equal to zero)* are the closest nodes to x, nodes j, h, k, and g. For this interpolation, the more a node is close to x, the more his influence (the associated value of the shape function) is important. We can note that the value of the shape functions associated with the neighbors of x are almost identical, because, for this example, x is roughly at the same distance from them. This is not the same for the finite elements linear shape functions. For a Delaunay tessellation represented in Figure1 (on the right), the neighbors of x are only j, g and

k (for k the associated shape function value is practically zero). For this tesselation h is not a neighbor of x although h being at a similar distance from x.

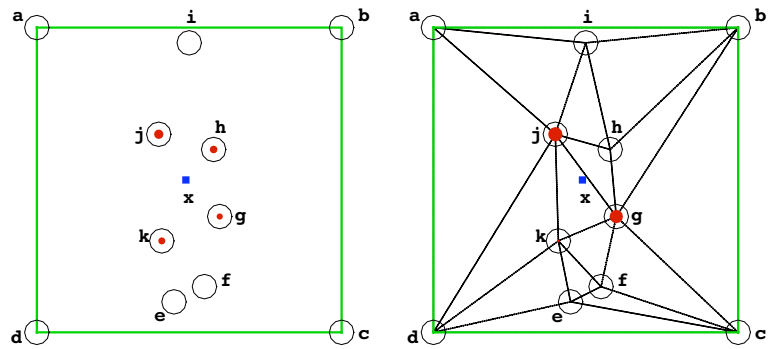


Figure 1: Values of neighbors Sibson NN shape functions *(left)* - Finite elements linear shape functions*(right)*

Compared with other meshless shape functions, the NN shape functions have a significant advantage. On the domain border, the value of the interpolated field by NN shape functions depends only on the value of this field on border node, and linearly (figure 2). In other words, the internal nodes have no influence on border of the domain, the associated shape functions vanished on the border. This property is only valid for convex domain.

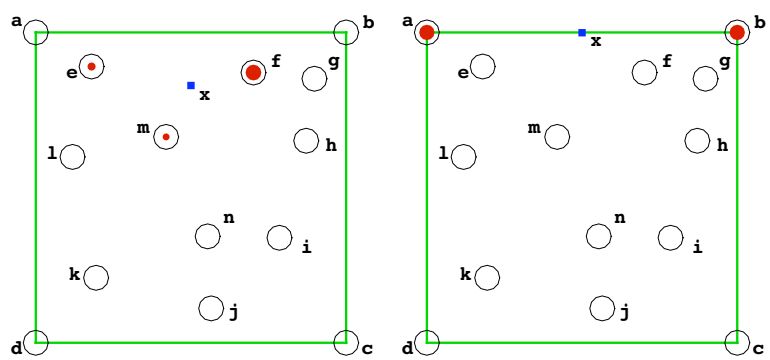


Figure 2: Values of neighbors Sibson NN shape functions close to convex border

For a **non-convex domain**, on or close to a **non-convex border**, the interpolated field may not only depend on internal nodes but also on nodes being located on the other side of this border (figure 3 on the left). The

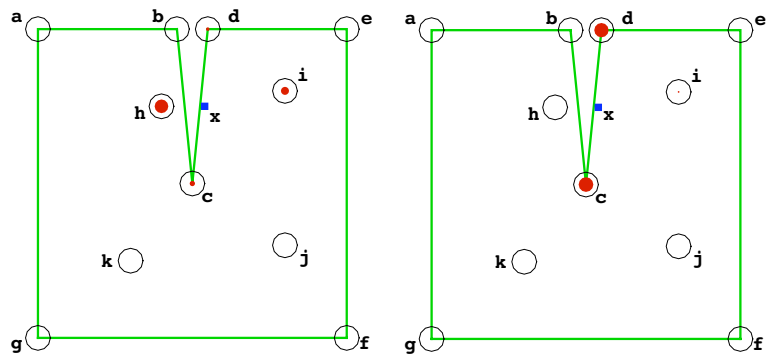


Figure 3: Sibson shape function close to a **non-convex** border: NN *(left)* / CNN *(right)*

constrained natural neighbor shape functions **CNN** [15] were introduced to cure this gap. The CNN shape functions are an extension of the NN shape functions for a **non-convex domain**: the calculation of NN shape functions is based on the *Voronoï* diagram, the calculation of CNN shape functions is based on the *Constrained Voronoï* diagram. The constrained Voronoï diagram is contrained by the border of the domain. Figure 3 illustrates the contribution of the CNN shape functions close to a non-convex border (on the left) comparatively to the NN shape functions (on the right).

3. Computation of CNN shape functions

In this section, for convenience reasons, we often refer to the two-dimensional case ($n = 2$), followed in italic by a generalization to the three-dimensional case ($n = 3$) or more ($n > 3$). We consider a domain Ω from R^n the border of which is denoted Γ. We also consider a set, S, of nodes spread over Ω and its border Γ.

3.1. Voronoï diagram and Delaunay tessellation

Let p be a node of S. The Voronoï domain [16] associated with p, noted $Vor_S(p)$, is the set of points of R^n closer to p than any others node q of S or equal distance:

$$Vor_S(p) = \{x \in R^n : d(x,p) \leq d(x,q), \forall Q \in S, q \neq p\} \tag{1}$$

where $d(.,.)$ is the Euclidean distance between two points. The Voronoï diagram is the dual of the Delaunay tessellation [17]. The circumcircle *(circumsphere, circum-hyper-sphere)* of each triangle *(tetrahedron, simplex)* does not contain any node inside itself.

Let us define some vocabulary:

- Voronoï vertex: center of the circle *(sphere, hyper-sphere)* which circumscribes a *Delaunay* triangle *(tetrahedron, simplex)*. This vertex is thus associated with three generator nodes *((n+1) in nd)*, it is in connection with three other vertices *((n+1) in nd)*, those associated with adjacent *Delaunay* triangles *(tetrahedron, simplex)*.

- Voronoï edge: segment connecting two adjacent vertices. Each edge is associated with two geneartor nodes *(n in nd)*

- Voronoï face: median plan *(hyper-plan)* between two neighboring nodes *(generator nodes)*.

- Voronoï cell: area of Voronoï.

3.2. Constrained Voronoï diagram and constrained Delaunay tessellation

The definition of constrained Voronoï diagram is based on the visibility criterion [18]: a node q is known as visible from another node p, and reciprocally, if, and only if, the segment $[p, q]$ do not pass through the domain border Γ of Ω, and is not outside this last.

In Figure 4, the segments $[a, b]$, $[a, c]$, and $[a, d]$ pass through Γ. The nodes b, c, and d are thus not visible from a and reciprocally. In addition, the segment $[e, c]$ is outside the domain, the nodes e and c are thus not visible from each other.

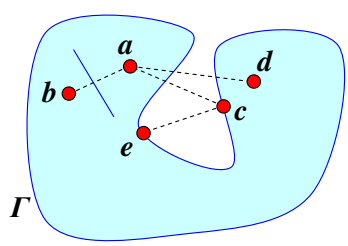


Figure 4: The visibility criterion

The definition of the constrained Voronoï area associated with node p, $Vor_S^{c\Gamma}(p)$, found classically in the literature [19], is as follow:

$$Vor_S^{c\Gamma}(p) = \{x \in R^n : d(x,p) \leq d(x,q), \forall q \in S, q \neq p,\ x \text{ visible from } p \text{ and } q\} \tag{2}$$

This definition restricts $Vor_S^{c\Gamma}(p)$ to its internal domain part. In order to be able to compute shape functions thereafter, this area should not be **truncated** by the domain border.

A more extended definition is:

$$Vor_S^{c\Gamma}(p) = \{x \in R^n : d(x,p) \leq d(x,q), \forall q \in S,\ q \neq p \wedge p \text{ visible from } q\} \tag{3}$$

In Figure 5 is represented, on the left, the (no constrained) Voronoï cell associated with node p. On the right-hand side drawings, two constrained

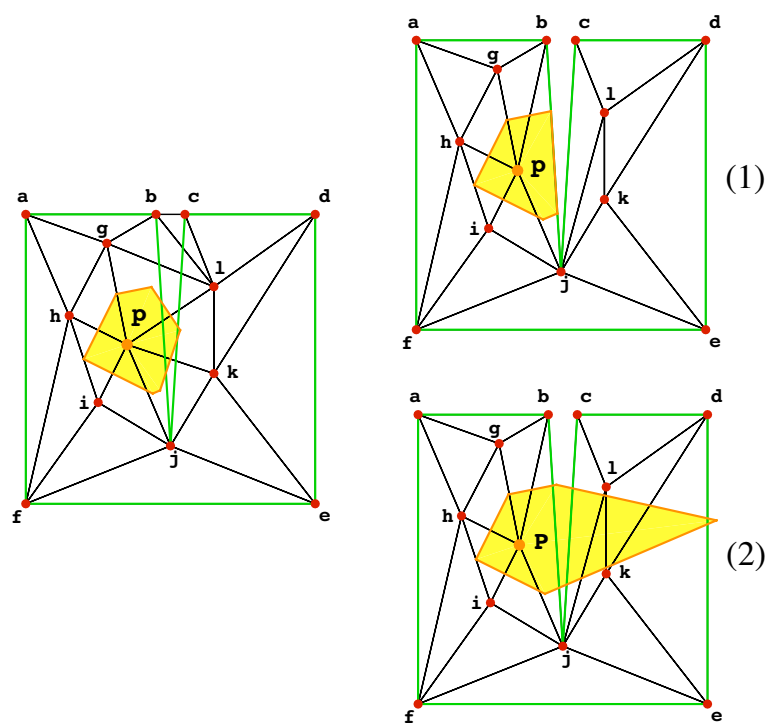


Figure 5: Voronoï cell of a node p, no-constrained *(left)*, constrained *(right)*

cells associated with the same node p are present: on the top the cell resulting form the definition (2), on the bottom the cell resulting from the definition

(3). The border of the domain is defined by the green segments. In the definition (3), the visibility criterion is applied only to **nodes** of the domain whereas in the definition (2) the visibility criterion is applied to all the **points** of the domain.

It must be noticed that the Voronoï diagram of a cloud of nodes, is the constrained Voronoï diagram, constrained by the **convex hull** of this cloud of nodes. For a non-convex domain, the convex hull of the cloud is not anymore its border. This explains the difference between the two diagrams, constrained and not constrained, for non-convex domains.

The constrained Voronoï diagram (CVD) is the dual of the constrained Delaunay tessellation (CDT)[20]. For this tessellation the circum circle *(sphere, hyper-sphere)* of each triangle *(tetrahedron, simplex)* does not contain any **visible** node from the nodes of this triangle. In $2d$ the constrained Delaunay tessellation always exists, for convex domain or not. This is unfortunately not true anymore in $3d$ for certain non-convex domains. An example is given with the Schönhardt polyhedron [21] (figure 6). A solution in these cases consists in adding some nodes on the border of the domain.

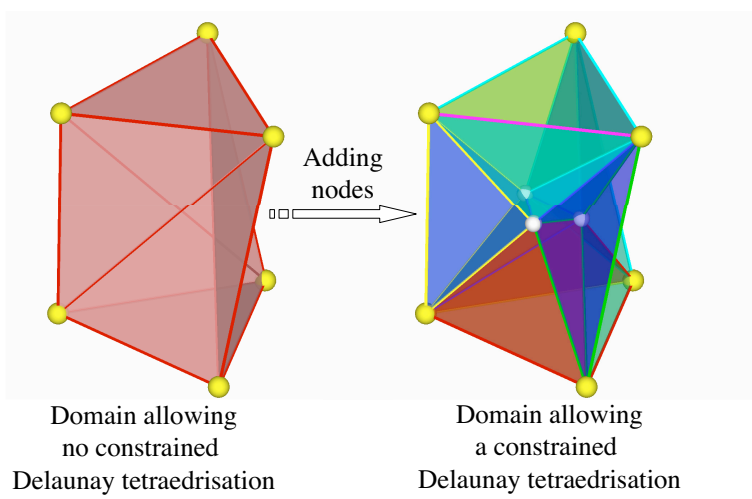


Figure 6: Schönhardt polyhedron

In our implementation, the construction of the CVD is done through its dual, the CDT. To build this CDT we use the algorithm proposed by Si and Gärtner [22] and implemented in TETGEN code [23] by the author. The input data of this algorithm are the nodes of the domain and the border of the domain described in the form of PLC *(Piecewise Linear Complex)*[24].

In our case, for a $3d$ domain, the tessellation of the border is a very simple PLC where all the faces are triangles.

3.3. Definition of the Sibson CNN shape functions

Let x be a point inside Ω, the point where we want to evaluate the interpolation, such as $x \notin S \wedge x \notin \Gamma$. We denote V_S^Γ the constrained Voronoï diagram of Ω, V_{S+x}^Γ the constrained Voronoï diagram of Ω with addition of x as node and D_{S+x}^Γ the corresponding constrained Delaunay tessellation, $c_x^{'}$ the Voronoï cell associated with x in V_{S+x}^Γ.

The constrained natural neighbors of x are the nodes of Ω whose Voronoï cell are directly close to the cell of x in V_{S+x}^Γ, in other words, the nodes of Ω connected to x by a Delaunay edge in D_{S+x}^Γ.

Let say that v_i is such a neighbor, and c_{v_i} its associated Voronoï cell in V_S^Γ. Similarly to the NN shape function [11][25][26], the value of Sibson CNN shape function associated with v_i at point x, denoted $\phi_{v_i}^s(x)$, is given by:

$$\phi_{v_i}^s(x) = \frac{\left\| c_x^{'} \cap c_{v_i} \right\|_L}{\left\| c_x^{'} \right\|_L}, \tag{4}$$

where $\|.\|_L$ is the algebraic Lebesgue measure *(length in 1d, area in 2d, volume in 3d...)*. For CNN shape functions, unlike NN shape functions, this measure can be negative or equal to zero for strongly non-convex $3d$ domains. Hence, for these domains, there are points where shape functions are non-definite. We return on this aspect in Appendix A.

A diagram illustrating the computation of Sibson shape functions in $3d$ is given Figure 7. In Figure 8 we can see the difference resulting from the use of constrained Voronoï diagram versus Voronoï diagram on a Sibson shape function *(2d)* (for an internal node near to non-convex border*(top)*, border node *(bottom)*).

3.4. Properties of CNN shape functions

a) Delta Kronecker property: like NN shape functions, the CNN shape functions are interpolating:

$$\phi_{n_i}(n_j) = \delta_{ij}, \ \forall (n_i, n_j) \in S^2 \tag{5}$$

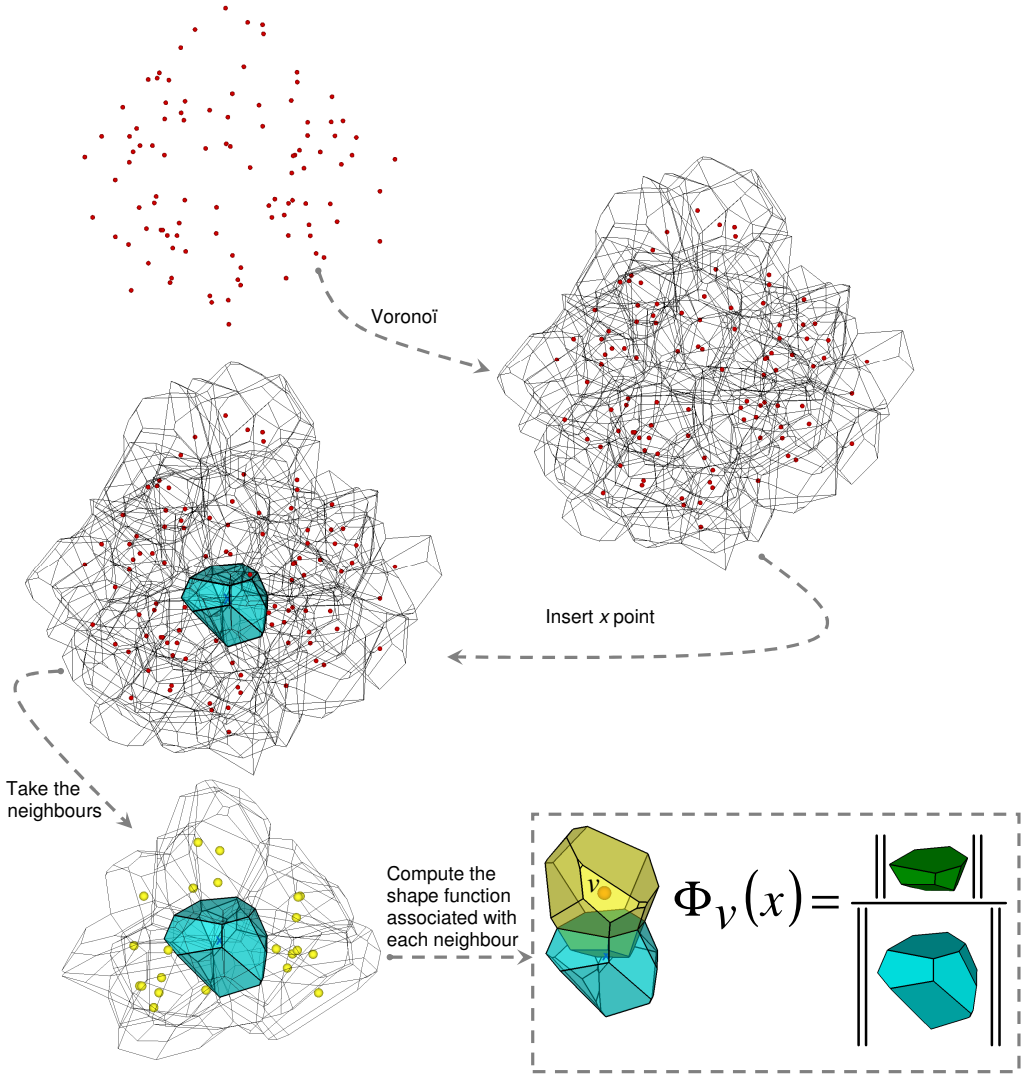


Figure 7: Computation of Sibson shape fonctions in 3d

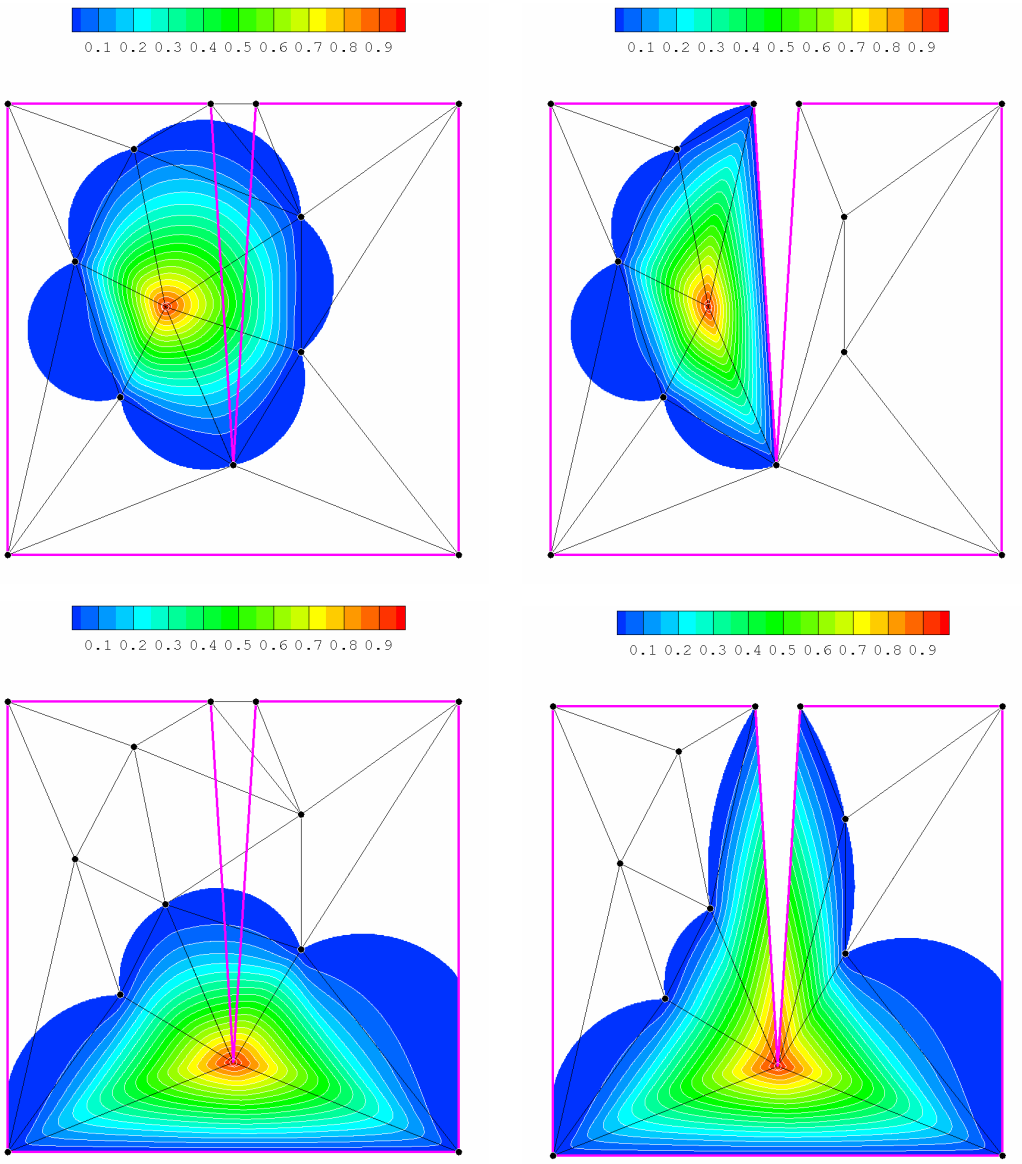


Figure 8: Contour of Sibson shape functions: NN *(left)* versus CNN *(right)*

b) Partition of unity: by construction CNN shape functions check the partition of unity:

$$\sum_{i=1}^{n(x)} \phi_{v_i}(x) = 1, \ \forall \ x \in \Omega \tag{6}$$

where *n(x)* is the number of neighbors of x.

c) Local coordinate property and linear consistency: Sibson[11][25][26] and Piper[27] showed that NN shape function reproduce exactly coordinates of this point, and by extention CNN also verifies:

$$x = \sum_{i=1}^{n(x)} \phi_{v_i}(x) \, x_i \tag{7}$$

Properties *b)* and *c)* imply that CNN shape functions reproduce exactly the linear functions.

d) Linearity on the border: on Γ, the CNN shape functions are strictly linear [15]. For recall, this border is: the convex hull of the nodes of Ω if we uses a Voronoï diagram, the real border of Ω, if we uses a **constrained** Voronoï diagram.

Properties *a)* and *d)* imply that on a segment *(triangle in 3d)* of the border, the CNN interpolation depends only on the DOF of the nodes of this segment *(triangle)*.

e) Support of CNN shape functions: the support of a CNN shape function associated with a given node i of Ω, is the set of the circles *(sphere, hyper-sphere)* circumscribed with the Delaunay triangles *(tetrahedron, simplexes)* connected to i. This support is truncated by Γ (see Figure 8, right).

f) Continuity: Sibson NN or CNN shape functions are C^0 on the nodes, C^1 on the Delaunay circles *(sphere, hyper-sphere)*, and C^∞ everywhere else [28].

3.5. Commputation of Sibson CNN shape functions

We denote by P^{int} the polyhedron defining the common domain of $c_x^{'}$ and c_v ($P^{int} = c_x^{'} \cap c_v$), v being one of the neighbors of x in V_{S+x}^{Γ}. The computation of $\phi_v^s(x)$ is made in two steps: the first one is the local modification of V_S^{Γ} by the insertion of the point x, the second one concerns the calcultation of $\|P^{int}\|_L$.

For the *first step* the use of an incremental algorithm is very well adapted. Two similar algorithms can be used [29]: the Bowyer algorithm [30], which works with Voronoï vertices, or the Watson algorithm [31] which works with Delaunay tetrahedrons. These two algorithms, initially developed for the construction of VD/DT, can be used in a incremental way on an existing CVD/CDT. Both of them maintain the constrained character of the initial diagram V_S^Γ or Delaunay tesselation D_S^Γ. The firts step can be split in three main sub-steps:

1. Search for the tetrahedron which contains the point x. For that it is possible to use a gradient descent based on Voronoï graph *(Schmitt and Borouchaki [32])*, or that based on the Delaunay graph *(Green and Sibson [33])*. In the case of non-convex domains this two search can fail if search starting from a no visible tetrahedron by the point x. It is thus necessary to implement other types of search, such frontal search, or octree search.
2. Search for all the tetrahedrons whose circumscribed sphere contains the point x and break them. This search is local, it is done by visiting the nearest tetrahedrons to the first one.
3. Remesh obtained cavity by connecting each cavity triangle to the x.

The *second step* can be done in different ways:

a) By Lasserre recursive algorithm

Alfaro et al. [34], like Sambridge et al. [35][36], propose to use the recursive Lasserre algorithm. The Lasserre algorithm can be used to compute the volume of convex polyhedrons described in a H-representation way (volumes delimited by a set of half-planes).

In the case of convex domain, the Voronoï cells are convex and, as being the intersection of two convex polyhedrons, P^{int} is also convex. P^{int} can then be described as a volume delimited by a set of half-planes. This set consists of all the faces of c_v (cell of v in V_S^Γ), and the common face to $c_v^{'}$ (cell of v in V_{S+x}^Γ) and $c_x^{'}$ (see Figure 9). The Lasserre recursive algorithm can therefore be used to compute the volume of P^{int}. This algorithm can be succinctly described as follow:

Let P a convex polyhedron in R^n, delimited by m half-planes. Each one of these half-planes can be represented by an inequation, which can be translated in a matrix form by the inequation:

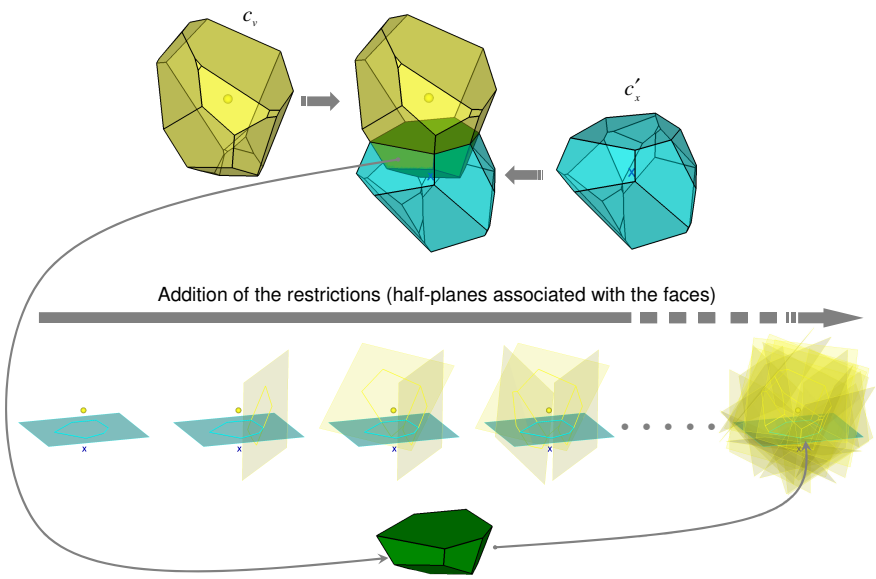


Figure 9: Computation of $\left\|P^{int}\right\|_L$ with Lasserre algorithm

$$\underline{\underline{A}} \cdot \underline{x} \leq \underline{b}, \tag{8}$$

where $\underline{\underline{A}}$ is an $m \times n$ matrix, $\underline{x}$ the column of the n components of the point x, and $\underline{b}$ a column of m terms.
The volume measurement of P, denoted $V(n, \underline{\underline{A}}, \underline{b})$, can be calculated by the following recursive formula:

$$V(n, \underline{\underline{A}}, \underline{b}) = \frac{1}{n} \sum_{i=1}^{m} \frac{b_i}{\|\underline{a}_i\|} V_i(n-1, \underline{\underline{A}}, \underline{b}), \tag{9}$$

where $V_i(n-1, \underline{\underline{A}}, \underline{b})$ is the volume measurement in R^{n-1} of the face f_i:

$$f_i = \left\{x |\ \underline{a}_i^T \cdot \underline{x} = b_i,\ \underline{\underline{A}} \cdot \underline{x} \leq \underline{b}\right\}.$$

b) By complementary volume

The Volume measurement of P^{int} can also be defined as the measurement of volume variation of its associated Voronoï cell after the insertion of the

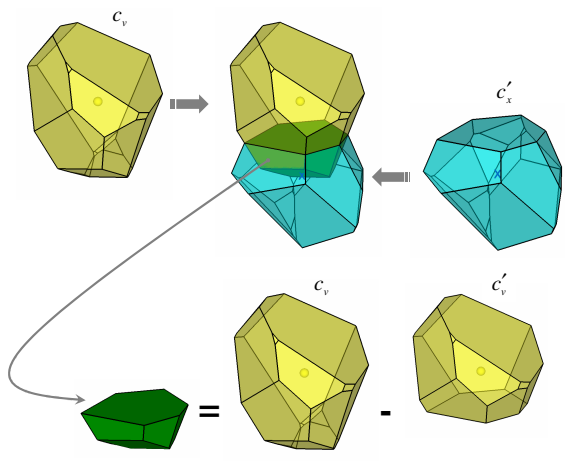


Figure 10: Computation of $\|P^{int}\|_L$ with complementary volume

point x (see Figure 10).
This approach can be applied only if the cell of each neighbour node is non-infinite. If it is not the case, it is impossible to calculate the measurement of its volume, but considering that we need only for the measurement of the variation of volumes of the cells after insertion of the point x, and not of their volume, this problem can be solve by truncating the infinite cells (addition of faces).

c) and d) By topological approaches

It is possible to rebuild the topology of P^{int} polyhedron (faces, edges, vertices) with only *c)* CVD connectivity [37] or *d)* CDT connectivity [38]. No computations of geometrical intersections are needed. In both of these approaches the faces of the P^{int} polyhedron are classified into three categories (see Figure 11):
- $f_i c_x^{'}$, the face of $c_x^{'}$ cell having as second node v,
- F_{trq}, the set of c_v cell faces truncated by the edges of $f_i c_x^{'}$,
- and possibly $F_{\overline{trq}}$, the set of c_v cell faces being completely inside $c_x^{'}$.

e) By Watson algorithm

Watson [39] proposes to use the Lawrence polyhedron decomposition [40] for the computation of $\|P^{int}\|_L$. Taking a polyhedron P, and an hyperplane

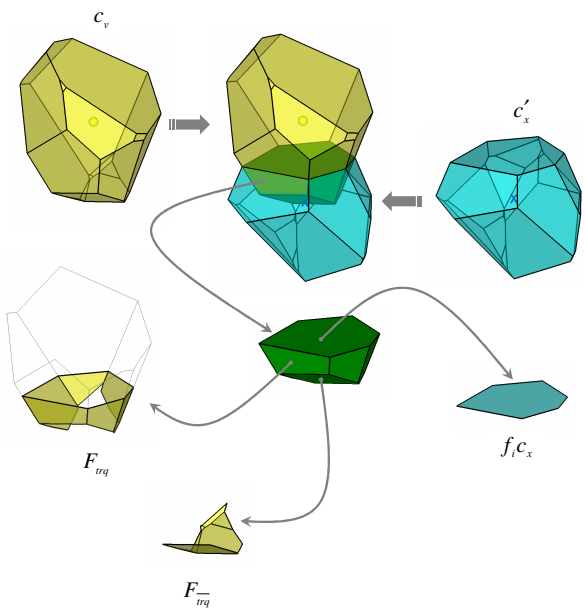


Figure 11: Topological decomposition of the P^{int} polyhedron

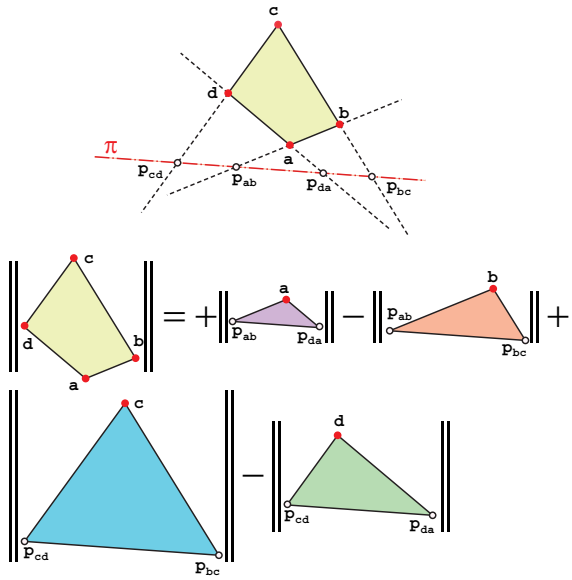


Figure 12: Lawrence decomposition in $2d$

π which is not parallel to any face of P, its volume measurement can then be calculated by summing <u>*algebraically*</u> the volume measures of the simplexes spx_s^{Law}. spx_s^{Law} are build for each vertex s of P by taking in addition to the hyperplane π, the hyperplanes associated with the faces of P having s as vertex. Each one of these hyperplanes divides space into two signed parts, the positive one includes P. The sign of the volume measurement of a simplex spx_s^{Law} is the sign of the part of the space in which it is located, which is the product of the signs of this space part compared to each hyperplane associated with spx_s^{Law} *(the hyperplane π is not considered)*. In Figure 12, we can see an example of the Lawrence polyhedron decomposition in $2d$. Watson takes, for his algorithm, the π plane as the median plane between the point x and v. An illustration of this is given in Figure 13.

The Watson algorithm is not applicable on evaluation point x being located on a face of any Delaunay tetrahedron. If the calculation of the interpolation is needed at such a point an other algorithm must be used.

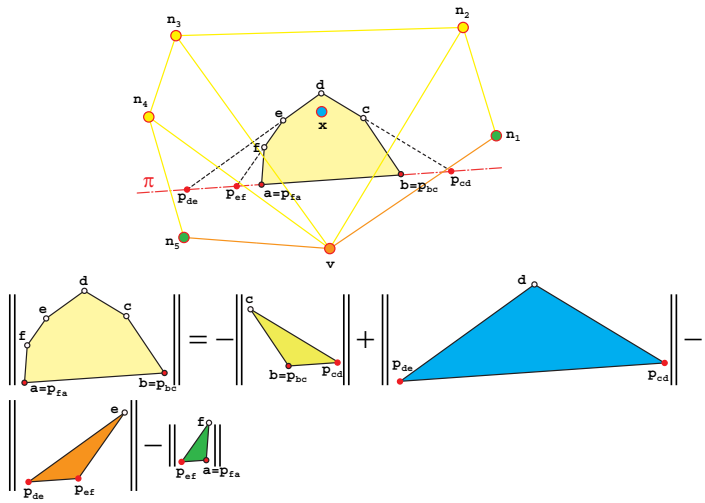


Figure 13: Computation of $\left\|P^{int}\right\|_L$ with Watson algorithm in $2d$

3.6. Efficiency comparison of the five proposed algorithms

The five methods of Sibson shape function calculation were implemented (C, C++ languages). For the method based on the Lasserre recursive algorithm, we use the Vinci code [41], in which an improved version of this algorithm is implemented, the r-Las algorithm [42].

For the comparison, we takes a unit cube of dimension (1×1×1) with 10 000 random distributed nodes inside. The shape functions are evaluated at the barycentre of each Delaunay tetrahedron. Only the points for which the five methods are applicable are taken into account.

Given times (in table 1) are average times corresponding to 1000 evaluations. The total computing time includes:
-T_{cav}: time taken for the cavity construction (identical for the five algorithms).
-T_{sf}: needed time for shape functions computation once the cavity is built.

The Watson algorithm is clearly the more powerful. For points where this algorithm doesn't work (points located on or close to a Delaunay face), the topological algorithm based on the CDT is recommended. In practice the rate of failure of the Watson algorithm for random evaluation points is about $\frac{1}{1000}$.

For time measurements the characteristics of the used computer are:

Algorithm	Lasserre	Vol-comp	Topo-DVC	Topo-TDC	Watson
T_{cav} $(m.s)$	35				
$T_{sf}(m.s)$	4136	544	413	192	51
$\frac{T_{sf}\ Alg}{T_{sf}(Watson)}$	82	11	8	4	1

Table 1: Comparison of the computation efficiency of $\left\|P^{int}\right\|_L$

- OS: microsoft windows xp,
- Compiler: Microsoft (R) Incremental Linker Version 7.10.3077,
- Processor: Mobile Intel Pentium M 745 - 1800 MHz,
- RAM: PC2700 DDR SDRAM.

4. Numerical integration of CNN shape functions for the CNEM

Numerical integration over Ω are needed in order to solve mechanical problems. In the NEM, and thus by extension in the CNEM, two numerical integrations are generally used: the Gauss integration, performed on the Delaunay tetrahedrons, and the stabilized conforming nodal integration (SCNI). As CNN shape functions are rational, the first one requires a high number of Gauss points in order to minimize the integration error. To improve this direct nodal integration, Chen et al. [43][44] proposed the SCNI approach for the RK interpolation in the RKPM method [45]. The SCNI approach was taken again by Gonzàlez et al. [46] for the NEM method.

In the SCNI approach a strain smoothing stabilization is performed to stabilize the nodal integration. The SCNI is based on the assumed strain method in which an average gradient (called stabilized gradient) is introduced for each node n_i:

$$\tilde{\nabla} A_i = \frac{1}{\left\|\Omega_i\right\|_L} \int_{\Omega_i} \nabla A \; d\Omega_i, \tag{10}$$

where A is field interpolated over Ω and Ω_i a sub-domain surrounding n_i. Using the divergence theorem, (10) can be rewrite as:

$$\tilde{\nabla} A_i = \frac{1}{\left\|\Omega_i\right\|_L} \int_{\Gamma_i} A \; \vec{n} \; d\Gamma_i, \tag{11}$$

where Γ_i is the border of Ω_i and $\vec{n}$ the outward normal on Γ_i. This allows to pass from a volume integral to a surface integral, and to integrate directly the field rather than its gradient.

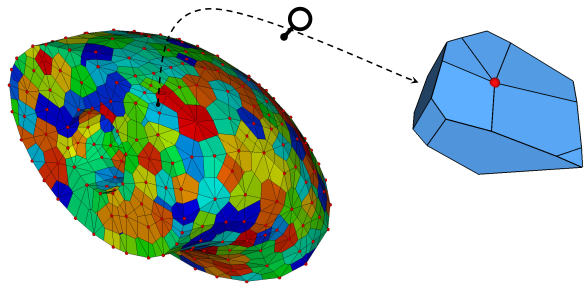


Figure 14: Domain partition for slightly non-convex domains

In a large transformations context, SCNI avoid fields projection between two configurations during a reactualization, all variables *(displacement, strain, stress)* being associated with the nodes (colocation approach).

For the integration of the weakforms it is necessary that the set of Ω_i forms a partition of Ω. In the case of convex, or slightly non-convex domain, the use of the Voronoï cells is a "natural" partition of the domain as they remain convex at interior domain (Figure 14). It is nevertheless necessary to truncate the cells which overflow domain border. This is the case of cells whose nodes are on the border and for some nodes close to the border. A fast and robust algorithm is presented in [37] to do this operation.

For strongly non-convex domains, the partition based on nodal Voronoï cells is no longer valid. For such domains, cells associated with border nodes, or close to it, can be non-convex *inside* the domain.

For the non-convex domain depicted in Figure 15 the phenomenon occurs at node n. The cell associated with n (visible in Figure 16) is non-convex inside Ω (the surrounded point with a circle belongs to the auto-intersection curve of the cell associated with node n). This problem can be avoided by using another nodal partition. The most simplest consists in subdividing each Delaunay tetrahedron in four parts. The set of all the parts of tetrahedron attached to a node n_i defines the sub-domain $\tilde{\Omega}_i$ associated with n_i (Figure 17). The set of $\tilde{\Omega}_i$ is a partition of Ω.

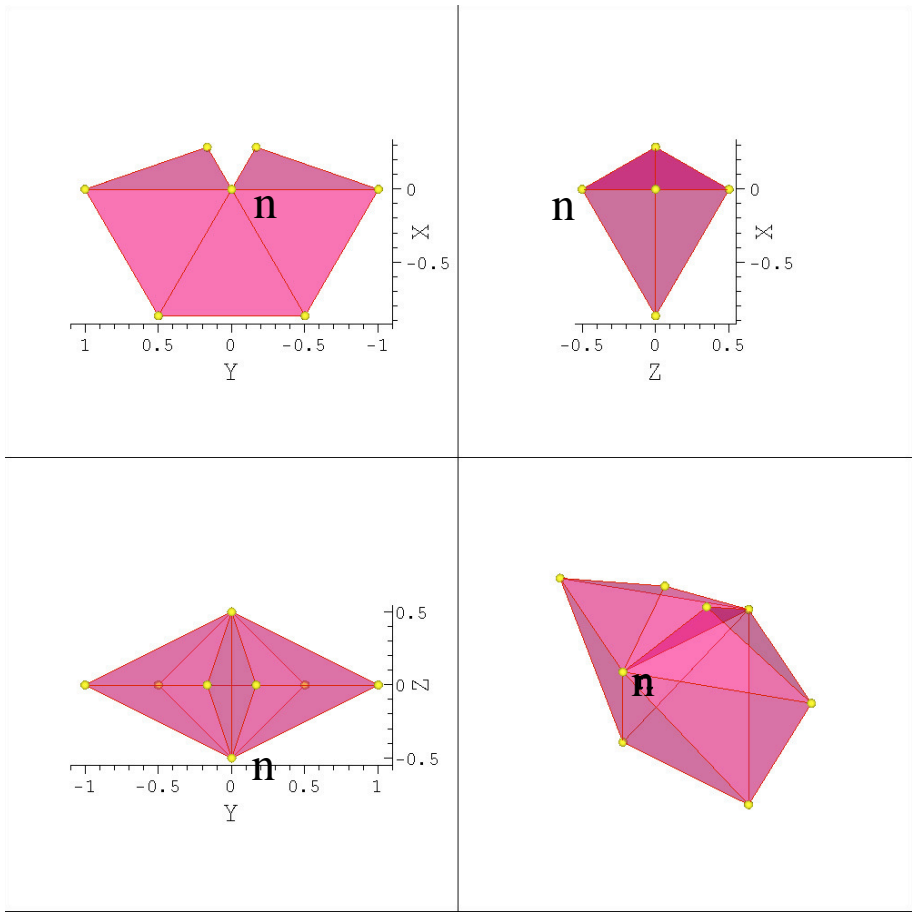


Figure 15: Example of non-convex domain

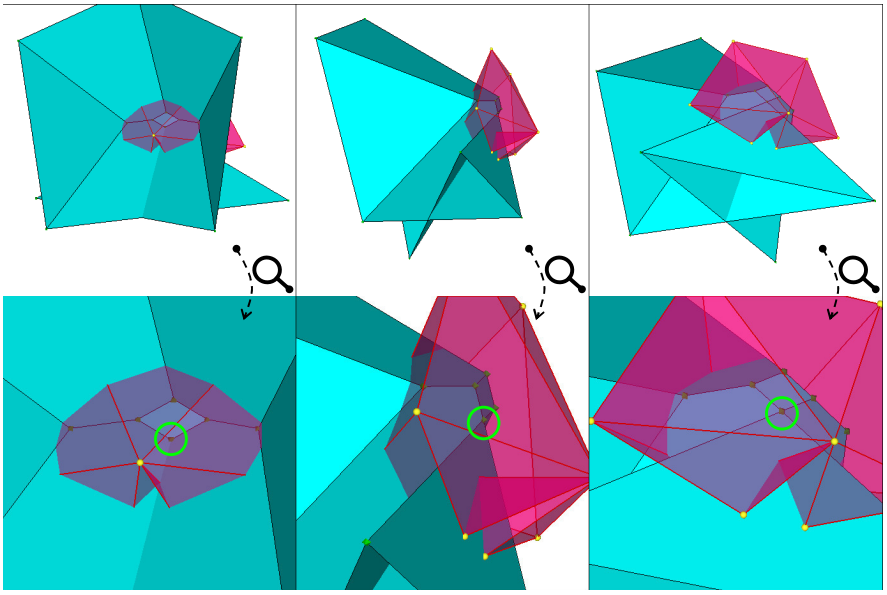


Figure 16: Example of a Voronoï cell having a non-convexity inside the domain

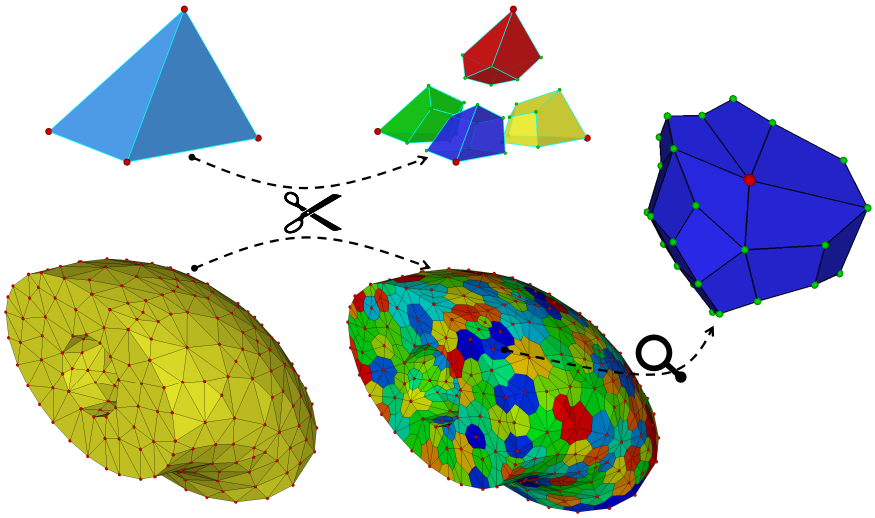


Figure 17: Domain partition for non-convex domains

5. 3d elastic Benchmarks

In order to validate the CNEM in 3d, we carried out simulations for two standard problems in isotropic elasticity (small displacement and small strain): the hollow sphere under pressure and the cylindrical excavation in an infinite medium subjected to constraints ad infinitum. An analytical solution exists for both of them.

We do not give here a description of the implementation of the CNEM in elasticity but the reader car refer to Yvonnet et al. [13].

When time calculation are examinated, we use the sparse iterative solver MKL [47] for the resolution of the generated linear system. The characteristics of the used computer are the following:
- Operating System: Linux RedHat 64 bits,
- Compiler: Intel c++ compiler 9.1,
- Processor: Intel Itanium 2 Monticito - 1,6GHz - 1 core used,
- RAM: 32G - DDR2 - 667MHz.

5.1. Hollow sphere under internal pressure

The exact solution of the problem (*stresses*) in spherical coordinates (r, θ, ϕ), is given by:

$$\sigma_{rr} = -\frac{{R_{int}}^3}{{R_{ext}}^3 - {R_{int}}^3}\left(\frac{{R_{ext}}^3}{r^3} - 1\right)P \tag{12}$$

$$\sigma_{\theta\theta} = \sigma_{\phi\phi} = -\frac{{R_{int}}^3}{{R_{ext}}^3 - {R_{int}}^3}\left(\frac{{R_{ext}}^3}{2r^3} + 1\right)P \tag{13}$$

$$\sigma_{r\theta} = \sigma_{r\phi} = \sigma_{\theta\phi} = 0 \tag{14}$$

where P is the pressure inside the sphere, R_{int} is the internal sphere radius and R_{ext} is the external sphere radius.

For simulations, the pressure inside the sphere is taken equal to 1 MPa and due to the symmetry of the problem only the eighth of the sphere is considered (Figure 18). Figure 19 shows the distribution of Von Mises equivalent stress (CNEM simulation with Sibson SCNI integration for 300 000 nodes). In this Figure, the equivalent stress field is not smoothed. The stress is constant on each node cell.

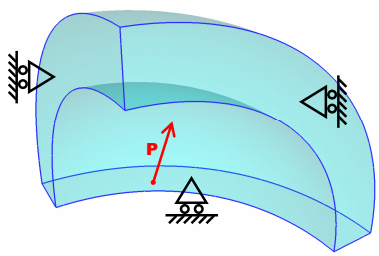


Figure 18: Hollow sphere under internal pressure – Boundary conditions

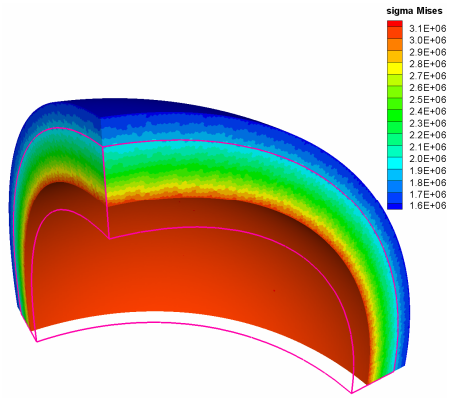


Figure 19: Hollow sphere – Von Mises equivalent stress (No. nodes $\approx$ 300,000, SCNI - Sibson shape function)

5.2. Cylindrical excavation in an infinite medium

This is a plane strain problem with a bi-axial load but we solve it in 3d. The exact solution of this problem is the following [48] (cylindrical coordinates (r,θ,z)):

$$\sigma_{rr} = \frac{\sigma_{1\infty} + \sigma_{2\infty}}{2}\left(1 - \left(\frac{a}{r}\right)^2\right) + \frac{\sigma_{1\infty} - \sigma_{2\infty}}{2}\left(1 - 4\left(\frac{a}{r}\right)^2 + 3\left(\frac{a}{r}\right)^4\right) \quad (15)$$

$$\sigma_{\theta\theta} = \frac{\sigma_{1\infty} + \sigma_{2\infty}}{2}\left(1 + \left(\frac{a}{r}\right)^2\right) - \frac{\sigma_{1\infty} - \sigma_{2\infty}}{2}\left(1 + 3\left(\frac{a}{r}\right)^4\right) cos\,(2\theta) \quad (16)$$

$$\sigma_{r\theta} = -\frac{\sigma_{1\infty} - \sigma_{2\infty}}{2}\left(1 + 2\left(\frac{a}{r}\right)^2 - 3\left(\frac{a}{r}\right)^4\right) sin\,(2\theta) \quad (17)$$

$$\sigma_{zz} = \nu\left(\sigma_{rr} + \sigma_{\theta\theta}\right) \quad (18)$$

$$\sigma_{rz} = \sigma_{\theta z} = 0 \quad (19)$$

For the stresses ad infinitum we take $\sigma_{1\infty} = 0$ and $\sigma_{2\infty} = 1\ MPa$. Only a quarter of the domain is studied (the problem has two symmetry plans). The boundary conditions are illustrated in Figure 20 and the calculated Von Mises equivalent stress field is depicted in Figure 21.

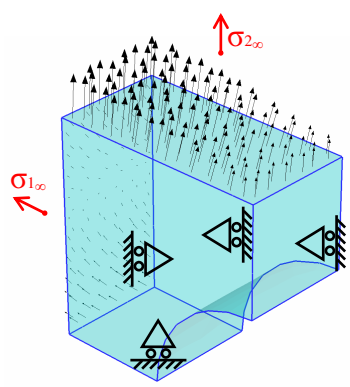


Figure 20: Cylindrical excavation – Boundary conditions

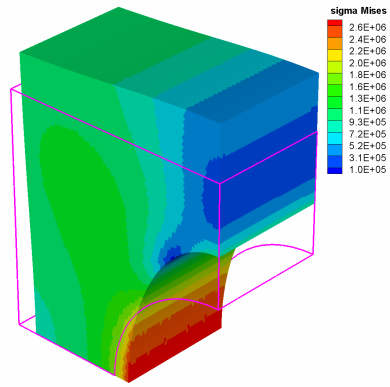


Figure 21: Cylindrical excavation – Von Mises equivalent stress (No. nodes $\approx$ 370 000, SCNI, Sibson shape function)

5.3. Convergence curves

For each of the two problems, and for a given homogeneous nodal distribution, following simulations were carried out:
- degree 0 Gauss integration with linear shape function on Delaunay mesh (Gauss FEM-L M-D),
- degree 0 Gauss integration with linear shape function on an optimized mesh (Gauss FEM-L M-O),
- degree 4 Gauss integration with Sibson shape function (Gauss Sibson),
- SCNI with linear shape function on Delaunay mesh (SCNI FEM-L M-D),
- SCNI with linear shape function on an optimized mesh (SCNI FEM-L M-O),
- SCNI with Laplace shape function (SCNI Laplace),
- SCNI with Sibson shape function (SCNI Sibson).

For SCNI, calculations are made on nodal volumes resulting from the quarter tetrahedron discretization and by taking one Gauss point integration per face (quadrangular).

Figure 22 and Figure 23 show the convergence curves of the error between computed and analytic solution for hollow sphere and cylindrical excavation respectively. The error is base on strain energy:

$$\|e\| = \sqrt{\frac{\int_\Omega (\sigma - \hat{\sigma}) : C^{-1} : (\sigma - \hat{\sigma}) d\Omega}{\int_\Omega \hat{\sigma} : C^{-1} : \hat{\sigma} d\Omega}}, \tag{20}$$

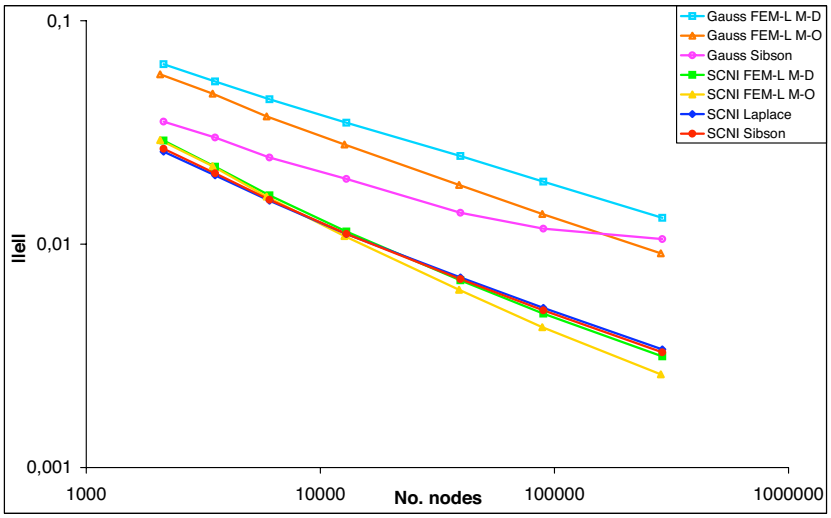


Figure 22: Hollow sphere – Convergence curves

where C is the elasticity tensor, σ is the computed stress tensor, and $\hat{\sigma}$ is the exact stress tensor.

For the two studied examples, although having the same rate of convergence than the linear finite elements, the CNEM divides the total error by 3. This profit of precision is mainly due to the use of the stabilized gradient. In fact, the stabilized gradient gives comparable results on linear interpolation and on CNN interpolation. We can also see that the rate of convergence obtained with the CNEM, with a Gauss integration over the Delaunay tetrahedrons, drop off when the number of nodes increase. This is due to the under integration of the Sibson shape function gradients.

Figure 24 gives, for the hollow sphere, and for the largest number of nodes ($\approx 300\,000$), the computation time t_p for stiffness matrix preparation [integration + assembly + boundary conditions imposition] and the computation time t_r for the resolution of the generated linear system. The relative error $\|e\|$ is also present in the same figure. It is interesting to note that t_p is approximatively five time greater for Sibson approximation than for finite element approximation. This is due to the calculation of shape functions, and more precisely due to the calculation of stabilized gradients.

The bandwidth of the stiffness matrix is larger for sibson function than for linear interpolation on tetraedron. This is due to the larger connectivity

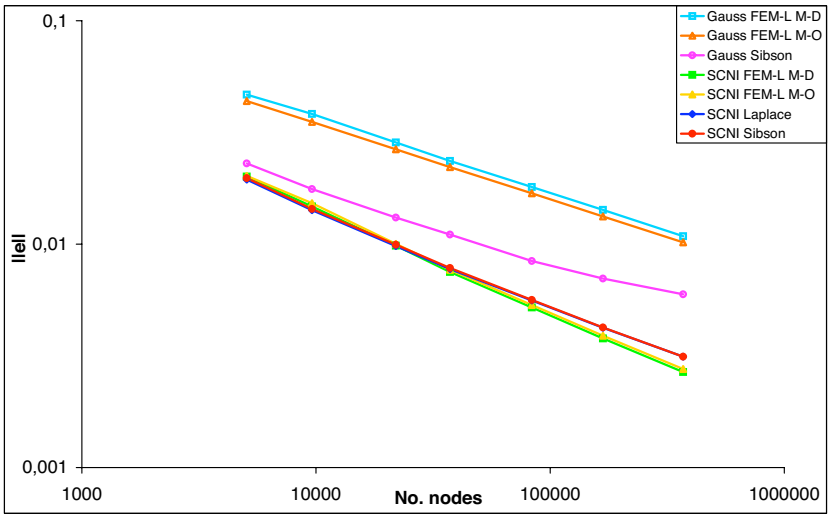


Figure 23: Cylindrical excavation – Convergence curves

of nodes with the CNEM. It is then not surprising that the resolution for CNEM takes larger time of resolution than FEM.

5.4. Study of the integration efficiency

As we saw in the section 4, the use of Voronoï cells based discretization is not always possible. We thus proposed to use the quarters of the tetrahedrons. In order to see whether the latter does not induce additional errors, the hollow sphere under pressure problem was treated with the two discretizations. The domain being slightly non-convex, the Voronoï cells based discretization is possible. Knowing that the Sibson shapes functions are rational, the number of integration point taken for the stabilized gradient computation can influence the error. In order to make sure that the error of integration is converged for each discretization, we have taken various numbers of integration points.

Following simulations were carried out:
a) For quarters of tetrahedron discretization based:
- 1 integration point for 3 faces,
- 1 integration point per faces *(barycenter)*,
- 1 Gauss integration point per triangle,
- 13 Gauss integration points per triangle *(integrand of order 7)*.

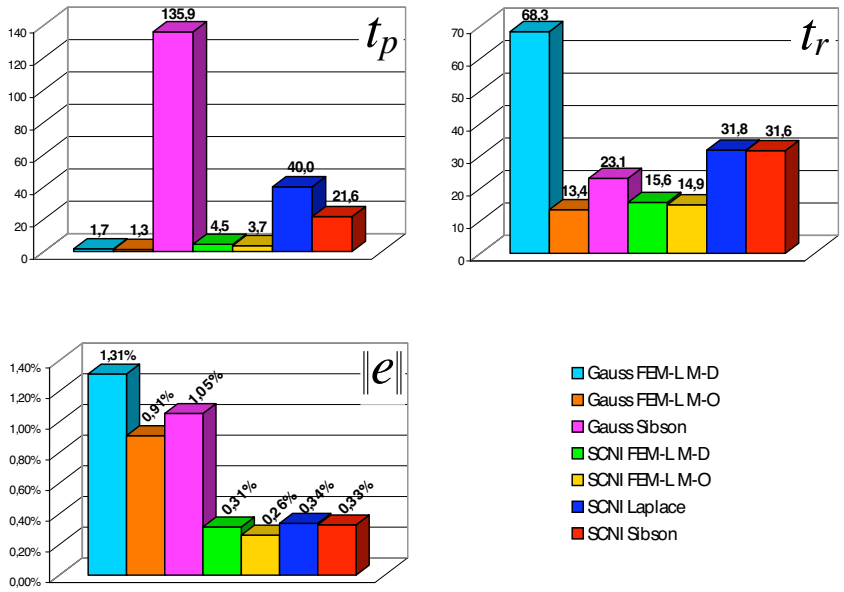


Figure 24: Hollow sphere – Computation time (in minutes) and error (No. nodes $\approx$ 300 000)

b) For Voronoï cells discretization based:
-1 integration point per faces *(barycenter)*,
-1 Gauss integration point per triangle,
-13 Gauss integration points per triangle *(26 per quadrangle)*.

The previous triangles are obtained by meshing the faces of elementary volumes attached to each node. These faces are quadrangles if we takes the quarters of the tetrahedron discretization based, and polygons if we takes the Voronoï cells discretization based. The study was done for Sibson shape function.

The obtained results are summarized in Figure 25. We can see that the quarters of tetrahedron discretization based does not induce more error than the Voronoï cells discretization based. Moreover, the error of integration with only one integration point per face is sufficient (identic error with that obtained with 26 points per face). Hence, we have chosen this integration for all CNEM simulations.

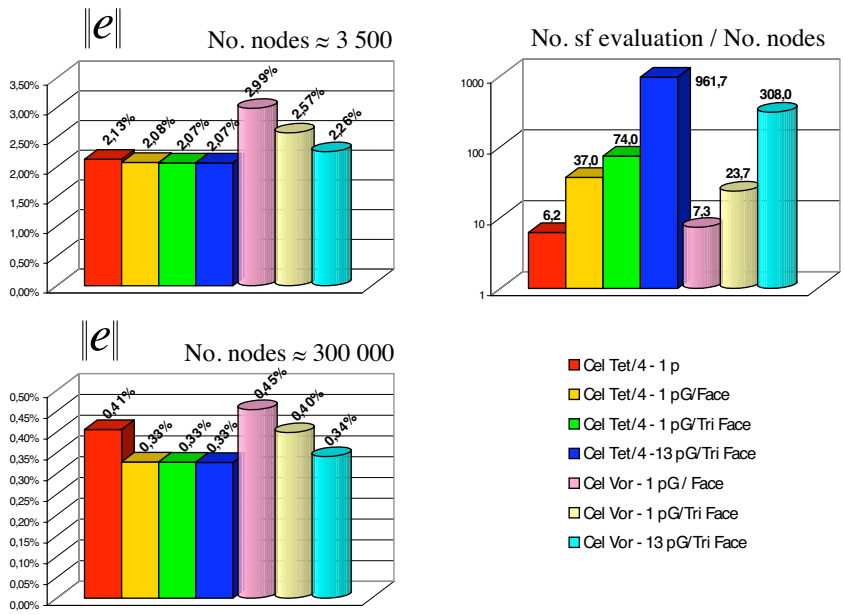


Figure 25: Integration efficiency

6. CNEM in finite transformations

In this section, we interest us in the CNEM in the context of large strains nonlinear elastoplastic, including dynamics, thermomechanical coupling, as well as localization in the shearing process. For this purpose, the constitutive equations of the model as well as algorithmic procedures are briefly described. A larger description can be find in [49].

6.1. Elastoplasticity formulation

For the integration of the constitutive equation we followed the approach proposed by [50]. The basic hypothesis underlying this approach to finite strains elastoplasticity is the multiplicative split of the deformation gradient, $\boldsymbol{F}$, into elastic and plastic parts:

$$\boldsymbol{F} = \boldsymbol{F}^e \cdot \boldsymbol{F}^p \tag{21}$$

This assumption, firstly proposed by Lee [51], admits the existence of a local unstressed intermediate configuration. The elastic part of the behavior

is base on a hyperelastic formulation which uses the Eulerian (or spatial) logarithmic strain tensor $\boldsymbol{E}^e = ln(\boldsymbol{U}^e)$. $\boldsymbol{U}^e$ comes from the polar decomposition of the elastic part of the deformation gradient: $\boldsymbol{F}^e = \boldsymbol{R}^e \cdot \boldsymbol{U}^e$.

Within the incremental numerical procedure for solving the history dependent problems, a numerical scheme to verify the material constitutive equation is needed to update stresses as well as the internal variables. We use J2 plasticity with a predicting correcting approach which has the same format that the standard return mapping used in the infinitesimal theory [52, 53, 54, 55].

6.2. Kinematic aspects

Large deformations generate large relative movements between nodes in particular when shearing occurs. To handle the modification of neighborhoods of such nodes we use an updated Lagrangian procedure.

After an updating at time $t^\star$, the interpolation of the total displacement of a particle m at t, denoted $\widetilde{\boldsymbol{u}}(m)_{0\to t}$, is defined by:

$$\widetilde{\boldsymbol{u}}(m)_{0\to t} \;=\; \widetilde{\boldsymbol{u}}(m)_{0\to t^\star} + \widetilde{\boldsymbol{u}}(m)_{t^\star\to t} \tag{22}$$

$$\widetilde{\boldsymbol{u}}(m)_{t^\star\to t} = \sum_{i=1}^{n^\star} \phi_i^\star\left(x^\star(m)\right).\boldsymbol{u}\left(n_i\right)_{t^\star\to t} \tag{23}$$

where $x^\star(m)$ are the coordinates of m at $t^\star$, $n^\star$ is the number of natural neighbors of m at $t^\star$, $\phi_i^\star(x)$ are the shape functions defined on the node repartition at $t^\star$ and $\boldsymbol{u}\left(n_i\right)_{t^\star\to t}$ are the displacements of the natural neighbor n_i of m at $t^\star$.

The gradient of the transformation from 0 to t can then be write:

$$\boldsymbol{F}_{0\to t} \;=\; \boldsymbol{F}_{t^\star\to t}.\boldsymbol{F}_{0\to t^\star} \tag{24}$$

6.3. Treatment of the additional points

As specified in section 3.2, the construction (or reconstruction when updating the configuration) of the CVD can require the addition of new points. These new points are always added on the border of the domain, and thus belong to one of the triangles describing this border.

We suggest handling these new points as "slaves" of the three nodes of the triangular facet on which they are situated. Their movement is thus defined as a linear combination of the displacements of these three "master" nodes.

$$\boldsymbol{u}_{slave} = \sum_{i=1}^{\mathcal{3}} \lambda_i . \boldsymbol{u}_i \tag{25}$$

where $\boldsymbol{u}_i$ is the displacement of each "master" node, and λ_i the associated barycentric coefficient location ($\sum_{i=1}^{\mathcal{3}} \lambda_i = 1$).

This approach presents several advantages: there is no additional unknown and the value of the time step (in the chosen conditionally stable explicit time integration scheme) is not affected. The volume of the "slave" cells is distributed, in proportion to λ_i, on its "masters" nodes.

6.4. Explicit thermomechanical Lagrangian procedure

The principle of virtual work may be formulated as follows:

$$\int_{\Omega^t} \rho(t)\ddot{\mathbf{u}}\boldsymbol{\eta} d\Omega^t + \int_{\Omega^t} \boldsymbol{\sigma}^t : \nabla_{\mathbf{x}}\boldsymbol{\eta} d\Omega^t = \int_{\Omega^t} \rho(t)\mathbf{b}\boldsymbol{\eta} d\Omega^t + \int_{\partial\Omega^t_\sigma} \tau\boldsymbol{\eta} d\Gamma^t \ \ \forall\eta \in \vartheta \tag{26}$$

where ρ is the density, $\mathbf{b}$ and τ represent the body forces and applied tractions respectively, $\boldsymbol{\sigma}^t$ is the Cauchy stress tensor at time t, η is the virtual displacement, and ϑ is the space of virtual displacements. The introduction of the CNEM discretization in the variational form (26), transported in the last update configuration, results in the discrete set of algebraic time dependent equations which may be expressed, in matrix form, as:

$$\mathbf{M}\ddot{\mathbf{u}}(t) = \mathbf{F}^{ext}(t) - \mathbf{F}^{int}(\mathbf{u}_n, t) \tag{27}$$

where t is the time, $\mathbf{M}$ denotes the mass matrix, $\mathbf{F}^{int}(\mathbf{u}, t)$ the internal forces vector, while $\mathbf{F}^{ext}(t)$ is the external forces vector. The use of the SCNI quadrature results in a diagonal matrix for $\mathbf{M}$. The diagonal terms are given by $m_i = \rho_0\Omega_i$, with Ω_i the area (volume in 3D) of the Voronoï cell related to node n_i.

To solve (27) the classical explicit central differences incremental scheme is used. In case of contact a predictor/corrector strategy is used according to the prescribed displacements or tractions.

The weak form of the heat balance can be expressed as:

$$\int_{\Omega^t} \rho(t)c(t)\dot{T}\varphi d\Omega^t + \int_{\Omega^t} k(t)\nabla_{\mathbf{x}}\mathbf{T}\cdot\nabla_{\mathbf{x}}\varphi d\Omega^t = \int_{\Omega^t} r^t\varphi d\Omega^t + \int_{\partial_2\Omega} \bar{q}\varphi d\Gamma^t \ \ \forall\varphi \in \psi \tag{28}$$

where $c(t)$ is the specific heat, $k(t)$ is the thermal conductivity for isotropic conduction, and r^t a heat source related to the inelastic deformations, defined at time t. We have made the assumption that heat generated by elastic deformation is negligible with respect to heat generated by plastic deformation.

In the following, we assume c and k constant in time. φ is the virtual temperatures, ψ is the space of virtual temperatures, and $\bar{q}$ represents the heat transfer at the boundary $\partial_2\Omega$. Following similar arguments as in the mechanical problem (26), the weak form of the heat balance results, with the CNEM discretization, in the ODE system:

$$\mathbf{C}\dot{\mathbf{T}}(t) + \mathbf{K}\mathbf{T}(t) = \mathbf{Q}(t) \tag{29}$$

which in the context of an explicit scheme can be written as

$$\mathbf{C}\mathbf{T}^{n+1} = [\mathbf{C} + \Delta t\mathbf{K}]\mathbf{T}^n + \Delta t\mathbf{Q}^n$$

The matrix $\mathbf{C}$ becomes diagonal in the context of CNEM when SCNI is used. The heat source resulting from the inelastic deformations is given by

$$r^t = \chi\boldsymbol{\sigma}^t : \hat{\mathbf{D}}_t^p \ , \qquad \hat{\mathbf{D}}_t^p = sym\left[\left(\dot{\mathbf{F}}_t^p\right)(\mathbf{F}_t^p)^{-1}\right] \tag{30}$$

where χ is the Taylor-Quinney parameter [56] representing the fraction of plastic work converted into heat. In the present work, we have used $\chi = 0.9$ [57].

When mechanical and thermal problems are coupled, a staggered solution approach is usually adopted, which solves the mechanical and the thermal problems in an uncoupled manner with data exchange performed at the end of each time step or increment. In particular, the nodal temperatures are transferred to the mechanical procedure, while plastic work is communicated to the thermal solver.

6.5. Constitutive law

For the examples below (section 7), we use the Johnson-Cook hardening law (31)[58]. This law allows, in a simple way, to take into account a thermo-mechanical coupling: effects of inelastic deformations and softening effects due to temperature.

$$\sigma_y(\bar{\epsilon}^p) = [A + B(\bar{\epsilon}^p)^n]\left[1 + Cln\left(\frac{\dot{\bar{\epsilon}}^p}{\dot{\bar{\epsilon}}_0^p}\right)\right]\left[1 - \left(\frac{T-T_0}{T_f-T_0}\right)^m\right] \tag{31}$$

A, B, C, m, n and $\dot{\bar{\epsilon}}_0^p$ are material parameters, $\bar{\epsilon}^p$ represents the equivalent plastic strain and $\dot{\bar{\epsilon}}^p$ represents the rate of equivalent plastic strain. T and T_0 correspond to the current and initial temperature respectively.

7. Simulation of manufacturing processes

The objective of this section is to show the ability of the CNEM to simulate problems where very large strains (around 10) are present. We chose an example of high speed blanking and two examples of machining (orthogonal cutting and oblique cutting). For these examples we give theses following data:

- Nodes: the number of nodes at the beginning / end of the simulation,
- T_{tot}: total duration of the simulation,
- N_I: number of corresponding increments,
- N_U: number of updates of the updated Lagrangian procedure,
- T_U: average time for one update,
- R_U: percentage of total time taken for all the updates versus T_{tot}.

For simplicity reason we took the same material for all these examples (table 2). All the simulations were launched on the same computer:
- OS: Linux RedHat 64 bits
- Compiler: Intel c++ compiler 9.1
- Processor: 4 x Intel Itanium 2 Monticito - 1,6GHz - 8 cores used
- RAM: 32G - DDR2 - 667MHz

SCNI and constitutive law integration are parallelized by using TBB (Threading Building Blocks) task primitive.

A(MPa)	B(MPa)	C	n	m	$\dot{\varepsilon}_{p0}(s^{-1})$	T_t(C)	T_f(C)
866	318	0.008	0.25	1.055	$5.77\,10^{-2}$	20	1670

Table 2: TA6V – Johnson-Cook coefficients (from [59])

Figures 27, 29 and 30 are structured in the same way for the three examples. On the top is represented the equivalent plastic strain (for three different instants), and on the bottom the equivalent Von Mises stress (for the same instants).

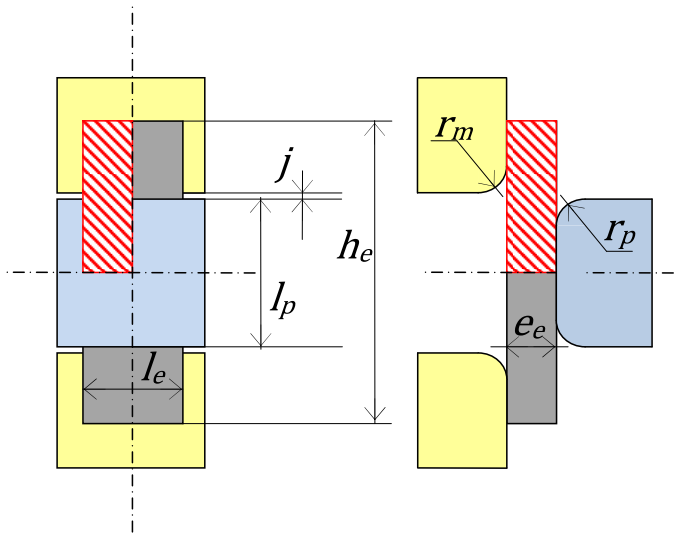


Figure 26: High speed blanking – Diagram of the blanking apparatus

7.1. High speed blanking

Figure 26 gives the diagram of the studied example: the punch is blue, the sample is grey/red (due to symmetry reason only a quarter is modeled), and the die is yellow. The table 3 gives the corresponding data. v is the relative speed between the punch and the die.

v	e_e	l_e	h_e	r_p/r_m	l_p	j
10 m/s	2 mm	17 mm	32 mm	0.5 mm	16 mm	0.1 mm

Table 3: High speed blanking – Geometric and process data

7.2. Orthogonal and oblique cutting

Figure 28 gives the diagram of the studied cutting examples. γ is equal to 0 for orthogonal cutting, otherwise we are in oblique cutting. Table 4 gives the corresponding data. v is the relative speed between the tool and the workpiece.

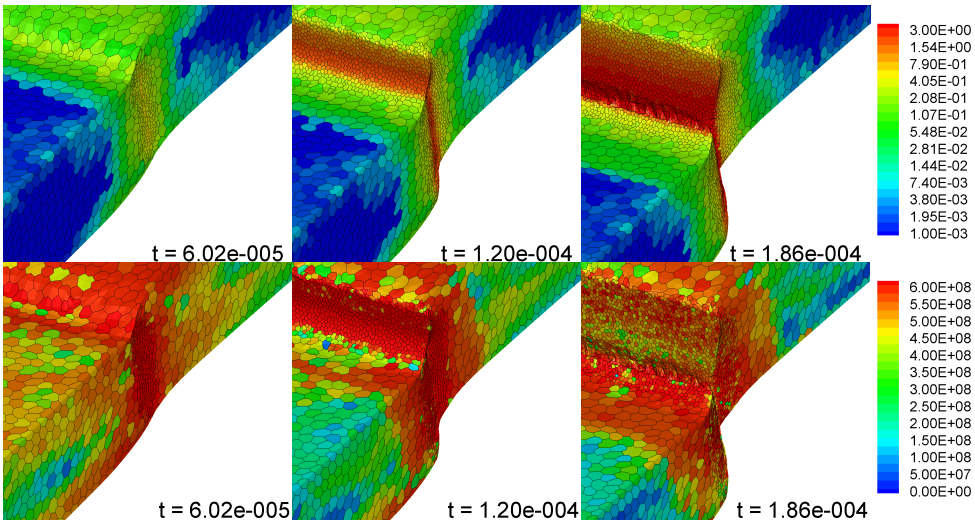


Figure 27: High speed blanking – Equivalent plastic strain and stress contours

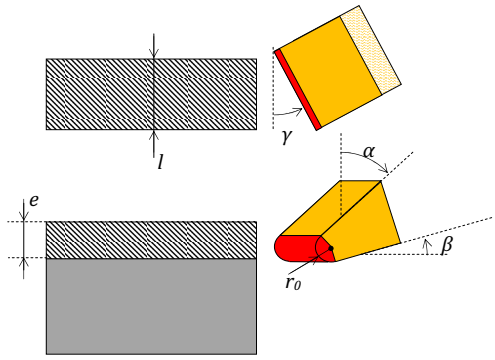


Figure 28: Machining – Diagram of the apparatus

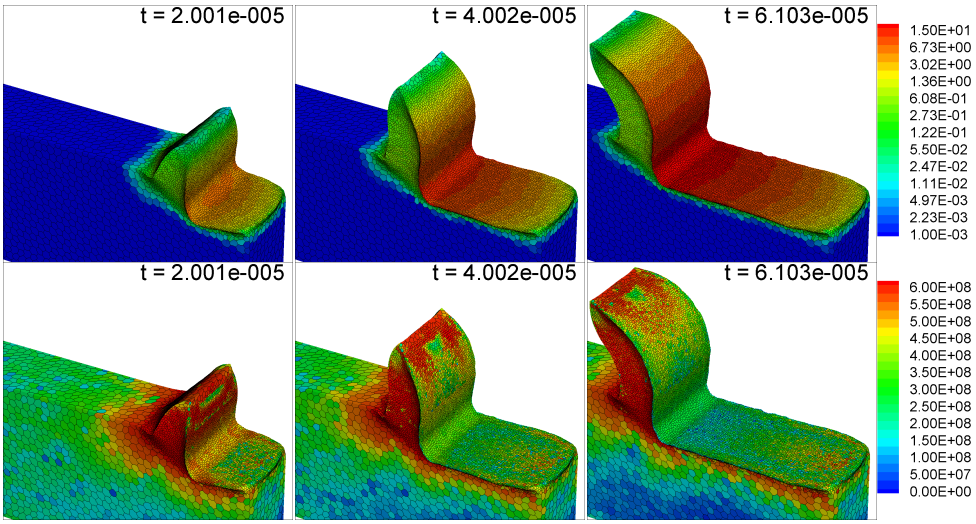


Figure 29: Orthogonal cutting – Equivalent plastic strain and stress contours

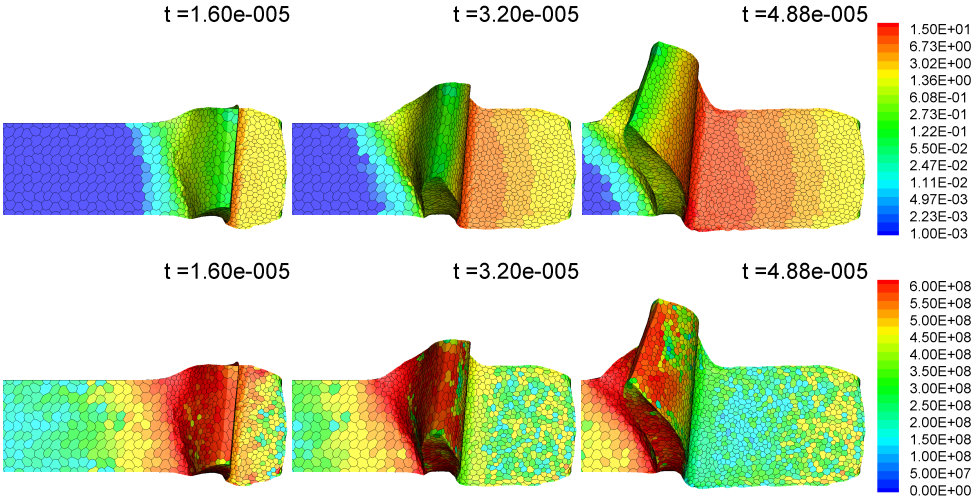


Figure 30: Oblique cutting – Equivalent plastic strain and stress contours

	v	e	l	r_0	α	β	γ
Orthogonal cutting	20 m/s	0.2 mm	0.5 mm	0.05 mm	5°	30°	0°
Oblique cutting	20 m/s	0.2 mm	0.5 mm	0.05 mm	5°	30°	5°

Table 4: Machining – Geometric and process data

7.3. Results

The computational data are given in table 5. One can see that the needed time for all the updates of configuration is non-negligible with respect to the total computational time. In general the ratio is near 30%. It is interesting to see that the number of updates remains relatively low with regard to the number of increments: $\frac{N_I}{N_U}$ is always greater than 300.

	Nodes	T_{tot}	N_I	N_U	T_U	R_U
Blanking	73385/147974	147 h	305944	153	6.82 min	12%
Oth. cutting	14386/78092	32.2 h	148651	449	2.3 min	54%
Obl. cutting	14386/20699	10.55 h	149093	254	0.75 min	30%

Table 5: Computational data

8. Conclusion

We can say that up to now the main difficulties inherent to the implementation of the CNEM 3d are overcome: construction of the constrained Voronoï diagram in $3d$, optimization of the calculation of Sibson shape functions, domain discretization in nodal elementary volumes for the stabilized nodal integration. The elastostatic tests are successful. Although having the same rate of convergence of linear finite elements, the CNEM divides the global error by 3. In finite transformations, for thermo-elasto-viscoplastic behavior, the proposed approach also gives very promising results.

However, the CNEM requires a correct description of the border (no auto-intersection) for the construction of the constrained Voronoï diagram. This involves a possible update of this last one: a remeshing of the border in the area where there are surface folds, excessive elongations or where nodes become to close. For high speed processes (cutting or blanking for example) it is still necessary, to realize more representative simulations, to take into account dynamic effects of the tools and the machine (die, punch, ...). To simulate a process like blanking, it is also necessary to be able to take into

account matter separation. A simple manner to introduce such a separation may concist in using a "kill cell" approach.

If numerical simulation is a key issue for the development of efficient cutting or forming processes, the definition, and the identification, of constitutive laws adapted to high speeds and high temperatures remain still a major difficulty.

9. Acknowledgement

This research was supported by the Cetim (French Industrial and Mechanical Technical Centre), and it is gratefully acknowledged.

Appendix A. Study the CNN shape functions near highly non-convex borders

For non-convex domains, there exist locations for which the insertion of a point in the CVD creates a non-convex Voronoï cell. In order to illustrate that, let take the domain Ω depicted in Figure A.31. The insertion of the point x (coordinates $[-0.1, 0, 0]$), in the CVD of Ω, leads to a non-convex cell (see Figure A.32). We can also see in this Figure, that the cell associated with the neighbor node v (n_0 or n_1) is also non-convex. When x moves the topology of its associated cell evolves continuously from a non-convex topology to a convex topology. In order to highlights this phenomenon, the point x moves gradually, in Figure A.33, from the position $[-0.25,\ 0,\ 0]$ to $[0^-,\ 0,\ 0]$ (displacement along the X axis).

Having noticed this non-convex property, it is interesting to study the evolution of the shape functions according to different positions of a point x. The graph A.34 gives, according to the point x coordinate along the X axis, the volume measures of:
- the point x cell, ($\mathrm{Vol}(c_x)$),
- the polyhedron intersections associated with each neighbor of x.

The graph A.35 gives, again according to the the point x coordinate along the X axis, the value of the shape functions associated with each neighbor of x. All shape function properties stated in 3.4 remain verified. We can however notice that there is a position x_0 for which the volume of the cell of the point x cancels. This gives place to an indetermination of the Sibson shape function (division by 0). Such position is not unique. Indeed, if we

calculate the volume measure of the point x cell in a vicinity of x_0, the location where the volume measure cancel forms a surface (Figure A.36).

It should be noted that even for Laplace CNN shape functions, places for which they are unspecified also exist, as the graph in the Figure A.37 shows it.

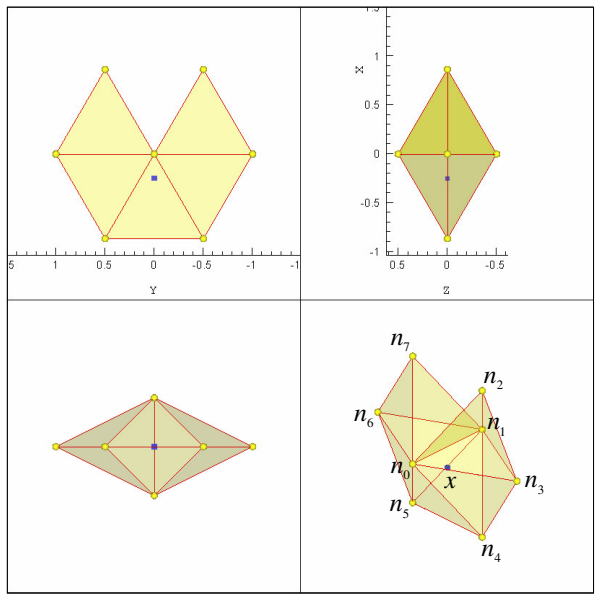


Figure A.31: Non-convex domain – Location of the point x

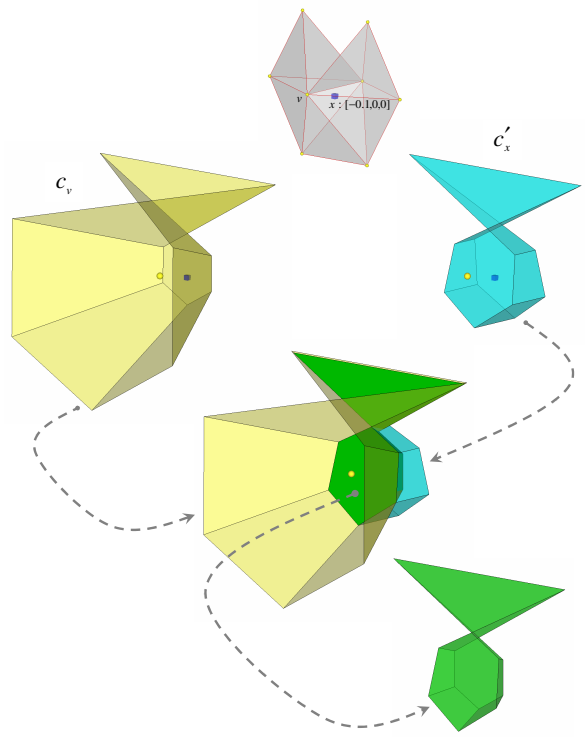


Figure A.32: Non-convex domain – Cell $c_x^{'}$ – Intersection with the cell c_v of a neighbor node

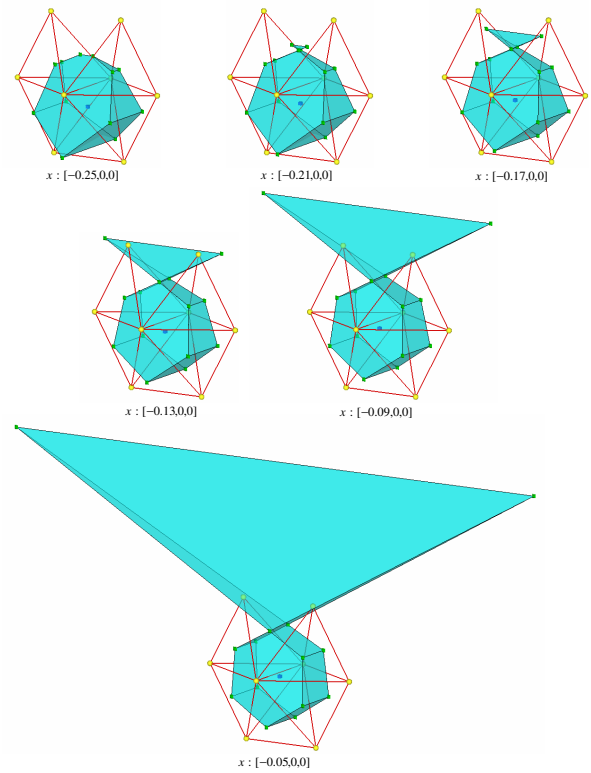


Figure A.33: From a convex cell to a non-convex cell according to the position of x.

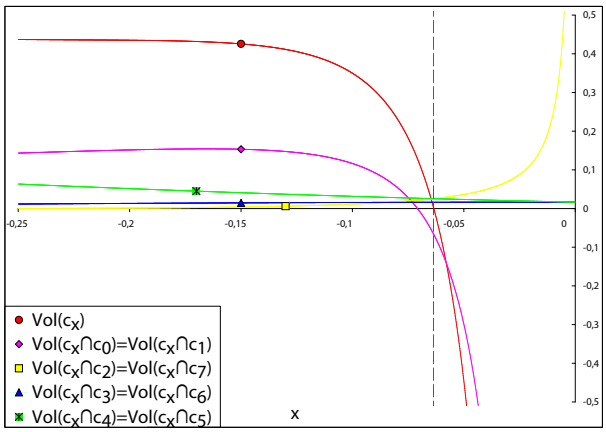


Figure A.34: Volume measurement – Cell of node x – Intersection with neighboring cells

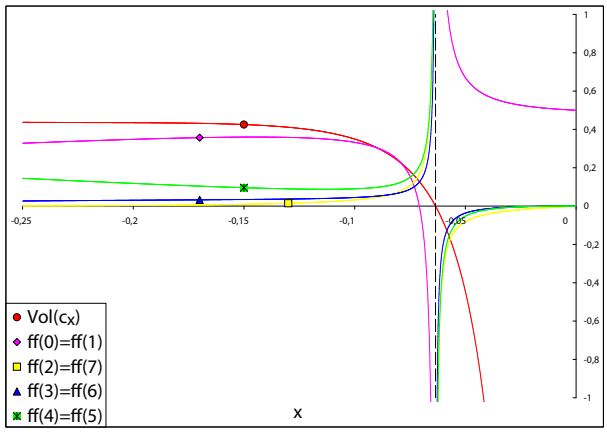


Figure A.35: Sibson shape functions at x for different neighbors

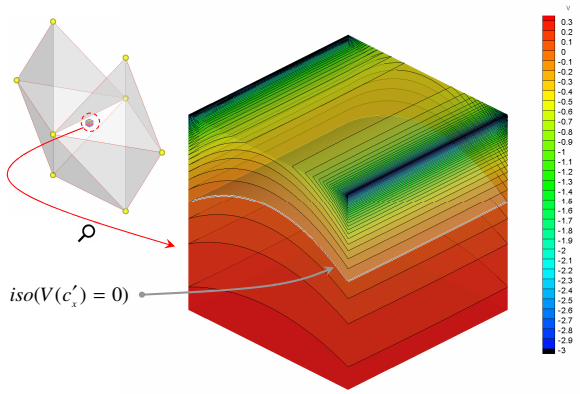


Figure A.36: Space location where the volume of x cell is null

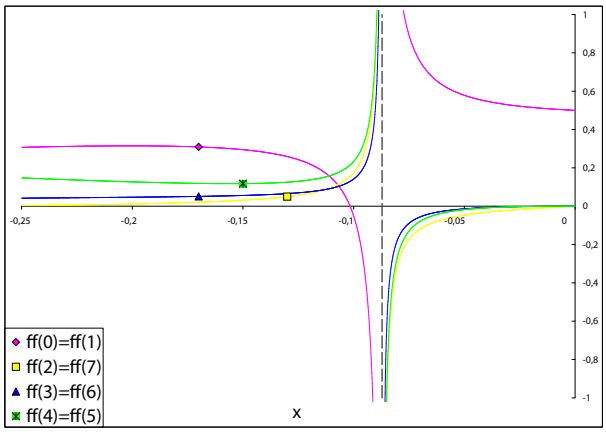


Figure A.37: Laplace shape functions at x for different neighbors

References

[1] P. S. Jensen, Finite difference techniques for variable grids, Computers & Structures 2 (1972) 17 – 29.

[2] L. B. Lucy, A numerical approach to the testing of the fission hypothesis, The Astronomical Journal 82(12) (1977) 1013–1024.

[3] M. Liu, G. Liu, Smoothed particle hydrodynamics (sph): an overview and recent developments, Archives of Computational Methods in Engineering 17 (2010) 25–76.

[4] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Computers & Structures 11 (1980) 83–95.

[5] T. Liszka, An interpolation method for an irregular net of nodes, Int. J. Numer. Meth. Engng. 20 (1984) 1599–1612.

[6] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: Diffuse approximation and diffuse elements, Computational Mechanics 10 (1992) 307–318.

[7] T. Belytschko, Y. Y. Lu, L. Gu, Element-Free Galerkin Methods, International Journal for Numerical Methods in Engineering 37 (1994) 229–256.

[8] W. K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods, International Journal for Numerical Methods in Engineering 38 (1995) 1655–1679.

[9] J. Braun, M. Sambridge, A numerical method for solving partial differential equations on highly irregular evolving grids, Nature 376 (1995) 655–660.

[10] S. De, K. J. Bathe, The method of finite spheres, Computational Mechanics 25 (2000) 329–345.

[11] R. Sibson, A vector identity for the dirichlet tessellation, Scandinavian Journal of Statistics 87 (1980) 151–155.

[12] E. Cueto, M. Doblaré, L. Gracia, Imposing essential boundary conditions in the Natural Element Method by means of density-scaled α-shapes., International Journal for Numerical Methods in Engineering 49-4 (2000) 519–546.

[13] J. Yvonnet, D. Ryckelynck, P. Lorong, F. Chinesta, A new extension of the Natural Element method for non-convex and discontnuous problems: the Constrained Natural Element method, International Journal for Numerical Methods in Enginering 60(8) (2004) 1452–1474.

[14] M. Vaz, D. Owen, V. Kalhori, M. Lundblad, L.-E. Lindgren, Modelling and simulation of machining processes, Archives of Computational Methods in Engineering 14 (2007) 173–204.

[15] J. Yvonnet, Nouvelles approches sans maillage basées sur la méthode des éléments naturels pour la simulation numérique des procédés de mise en forme, Ph.D. thesis, ENSAM, LMSP, 2004.

[16] G. Voronoï, Nouvelles applications des paramètres continus à la théorie des formes quadratiques, deuxième mémoire : recherches sur les parallélloèdres primitifs, J. Reine Angew. Math 134 (1908) 198–287.

[17] N. D. Boris, Sur la sphère vide, Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk 7 (1934) 793–800.

[18] D. Organ, M. Fleming, T. Terry, T. Belytschko, Continuous meshless approximations for nonconvex bodies by diffraction and transparency, Computational Mechanics 18 (1996) 225–235.

[19] B. Joe, C. A. Wang, Duality of constrained voronoi diagrams and delaunay triangulations, Algorithmica 9 (1993) 142–155.

[20] J. R. Shewchuk, A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations, in: Proceedings of the fourteenth annual symposium on Computational geometry, pp. 76–85.

[21] E. Schönhardt, Über die zerlegung von dreieckspolyedern in tetraeder, Mathematische Annalen 98 (1928) 309–312.

[22] H. Si, K. Gärtner, Meshing piecewise linear complexes by constrained delaunay tetrahedralizations, in: 14th International Meshing Roundtable, pp. 147–163.

[23] H. Si, Tetgen version 1.4, a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, 2006. ©2002, 2004, 2005, 2006. http://tetgen.berlios.de/.

[24] G. L. Miller, D. Talmor, S.-H. Teng, N. Walkington, H. Wang, Control volume meshes using sphere packing: Generation, refinement and coarsening, in: 5th International Meshing Roundtable, pp. 47–61.

[25] R. Sibson, The dirichlet tessellation as an aid in data analysis, in: Mathematical Proceedings of the Cambridge Philosophical Society, volume 7, pp. 14–20.

[26] R. Sibson, A brief description of natural neighbor interpolation, in: V. Barnett (Ed.), Interpreting Multivariate Data, John Wiley and Sons Ltd, 1981, pp. 21–36.

[27] B. Piper, Properties of local coordinates based on dirichlet tessellations, Computing Suppl. 8 (1993) 227–239.

[28] H. Hiyoshi, K. Sugihara, Voronoï-based interpolation with higher continuity, in: Proceedings of the 16th Annual ACM Symposium on Computational Geometry, pp. 242–250.

[29] E. Bertin, Diagramme de Voronoï 2D et 3D : applications en analyse d'images, Ph.D. thesis, Université Joseph Fourier - Grenoble 1, TIMC - Institut IMAG, 1994.

[30] A. Bowyer, Computing dirichlet tessellations, Computer Journal 24 (1981) 162–166.

[31] D. Watson, Computing the n-dimensional delaunay tessellation with application to voronoi polytopes, Computer Journal 24 (1981) 167–172.

[32] F. Schmitt, H. Borouchaki, Algorithme rapide de maillage de delaunay dans rd, in: journées de géométrie algorithmique, pp. 131–133.

[33] P. J. Green, R. Sibson, Computing dirichlet tessellation in the plane, Computer Journal 21 (1978) 168–173.

[34] I. Alfaro, J. Yvonnet, F. Chinesta, E. Cueto, A study on the performance of natural neighbour-based galerkin methods, Int. J. Numer. Meth. Engng 71 (2007) 1436–1465.

[35] M. Sambridge, J. Braun, H. McQueen, New computational methods for natural neighbour interpolation in two and three dimensions, in: Computational Techniques and Applications : CTAC95, pp. 685–692.

[36] M. Sambridge, J. Braun, H. McQueen, Geophysical parametrization and interpolation of irregular data using natural neighbours, Geophysical Journal International 122 (1995) 837–857.

[37] L. A. Illoul, Mise en œuvre de la méthode des éléments naturels contrainte en 3D – Application au cisaillage adiabatique, Ph.D. thesis, Ecole nationale supérieure d'arts et métiers, 2008.

[38] J.-D. Boissonnat, F. Cazals, Smooth surface reconstruction via natural neighbour interpolation of distance functions, Computational Geometry 22 (2002) 185–203.

[39] d. David F. Watson, Compound signed decomposition, the core of natural neighbour interpolation in n-dimensional space. unpublished manuscript, 2001. ©2001 D.F.Watson. http://www.iamg.org/naturalneighbour.html.

[40] J. Lawrence, Polytope volume computation, Mathematics of Computation 57 (1991) 259–271.

[41] A. Enge, B. Büeler, K. Fukuda, Vinci version 1.0.5, Computing volumes of convex polytopes, 2003. http://www.lix.polytechnique.fr/Labo/Andreas.Enge/Vinci.html.

[42] B. Büeler, A. Enge, K. Fukuda, Exact volume computation for polytopes: A practical study, Polytopes - Combinatorics and Computation 29 (2000) 131–154.

[43] J.-S. Chen, C.-T. Wu, S. Yoon, Y. You, Stabilized conforming nodal integration for galerkin mesh-free methods, International Journal for Numerical Methods in Engineering 50 (2001) 435–466.

[44] J.-S. Chen, S. Yoon, C.-T. Wu, Non-linear version of stabilized conforming nodal integration for galerkin mesh-free methods, International Journal for Numerical Methods in Engineering 53 (2002) 2587–2615.

[45] W. K. Liu, S. Jun, Y. F. Zhang, Reproducing kernel particle methods, International Journal for Numerical Methods in Fluids 20 (1995) 1081–1106.

[46] D. González, E. Cueto, M. A. Martínez, M. Doblaré, Numerical integration in natural neighbour galerkin methods, International Journal for Numerical Methods in Engineering 60 (2004) 2077–2104.

[47] Intel, Math kernel library, 2007. Copyright ©Intel Corporation, 2007. http://www.intel.com/software/products/mkl.

[48] J. Jaeger, N. Cook, Fundamentals of Rock Mechanics, Chapman and Hall, 1976.

[49] P. Lorong, J. Yvonnet, G. Coffignal, S. Cohen, Contribution of computational mechanics in numerical simulation of machining and blanking, Archives of Computational Methods in Engineering 13 (2006) 45–90.

[50] K.-J. Bathe, Finite Element Procedures, Prentice Hall, 1995.

[51] E. Lee, Elastic-plastic deformation at finite strains, J. Applied Mech. ASME 36 (1969) 1–6.

[52] A. Eterovic, K.-J. Bathe, A hyperelastic-based large strain elasto-plasstic constitutive formulation with combined isotropic-kiematic hardening using the logarithmic stress and strain measures, Int. J. Numer. Meth. in Engng 30 (1990) 1099–1114.

[53] J. Simo, Algorithms for static and dynamic multiplicative plasticity that preserve the classical return-mapping algorithm schemes of the infinitesimal theory, Comp. Meth. Applied Mech. and Eng. 99 (1992) 61–112.

[54] A. Cuitino, M. Ortiz, A material-independent method for extending stress-update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics, Engineering computations 9 (1992) 437–451.

[55] J. Simo, C. Miehe, Associative coupled thermoplasticity at finite strains: formulation numerical analysis and implementation, Comp. Meth. Applied Mech. and Eng. 98 (1992) 41–104.

[56] G. Taylor, H. Quinney, The latent energy remaining in a metal after cold working, Proceedings of the Royal Society 143 (1934) 307–326.

[57] T. Marusich, M. Ortiz, Modelling and simulation of high-speed machining, Int. J. Numer. Meth. in Engng 38 (1995) 3675–3694.

[58] G. R. Johnson, W. H. Cook, A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures, in: Proceeding of the 7th International Symposium on Ballistics, The Hague, Holland, pp. 541–547.

[59] N. Ranc, Etude des champs de température et de déformation dans les matériaux métalliques sollicités à grande vitesse de déformation, Ph.D. thesis, Université Paris X, Nanterre, 2004.