Mise en oeuvre de la méthode des éléments naturels contrainte en 3D : application au cisaillage adiabatique

Amran Lounès Illoul

▶ **To cite this version:**

Amran Lounès Illoul. Mise en oeuvre de la méthode des éléments naturels contrainte en 3D : application au cisaillage adiabatique. Sciences de l'ingénieur [physics]. Arts et Métiers ParisTech, 2008. Français. NNT : 2008ENAM0015 . pastel-00004035

HAL Id: pastel-00004035
https://pastel.hal.science/pastel-00004035

Submitted on 5 Dec 2008

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.

N°: 2008 ENAM 0015

Ecole doctorale n° 432 : Sciences des Métiers de l'Ingénieur

T H È S E

pour obtenir le grade de

Docteur

de

l'École Nationale Supérieure d'Arts et Métiers

Spécialité "Mécanique"

présentée et soutenue publiquement
par

Amran Lounès ILLOUL

le 9 juillet 2008

MISE EN ŒUVRE DE LA MÉTHODE DES ÉLÉMENTS NATURELS CONTRAINTE EN 3D

APPLICATION AU CISAILLAGE ADIABATIQUE

Directeur de thèse : ***Francisco CHINESTA***

Co-encadrement de la thèse : ***Philippe LORONG***

Jury :

M. Yvan CHASTEL, Professeur, CEMEF, MINES ParisTech ..... Président
M. Jean-Daniel BOISSONNAT, Directeur de Recherche, INRIA, Sophia Antipolis ..... Rapporteur
M. Serge CESCOTTO, Professeur, ArGEnCo, Université de Liège ..... Rapporteur
M. Francisco CHINESTA, Professeur, LMSP, ENSAM, Paris. ..... Examinateur
M. Elias CUETO, Professeur associé, GEMM, Université de Saragosse ..... Examinateur
M. Philippe LORONG, Maître de Conférences HDR, LMSP, ENSAM, Paris ..... Examinateur
M. Florian DE VUYST, Professeur, MAS, Ecole Centrale, Paris ..... Invité

Laboratoire de Mécanique des Systèmes et des Procédés
ENSAM, CER de Paris

L'ENSAM est un Grand Etablissement dépendant du Ministère de l'Education Nationale, composé de huit centres :
AIX-EN-PROVENCE ANGERS BORDEAUX CHÂLONS-EN-CHAMPAGNE CLUNY LILLE METZ PARIS

Je remercie les membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer mes travaux de thèse.

Je remercie également le CETIM pour l'intérêt qu'il a bien voulu porter au sujet de thèse au travers du financement de celle-ci.

Je tiens à exprimer ma gratitude ainsi que mes plus sincères remerciements à mes encadrants, Mr Francisco Chinesta et Mr Philippe Lorong, pour tout l'aide qu'ils ont pu m'apporter, tout le temps qu'ils ont pu me consacrer, tout le savoir qu'ils ont pu me transmettre, et surtout tout le soutien qu'ils ont pu m'apporter.

Je tiens à remercier :

Mr Maurice Touratier, ancien directeur du LMSP pour m'avoir accepté dans son laboratoire, et pour avoir fait tout ce qu'il fallait afin que le financement de cette thèse démarre dans les temps,

Mr Christophe Gengembre pour l'aide qu'il m'a apportée, et le temps qu'il à bien voulu me consacrer,

Mr Julien Yvonnet, pour tous les éclaircissements qu'il m'a apportés concernant la CNEM,

Mr Stéphane Clénet pour les collaborations fructueuses que l'on a eues,

Mr Alain Racineux pour toutes les lumières qu'il m'a apporté concernant la problématique du maillage,

Mr André Barraco, Mr Gérard Coffignal, Mr Guy Ishiomin, Mr Sébastien Laporte, et Mr Arnaud Larue, pour avoir bien voulu me confier tout ou une partie des travaux pratiques ou travaux dirigés de leurs unités d'enseignement,

Mr Hang Si ainsi que Mr Andreas Enge pour leur précieuse collaboration,

Mme Farida Zidekhile, Mme Monique Blanc, Mr Jacques Gros La Faige, et Mr Claude Laudet, ainsi que tous les membres du LMSP pour l'accueil qu'ils m'ont réservé au sein du laboratoire.

Que ceux que j'ai oublié de citer veuillent bien me pardonner.

Table des matières

1 Introduction **11**

2 Présentation de l'approche CNEM **15**
2.1 Méthode des éléments naturels **(NEM)** - Méthode des éléments naturels **contrainte (CNEM)** 16
2.2 Les fonctions de forme CNEM 20
2.2.1 Préliminaire 20
2.2.2 Calcul des fonctions de forme CNEM 22
2.2.3 Calcul du gradient des fonctions de forme CNEM de type Sibson 26
2.2.4 Propriétés des fonctions de forme CNEM 27
2.3 Intégration numérique et gradient stabilisé 28
2.4 Conclusion 31

3 Mise en œuvre de la CNEM en 3d **33**
3.1 Structuration adoptée pour les données du diagramme de Voronoï contraint 34
3.2 Construction du diagramme de Voronoï contraint 37
3.2.1 Construction de la tétraèdrisation de Delaunay contrainte d'un domaine $3d$ 37
3.2.2 Passage de la tétraèdrisation de Delaunay contrainte au diagramme de Voronoï contraint 39
3.3 Calcul des fonctions de formes CNEM de type Sibson 42
3.3.1 Etape-1 : Insertion du point x dans le diagramme de Voronoï contraint existant 42
3.3.2 Etape-2 : Calcul de la mesure du volume commun à $\acute{c}_x$ et c_v . . 45
3.3.3 Test comparatif des différents algorithmes 58
3.3.4 Mise en évidence de phénomènes associés aux fonctions de formes CNEM apparaissant en 3d pour des domaines non convexes . . 60
3.4 discrétisation du domaine en vue de l'intégration numérique 67
3.4.1 Cas des domaines convexes ou faiblement non convexes 67
3.4.2 Cas des domaines fortement non convexes 79
3.5 Conclusion 80

4 Validation **83**
4.1 Simulations menées et résultats 84
4.1.1 Sphère creuse sous pression 85

4.1.2 Excavation cylindrique dans un milieu infini 90
4.2 Commentaires sur les résultats obtenus 93
4.3 Choix de la méthode de calcul du gradient stabilisé 93
4.4 Conclusion 95

5 Applications de la CNEM aux grandes transformations 97
5.1 Mise en œuvre des grandes transformations dans Nessy 98
5.1.1 Contexte général de la mise en œuvre 98
5.1.2 Formulation Lagrangienne Actualisée (**FLA**) 98
5.1.3 Traitement des points additionnels 99
5.1.4 Schéma d'intégration temporelle 100
5.1.5 Traitement du contact 100
5.1.6 Intégration de la relation de comportement 100
5.2 Applications 103
5.2.1 Barre de Taylor 103
5.2.2 Cisaillage adiabatique 109
5.3 Conclusion 118

6 Conclusion 119

Table des figures

2.1 Fonction de forme NEM Sibson *(à gauche)* / FEM *(à droite)* 17
2.2 Fonctions de forme NEM Sibson près d'un bord convexe 18
2.3 Fonctions de forme Sibson près d'un bord **non-convexe**, NEM *(à gauche)* / CNEM *(à droite)* 19
2.4 Critère de visibilité 21
2.5 Région de Voronoï associée à un nœud, non-contrainte*(à gauche)*,contrainte *(à droite)* 22
2.6 Tétraèdrisation contrainte de Delaunay du polyèdre de **Schönhardt** par ajout de nœuds 23
2.7 Calcul des fonctions de forme Sibson 24
2.8 Fonction de forme en 2d EN Sibson / EF linéaires 25
2.9 Fonction de forme en $3d$, EN Sibson / EF linéaires 25
2.10 Résultat de l'utilisation d'un diagramme de Voronoï non contraint / contraint pour le calcul d'une fonction forme Sibson 26
2.11 Fonctions de forme éléments naturels 2d Laplace / Sibson 29

3.1 Conventions pour les sommets 34
3.2 Conventions pour les arêtes 35
3.3 Conventions pour les faces 36
3.4 Connexion entre les différentes entités du diagramme de Voronoï 36
3.5 Exemple de PLC 37
3.6 Arêtes et faces connectées à un sommet s 40
3.7 Calcul des fonctions de formes NEM Sibson 43
3.8 Cas de figure pouvant conduire à la création d'un tétraèdre plat 44
3.9 Etape-2 : Calcul de la mesure de volume du polyèdre P^{int} par Lasserre 46
3.10 Projection orthogonale de la face f_i sur un plan de normale $\vec{e_j}$ 48
3.11 Etape-2 : Calcul de la mesure de volume du polyèdre P^{int} par le biais du volume complémentaire 49
3.12 Contribution du volume associé à une arrête au volume d'une cellule . . 50
3.13 Volumes A et B qui s'interpénètrent 51
3.14 Etape-2 : Calcul de la mesure de volume du polyèdre P^{int} par approche topologique basée sur le DVC 52
3.15 Identification des entités utilisées dans l'approche topologique basée sur le DVC 55

3.16 Etape-2 : Calcul de la mesure de volume des polyèdres P^{int} par approche topologique basée sur la TDC . . . 56
3.17 Décomposition de Lawrence en $2d$ . . . 58
3.18 Algorithme de Watson en $2d$ . . . 59
3.19 Domaine non convexe et positionnement du point x . . . 62
3.20 Domaine non convexe : cellule $c_x^{'}$ – Intersection avec la cellule du nœud voisin c_v . . . 63
3.21 Passage d'une cellule convexe à une cellule non convexe en fonction de la position de x . . . 64
3.22 Mesure de volumes : cellule du nœud x, et intersections avec les cellules voisines . . . 65
3.23 Fonction de formes Sibson en x pour les différents voisins . . . 65
3.24 Lieux de l'espace où le volume de la cellule du nœud x est nul . . . 66
3.25 Fonction de formes Laplace en x pour les différents voisins . . . 66
3.26 Exemple de cellule de Voronoï tronquée par la frontière . . . 67
3.27 Analyse d'une cellule de Voronoï tronquée par la frontière . . . 68
3.28 Intersection entre une arête de la triangulation et les faces de Voronoï . . . 69
3.29 Intersections [arête Voronoï/face triangulation] . . . 71
3.30 Discrétisation obtenue par le biais des quarts de tétraèdre . . . 79
3.31 Domaine produisant des cellules de Voronoï non convexes en son intérieur 80
3.32 Cellule de Voronoï non convexe à l'intérieur du domaine . . . 81

4.1 Conditions aux limites . . . 86
4.2 Contrainte équivalente de Von Mises - $\approx$ **3.500** nœuds *(integration stabilisée - FF Sibson)* . . . 86
4.3 Contrainte équivalente de Von Mises - $\approx$ **300.000** nœuds *(integration stabilisée - FF Sibson)* . . . 87
4.4 Courbes de convergence $\|e\| = f(nb\ nœud)$ - *double échelle logarithmique* 87
4.5 Courbes d'éffectivité *(degré de convergence)* - *échelle logarithmique pour le nombre de nœuds* . . . 88
4.6 Temps de calcul et erreur - $\approx$ **300.000** nœuds . . . 89
4.7 Conditions aux limites . . . 90
4.8 Contrainte équivalente de Von Mises - $\approx$ **5.000** nœuds *(integration stabilisée - FF Sibson)* . . . 91
4.9 Contrainte équivalente de Von Mises - $\approx$ **370.000** nœuds *(integration stabilisée - FF Sibson)* . . . 91
4.10 Courbes de convergence $\|e\| = f(nb\ nœud)$ - *double échelle logarithmique* 92
4.11 Courbes d'éffectivité - *échelle logarithmique pour le nombre de nœuds* . . . 92
4.12 Erreurs associées aux différentes méthodes de calcul du gradient stabilisé 94

5.1 Barre de Taylor *(Schema)* . . . 103
5.2 Comparaison entre les profils obtenus : expérience / FEM/ CNEM . . . 104
5.3 Comparaison entre profils obtenus : conduction / adiabatique / T constant 105
5.4 Barre de Taylor - σ Von-Mises - $0s < t < 3,84.10^{-5}s$ . . . 106
5.5 Barre de Taylor - p - $0s < t < 3,84.10^{-5}s$ . . . 106

5.6 Barre de Taylor - σ Von-Mises - $3,84.10^{-5}s < t < 9,14.10^{-5}s$ 107
5.7 Barre de Taylor - p - $3,84.10^{-5}s < t < 9,14.10^{-5}s$ 107
5.8 Barre de Taylor - σ Von-Mises - $9,14.10^{-5} < t < 1,18.10^{-4}$ 108
5.9 Barre de Taylor - p - $9,14.10^{-5} < t < 1,18.10^{-4}$ 108
5.10 Cisaillage adiabatique *(Schemas)* 109
5.11 Cisaillage 304L - $0s < t < 1,5.10^{-6}s$ 112
5.12 Cisaillage 304L - $1,5.10^{-6}s < t < 2,3.10^{-6}s$ 113
5.13 Cisaillage 304L - $2,3.10^{-6}s < t < 2,9.10^{-5}s$ 114
5.14 Cisaillage 304L - $2,9.10^{-5}s < t < 3,9.10^{-5}s$ 115
5.15 Cisaillage TA6V - $0s < t < 7,5.10^{-6}s$ 116
5.16 Cisaillage TA6V - $7,5.10^{-6}s < t < 5.10^{-5}s$ 117

Chapitre 1

Introduction

La simulation numérique peut constituer un apport décisif dans la recherche de conditions optimales lors de la mise en œuvre de procédés de fabrication. En plus de limiter les campagnes d'essais elle peut participer à la compréhension de phénomènes complexes en donnant accès à des données *(températures, contraintes, ...)* difficilement accessibles par des démarches expérimentales. Ceci est notamment le cas pour le cisaillage à grande vitesse, également appelé cisaillage adiabatique, pour lequel la brièveté des phénomènes *(pour une éprouvette de 5 millimètres d'épaisseur la propagation de la bande cisaillement se fait sur une durée de quelques dizaines de micro-secondes)* rend très délicate l'observation et la mesure. Aujourd'hui, pour ce procédé, la définition des paramètres optimaux *(vitesse poinçon, jeu poinçon/matrice, ...)* est encore loin d'être maîtrisée.

La simulation numérique en $3d$ de ce procédé, et d'une manière générale de tous les procédés mettant en œuvre de très grandes déformations de la matière, par une approche de type éléments finis est confrontée à des difficultés pratiques suite aux fortes distorsions des éléments. Des procédures de remaillages doivent être mises en œuvre. Celles-ci sont coûteuses en temps de calcul et rendent nécessaire, à chaque remaillage, la projection de champs ce qui tend à dégrader la qualité de la solution. Une alternative possible aux approches éléments finis sont les méthodes sans maillage [BKO+96]. Ces approches se libèrent de la contrainte de maintien de la qualité géométrique des éléments supports de l'interpolation et donc de la nécessité de déplacement de nœuds vis à vis de la matière. Ainsi, tout au long d'une simulation, elles rendent possible la conservation des nœuds initiaux *(en affinant éventuellement l'interpolation par ajout de nœuds dans certaines zones)* qui sont de véritables points matériels. En définissant toutes les variables aux nœuds *(typiquement déplacements, contraintes, température, déformations plastiques, ...)* il est alors aisé de suivre l'évolution des variables qui y sont associées sans projection. Les méthodes sans maillage présentent cependant deux inconvénients : le choix de la taille du support des fonctions de formes de l'interpolation *(car non automatique)*, et l'imposition des conditions aux limites.

Une troisième voie existe. Il s'agit de l'approche NEM *(Natural Element Method)* qui est à mi-chemin des méthodes sans maillage et de la méthode des éléments finis. La NEM propose une interpolation basée sur le diagramme de Voronoï associé au nuage de nœuds répartis sur le domaine à étudier. Ce diagramme de Voronoï est le dual du

maillage de Delaunay. Il y a donc un maillage utilisé pour la construction de l'interpolation. Cependant, comme le montrent les exemples présentés dans ce document, la qualité de cette interpolation ne dépend pas de la forme des triangles *(problèmes bidimensionnels)* ou tétraèdres *(problèmes tridimensionnels)* présents dans le maillage de Delaunay. Ce dernier est construit de façon systématique sans nécessiter le déplacement de nœuds. Avec la NEM le choix du support des fonctions de forme est automatique et optimal dans le sens où le voisinage des nœuds est pris en compte au mieux pour définir l'interpolation. Pour ce qui est de l'imposition des conditions aux limites, pour des domaines convexes, elle est directe et procède de la même démarche que les éléments finis : l'influence des nœuds internes à un domaine donné s'annulent sur les bords de ce dernier. La NEM cumule, pour les domaines convexes, les avantages des méthodes sans maillage et des approches éléments finis même si, vis à vis de ces dernières, un surcoût existe pour la construction de l'interpolation.

La méthode des éléments naturels contrainte CNEM *(Constrained Natural Element Method)* a été introduite afin de garder les propriétés de la NEM pour des domaines non convexes. L'interpolation CNEM est construite sur le diagramme de Voronoï **contraint** qui est le dual du maillage de Delaunay **contraint**. Ainsi, en plus du nuage de nœuds, une description valide de la frontière du domaine doit être introduite. Le maillage de Delaunay est contraint à respecter cette frontière.

Le travail de cette thèse a principalement deux objectifs. Le premier est la mise en œuvre de la méthode des éléments naturels contrainte dans un contexte tridimensionnel, et le second est son utilisation dans la simulation des transformations finies et en particulier celle du cisaillage à grande vitesse. La première partie du document *(chapitre 2)* débute par une présentation générale de la méthode des éléments naturels contrainte. Afin d'introduire avec plus de précision cette méthode, les définitions s'y afférent sont ensuite données. Dans la seconde partie *(chapitre 3)* est abordée la mise en œuvre en 3d de l'approche. Trois aspects sont concernés : la construction du diagramme de Voronoï contraint, le calcul des fonctions de formes éléments naturels de type Sibson, et la discrétisation du domaine en vu de l'intégration nodale stabilisée conforme SCNI. Pour ces trois aspects nous avons soit développé nos propres algorithmes et/ou intégré et évalué des algorithmes issus du travail d'autres équipes*(certains de ces algorithmes ont été reprogrammés à cette fin)*. Deux critères ont été pris en considération : la robustesse et les performances en temps de calcul. La troisième partie *(chapitre 4)* porte sur la validation de la CNEM en 3d au travers de deux exemples en élasticité linéaire *(sous l'hypothèse des petites perturbations)*. Y est notamment abordé l'importance des choix faits pour le calcul de l'intégration stabilisée. Cette validation s'appuie sur des comparaisons, en terme d'erreur et de taux de convergence *(effectivité)* vis à vis des solutions analytiques et des solutions obtenues avec la méthode des éléments finis. La dernière partie *(chapitre 5)* présente la démarche adoptée pour l'utilisation de la CNEM en grandes transformations. Deux exemples sont étudiés. Le premier exemple porte sur le test de la barre de Taylor. Cet exemple est bien référencé dans la littérature [Ban05]. Il a permis de valider et comparer notre mise en œuvre vis à vis d'autres approches. Le second exemple porte sur le cisaillage à grande vitesse. Sont notamment étudiés les premiers instants du cisaillage et l'importance du type de matériau sur la localisation

des bandes de cisaillement. Dans la conclusion nous faisons un bilan de ce travail et jetons les bases des développements à venir.

Les travaux présentés dans ce document s'intègrent dans un projet plus large qui associe le CNRS, l'ENSAM et le CETIM autour d'un laboratoire commun : le LASIP *(Laboratoire pour la Simulation des Procédés)*. L'objectif principal de ce laboratoire est d'aller vers une meilleure compréhension et maîtrise du cisaillage à grande vitesse. Ainsi, en parallèle de la présente démarche simulation, un banc d'essais de cisaillage à grande vitesse à été mis en place sur le site de l'ENSAM Paris. Ce banc d'essais vise, par une maîtrise "optimale" des conditions opératoires, à faire des observations et des mesures pour aller vers une définition et une identification de modèles de comportement. Ces modèles devront être suffisamment représentatifs pour que, utilisés dans l'approche CNEM, les simulations soient prédictives.

Chapitre 2

Présentation de l'approche CNEM

L'objet du présent chapitre est de présenter l'approche C-NEM sans entrer dans les détails de sa mise en œuvre. Après avoir esquissé les grands traits de l'approche et avoir situé celle-ci vis à vis de la NEM, nous revenons sur la définition du diagramme de Voronoï Contraint. Ce diagramme est à la base de la construction de l'interpolation CNEM. Nous rappelons également les particularités et propriétés vérifiées par cette interpolation avant de finir le chapitre par une présentation de la technique d'intégration utilisée dans notre mise en œuvre de la CNEM.

2.1 Méthode des éléments naturels (NEM) - Méthode des éléments naturels contrainte (CNEM)

La méthode des éléments naturels *(NEM)* [SBM96][Suk98] propose une interpolation définie à partir de constructions géométriques se basant sur un diagramme de Voronoï. Ce diagramme de Voronoï est celui associé au nuage de noeuds répartis sur le domaine étudié. Les nœuds sont des points du domaine où sont définis les degrés de libertés (*DDL*) de l'interpolation. L'interpolation proposée est une combinaison linéaire de fonctions de forme, une par nœud, locales en espace et interpolantes *(la valeur de l'interpolation en un nœud est égale à la valeur des DDL du nœud)*.

Le diagramme de Voronoï est le dual du maillage de delaunay associé au nuage de nœuds. Cependant, contrairement à des fonctions de forme dont la construction est basée sur un maillage *(éléments finis par exemple)*, les fonctions de forme NEM ne dépendent pas du maillage de Delaunay sous-jacent, mais plutôt de la répartition spatiale des nœuds.

Afin de mieux illustrer cela, la figure 2.1 donne, pour une distribution nodale fixée, et en un point x donné, la valeur de la fonction de forme associée à chaque nœud *(diamètre des cercles rouges)* : pour l'interpolation NEM Sibson à gauche, et FEM *(éléments finis linéaires)* à droite pour deux maillages différents.

Pour la NEM, les voisins de x *(nœuds du domaine pour lesquels les fonctions de forme associées sont non nulles)* sont les nœuds les plus proches de x, nœuds j, h, k, et g. Pour cette interpolation, plus un nœud est proche du lieu ou l'on évalue l'interpolation, plus son influence *(la valeur de la fonction de forme qui lui est associée)* est grande. On peut constater que les valeurs des fonctions de forme associées aux voisins de x sont presque identiques, car x est sensiblement à une même distance de ces derniers.

Ceci n'est pas vérifié pour la FEM, que ce soit pour un maillage fortement distordu *(maillage du haut)*, ou pour un maillage de Delaunay qui minimise cette distorsion *(maillage du bas)*. Pour le maillage du haut, les voisins de x sont i, e, et f, qui ne sont pas les nœuds les plus proches de x. Pour le maillage du bas, les voisins sont j et g *(k est aussi un voisin mais la valeur de la fonction de forme associée est pratiquement nulle)*, alors que k et h sont à une distance similaire de x.

Comparée aux autres méthodes sans maillages, l'approche NEM procède un avantage important. Sur la frontière du domaine, la valeur du champ interpolé par des fonctions de forme NEM, ne dépend que de la valeur de ce champ aux nœuds de la frontière, et ce linéairement *(figure 2.2)*. Autrement dit, les nœuds internes au domaine n'ont pas d'influence sur la frontière du domaine : les fonctions de forme associées à ces derniers s'**annulent** à la frontière. Ceci permet d'imposer de façon stricte des conditions aux limites qui sont indépendantes de toute valeur aux nœuds internes au domaine.

Cependant, cette propriété n'est valable que pour des domaines convexes. Pour des domaines **non-convexes** sur et près d'un bord **non-convexe**, le champ interpolé peut

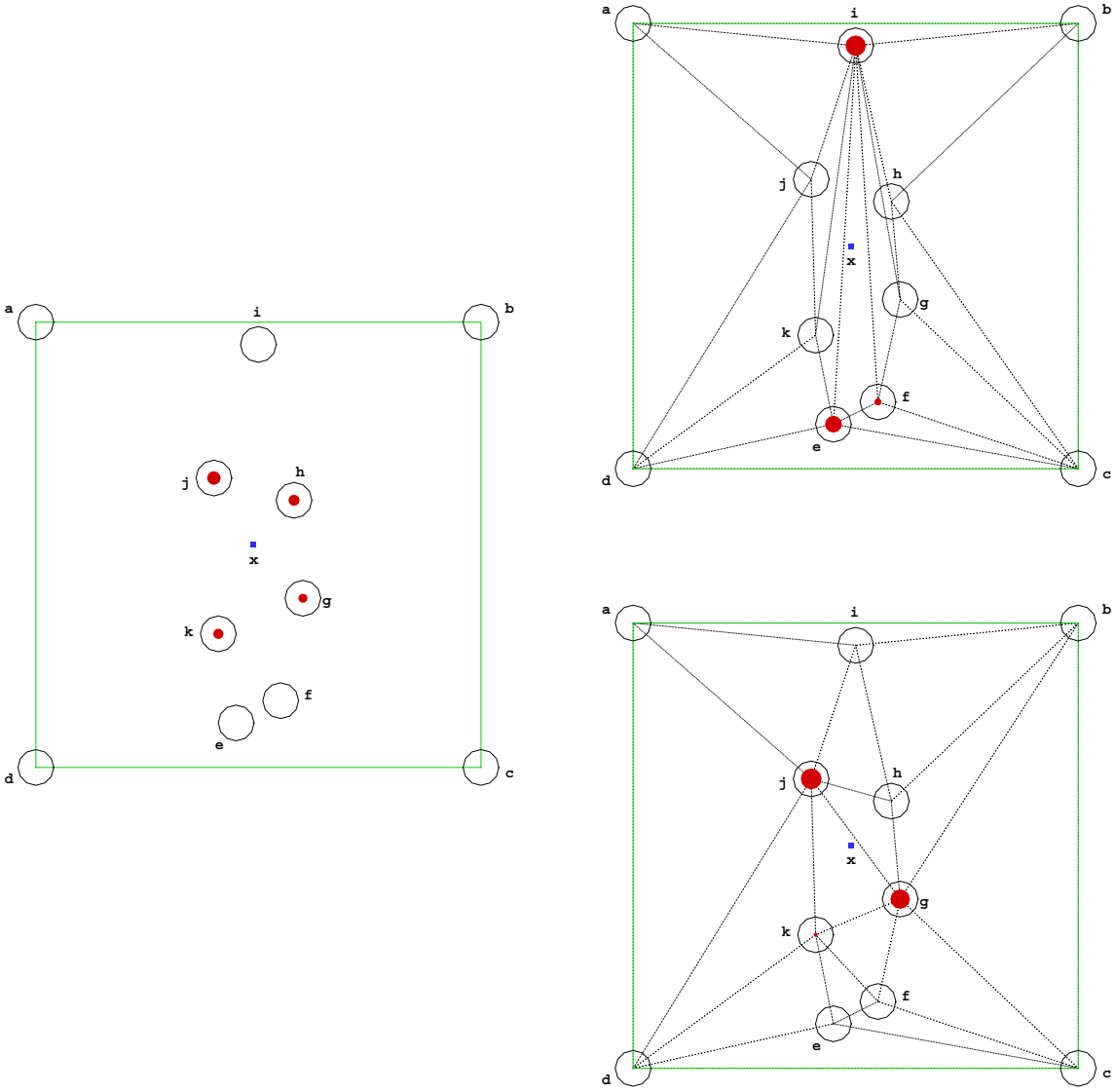

FIG. 2.1 – Fonction de forme NEM Sibson *(à gauche)* / FEM *(à droite)*

dépendre non seulement de nœuds internes au domaine mais aussi de nœuds se situant de l'autre coté de ce bord *(voir figure 2.3 à gauche)*.

La CNEM *(Constrained Natural Element method)* [Yvo04] a été introduite afin remédier à cette lacune. La CNEM n'est qu'une extension de la NEM pour des domaines **non-convexes**. Ainsi, pour la NEM le calcul des fonctions de forme se base

sur le diagramme de *Voronoï*, pour la CNEM il se base sur le diagramme de Voronoï **contraint** par la frontière du domaine.

La figure 2.3 illustre l'apport de la CNEM *(dessins à droite)* vis à vis de la NEM *(dessins à gauche)* près d'un bord non convexe. Comme pour la figure 2.1, le diamètre des cercles rouges représente la valeur des fonctions de forme en x.

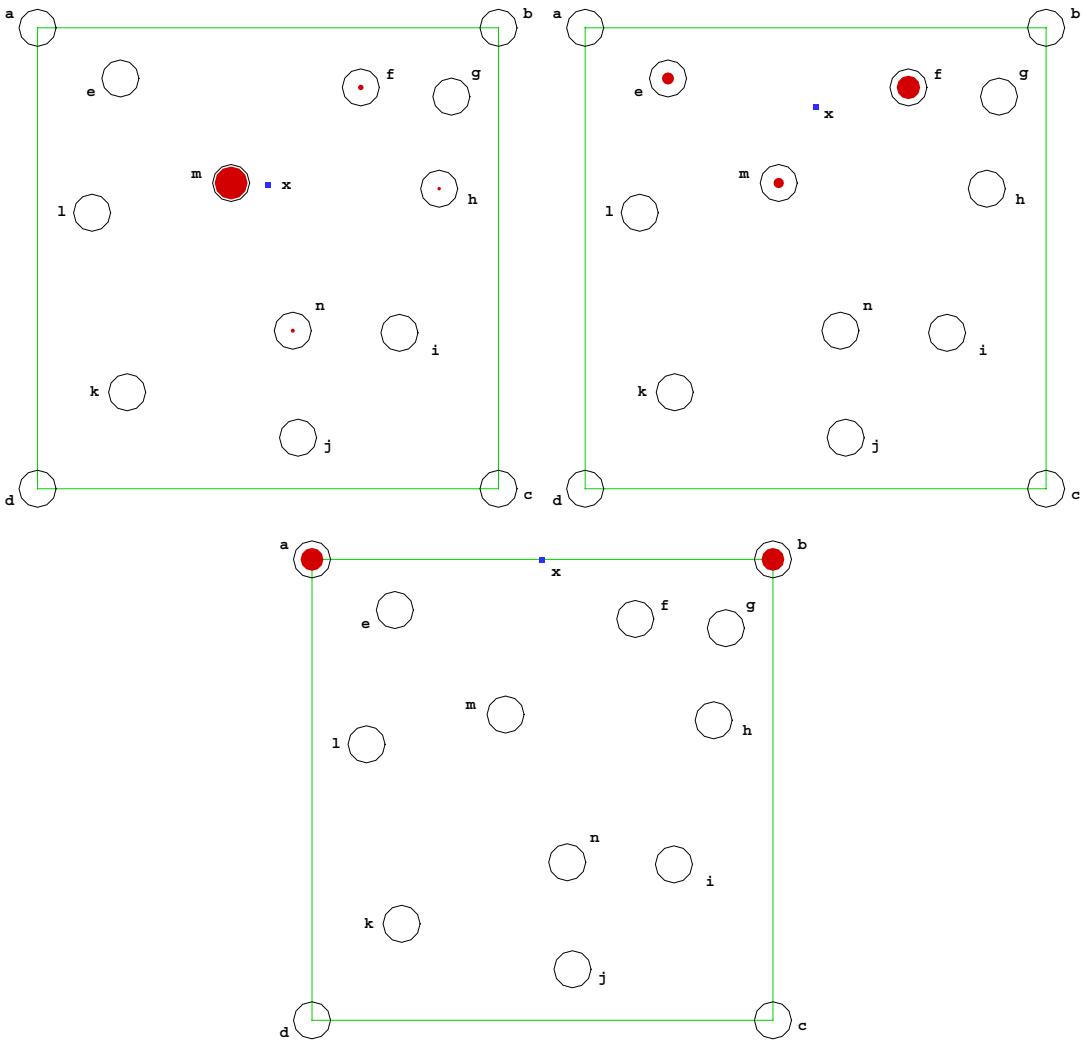

FIG. 2.2 – Fonctions de forme NEM Sibson près d'un bord convexe

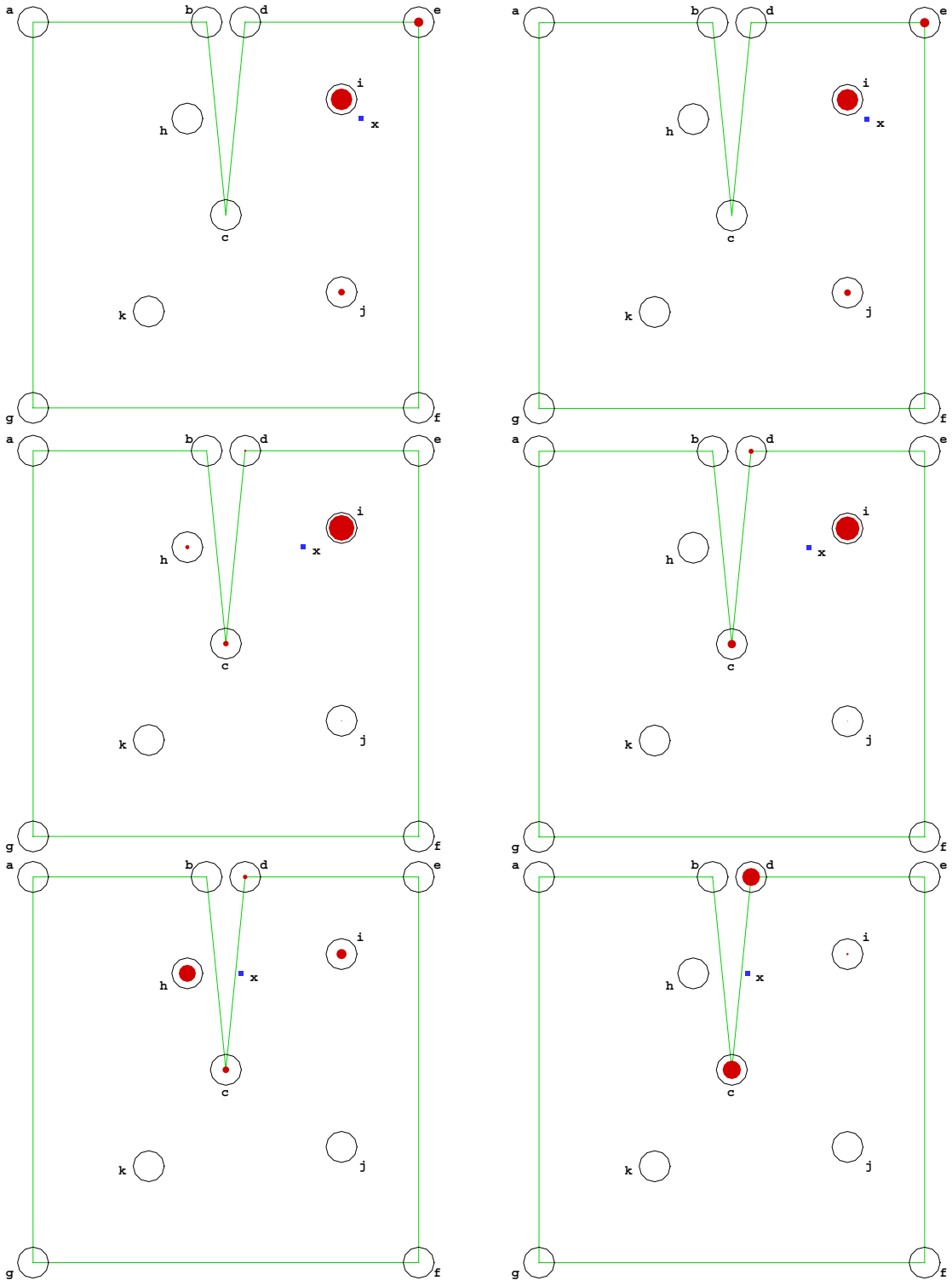

FIG. 2.3 – Fonctions de forme Sibson près d'un bord **non-convexe**, NEM *(à gauche)* / CNEM *(à droite)*

2.2 Les fonctions de forme CNEM

Dans ce paragraphe, pour des raisons de commodité, nous nous référons souvent au cas bidimensionnel ($n = 2$), suivi en italique d'une généralisation au cas tridimensionnel ($n = 3$) ou plus ($n > 3$). D'une manière générale nous nous intéressons à un domaine Ω de R^n dont la frontière est notée Γ.

2.2.1 Préliminaire

Diagramme de Voronoï et triangulation de Delaunay

Soit S un ensemble de nœuds de R^n, et soit p un nœud de S. La région de Voronoï [G.M08] associée à p, notée $Vor_S(p)$, est l'ensemble des points de R^n plus proches de p que de tout autres nœud q de S.

$$Vor_s(p) = \{x \in R^n, d(x,p) \leq d(x,q), \forall q \in S, q \neq p\}$$

où $d(.,.)$ est la distance euclidienne entre deux points.

Le diagramme de Voronoï est le dual de la triangulation de Delaunay [N.34], triangulation pour laquelle le cercle *(sphère, hyper-sphère)* circonscrit*(e)* à chaque triangle *(tétraèdre, simplexe)* ne contient aucun nœud à l'intérieur.

Vocabulaire

- Sommet : centre du cercle *(sphère, hyper-sphère)* qui circonscrit un triangle *(tétraèdre, simplexe)* de *Delaunay*. Ce sommet est donc associé à trois nœuds générateurs *((n+1) en nd)*, il est en liaison avec à trois autres sommets *((n+1) en nd)*, ceux associés aux triangles *(tétraèdre, simplexe)* de *Delaunay* adjacents.
- Arête : segment reliant deux sommets adjacents, elle est associée à deux nœuds générateurs *(n en nd)*
- Face : plan *(hyper-plan)* médian entre deux nœuds voisins *(nœuds générateurs)*.
- Cellule : région de Voronoï.

Diagramme de Voronoï contraint et triangulation de Delaunay contrainte

La définition du diagramme de Voronoï contraint s'appuie sur le critère de visibilité [OFTB96] :

Un nœud q est dit visible d'un autre nœud p, et réciproquement, si, et seulement si, le segment de droite $[p,q]$ les reliant ne transperce pas la frontière Γ du domaine Ω, et n'est pas à l'extérieur de ce dernier.

Sur la figure 2.4, les segments [a,b], [a,c], et [a,d] transpercent Γ. Les nœuds b, c, et d ne sont donc pas visibles par a et réciproquement. Par ailleurs, le segment [e,c] est à l'extérieur du domaine, les nœuds e et c ne sont donc pas visibles l'un de l'autre.

La définition de la région de Voronoï contrainte associée à un nœud p, $Vor_S^c(p)$, trouvée classiquement dans la littérature [JW93], est la suivante :

$$Vor_s^c(p) = \{x \in R^n \ : \ d(x,p) \leq d(x,q), \forall q \in S, \ q \neq p, \ x \ p \ visible, \ x \ q \ visible\} \quad (2.1)$$

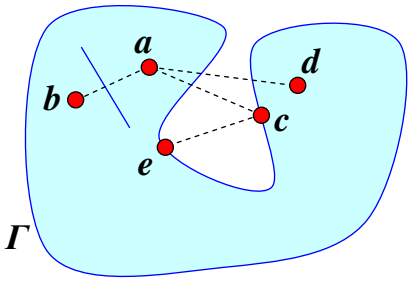

FIG. 2.4 – Critère de visibilité

Cette définition restreint la région de Voronoï contrainte associée à un nœud donné, à sa partie interne au domaine. Afin de pouvoir calculer des fonctions de forme par la suite, cette région ne doit pas être **tronquée** par la frontière du domaine.

Une définition de la région de Voronoï contrainte associée à un nœud, plus étendue, dans le sens où elle ne tronque pas cette région par la frontière, est la suivante :

$$Vor_s^c(p) = \{x \in R^n \ : \ d(x,p) \leq d(x,q), \forall q \in S, \ q \neq p \wedge p \ q \ visible\} \tag{2.2}$$

Sur la figure 2.5 est représentée à gauche une cellule de Voronoï pour un diagramme non contraint, et sur les deux dessins de droite, les cellules associées au même nœud pour un diagramme de voronoï contraint. La cellule en haut à droite est celle issue de la définition 2.1, la cellule en bas à droite correspond à la définition 2.2. Pour le diagramme contraint issu de la définition 2.2, seule l'application aux nœuds du critère de visibilité sert à définir la cellule du nœud p. Les nœuds k, l ne sont pas visibles par le nœud p, car les segments $[p,l]$ et $[p,k]$ transpercent la frontière du domaine.

Il faut souligner que le diagramme de Voronoï d'un nuage de nœuds, est un diagramme de Voronoï contraint par l'**enveloppe convexe** de ce nuage de nœuds. Pour un domaine non convexe, l'enveloppe convexe des nœuds n'est plus sa frontière, d'où la différence entre les deux diagrammes pour de tels domaines.

Le diagramme de Voronoï contraint est le dual de la triangulation de Delaunay contrainte[She98]. Pour cette triangulation le cercle *(sphère,hyper-sphère)* circonscrit(e) à chaque triangle *(tétraèdre, simplexe)* ne contient aucun nœud **visible** par les nœuds de ce triangle. En $2d$ la triangulation de Delaunay contrainte existe toujours que ce soit pour des domaines convexes ou non. Ceci n'est malheureusement plus vrai en $3d$ pour certains domaines non convexes comme, par exemple, le polyèdre de Schönhardt [E.28] *(voir figure 2.6)*. Une solution dans ces cas consiste à ajouter des nœuds supplémentaires sur la frontière du domaine. Nous reviendrons plus en détail sur cet aspect dans le paragraphe 3.2.1.

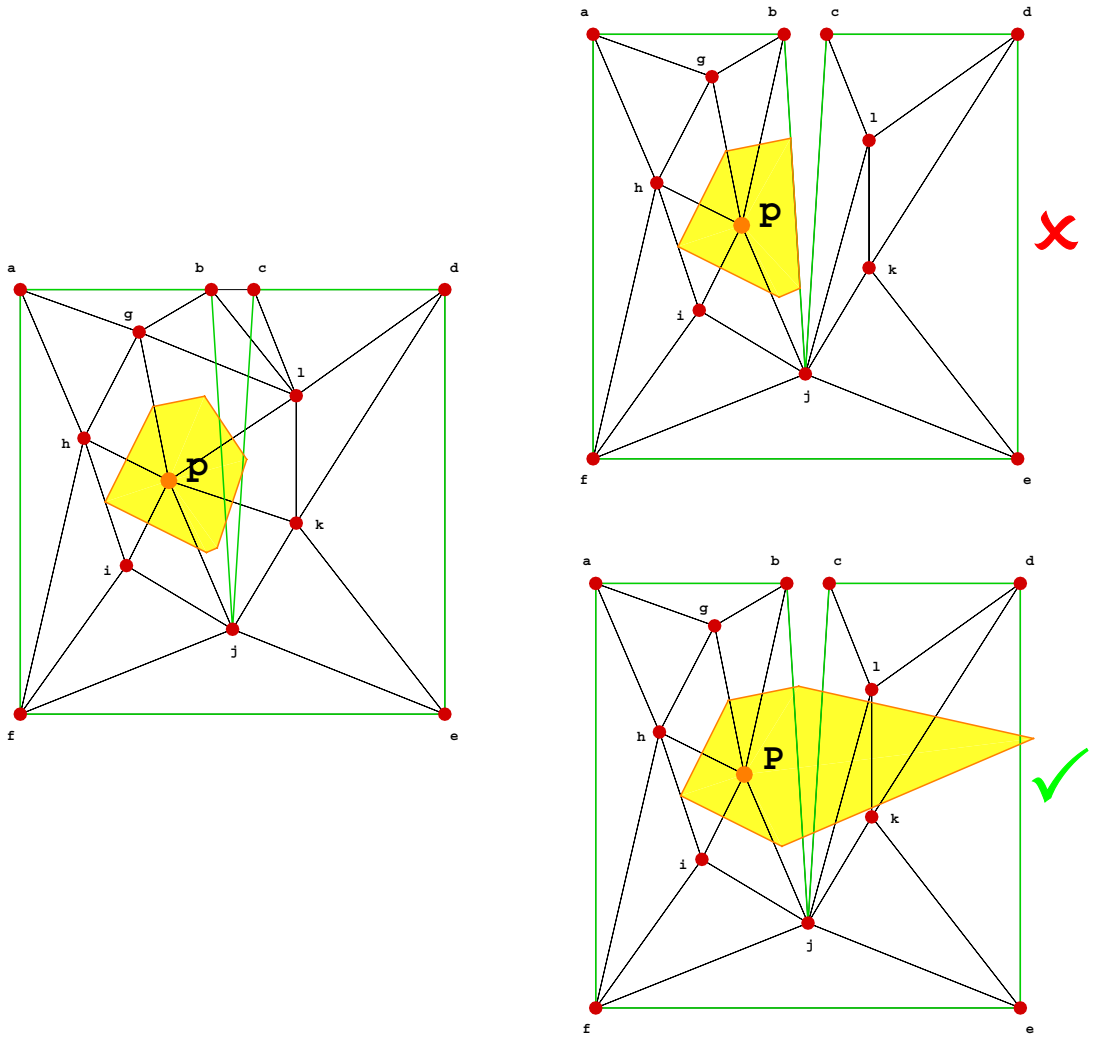

FIG. 2.5 – Région de Voronoï associée à un nœud, non-contrainte*(à gauche)*,contrainte *(à droite)*

2.2.2 Calcul des fonctions de forme CNEM

Soit Ω un domaine dans R^n, Γ sa frontière, et S un ensemble de nœuds répartis sur Ω.

Soit x un point à l'intérieur du domaine Ω, point où l'on veut évaluer l'interpolation, tel que :

- $x \notin S$
- $x \notin \Gamma$

On note :

- DV_Ω^c le diagramme de Voronoï contraint de Ω *(ou plus précisément diagramme de Voronoï associé aux nœuds de S et contraint par la frontière Γ de Ω).*

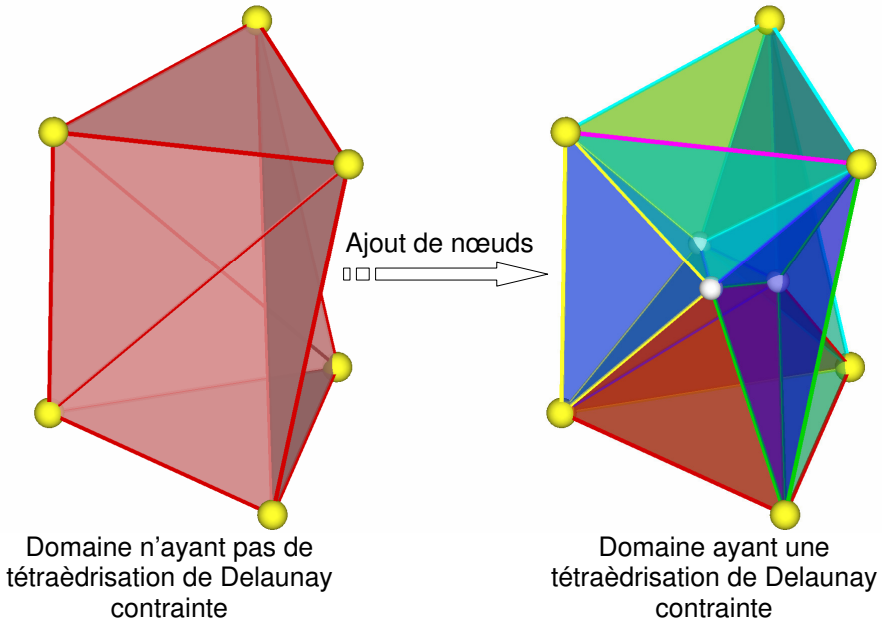

FIG. 2.6 – Tétraèdrisation contrainte de Delaunay du polyèdre de **Schönhardt** par ajout de nœuds

– $DV^c_{\Omega+x}$ le diagramme de Voronoï contraint de Ω avec ajout du point x.
– $TD^c_{\Omega+x}$ la triangulation de Delaunay contrainte de Ω avec ajout du point x.
– c_x la cellule de Voronoï associée au point x dans $DV^c_{\Omega+x}$.

Les voisins naturels du point x sont les nœuds de Ω dont la cellule de Voronoï est voisine directe de la cellule de x dans $DV^c_{\Omega+x}$. Autrement dit ce sont les nœuds de Ω qui sont liés à x par une arête de Delaunay dans $TD^c_{\Omega+x}$.

Soit v_i un tel voisin, et c_{v_i} sa cellule de Voronoï associée dans DV^c_Ω. La valeur en x de la fonction de forme éléments naturels Sibson [Sib80b][Sib80a] [Sib81] associée à ce nœud voisin, fonction notée $\phi^s_{v_i}(x)$, est donnée par :

$$\phi^s_{v_i}(x) = \frac{\|c_x \cap c_{v_i}\|_L}{\|c_x\|_L} \tag{2.3}$$

où $\|.\|_L$ désigne la mesure de Lebesgue *(longueur en 1d, aire en 2d, mesure de volume en 3d...)*. Il s'agit d'une mesure algébrique, elle peut être négative pour des frontières fortement non convexes *(voir 3.3.4)*.

Un exemple de construction de ces fonctions de forme en $2d$ est donné figure 2.7.

Sur la figure 2.8 sont représentées les courbes iso-valeurs d'une fonction de forme associée à un nœud, élément naturel Sibson $2d$ *(à gauche)*, et éléments finis linéaire *(à droite)* construite sur la triangulation de Delaunay duale pour comparaison.

Sur la figure 2.9 sont représentées les surfaces iso-valeurs d'une fonction de forme associée à un nœud, élément naturel Sibson $3d$ *(en haut)*, et éléments finis linéaires construite sur la tétraèdrisation de Delaunay duale *(en bas)* pour comparaison.

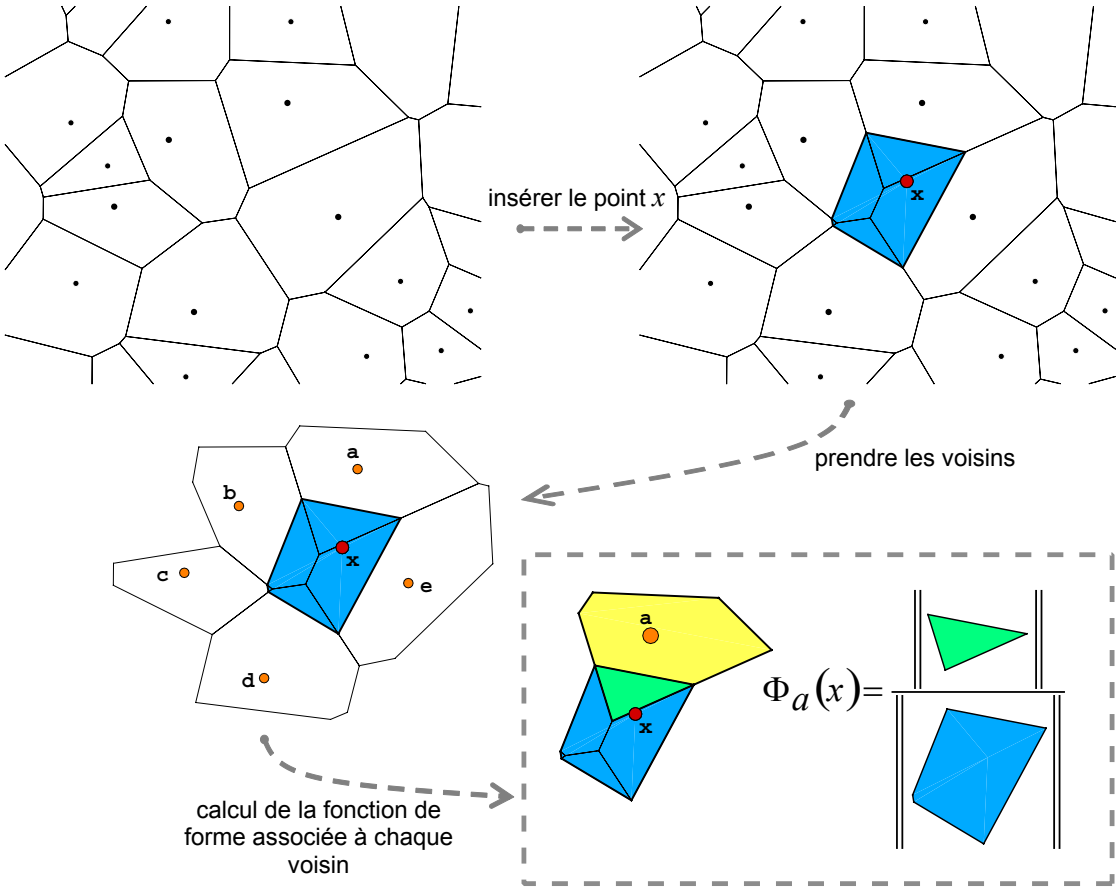

FIG. 2.7 – Calcul des fonctions de forme Sibson

On peut enfin voir figure 2.10, la différence qui résulterait pour un domaine non convexe *2d*, de l'utilisation d'un diagramme de Voronoï non contraint *(à gauche)* et d'un diagramme de Voronoï contraint *(à droite)* pour une fonction de forme Sibson associée à :

- un nœud proche de la frontière *(en haut)*,
- un nœud sur la frontière *(en bas)*.

Il est également possible de définir, pour ce même nœud voisin, une autre fonction forme éléments naturels, dite non Sibson ou Laplace [BS00], au point x, $\phi^l_{v_i}(x)$ qui a pour expression :

$$\phi^l_{v_i}(x) = \frac{\frac{\|f_i\|_L}{\|\vec{xv_i}\|}}{\sum_{j=1}^{n} \frac{\|f_j\|_L}{\|\vec{xv_j}\|}} \tag{2.4}$$

ou f_i désigne la face de c_x associée au nœud voisin v_i, et n le nombre de voisins naturels.

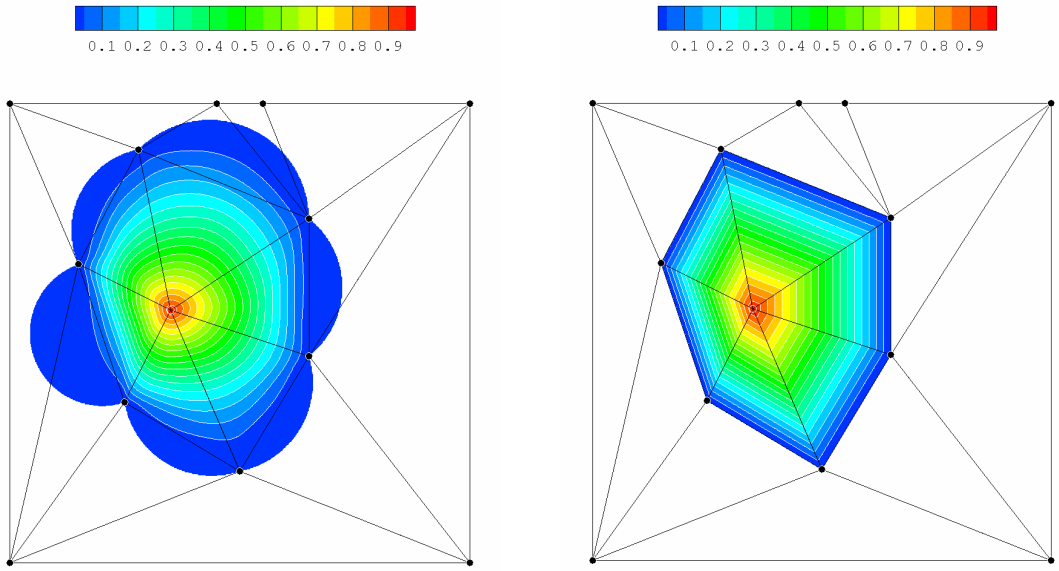

FIG. 2.8 – Fonction de forme en 2d EN Sibson / EF linéaires

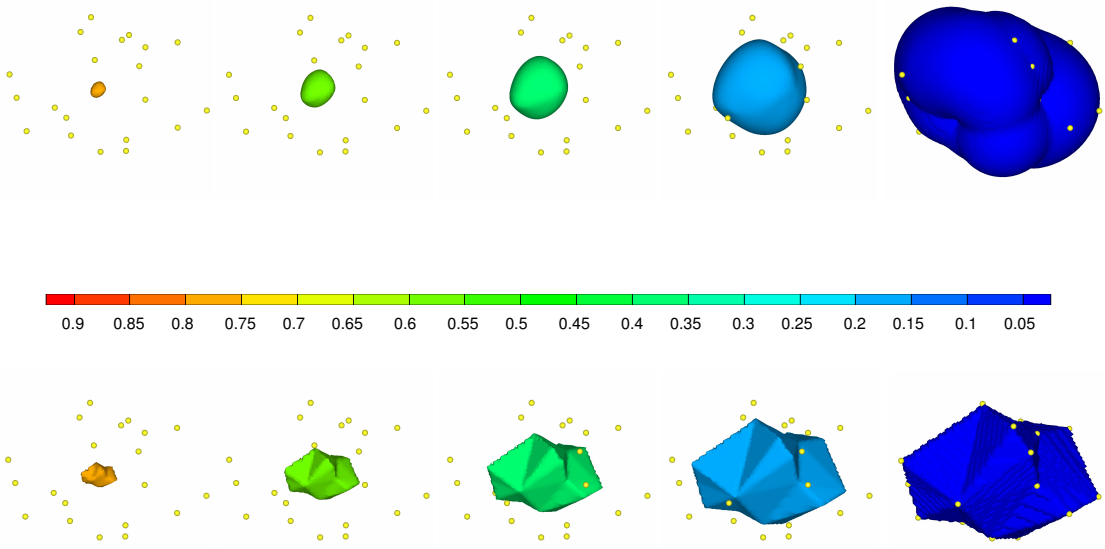

FIG. 2.9 – Fonction de forme en 3*d*, EN Sibson / EF linéaires

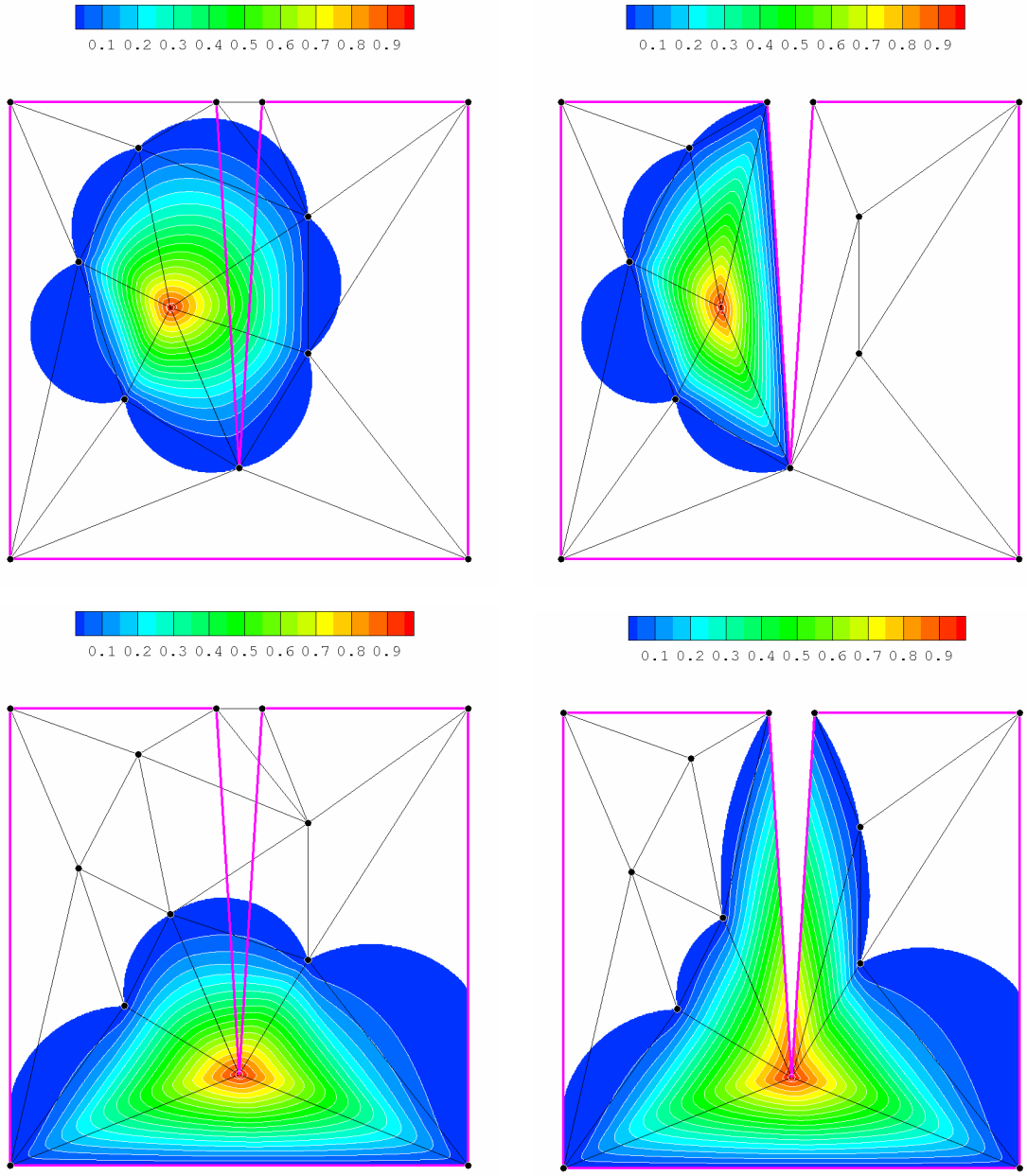

FIG. 2.10 – Résultat de l'utilisation d'un diagramme de Voronoï non contraint / contraint pour le calcul d'une fonction forme Sibson

2.2.3 Calcul du gradient des fonctions de forme CNEM de type Sibson

Piper [Pip93] donne l'expression exacte du gradient de la contribution $\lambda_i\left(\|c_x \cap c_{v_i}\|_L\right)$ d'un nœud voisin v_i donné, et ce en dimension quelconque :

$$\nabla\lambda_i(x) = \frac{\|f_i\|_L}{\left\|\vec{xv_i}\right\|}\,\vec{xC_i} \tag{2.5}$$

avec :

1. f_i : face de c_x associée au nœud voisin v_i.
2. C_i : centroïde *(barycentre)* de la face f_i, défini par la l'équation 2.6.

$$C_i = \frac{1}{\|f_i\|_L} \int_{q \in f_i} q \; dq \tag{2.6}$$

2.3 peut se réécrire :

$$\phi^s_{v_i}(x) = \frac{\lambda_i(x)}{\sum_{j=1}^{n} \lambda_j(x)}, \text{ avec } n \text{ le nombre de voisins.} \tag{2.7}$$

Cette dernière expression permet d'aboutir à l'expression ci-dessous pour le gradient des fonctions de forme Sibson :

$$\nabla\phi^s_{v_i}(x) = \frac{\nabla\lambda_i(x) - \phi^s_{v_i}(x) \sum_{j=1}^{n} \nabla\lambda_j(x)}{\sum_{j=1}^{n} \lambda_j(x)} \tag{2.8}$$

2.2.4 Propriétés des fonctions de forme CNEM

a) Propriété delta Kronecker

Les fonctions de forme éléments naturels sont interpolantes :

$$\phi_{n_i}(n_j) = \delta_{ij} \tag{2.9}$$

b) Partition de l'unité

Par construction les fonctions de forme éléments naturels vérifient la partition de l'unité. En un point x d'un domaine Ω, on a :

$$\sum_{i=1}^{n} \phi_{v_i}(x) = 1, \; \forall \; x \in \Omega, \text{ avec } n \text{ le nombre de voisins du point } x. \tag{2.10}$$

c) Propriété de coordonnées locales et consistance linaire

Sibson[Sib80b][Sib80a][Sib81] et Piper[Pip93] montrent qu'en un point x donné d'un domaine Ω, les fonctions de forme éléments naturels reproduisent exactement les coordonnés géométriques de ce point :

$$x = \sum_{i=1}^{n} \phi_{v_i}(x)\, x_i, \text{ avec } n \text{ le nombre de voisins de } x. \tag{2.11}$$

Les deux propriétés (**b**) et (**c**) impliquent que les fonctions de forme éléments naturels reproduisent exactement les fonctions linéaires.

d) Linéarité sur le bord

Sur la frontière qui contraint le diagramme de Voronoï d'un domaine Ω, les fonctions de forme éléments naturels sont strictement linaires [Yvo04].

Pour rappel, la frontière qui contraint un diagramme de Voronoï d'un domaine Ω, est :
- l'enveloppe convexe des nœuds de Ω, si l'on utilise un diagramme de Voronoï,
- la frontière réelle de Ω, si l'on utilise un diagramme de Voronoï **contraint**.

Ces deux dernières sont confondues pour des domaines convexes.

Les propriétés (**a**) et (**d**) impliquent que sur un segment *(polygone en 3d)* de la frontière, l'interpolation éléments naturels est linéaire et ne dépend que de la valeur du champ aux nœuds de ce segment *(polygone en 3d)*. Il est ainsi possible d'imposer directement les conditions aux limites sur les nœuds de la frontière.

e) Support

Le support d'une fonction de forme éléments naturels associés à un nœud i donné de Ω, est l'ensemble des cercles *(sphère, hyper-sphère)* circonscrits aux triangles *(tétraèdres, simplexes)* de Delaunay connectés à i. Ce support est tronqué par la frontière qui contraint le diagramme de Voronoï du domaine Ω *(voir figure 2.10)*. Sur la figure 2.8 on peut voir que le support d'une fonction de forme éléments naturels est plus étendu que le support d'une fonction éléments finis linéaires qui serait associée au même nœud, et construite sur le maillage de Delaunay dual. Ainsi, contrairement aux éléments finis, pour l'interpolation éléments naturels en $2d$, plus de 3 fonctions de forme peuvent être non nulles en un point *(4 en 3d)*.

f) La continuité

Les fonctions de forme éléments naturels Sibson *(figure 2.11 à droite)* sont C^0 aux nœuds, C^1 sur les cercles *(sphères, hyper-sphères)* de Delaunay, et C^∞ partout ailleurs [HS00].

Les fonctions de forme éléments naturels Laplace *(figure 2.11 à gauche)* sont C^0 aux nœuds et sur les cercles *(sphères, hyper-sphères)* de Delaunay, et C^∞ partout ailleurs [HS00].

2.3 Intégration numérique et gradient stabilisé

Deux types d'intégration sont généralement utilisées dans la NEM, et donc par extension dans la CNEM :

1. l'intégration de Gauss sur les tétraèdres de Delaunay,
2. l'intégration nodale stabilisée conforme **SCNI** *(Stabilized Conforming Nodal Integration)*.

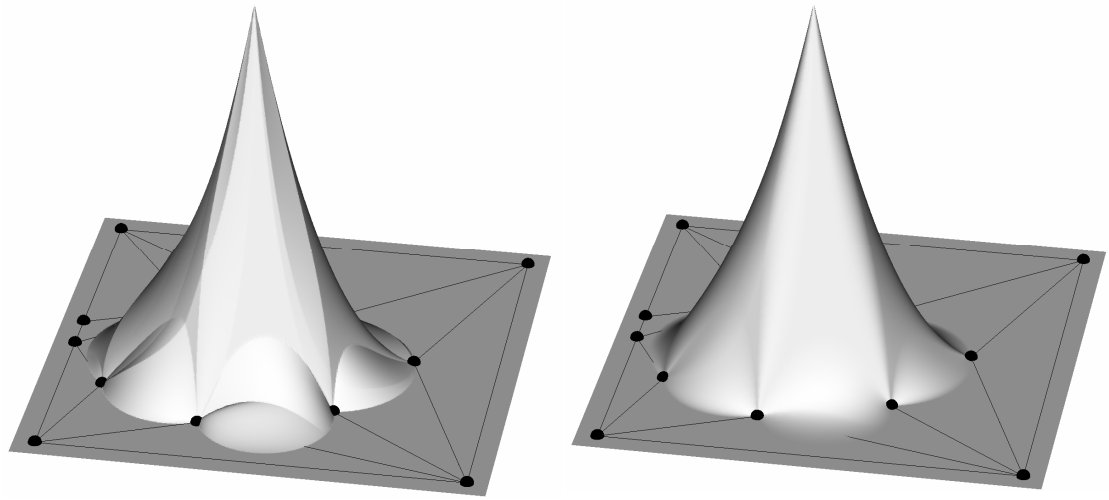

FIG. 2.11 – Fonctions de forme éléments naturels 2d Laplace / Sibson

La première nécessite un nombre élevé de points de Gauss afin de minimiser l'erreur d'intégration. Ceci est dû au fait que les fonctions de forme éléments naturels sont rationnelles *(elles ne peuvent pas être intégrées numériquement de façon exacte par la méthode de Gauss)*.

La seconde, l'intégration nodale stabilisée conforme **SCNI**, a été introduite par Chen [CWYY01][CYW02] pour des approximations sans maillage de type RK dans la méthode RKPM [LJZ95], afin d'améliorer l'intégration nodale directe. Elle fut reprise par Cueto [GCMD04] pour des approximations de type éléments naturels dans la NEM.

L'intégration **SCNI** consiste à substituer le gradient ∇A d'un champ interpolé A, en chaque nœud n_i d'un domaine Ω, par un gradient moyen *(appelé gradient stabilisé)* $\tilde{\nabla} A_i$. Ce gradient moyen est calculé sur un sous-domaine ω_i de Ω, qui entoure le nœud n_i :

$$\tilde{\nabla} A_i = \frac{1}{\|\omega_i\|} \int_{\omega_i} \nabla A d\omega_i \tag{2.12}$$

En utilisant le théorème de la divergence, 2.12 peut se simplifier en :

$$\tilde{\nabla} A_i = \frac{1}{\|\omega_i\|} \int_{\Gamma\omega_i} A \, \vec{n} \, d\Gamma\omega_i \tag{2.13}$$

avec :
- $\Gamma\omega_i$: le bord de ω_i
- $\vec{n}$: la normale extérieure à $\Gamma\omega_i$

La formule 2.13 permet :

– de passer d'une intégrale volumique à une intégrale surfacique,
– d'intégrer directement le champ plutôt que son gradient.

Pour le calcul du gradient stabilisé, il est nécessaire d'introduire une discrétisation du domaine Ω en sous-domaines "nodaux" entourant chaque nœud. La façon d'obtenir une telle discrétisation est abordé dans le paragraphe 3.4.

L'intégration stabilisée permet d'associer un gradient unique au sous-domaine ω_i. C'est ce gradient qui sera utilisé par la suite pour le calcul des déformations et contraintes moyennes associées à chaque nœud du domaine. Cette démarche permet, dans un contexte de grandes transformations, d'éviter la projection de champs entre deux configurations lors d'une réactualisation, toutes les variables *(déplacement, déformation, contrainte,...)* étant définies aux nœuds *(colocation)*.

2.4 Conclusion

Ce chapitre nous a permis de définir les notions de base de l'interpolation CNEM. Le chapitre qui suit va se concentrer sur la mise en œuvre, en 3d, des algorithmes utilisés pour la construction du DVC, le calcul des fonctions de forme CNEM ainsi que pour la discrétisation en vue de l'intégration SCNI.

Chapitre 3

Mise en œuvre de la CNEM en 3d

L'objet de ce chapitre est de détailler, dans un contexte **tridimensionnel**, les trois principaux aspects de la mise en œuvre de la CNEM, que sont, la construction du DVC, le calcul des fonctions de forme Sibson, et la discrétisation du domaine en vue de l'intégration SCNI.
On retrouvera ainsi, exposé dans le paragraphe 3.1, l'organisation choisie pour représenter de façon topologique le diagramme de Voronoï contraint qui est à la base de l'approche CNEM.
Ce diagramme de Voronoï contraint est déduit d'une tétraèdrisation de Delaunay contrainte réalisée à l'aide du code TetGen [HS06] externe au notre. Après une présentation rapide de la démarche utilisée dans TetGen paragraphe 3.2.1, l'algorithme permettant de passer de la tétraèdrisation contrainte au diagramme de Voronoï est donné dans le paragraphe 3.2.2.
Cela établi, les différentes approches possibles pour le calcul des fonctions de forme CNEM Sibson sont présentées dans le paragraphe 3.3. Une comparaison des performances de ces différentes approches ainsi que les particularités de l'interpolation CNEM au voisinage des zones non convexes d'un domaine sont également données dans ce paragraphe.
On abordera finalement dans le paragraphe 3.4 la discrétisation du domaine en vu de l'intégration SCNI. Certains aspects problématiques dans le cas de domaine fortement non convexe sont mis en évidence et résolus.

3.1 Structuration adoptée pour les données du diagramme de Voronoï contraint

Pour décrire en détail le diagramme de Voronoï, et pour faciliter l'écriture des algorithmes géométriques, nous avons choisi de le représenter de façon topologique. Les entités de bases dont est composé un diagramme de Voronoï en *3d* sont de quatre types : **sommet**, **arête**, **face**, **cellule** *(les **nœuds** ne sont pas considérés comme des entités de Voronoï)*. Les entités de même type sont stockées dans une même liste.

a) L'entité sommet :

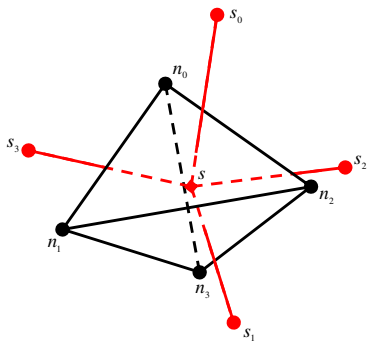

FIG. 3.1 – Conventions pour les sommets

Chaque sommet est connecté à :
- ses 4 nœuds générateurs *(Bowyer nœud)*[Wat81], nœuds du tétraèdre dual,
- ses 4 sommets voisins *(Bowyer sommet)*[Wat81], les tétraèdres voisins du tétraèdre dual,
- ses coordonnées, centre de la sphère circonscrite au tétraèdre.

Soit s le sommet considéré :
- Ses 4 nœuds générateurs $n_0,\ n_1,\ n_2,\ n_3$ sont ordonnés de manière à ce que la normale à la face du tétraèdre dual à s formée par les nœuds $n_0,\ n_1,\ n_2$, $\overrightarrow{n} = \overrightarrow{n_0\,n_1} \wedge \overrightarrow{n_0\,n_2}$, pointe vers l'extérieur du tétraèdre.
- Ses 4 sommets voisins $s_0,\ s_1,\ s_2,\ s_3$ sont ordonnés de manière à ce que s_i, pour $i \in \{0, 1, 2, 3\}$, soit le sommet du tétraèdre dont les 3 nœuds $n_{(i)},\ n_{(i+1)mod(4)}$, $n_{(i+2)mod(4)}$ (mod : modulo) soient les nœuds en commun avec le tétraèdre dual à s.

Sommet à l'infini : Un sommet *(tétraèdre)* à l'infini est un sommet qui ne possède que trois nœuds, nœuds d'un triangle de la frontière du domaine. Le quatrième qui est à l'infini, dirigé vers l'extérieur, aura comme index (-1) dans la structure de données associée au sommet. Ce sommet à l'infini ne possède qu'un seul sommet voisin, sommet ayant en commun les trois mêmes nœuds, les trois autres sommets voisins auront

comme index (-1) dans sa structure de données. Enfin ce sommet à l'infini n'a pas de coordonnées, mais se situe à l'infini sur l'arrête de Voronoï ayant comme nœuds les trois nœuds du sommet.

b) L'entité arête :

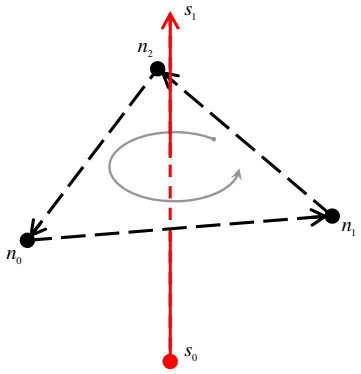

FIG. 3.2 – Conventions pour les arêtes

Chaque arête est connectée à :
- ses trois nœuds générateurs, nœuds sommets de la face de tétraèdre associée à cette arête *(face de tétraèdre duale de l'arête)*,
- ses deux sommets.

Soit a l'arête considérée. Ses 3 nœuds générateurs $n_0,\ n_1,\ n_2$ et ses deux sommets $s_0,\ s_1$ sont ordonnés de manière à ce que :
- la normale à la face de tétraèdre duale à a, formée par les nœuds $n_0,\ n_1,\ n_2$, $\overrightarrow{n} = \overrightarrow{n_0\,n_1} \wedge \overrightarrow{n_0\,n_2}$, pointe dans le même sens que le vecteur $\overrightarrow{s_0\,s_1}$

Remarque : $n_0,\ n_1,\ n_2$ ne sont pas stockés, car ils se déduisent directement à partir de s_0 et s_1.

c) L'entité face :

Chaque face est connectée à :
- ses deux nœuds générateurs, nœuds de l'arête de tétraèdre duale,
- sa liste d'arêtes signées et chaînées.

Soit f la face considérée. Ses deux nœuds générateurs $n_0,\ n_1$ et sa liste d'arêtes $a_0,\ a_1,\ \ldots\ a_n$ sont ordonnés de manière à ce que :
- les arêtes soient chaînées et que le contour formé par la liste d'arêtes s'enroule dans le sens direct autour de l'axe orienté $(n_0,\ n_1)$.

Dans la liste des arêtes, les index sont signés. Une arête sera considérée comme étant positive, si elle est orientée dans le même sens qu'est orienté le contour.

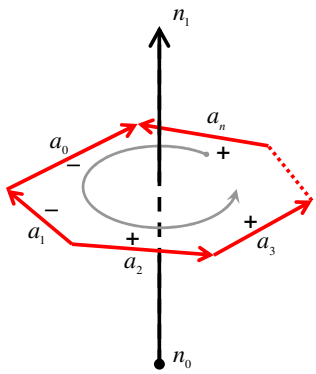

FIG. 3.3 – Conventions pour les faces

d) L'entité cellule :

Chaque cellule est connectée à :
– son nœud générateur,
– sa liste de faces signées.

Une face sera considérée comme étant positive, si sa normale $\overrightarrow{n} = \overrightarrow{n_0\, n_1}$ (n_0, n_1 nœuds de la face) pointe vers l'extérieur de la cellule.

La figure 3.4 donne une représentation synthétique de la structure des données adoptée pour le diagramme de Voronoï.

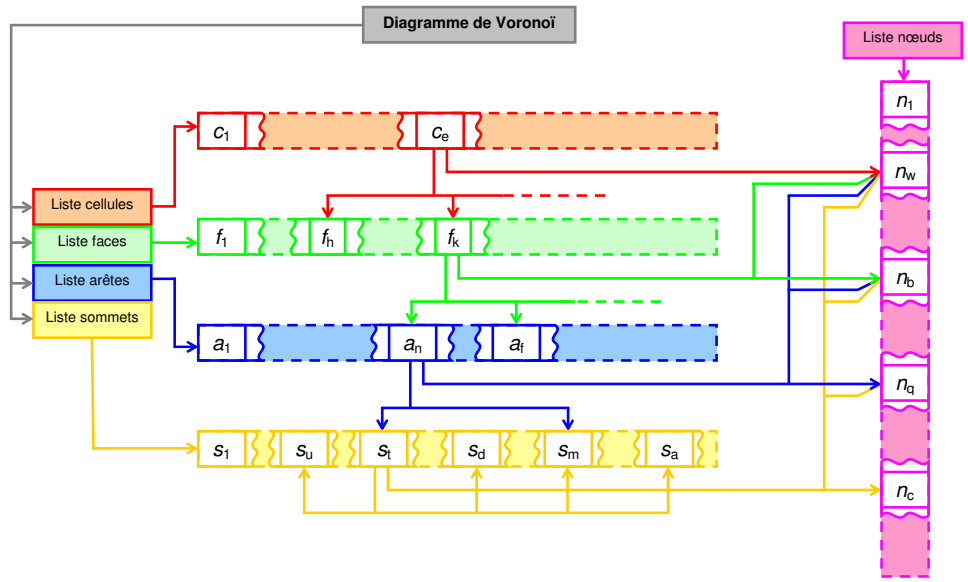

FIG. 3.4 – Connexion entre les différentes entités du diagramme de Voronoï

3.2 Construction du diagramme de Voronoï contraint

La construction du diagramme de Voronoï contraint se fait par le biais de son dual, la tétraèdrisation de Delaunay contrainte.

3.2.1 Construction de la tétraèdrisation de Delaunay contrainte d'un domaine $3d$

Cette construction est réalisée grâce à l'algorithme proposé par Si et Gärtner [SG05]. Les données d'entrée de cet algorithme sont les nœuds du domaine, et sa frontière décrite sous forme de PLC (*Piecewise Linear Complexe*). Un PLC consiste en une description linéaire par morceaux d'une frontière [MTT+96]. Pour un domaine $3d$ cette description est composée d'un ensemble de faces polygonales planes *(pouvant être non convexes)*, de segments, et de nœuds *(un exemple de PLC est donnée figure 3.5)*, tels que :

- deux faces ne peuvent être que disjointes ou avoir uniquement en commun un ensemble de segments et/ou de nœuds,
- deux segments ne peuvent être que disjoints ou avoir en commun un unique nœud.

Une triangulation est un PLC très simple où toutes les faces sont des triangles.

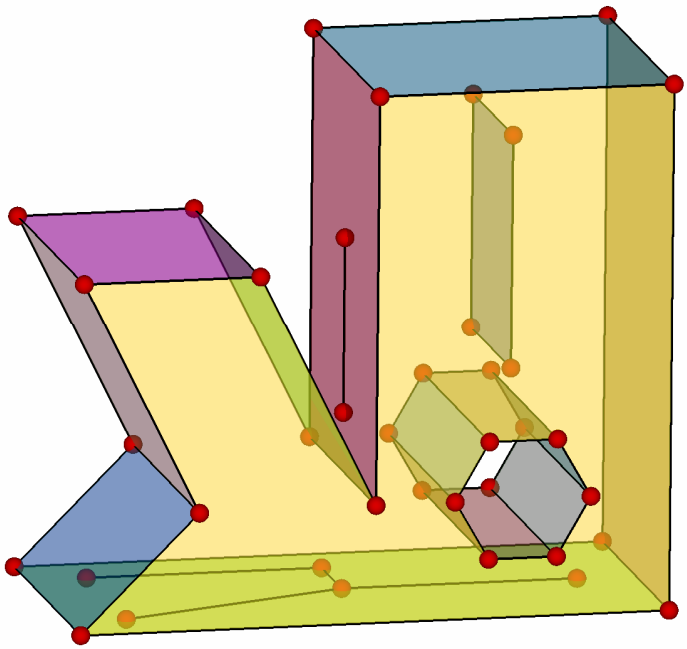

FIG. 3.5 – Exemple de PLC

L'algorithme proposé par Si et Gärtner comprend quatre étapes principales, et se base sur le théorème qu'ils énoncent :

Théorème 1. *Soit X un PLC, si la tétraèdrisation de Delaunay des nœuds de X contient tous les segments de X, et ne contient aucune dégénérescence locale, alors la tétraèdrisation de Delaunay contrainte de X existe.*

Définition : Il y a dégénérescence locale si cinq nœuds ou plus de X se situent sur une même sphère sans qu'aucun autre nœud ne soit à l'intérieur de cette sphère.

Etape 1 : Construction de la tétraèdrisation de Delaunay des nœuds de X

Cette construction se fait par le biais d'algorithmes incrémentaux standard *(voir[Ber94])*. Soit D_1 cette tétraèdrisation.

Etape 2 : Traitement des segments de X non compatibles

Tous les segments de X ne sont pas forcement des arêtes de D_1. Le but de cette étape est d'insérer de manière incrémentale des nœuds dans X *(modification du PLC, et en particulier modification des segments concernés par ajout d'un ou plusieurs nœuds sur ces derniers)* de manière à obtenir une tétraèdrisation de Delaunay D_2 pour laquelle tous les segments du nouveau PLC X' sont des arêtes de D_2.

Etape 3 : Suppression des dégénérescences locales

Afin de pouvoir appliquer le théorème 1, les dégénérescences locales doivent être supprimées. Cette suppression se fait par perturbation locale de la position de certains nœuds, ou par insertion d'un nœud le cas échéant. En effet, certains nœuds de X ne peuvent être perturbés sans compromettre la description linéaire de la frontière, par exemple, pour un noeud se situant à l'intersection de 3 faces de X. Soit X'' ce nouveau PLC et D_3 la tétraèdrisation de Delaunay associée.

Etape 4 : Récupération des faces

A la suite de l'étape 3, le théorème 1 garantit l'existence de la tétraèdrisation contrainte de X''. D_3 n'est pas cette tétraèdrisation car certaines faces de X'' peuvent ne pas être entièrement formées par l'union de faces triangulaires de D_3 : certains tétraèdres de D_3 peuvent transpercer des faces de X'', car jusqu'à présent nous n'avons pas introduit le critère de visibilité *(voir 2.2.1)*.

La suppression des tétraèdres transperçant une face F de X'' crée une ou plusieurs cavités, chacune d'entres elles provient du retrait de tétraèdres adjacents et peut ne pas être convexe. Chaque cavité est ensuite séparée en deux par F *(introduction du critère de visibilité : les faces de X'' sont opaques)* et un maillage de Delaunay est réalisé pour chacune des deux moitiés à partir des nœuds de chacune d'entres elles. Seuls les tétraèdres internes aux demie-cavités sont conservés.

Il est à noter que la sphère circonscrite à certains tétraèdres d'une demi-cavité peut contenir des nœuds de l'autre demie-cavité, car ils ne sont plus visibles.

Une fois toutes les faces F traitées, une tétraèdrisation D_4 est obtenue pour laquelle toutes les faces de X'' sont formées par des ensembles de faces triangulaires de D_4.

Pour finir les tétraèdres extérieurs au domaine sont supprimés.

La tétraèdrisation D_4 est la tétraèdrisation de Delaunay contrainte de X''.

L'algorithme a été implémenté par les auteurs dans un code c++*(TetGen[HS06])*, que nous utilisons pour réaliser cette tétraèdrisation contrainte.

Dans le cadre des grandes transformations, les nœuds ajoutés feront l'objet d'un traitement spécifique précisé dans le chapitre 5. Le calcul des fonctions de forme CNEM 3.3 se base sur D_4 garantissant ainsi toutes les propriétés énoncées en 2.2.4.

3.2.2 Passage de la tétraèdrisation de Delaunay contrainte au diagramme de Voronoï contraint

Une fois la tétraèdrisation contrainte établie, la construction de la base de données topologique du diagramme de Voronoï adoptée est directe.

La donnée d'entrée de l'algorithme qui effectue cette tâche, donnée produite par le code TetGen[HS06], est la liste des sommets *(tétraèdres)*, avec pour chaque sommet :

- ses coordonées *(centres de la sphères circonscrites au tétraèdre de Delaunay dual)*
- sa connexion aux 4 nœuds de son tétraèdre dual *(Bowyer nœud [Bow81])*,
- sa connexion aux 4 sommets voisins, sommets des 4 tétraèdres voisins au tétraèdre du sommet en question *(Bowyer sommet [Bow81])*.

L'algorithme de remplissage de la base de données du diagramme de Voronoï a été conçu de manière à ne créer qu'une fois chaque entité du diagramme de Voronoï tout en limitant les coûts générés par la vérification de l'existence d'une entité avant de la créer. Ainsi, nous savons qu'à partir de chaque sommet il est possible de créer *(cf figure 3.6)* :

→ *4 arêtes* : arêtes reliant ce dernier aux quatre sommets voisins. Chacune des ces arêtes est susceptible d'être créée deux fois. Afin d'éviter cela nous ne créerons une arête qui si l'index du sommet considéré est supérieur à l'index du sommet voisin.

→ *6 faces* : chaque face est commune à tous les sommets ayant les deux nœuds de la face en commun *(grappe de tétraèdress s'enroulant autour de l'arête de Delaunay formée par les deux nœuds de la face)*. Chaque face est donc successible d'être créée autant de fois qu'il y a de ces sommets *(nombre de tétraèdres de la grappe)*. Une solution à moindre coût est de vérifier dans la liste des faces de la cellule associée à l'un des deux nœuds de la face considérée *(peu importe le choix)*, si elle n'a pas déjà été créée. Nous évitons ainsi la vérification dans la liste globale des faces du diagramme qui a une complexité en $O(n^2)$.

Remarque : la liste des cellules forme implicitement une table de hachage des faces, la clé d'accès associée à une face donnée est l'index de l'un de ses deux nœuds. Afin de minimiser la taille des listes des faces qui ont la même clé, qui est dans notre cas de l'ordre de la dizaine *(nombre de face d'une cellule)*, on pourrait construire une table de hachage des faces explicitement, en associant aux faces une clé basée sur les indexes de ses deux nœuds.

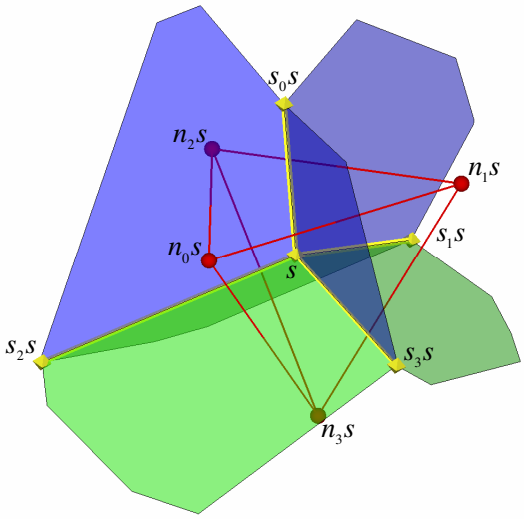

FIG. 3.6 – Arêtes et faces connectées à un sommet s

Algorithme :

Données d'entrée :
– La tétraèdrisation de Delaunay contrainte.
Données de sortie :
– Le diagramme de Voronoi contraint.

Pour décrire l'algorithme de remplissage de la base de données nous introduisons les notations et conventions suivantes :
– L'indexation commence à partir de 0
– n : nœud
– s : sommet de Voronoï
– a : arête de Voronoï
– f : face de Voronoï
– c : cellule de Voronoï
– xy, avec x,y $\in \{n, s, a, f, c\}$: entité x de l'entité y
– x_iy : $i^{ème}$ entité x de l'entité y. Si la dernière entité *(la plus à droite)* comporte un indice, il s'agit de son numéro *(index)* dans la structure de données du diagramme de Voronoï.
– *mod* : modulo

Exemples :
– n_is : $i^{ème}$ nœud du sommet s
– a_if_jc : $i^{ème}$ arête la $j^{ème}$ face de la cellule c
– n_ia_j : $i^{ème}$ nœud de la $j^{ème}$ arête du diagramme de Voronoï
– s_is : $i^{ème}$ sommet du sommet s *($i^{ème}$ sommet voisin au sommet s)*

Pour chaque sommet s^v de la TDC **faire**
  créer un tableau tampon A de 4 arêtes
  Pour i **de** 0 **à** 3 **faire**
    Si $(index(s^v) > index(s_j s^v))$ **Alors**
      créer une arête a^v
      $s_0 a^v \leftarrow s_i$; $s_1 a^v \leftarrow s_i s^v$; $A[i] \leftarrow a^v$
    Fin Si
  Fin Pour
  Pour i **de** 0 **à** 2 **faire**
    Si $((A[0]$ créé$)\vee(A[(i+2)mod(3)+1]$ créé$))$ **Alors**
      créer une face tampon f
      Si (dans $cn_i s^v$ il existe une face ayant comme un de ses deux nœuds $n_{(i+1)mod(3)} s^v$) **Alors**
        affecter cette face à f^v
      Sinon
        $n_0 f^v \leftarrow n_i s^v$; $n_1 f^v \leftarrow n_{(i+1)mod(3)} s^v$
        ajouter f^v à $cn_i s^v$; ajouter $-f^v$ à $cn_{(i+1)mod(3)} s^v$
      Fin Si
      Si $(A[0]$ créé$)$ **Alors**
        ajouter $A[0]$ à f^v
      Fin Si
      Si $(A[(i+2)mod(3)+1]$ créé$)$ **Alors**
        ajouter $-A[(i+2)mod(3)+1]$ à f^v
      Fin Si
    Fin Si
  Fin Pour
  Pour i **de** 0 **à** 2 **faire**
    Si $((A[(i+1)mod(3)+1]$ créé$)\vee(A[(i+2)mod(3)+1]$ créé$))$ **Alors**
      créer une face tampon f^v
      Si (dans $cn_3 s^v$ il existe une face ayant comme un de ses deux nœuds $n_i s^v$) **Alors**
        affecter cette face à f^v
      Sinon
        $n_0 f^v \leftarrow n_3 s^v$; $n_1 f^v \leftarrow n_i s^v$
        ajouter f^v à $cn_3 s^v$; ajouter $-f^v$ à $cn_i s^v$
      Fin Si
      Si $(A[(i+1)mod(3)+1]$ créé$)$ **Alors**
        ajouter $A[(i+1)mod(3)+1]$ à f^v
      Fin Si
      Si $(A[(i+2)mod(3)+1]$ créé$)$ **Alors**
        ajouter $-A[(i+2)mod(3)+1]$ à f^v
      Fin Si
    Fin Si
  Fin Pour
Fin Pour

Algorithme 1: Construction de la topologie du DVC à partir de la TDC

3.3 Calcul des fonctions de formes CNEM de type Sibson

Le calcul des fonctions de formes CNEM de Type Sibson comporte deux étapes principales. La première étape concerne la modification locale du diagramme de Voronoï contraint suite à l'ajout du point x, cette étape est présentée au paragraphe 3.3.1. La seconde concerne le calcul de la mesure du volume commun à la cellule associée au point x, et chaque cellule associée à un nœud voisin de x *(au sens Voronoï dans le nouveau diagramme)* avant l'insertion du point x. Ce calcul peut se faire par différentes approches. Celles-ci sont présentées dans le paragraphe 3.3.2 et leurs performances sont comparées au paragraphe 3.3.3. Une illustration de la démarche complète est donnée figure 3.7. Enfin, certains phénomènes associés aux fonctions de formes CNEM apparaissant en 3d pour des domaines non convexes sont mis en évidence au paragraphe 3.3.4.

3.3.1 Etape-1 : Insertion du point x dans le diagramme de Voronoï contraint existant

Vu que nous voulons insérer un point dans un diagramme existant, l'utilisation d'un incrément d'algorithme incrémental s'impose. Deux algorithmes incrémentaux, concourants dans leur démarche, sont à notre disposition [Ber94], celui de Bowyer [Bow81] qui travaille sur les sommets, et celui de Watson [Wat81] qui travaille sur les tétraèdres. Ces derniers ont été élaborés pour la construction de diagramme de Voronoï **non contraint**, mais en les utilisant de façon incrémental sur un diagramme de Voronoï contraint existant, ils maintiennent le caractère contraint du diagramme.

Chaque incrément de ces deux algorithmes se décompose en deux étapes principales :

1. Recherche de tous les tétraèdres dont la sphère circonscrite contient le point x, et les casser. Cette étape se décompose elle-même en deux autres :
   (a) Recherche du tétraèdre qui contient le point x *(voir détail juste en dessous)*
   (b) Recherche de tous les tétraèdres dont la sphère circonscrite contient le point x : cette recherche est locale, elle se fait en "visitant" les tétraèdres qui sont à proximité du premier.
2. Mailler la cavité ainsi obtenue en étoilé par rapport au point x, et mettre à jour les relations de voisinage entre les tétraèdres.

Recherche du tétraèdre qui contient le point x

Pour cela il est possible d'utiliser la descente en gradient basée sur le graphe de Voronoï proposée par Schmitt et Borouchaki [SB90], ou celle basée sur le graphe de Delaunay proposée par Green et Sibson [GS78]. Dans le cas de domaines non convexes ces deux recherches peuvent échouer si l'on démarre la recherche à partir d'un tétraèdre non visible du point x. Il donc nécessaire de mettre en place d'autres types de recherches, comme par exemple par "tache d'huile", ou par octree.

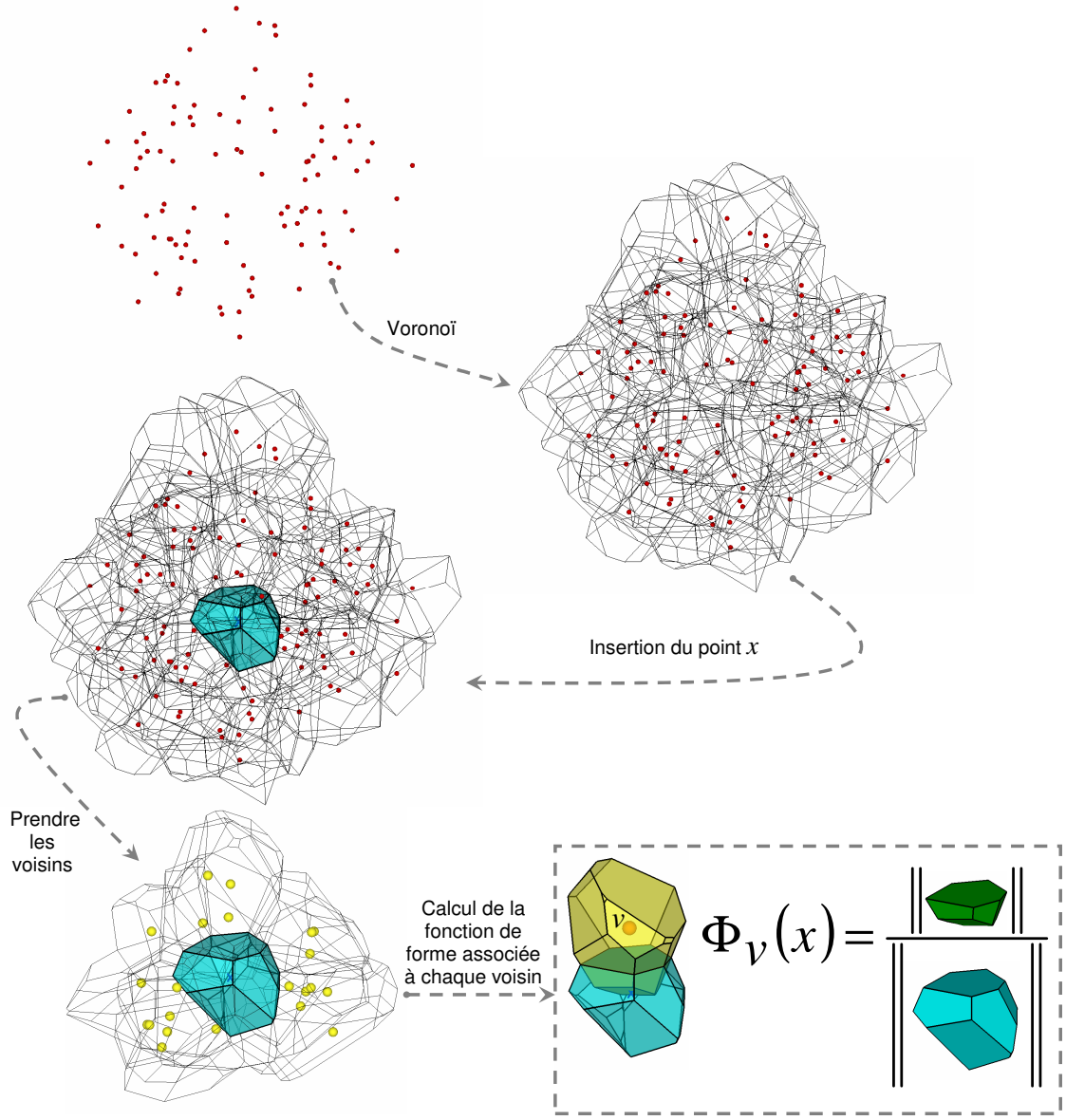

FIG. 3.7 – Calcul des fonctions de formes NEM Sibson

Remarque concernant le problème des tétraèdres plats

Des tétraèdres plats peuvent apparaître lors du remaillage de la cavité en étoilé. Ceci se produit lorsque le point x est sur le cercle circonscrit à une face de tétraèdre de Delaunay *(ou s'il en est très proche)*. Suivant la tolérance que l'on prend pour le test de la sphère vide, on peut éliminer les deux tétraèdres Ta et Tb adjacents *(figure 3.8)*, ou seulement l'un d'entre eux *(alors que les deux doivent être éliminés)*, auquel cas, la face commune à ces derniers sera une face de la cavité, ce qui aura pour conséquence la création d'un tétraèdre de Delaunay plat *(ou quasi plat)* lors du remaillage en étoilé par rapport au point x. La recherche des coordonnées du sommet associé à un tel

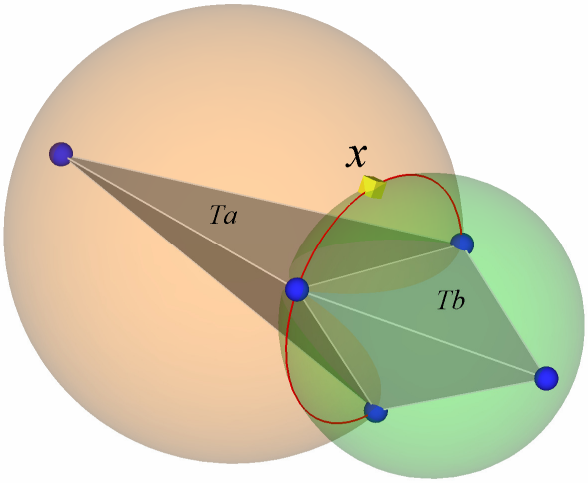

FIG. 3.8 – Cas de figure pouvant conduire à la création d'un tétraèdre plat

tétraèdre mènera à une indétermination *(une infinité de sphères passent par ce cercle)*. Il est possible d'éviter de créer de tels tétraèdres, pour cela, il suffit de vérifier avant de remailler la cavité, s'il y a une *(ou plusieurs)* face*(s)* de cette dernière susceptible*(s)* de créer un tétraèdre plat *(point très proche du plan formé par les 3 nœuds de la face)*. Si de telle faces sont trouvées, on ne peut qu'être en présence du cas précédemment cité *(point x proche du cercle circonscrit à la face, preuve ci-dessous)*, et le tétraèdre adjacent à cette face *(celui qui n'a pas été supprimé)* peut être donc supprimé, chose que l'on fera*(sans oublier de mettre à jour la cavité)*. Ce n'est qu'après avoir vérifié toutes les faces de la cavité que le remaillage en étoilé sera effectué.

Preuve

Imaginons qu'il y ait une face de la cavité pour laquelle le point x appartient au plan passant par les trois nœuds de la face *(ou s'il en est très proche)*, trois cas de figures sont possibles :

1. le point x est à l'extérieur du cercle circonscrit à la face,
2. le point x est à l'intérieur du cercle circonscrit à la face,
3. le point x est sur le cercle circonscrit à la face *(ou très proche de ce cercle)*.

La première proposition est impossible, car le second tétraèdre partageant la face n'aurait pas pu être éliminé, vu que la sphère qui le circonscrit ne peut contenir le point x, ce qui implique que cette face n'aurait pas fait partie de la cavité.
La seconde proposition l'est aussi, car dans ce cas le point x serait à l'intérieur de la sphère circonscrite au tétraèdre, donc ce dernier aurait dû être éliminé, du coup cette face n'aurait pas fait partie de la cavité.
Finalement le point x ne peut être que sur le cercle circonscrit à la face.

3.3.2 Etape-2 : Calcul de la mesure du volume commun à $\acute{c}_x$ et c_v

Une fois le point x inséré dans le diagramme, il nous faut calculer la mesure du volume commun à la cellule $\acute{c}_x$ associée au point x, et la cellule c_v associée à chaque nœud v voisin de x *(dans le nouveau diagramme)* avant l'insertion de x, volume que l'on nommera dans tout ce qui suit par le polyèdre P^{int}. Afin de calculer cette mesure nous pouvons procéder de plusieurs façons :

a) Par l'algorithme récursif de *Lasserre*

Cueto et al [AYCC06] ainsi que Sambridge et al [SBM96][SBM95], proposent d'utiliser l'algorithme récursif de Lasserre, qui n'a besoin pour le calcul de la mesure de volume d'un polyèdre *convexe* dans R^n que de l'ensemble des demi-plans le définissant, s'affranchissant ainsi d'une description topologique explicite de ce dernier.

Dans le cas d'un domaine convexe, les cellules de Voronoï des nœuds de ce dernier sont convexes, le polyèdre P^{int} étant par conséquence convexe *(intersection de deux polyèdre convexes)* peut être alors décrit comme un volume délimité par un ensemble fini de demi plan *(H-représentations)*. Pour un nœud voisin v_i donné de x, cet ensemble est constitué de toutes les faces de c_{v_i}, cellule du nœud v_i avant insertion du point x, et de la face commune aux nœuds v_i et x (*voir figure 3.9)*. Le polyèdre P^{int} étant décrit par une *(H-représentations)*, nous pouvons dès lors calculer la mesure de son volume en utilisant l'algorithme récursif de *Lasserre* :

Soit un polyèdre convexe P dans R^n, délimité par m demi-plans, chacun de ces derniers peut être représenté par une inéquation, ce qui peut se traduire sous une forme matricielle par l'inéquation :

$$\underline{\underline{A}} \cdot \underline{x} \leq \underline{b} \tag{3.1}$$

où $\underline{\underline{A}}$ est une matrice $m \times n$, $\underline{x}$ la colonne des n composantes du point x, et $\underline{b}$ une colonne de m termes.
La mesure de volume de P, noté $V(n, \underline{\underline{A}}, \underline{b})$, peut être alors calculée par la formule suivante :

$$V(n, \underline{\underline{A}}, \underline{b}) = \frac{1}{n} \sum_{i=1}^{m} \frac{b_i}{\|\underline{a}_i\|} V_i(n-1, \underline{\underline{A}}, \underline{b}) \tag{3.2}$$

Avec : $V_i(n-1, \underline{\underline{A}}, \underline{b})$ la mesure de volume dans R^{n-1} de la face f_i :

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

Remarque : 3.2 peut être mise sous la forme bien connue en géométrie $2d$, $3d$:

$$V(P) = \frac{1}{n} \sum_{i=1}^{m} Dist(o, f_i) \cdot V(f_i) \tag{3.3}$$

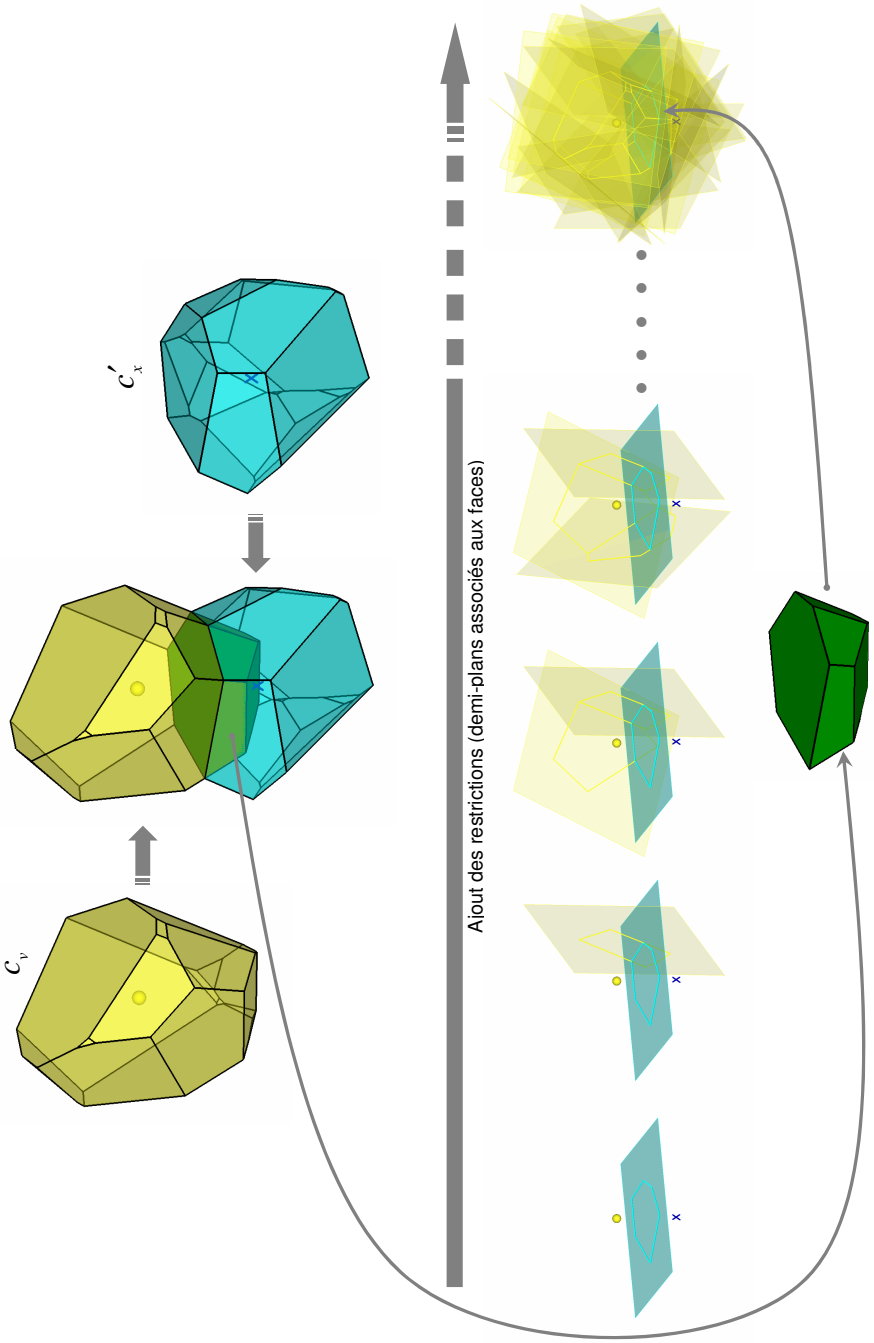

FIG. 3.9 – Etape-2 : Calcul de la mesure de volume du polyèdre P^{int} par Lasserre

Avec : o un point quelconque, et $Dist(o, f_i)$ la ditance signée de o à la face f_i ($\vec{op} \cdot \vec{n}$, avec p un point de f_i, et $\vec{n}$ la normale extérieure unitaire de f_i).

$V_i(n-1, \underline{\underline{A}}, \underline{b})$ ne peut être calculé directement, car nous ne connaissons pas explicitement les sommets du polyèdre définissant la face f_i, Lasserre propose donc de

projeter cette dernière dans un sous-espace de dimension $n-1$:

La face f_i étant définie en fixant la $i^{ème}$ contrainte dans le système 3.1 ($\underline{a_i}^T \cdot \underline{x} = b_i$). On choisit un pivot $a_{ij} \neq 0$, afin de substituer la variable $x_j = b_i - \sum_{k=1,k\neq j}^{n} \frac{a_{ik} \cdot x_k}{a_{ij}}$ dans les $m-1$ autres inéquations, pour aboutir au système :

$$\underline{\underline{\tilde{A}}} \cdot \underline{\tilde{x}} \leq \underline{\tilde{b}} \tag{3.4}$$

qui définit la face $\tilde{f_{ij}}$, projection orthogonale de la face f_i sur un hyperplan de normale $\vec{e_j}$.

$\underline{\underline{\tilde{A}}}$ est une matrice $\tilde{m} \times \tilde{n}$, $\underline{\tilde{b}}$ une colonne de $\tilde{m}$ termes, où $\tilde{m} = m-1$ et $\tilde{n} = n-1$. $\underline{\tilde{x}}$ est la colonne *($\tilde{n}$ termes)* des composantes de x à laquelle a été retiré x_j *(la variable substituée)*.

Il est alors possible d'écrire :

$$V_i(n-1, \underline{\underline{A}}, \underline{b}) = \frac{1}{\lambda} \cdot V_i(\tilde{n}, \underline{\underline{\tilde{A}}}, \underline{\tilde{b}}) \tag{3.5}$$

où λ est le rapport entre la mesure de volume de la face f_i et sa projection orthogonale $\tilde{f_{ij}}$:

$$\vec{n}(f_i) \cdot \vec{e_j} = \frac{a_{ij}}{\|\underline{a_i}\|}$$

$\vec{n}(f_i)$ étant la normale unitaire à la face f_i.
Pour illustrer cela, voir la figure 3.10, qui représente un cas 3D.

En remplaçant 3.5 dans 3.2 on obtient l'algorithme récursif de Lasserre :

$$V(n, \underline{\underline{A}}, \underline{b}) = \frac{1}{n} \sum_{i=1}^{m} \frac{b_i}{a_{ij}} V_i(\tilde{n}, \underline{\underline{\tilde{A}}}, \underline{\tilde{b}}) \tag{3.6}$$

Après $n-1$ substitutions, à la fin de chaque branche du schéma récursif, on aboutit à un domaine dans R, qui est représenté par un système de $m-n+1$ inéquations, et qui est de la forme $\underline{\alpha} \cdot x_l \leq \underline{\beta}$, avec $\underline{\alpha}$ et $\underline{\beta}$ des colonnes de taille $m-n+1$, et x_l la variable restante. La mesure de volume d'un tel domaine est égale à :

$$max\left\{0, \left(min\left\{\frac{\alpha_i^{positif}}{\beta_i}\right\} - max\left\{\frac{\alpha_i^{négatif}}{\beta_i}\right\}\right)\right\} \tag{3.7}$$

Remarque importante : Cette méthode ne peut être appliquée qu'à des polyèdres P^{int} convexes. Pour des domaines $\underline{non-convexe}$ certaines cellules peuvent ne pas être convexes *(voir 3.3.4)*, ce qui interdit l'usage de l'algorithme récursif de *Lasserre* dans ce contexte.

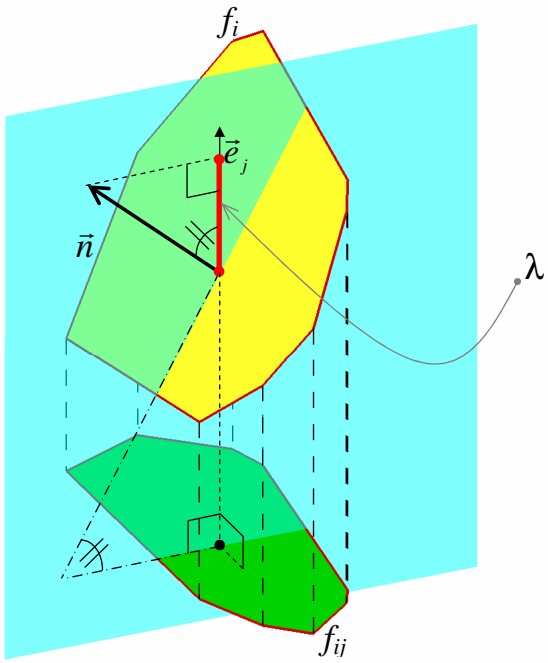

FIG. 3.10 – Projection orthogonale de la face f_i sur un plan de normale $\vec{e_j}$

b) Par le biais du volume complémentaire

La mesure de volume du polyèdre P^{int} pour un voisin donné v_i peut être aussi définie comme étant la mesure de la variation du volume de la cellule de ce nœud après insertion du point x *(voir figure 3.11)*.

Soit $N_V(x)$ l'ensemble des nœuds voisins à x, et soit $S(N_V(x))$ l'ensemble des sommets des cellules des nœuds de l'ensemble $N_V(x)$ *(après insertion du point x)*.

Grâce à $S(N_V(x))$ il est possible de reconstruire la topologie *(cellules, faces, arêtes et sommets)* des cellules de l'ensemble $N_V(x)$, et de x également, et ce en utilisant l'algorithme du paragraphe 3.2.2. Nous pourrons ainsi par simple soustraction en déduire la variation des mesures de volume des cellules associées aux nœuds voisins du point x.

Cette méthode ne peut être appliquée que si la cellule de chaque nœud voisin est non infinie. Si ce n'est pas le cas, il est impossible de calculer la mesure de son volume, mais vu que nous n'avons besoin que de la mesure de la variation de volumes des cellules après insertion du point x, et non de leur volume, cette impossibilité peut être contournée en tronquant les cellules infinies *(ajout de faces)*.

La reconstruction de la topologie des cellules de l'ensemble $N_V(x)$ et de celle de x est nécessaire pour calculer leur Volume par 3.3. Afin de réduire les temps de calcul, ces volumes peuvent être calculés d'une autre façon, grâce à une décomposition en volumes élémentaires associés à chacune des arêtes de chaque cellule *(voir figure 3.12)*.

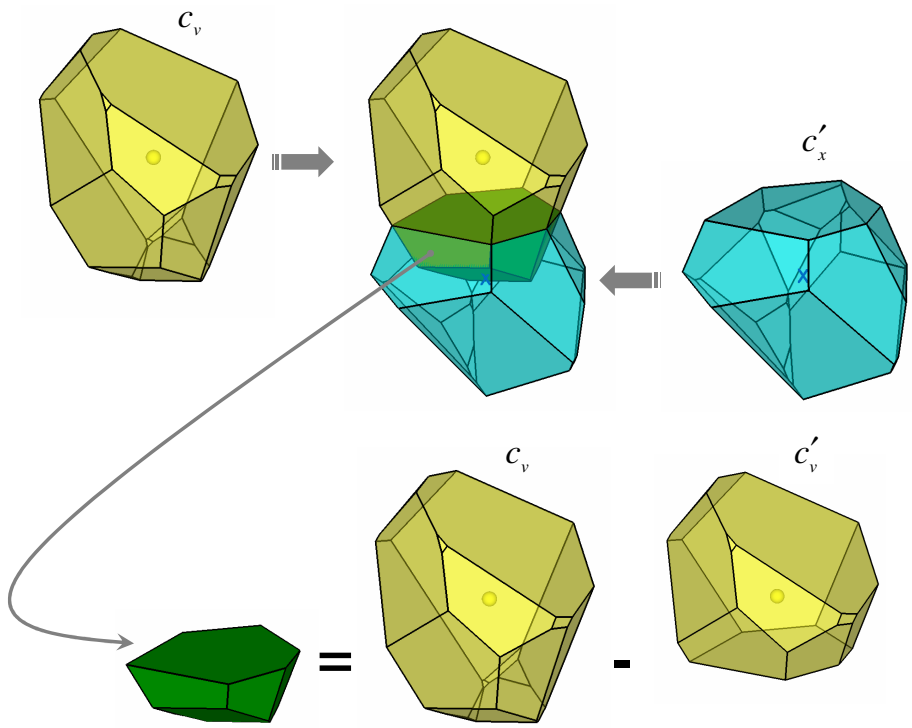

FIG. 3.11 – Etape-2 : Calcul de la mesure de volume du polyèdre P^{int} par le biais du volume complémentaire

Ces volumes élémentaires sont construits comme suit :

Soit a l'arête considérée, s_1, s_2 ses deux sommets, n_1, n_2, n_3 ses trois nœuds générateurs. La contribution de l'arrête a au volume de la cellule ayant comme nœud n_1 sera formée par les deux tétraèdres T_1, T_2, avec :
- T_1 tétraèdre ayant pour sommets : n_1, S_1, S_2, n_{12} *(milieux du segment* (n_1, n_2)*)*.
- T_2 tétraèdre ayant pour sommets : n_1, S_2, S_1, n_{13}*(milieux du segment* (n_1, n_3)*)*.
Il suffira donc de parcourir toutes les arêtes formées par les sommets de l'ensemble $S(N_V(x))$, et de rajouter la mesure de volume élémentaire associé à chacune des cellules des nœuds *(inclut dans l'ensemble* $N_V(x)$*)* de l'arête considérée.

c) Par approche topologique basée sur le DVC

Le polyèdre P^{int} généré par l'intersection de $\acute{c}_x$ cellule du point x, et c_{v_i} cellule d'un nœud voisin v_i avant insertion du nœud x, n'est pas quelconque. Il possède une propriété particulière grâce à laquelle il est possible de reconstruire sa topologie *(faces, arêtes, sommets)* en se basant uniquement sur les connectivités du DVC. Aucune opération d'intersection géométrique proprement dite n'est nécessaire.

En effet, dans le cas général si l'on prend deux volumes A et B qui s'interpénètrent, on peut décomposer la surface du volume intersection en deux parties bien distinctes,

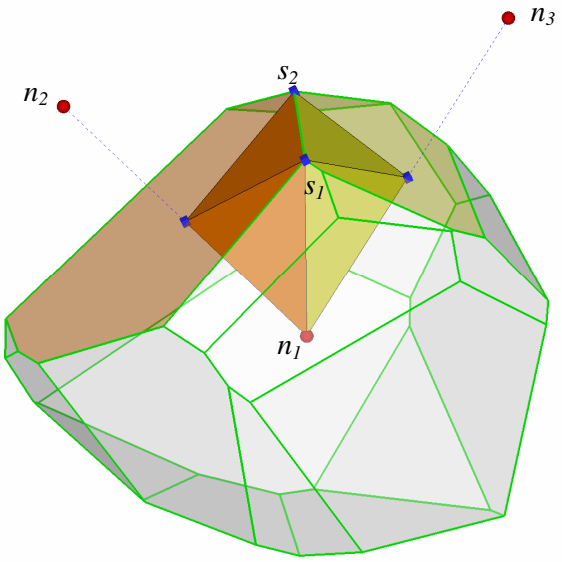

FIG. 3.12 – Contribution du volume associé à une arrête au volume d'une cellule

ΓA_B, partie de la surface ΓA du volume A qui est à l'intérieur du volume B, et ΓB_A, partie de la surface ΓB du volume B qui est à l'intérieur du volume A. ΓA_B et ΓB_A ont en commun une courbe S, intersection de ΓA et de ΓB *(voir figure 3.13)*.

Dans notre cas, si l'on considère la cellule $\acute{c}_x$ comme étant le volume A, et c_{v_i} comme étant le volume B, ΓA_B se réduit exactement à $f_i\acute{c}_x$ face de la cellule $\acute{c}_x$ ayant comme second nœud le nœud v_i.

Quant à ΓB_A, elle se décompose en deux parties, F_{trq} ensemble des faces de la cellule c_{v_i} tronquées par les arêtes de $f_i\acute{c}_x$, et éventuellement $F_{\overline{trq}}$ ensemble des faces de la cellule c_{v_i} étant complètement à l'intérieur de $\acute{c}_x$ *(voir figure 3.14)*.

L'algorithme se composera donc de deux étapes, la première qui construira l'ensemble F_{trq}, et la seconde qui construira l'ensemble $F_{\overline{trq}}$, et ce pour chaque nœud v_i voisin du nœud x. Le volume construit, sa mesure est calculée avec 3.3. Pour ce qui suit se référer à la figure 3.15.

Soit v_i le voisin considéré, et c_{v_i} sa cellule associée. Et soit $f_i\acute{c}_x$ la face de la cellule du point x ayant v_i comme un de ces deux nœuds.

Chaque arête $a_jf_i\acute{c}_x$ de $f_i\acute{c}_x$ tronque une face $f_jc_{v_i}$ de c_{v_i}, face ayant comme l'un de ses deux nœuds le nœud v_{ij}, nœud de l'arête $a_jf_i\acute{c}_x$ différent de x et de v_i. Les deux sommets de l'arête $a_jf_i\acute{c}_x$ tronquent deux arêtes de $f_jc_{v_i}$, $a_kf_jc_{v_i}$ et $a_lf_jc_{v_i}$, arêtes ayant pour nœuds v_{ij}^- v_{ij}^+ respectivement, nœuds d'arête différents de x et de v_i, des arêtes $a_j^-f_i\acute{c}_x$ et $a_j^+f_i\acute{c}_x$, arêtes de $f_i\acute{c}_x$ qui précède et qui succède $a_jf_i\acute{c}_x$. Soit A le sous ensemble d'arêtes de $f_jc_{v_i}$ qui succède l'arête $a_kf_jc_{v_i}$ jusqu'à $a_lf_jc_{v_i}$.

La liste d'arêtes de la face $f_j c_{v_i}$ tronquée sera composé de $a_k f_j c_{v_i}$ tronquée, suivi du sous ensemble d'arêtes A, de l'arête $a_l f_j c_{v_i}$ tronquée, et enfin de l'arête $(-)a_j f_i \acute{c}_x$.

Avec chaque arête a de A, en construira des couples *(arête, nœud)*, où "arête" est a, et "nœud" v_i, que l'on ajoutera à une liste de couples L.

Une fois l'ensemble F_{trq} construit, on retrouvera les faces de l'ensemble $F_{\overline{trq}}$ par adjacence aux faces de F_{trq} grâce aux arêtes de la liste de couples L, et ce comme suit :

- tant que la liste L est non vide, on retire le couple c en queue. Soit f la face de c_{v_i} ayant pour nœud v, nœud de l'arête du couple c différent de v_i et du nœud du couple c. f fait partie de l'ensemble $F_{\overline{trq}}$, elle est donc ajoutée à ce dernier. Pour chaque arête a de f différente de l'arête du couple c, on vérifie si elle est pressente dans un couple de L, si oui, ce couple est supprimé, si non, on rajoute à L le couple *(a,v)*.

L'approche topologique présente les avantages suivants par rapport aux deux approches précédentes :

- gain en temps de calcul *(d'un facteur de l'ordre de 10 par rapport à Lasserre, comme il est montré plus loin)*,
- indifférente au fait que les cellules voisines soient infinies ou non *(on ne calcule pas la variation de la mesure de volume d'une cellule voisine mais la mesure de volume du polyèdre intersection P^{int})*,
- indifférente au fait que les cellules voisines, et notamment celle du point x, soient convexes ou non.

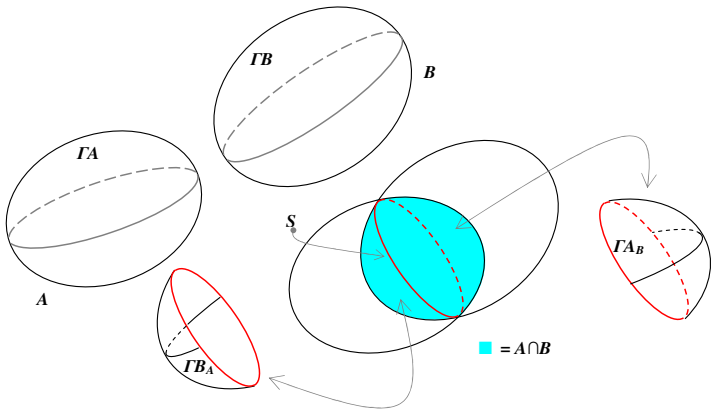

FIG. 3.13 – Volumes A et B qui s'interpénètrent

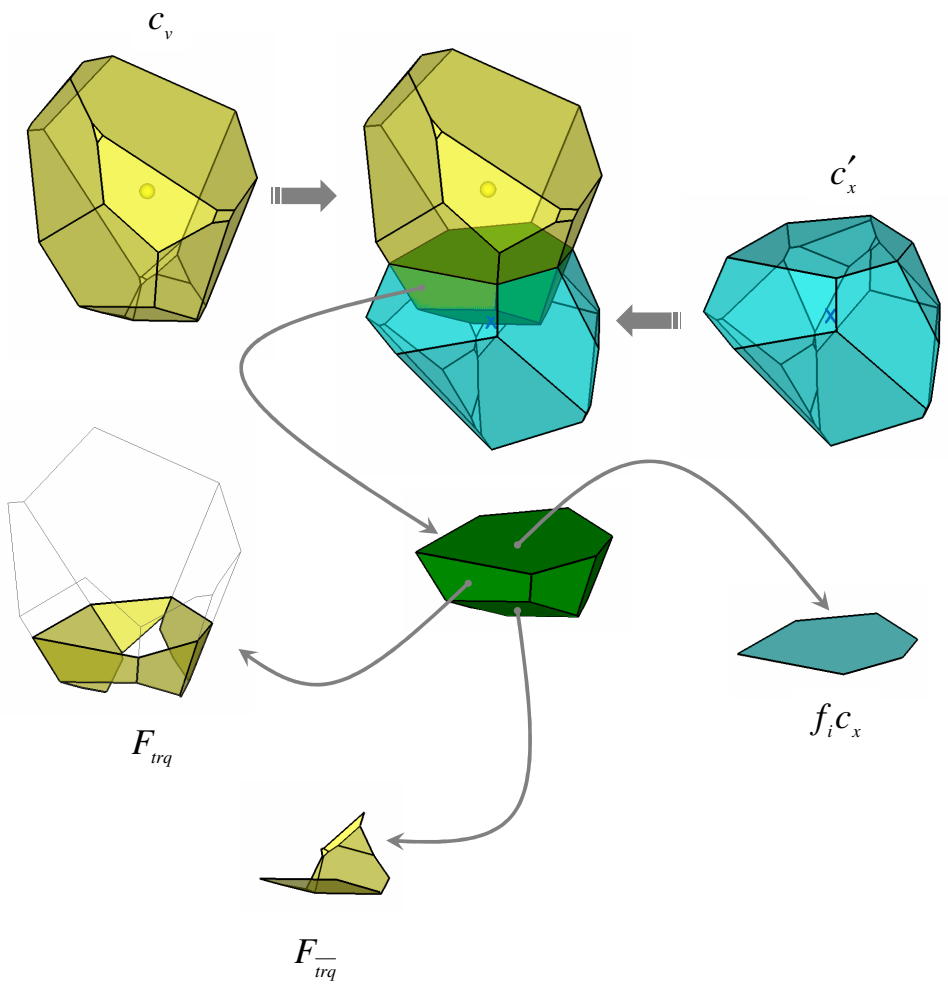

FIG. 3.14 – Etape-2 : Calcul de la mesure de volume du polyèdre P^{int} par approche topologique basée sur le DVC

Algorithme :

Données d'entrée :
- Le diagramme de Voronoï contraint *(sans le point x, point où l'on veut évaluer l'interpolation).*
- La cellule de Voronoï associée au point x.

Données de sortie :
- L'interpolation **CNEM** de type *Sibson* au point x.

L=nombre de faces de $c\acute{}_x$
Pour i **de** 0 **à** $L-1$ **faire**
  v_i : nœud tampon ; $v_i \leftarrow$ nœud de la face $f_i c\acute{}_x \neq$ de x
  P : polyèdre tampon ; rajouter la face $f_i c\acute{}_x$ à P
  $L^{(a-n)}$: liste de couples $(arête, nœud)$ tampon
  N=nombre d'arêtes de $f_i c\acute{}_x$
  Pour j **de** 0 **à** $N-1$ **faire**
    v_{ij}^-, v_{ij}, v_{ij}^+ : nœud tampon
    $v_{ij}^- \leftarrow$ nœud de l'arête $a_{(j-1)mod(N)} f_i c\acute{}_x \neq$ de x et de v_i
    $v_{ij} \leftarrow$ nœud de l'arête $a_j f_i c\acute{}_x \neq$ de x et de v_i
    $v_{ij}^+ \leftarrow$ nœud de l'arête $a_{(j+1)mod(N)} f_i c\acute{}_x \neq$ de x et de v_i
    f^{trq} : face tampon ; rajouter l'arête $(-)a_j f_i c\acute{}_x$ à f^{trq}
    f : face tampon ; $f \leftarrow$ face de c_{v_i} dont un des deux nœuds est v_{ij}
    M=nombre d'arêtes de f
    Pour k **de** 0 **à** $M-1$ **faire**
      n^k : nœud tampon ; $n^k \leftarrow$ nœud de l'arête $a_k f \neq$ de v_i et de v_{ij}
      Si $(n^k = v_{ij}^-)$ **Alors**
        a_{trq}^{deb} : arête tampon ; $a_{trq}^{deb} \leftarrow a_k f$; $s_0 a_{trq}^{deb} \leftarrow s_0 a_j f_i c\acute{}_x$
        Rajouter l'arête a_{trq}^{deb} à f^{trq}
        $k \leftarrow (k+1)mod(M)$; $n^k \leftarrow$ nœud de l'arête $a_k f \neq$ de v_i et de v_{ij}
        Tant que $(n^k \neq v_{ij}^+)$ **faire**
          Rajouter $a_k f$ à f^{trq}
          Rajouter le couple $(a_k f, n^k)$ à $L^{(a-n)}$
          $k \leftarrow (k+1)mod(M)$
          $n^k \leftarrow$ nœud de l'arête $a_k f \neq$ de v_i et de v_{ij}
        Fait
        a_{trq}^{fin} : arête tampon ; $a_{trq}^{fin} \leftarrow a_k f$; $s_1 a_{trq}^{fin} \leftarrow s_1 a_j f_i c\acute{}_x$
        Rajouter l'arête a_{trq}^{fin} à f^{trq}
        Sortir de la boucle k
      Fin Si
    Fin Pour
    Rajouter f^{trq} à P
  Fin Pour
  ...suite de l'algorithme page suivante...
Fin Pour

Algorithme 2: Calcul des fonctions de forme de type Sibson

Pour i **de** 0 **à** $L-1$ **faire**
  ...
  ..
  .
  Tant que ($L^{(a-n)}$ non vide) **faire**
    Retirer le couple (a^{queue}, n^{queue}) en queue de $L^{(a-n)}$
    $f^{\overline{trq}}$: face tampon ; $f^{\overline{trq}} \leftarrow$ face de c_{v_i} dont un des deux nœuds est n^{queue}
    Rajouter $f^{\overline{trq}}$ à P
    Pour chaque arête a de $f^{\overline{trq}}$ **faire**
      Si $(a \neq a^{queue})$ **Alors**
        Si (le couple $((-)a, n^{queue})$ existe dans $L^{(a-n)}$) **Alors**
          Supprimer ce couple de $L^{(a-n)}$
        Sinon
          n : nœud tampon ; $n \leftarrow$ nœud de l'arête $a_j f^{\overline{trq}} \neq$ de v_i et de n^{queue}
          Rajouter le couple (a, n) à $L^{(a-n)}$
        Fin Si
      Fin Si
    Fin Pour
  Fait
  $\Phi(v_i, x) = \frac{\text{volume de } P}{\text{volume de } c_x^{'}}$
Fin Pour

Algorithme 3: Calcul des fonctions de forme de type Sibson suite et fin

d) Par approche topologique basée sur la TDC

Boissonnat et al [BC02] proposent un algorithme basé entièrement sur la TD *(ou TDC)*. La même décomposition que précédemment du polyèdre P^{int} est faite. Soit F_{trq} l'ensemble de toutes les faces tronquées de tous les polyèdres P^{int} associées à tous les voisins de x, et soit $F_{\overline{trq}}$ l'ensemble de toutes les faces non tronquées de tous les polyèdres P^{int} associées à tous les voisins de x. La construction de ces deux ensembles se fait comme suit :

- Soit C la cavité engendrée par l'insertion du point x dans la TDC *(ensemble de tétraèdres dont la sphère circonscrite contient le point x)*. Soint A l'ensemble des arêtes de tétraèdre de C. A se décompose en deux sous ensembles, A_{int} sous ensemble des arêtes internes à C, et A_{ext} sous ensemble des arêtes externes à C *(arête se situant sur une face (de tétraèdre) externe à C)*.

  Chaque arête externe a_{ext} *(arête a-b figure 3.16 à gauche)* de A_{ext} a pour dual une face f_{trq} *(face rouge)* de F_{trq}. Les sommets associés aux tétraèdres de C ayant a_{ext}

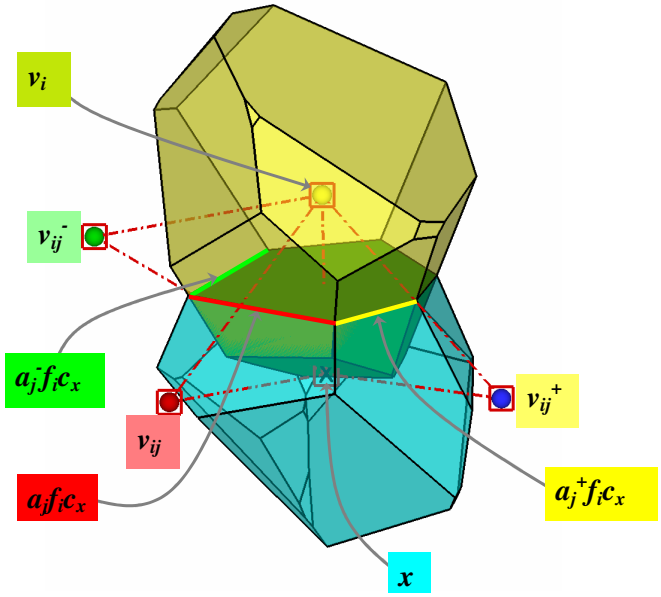

FIG. 3.15 – Identification des entités utilisées dans l'approche topologique basée sur le DVC

pour arête constituent une partie de la liste des sommets de f_{trq}. Pour compléter cette liste, il faut rajouter à ses extrémités les deux nouveaux sommets associés aux tétraèdres ayant pour nœuds le point x et les nœuds des deux faces externes de la cavité qui ont pour arête a_{ext} *(faces a-b-c et b-a-e)*.

Chaque arête a_{int} *(arête a-b figure 3.16 à droite)* de A_{int} a pour dual une face $f_{\overline{trq}}$ *(face rouge)* de $F_{\overline{trq}}$, cette face a pour sommets les sommets associés aux tétraèdres de C ayant a_{int} pour arête.

Une fois ces deux ensembles construits, il sufit d'ajouter chaque face de ces deux ensembles au polyèdre P^{int} associé à chacun des deux nœuds de l'arête de tétraèdre duale à cette face. En rajoutera enfin à chaque polyèdre P^{int} associé à un nœud voisin de x donné, la face de la cellule de x ayant comme l'un de ses deux nœuds de face ce voisin.

En plus de procéder les mêmes avantages que l'algorithme topologique basé sur le DVC, et de diviser les temps de calcul de ce dernier par un facteur 2, cet algorithme est entièrement basé sur la TDC, la construction du DVC et donc inutile.

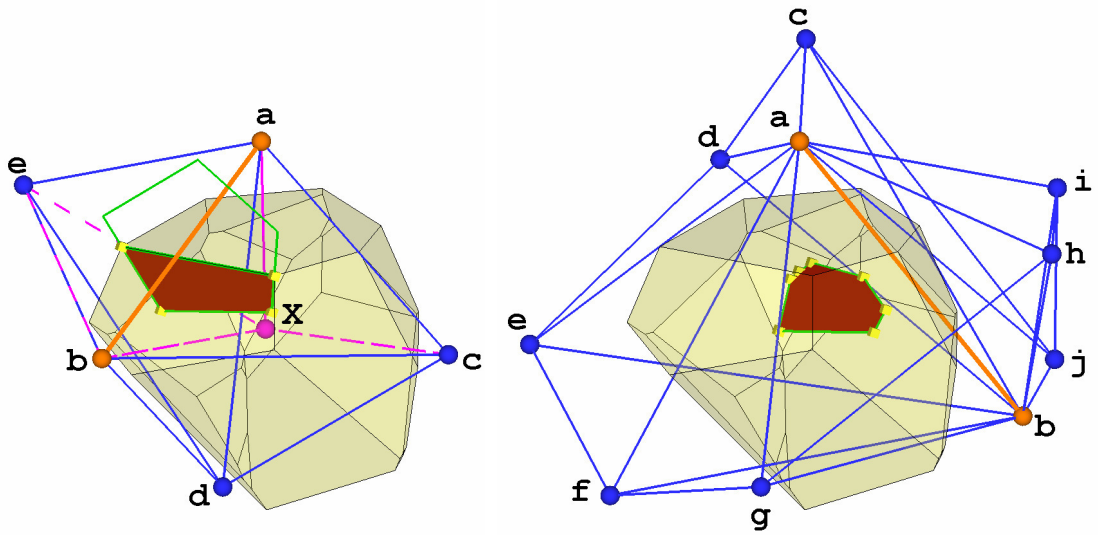

FIG. 3.16 – Etape-2 : Calcul de la mesure de volume des polyèdres P^{int} par approche topologique basée sur la TDC

e) Par l'algorithme de Watson

Préliminaire : Calcul de la mesure de Volume d'un polyèdre par décomposition de Lawrence [Law91]

Soit un polyèdre P, et soit π un hyperplan qui n'est parallèle à aucune face de P. La mesure de volume du polyèdre P peut être alors calculée en sommant *algébriquement* les mesures de volume de simplexes spx_s^{Law}, que l'on construit en prennant pour chaque sommet s de P, en plus de l'hyperplan π, les hyperplans associés aux faces de P ayant s pour sommet. Chacun de ces hyperplans divise l'espace en deux parties signées, la partie positive étant celle incluant P. Le signe de la mesure de volume d'un simplexe spx_s^{Law} est le signe de la partie de l'espace dans laquelle il se situe, qui est le produit des signes de cette partie l'espace par rapport à chacun des hyperplans associés à spx_s^{Law} *(l'hyperplan π n'étant pas pris en compte)*.

Sur la figure 3.17, est illustré un exemple de cela en $2d$. Le polyèdre P ayant pour sommets a, b, c, d, est décomposé en triangles, et ce en prenant pour chaque sommet s de P, les droites associées aux arêtes de P ayant s pour sommets, et la droite π, ce qui nous donne :

- Pour $a \rightarrow$ droites (da), (ab), $\pi \rightarrow$ triangle (p_{da}, a, p_{ab}), signe $+ = -_{(da)} \times -_{(ab)}$
- Pour $b \rightarrow$ droites (ab), (bc), $\pi \rightarrow$ triangle (p_{ab}, b, p_{bc}), signe $- = -_{(ab)} \times +_{(bc)}$
- Pour $c \rightarrow$ droites (bc), (cd), $\pi \rightarrow$ triangle (p_{bc}, c, p_{cd}), signe $+ = +_{(bc)} \times +_{(cd)}$
- Pour $d \rightarrow$ droites (cd), (da), $\pi \rightarrow$ triangle (p_{cd}, d, p_{da}), signe $- = +_{(cd)} \times -_{(da)}$

Watson [DFW] propose d'utiliser la décomposition de polyèdre de Lawrence pour le calcul de la mesure de volume des polyèdres intersections P^{int}, qu'il a modifié en prenant comme plan π un plan passant par une des faces de P^{int}. La mesure de Volume

du polyèdre P^{int} s'obtient alors en sommant *algébriquement* les mesures de volume des simplexes spx_s^{Law} associées aux sommets ne se situant pas sur π, $S_{\notin}\pi$. La face de P^{int} par laquelle passe le plan π n'est pas choisi au hasard, c'est la face de P^{int} ayant pour nœud *(au sens Voronoï)* le point x, point d'évaluation de l'interpolation, et le nœud v voisin de x rattaché au polyèdre intersection P^{int} considéré, autrement dit c'est le plan médiant entre le point x et le nœud voisin v.

Les hyperplans associés aux faces de P^{int} ayant pour sommet $s_{\notin}\pi$ de $S_{\notin}\pi$ sont des plans médiants associés aux deux nœuds de chaque arête du simplexe de Delaunay dual à $s_{\notin}\pi$ ayants v pour nœud.

Les sommets du simplexe $spx_{s_{\notin}\pi}^{Law}$ sont donc :

- le sommet de Voronoï $s_{\notin}\pi$
- les centres des hypersphères circonscrites aux simplexes ayant pour sommets le point x et les nœuds de chacune des faces du simplexe de Delaunay dual au sommet de Voronoï $s_{\notin}\pi$, ayant le nœud v comme nœud.

Afin d'illustrer cela, prenons l'exemple $2d$ donné figure 3.18. Le polyèdre *(a,b,c,d,e,f)* est le polyèdre intersection P^{int} associé à un nœud voisin v du point x. Chaque sommet du polyèdre P^{int} ne se situant pas sur la droite π *(sommets c,d,e,f)* est un sommet de Voronoï de la cavité engendrée par l'insertion du point x, ayant v pour nœud *(au sens Voronoï)*, autrement dit c'est le centre d'un cercle circonscrit à un triangle de Delaunay de la cavité ayant v pour nœud. La contribution à la mesure de volume de P^{int} associée à un sommets $s_{\notin}\pi$, est le triangle construit avec les deux droites associées aux arêtes de P^{int} ayant $s_{\notin}\pi$ pour sommet, et la droite π. Vu que les deux droites passant par $s_{\notin}\pi$ sont des médianes d'arêtes du triangle de Delaunay dual à $s_{\notin}\pi$ ayants v pour nœud, et vu que la droite π est aussi une médiane, droite médiane entre le point x et au nœud v, les points d'intersections des deux droites passant par $s_{\notin}\pi$ avec la droite π sont par conséquence les centres des cercles circonscrits aux triangles ayant pour nœud le point x, et les deux nœuds de chacune des deux arêtes du triangle de Delaunay dual à $s_{\notin}\pi$ ayant v pour nœud. Exemple, pour le sommet c :

- L'intersection de la première droite passant par c, droite (bc) et π, point p_{bc}, est le centre du cercle circonscrit au triangle ayant pour nœud x et v, n_1 qui sont les nœuds d'une arête du triangle de Delaunay dual au sommet de Voronoï c ayant v pour nœud.
- L'intersection de la deuxième droite qui passant par ce sommet, droite (cd) et π, point p_{bc}, est le centre du cercle circonscrit au triangle ayant pour nœud x, et v, n_2 qui sont aussi les nœuds d'une arête du triangle de Delaunay dual au sommet de Voronoï c ayant v pour nœud.

Cet algorithme est actuellement le plus efficace que nous ayons trouvé pour le calcul des fonctions de forme Sibson. Sa simplicité et sa rapidité sont dues au fait qu'il ne construit pas les polyèdres intersection P^{int} *(faces, arêtes, sommets)* pour en suite calculer la mesure de leurs volumes. Il le fait directement en bouclant sur les simplexes de la cavité, chacun de ces derniers apporte une contribution à la mesure de volume de chacun des polyèdre P^{int} rattaché aux nœuds de ce simplexe. Il faut cependant signaler que l'algorithme de Watson n'est pas applicable pour des points d'évaluation de l'interpolation x se situant sur une face de tétraèdre de Delaunay. En effet, dans ce

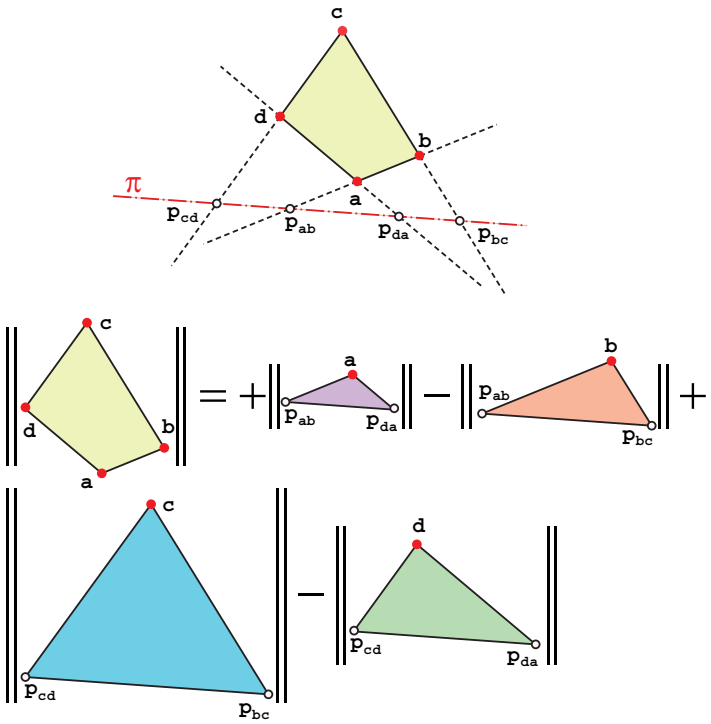

FIG. 3.17 – Décomposition de Lawrence en $2d$

cas, cette face sera forcement une face d'un tétraèdre de la cavité. Un des simplexes $spx^{Law}_{s\notin\pi}$ aura donc comme sommet l'orthocentre d'un simplexe ayant pour sommet les nœuds de cette face et le point x, qui est indéterminé, car les sommets de ce dernier sont coplanaires.

3.3.3 Test comparatif des différents algorithmes

Les cinq méthodes de calcul de fonction de forme de type Sibson ont été implémentées (langages c, c++). Pour la méthode de calcul se basant sur l'algorithme récursif de Lasserre, nous utilisons le code Vinci [EBF03], dans lequel est implémenté une version améliorée de ce dernier, l'algorithme r-Las [BEF00].

Calculateur utilisé

- OS : microsoft windows xp
- Compilateur : Microsoft (R) Incremental Linker Version 7.10.3077
- Processeur : Mobile Intel Pentium M 745 - 1800 MHz
- RAM : 1024 Mo - PC2700 DDR SDRAM

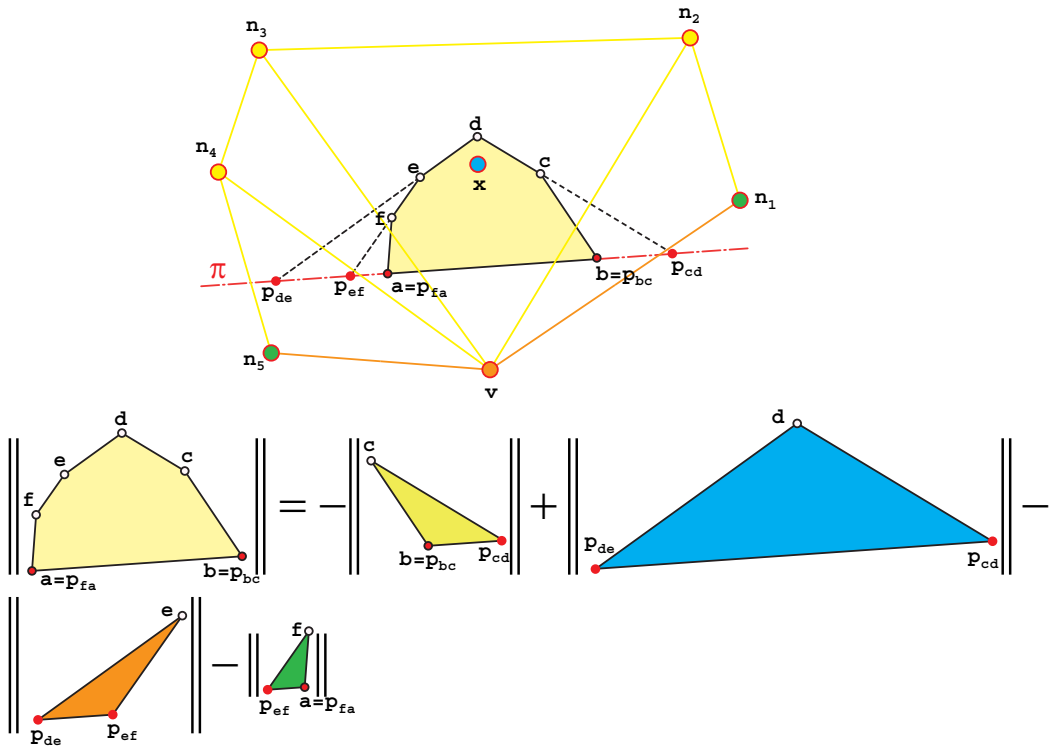

FIG. 3.18 – Algorithme de Watson en $2d$

Le test

On prend un cube de dimension unitaire (1x1x1) avec 10000 nœuds répartis de façon aléatoire à l'intérieur. Les fonctions de formes sont évaluées aux barycentres de chaque tétraèdre de Delaunay. Ceci évite de comptabiliser le temps pris par la localisation du point où l'on veut évaluer la fonction de forme dans la tétraèdrisation. Seuls les points pour lesquels les 5 méthodes sont applicables ont servi pour les tests,ont été écartés les points où au moins l'une des cellules des nœuds voisins au point d'évaluation est infinie *(méthode de calcul par volume complémentaire non applicable)*.

Résultats

Les temps donnés sont des temps moyens correspondants à 1000 évaluations. Le temps de calcul global est divisé en deux :

- $T_{cavité}$: temps pris pour la construction de la cavité, il est évidemment constant pour les cinq algorithmes testés.
- T_{cal-ff} : temps pris par le calcul des fonctions de formes une fois la cavité construite.

Algorithme	Lasserre	Vol-comp	Topo-DVC	Topo-TDC	Watson
$T_{cavité}\ (m.s)$	≈ 35				
$T_{cal-ff}(m.s)$	≈ 4136	≈ 544	≈ 413	≈ 192	≈ 51
$\frac{T_{cal-ff}\ Alg}{T_{cal-ff}\ Alg-Wat}$	≈ 82	≈ 11	≈ 8	≈ 4	1

Comme le montrent ces mesures, l'algorithme de Watson est le plus performant.

Afin de situer ces temps de calculs par rapport aux temps que pourrait prendre d'autre fonctions de formes, le même calcul à été mené en prenant cette fois ci des fonctions de formes éléments naturels Laplace et éléments finis linaire :

f.f	Sibson*(Watson)*	Laplace	EF Linéaire
$T(m.s)$	≈ 87	≈ 150	≈ 3
$\frac{T}{T(EF\ Linéaire)}$	≈ 33	≈ 56	1

Conculsion

En conclusion, l'algorithme de Watson semble être le plus approprié pour le calcul des fonctions des formes de type Sibson dans le cas de la CNEM. Comme nous l'avons vu cet algorithme ne fonctionne pas pour des points d'évaluation se situant sur et proche d'une face de Delaunay. Pour ces points l'algorithme topologique basé sur la TDC est préconisé. Dans la pratique le taux d'échec de l'algorithme de Watson pour des points d'évaluations positionnés de façon aléatoire, est de l'ordre de $\frac{1}{1000}$.

3.3.4 Mise en évidence de phénomènes associés aux fonctions de formes CNEM apparaissant en 3d pour des domaines non convexes

Pour des domaines non convexes, il existe des lieux géométriques pour lesquels l'insertion d'un point dans le diagramme de Voronoï contraint, conduit à la création d'une cellule de Voronoï associée non convexe. Afin d'illustrer cela, prenons le domaine Ω de la figure 3.19. L'insertion du point x de coordonnées $[-0.1, 0, 0]$ dans le diagramme de Voronoï contraint de Ω, conduit à la création d'une cellule non convexe, comme l'illustre la figure 3.20. Sur cette même figure, on peut voir que le polyèdre P^{int} associée au nœud voisin v *(n_0, ou n_1)* l'est aussi.

Le passage d'une topologie convexe à une topologie non convexe de la cellule associée au point x, se fait de façon continue quand on déplace ce dernier. La figure 3.21, met en évidence cela en faisant passer progressivement le point x de la position $[-0.25,\ 0,\ 0]$ à la position $[0^-,\ 0,\ 0]$ *(déplacement suivant l'axe* $\boldsymbol{X}$*)*.

Ayant remarqué cette propriété de non convexité, il est alors intéressant d'étudier l'évolution des fonctions de forme en fonction de la position du point x. Le graphe 3.22 commence par donner, en fonction de la coordonnée suivant l'axe X du point x, les mesures de volumes :

- de la cellule associée au point x *($Vol(c_x)$)*.
- des polyèdre intersections P^{int} associées à chaque voisin de x.

Le graphe 3.23 permet enfin de préciser, toujours en fonction de la coordonnée suivant l'axe X du point x, la valeur des fonctions de forme associées aux nœuds voisins de x.

Il est facile de vérifier numériquement que toutes les propriétés relatives aux fonctions de formes énoncées dans le chapitre 2.2.4 sont vérifiées. On peut cependant voir qu'il existe une position x_0 pour laquelle le volume de la cellule associée au point x s'annule. Ceci donne lieu à l'indétermination des fonctions de formes de type Sibson en ce point *(division par 0)*. De telles positions ne sont pas uniques. En effet, si l'on calcule la mesure de volume de la cellule associée au point x dans un voisinage de x_0, les point correspond aux lieux de l'espace où la mesure de volume de la cellule associée au point x est nulle forment une surface*(voir figure 3.24)*.

Il est à noter que même pour d'autres fonctions de forme éléments naturels comme Laplace, il existe aussi des lieux pour lesquels elles sont indéterminées, comme le montre le graphe de la figure 3.25.

Ce problème a été mis en évidence récemment et son influence sur le comportement de la **CNEM** n'a pas encore été étudié en détail. Pour les simulations menées à ce jour, nous n'avons pas observé de comportement suspect au voisinage des zones non convexes du domaine.

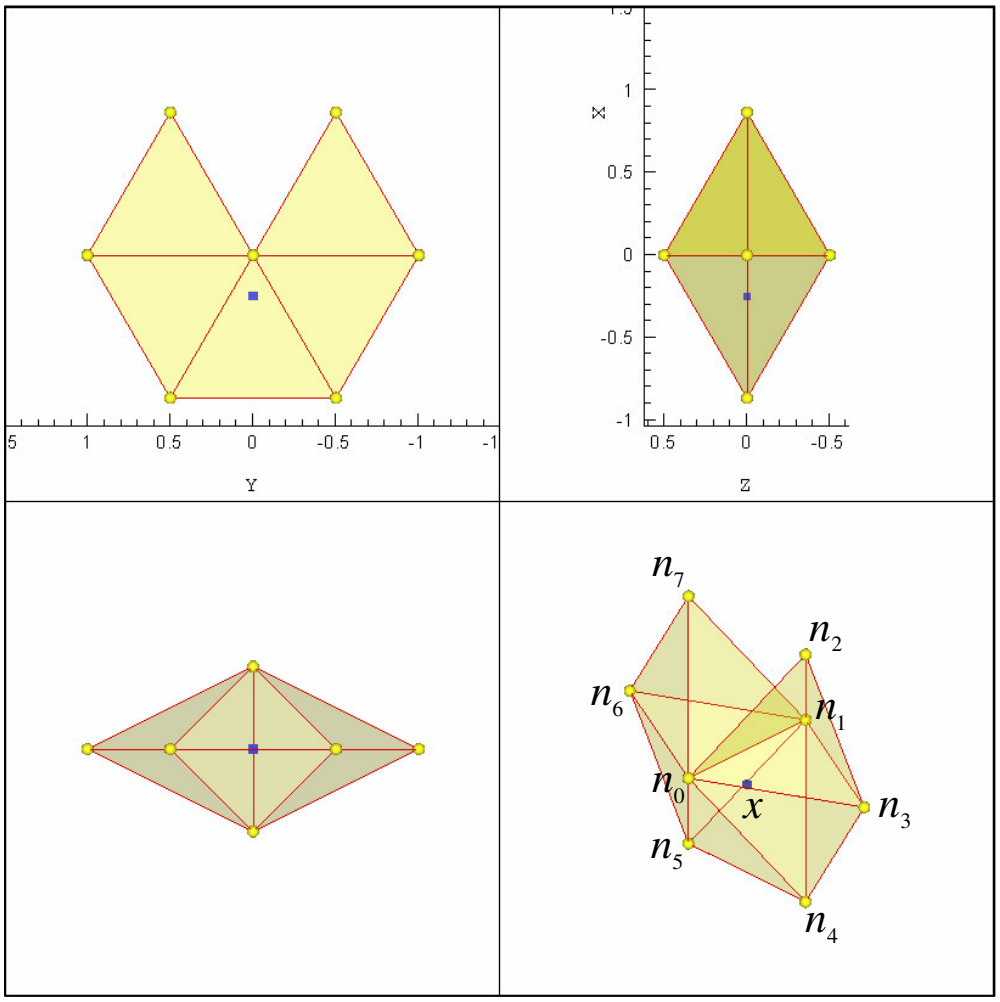

FIG. 3.19 – Domaine non convexe et positionnement du point x

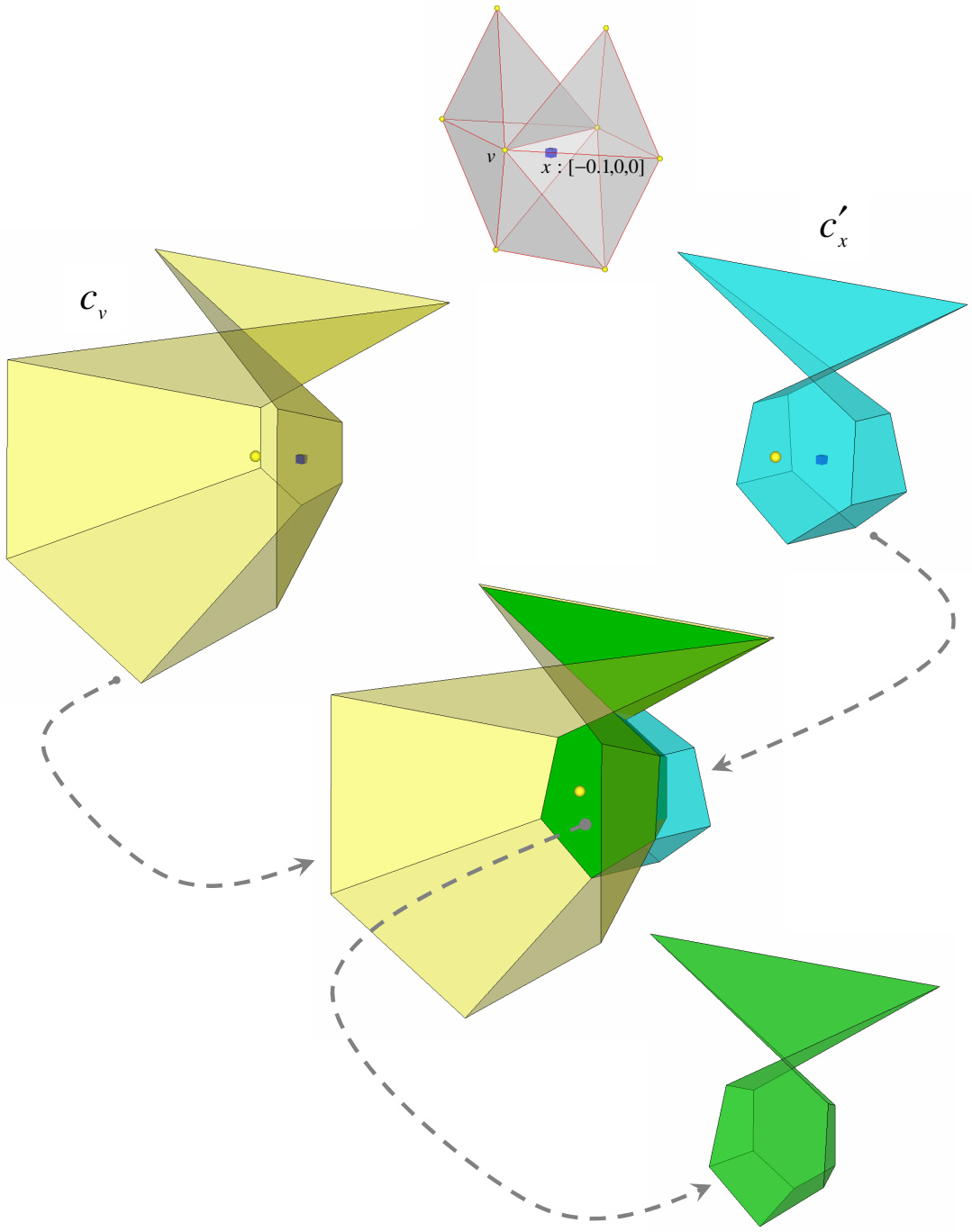

FIG. 3.20 – Domaine non convexe : cellule $c_x^{'}$ – Intersection avec la cellule du nœud voisin c_v

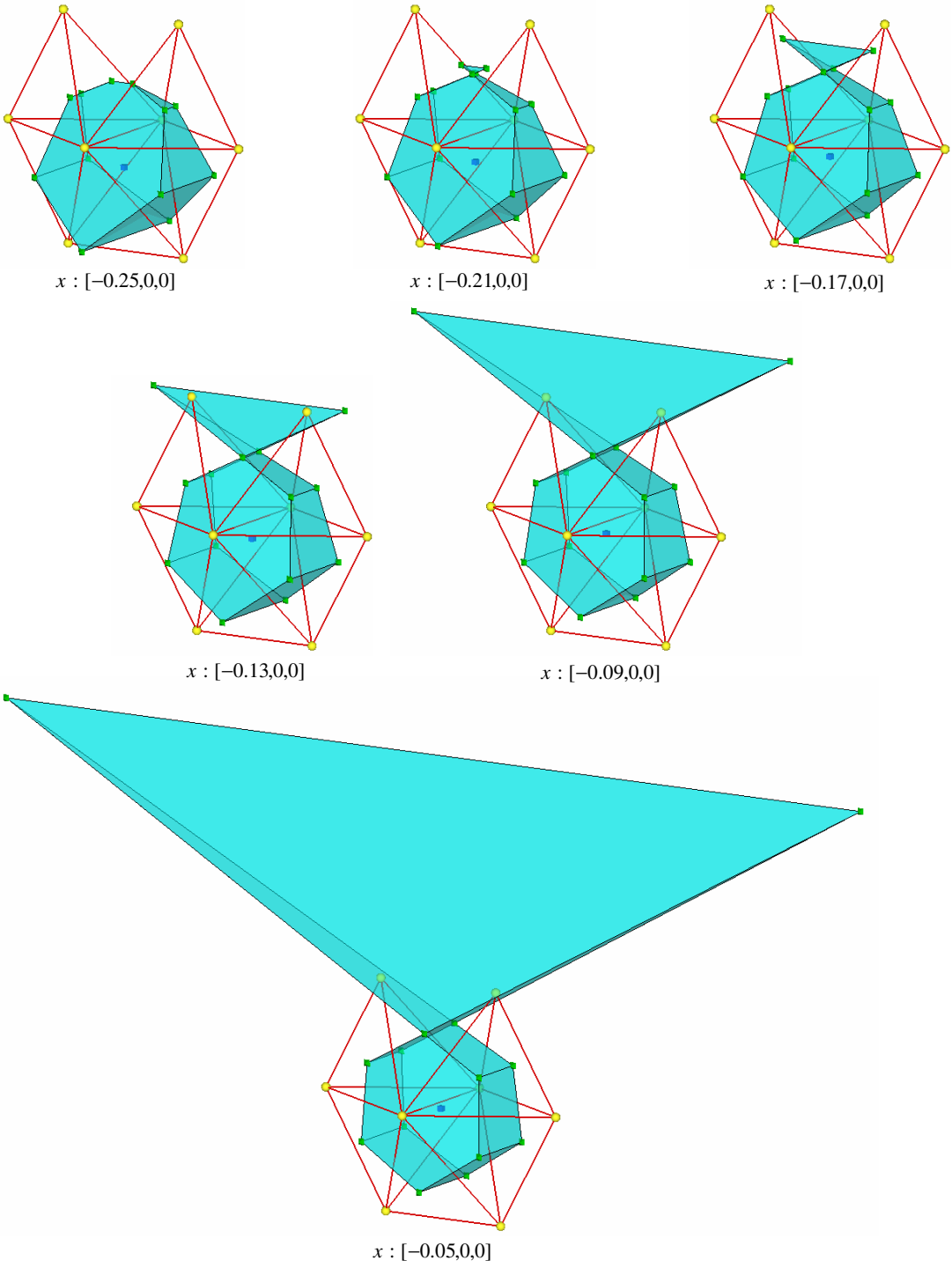

FIG. 3.21 – Passage d'une cellule convexe à une cellule non convexe en fonction de la position de x

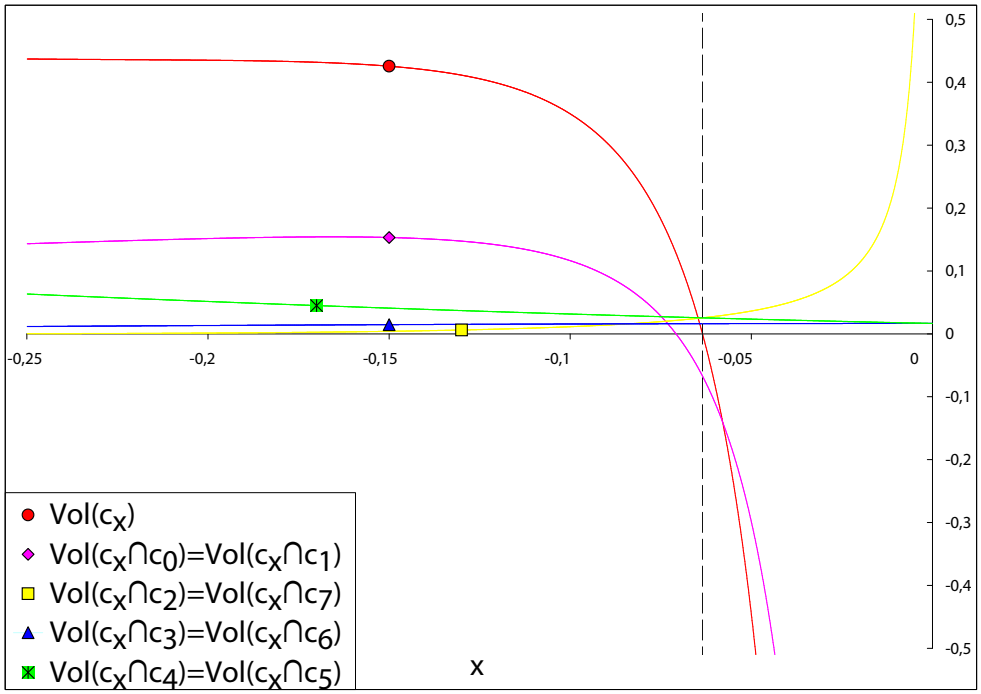

FIG. 3.22 – Mesure de volumes : cellule du nœud x, et intersections avec les cellules voisines

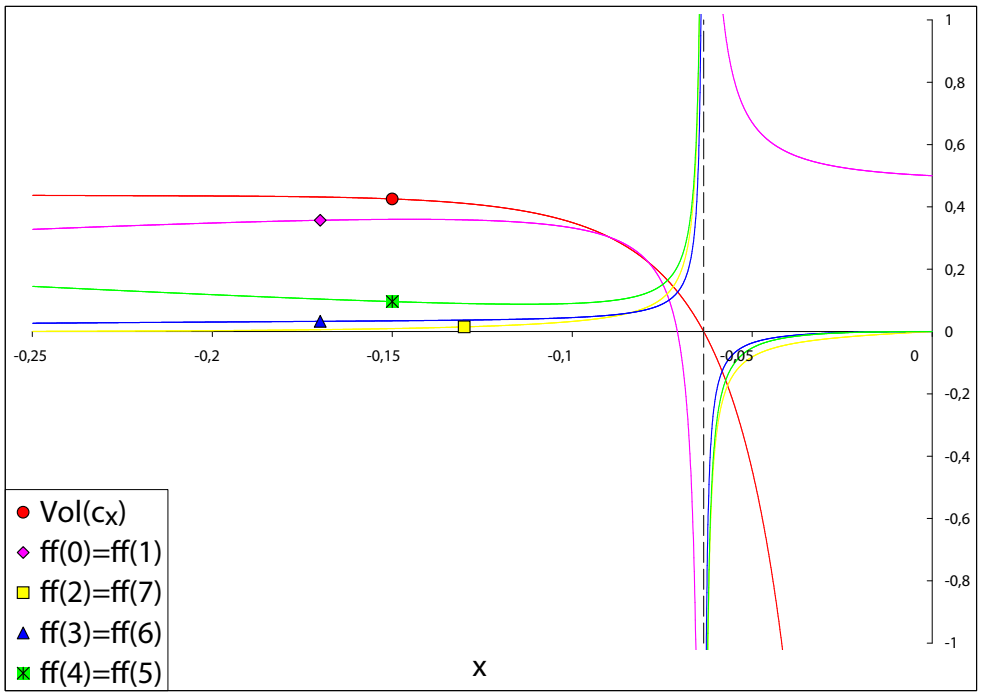

FIG. 3.23 – Fonction de formes Sibson en x pour les différents voisins

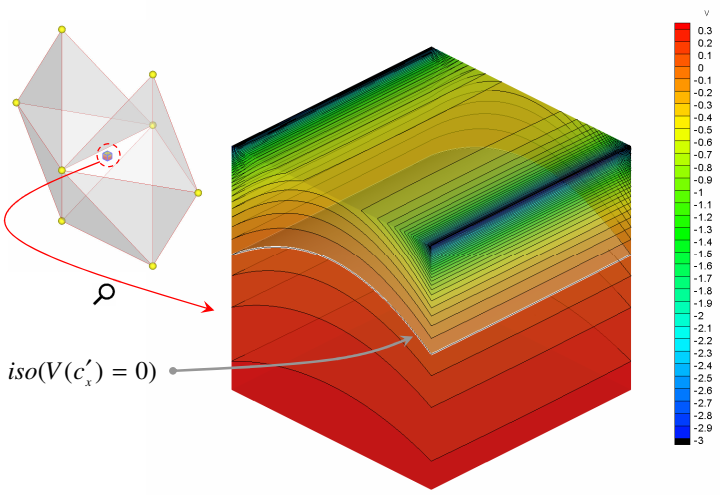

FIG. 3.24 – Lieux de l'espace où le volume de la cellule du nœud x est nul

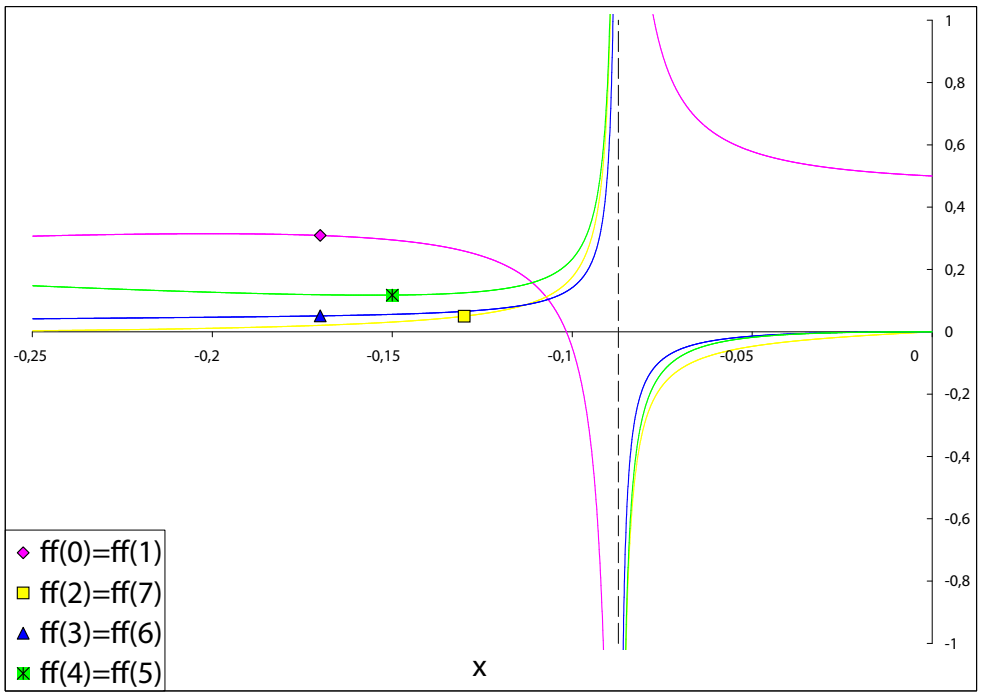

FIG. 3.25 – Fonction de formes Laplace en x pour les différents voisins

3.4 discrétisation du domaine en vue de l'intégration numérique

Comme nous l'avons vu en 2.3, pour un domaine donné Ω, l'intégration nodal nécessite une discrétisation en volumes élémentaires entourant chaque nœud.

3.4.1 Cas des domaines convexes ou faiblement non convexes

Dans le cas de domaines convexes ou faiblement non convexes, l'utilisation des cellules de Voronoï constitue une partition "naturelle" du domaine *(figure 3.26)*. Il est néanmoins nécessaire de tronquer les cellules qui débordent du domaine, cellules dont le nœud est sur la frontière, et certaines dont le nœud est proche de la frontière. Dans cette partie nous allons présenter les algorithmes mis en œuvre afin de tronquer ces cellules par la frontière, ils ne sont applicables que s'il n'y a pas de cellules non convexes s'auto-intersectant à l'intérieur du domaine, ce qui peut se produire si ce dernier est localement fortement non convexe *(voir 3.4.2)*.

Le fait de disposer de tels algorithmes nous a permis d'étudier l'influence du choix fait pour la partition du domaine sur la qualité des résultats obtenus par la CNEM *(voir 4.3)*.

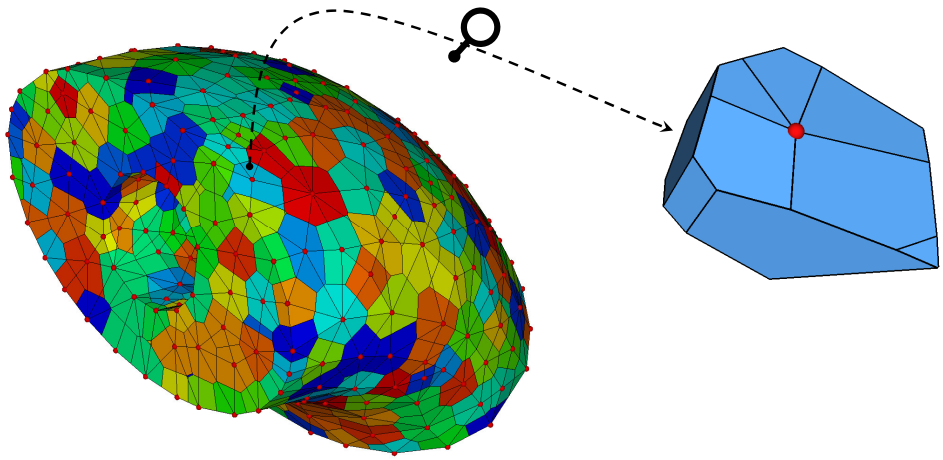

FIG. 3.26 – Exemple de cellule de Voronoï tronquée par la frontière

Si l'on étudie une cellule de Voronoï tronquée par la frontière *(figure 3.27)*, on constate que cette dernière est composée de faces de Voronoï (*faces violettes*), de faces de Voronoï tronquées par la frontière (*faces bleues*), et de sous-faces de la triangulation de la frontière (*faces jaunes*). Le contour des faces de Voronoï tronquées par la frontière, est composé d'arêtes de Voronoï, d'arêtes de Voronoï tronquées par la frontière, et d'arêtes se situant sur la triangulation de la frontière. Les sous-faces

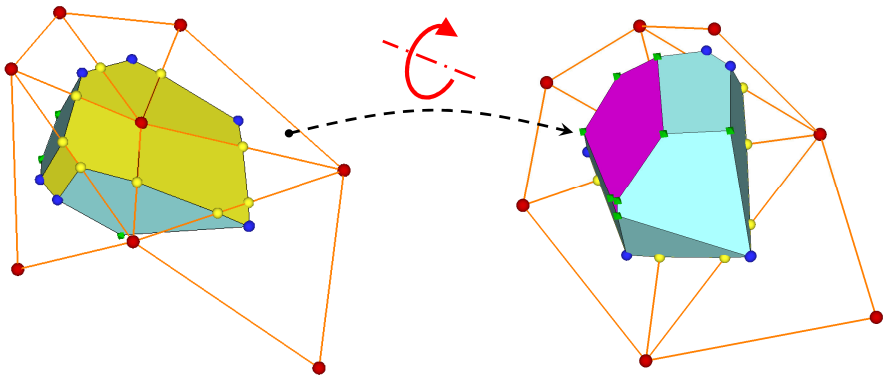

FIG. 3.27 – Analyse d'une cellule de Voronoï tronquée par la frontière

de la triangulation de la frontière sont formées exclusivement d'arêtes se situant sur la frontière. Les arêtes de Voronoï tronquées par la frontière peuvent avoir comme sommets, un sommet de Voronoï ou un point d'intersection [arête Voronoï/face triangulation] (*points bleus*). Les arêtes se situant sur la triangulation de la frontière peuvent avoir comme sommet, un point d'intersection [arête triangulation/face Voronoï] (*points jaunes*), un point d'intersection [arête Voronoï/face triangulation], ou encore un nœud. Les nouveaux sommets des faces décrivant une cellule tronquée peuvent ainsi être de trois types :

1. des nœuds,
2. des points intersection entre arêtes de triangulation et faces de Voronoï,
3. des points intersection entre arêtes de Voronoï et faces de triangulation.

La détermination des sommets de la première famille *(les nœuds)* est immédiate. La détermination de ceux des deux familles suivantes l'est moins, en particulier dans un contexte d'optimisation des algorithmes. Cette détermination fait l'objet, successivement pour chaque famille, des paragraphes ***a*** et ***b***.

Une fois tous ces sommets construits, il ne reste plus qu'à les utiliser pour décrire les nouvelles arêtes et nouvelles faces des cellules tronquées par la frontière. Ceci est décrit dans le paragraphe ***c***.

a)Points d'intersection entre arêtes de la triangulation de la frontière et faces de Voronoï

Soit a^T une arête de la triangulation de la frontière, et soient n^a et n^b les deux nœuds extrémités de a^T. Soient enfin c_{n^a} et c_{n^b} les deux cellules de de Voronoï associées à n^a et n^b. En plus de ces deux cellules, l'arête a^T peut transpercer plusieurs autres cellules c_{n^x} *(figure 3.28)*. Les cellules transpercées par a^T forment une chaîne de cellules adjacentes *(cellules qui partagent une face de Voronoi : face transpercée par* a^f*)* dont les deux extrémités sont les cellules c_{n^a} et c_{n^b}. Etant donné que les cellules sont supposées être convexes à l'intérieur du domaine, on peut affirmer que c_{n^a} et c_{n^b}

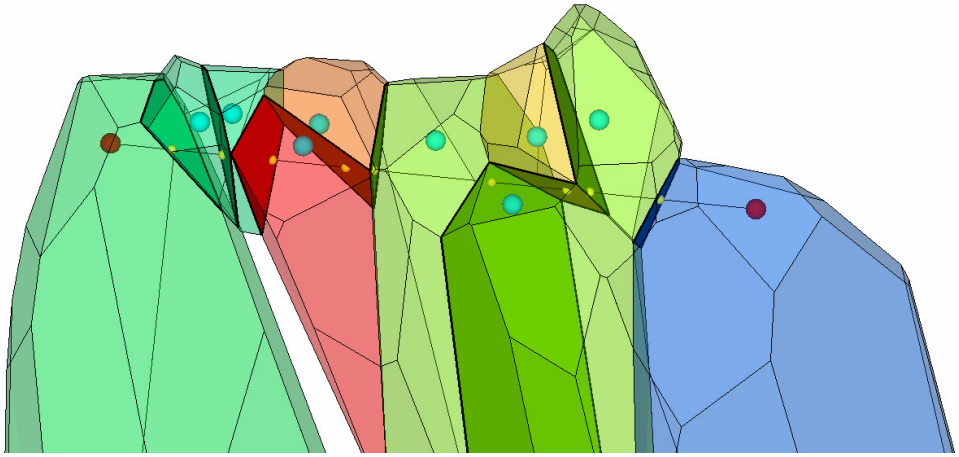

FIG. 3.28 – Intersection entre une arête de la triangulation et les faces de Voronoï

ne sont transpercées qu'une seule fois par a^T, et que les cellules c_{n^x} sont transpercées deux fois par a^T.

Algorithme :

Données d'entrée :
- Le diagramme de Voronoï contraint.
- La liste des arêtes de la triangulation de la frontière, avec :
  pour chaque arête a^T ses deux nœuds n_1a^T et n_2a^T, ainsi que les deux triangles auxquels elle appartient t_1a^T, t_2a^T.

Données de sortie :
- Les points d'intersections [arête triangulation/face Voronoï] avec :
  pour chaque point p ses coordonnées $[XYZ]p$, l'arête de la triangulation a^Tp et la face de Voronoï f^Vp qui l'ont engendré.
  Ces points, ainsi que les points d'intersection [arête Voronoï/face triangulation] que l'on verra dans la section suivante, seront stockés dans la liste des sommets. On rajoutera une variable à la structure de données associée à l'entité *sommet* afin d'indiquer son type.
- Pour chaque arête de la triangulation a^T la liste de faces de Voronoï qu'elle intersecte L^fa^T.

Pour chaque arête a^T de la triangulation de la frontière **faire**
  n : nœud tampon
  f : face tampon
  $n \leftarrow n_0 a^T$
  $f \leftarrow \{\}$
  Répéter
    f^{int} : face tampon
    Si (n=$n_0 a^T$) **Alors**
      $f^{int} \leftarrow$ face de c_n intersectée par a^T
    Sinon
      $f^{int} \leftarrow$ face de $c_n \neq$ de f, intersectée par a^T
    Fin Si
    p^{int} : sommet tampon
    $[XYZ]p^{int} \leftarrow$ Coordonnée du point d'intersection de a^T avec f^{int}
    $a^T p^{int} \leftarrow a^T$
    $f^V p^{int} \leftarrow f^{int}$
    Stocker p^{int} dans la liste des sommets du diagramme de Voronoï
    Rajouter f^{int} à $L^f a^T$
    $n \leftarrow$ nœud de la face $f^{int} \neq n$
    $f \leftarrow f^{int}$
  jusqu'à ce que ($n = n_1 a^T$)
Fin Pour

Algorithme 4: Intersection [arête triangulation/face Voronoï]

b)Points d'intersection entre arêtes de Voronoï et faces de la triangulation de la frontière

Soit f^T une face de la triangulation de la frontière. Chacune des 3 arêtes de f^T transperce un ensemble de faces de Voronoï. Soit F^b *(faces bleues figure 3.29)* l'union des trois ensembles de faces intersectées par les trois arêtes de f^T. F^b "borne" l'ensemble F^j *(faces jaunes figure 3.29)* des faces de Voronoï intersectées par la face triangulaire f^T.

Il est ainsi possible, grâce aux faces de l'ensemble F^b, de reconstituer par adjacence les faces de l'ensemble F^j, et par conséquence de trouver les points d'intersection [arête Voronoï/face triangulation], car tous ces derniers se situent sur des arêtes de Voronoï appartenant aux faces de l'ensemble F^j.

A chaque fois que l'on trouve un point d'intersection [arête Voronoï/face triangulation], on crée une arête de Voronoï tronquée, qui sera un attribut de chaque arête de Voronoï qui intersecte la frontière. Avant de créer une telle arête, on doit vérifier au préalable si cela n'a pas déjà été fait, car il possible qu'une arête de Voronoï puisse être tronquée des deux cotés.

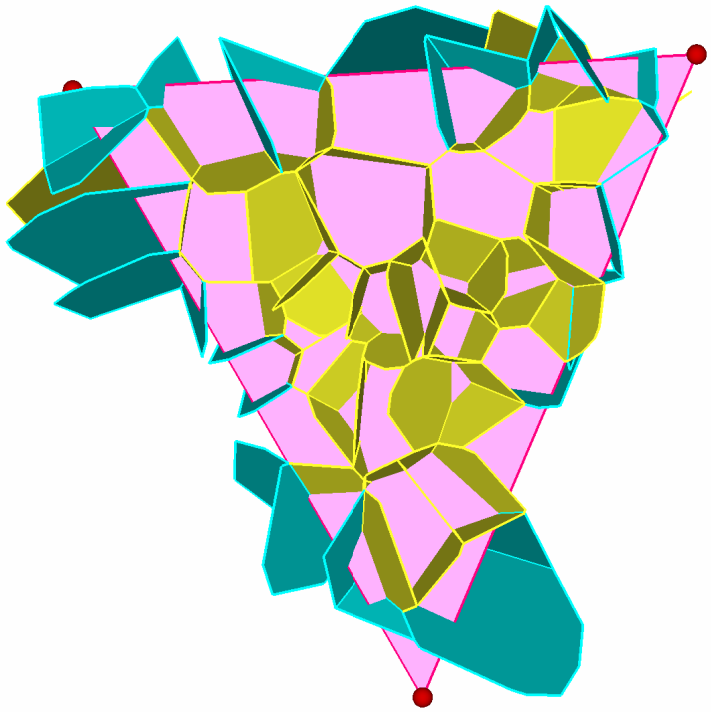

FIG. 3.29 – Intersections [arête Voronoï/face triangulation]

Algorithme :

Données d'entrée :

- Le diagramme de Voronoï contraint.
- Pour chaque arête de Voronoï a, les 3 faces de Voronoï aux quelles elle appartient f_1a f_2a et f_3a.
- La liste des triangles de la frontière.

Données de sortie :

- Les points d'intersections [arête Voronoi/face triangulation] avec :
  pour chaque point p ses coordonnées $[XYZ]p$, l'arête de Voronoi a^Vp et la face de la triangulation f^Tp qui l'ont engendré.

- Pour chaque arête de Voronoï a qui intersecte la frontière, son arête tronquée $a^{trq}a$.

Pour chaque triangle f^T de la triangulation de la frontière **faire**
  $L^{(f-a)}$: liste tampon de couples *(face Voronoï,arête Voronoi)*
  Pour chaque arête a du triangle f^T **faire**
    Pour chaque face f de $L^f a$ **faire**
      Si (f présente dans un couple de $L^{(f-a)}$) **Alors**
        Supprimer ce couple de $L^{(f-a)}$
      Sinon
        Rajouter le couple $(f, \{\})$ à $L^{(f-a)}$
      Fin Si
    Fin Pour
  Fin Pour
  Tant que ($L^{(f-a)}$ non vide) **faire**
    Retirer le couple (f^{queue},a^{queue}) en queue de $L^{(f-a)}$
    a : arête tampon
    $a \leftarrow$ arête de $f^{queue} \neq$ de a^{queue} (*si elle existe*) qui intersecte la face du triangle f^T
    p^{int} : sommet tampon
    $[XYZ]p^{int} \leftarrow$ Coordonnées du point d'intersection de a avec f^T
    $a^V p^{int} \leftarrow a$
    $f^T p^{int} \leftarrow f^T$
    Stocker p^{int} dans la liste des sommets du diagramme de Voronoï
    Si (a possède déjà une arête tronquée) **Alors**
      Remplacer dans $a^{trq}a$ le sommet qui est au dessus du triangle f^T par p^{int}
    Sinon
      a^{trq} : arête tampon
      $a^{trq} \leftarrow a$
      Remplacer dans a^{trq} le sommet qui est au dessus du triangle f^T par p^{int}
      Stocker a^{trq} dans la liste des arêtes du diagramme de Voronoï
      $a^{trq}a \leftarrow a^{trq}$
    Fin Si
    Pour i **de** 0 **à** 2 **faire**
      Si ($f_i a \neq f^{queue}$) **Alors**
        Si ($f_i a$ présente dans un couple de $L^{(f-a)}$) **Alors**
          Supprimer ce couple de $L^{(f-a)}$
        Sinon
          Rajouter le couple ($f_i a$,a) à $L^{(f-a)}$
        Fin Si
      Fin Si
    Fin Pour
  Fait
Fin Pour

Algorithme 5: Intersection [arête Voronoï/face triangulation]

c)Construction des cellules de Voronoï tronquées par la frontière

Cette construction se fait en 2 étapes :

Etape 1 : Création des faces de Voronoï tronquées par la frontière

Grâce aux trois faces de Voronoï auxquelles chaque arête de Voronoï tronquée appartient, on recense toutes les faces de Voronoï qui intersectent la triangulation de la frontière. Ces faces sont par la suite rassemblées dans une liste. Pour chaque face f de cette liste, on crée une face de Voronoï tronquée f^{trq}, qui est un attribut de f. Les arêtes de f^{trq} peuvent être de trois type :

- Type a : Arêtes de Voronoï.
- Type b : Arêtes de Voronoï tronquées par la frontière *(déjà créées)*.
- Type c : Arêtes se situant sur la triangulation de la frontière *(à créer)*.

Ces dernières ont pour extrémités des points d'intersections [arête triangulation/face Voronoï], ou [arête Voronoï/face Triangulation], appartenant à la face de Voronoï considérée, et se situant sur une même face de la triangulation. Le contour d'une face de Voronoï tronquée par la frontière forme une séquence d'arêtes $\boxed{\text{b}} \circ \{n \times \boxed{\text{a}}\} \circ \boxed{\text{b}} \circ \{m \times \boxed{\text{c}}\}$, qui se répète une ou plusieurs fois. Afin de former une séquence, on recherche dans la face f une arête qui possède une arête tronquée et dont le premier sommet et un point d'intersection, on rajoute cette arête à f^{trq}. Si le second point de cette arête n'est pas un point d'intersection, on rajoute à f^{trq} toutes les arêtes de f qui suivent jusqu'a ce que l'on tombe sur une arête qui possède une arête tronquée dont le second sommet est un point d'intersection. Soit p ce point, on recherche parmi les points d'intersection de la face associée à f, le point qui se situe sur la même face que p, et l'on crée l'arête reliant ces deux points *(arête de type c)*. Et ainsi de suite jusqu'à ce que le second point de l'arête créée soit un point d'intersection [arête Voronoï/face triangulation]. On procède de la même façon pour former les séquences qui suivent, jusqu'à ce que le contour de f^{trq} soit totalement formé.

Algorithme :

Données d'entrée :

- Le diagramme de Voronoi contraint.
- La liste des faces à tronquer L^f.
- La liste des points d'intersection associés à chaque face f de L^f, $L^P f$.

Données de sortie :

- Pour chaque face de Voronoï f qui intersecte la frontière, sa face tronquée $f^{trq} f$.

Pour chaque face à tronquer f de L^f **faire**
  $nb_a \leftarrow$ nombre d'arêtes de la face f
  $i \leftarrow 0$
  f^{trq} : face tampon
  Répéter
    Tant que ($a_i f$ n'a pas d'arête tronquée $\vee$ $s_0 a^{trq} a_i f$ n'est pas un point d'intersection) **faire**
      $i \leftarrow (i+1) mod(nb_a)$
    Fait
    Rajouter $a^{trq} a_i f$ à f^{trq}
    Tant que ($a_i f$ n'a pas d'arête tronquée $\vee$ $s_1 a^{trq} a_i f$ n'est pas un point d'intersection) **faire**
      $i \leftarrow (i+1) mod(nb_a)$
      Si ($a_i f$ n'a pas d'arête tronquée) **Alors**
        Rajouter $a_i f$ à f^{trq}
      Sinon
        Rajouter $a^{trq} a_i f$ à f^{trq}
      Fin Si
    Fait
    s^a : sommet tampon
    $s^a \leftarrow s_1 a^{trq} a_j f_{tmp}$
    Retirer s^a de $L^p f$
    Tant que (il y a un point s^b de $L^p f$ sur le même triangle que s^a) **faire**
      Retirer s_b de $L^{p^{int}} f$
      a : arête tampon
      $s_0 a \leftarrow s^a$
      $s_1 a \leftarrow s^b$
      Stocker a dans la liste des arêtes du diagramme de Voronoï
      Rajouter a à f^{trq}
      $s^a \leftarrow s^b$
    Fait
  jusqu'à ce que ($L^p f$ soit vide)
  Rajouter f^{trq} à la liste des faces du diagramme de Voronoï
  $f^{trq} f \leftarrow f^{trq}$
Fin Pour

Algorithme 6: Création des faces de Voronoï tronquées par la frontière

Etape 2 : Création des cellules de Voronoï tronquées par la frontière

Cette étape est constituée elle-même de deux sous étapes :

Etape 2-1 :

Grâce aux nœuds des faces de Voronoï qui intersectent la frontière on recense toutes les cellules qui intersectent la frontière que l'on rassemble dans une liste. Pour chaque cellule c de cette liste, on crée une cellule de Voronoï tronquée c^{trq}, qui sera un attribut de c. On commence par rajouter à c^{trq} les faces tronquées de c, gârce à ces faces, et par adjacence, il est possible de retrouver les faces de c non tronquées internes au domaine, que l'on rajoute à c^{trq}.

Algorithme :

Données d'entrée :
- Le diagramme de Voronoi contraint.
- La liste des cellules à tronquer L^c.

Données de sortie :
- Pour chaque cellule de Voronoi c qui intersecte la frontière, sa cellule tronquée $c^{trq}c$ à laquelle il ne manque que les faces se situant sur la frontière du domaine.

Pour chaque cellule à tronquer c de L^c **faire**
  c^{trq} : cellule de Voronoi tampon
  $L^{(a-n)}$: liste de couples $(arête, nœud)$ tampon
  L^f : liste de faces de Voronoi tampon
  Pour chaque face f de c **faire**
    Si (f a une face tronquée) **Alors**
      Rajouter $f^{trq}f$ à la liste de faces de c^{trq}
      Pour chaque arête a de $f^{trq}f$ **faire**
        Si (a arête de Voronoi non tronquée) **Alors**
          n : nœud tampon
          $n \leftarrow$ nœud de l'arête $a \neq$ de $n_0 f$ et $n_1 f$
          Rajouter le couple (a,n) à $L^{(a-n)}$
        Fin Si
      Fin Pour
    Sinon
      Rajouter f à L^f
    Fin Si
  Fin Pour
  Tant que ($L^{(a-n)}$ non vide) **faire**
    Retirer le couple (a^{queue}, n^{queue}) en queue de $L^{(a-n)}$
    Trouver et retirer dans L^f la face f dont un des deux nœuds est n^{queue}
    Rajouter f à c^{trq}
    Pour chaque arête a de f **faire**
      Si ($a \neq a^{queue}$) **Alors**
        Si (le couple $(-a, n^{queue})$ existe dans $L^{(a-n)}$) **Alors**
          Supprimer ce couple de $L^{(a-n)}$
        Sinon
          n : nœud tampon
          $n \leftarrow$ nœud de l'arête $a \neq$ de $n_0 f$ et $n_1 f$
          Rajouter le couple (a,n) à $L^{(a-n)}$
        Fin Si
      Fin Si
    Fin Pour
  Fait
  $c^{trq}c \leftarrow c^{trq}$
Fin Pour

Algorithme 7: Création des cellules de Voronoï tronquées par la frontière - 1/2

Etape 2-2 :

L'étape précédente produit des cellules tronquées auxquelles il ne manque que des faces se situant sur la frontière du domaine. Ces faces sont donc des sous-faces des triangles de la frontière. Le but de cette étape est de les créer. Pour chaque cellule tronquée, les sous-faces de la triangulation qui lui manquent ont pour arêtes :

– les arêtes de ses faces de Voronoï tronquées par la frontière, qui se situent sur la frontière *(déjà créées)*.
– des sous-arêtes d'arêtes de la triangulation de la frontière *(à créer)*.

Ces dernières ont pour sommets soit un point d'intersection [arête triangulation/face Voronoï], soit un nœud *(nœud de la cellule)*. On rassemble dans une liste tous les points intersections [arête triangulation/face Voronoï] de la cellule concernée. Pour chaque point de cette liste on recherche s'il existe un autre point dans cette même liste qui se situe sur la même arête de la triangulation. Si oui, on crée l'arête qui relie ces deux points. Si non, on crée l'arête reliant ce point au nœud de la cellule.

Grâce à l'ensemble des arêtes des deux types précédents, il est possible de créer les faces manquantes en rassemblant les arrêtes qui se situent sur une même face de la triangulation. Ces arêtes seront chaînées par la suite.

Algorithme :

Données d'entrée :
– Le diagramme de Voronoi contraint.
– La liste des cellules tronquées L^c.
Données de sortie :
– Pour chaque cellule tronquée c les faces qui lui manquent se situant sur la frontière du domaine.

Pour chaque cellule tronquée c de L^c **faire**
- $L^{(a-t)}$: liste de couples $(arête, triangle)$ tampon
- Récupérer dans les faces tronquées de c les arêtes se situant sur un triangle de la frontière, et rajouter les couples associés à $L^{(a-t)}$
- L^p : liste de points d'intersection tampon
- Récupérer dans les faces tronquées de c les points d'intersections [arête triangulation/face Voronoï], et les rajouter à L^p
- **Tant que** (L^p non vide) **faire**
  - a : arête tampon
  - Retirer le point p^{queue} en queue de L^p
  - a^T : arête de la triangulation tampon
  - $a^T \leftarrow a^T p^{queue}$
  - **Si** (un des deux nœuds de a^T est le nœud nc) **Alors**
    - $s_0 a = p^{queue}$
    - $s_1 a = nc$
  - **Sinon**
    - Rechercher dans L^p le point p se situant sur a^T
    - $s_0 a = p^{queue}$
    - $s_1 a = p$
  - **Fin Si**
  - Stocker a dans la liste des arêtes du diagramme de Voronoï
  - Rajouter le couple $(a, t_0 a^T)$ à la liste de couple $L^{(a-t)}$
  - Rajouter le couple $(a, t_1 a^T)$ à la liste de couple $L^{(a-t)}$
- **Fait**
- **Tant que** ($L^{(a-t)}$ non vide) **faire**
  - Retirer le couple (a^{queue}, t^{queue}) en queue de $L^{(a-t)}$
  - f face tampon
  - Rajouter a^{queue} à f
  - Retirer tous les couples ayant t^{queue} comme triangle, et rajouter leur arête à f
  - Ranger et orienter les arêtes de la face f
  - Stocker f dans la liste des faces du diagramme de Voronoï
  - Rajouter f à la liste de faces de c
- **Fait**

Fin Pour

Algorithme 8: Création des cellules de Voronoï tronquées par la frontière - 2/2

3.4.2 Cas des domaines fortement non convexes

Pour certains domaines fortement non convexes, la discrétisation par le biais des cellules de Voronoï décrite dans la section précédente devient problématique. Les algorithmes proposés ont comme hypothèse de départ la convexité des cellules de Voronoï à l'intérieur du domaine, hypothèse qui peut ne plus être vérifiée pour des domaines fortement non convexes. Pour de tels domaines certaines cellules associées à des nœuds de la frontière ou proches de celle-ci deviennent non convexes à l'intérieur du domaine. Il est possible de mettre en évidence ce phénomène par l'exemple illustré figure 3.32 *(le domaine pris est présenté figure 3.31)*. Sur cette figure on peut voir que la cellule de Voronoï associée au nœud n est non convexe à l'intérieur du domaine *(le point entouré avec un cercle appartient à la courbe d'auto-intersection de la cellule du nœud n, et est à l'intérieur du domaine)*.

Ce problème peut être contourné en utilisant une partition autre que celle basée sur les cellules de Voronoï. La plus évidente et la plus simple à mettre en œuvre, consiste à subdiviser chaque tétraèdre de la tétraèdrisation de Delaunay contrainte que l'on a à disposition en quatre parties, et d'associer chaque partie au nœud du tétraèdre auquel elle est rattachée. On obtient ainsi une partition qui associe un volume élémentaire à chaque nœud du domaine *figure 3.30*.

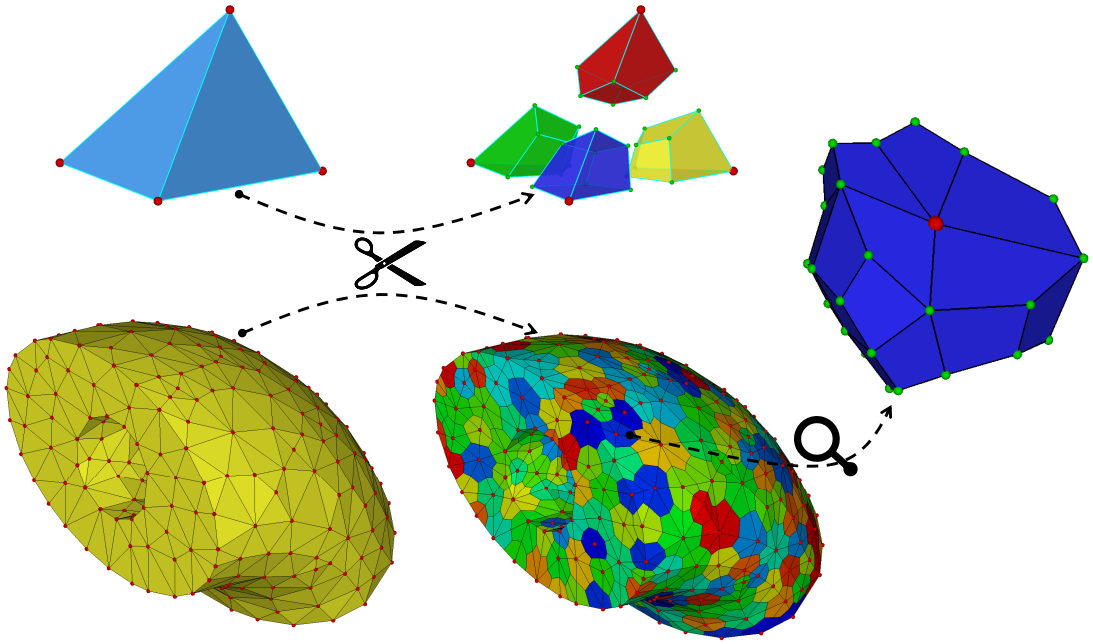

FIG. 3.30 – Discrétisation obtenue par le biais des quarts de tétraèdre

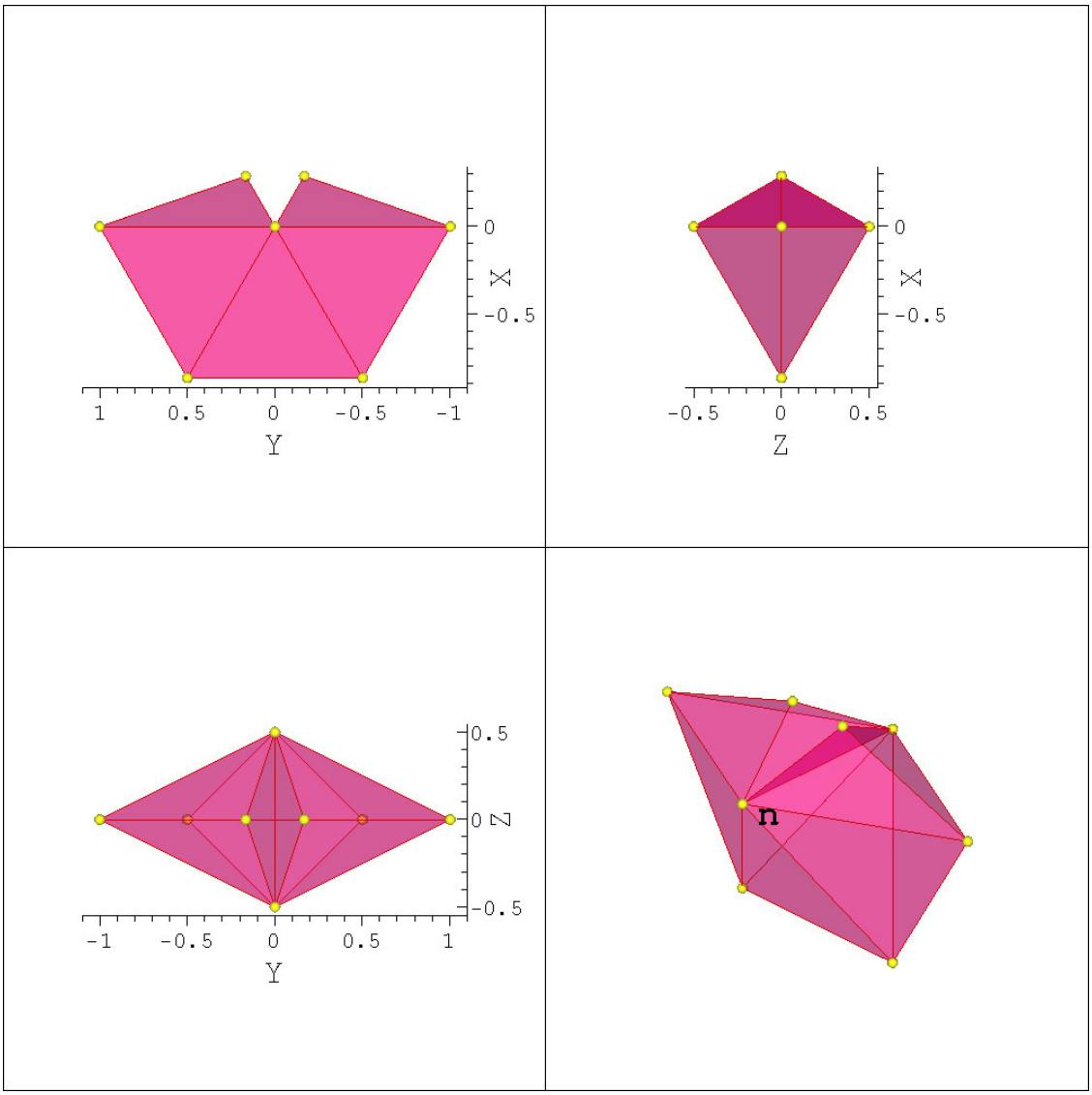

FIG. 3.31 – Domaine produisant des cellules de Voronoï non convexes en son intérieur

3.5 Conclusion

Dans ce chapitre ont été présentés les algorithmes associés à la mise en œuvre de la CNEM en 3*d*. Grâce à l'utilisation de l'algorithme de Watson, les temps de calcul des fonctions de formes Sibson ont été pratiquement divisés par un facteur 100. En cas d'échec de ce dernier (*environ* $\frac{1}{1000}$) l'algorithme proposé par Boissonnat et al garantit leurs calculs.
Pour la discrétisation du domaine en volumes élémentaires en vue de l'intégration SCNI, et dans le cas de domaine convexe et faiblement non convexe, les cellules de Voronoï *(internes+tronquées par le bord)* sont une solution. Un algorithme robuste et local est proposé afin de tronquer les cellules de Voronoï qui débordent sur la frontière.

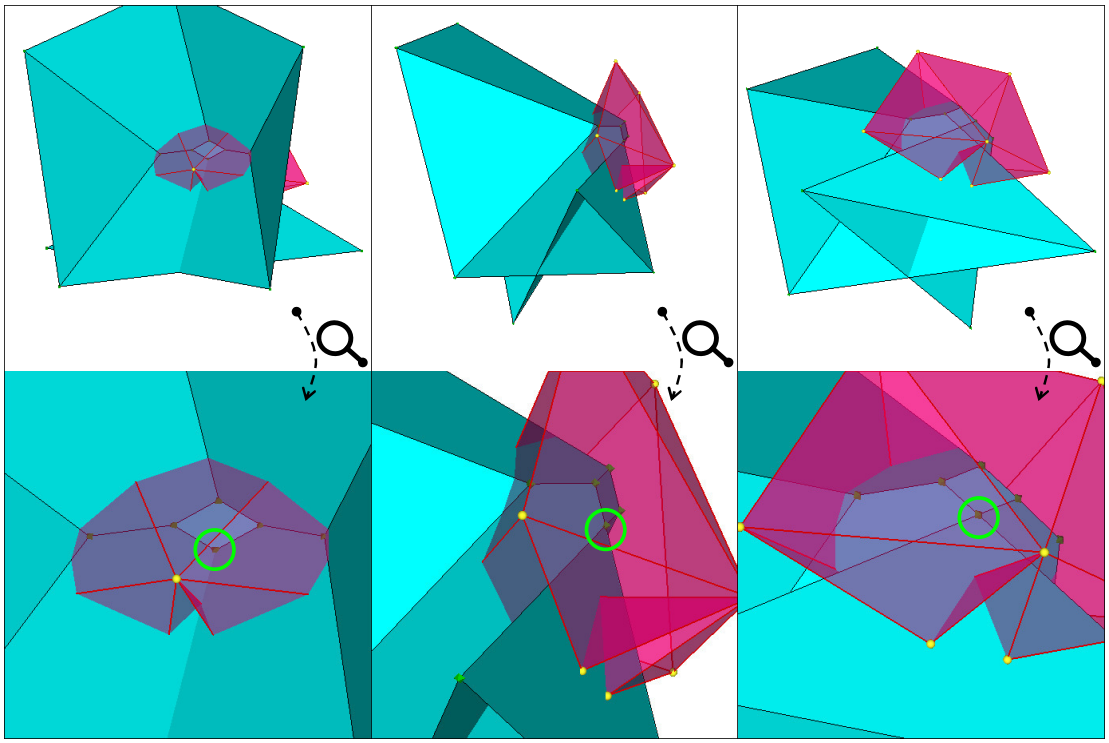

FIG. 3.32 – Cellule de Voronoï non convexe à l'intérieur du domaine

Dans le cas général, la discrétisation basée sur les quarts de tétraèdres rattachés aux nœuds constitue une solution efficace à moindre coût.

Chapitre 4

Validation

Avant d'appliquer la CNEM à des problèmes en transformation finies, nous présentons dans ce petit chapitre, deux exemples en petites perturbations pour des problèmes tridimensionnels d'élasticité.
Ces exemples permettent de valider l'aptitude de la CNEM, en $3d$, à traiter de tels problèmes et de comparer les résultats obtenus avec la CNEM à ceux obtenus avec les éléments finis *(éléments tétraédriques à 4 nœuds)*. Ces exemples ont également permis de calibrer certains paramètres pour l'intégration nodale stabilisée.
Après avoir précisé la nature des simulations réalisées *(identique pour les deux exemples)* ainsi que la mesure d'erreur employée, nous donnons les résultats obtenus pour ces deux exemples : la sphère creuse sous pression et l'excavation cylindrique dans un milieu "infini".
Nous étudions enfin l'importance des choix faits pour l'intégration *(volume entourant les nœuds, nombre de points d'intégration)* quant à la précision des résultats obtenus.

4.1 Simulations menées et résultats

Afin de valider la CNEM en 3d, nous avons réalisé une série de simulations pour chacun des deux exemples choisis. Notre choix s'est porté sur deux problèmes types en élasticité *(hypothèse petites perturbations)* ayant une solution analytique [MH01] : la sphère creuse sous pression et l'excavation cylindrique dans un milieu infini soumise à des contraintes à l'infini.

La norme de l'erreur utilisée est la norme en énergie de déformation :

$$\|e\| = \sqrt{\frac{\int_\Omega (\sigma - \hat{\sigma}) : C^{-1} : (\sigma - \hat{\sigma}) d\Omega}{\int_\Omega \hat{\sigma} : C^{-1} : \hat{\sigma} d\Omega}} \tag{4.1}$$

avec :
- C : le tenseur d'élasticité,
- σ : le tenseur de contraintes calculées,
- $\hat{\sigma}$: le tenseur de contraintes exactes.

La résolution du système linaire engendré est réalisée grâce au solveur itératif *(gradient conjugué)* creux de la librairie mathématique MKL [Int].

Les caractéristiques du calculateur utilisé sont les suivantes :
- OS : Linux RedHat 64 bits
- Compilateur : Intel c++ compiler 9.1
- Processeur : Intel Itanium 2 - 1,6GHz
- RAM : 32G - DDR2 - 667MHz

Pour chacun des deux problèmes, et pour une distribution nodale homogène donnée, les simulations suivantes ont été menées :
- Intégration de Gauss de degré 4, avec fonction de forme Sibson.
- Intégration de Gauss de degré 0, avec fonction de forme éléments finis linéaires sur maillage de Delaunay.
- Intégration de Gauss de degré 0, avec fonction de forme éléments finis linéaires sur maillage optimisé.
- Intégration stabilisée, avec fonction de forme Sibson.
- Intégration stabilisée, avec fonction de forme Laplace.
- Intégration stabilisée, avec fonction de forme éléments finis linéaires sur maillage de Delaunay.
- Intégration stabilisée, avec fonction de forme éléments finis linéaires sur maillage optimisé.

Pour l'intégration de Gauss, elle est réalisée, selon le cas, soit sur les tétraèdres de Delaunay soit sur les tétraèdres du maillage éléments finis optimisé.

Pour l'intégration stabilisée, elle est faite sur les volumes nodaux issus de la discrétisation quart de tétraèdre *(tétraèdres proviennent, selon le cas, soit du maillage de Delaunay soit du maillage éléments finis optimisé)*, en prenant un point de Gauss par face *(quadrangle)*. Ce choix est justifié un peu plus loin dans le paragraphe 4.3.

Les résultats obtenus pour la sphère creuse appellent aux mêmes commentaires que ceux obtenus pour l'excavation cylindrique. Ces commentaires sont donnés section 4.2.

4.1.1 Sphère creuse sous pression

Dans cet exemple, on considère une sphère creuse sous pression interne. La sphère est parfaitement élastique, et possède des propriétés mécaniques isotropes. La solution exacte du problème *(contraintes)* en coordonnées sphériques (r, θ, ϕ), est donnée par :

$$\sigma_{rr} = -\frac{{R_{int}}^3}{{R_{ext}}^3 - {R_{int}}^3}\left(\frac{{R_{ext}}^3}{r^3} - 1\right) P \tag{4.2}$$

$$\sigma_{\theta\theta} = \sigma_{\phi\phi} = -\frac{{R_{int}}^3}{{R_{ext}}^3 - {R_{int}}^3}\left(\frac{{R_{ext}}^3}{2r^3} + 1\right) P \tag{4.3}$$

$$\sigma_{r\theta} = \sigma_{r\phi} = \sigma_{\theta\phi} = 0 \tag{4.4}$$

Avec :
- P : la pression à l'intérieur de la sphère,
- R_{int} : rayon interne de la sphère,
- R_{ext} : rayon externe de la sphère.

Pour les simulations, la pression P régnant à l'intérieur de la sphère est prise égale à 1 MPa.
Vue la symétrie du problème on ne considère que le huitième de la sphère. Les conditions aux limites sont illustrées figure 4.1.

Les figures 4.2 et 4.3 montrent la répartition de la contrainte équivalente de Von Mises, issue du calcul CNEM avec fonction de forme Sibson et intégration SCNI, et ce pour deux nombres de nœuds différents*(3.500, et 300.000 nœuds)*. Ces champs de contrainte équivalente ne sont pas lissés : la contrainte est constante sur le volume entourant chaque nœud.

Les courbes de convergence et d'effectivité *(taux de convergence)* sont données figures 4.4 et 4.5.

A titre indicatif, le graphique 4.6 donne, pour le modèle de la sphère creuse comportant le plus grand nombre de nœuds, et pour chaque simulation :
- les temps de calcul associés à l'intégration + l'assemblage + l'imposition des conditions aux limites,
- les temps de calcul associés à la résolution du système linaire engendré,
- les valeurs chiffrées de l'erreur relative $\|e\|$ vis à vis de la solution exacte.

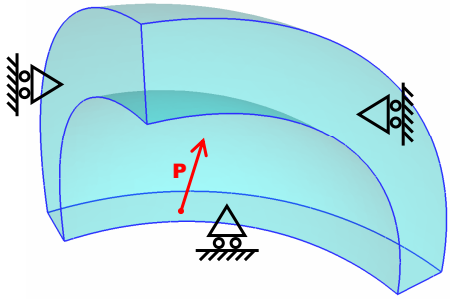

FIG. 4.1 – Conditions aux limites

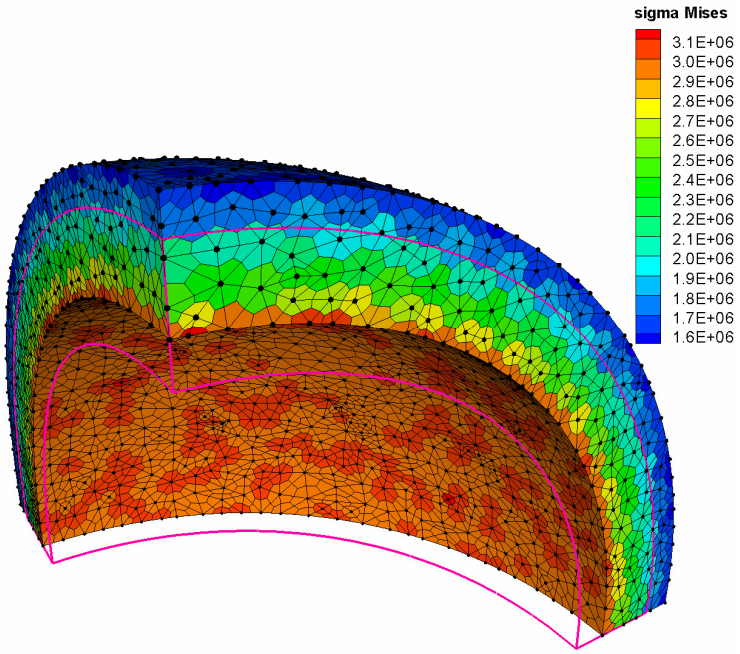

FIG. 4.2 – Contrainte équivalente de Von Mises - ≈ **3.500** nœuds *(integration stabilisée - FF Sibson)*

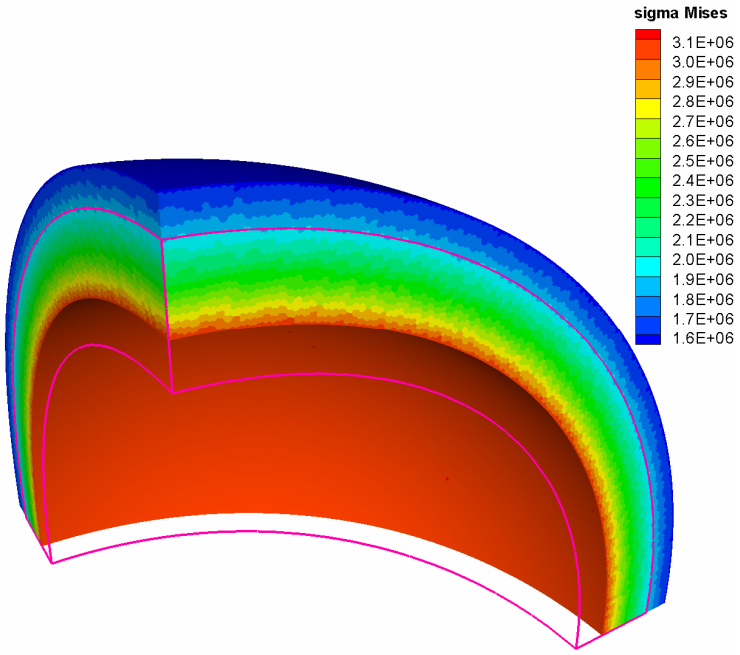

FIG. 4.3 – Contrainte équivalente de Von Mises - ≈ **300.000** nœuds *(integration stabilisée - FF Sibson)*

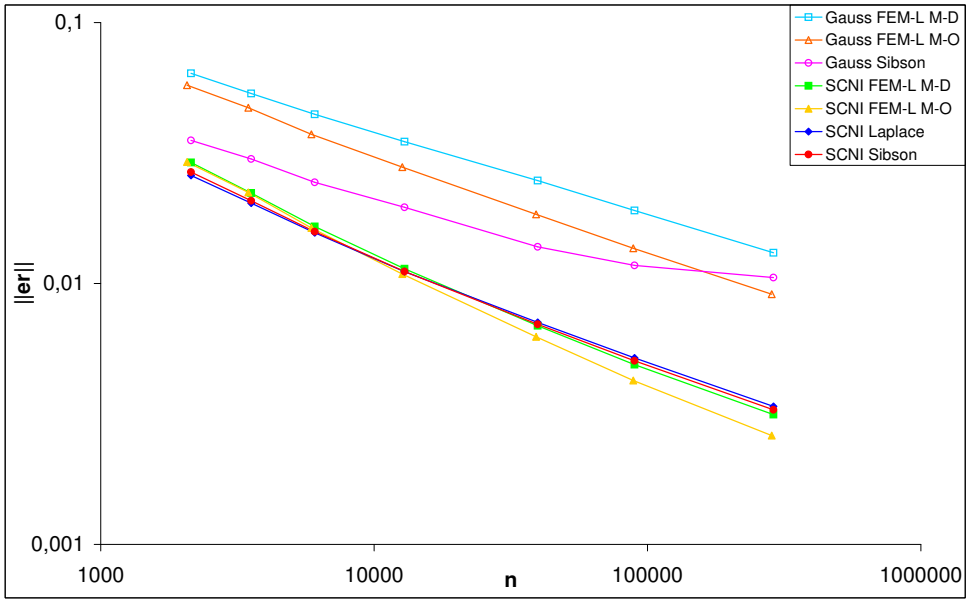

FIG. 4.4 – Courbes de convergence $\|e\| = f(nb\ n\text{œ}ud)$ - *double échelle logarithmique*

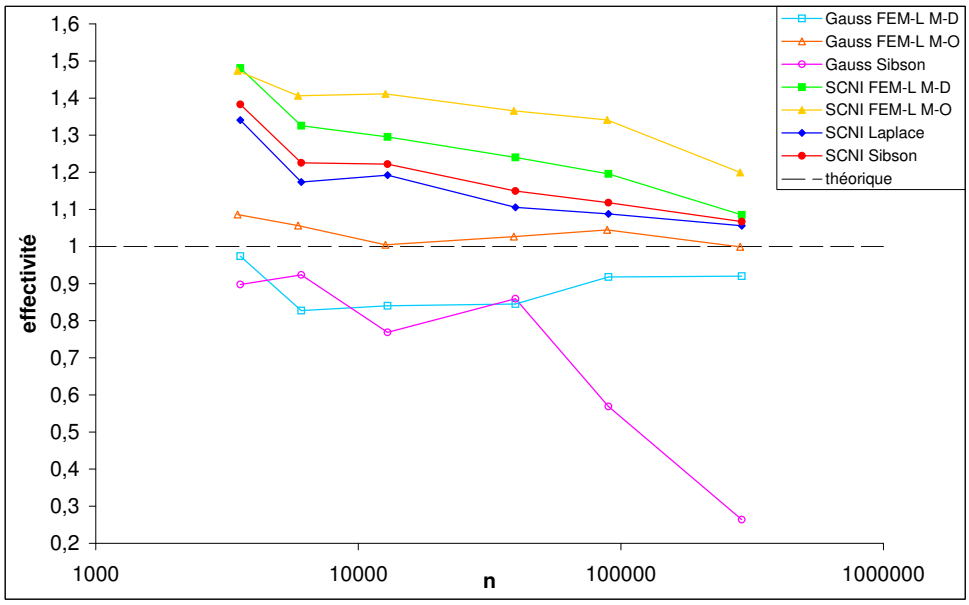

FIG. 4.5 – Courbes d'éffectivité *(degré de convergence) - échelle logarithmique pour le nombre de nœuds*

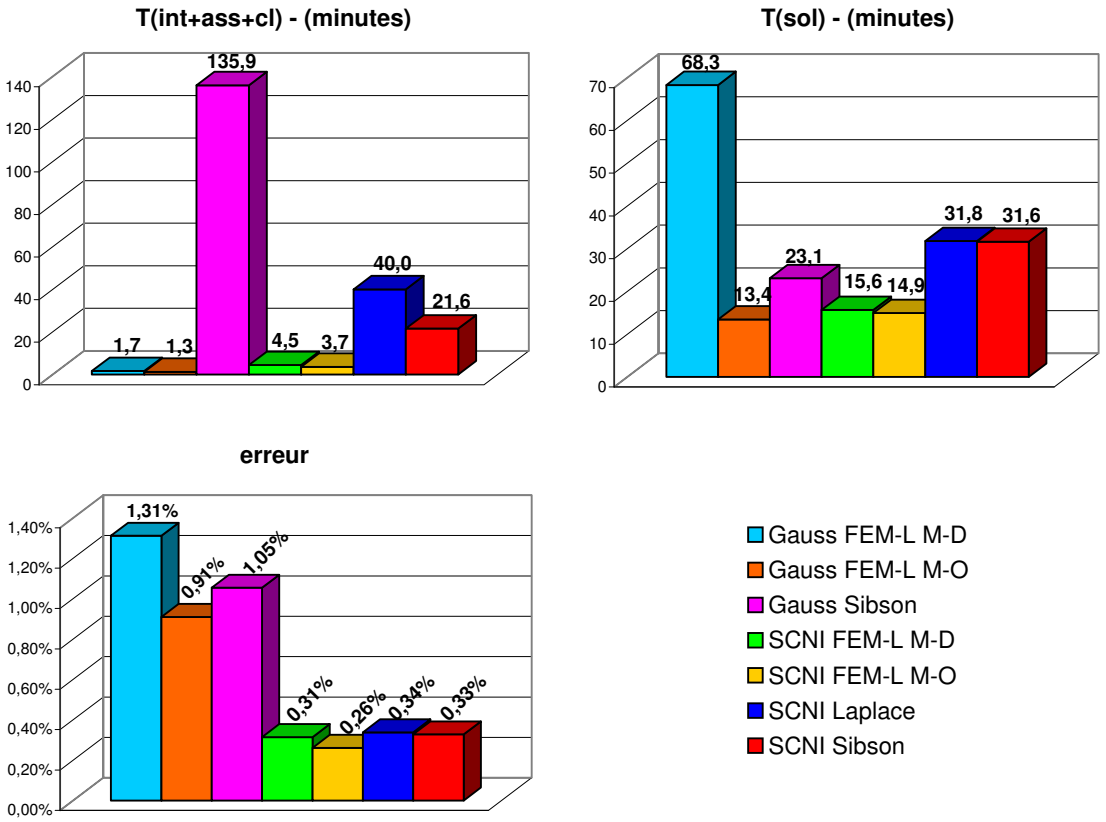

FIG. 4.6 – Temps de calcul et erreur - $\approx$ **300.000** nœuds

4.1.2 Excavation cylindrique dans un milieu infini

Dans ce deuxième exemple, on considère une excavation cylindrique dans un milieu élastique isotrope infini, soumise à des contraintes à l'infini dans le plan normal à l'axe du cylindre. On peut trouver une solution exacte du problème en déformations planes, dans [JC76] *(Coordonnées cylindriques (r,θ,z))* :

$$\sigma_{rr} = \frac{\sigma_{1\infty} + \sigma_{2\infty}}{2}\left(1 - \left(\frac{a}{r}\right)^2\right) + \frac{\sigma_{1\infty} - \sigma_{2\infty}}{2}\left(1 - 4\left(\frac{a}{r}\right)^2 + 3\left(\frac{a}{r}\right)^4\right) \tag{4.5}$$

$$\sigma_{\theta\theta} = \frac{\sigma_{1\infty} + \sigma_{2\infty}}{2}\left(1 + \left(\frac{a}{r}\right)^2\right) - \frac{\sigma_{1\infty} - \sigma_{2\infty}}{2}\left(1 + 3\left(\frac{a}{r}\right)^4\right) cos\,(2\theta) \tag{4.6}$$

$$\sigma_{r\theta} = -\frac{\sigma_{1\infty} - \sigma_{2\infty}}{2}\left(1 + 2\left(\frac{a}{r}\right)^2 - 3\left(\frac{a}{r}\right)^4\right) sin\,(2\theta) \tag{4.7}$$

$$\sigma_{zz} = \nu\,(\sigma_{rr} + \sigma_{\theta\theta}) \tag{4.8}$$

$$\sigma_{rz} = \sigma_{\theta z} = 0 \tag{4.9}$$

Pour les contraintes à l'infini on prend, $\sigma_{1\infty} = 0$, et $\sigma_{2\infty} = 1\ MPa$.
Etant donnée la symétrie du problème, seul le quart du domaine est utilisé pour les simulations. Les conditions aux limites sont illustrées figure 4.7.

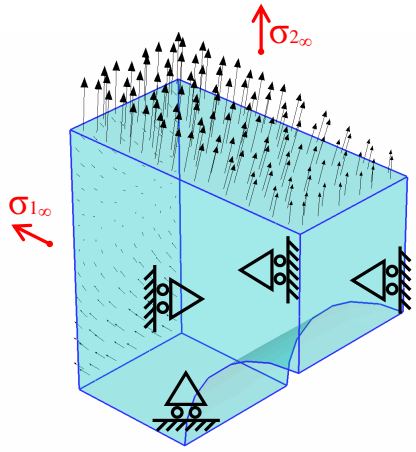

FIG. 4.7 – Conditions aux limites

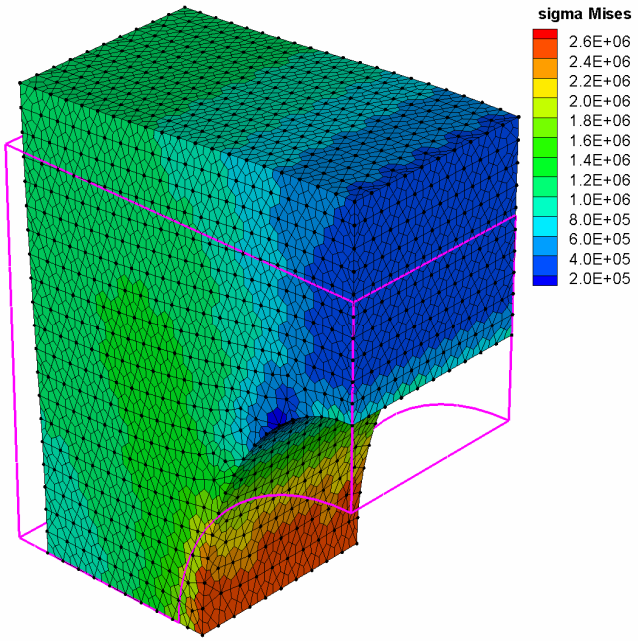

FIG. 4.8 – Contrainte équivalente de Von Mises - ≈ **5.000** nœuds *(integration stabilisée - FF Sibson)*

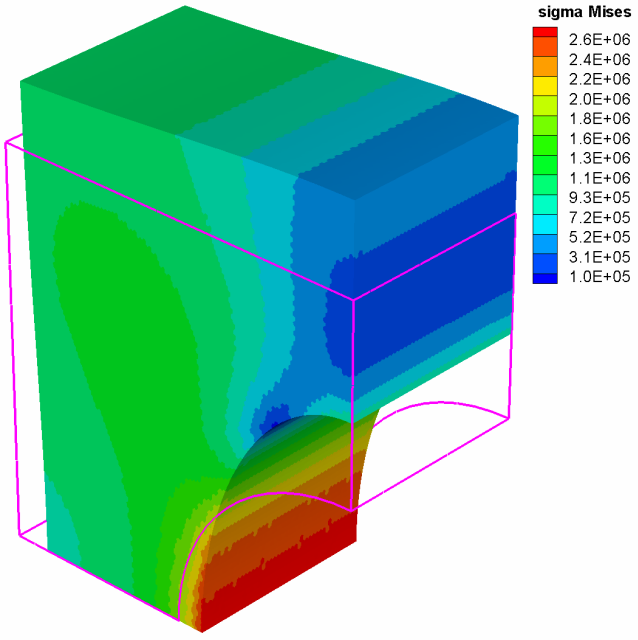

FIG. 4.9 – Contrainte équivalente de Von Mises - ≈ **370.000** nœuds *(integration stabilisée - FF Sibson)*

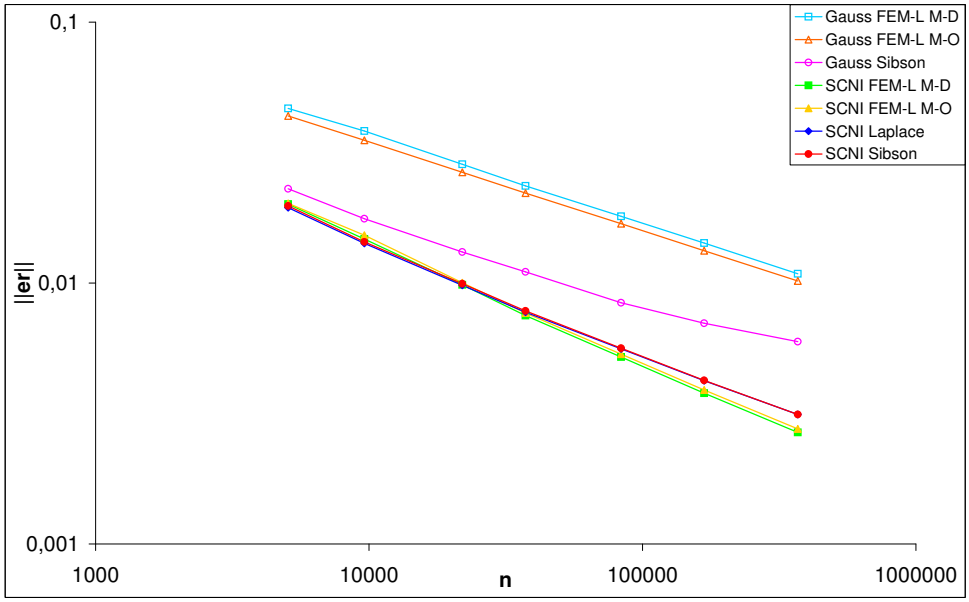

FIG. 4.10 – Courbes de convergence $\|e\| = f(nb\ n\text{œ}ud)$ - *double échelle logarithmique*

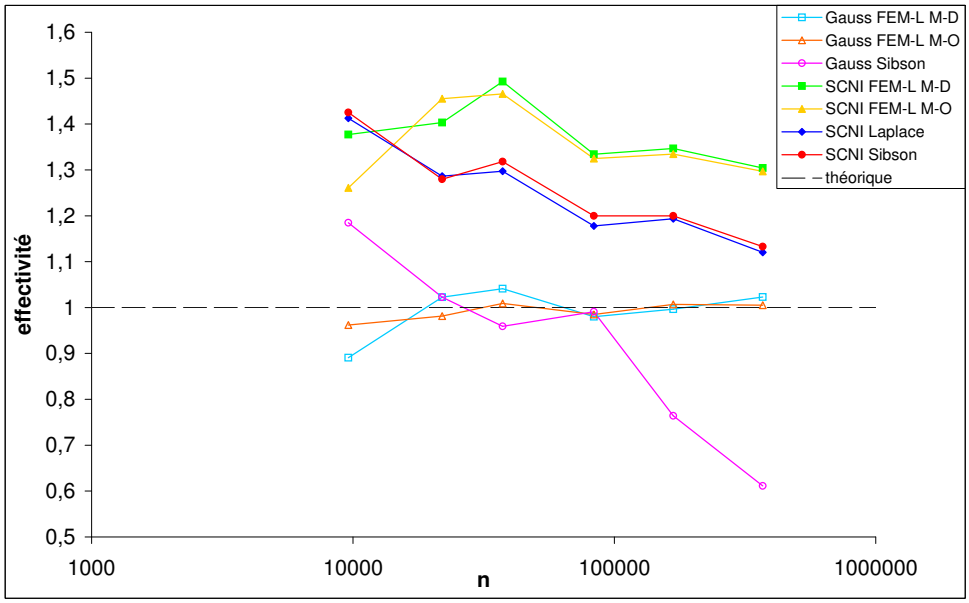

FIG. 4.11 – Courbes d'éffectivité - *échelle logarithmique pour le nombre de nœuds*

4.2 Commentaires sur les résultats obtenus

Bien qu'ayant le même taux de convergence que les éléments finis linéaires, pour les deux exemples étudiés, la CNEM divise l'erreur globale par 3. Ainsi pour atteindre une erreur globale fixée cela revient, en $3d$, à avoir 30 fois plus de nœuds pour une approche éléments finis linéaires que pour la CNEM. Ce gain de précision est principalement dû à l'utilisation du gradient stabilisé qui, s'il est utilisé sur approximation éléments finis, donne des résultats alors comparables pour l'interpolation éléments naturels et éléments finis. On peut remarquer que l'erreur obtenue avec la CNEM avec une intégration de Gauss sur les tétraèdres de Delaunay, augmente avec l'augmentation du nombre de nœuds, ce qui peut être expliqué par une sous intégration des gradients de fonctions de formes de type Sibson.

4.3 Choix de la méthode de calcul du gradient stabilisé

***Remarque** : l'étude a été menée pour des fonctions de forme élément naturel **Sibson**.*

Comme nous l'avons vu dans le paragraphe 3.4, l'utilisation de la discrétisation basée sur les cellules de Voronoï n'est pas toujours possible. Nous avons donc proposé d'utiliser la discrétisation basée sur les quarts de tétraèdres. Afin de voir si cette dernière n'induit pas d'erreurs supplémentaires, le problème de la sphère creuse sous pression à été traité avec les deux discrétisations. Le domaine étant faiblement non convexe, la discrétisation basée sur les cellules de Voronoï est possible. Sachant que les fonctions de formes Sibson sont rationnelles, le nombre de point d'intégration pris pour le calcul du gradient stabilisé peut influer sur l'erreur de calcul. Afin de s'assurer que l'erreur d'intégration est convergée pour chaque discrétisation, on prend différents nombres de points d'intégrations.

Les simulations suivantes ont été menées :

- Intégration stabilisée, discrétisation basée sur les quarts de tétraèdre, avec :
  - 1 point d'intégration pour 3 faces,
  - 1 point d'intégration par face *(barycentre)*,
  - 1 point de Gauss par triangle,
  - 13 point de Gauss par triangle.
- Intégration stabilisée, discrétisation basée sur les cellules de Voronoï, avec :
  - 1 point d'intégration par face *(barycentre)*,
  - 1 point de Gauss par triangle,
  - 13 point de Gauss par triangle *(26 par quadrangle)*.

***Remarque** : Les triangles sont obtenus en maillant les faces des volumes élémentaires entourant chacun des nœuds. Ces faces sont des quadrangles si l'on prend la discrétisation basée sur les quarts de tétraèdres, et des polygones si l'on prend la discrétisation basée sur les cellules de Voronoï.*

Les résultas obtenus sont résumés sur la figure 4.12. On peut voir sur cette dernière, que la discrétisation basée sur les quarts de tétraèdre n'induit pas plus d'erreur que la

discrétisation basée sur les cellules de Voronoï *(pour un nombre de point d'évaluation de fonction de forme par nœud presque équivalent)*. De plus, l'erreur d'intégration avec seulement un point d'intégration par face est largement suffisante *(identique à celle obtenue avec 26 points par face)*. C'est cette intégration que nous avons choisie pour tous les calculs CNEM.

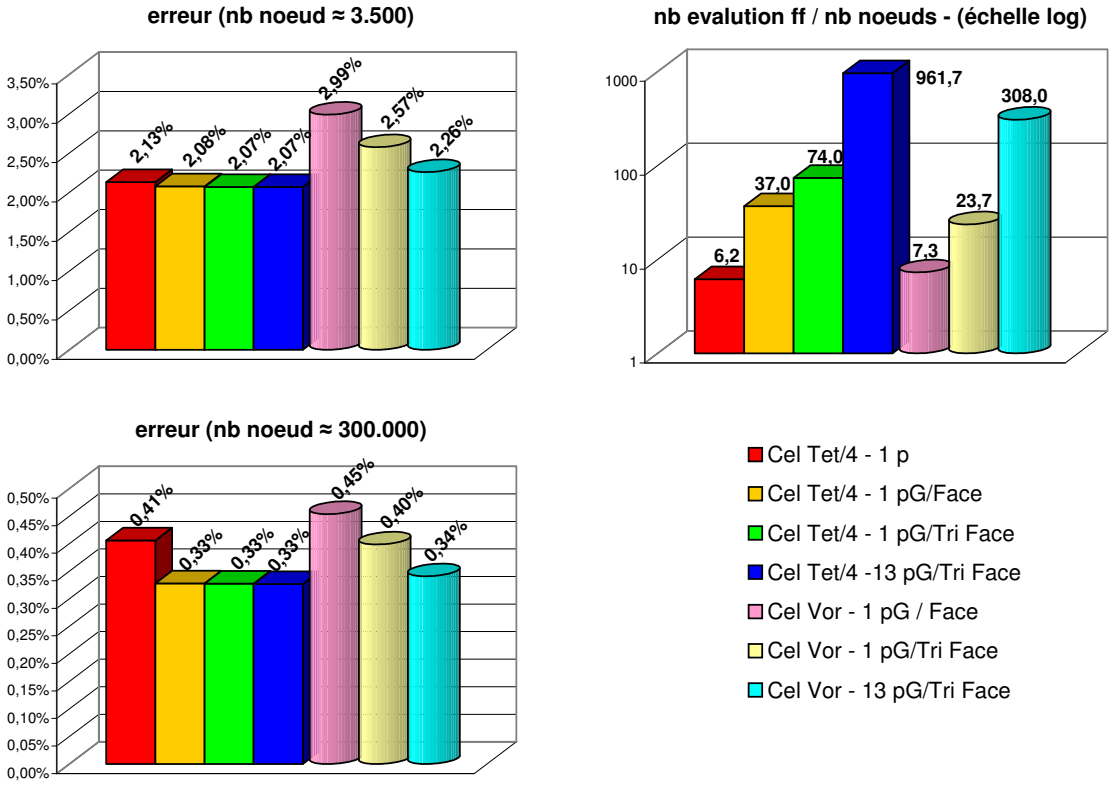

FIG. 4.12 – Erreurs associées aux différentes méthodes de calcul du gradient stabilisé

4.4 Conclusion

Les résultats donnés dans ce chapitre ont permis, d'une part, de positionner la CNEM vis à vis des éléments finis linéaires *(tétraèdres à 4 nœuds)*, et d'autre part de calibrer certains paramètres concernant l'intégration stabilisée *(influence du domaine d'intégration et choix du nombre de points d'intégration)*. Un point marquant est que l'approche CNEM donne des résultats d'une qualité supérieure aux éléments finis linéaires, que le maillage éléments finis soit optimisé ou non. Il faut cependant noter l'importance du gradient stabilisé. Ainsi, si une démarche gradient stabilisé est appliquée à l'interpolation éléments finis, pour un maillage de Delaunay ou un maillage optimisé, les approches CNEM et éléments finis donnent, en élasticité linéaire, des résultats de qualité similaire.

Chapitre 5

Applications de la CNEM aux grandes transformations

Dans cette dernière partie nous abordons l'utilisation de la CNEM dans un contexte de grandes transformations *(grands déplacements et grandes déformations)*. Comme nous le verrons, même si un nombre important de fonctionnalités sont opérationnelles, certains développements sont, encore à ce jour, en cours de réalisation ou à venir. Nous commençons par présenter la démarche mise en œuvre pour les transformations finies : schéma d'intégration temporelle, particularités liées à l'actualisation de la configuration, modèle de comportement *(modèle de type Johnson-Cook)*, contact. Nous nous intéressons ensuite à deux exemples particuliers. Le premier, le test de la barre de Taylor, est destiné à positionner les résultats obtenus avec la CNEM vis à vis de ceux obtenus avec d'autres approches. Nous proposons également, sur cet exemple, une étude sur l'influence de la conduction thermique et de l'adoucissement thermique *(terme d'adoucissement dans la loi de Johnson-Cook)* sur les résultats de simulation. Le second exemple est celui du cisaillage à grande vitesse. Deux aspects sont abordés : étude des premiers instants du cisaillage et des durées caractéristiques des phénomènes rencontrés, puis étude de la dépendance de la localisation des déformations plastiques en fonction du matériau cisaillé *(deux matériaux différents sont comparés)*.

5.1 Mise en œuvre des grandes transformations dans Nessy

5.1.1 Contexte général de la mise en œuvre

L'objet de ce paragraphe est de préciser la démarche que nous avons adoptée pour utiliser la CNEM dans un contexte de grandes transformations. Cette démarche intègre :

- des aspects cinématiques :
  - utilisation d'une formulation lagrangienne actualisée,
  - utilisation d'un schéma d'intégration temporelle explicite,
- un traitement des contacts : contact avec frottement sur surfaces rigides uniquement,
- une intégration de la relation de comportement qui se base sur une décomposition multiplicative du tenseur gradient de la transformation $\mathcal{F}$ en partie élastique et partie plastique.

Ces différents points sont abordés un à un dans les paragraphes qui suivent. La plupart des choix faits sont identiques à ceux présentés dans la thèse d'Yvonnet [Yvo04].

Le modèle de comportement pris est le modèle de Johnson-Cook 5.1. Ce modèle est très utilisé pour les phénomènes à grande vitesse de déformation car il prend en compte de façon simple les principaux phénomènes physiques rencontrés :

- l'écrouissage,
- le durcissement dynamique,
- l'adoucissement thermique.

$$\sigma\left(\varepsilon_{eq}, \dot{\varepsilon}_{eq}, T\right) = \underbrace{\left(A + B.(\varepsilon_{eq})^n\right)}_{\text{écrouissage}} . \underbrace{\left(1 + C.ln\left(\frac{\dot{\varepsilon}_{eq}}{\dot{\varepsilon}_0}\right)\right)}_{\text{durcissement dynamique}} . \underbrace{\left(1 - \left(\frac{T - T_t}{T_f - T_t}\right)^m\right)}_{\text{adoucissement thermique}} \tag{5.1}$$

avec :

- A : la contrainte d'écoulement à déformation plastique nulle,
- B : le paramètre linéaire d'écrouissage,
- n : le paramètre non linéaire d'écrouissage,
- C : le coefficient de sensibilité à la vitesse de déformation,
- $\dot{\varepsilon}_0$: la vitesse de déformation de référence,
- T_t : la température de transition,
- T_f : la température de fusion,
- m : l'exposant d'adoucissement thermique.

5.1.2 Formulation Lagrangienne Actualisée (FLA)

Un des coût principaux de la CNEM, par rapport aux approches de type éléments finis, est lié à la construction de l'interpolation et au calcul des gradients stabilisés.

A cause de grandes déformations il peut y avoir localement de grands déplacements des nœuds les uns vis à vis des autres *(notamment en présence de cisaillement)*. Ainsi, les voisins naturels initiaux d'un nœud peuvent ne pas le rester, d'où la nécessité de réactualiser l'interpolation. Le déplacement d'une particule m à l'instant t, noté $\widetilde{u}(m)_{0\to t}$ est alors défini par :

$$\widetilde{u}(m)_{0\to t} \;=\; \widetilde{u}(m)_{0\to t_{act}} + \widetilde{u}(m)_{t_{act}\to t} \tag{5.2}$$

$$\widetilde{u}(m)_{t_{act}\to t} = \sum_{i=1}^{nb\ voisin} \phi_{i t_{act}}\left(x(m)_{t_{act}}\right).u\left(m_i\right)_{t_{act}\to t} \tag{5.3}$$

avec :

- t_{act} : instant correspondant à la dernière actualisation,
- $x(m)_{t_{act}}$: coordonnées de la particule m à l'instant t_{act},
- $\phi_{i t_{act}}(x)$: fonctions de forme construites sur la configuration à l'instant t_{act} au point de coordonnées x, associée au voisin i,
- $u\left(m_i\right)_{t_{act}\to t}$: déplacement associé à la $i^{ème}$ particule voisine de l'instant t_{act} à l'instant t.

Il en découle que le gradient total de la transformation en un point m est défini par :

$$\mathcal{F}_{0\to t} \;=\; \mathcal{F}_{t_{act}\to t}.\mathcal{F}_{0\to t_{act}} \tag{5.4}$$

5.1.3 Traitement des points additionnels

Comme nous l'avons vu dans le paragraphe 2.2.1, la construction *(ou reconstruction en FLA)* du diagramme de Voronoï contraint peut nécessiter l'ajout de nouveaux nœuds. Ces nouveaux nœuds sont toujours ajoutés sur la surface du domaine, et appartiennent donc à un des triangles décrivant cette frontière.

Ces nouveaux nœuds sont traités comme des nœuds "esclaves" des 3 nœuds de la facette sur laquelle ils se situent. Leur déplacement est défini comme une combinaison linéaires des nœuds "maîtres".

$$u_{esclave} = \sum_{i=1}^{nb\ maître} \lambda_i.u_i \tag{5.5}$$

où u_i est le déplacement de chaque nœud maître, et λ_p le coefficient de pondération associé ($\sum_{i=1}^{nb\ maître} \lambda_i = 1$).

Cette démarche a plusieurs avantages. Elle permet de ne pas ajouter d'inconnues supplémentaires au problème à traiter et surtout évite de diminuer le pas de temps adopté dans le schéma d'intégration temporelle explicite *(schéma conditionnellement stable)*.

Le volume de chaque cellule esclave est réparti, au prorata des λ_i, sur ses nœuds maîtres.

5.1.4 Schéma d'intégration temporelle

Le schéma d'intégration en temps choisi est celui des différences centrées [GR96]. Il s'agit d'un schéma explicite, conditionnellement stable. Il est classiquement présenté sous la forme :

$$\ddot{u}_n = M^{-1}(F_n^{ext} - F_n^{int}) \tag{5.6}$$

$$\dot{u}_{n+\frac{1}{2}} = \dot{u}_{n-\frac{1}{2}} + \frac{\Delta t_n + \Delta t_{n-1}}{2}\ \ddot{u}_n \tag{5.7}$$

$$u_{n+1} = u_n + \Delta t_n\ \dot{u}_{n+\frac{1}{2}} \tag{5.8}$$

avec :
- M : la matrice masse, cette dernière est prise diagonale *(pour chaque nœud il s'agit de la masse de la cellule associée)*,
- F_n^{ext} : colonne des efforts extérieurs généralisés,
- F_n^{int} : colonne des efforts intérieurs généralisés.

5.1.5 Traitement du contact

Actuellement seul le contact avec frottement *(frottement de coulomb)* sur des surfaces rigides mobiles a été implanté. Ce traitement se base sur une approche prédiction correction où les composantes de F_n^{ext} sont calculées de manière à respecter les conditions de contact à t_{n+1}.

5.1.6 Intégration de la relation de comportement

Pour l'intégration de la relation de comportement nous avons suivi la démarche proposée par [Bat95]. Cette démarche s'appuie sur une description de l'état plastique de la matière à partir d'une décomposition multiplicative du tenseur gradient de la transformation totale $\mathcal{F}$. Ce tenseur est décomposé en une partie élastique et une partie plastique :

$$\mathcal{F} = \mathcal{F}^e.\mathcal{F}^p \tag{5.9}$$

Soit la configuration S_{t_n}, configuration à l'instant t_n , pour laquelle on connaît :
- σ_n : le tenseur de contrainte de Cauchy,
- $\mathcal{F}_n$: le gradient de la transformation totale,
- $\mathcal{F}_n^P$: la partie plastique du gradient de la transformation totale,
- p_n : la déformation plastique cumulée.

Et soit la configuration $S_{t_{n+1}}$, la configuration à l'instant $t_{n+1} = t_n + \Delta t$.

L'intégration de la relation de comportement consiste à trouver σ_{n+1}, $\mathcal{F}_{n+1}^P$, et p_{n+1} on se donnant uniquement le gradient de la transformation totale $\mathcal{F}_{n+1}$ à l'instant t_{n+1}.

Le schéma d'intégration en temps permet de calculer en chaque nœud $\mathcal{F}_{n+1}$.

$$\mathcal{F}_{n+1} = \mathcal{I} + \overline{\overline{\nabla}}_{S_0}(u_{n+1}) \tag{5.10}$$

Remarque : dans un contexte de FLA $\mathcal{F}_{n+1} = \left(\mathcal{I} + \overline{\overline{\nabla}}_{S_{t_{act}}}(u_{t_{act}\to t_{n+1}})\right).\mathcal{F}_{0\to t_{act}}$

L'intégration de la relation de comportement débute par une prédiction élastique qui revient à faire l'hypothèse que $\mathcal{F}^p_{n+1} = \mathcal{F}^p_n$.

Prédiction élastique :

Le gradient élastique prédictif ${}^*\mathcal{F}^e_{n+1}$ est défini par :

$$ {}^*\mathcal{F}^e_{n+1} = \mathcal{F}_{n+1}.(\mathcal{F}^p_n)^{-1} = {}^*\mathcal{R}^e.{}^*\mathcal{U}^e \tag{5.11}$$

où ${}^*\mathcal{R}^e$ et ${}^*\mathcal{U}^e$ sont respectivement le tenseur de rotation pure et le tenseur de déformation pure gauche déduits de la décomposition polaire de ${}^*\mathcal{F}^e_{n+1}$.

Le tenseur des contraintes de Cauchy tourné prédictif, ${}^*\hat{\sigma}$, est défini par la relation de comportement élastique écrite dans les deux équations suivantes :

$$ Tr({}^*\hat{\sigma}) = 3\ \kappa\ Tr({}^*\mathcal{E}^e),\ dev({}^*\hat{\sigma}) = 2\ \mu\ dev({}^*\mathcal{E}^e) \tag{5.12}$$

$$ {}^*\hat{\sigma} = J.({}^*\mathcal{R}^e)^T.{}^*\sigma.{}^*\mathcal{R}^e\ ,\ avec\ J = det({}^*\mathcal{F}^e_{n+1}) = det(\mathcal{F}_{n+1}) \tag{5.13}$$

avec :
- ${}^*\mathcal{E}^e = ln({}^*\mathcal{U}^e)$: tenseur des déformations logarithmiques de Henky,
- $3\ \kappa = \frac{E}{1-2\nu}$: module de compression hydrostatique (E module d'Young, ν coefficient de poisson),
- $2\ \mu = \frac{E}{1+\nu}$: module de cisaillement transverse.

La contrainte prédictive équivalente ${}^*\sigma^{eq}$ *(contrainte équivalente de Von Mises)* vaut ainsi :

$$ {}^*\sigma^{eq} = \frac{1}{J}\ {}^*\hat{\sigma}^{eq} = \frac{1}{J}\sqrt{\frac{3}{2}\ dev({}^*\hat{\sigma}) :\ dev({}^*\hat{\sigma})} \tag{5.14}$$

En notant σ^y_n la contrainte seuil en début d'incrément, deux cas sont à envisager :
- ${}^*\sigma^{eq} \leq \sigma^y_n\ \rightarrow$ on reste dans le domaine élastique, la prédiction est donc la solution.
- ${}^*\sigma^{eq} > \sigma^y_n\ \rightarrow$ on est dans le domaine plastique, il faut donc appliquer le correcteur plastique.

Correction plastique :

On suppose que l'écoulement plastique se fait dans la direction du déviateur des contraintes *(loi de normalité)* :

$$ \hat{\mathcal{D}}^p = \dot{p}\ \frac{3}{2}\ \frac{dev(\hat{\sigma})}{\hat{\sigma}^{eq}} \tag{5.15}$$

et donc, par intégration Euler arrière *(implicite)*, que :

$$\Delta\mathcal{E}^p = \Delta p \, \frac{3}{2} \, \frac{dev(\hat{\sigma})}{\hat{\sigma}^{eq}} \tag{5.16}$$

de laquelle est déduit la déformation élastique $\mathcal{E}^e$:

$$\mathcal{E}^e = {}^*\mathcal{E}^e - \Delta\mathcal{E}^p \tag{5.17}$$

A partir de la déformation élastique il est possible de calculer le déviateur du tenseur des contraintes tournées $dev(\hat{\sigma})$:

$$dev(\hat{\sigma}) = 2 \, \mu \, dev(\mathcal{E}^e) = 2 \, \mu \, dev({}^*\mathcal{E}^e) - 2 \, \mu \, \Delta dev(\mathcal{E}^p) \tag{5.18}$$

ou encore :

$$dev(\hat{\sigma}) = dev({}^*\hat{\sigma}) - 3 \, \mu \, \Delta p \, \frac{dev(\hat{\sigma})}{\hat{\sigma}^{eq}} \tag{5.19}$$

Soit finalement :

$$\hat{\sigma}^{eq} = {}^*\hat{\sigma}^{eq} - 3 \, \mu \, \Delta p \tag{5.20}$$

L'écoulement plastique ayant lieu, on a :

$$\hat{\sigma}^{eq} = \sigma^y(p_n, \Delta p, ...) \tag{5.21}$$

En remplaçant l'expression de $\hat{\sigma}^{eq}$ donnée par 5.21 dans 5.20, nous obtenons une équation non linaire dont la résolution permet de déterminer l'incrément de déformation plastique cumulée Δp. Une fois Δp connu, 5.19 nous donne $dev(\hat{\sigma})$.

Vu que l'écoulement plastique est incompressible ($\det \mathcal{F}^p = 1$) on a $Tr(\mathcal{E}^e) = Tr({}^*\mathcal{E}^e)$ et donc :

$$Tr(\hat{\sigma}) = 3 \, \kappa \, Tr(\mathcal{E}^e) = 3 \, \kappa \, Tr({}^*\mathcal{E}^e) \tag{5.22}$$

D'où :

$$\hat{\sigma}_{n+1} = dev(\hat{\sigma}) + \frac{1}{3} Tr(\hat{\sigma}) \tag{5.23}$$

Pour remonter à $\mathcal{F}^p_{n+1}$ il suffit de suivre la procédure :

$$\mathcal{U}^e = exp(\mathcal{E}^e) \quad \rightarrow \quad \mathcal{F}^e_{n+1} = {}^*\mathcal{R}^e.\mathcal{U}^e \quad \rightarrow \quad \mathcal{F}^p_{n+1} = (\mathcal{F}^e_{n+1})^{-1}.\mathcal{F}_{n+1} \tag{5.24}$$

5.2 Applications

5.2.1 Barre de Taylor

Présentation du test

Le test de la barre de Taylor consiste à impacter une barre *(cylindrique)* métallique dotée d'une vitesse initiale v_c sur un plan rigide *(figure5.1)*. Pour nos simulations, l'axe du cylindre ainsi que la vitesse initiale du cylindre, sont pris parallèles à la normale au plan impacté.

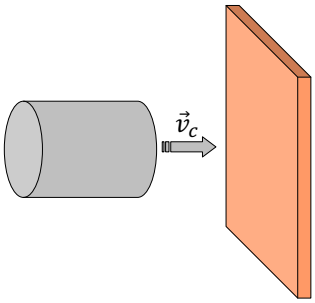

FIG. 5.1 – Barre de Taylor *(Schema)*

Les données, résultats expérimentaux, ainsi que les résultats des simulations éléments finis *(LS-Dyna,avec couplage thermo-mécanique)* de ce test ont été repris de [Ban05].

Les données utilisées sont les suivantes :

- Matériau : Cu-ETP *(Cu-a1)* dont le comportement est modélisé via un modèle de type Johnson-Cook :

A(MPa)	B(MPa)	C	n	m	$\dot{\varepsilon}_{p0}(s^{-1})$	$T_t(^\circ\mathrm{K})$	$T_f(^\circ\mathrm{K})$
90	292	0,025	0,31	1,09	1	294	1356

- Dimensions :
  - longueur initiale : 30 mm
  - diamètre initial : 6 mm
- Vitesse d'impacte : 188 m/s
- Température initiale : 727 $^\circ$K
- Le contact se fait sans frottement et le plan impacté est supposé infiniment rigide.

Sur la figure 5.2 sont représentés les profils du cylindre après impact pour : le résultat expérimental, une simulation éléments finis menée avec LS-Dyna et enfin la simulation menée avec la CNEM. Pour les deux simulations, la conduction thermique à été prise en compte.

Les longueurs et diamètres finaux sont listés dans le tableau ci-dessous.

	expérimental	LS-Dyna	CNEM
langueur finale *(mm)*	16,8	20,2	15,7
$(l - l_{exp})/l_{exp}$	-	20,5%	-6,3%
diamètre final*(mm)*	11,0	12,0	14,8
$(d - d_{exp})/d_{exp}$	-	8,8%	34,7%

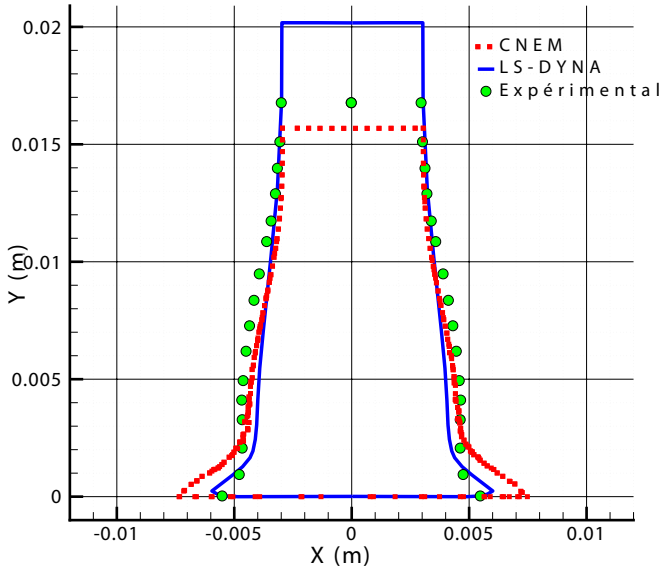

FIG. 5.2 – Comparaison entre les profils obtenus : expérience / FEM/ CNEM

Les figures 5.4 à 5.2.1 retracent différentes étapes de l'écrasement du cylindre. Deux grandes phases sont identifiées. La première au cours de laquelle seule l'extrémité s'évase, la seconde où apparaît un renflement dans la partie centrale. Tout à la fin de la simulation ($t = 1,18.10^{-4}s$) on observe un relâchement des contraintes : ne subsistent que les contraintes résiduelles auxquelles s'ajoutent des allers retours d'ondes élastiques.

Les résultats CNEM et éléments finis semblent encadrer le résultat expérimental (la CNEM donne une meilleure prédiction sur la longueur, les éléments finis une meilleure prédiction sur le diamètre). Il est cependant à noter que ces simulations n'intègrent pas le frottement cylindre/plan qui aura tendance à réduire le diamètre et à augmenter la longueur du cylindre après impact.

Importance des phénomènes thermiques

Afin d'étudier l'influence des phénomènes thermiques sur le résultat obtenu, la même simulation a été menée :

– en adiabatique,

– avec température constante *(pas d'adoucissement thermique)*.

Les profils de cylindre obtenus sont présentés figure 5.3. Les longueurs ainsi que les diamètres finaux sont donnés dans le tableau ci-dessous.

	conduction	adiabatique	T constant
langueur finale*(mm)*	15,7	15,7	16,0
diamètre final*(mm)*	14,8	14,8	13,8
max T(°K)	960	980	718
max p	3,41	3,41	2,93

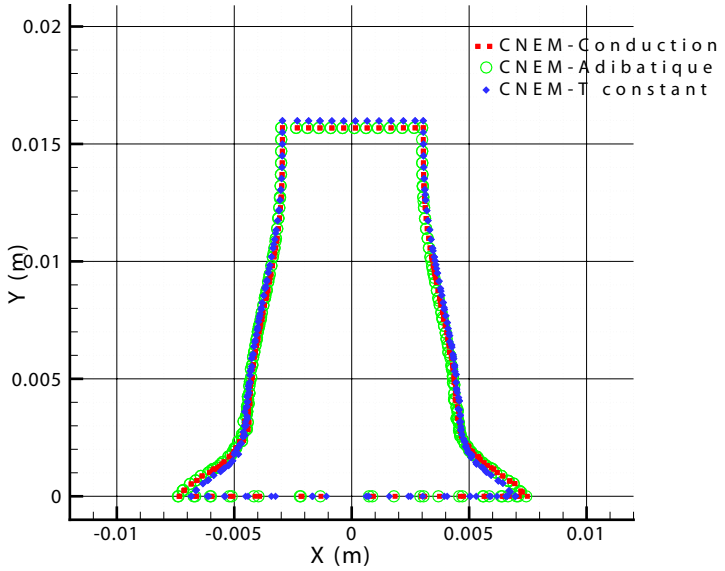

FIG. 5.3 – Comparaison entre profils obtenus : conduction / adiabatique / T constant

Pour ce qui est du profil final du cylindre, peu de différences sont observées pour les 3 simulations. Seule la suppression de l'adoucissement thermique a une très légère influence sur le résultat *(moins d'écrasement)*, cette différence est par contre plus significative pour les températures *(et les déformations plastiques cumulées)* : $718^{\circ}C$ le cas sans adoucissement contre $960^{\circ}C$ ou $980^{\circ}C$ pour les deux autres simulations.

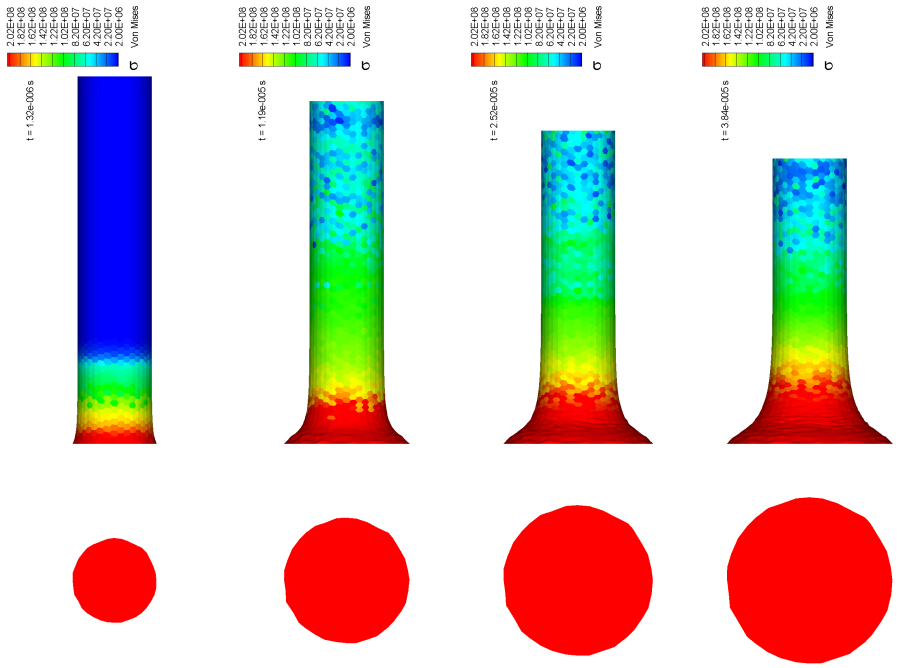

FIG. 5.4 – Barre de Taylor - σ Von-Mises - $0s < t < 3,84.10^{-5}s$

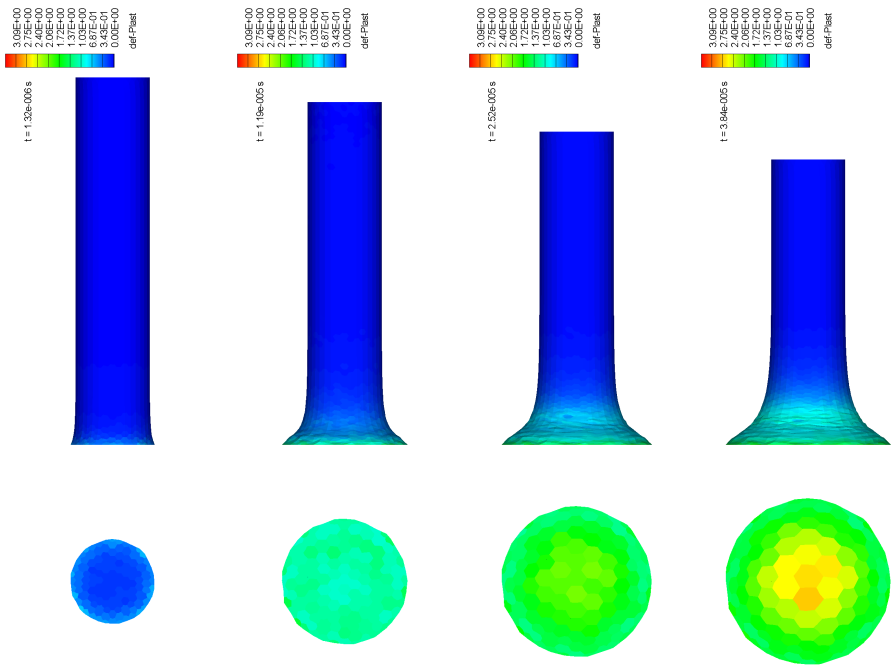

FIG. 5.5 – Barre de Taylor - p - $0s < t < 3,84.10^{-5}s$

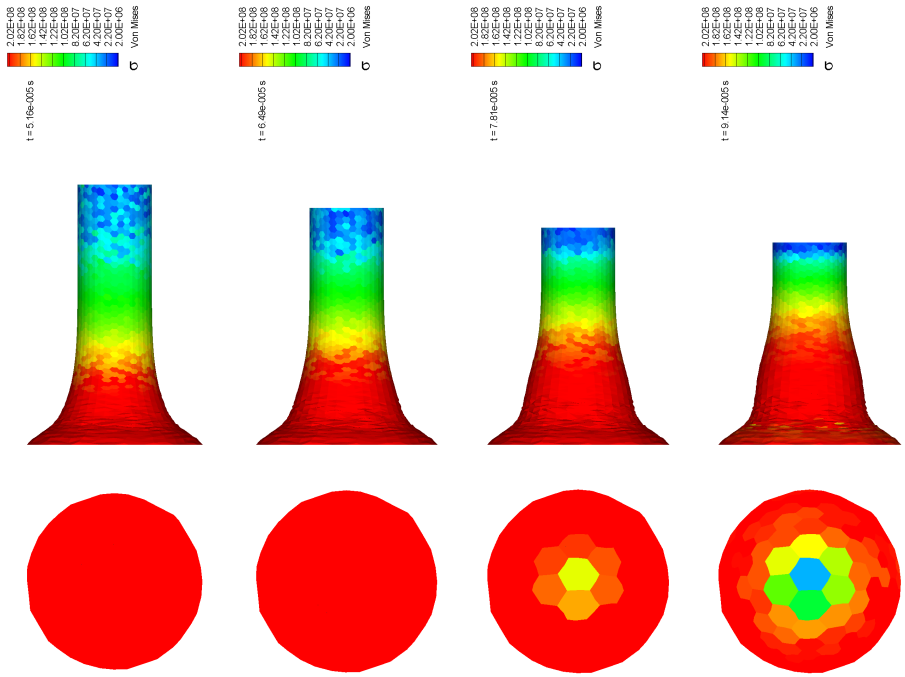

FIG. 5.6 – Barre de Taylor - σ Von-Mises - $3,84.10^{-5}s < t < 9,14.10^{-5}s$

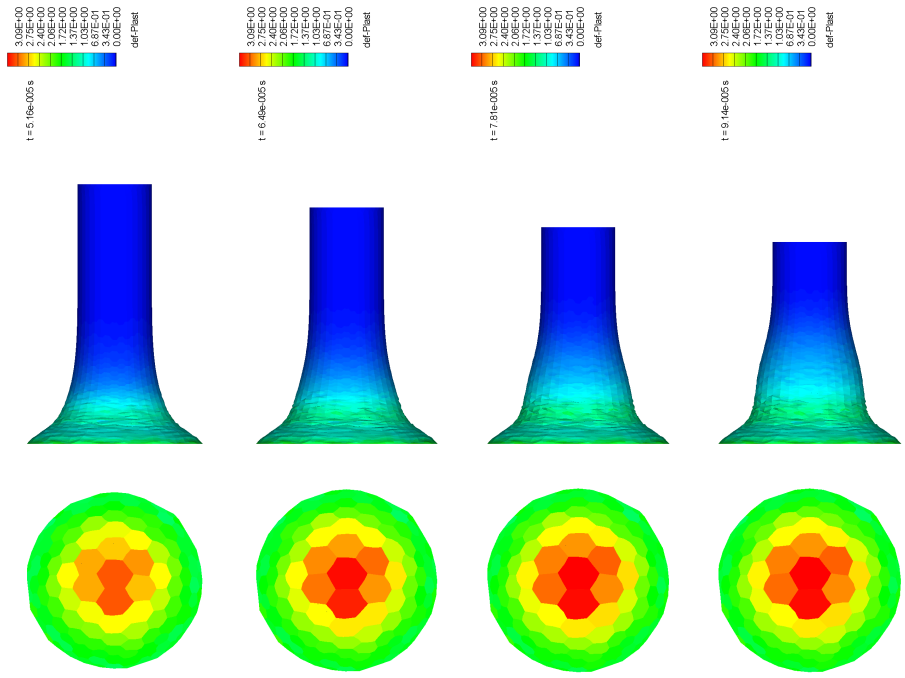

FIG. 5.7 – Barre de Taylor - p - $3,84.10^{-5}s < t < 9,14.10^{-5}s$

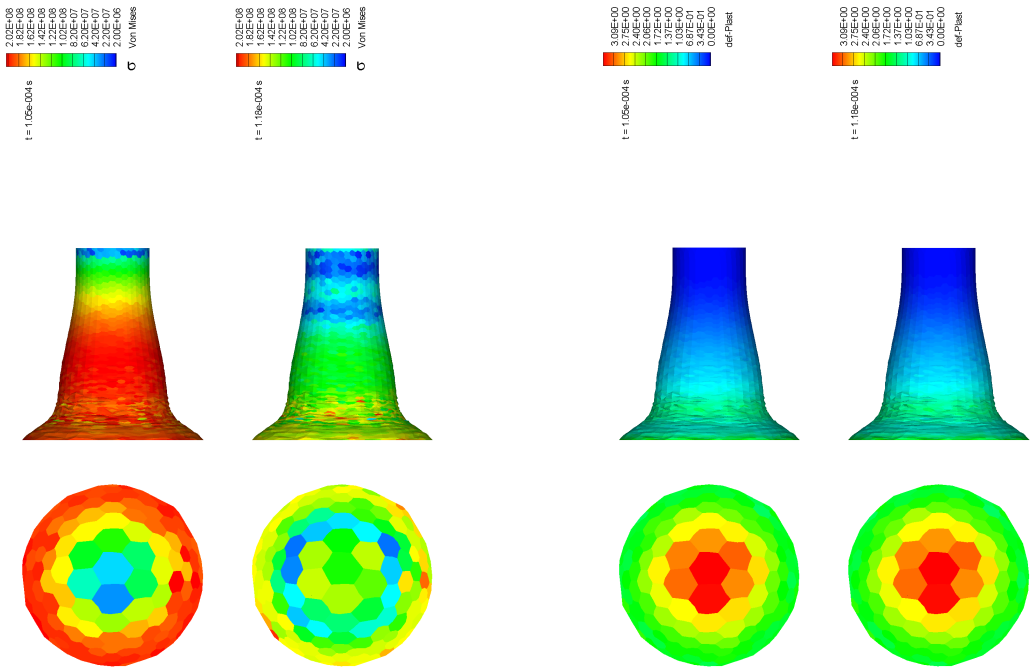

FIG. 5.8 – Barre de Taylor - σ Von-Mises - $9,14.10^{-5} < t < 1,18.10^{-4}$

FIG. 5.9 – Barre de Taylor - p - $9,14.10^{-5} < t < 1,18.10^{-4}$

5.2.2 Cisaillage adiabatique

Une banc d'essais de cisaillage à grande vitesse a été développé au sein du LMSP. Une des finalités de notre travail, et donc de cette mise en œuvre de la CNEM, est de simuler les essais réalisé sur ce banc et de comparer résultats de simulation et résultats expérimentaux. Le dispositif expérimental peut être schématisé comme indiqué sur la figure 5.10. Ce dispositif comprend un poinçon qui, lancé avec une vitesse initiale $\overrightarrow{v}_p$, vient impacter une éprouvette parallélépipèdique simplement posée sur une matrice elle-même montée sur un tube de Hopkinson *(tube non représenté sur la figure)*.

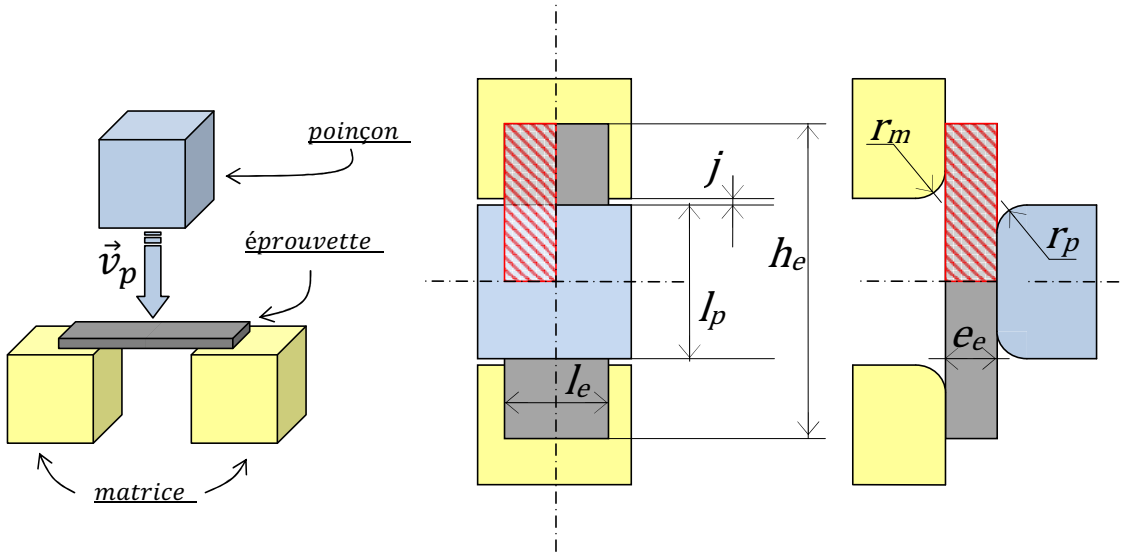

FIG. 5.10 – Cisaillage adiabatique *(Schemas)*

Pour les simulations, le poinçon et la matrice seront considérés rigides, la matrice étant fixe et le poinçon étant supposé opérer à vitesse constante. L'ensemble matrice-poinçon-éprouvette possédant deux plans de symétrie, seul un quart du dispositif est modélisé. La conduction thermique n'a pas été prise en compte. Les contacts poinçon/éprouvette et éprouvette/matrice sont supposés sans frottement. La procédure d'actualisation de la configuration n'a pas été appliquée sur cet exemple.

Nous présentons ci-dessous des résultats pour deux matériaux différents : un acier inoxydable, le 304L, et un alliage de titane, le TA6V. Dans les deux cas un modèle de comportement *(seuil)* de Johnson-Cook est utilisé. Les paramètres de ces modèles sont les suivants :

Matériau-a : 304L *(d'après [Yvo04])*

A(MPa)	B(MPa)	C	n	m	$\dot{\varepsilon}_{p0}(s^{-1})$	$T_t(^\circ\text{K})$	$T_f(^\circ\text{K})$
253,32	685,1	0,097	0,3128	2,044	1	296	1698

Matériau-b : TA6V *(d'après [Ran04])*

A(MPa)	B(MPa)	C	n	m	$\dot{\varepsilon}_{p0}(s^{-1})$	$T_t(^\circ C)$	$T_f(^\circ C)$
866	318	0,008	0,25	1,055	$5{,}77.10^{-2}$	20	1670

Les simulations de cisaillage présentées ont en commun les données suivantes :
- Jeu poinçon matrice j : 0,3 mm
- Largeur poinçon l_p : 11,4 mm
- Rayon d'arrondi poinçon r_p : 0,2 mm
- Rayon d'arrondi matrice r_m : 0,2 mm
- Epaisseur éprouvette e_e : 5 mm
- Largeur éprouvette l_e : 8 mm
- Longueur éprouvette h_e : 24 mm
- Vitesse du poinçon : 10 m/s
- Température initiale : 293 $^\circ$K

Etude du début du cisaillage *(acier 304L)*

Sur les figures 5.11 et 5.12 sont détaillés les premiers instants du cisaillage ($0 \leqslant t \leqslant 2,3.10^{-6}s$). La propagation des ondes élastiques est clairement visible : les ondes traversent l'éprouvette *(épaisseur 5mm)* en environ $1.10^{-6}s$. Un aller retour prend ainsi $2.10^{-6}s$, durée au bout de laquelle on observe déjà très nettement la localisation des bandes de cisaillement. Le niveau de contrainte atteint est tel que la plastification dans cette zone a déjà débutée ce qui montre que le cisaillage a commencé alors même que l'équilibre dynamique est très loin d'être établi.

Il est cependant important de noter que pour cette simulation le poinçon et la matrice sont supposés rigides. Dans la pratique, module d'Young et masse volumique de ces derniers sont voisins de ceux du matériau cisaillé *(inox 304L)*. Le comportement élastique de la matrice et du poinçon ne peuvent ainsi être négligés pour les premiers instants de la découpe. Le fait de les considérer rigides induit une surestimation d'un facteur 2 environ de la vitesse de l'interface poinçon/éprouvette et donc des contraintes présentes dans les ondes élastiques.

La suite de la simulation confirme la localisation observée aux premiers instants *(figures 5.13 et 5.14)*. Dans sa version actuelle, la simulation, suite à l'absence de réadaptation de la frontière du domaine, ne peut dépasser $39.10^{-6}s$. Cette durée correspond à environ 20% de la durée prise pour le cisaillage complet *(d'après une estimation expérimentale)*.

Etude comparée de la localisation du cisaillement - 304L/TA6V

Nous avons également réalisé une simulation pour le TA6V *(figures 5.15, 5.16)*. Ce matériau possède un comportement sensiblement différent du 304L. Comme le montre

ces figures, les simulations permettent de retrouver ce qui est observé expérimentalement à savoir une plus grande localisation de la bande de cisaillement pour le TA6V. Pour le 304L une très forte concentration des contraintes est observée au niveau des congés du poinçon et de la matrice : il y a un facteur 2 sur les contraintes entre ces zones et le centre de la bande de cisaillement. Pour le TA6V la contrainte est presque homogène dans toute la bande de cisaillement.

Il faut cependant garder à l'esprit que ces résultats dépendent fortement de l'identification des paramètres utilisés dans le modèle de comportement. Ils indiquent simplement une tendance.

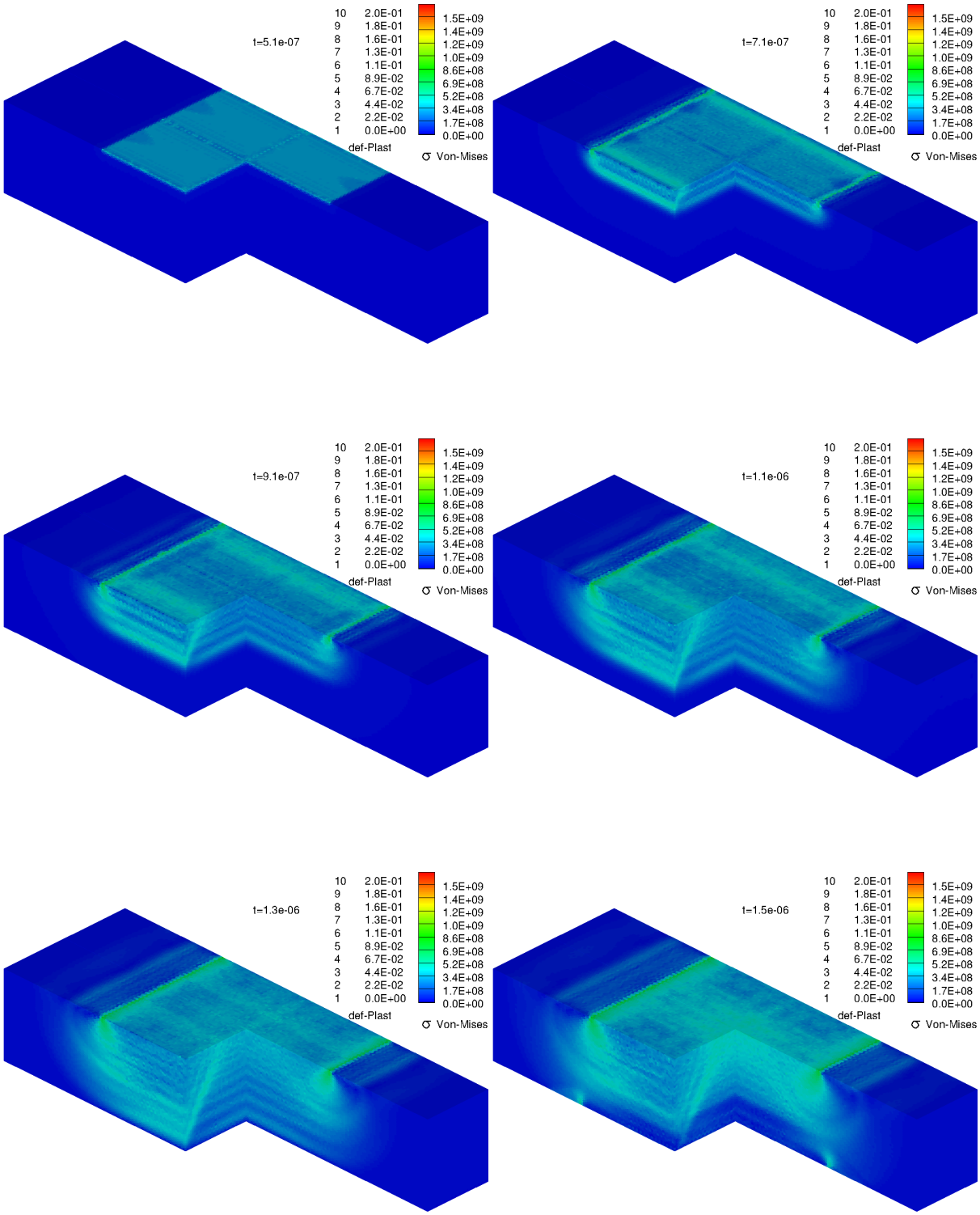

FIG. 5.11 – Cisaillage 304L - $0s < t < 1,5.10^{-6}s$

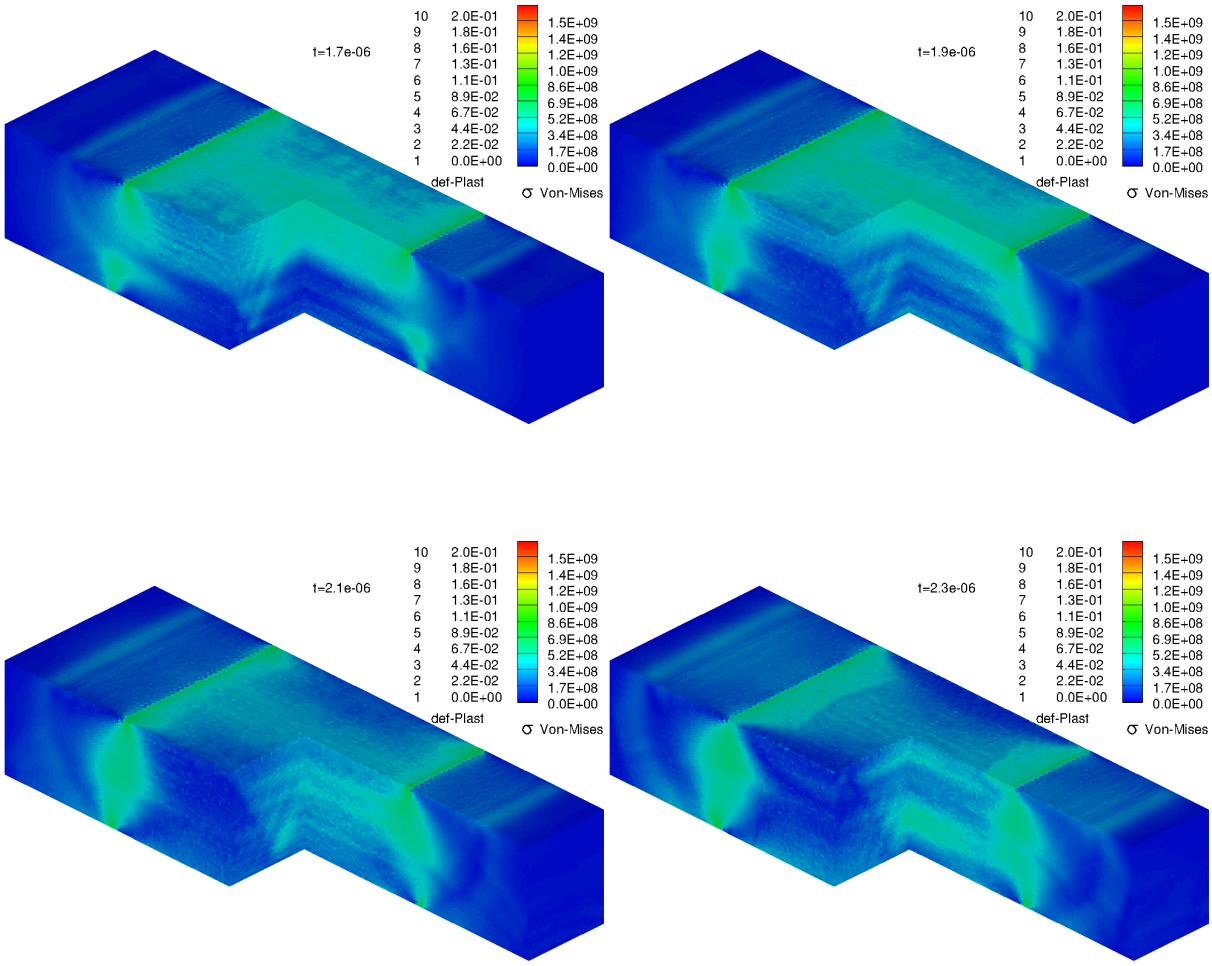

FIG. 5.12 – Cisaillage 304L - $1,5.10^{-6}s < t < 2,3.10^{-6}s$

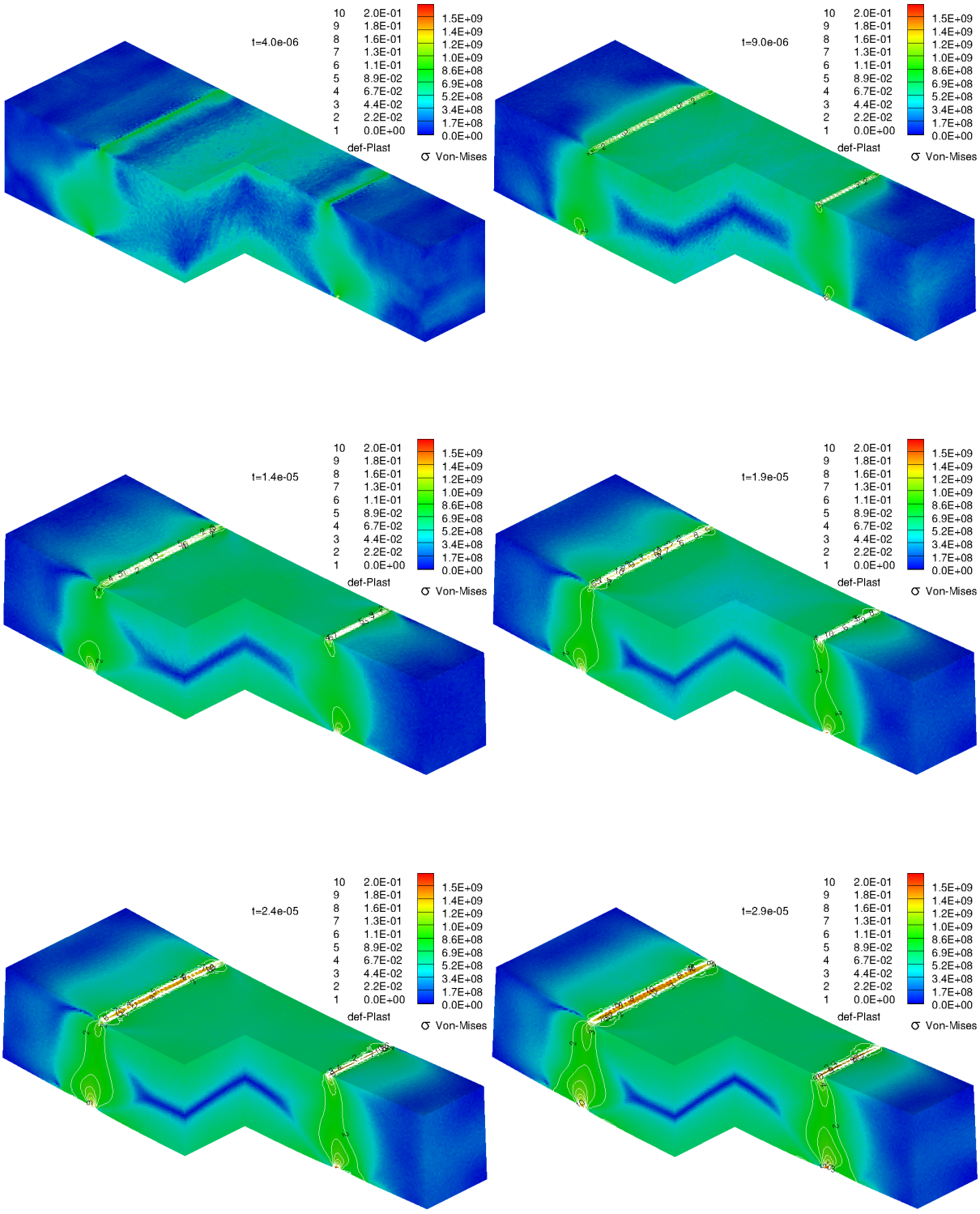

FIG. 5.13 – Cisaillage 304L - $2,3.10^{-6}s < t < 2,9.10^{-5}s$

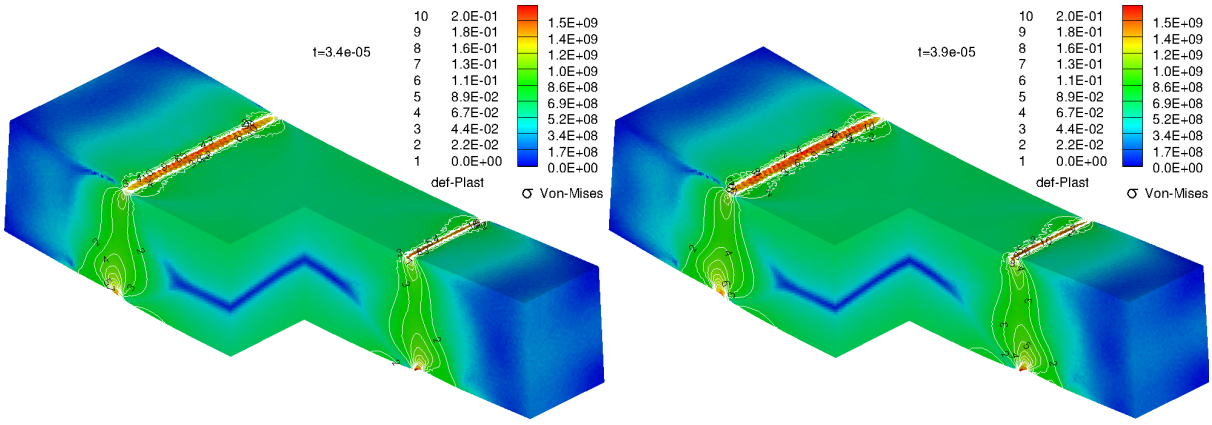

FIG. 5.14 – Cisaillage 304L - $2,9.10^{-5}s < t < 3,9.10^{-5}s$

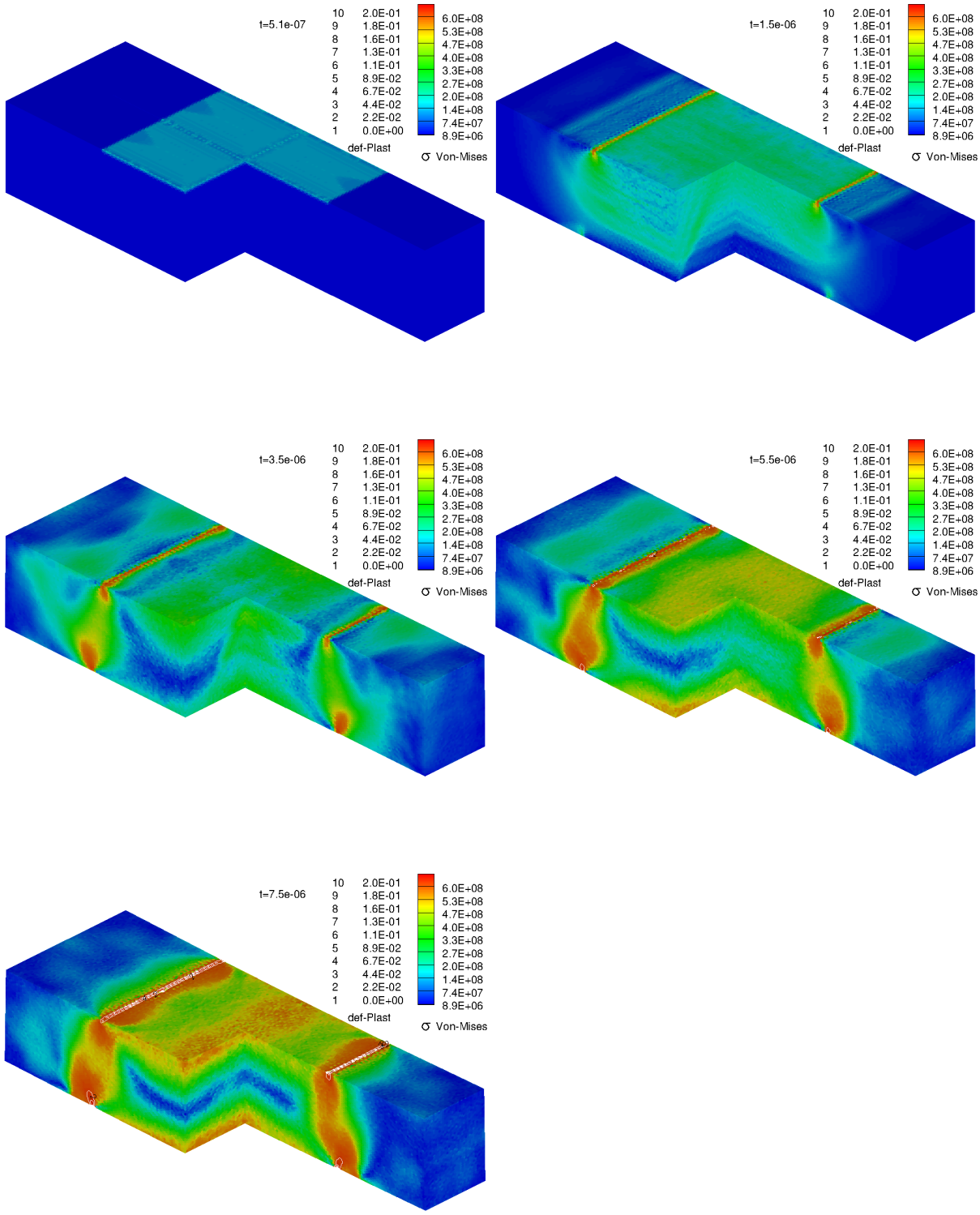

FIG. 5.15 – Cisaillage TA6V - $0s < t < 7,5.10^{-6}s$

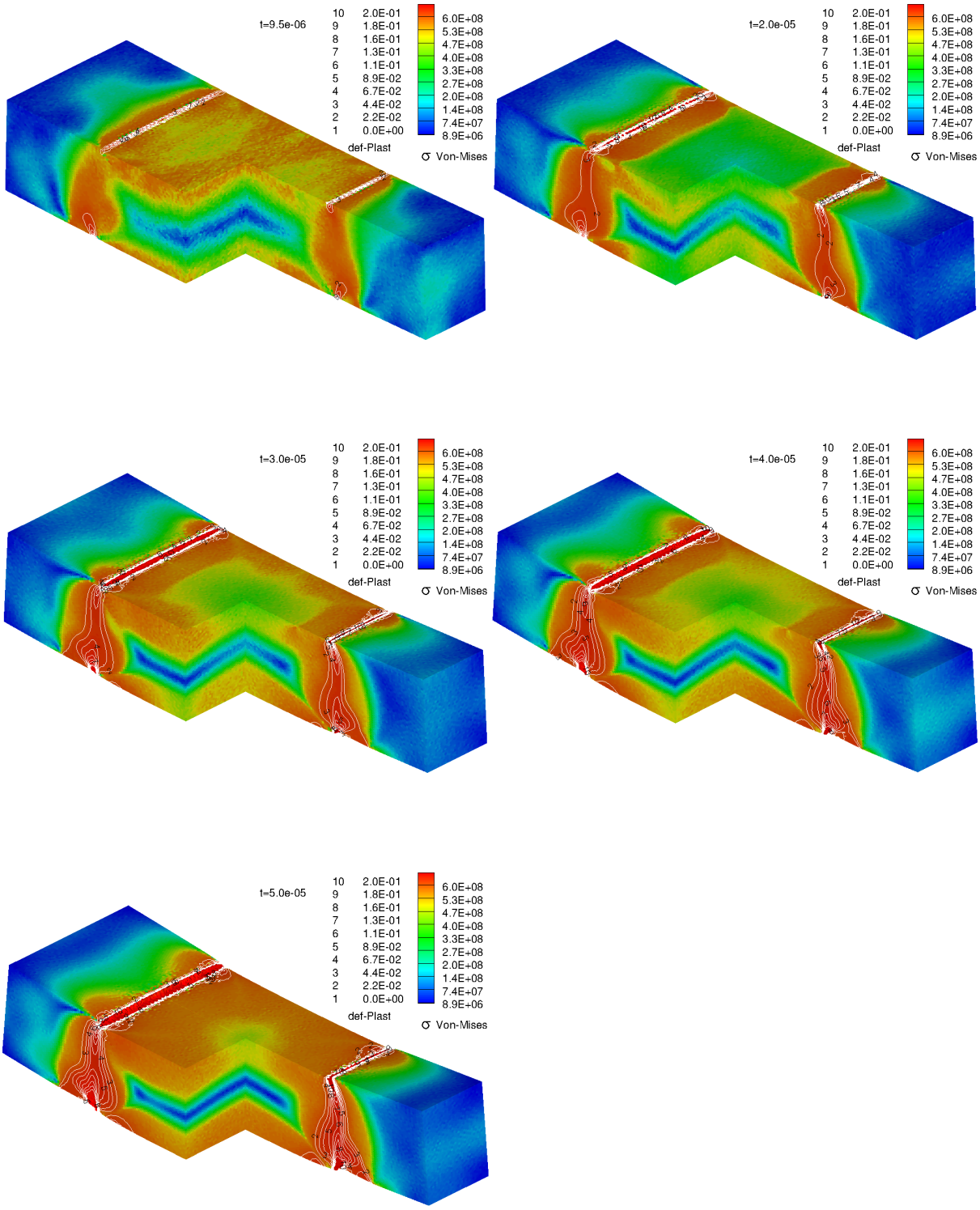

FIG. 5.16 – Cisaillage TA6V - $7,5.10^{-6}s < t < 5.10^{-5}s$

5.3 Conclusion

Les simulations menées sur le test de la barre de Taylor nous ont permis de comparer les résultats obtenus par la CNEM avec ceux obtenus par LS-Dyna. Ces résultats dépendent des algorithmes utilisés dans la CNEM mais également du modèle de comportement choisi et de l'identification de ses paramètres. Ces différents aspects expliquent, sans qu'il soit possible de cerner l'influence de chacun, les différences observées entre résultats expérimentaux et résultats numériques. La principale conclusion est que la CNEM, et l'implémentation que nous en avons faite, permettent d'obtenir des résultats, pour cet exemple, de qualité tout à fait comparable aux approches concurrentes. Enfin, toujours pour la barre de Taylor, et pour une vitesse d'impact de $188m/s$, les simulations montrent qu'il est tout à fait valide de se placer en adiabatique.

Les simulations menées sur l'exemple du cisaillage se limitent actuellement aux premiers instants du cisaillage. Si la localisation des déformations est déjà clairement observée, deux fonctionnalités nous font encore défaut pour aller plus loin dans la simulation : la possibilité d'ajout de nœuds en cours de simulation, et la séparation de la matière. La première fonctionnalité est en cours de finalisation. La seconde se basera, dans un premier temps sur une technique de type "kill-cell" analogue à la démarche "kill-element". Ce type de stratégie étant relativement simple à intégrer nous espérons avoir très prochainement des simulations complètes du cisaillage.

Chapitre 6

Conclusion

Nous pouvons dire qu'à ce jour la majorité des difficultés inhérentes à la mise en œuvre de la CNEM en 3d ont été surmontées : construction du diagramme de Voronoï en $3d$, optimisation des temps de calcul associés à l'évaluation des fonctions de forme Sibson, discrétisation du domaine en volumes élémentaires nodaux pour l'intégration nodale stabilisée *(SCNI)*.

Les tests de validations en élasto-statique ont été menés avec succès. Bien qu'ayant le même taux de convergence que les éléments finis linéaires, la CNEM divise l'erreur globale par 3.

En transformations finies, pour des comportements élasto-viscoplastique, l'approche proposée donne également des résultats très prometteurs. Ainsi sans utiliser une réactualisation de configuration *(formulation lagrangienne totale)*, pour le test de la barre de Taylor, les résultats obtenus sont de qualité au moins équivalente à ceux obtenus avec une approche éléments finis.

Il est cependant nécessaire d'aller plus loin dans le développement de la CNEM et dans les comparaisons avec les approches éléments finis. Pour ce faire, une procédure d'adaptation automatique de l'interpolation en cours de simulation doit être ajouté à l'approche actuelle. Un indicateur basé sur une technique de reconstruction des contraintes de type Zienkiewicz-Zhu [ZBZ99a][ZBZ99b] a été programmé [SAN07] mais doit encore être couplé à une procédure de raffinement. Par ailleurs, la CNEM exige une description correcte de la frontière *(pas d'auto-intersection)* pour la construction du diagramme de Voronoï contraint. Ceci implique une éventuelle mise à jour de cette dernière : remaillage surfacique dans les zones où des repliements, des allongements ou rapprochements excessifs sont observés. L'ajout de nœuds permettra d'exploiter complètement la formulation lagrangienne actualisée pour aller plus loin dans les simulations telles que le cisaillage par exemple.

Pour ce qui est de l'application de la CNEM au cisaillage à grande vitesse, il faut encore, pour réaliser des simulations plus représentatives, tenir compte des effets dynamiques au sein des outillages *(matrice et poinçon)*. A titre d'exemple, sur le banc de cisaillage à grande vitesse du laboratoire, le cisaillage d'un acier S600 de $2mm$

d'épaisseur dure environ $200\mu s$. Un aller retour des ondes élastiques dans le poinçon prend environ $80\mu s$, ce qui modifie l'interaction poinçon/éprouvette en cours de cisaillage. Toujours pour la simulation du cisaillage, il est également nécessaire de pouvoir prendre en compte une séparation de la matière. A cette fin une technique de type "kill element" va dans un premier temps être mise en œuvre.

Bibliographie

[AYCC06] I. Alfaro, J. Yvonnet, F. Chinesta, and E. Cueto. A study on the performance of natural neighbour-based galerkin methods. *Int. J. Numer. Meth. Engng*, 00 :1–28, 2006.

[Ban05] Biswajit Banerjee. Taylor impact tests : Detailed report. Technical report, Department of Mechanical Engineering, University of Utah, Salt Lake City, UT 84112, USA, November 2005. Report No. C-SAFE-CD-IR-05-001. www.csafe.utah.edu/documents/C-SAFE-CD-IR-05-001.pdf.

[Bat95] Klaus-Jurgen Bathe. *Finite Element Procedures*. Prentice Hall, 1995.

[BC02] J-D. Boissonnat and F. Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry*, 22 :185–203, 2002.

[BEF00] B. Büeler, A. Enge, and K. Fukuda. Exact volume computation for polytopes : A practical study. *Polytopes - Combinatorics and Computation*, 29 :131–154, 2000.

[Ber94] Etienne Bertin. *DIAGRAMMES DE VORONOI 2D ET 3D : APPLICATIONS EN ANALYSE D'IMAGES*. PhD thesis, UNIVERSITE Joseph FOURIER - GRENOBLE 1, TIMC - Institut IMAG, Janvier 1994.

[BKO+96] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods : An overview and recent developments. *Computer methods in applied mechanics and engineering*, 139 :3–47, 1996.

[Bow81] A. Bowyer. Computing dirichlet tessellations. *THE COMPUTER JOURNAL*, 24(2) :162–166, 1981.

[BS00] Vitali V. Belikov and Andrei Yu. Semenov. Non-sibsonian interpolation on arbitrary system of points in euclidean space and adaptive isolines generation. *Applied Numerical Mathematics*, 32 :371–387, 2000.

[CWYY01] Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, and Yangu You. Stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50 :435–466, 2001.

[CYW02] Jiun-Shyan Chen, Sangpil Yoon, and Cheng-Tang Wu. Non-linear version of stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53 :2587–2615, 2002.

[DFW] (dave@gumforest.com) David F. Watson. Compound signed decomposition, the core of natural neighbour interpolation in n-dimensional space.(unpublished manuscript). ©2001 D.F.Watson. http://www.iamg.org/naturalneighbour.html.

[E.28] Schönhardt E. Über die zerlegung von dreieckspolyedern in tetraeder. *Mathematische Annalen*, 98 :309–312, 1928.

[EBF03] Andreas Enge(enge@lix.polytechnique.fr), Benno Büeler(bueeler@ifor.math.ethz.ch), and Komei Fukuda(komei.fukuda@ifor.math.ethz.ch). Vinci version 1.0.5, computing volumes of convex polytopes, July 2003. http://www.lix.polytechnique.fr/Labo/Andreas.Enge/Vinci.html.

[GCMD04] D. González, E. Cueto, M. A. Martínez, and M. Doblaré. Numerical integration in natural neighbour galerkin methods. *International Journal for Numerical Methods in Engineering*, 60 :2077–2104, 2004.

[G.M08] Voronoï G.M. Nouvelles applications des paramètres continus à la théorie des formes quadratiques, deuxième mémoire : recherches sur les parallélloèdres primitifs. *J. Reine Angew. Math*, 134 :198–287, 1908.

[GR96] M. Géradin and D. Rixen. *Théorie des vibrations : application à la dynamique des structures*. MASSON, 1996.

[GS78] P. J. Green and R. Sibson. Computing dirichlet tessellation in the plane. *Computer Journal*, 21 :168–173, 1978.

[HS00] Hisamoto Hiyoshi and Kokichi Sugihara. Voronoï-based interpolation with higher continuity. In *Proceedings of the 16th Annual ACM Symposium on Computational Geometry*, pages 242–250, 2000.

[HS06] (si@wias-berlin.de) Hang Si. Tetgen version 1.4, a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, January 2006. ©2002, 2004, 2005, 2006. http://tetgen.berlios.de/.

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

[JC76] J.C. Jaeger and N.G.W. Cook. *Fundamentals of Rock Mechanics*. Chapman and Hall, 1976.

[JW93] Barry Joe and Cao An Wang. Duality of constrained voronoi diagrams and delaunay triangulations. *Algorithmica*, 9 :142–155, 1993.

[Law91] Jim Lawrence. Polytope volume computation. *Mathematics of Computation*, 57 :259–271, 1991.

[LJZ95] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20 :1081–1106, 1995.

[MH01] Philippe Mestat and Pierre Humbert. Référentiel de tests pour la vérification de la programmation des lois de comportement dans les logiciels élément finis. *Bulletin des laboratoires des ponts et chaussées*, 230 :23–38, 2001.

[MTT+96] Gray L. Miller, Dafna Talmor, Shang-Hua Teng, Noel Walkington, and Han Wang. Control volume meshes using sphere packing : Generation, refinement and coarsening. In *5th International Meshing Roundtable*, pages 47–61, 1996.

[N.34] Delaunay Boris N. Sur la sphère vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7 :793–800, 1934.

[OFTB96] D. Organ, M. Fleming, T. Terry, and T. Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics*, 18 :225–235, 1996.

[Pip93] B. Piper. Properties of local coordinates based on dirichlet tessellations. *Computing Suppl.*, 8 :227–239, 1993.

[Ran04] Nicolas Ranc. *Etude des champs de température et de déformation dans les matériaux métalliques sollicités à grande vitesse de déformation.* PhD thesis, Université Paris X, Nanterre, Décembre 2004.

[SAN07] Rafael DALA ROSA DOS SANTOS. Projet d'expertise, cisaillage adiabatique : De la modelisation a la realite. Technical report, ENSAM Paris, 2006-2007.

[SB90] F. Schmitt and H. Borouchaki. Algorithme rapide de maillage de delaunay dans rd. In *journées de géométrie algorithmique*, pages 131–133, 1990.

[SBM95] Malcolm Sambridge, Jean Braun, and Herbert McQueen. Geophysical parametrization and interpolation of irregular data using natural neighbours. *Geophysical Journal International*, 122 :837–857, 1995.

[SBM96] M. Sambridge, J. Braun, and H. McQueen. New computational methods for natural neighbour interpolation in two and three dimensions. In *Computational Techniques and Applications : CTAC95*, pages 685–692, 1996.

[SG05] H. SI and K. GÄRTNER. Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. In *14th International Meshing Roundtable*, pages 147–163, 2005.

[She98] Jonathan Richard Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations. In *Proceedings of the fourteenth annual symposium on Computational geometry*, pages 76–85, 1998.

[Sib80a] R. Sibson. The dirichlet tessellation as an aid in data analysis. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 7, pages 14–20, 1980.

[Sib80b] R. Sibson. A vector identity for the dirichlet tessellation. *Scandinavian Journal of Statistics*, 87 :151–155, 1980.

[Sib81] R. Sibson. A brief description of natural neighbor interpolation. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 21–36. John Wiley and Sons Ltd, 1981.

[Suk98] Natarajan Sukumar. *The Natural Element Methode in Solid Mechanics.* PhD thesis, Northwestern University, Evanston, Illinois, June 1998.

[Wat81] D.F. Watson. Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *Computer Journal*, 24 :167–172, 1981.

[Yvo04] Julien Yvonnet. *Nouvelles approches sans maillage basées sur la méthode des éléments naturels pour la simulation numérique des procédés de mise en forme*. PhD thesis, ENSAM, LMSP, Décembre 2004.

[ZBZ99a] O.C. Zienkiewicz, B. Boroomand, and J.Z. Zhu. Recovery procedures in error estimation and adaptivity. part i : Adaptivity in linear problems. *Computer methods in applied mechanics and engineering*, 176 :111–125, 1999.

[ZBZ99b] O.C. Zienkiewicz, B. Boroomand, and J.Z. Zhu. Recovery procedures in error estimation and adaptivity. part ii : Adaptivity in nonlinear problems of elasto-plasticity behaviour. *Computer methods in applied mechanics and engineering*, 176 :127–146, 1999.

Mise en œuvre de la méthode des éléments naturels contrainte en 3d: Application à la simulation du cisaillage adiabatique

RESUME: Ce travail porte sur la mise en œuvre en 3d de la méthode des éléments naturels contrainte CNEM en vue de son utilisation pour la simulation du cisaillage à grande vitesse. La CNEM est une approche à mi-chemin des approches sans maillage et des éléments finis. La construction de son interpolation utilise le diagramme de Voronoï contraint *(dual du maillage de Delaunay contraint)* associé à un nuage de nœud réparti sur le domaine étudié muni d'une description de sa frontière.
La mise en œuvre de la CNEM comporte trois aspects principaux :i) la construction du diagramme de Voronoï contraint, ii) le calcul des fonctions de forme éléments naturels Sibson, iii) la discrétisation d'une formulation variationnelle générique par utilisation de l'intégration nodale stabilisée conforme, SCNI, introduite par Chen et Al en 2001. Une partie importante de ce travail concerne les deux derniers points. Pour le calcul des fonctions de formes Sibson 3d cinq algorithmes sont présentés, dont deux développés au cours de la thèse, et sont comparés en terme de performance. Par ailleurs, une discrétisation est proposée pour être applicable au cas des domaines fortement non convexes.
La mise en œuvre proposée est validée sur des exemples en élasticité linéaire 3d en petites perturbations *(vis à vis de solutions analytiques et de résultats éléments finis) p*uis en grandes transformations (*test de la barre de Taylor)*. L'application de la CNEM au cisaillage grande vitesse est finalement abordée.
Les développements effectués ont été intégrés à la plateforme logicielle Nessy. Cette plateforme a pour objectif la capitalisation du savoir faire du LMSP en simulation numérique.

Mots-clés: *méthode des éléments naturels contrainte 3d (CNEM), diagramme de Voronoï contraint (DVC), tétraèdrisation de Delaunay contrainte (TDC), fonction de forme Sibson, transformations finie, dynamique explicite, barre de Taylor, cisaillage adiabatique.*

Implementation of the constrained natural element method in 3d: Application to the simulation of adiabatic shearing

ABSTRACT: This work deals with the implementation in 3D of the Constrained Natural Element Method CNEM in order to use it in the simulation of high speed shearing problems. The CNEM is a member of the large family of mesh-free methods, that is at the same time very close to the finite element method. The CNEM's shape function is built using the constraint Voronoï diagram *(the dual of the constraint Delaunay tessellation)* of a set of nodes defining both the whole domain and its boundaries.
The implementation comprises three principal aspects :i) the built of the constrained Voronoï diagram, ii) the Sibson-type CNEM shape functions computation, iii) the discretization of a generic variational formulation by invoking an "stabilized conforming nodal integration", SCNI, introduced by Chen & al. in 2001. 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 in the context of this PhD. For the integration task, a discretization strategy is proposed to handle domains with strong non-convexities.
These approaches are validated on some benchmarks, first in 3D elasticity under the hypothesis of small transformations. The results are compared with the analytical solutions and with approximate finite elements results. Furthermore, a validation in large strain with plasticity effects *(Taylor-bar impact)* is achieved and gives reasonable results. Finally, the 3D CNEM is applied for addressing high speed shearing models.
The developed simulation code (*in Fortran and C++)* is integrated in the LMSP software platform NESSY. NESSY aims the capitalization of the LMSP know-how in numerical simulation.

Keywords: *3d constrained natural elements method (CNEM), constrained Voronoï diagram (CVD), constrained Delaunay tetraedrisation (CDT), Sibson shape function, Finite Transformations , explicit dynamics, Taylor bare impact, adiabatic shearing.*

ARTS ET MÉTIERS ParisTech