Transfert de déformations géométriques lors des couplages de codes de calcul - Application aux dispositifs expérimentaux du réacteur de recherche Jules Horowitz.

Benjamin Duplex

▶ **To cite this version:**

Benjamin Duplex. Transfert de déformations géométriques lors des couplages de codes de calcul - Application aux dispositifs expérimentaux du réacteur de recherche Jules Horowitz.. Modélisation et simulation. Université de la Méditerranée - Aix-Marseille II, 2011. Français. NNT : . tel-00679015

HAL Id: tel-00679015

https://theses.hal.science/tel-00679015

Submitted on 14 Mar 2012

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.

AIX-MARSEILLE UNIVERSITÉ
UNIVERSITÉ DE LA MÉDITERRANÉE

ÉCOLE DOCTORALE 184
MATHÉMATIQUES ET INFORMATIQUE

T H È S E

pour obtenir le titre de

Docteur de l'Université de la Méditerranée

Spécialité : INFORMATIQUE

Présentée et soutenue par

Benjamin DUPLEX

Transfert de déformations géométriques lors des couplages de codes de calcul - Application aux dispositifs expérimentaux du réacteur de recherche Jules Horowitz.

soutenue le 14 décembre 2011 devant le jury composé de :

Président :	Pr. Stefanie HAHMANN	- Institut National Polytechnique de GRENOBLE
Rapporteurs :	Pr. Samir AKKOUCHE	- Université Claude-Bernard, LYON I
	Pr. Houman BOROUCHAKI	- Université de Technologie de TROYES
Directeur :	Pr. Marc DANIEL	- Université AIX-MARSEILLE II
Co-directeur :	Dr. Gilles GESQUIÈRE	- Université AIX-MARSEILLE I
Encadrants :	Dr. Fabien PERDU	- CEA* (GRENOBLE)
	Dr. Marc GRANDOTTO	- CEA* (CADARACHE)
Examinateur :	Dr. Vincent BERGEAUD	- CEA* (SACLAY)

*CEA : Commissariat à l'Energie Atomique et aux énergies alternatives

à la mémoire de Chloé et de Claude,
à mes parents, ...

Remerciements

Je tiens tout d'abord à remercier les personnes qui ont encadré ce doctorat, mon directeur Marc Daniel, mon co-directeur Gilles Gesquière, ainsi que Fabien Perdu et Marc Grandotto, Ingénieurs-Chercheurs au CEA (Commissariat à l'Energie Atomique et aux énergies alternatives), pour les nombreuses discussions, scientifiques ou non, que nous avons pu partager. Merci à vous pour votre aide et vos compétences ainsi que pour la confiance et la liberté que vous m'avez accordées. Je tiens également à remercier Guy Willermoz pour m'avoir proposé ce sujet de thèse.

Je remercie Stefanie Hahmann pour avoir accepté de présider mon jury de thèse. J'adresse mes remerciements aux rapporteurs, Samir Akkouche et Houman Borouchaki, pour avoir lu attentivement mon manuscrit. Merci également à Vincent Bergeaud pour avoir accepté de juger mon travail.

Je remercie Bernard Faydide, chef de service, et Frédéric Ducros, chef de laboratoire, pour m'avoir accueilli au sein du Service de Simulation en Thermo-Hydraulique (SSTH) et du Laboratoire de Développements et d'Applications à l'échelle Locale (LDAL). Merci à toute l'équipe SSTH de Cadarache pour ces trois années passées à vos côtés. Les discussions partagées ont toujours été pour moi très enrichissantes.

Je tiens également à remercier les équipes de développement du code Trio_U, du code Cast3m et de la plateforme SALOME pour leur soutien et leur aide dans la mise en application du cas pratique. Plus particulièrement, merci à Thomas Fortin et Olivier Cioni pour avoir programmé le nouvel opérateur de diffusion dans le code Trio_U, à Erwan Adam et Francis Kloss pour les réponses aux questions sur SALOME et à Gauthier Fauchet pour sa participation. Merci également à Patrick Roux et Christian Gonnier pour les discussions techniques concernant le RJH et ses dispositifs expérimentaux.

Je remercie les membres de l'équipe G-Mod du Laboratoire des Sciences de l'Information et des Systèmes (LSIS) pour m'avoir accueilli chaleureusement à l'ESIL et à l'IUT d'Arles.

Je souhaite également remercier les secrétaires de laboratoire pour m'avoir guidé et assisté dans mes tâches administratives.

Enfin, je citerai les différentes personnes que ce doctorat m'a permis de rencontrer et de cotoyer : Axel (ça pique un peu !), Carole, Chady, Claire (Beamer inside), Clément, Dimitri, Ibrahim, Naibin, les trois Nico :), Olivier (et sa clenche

magique), Raouf, Simon (ça me fait une sorte de fussoir !), Thierry, Valentin, Zhou, ... Leur sympathie, leurs conseils et leur bonne humeur ont indéniablement contribué à l'agréable déroulement de ces trois années enrichissantes.

Mes derniers remerciements sont adressés à ma famille pour leur soutien constant, tout au long de mes études et de mon doctorat.

Sincèrement, merci à tous.

Table des matières

Table des figures ix

Sigles utilisés xiii

Nomenclature xv

1 Cadre de l'étude 1
- 1.1 Simulations numériques ... 3
- 1.2 Problématique de la thèse ... 5
  - 1.2.1 Centralisation des données ... 5
  - 1.2.2 Transfert de déformations géométriques ... 6
- 1.3 Synthèse ... 7

2 Etat de l'art 11
- 2.1 Couplage de codes de calcul ... 13
  - 2.1.1 ISAS ... 14
  - 2.1.2 Travail de doctorat de T. Bonaccorsi ... 14
  - 2.1.3 SALOME ... 15
  - 2.1.4 MpCCI ... 16
  - 2.1.5 ICoCo ... 16
  - 2.1.6 Bilan ... 17
- 2.2 Modèle commun ... 17
  - 2.2.1 Modélisation géométrique ... 17
  - 2.2.2 Modélisation sémantique ... 19
  - 2.2.3 Modélisation déclarative ... 19
  - 2.2.4 Modélisation par entités ... 20
  - 2.2.5 Bilan ... 21
- 2.3 Formats d'échange ... 21
  - 2.3.1 Formats spécifiques ... 21
  - 2.3.2 Formats génériques ... 22
  - 2.3.3 Bilan ... 24
- 2.4 Caractérisation de déformations ... 24
  - 2.4.1 Déformations d'espaces et de sous-espaces ... 25
    - 2.4.1.1 Déformations de formes ... 25
    - 2.4.1.2 Déformations utilisant des coordonnées différentielles 26
    - 2.4.1.3 Fonctions de base radiale ... 26
    - 2.4.1.4 Déformations physiques ... 28
  - 2.4.2 Surfaces déformables ... 29
    - 2.4.2.1 Multirésolution ... 29
    - 2.4.2.2 Déformations physiques ... 30

2.4.3 Méthodes de transfert de déformations 30
2.4.4 Bilan 31
2.5 Méthodes de simplification 32
2.5.1 Fusions de faces 33
2.5.2 Ré-échantillonnage de surfaces polygonales 33
2.5.3 Plongement dans une grille régulière (*vertex clustering*) 34
2.5.4 Décimation 35
2.5.5 Contraction d'arêtes/faces 36
2.5.6 Bilan 39
2.6 Synthèse 40

3 Couplage de codes de calcul et modèle commun 41
3.1 Procédure de couplage de codes 43
3.2 Architecture du modèle commun 45
3.2.1 Intégration de données 45
3.2.2 Définition des caractéristiques 47
3.2.3 Niveau de finesse 48
3.2.4 Définition des entités 49
3.2.4.1 Entité de base 49
3.2.4.2 Entité abstraite 50
3.2.4.3 Entité élémentaire 51
3.2.4.4 Entité complexe 52
3.3 Modèle commun mis en place 56
3.4 Synthèse 56

4 Etudes préliminaires 59
4.1 Protocole d'évaluation mis en place 61
4.2 Interpolation linéaire 62
4.3 Fonctions de base radiale 66
4.3.1 RBF multiquadrique 67
4.3.2 RBF Thin-Plate Spline 71
4.3.3 CS-RBF 73
4.4 Synthèse 76

5 Transfert de déformations 77
5.1 Localisation des déformations géométriques 79
5.2 Définition de nouvelles fonctions 84
5.2.1 Méthode de validation mise en place 84
5.2.2 Fonction à support global F_α 86
5.2.2.1 Présentation 87
5.2.2.2 Etape de simplification et paramètre α 87
5.2.2.3 Etape d'interpolation 90
5.2.2.4 F_α et modèle commun 90
5.2.2.5 Résultats 90

5.2.3 Fonction à support local F_α^σ 93
5.2.3.1 Présentation 94
5.2.3.2 Etape de simplification et paramètre α 94
5.2.3.3 Taille de support et paramètre σ 94
5.2.3.4 F_α^σ et modèle commun 95
5.2.3.5 Résultats 95
5.2.4 Bilan 98
5.3 Application 99
5.3.1 Entre des représentations d'une même géométrie 99
5.3.2 Entre des représentations de géométries adjacentes 107
5.4 Synthèse 111

6 Mise en application **113**
6.1 Cas d'application 115
6.1.1 Réacteur Jules Horowitz 115
6.1.2 Dispositifs expérimentaux 116
6.1.3 Présentation du cas d'application 117
6.2 Couplage réalisé 117
6.2.1 Calcul thermo-mécanique 121
6.2.2 Première étape intermédiaire 123
6.2.3 Calcul thermo-hydraulique 124
6.2.4 Seconde étape intermédiaire 126
6.2.5 Arrêt du couplage 126
6.3 Apport du transfert des déformations géométriques 127
6.4 Résultats obtenus 127
6.4.1 Maillage du fluide 127
6.4.2 Temps de calcul 129
6.4.3 Température du fluide 130
6.4.4 Simulation de l'écoulement du fluide 131
6.5 Synthèse 132

7 Synthèse et perspectives **135**

Annexe

A Calculs complémentaires liés au cas pratique **139**
A.1 Epaisseur moyenne de la lame d'eau 141
A.2 Diamètre hydraulique 142
A.3 Vitesse d'entrée du fluide 142
A.4 Nombre de Reynolds 143
A.5 Valeurs d'énergie cinétique turbulente et de sa dissipation 143
A.6 Puissance thermique 144
A.6.1 Puissance déposée dans les tubes 144
A.6.2 Puissance thermique évacuée par l'eau 144
A.6.3 Bilan possible et prévision des résultats 145

Liste des auteurs **146**

Bibliographie **151**

Table des figures

Chapitre 1

1.1 Différents types de maillages — extrait de [Bon06] 4
1.2 Centralisation des informations 6
1.3 Schéma illustrant le transfert de déformations entre des maillages issus d'une même géométrie 8
1.4 Schéma illustrant le transfert de déformations entre des maillages issus de géométries adjacentes 9

Chapitre 2

2.1 Présentation des composants SALOME — extrait de [Bon06] 15
2.2 Représentation B-Rep d'un cube avec une bosse 18
2.3 Représentation CSG d'un cube troué 18
2.4 Construction d'un *HybridTree* — extrait de [AGCA06] 18
2.5 Exemple de modèle sémantique — extrait de [DLBS*11] 19
2.6 Principe de fonctionnement des FFD — extrait de [GB08] 25
2.7 Déformations s'appuyant sur des cellules rigides — extrait de [BPWG07] (de gauche à droite : le maillage initial, l'ensemble des cellules volumiques déplacées, le maillage modifié) 28
2.8 Principe des approches par multirésolution — extrait de [BS08] 29
2.9 Transfert de déformations entre des objets de formes différentes — extrait de [SP04] 31
2.10 Simplification de maillages par fusions de faces — extrait de [DZ91] . 33
2.11 Simplification par ré-échantillonnage — extrait de [Tur92] 34
2.12 Plongement d'un maillage dans une grille régulière 34
2.13 Décimation d'un nœud 35
2.14 Méthode *"Simplification envelopes"* — extrait de [CVM*96] (à gauche, enveloppe interne et à droite, enveloppe externe) 35
2.15 Contraction d'une arête 36
2.16 Contraction d'une face 36
2.17 Contraction par la méthode *Progressive meshes* — extrait de [Hop96] 37

Chapitre 3

3.1 Méthode de couplage permettant un transfert de déformations entre codes couplés 44
3.2 Entité de base E_BASE 49
3.3 Entité de base E_GROUPE 49
3.4 Exemple d'entité abstraite 50
3.5 Exemple d'entité abstraite héritant de l'entité EA_TUBE 51

3.6 Exemple d'entité élémentaire héritant de EA_CANALISATION 51
3.7 Exemple d'entité élémentaire héritant de l'entité EE_CAN_100_14. 52
3.8 Exemple d'entité élémentaire utilisant une autre entité élémentaire... 53
3.9 Exemple d'entité complexe de groupement 54
3.10 Exemple d'entité complexe correspondant au modèle commun final .. 55

Chapitre 4

4.1 Protocole d'évaluation des résultats des études préliminaires 61
4.2 Maillages M_{1i} et M_{1d} de la phase de définition de la déformation..... 62
4.3 Maillage M_{1d} en fonction de trois plans de coupe..... 62
4.4 Schéma de la méthode d'interpolation linéaire 63
4.5 Résultats de la méthode d'interpolation linéaire 65
4.6 Schéma de la méthode utilisant un réseau de RBF 66
4.7 Influence des paramètres η et β sur la fonction multiquadrique 67
4.8 Résultats de la méthode RBF multiquadrique en fonction du paramètre β ($\eta = 0.0062$ constant, M_{2i} plus fin que M_1 contenant 39 800 nœuds) 69
4.9 Résultats de la méthode RBF multiquadrique en fonction du paramètre η ($\beta = -0.5$ constant, M_{2i} plus fin que M_1 contenant 39 800 nœuds) 70
4.10 Résultats de la méthode utilisant les TPS..... 72
4.11 Résultats de la méthode utilisant la CS-RBF multiquadrique en fonction de la taille du support s (M_{2i} plus fin que M_1 contenant 39 800 nœuds) 75

Chapitre 5

5.1 Maillages utilisés pour illustrer la méthode de localisation des déformations 79
5.2 Maillages initial et déformé localement de la tête..... 81
5.3 Distance euclidienne entre le maillage calculé et M_{1d} 82
5.4 Mise en évidence d'une discontinuité au bord d'une zone modifiée.... 83
5.5 Illustration du maillage utilisé pour la validation des fonctions mises en place..... 85
5.6 Méthode de validation des fonctions définies 85
5.7 Etape de simplification utilisant les QEM 88
5.8 Impact de l'étape de simplification sur la qualité du transfert de déformations (le plan π est utilisé pour montrer le bord du demi-cylindre)..... 89
5.9 Simplification d'un demi-cercle en fonction du paramètre ω 90
5.10 Conditionnement K_2 de la matrice Ψ de F_α pour $n = 30\ 500$ 91
5.11 Erreurs en fonction du paramètre α 92
5.12 Schéma illustrant la taille du support d'une CS-RBF en fonction du paramètre $\sigma = 3$ (ici $\alpha = 6$) 95

5.13 Conditionnement K_2 de la matrice Ψ de F_α^σ pour $n = 30\,500$ 96
5.14 Erreurs en fonction des paramètres α et σ 97
5.15 Méthode proposée pour gérer des géométries identiques 100
5.16 Maillages initial et déformé du *dragon* de Stanford 101
5.17 Maillages initial et déformé *Armadillo* de Stanford 101
5.18 Fonction F_α sur le maillage *dragon* de Stanford 102
5.19 Fonction F_α^σ sur le maillage *dragon* de Stanford 102
5.20 Maillages initial et déformé d'un demi-tube CAO 103
5.21 Fonction F_α sur le maillage *Armadillo* de Stanford 104
5.22 Fonction F_α^σ sur le maillage *Armadillo* de Stanford 104
5.23 Qualité du transfert sur le maillage du demi-tube CAO 106
5.24 Méthode proposée pour gérer des géométries adjacentes 107
5.25 Maillages issus de géométries adjacentes (M_{1i} à droite et M_{2i} à gauche) 108
5.26 Maillages déformés des deux demi-cubes adjacents 109
5.27 Coupe axiale du résultat du transfert d'une déformation entre deux cubes adjacents (vue de dessus) 110

Chapitre 6

6.1 Présentation du Réacteur Jules Horowitz — images extraites du site `www.cea.fr` 116
6.2 Zoom sur le cœur et le réflecteur du RJH — image extraite du site `www.cea.fr` 116
6.3 Schéma du cas d'application (l'unité utilisée est le mètre) 118
6.4 Schéma du cas d'application (coupe vue de dessus) 119
6.5 Schéma du cas test avec et sans prise en compte des déformations 120
6.6 Conditions limites du calcul de comportement thermo-mécanique 122
6.7 Conditions limites du calcul de comportement thermo-hydraulique 124
6.8 Maillage utilisé par le code Trio_U (l'unité utilisée est le mètre) 125
6.9 Vue de dessus d'une coupe du maillage nominal utilisé par Trio_U 128
6.10 Vue de dessus d'une coupe du maillage modifié utilisé par Trio_U 128
6.11 Température du fluide obtenue sans la prise en compte des déformations géométriques 131
6.12 Température du fluide obtenue avec la prise en compte des déformations géométriques 131
6.13 Ecoulement d'une particule fluide dans la géométrie initiale 133
6.14 Ecoulement d'une particule fluide dans la géométrie déformée 134

Chapitre 7

Annexe A

A.1 Schéma représentant la section de passage de la lame d'eau à l'entrée du cas test. 141

Sigles utilisés

Sigle	Définition (français puis anglais)
CEA	**C**ommissariat à l'**E**nergie **A**tomique et aux énergies alternatives French Atomic Energy and Alternative Energies Commission
MTR	réacteur d'irradiation technologique **M**aterial **T**esting **R**eactor
RJH	**R**éacteur **J**ules **H**orowitz Jules Horowitz Reactor
CAO	**C**onception **A**ssistée par **O**rdinateur Computer Aided Engineering
DAO	**D**essin **A**ssisté par **O**rdinateur Computed Aided Design
SI	**S**ystème **I**nternational d'unités International System of units
RBF	fonction de base radiale **R**adial **B**asis **F**unction
TPS	fonction-spline du type "plaque mince" **T**hin-**P**late **S**pline
CS-RBF	fonction de base radiale à support compact **C**ompactly **S**upported **R**adial **B**asis **F**unction

Nomenclature

$v^{(c)}$	composante c du point v.
${}^t x$	transposée du vecteur ou de la matrice x.
ϕ	fonction de base radiale.
F_α	fonction d'interpolation à support global présentée dans la partie 5.2.2.
F_α^σ	fonction d'interpolation à support local présentée dans la partie 5.2.3.
M_{1i}	maillage initial (*i.e.* non modifié) utilisé pour définir une déformation.
M_{1d}	maillage déformé définissant une déformation.
M_{2i}	maillage initial sur lequel sera transférée une déformation.
M_{2c}	maillage calculé, résultat de l'application du transfert de déformation sur le maillage M_{2i}.

Chapitre 1

Cadre de l'étude

Sommaire

1.1 Simulations numériques 3
1.2 Problématique de la thèse 5
1.2.1 Centralisation des données 5
1.2.2 Transfert de déformations géométriques 6
1.3 Synthèse 7

Depuis de nombreuses années, les simulations numériques sont utilisées pour représenter des phénomènes physiques par l'intermédiaire de programmes informatiques appelés codes de calcul. Moins onéreuses et moins contraignantes que les expérimentations, elles permettent d'étudier l'impact de la modification de paramètres et d'obtenir des données qui peuvent être difficilement mesurables en conditions réelles, aussi bien pour des raisons de sûreté que techniques. Il est ainsi possible d'explorer de nombreuses solutions et d'observer des comportements inattendus, pouvant suggérer des expériences. Pour ces raisons, les simulations numériques sont complémentaires aux expérimentations.

Elles ont malgré tout des limites. Tout d'abord, les modèles physiques utilisés pour simuler les phénomènes sont une approximation de la réalité qui a besoin d'être validée pour un domaine d'utilisation. Ensuite, la complexité de certaines études implique des puissances de calcul indisponibles actuellement et l'obligation de limiter spatialement le domaine étudié. Les codes développés s'attachent donc à résoudre un problème déterminé et simplifient les effets extérieurs en utilisant des données imposées (issues d'expérimentations), des moyennes ou des heuristiques. Depuis ces vingt dernières années, des couplages entre codes ont été réalisés afin d'améliorer ce point. Dans ce cas, les codes échangent des résultats obtenus chacun dans leur domaine de compétence. Les calculs réalisés peuvent ainsi prendre en compte les interactions entre différentes disciplines physiques, améliorant les résultats finaux.

Certains codes sont spécialisés dans le comportement mécanique d'objets soumis à des contraintes physiques. Leur résultat permet de rendre compte des évolutions de la géométrie. On s'intéresse dans ce travail de thèse au transfert, vers la représentation géométrique utilisée par un autre code de calcul, de ces déformations qui peuvent être localisées sur une petite zone ou globales à une étude physique. Les données à traiter sont donc potentiellement très nombreuses. Qu'il s'agisse de constantes physiques, de champs de valeurs ou de modifications géométriques, il est important que les codes utilisés puissent échanger des données entre eux. Il est donc nécessaire de définir une entité fédératrice rassemblant l'ensemble des informations.

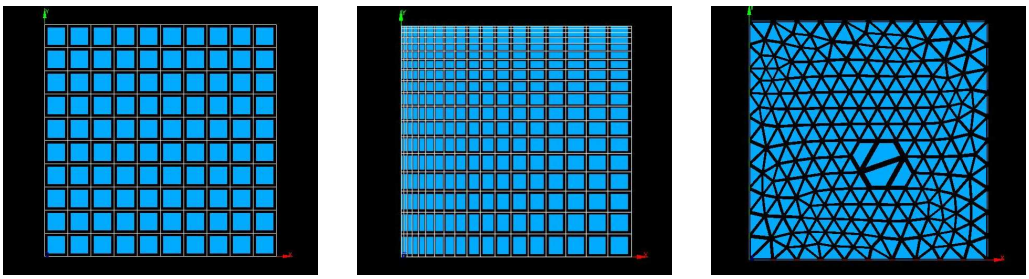
Les simulations numériques sont au cœur des couplages réalisés. Pour cette raison, elles sont présentées dans le paragraphe 1.1. La problématique du travail est détaillée dans le paragraphe 1.2.

1.1 Simulations numériques

Les simulations numériques permettent d'étudier des comportements physiques. Les ressources informatiques étant beaucoup moins coûteuses que les expérimentations, il est possible de réaliser de nombreuses études pour valider des choix techniques ou traiter des cas problématiques, comme l'optimisation, les études de sensibilité et les incertitudes de paramètres physiques. Différents programmes informatiques, nommés codes de calcul, sont développés. Du fait de la complexité et des nombreuses interactions entre les effets physiques, la réalisation d'un code global n'est pas envisageable. Les codes sont spécialisés dans la simulation d'un

phénomène, comme l'écoulement d'un fluide, et d'un type de problème. Ils sont très performants dans un domaine précis. Il faut alors noter que les échelles de grandeur peuvent aller du nanomètre en dynamique moléculaire (simulation de l'évolution d'un système de particules moléculaires) à plusieurs kilomètres dans le cas des études météorologiques. Dans ce travail de thèse, on se limite aux codes s'appuyant sur une représentation géométrique pour effectuer leurs calculs. Un classement possible de ces derniers peut être réalisé pour les séparer en deux groupes. Dans le premier, ils s'appuient sur un maillage de la géométrie de l'étude. La résolution des équations physiques aboutit à un résultat discrétisé sur ce dernier. On parle alors de codes déterministes. Dans le second cas, ils sont dits de Monte-Carlo. Il s'agit d'une approche stochastique utilisant une géométrie continue et faisant intervenir des techniques probabilistes. Elle permet de simuler des phénomènes caractérisés par une succession d'étapes dans lesquelles plusieurs résultats sont possibles pour chaque entité étudiée. Dans ce qui suit, seuls les codes déterministes sont traités. Il faut alors indiquer que, pour une géométrie donnée, la phase de discrétisation peut aboutir à des résultats très différents. Les maillages générés sont spécifiques aux besoins de chacun des codes de calcul. Ils peuvent être surfaciques ou volumiques, uniformes, à base de triangles, de quadrangles, ... La figure 1.1 illustre en deux dimensions trois exemples de maillages d'un carré. Le premier est régulier et structuré, c'est-à-dire qu'il ne contient que des mailles identiques quadrangulaires. Le second est toujours structuré car uniquement composé de rectangles, mais est irrégulier car ses mailles n'ont pas toutes la même taille en raison d'un raffinement en haut et à gauche. Le dernier est irrégulier et non-structuré car constitué de triangles et de quadrangles.

L'intérêt de coupler plusieurs codes est de pouvoir réaliser une étude en s'appuyant sur l'ensemble de leurs compétences. Les résultats obtenus par un code sont ainsi mis à disposition des autres pour qu'ils puissent utiliser des données d'entrée les plus précises possibles. Des échanges de données entre codes sont nécessaires. Des difficultés supplémentaires apparaissent lorsque les échelles de grandeur des codes sont différentes (nanomètre/mètre), que les maillages ne sont pas de même



(a) régulier structuré (b) irrégulier structuré (c) non structuré

Figure 1.1 – Différents types de maillages — extrait de [Bon06]

dimension (surfacique/volumique) et/ou qu'ils sont différents (forme des mailles, nombre et positionnement des nœuds).

Dans ce travail de doctorat, seul le cas du transfert de déformations géométriques est traité. Ces dernières sont calculées en fonction de contraintes thermo-mécaniques par un code dont la spécialité porte sur le comportement mécanique, et utilisées pour mettre à jour le maillage d'un autre code de calcul déterministe.

1.2 Problématique de la thèse

La problématique du transfert de déformations est récurrente dans de nombreux domaines, comme les films d'animation ou les jeux vidéos. Les cas d'applications peuvent être la modification d'un modèle à partir du déplacement de quelques points dans l'espace, comme avec la technique de suivi de mouvement (*motion tracking*) dans [JL08]. Les modèles géométriques manipulés pouvant être très complexes, les déformations sont parfois obtenues sur des supports plus simples issus de géométries différentes. Dans ce cas, le transfert est opéré en réalisant une approximation au mieux sur le modèle initial complet, comme dans [SP04].

Le cas traité dans cette thèse est différent car il s'applique aux simulations numériques. Un programme spécialisé dans le calcul de modifications géométriques doit pouvoir fournir son résultat à d'autres pour qu'ils puissent effectuer leurs calculs sur une géométrie ayant évolué. Pour cela, il est primordial que toutes les tâches s'appuient sur les mêmes informations, qu'il s'agisse de données géométriques ou liées au calcul physique. Il est donc indispensable de les regrouper au sein d'un modèle partagé par toutes les tâches.

Dans ce qui suit, nous allons donc détailler les besoins liés à la mise en œuvre d'un modèle commun centralisant les informations ainsi que ceux portant sur le transfert de déformations. Le travail réalisé n'étant pas restreint aux simulations numériques, les termes de "tâche à accomplir" ou de "traitement à effectuer" sont utilisés plutôt que celui du "code de calcul" dans un soucis de généricité.

1.2.1 Centralisation des données

La centralisation de l'ensemble des informations d'un calcul permet de regrouper plusieurs traitements partageant des données communes (figure 1.2). Elle impose un ensemble de règles pour y accéder et met à disposition les informations calculées par une tâche. Dans le cas présent, les besoins essentiels sont de pouvoir définir une géométrie et d'y attacher des données afin de certifier que tous les traitements s'appuient sur des informations identiques. Dans ce travail de thèse, toutes les données sont connues. Il est donc possible de définir complètement la géométrie qui sert de base aux représentations des tâches à effectuer. Le modèle géométrique permet également d'assurer une cohérence spatiale des résultats obtenus lors des calculs. Pour simplifier sa mise en œuvre, il est important de pouvoir intégrer des paramètres de construction, comme le rayon d'une sphère ou la hauteur d'un cylindre. Il sera

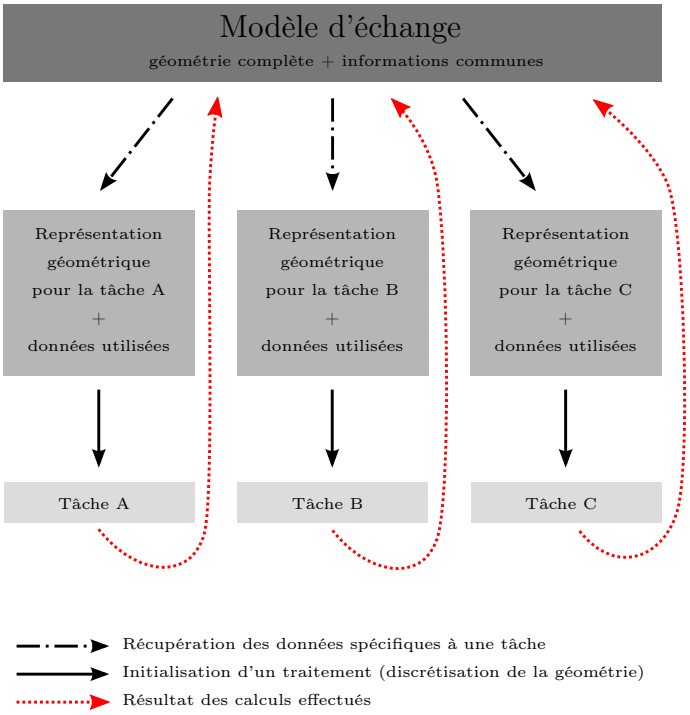


Figure 1.2 – Centralisation des informations

donc nécessaire de la définir de manière paramétrique et continue pour garantir une précision optimale.

Un des intérêts de mettre en commun des informations entre différentes tâches est de s'assurer de la cohérence de leurs résultats. Les données utilisées sont ainsi strictement identiques, aussi bien au niveau de la précision numérique que concernant le choix des valeurs physiques (pour les cas ambigus). Un autre est de pouvoir partager entre tous les traitements les résultats obtenus par une tâche en particulier. Il peut alors s'agir de champs de valeurs, comme une température, ou de déformations géométriques. L'échange de données nécessite donc une phase de transfert vers la représentation géométrique d'un second traitement. Le point particulier du transfert de modifications géométriques est détaillé dans la partie suivante.

1.2.2 Transfert de déformations géométriques

Le but principal du transfert de déformations est de permettre à des traitements d'utiliser une représentation géométrique à jour, c'est-à-dire prenant en compte les résultats obtenus par une tâche dont la spécialité est de calculer les évolutions de la géométrie. Le principe du transfert de modifications géométriques est défini en deux étapes distinctes. La première est appelée la phase de définition. Elle consiste à calculer la déformation de référence qui correspond, dans notre cas, au déplacement de certains nœuds du maillage utilisé. Elle peut être obtenue de différentes manières, par exemple par modification manuelle en utilisant un outil graphique tel Blender

([Blender]) ou par l'intermédiaire d'un logiciel de simulation physique appelé code de calcul. Dans tous les cas, les maillages avant et après déformation sont appariés, c'est-à-dire qu'ils partagent une même connectivité mais que tout ou partie des nœuds peut avoir été déplacé. La seconde phase réalise le transfert des déformations en s'appuyant sur le résultat de la première étape. L'approche envisagée devra faire intervenir des techniques d'interpolation afin de s'appuyer sur les positions exactes du maillage résultat de l'étape précédente. De plus, le transfert ne devra pas créer de nouveaux nœuds ou en supprimer car des données physiques y sont attachées. Seules leurs positions peuvent être mises à jour, c'est pourquoi une approche par remaillage est à proscrire. Il faut également ajouter que le transfert doit être réalisé en un temps relativement faible en comparaison de celui des codes de calcul ; il doit au moins être d'un ordre de grandeur inférieur.

Dans ce qui suit sont présentés les deux cas à traiter nécessitant un transfert de déformations, le premier entre deux maillages issus d'une même géométrie, le second entre deux maillages de géométries adjacentes.

Le premier échange concerne deux traitements réalisant leur calcul en s'appuyant sur une même géométrie. Cette dernière est discrétisée en deux maillages, différents en fonction des besoins de chacune des tâches. Ces représentations géométriques sont supposées quelconques, c'est-à-dire qu'il n'y a aucune contrainte *a priori* sur la position et le nombre des nœuds. Le résultat souhaité du transfert de déformations est détaillé dans la figure 1.3. L'étape de définition de la déformation est réalisée sur le maillage de gauche. Ce dernier est surfacique triangulaire irrégulier avec un raffinement sur la partie modifiée. La seconde phase du processus correspond au transfert de la déformation. Dans l'exemple proposé, il est réalisé sur le maillage de droite qui est volumique hexaédrique régulier.

Dans la seconde approche, les deux traitements utilisent des géométries qui sont adjacentes, c'est-à-dire qu'elles sont différentes mais possèdent des frontières communes. De même que précédemment, les maillages utilisés sont supposés quelconques. Le résultat souhaité du transfert de déformations dans ce cas est détaillé dans la figure 1.4. Dans cet exemple, une géométrie complète regroupe deux objets, notés A pour celui de gauche et B pour celui de droite. Les deux tâches discrétisent la partie de la géométrie complète sur laquelle ils vont effectuer leur calcul. La définition de la déformation est obtenue sur le maillage de l'objet A. Le but du transfert de déformations est de pouvoir modifier le maillage de l'objet B, non pas uniquement son enveloppe surfacique mais l'intégralité du maillage volumique, y compris les nœuds internes.

1.3 Synthèse

Dans ce chapitre, nous avons précisé les besoins de la méthode et défini les objectifs nécessaires à remplir pour y parvenir. Une centralisation des données est impérative pour permettre d'assurer la cohérence des résultats. Le modèle, commun à plusieurs tâches, doit également pouvoir fournir des maillages différents d'une

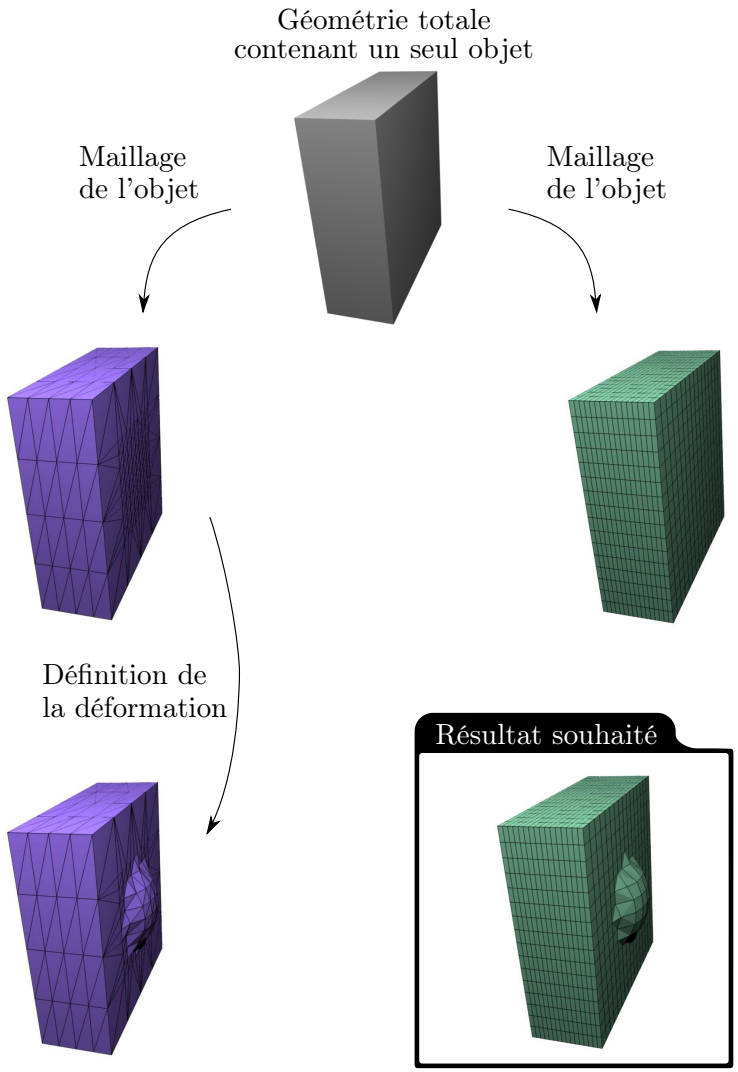


Figure 1.3 – Schéma illustrant le transfert de déformations entre des maillages issus d'une même géométrie

même géométrie. Pour ces raisons, la précision de cette dernière est importante, c'est pourquoi elle est définie de manière continue dans le modèle commun. La prise en compte des modifications géométriques implique la mise en place de méthodes de transfert. Pour cela, deux modes doivent être traités. Le premier, entre deux maillages issus d'une même géométrie et le second, entre des géométries adjacentes. Pour répondre à ce besoin, une étude des méthodes existantes doit être réalisée.

Les approches possibles pour résoudre la problématique de ce travail de doctorat sont présentées dans le chapitre 2. Il en résulte qu'il est nécessaire de mettre en place un modèle commun, proposé dans le chapitre 3, et que différentes méthodes apportent des éléments de réponse face au problème du transfert de déformations. Pour les tester, des études préliminaires sont exposées dans le chapitre 4. Deux nouvelles fonctions sont ensuite mises en place pour répondre au problème. Elles

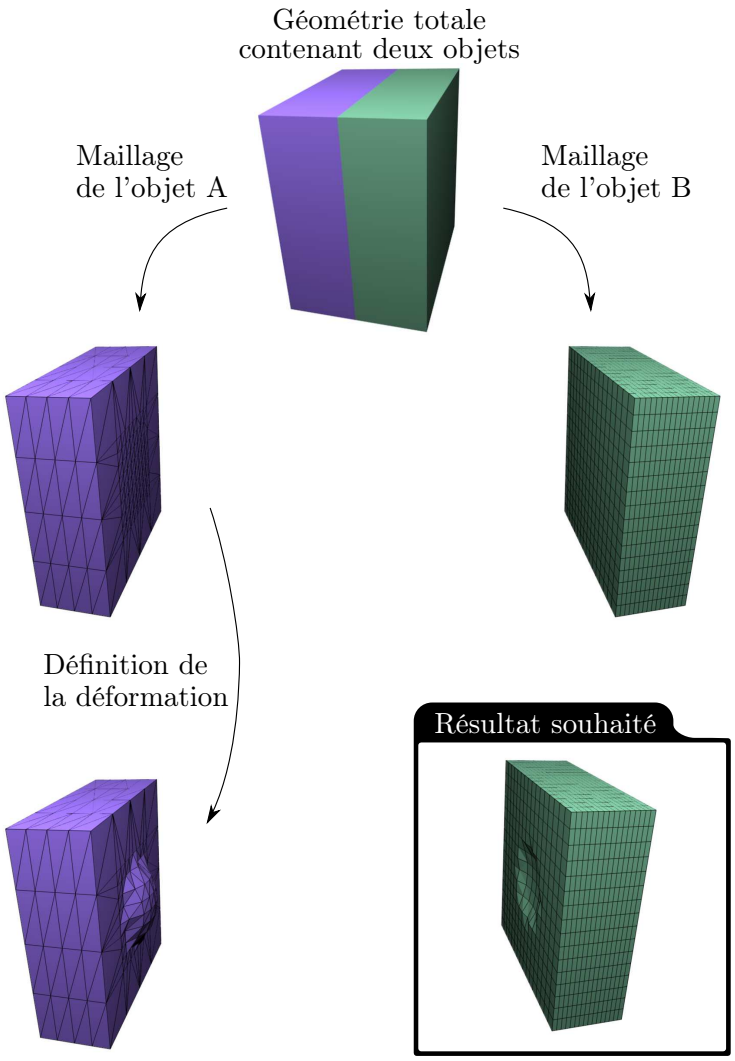


Figure 1.4 – Schéma illustrant le transfert de déformations entre des maillages issus de géométries adjacentes

font l'objet du chapitre 5. Le travail présenté dans ce manuscrit se veut général. Les possibilités d'applications sont nombreuses, comme par exemple la simulation du comportement d'un moteur thermique, l'étude du refroidissement d'un système, ... Le cas choisi est une étude proposée par le CEA qui porte sur les dispositifs expérimentaux du réacteur de recherche Jules Horowitz. Le chapitre 6 présente le traitement de ce cas d'application.

Chapitre 2

Etat de l'art

Sommaire

2.1 Couplage de codes de calcul 13
2.1.1 ISAS 14
2.1.2 Travail de doctorat de T. Bonaccorsi 14
2.1.3 SALOME 15
2.1.4 MpCCI 16
2.1.5 ICoCo 16
2.1.6 Bilan 17
2.2 Modèle commun 17
2.2.1 Modélisation géométrique 17
2.2.2 Modélisation sémantique 19
2.2.3 Modélisation déclarative 19
2.2.4 Modélisation par entités 20
2.2.5 Bilan 21
2.3 Formats d'échange 21
2.3.1 Formats spécifiques 21
2.3.2 Formats génériques 22
2.3.3 Bilan 24
2.4 Caractérisation de déformations 24
2.4.1 Déformations d'espaces et de sous-espaces 25
2.4.1.1 Déformations de formes 25
2.4.1.2 Déformations utilisant des coordonnées différentielles 26
2.4.1.3 Fonctions de base radiale 26
2.4.1.4 Déformations physiques 28
2.4.2 Surfaces déformables 29
2.4.2.1 Multirésolution 29
2.4.2.2 Déformations physiques 30
2.4.3 Méthodes de transfert de déformations 30
2.4.4 Bilan 31
2.5 Méthodes de simplification 32
2.5.1 Fusions de faces 33
2.5.2 Ré-échantillonnage de surfaces polygonales 33

2.5.3 Plongement dans une grille régulière (*vertex clustering*) 34
2.5.4 Décimation 35
2.5.5 Contraction d'arêtes/faces 36
2.5.6 Bilan 39
2.6 Synthèse 40

Les codes de calcul sont très utilisés et couvrent un large spectre de domaines différents : simulation de feux ([SGS11]), étude du vent autour des bâtiments ([TMY*08]), analyse de la dispersion de polluants dans l'atmosphère ([GBSvH11]), ... Il existe des solutions logicielles commerciales, comme Fluent (mécanique des fluides) ou ANSYS Structural (comportement mécanique). D'autres sont libres comme OpenFOAM (dynamique des fluides) ou FDS (simulation de la dynamique du feu). Certaines entreprises, instituts et organismes de recherche ont préféré opter pour un développement interne. Ils peuvent ainsi maîtriser complètement les simulations numériques et mettre en œuvre des codes spécialisés. Par exemple, l'ONERA [1] réalise un code multiphysique pour traiter les problèmes de l'énergétique et de la propulsion appelé CEDRE. Le CEA [2] en développe également, comme Cast3M (`www-cast3m.cea.fr`) pour les études en mécanique des structures ou Trio_U (`www-trio-u.cea.fr`) pour la mécanique des fluides.

Pour réaliser un couplage de plusieurs codes de calcul, des données doivent être échangées. Différentes approches sont possibles, les principales sont détaillées dans le paragraphe 2.1. Ensuite, la problématique du transfert de déformations telle qu'exposée dans le paragraphe 1.2 est abordée par une approche en deux temps.

Premièrement, il est important de centraliser et de partager les informations d'une étude faisant intervenir plusieurs traitements, qu'il s'agisse de données initiales ou de résultats. Il faut donc pouvoir définir un modèle qui serait commun aux différentes tâches à réaliser. Les travaux existants sur ce point sont détaillés dans le paragraphe 2.2 et les principaux formats d'échange sont recensés dans le paragraphe 2.3.

Deuxièmement, la méthode de transfert utilisée doit être à même de gérer tous les types de maillages, que ce soit en fonction de leur dimension, de la forme de leurs mailles ou du nombre de nœuds qui les composent. En plus de son côté générique, elle doit pouvoir être utilisée entre des maillages issus d'une même géométrie ou de géométries adjacentes. Les différentes approches sont détaillées dans le paragraphe 2.4. Elles montrent que les solutions possibles sont confrontées à des problèmes en raison de la complexité des maillages utilisés. Après avoir justifié leur emploi, des techniques de simplification de maillages sont présentées dans le paragraphe 2.5.

2.1 Couplage de codes de calcul

Pour mettre en place un couplage de codes, il est possible d'adopter différentes stratégies. Tout d'abord, les données peuvent être transmises d'un code à l'autre par échange de fichiers. Pour cela, tous les codes doivent être modifiés pour pouvoir intégrer l'ensemble des formats de fichiers qu'ils sont susceptibles de recevoir. Une telle approche est longue à mettre en œuvre et non pérenne dans le temps. Pour éviter cela, le superviseur ISAS, Integrated Safety Analysis System, a été mis en place par le CEA.

1. ONERA : Office National d'Etudes et de Recherches Aérospatiales, `www.onera.fr`
2. CEA : Commissariat à l'Energie Atomique et aux énergies alternatives, `www.cea.fr`

2.1.1 ISAS

En 1995, afin de pallier le problème du dialogue entre les codes de calcul, le CEA a décidé de développer le superviseur ISAS, Integrated Safety Analysis System [3]. Il s'agit d'une bibliothèque de communication programmée en OCAML, dont le noyau est écrit en C, utilisant le langage de commande spécifique CEA Gibiane et s'appuyant sur la librairie PVM (Parallel Virtual Machine) pour gérer les échanges de messages et de données entre les différents codes scientifiques.

Le langage, les formats de fichiers utilisés et les appels possibles sont, très souvent, différents d'un code à l'autre. Ainsi, la mise en place d'un coupleur est une tâche ardue. Son développement nécessite de connaître parfaitement l'ensemble des codes intervenant dans l'étude en cours car ils sont modifiés pour prendre en compte les résultats de ceux avec lesquels ils sont couplés. Les données échangées transitent par le superviseur ISAS, apportant ainsi une aide dans la standardisation de bas-niveau des échanges (types de données échangées). Il faut noter que les codes sont modifiés spécifiquement pour un cas étudié. Remplacer un code par un autre imposera de le connaître parfaitement et de réaliser des ajustements lourds au niveau du coupleur. Plusieurs couplages ont tout de même été réalisés grâce à ISAS ([GNP*97, GGTF98, PC00, BG02]). En raison de contraintes techniques et internes au CEA, ISAS n'est plus maintenu depuis les années 2000 et n'est plus compilable depuis trois ans. Le superviseur a donc définitivement été abandonné.

2.1.2 Travail de doctorat de T. Bonaccorsi

Durant son doctorat, T. Bonaccorsi ([Bon06]) s'est concentré sur la problématique de la modélisation pluridisciplinaire. Il a mis en place un modèle de données commun à plusieurs disciplines physiques et a proposé des outils adaptés au couplage permettant d'automatiser les simulations numériques. Pour cela, il définit un modèle commun s'appuyant sur des atomes de construction appelés *Entités technologiques*. Les jeux de données, ces fichiers requis par les codes pour décrire le problème à traiter et la méthode de résolution à utiliser, sont ensuite encapsulés dans une instance informatique correspondant au problème de l'étude. Elle intègre des fonctions de pilotage du code, comme l'initialisation et le lancement de la résolution, et impose une fonction d'accès aux données appelée *getResults*. Relativement générique, elle est spécialisée dans l'échange de champs de valeurs entre simulations. Pour cela, il est nécessaire de fournir une forme sur laquelle l'interpolation du champ doit être réalisée. L'inconvénient de cette approche est le fait que le travail d'interpolation est réalisé pour chaque code de calcul et que son développement est laissé à la charge des physiciens. De plus, il n'est pas possible de transférer de déformations géométriques entre les codes.

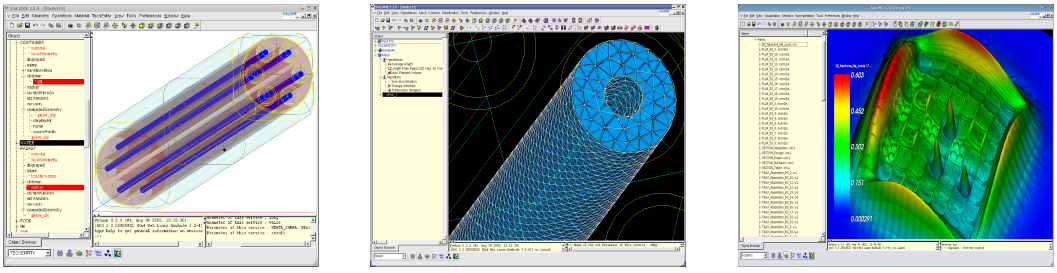
3. Initialement, les premiers couplages de codes sont mis en place dans le cadre de la sûreté nucléaire, ce qui explique les termes de l'acronyme ISAS.

2.1.3 SALOME

La plateforme SALOME, Simulation numérique par Architecture Logicielle en Open source et à Méthodologie d'Evolution (`http://www.salome-platform.org/`), est une solution logicielle développée en partenariat entre EDF [4], le CEA et OpenCascade majoritairement. Il est possible de l'utiliser à partir d'une interface graphique ou par un script écrit en langage python. De cette manière, un utilisateur peut réaliser les tâches courantes graphiquement et automatiser certaines opérations à partir de la console. SALOME fournit un pré- et un post-traitement aux simulations numériques par l'intermédiaire de composants (figure 2.1). Ces derniers sont des briques réalisant un travail donné. Il est possible d'en développer de nouveaux, comme dans [Bon06]. Les composants standards sont au nombre de quatre : GEOM, SMESH, VISU et YACS.

GEOM
Il permet de définir une géométrie de manière continue en combinant des primitives solides et des surfaces. De nombreuses opérations de génération et de transformation sont mises à disposition, comme les extrusions, les révolutions, les translations, les rotations, les explosions (un objet est décomposé en éléments de dimensions inférieures), ... Il fournit les principales tâches d'un outil de CAO et supporte les formats de fichiers standards (IGES, BREP, STEP, UNV, MED, ...).

SMESH
A partir d'une géométrie, cet outil réalise un maillage. Le module s'appuie sur des mailleurs existants comme NetGEN ou GHS3D, qu'il intègre. De cette manière, il est possible de définir des choix sur les maillages (*i.e.* régularité, dimension), sur les mailles (*i.e.* forme triangulaire, critère d'aspect) et sur les arêtes (*i.e.* nombre et répartition des nœuds). Les résultats possibles sont donc très variés. Ce composant permet également de réaliser des remaillages lorsque les frontières du modèle sont fermées. Par exemple, en deux dimensions, il est possible de remailler l'intérieur d'un carré si son contour est fourni.



(a) GEOM (b) SMESH (c) VISU

Figure 2.1 – Présentation des composants SALOME — extrait de [Bon06]

4. EDF : Electricité De France, `www.edf.fr`

VISU
Cet outil de post-traitement permet de visualiser les champs résultats des calculs d'un code. Il est possible d'afficher les données par niveaux de couleurs en fonction de leur valeur, de réaliser des seuillages, de tracer des lignes de courant, ... De plus, la modification et la combinaison de champs sont accessibles par script python et permettent d'en créer de nouveaux. Il est par exemple possible, à partir d'un champ de vitesses, d'en définir un nouveau où les valeurs sont élevées au carré. Ce composant permet également d'obtenir des informations sur la qualité du maillage.

YACS
La plateforme SALOME dispose d'un composant spécialisé dans le couplage de codes de calcul. Initialement nommé SUPERVISION, puis YACS, il a été utilisé lors de plusieurs études ([BCD*08, BL10, CAB*11]). Il permet de définir un enchaînement ou un couplage de codes, ces derniers devant préalablement être encapsulés en tant que composant SALOME pour correspondre au standard de la plateforme. Cette approche est très intéressante lorsque le couplage est bien défini à l'avance et fonctionnel. Elle n'est malheureusement pas applicable dans ce travail de doctorat car le problème du transfert de déformations géométriques n'y est pas traité. De plus, elle ne spécifie pas l'interface de chaque code, c'est-à-dire les différents appels à réaliser pour les piloter.

SALOME se positionne comme un outil d'aide au couplage de codes de calcul. Grâce à ses différents modules, la plateforme est à même de réaliser les étapes en amont (GEOM, SMESH), pendant (YACS) et en aval (VISU) d'un couplage.

2.1.4 MpCCI

MpCCI : Mesh-based parallel Code Coupling Interface (`http://www.mpcci.de`) est une interface permettant de réaliser des couplages de codes ([JK06]). Pour cela, ils doivent respecter une norme définie par l'interface. A l'heure actuelle, un grand nombre de codes commerciaux la supporte et il est possible d'effectuer des échanges de données et des transferts de déformations géométriques. L'interface étant propriétaire et commerciale, il n'est toutefois pas possible de réaliser des développements internes (modification ou ajout). Dans le cadre d'un travail de recherche, la souplesse quant aux méthodes utilisées, existantes ou nouvelles, est primordiale. Pour cette raison, le CEA a préféré développer sa propre interface. Ses codes de calcul ne supportent donc pas MpCCI. Ainsi, le CEA n'est pas dépendant d'une entreprise et peut gérer le support logiciel en interne de manière plus réactive.

2.1.5 ICoCo

L'interface de couplage proposée par le CEA est appelée ICoCo, Interface de Couplage de Codes ([Per06]). Elle décrit un ensemble de méthodes, la manière de les appeler et leur résultat attendu. Ainsi, ICoCo permet à un coupleur de piloter

l'avance en temps des codes qui l'implémente, ainsi que de fournir ou de récupérer des champs. La prise en charge de cette interface par les codes du CEA et d'EDF est en cours de développement. Au CEA, ceux spécialisés en thermo-hydraulique la supporte. Dans [VP10], les auteurs utilisent deux codes de calcul implémentant l'interface ICoCo pour réaliser un couplage. Le coupleur mis en place pilote les codes et modifie les pas de temps en fonction de l'importance de la variation de paramètres. Malheureusement, actuellement, l'interface ne gère pas le transfert de déformations géométriques.

2.1.6 Bilan

Parmi l'ensemble des outils existants, seuls les trois derniers pourraient permettre un couplage laissant la possibilité de réaliser un transfert de déformations géométriques. L'utilisation du module YACS n'offre pas la flexibilité requise dans le cadre d'un doctorat et nécessiterait des développements supplémentaires. MpCCI ne peut être supporté par l'ensemble des codes en raison de son côté propriétaire et commercial. Enfin, l'interface ICoCo est prometteuse mais un effort de développement est nécessaire pour que plus de codes de calcul la supporte et une mise à jour est impérative pour pouvoir répondre au besoin du transfert de déformations. Une nouvelle approche doit donc être mise en place. Elle peut s'appuyer sur les différents outils proposés par la plateforme SALOME. Elle doit intégrer une entité capable de centraliser les informations de chacun des codes. De cette manière, les données sont disponibles pour être utilisées par une autre simulation numérique. Les différentes techniques possibles sont discutées dans ce qui suit.

2.2 Modèle commun

La mise en place d'un modèle commun a pour but de centraliser toutes les informations des traitements. Les tâches réalisées s'appuient sur un maillage de la totalité ou d'une partie de la géométrie complète. Cette dernière doit être définie de manière à ne pas faire intervenir de biais par rapport à la réalité. Sa représentation doit donc être continue. Différentes modélisations sont possibles ([CDH*07]). Il faut rappeler que des données doivent y être intégrées. Il peut s'agir de paramètres géométriques comme une hauteur, ou de données physiques comme un champ de valeurs. Cette partie s'attache à confronter les principaux types de modèles.

2.2.1 Modélisation géométrique

La modélisation géométrique représente les objets à partir de solides et de surfaces ([Gar02]). Dans notre cas, les objets ont une réalité physique. S'ils sont modélisés par des surfaces, ces dernières doivent obligatoirement être fermées [5]. Les principales approches sont les représentations par frontières et par primitives.

5. Une surface fermée sépare l'espace en deux zones distinctes correspondant à l'intérieur et à l'extérieur de l'objet. Pour passer de l'une à l'autre, il faut traverser la surface.

Représentation par frontières

Un modèle par frontières (*B-Rep, Boundary representation*) représente un solide à partir de sa frontière, en connaissant l'intérieur de l'extérieur ([Gar02]). Les frontières peuvent être approchées par des facettes planes (triangles, carrés, ...) et/ou par des surfaces (Bézier, B-spline, NURBS, surface implicite).
La figure 2.2 schématise une représentation d'un cube avec une bosse où les faces sont orientées par leur normale vers l'extérieur (flèches). Le cube est constitué de cinq faces planes et d'une surface paramétrique.

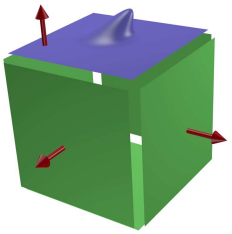


Figure 2.2 – Représentation B-Rep d'un cube avec une bosse

Représentation par primitives

Un modèle par primitives (*CSG, Constructive Solid Geometry*) réunit plusieurs primitives géométriques et les assemble grâce à des opérations élémentaires (union, intersection, différence). Elle s'appuie sur un arbre de construction.
La figure 2.3 représente l'arbre de construction d'un cube troué. Deux primitives, un cube et un cylindre, sont regroupées par une opération de différence.

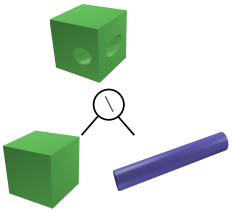


Figure 2.3 – Représentation CSG d'un cube troué

BlobTree

Une extension des CSG a été proposée dans [WGG99]. Nommée *BlobTree*, elle permet d'intégrer des surfaces implicites et ajoute des fonctions de modifications comme la torsion ou la mise à l'échelle. Cette approche a été améliorée dans [AGCA06] pour permettre de prendre en compte les maillages dans un tel arbre. Cette dernière est appelée *HybridTree* et un exemple de construction est illustré sur la figure 2.4.

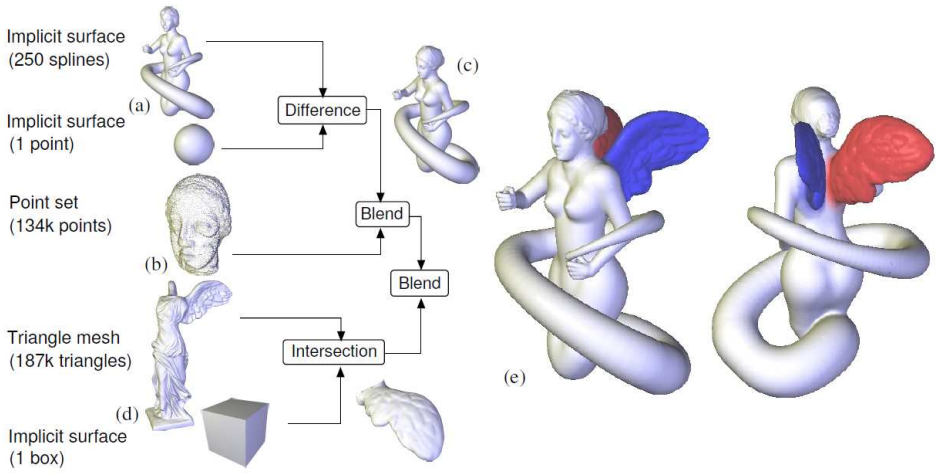


Figure 2.4 – Construction d'un *HybridTree* — extrait de [AGCA06]

Ces modèles géométriques permettent de représenter les objets physiques de manière surfacique ou volumique. Ils répondent bien aux problèmes liés à la géométrie mais ne permettent pas d'intégrer d'informations autres. Pour combler cela, la modélisation sémantique a été mise en place.

2.2.2 Modélisation sémantique

Elle permet d'attacher différentes informations à des objets géométriques pour les définir et pouvoir spécifier des relations entre eux. De cette manière, il est possible d'indiquer qu'un objet géométrique est de couleur rouge, s'appelle "toit" et qu'il est positionné au dessus d'un second objet de couleur blanche nommé "mur". Un exemple d'application à la reconstruction architecturale est disponible dans [DLBS*11]. Dans la figure 2.5, un château est modélisé sémantiquement. Grâce aux données intégrées, il est possible de connaître le nom des différentes parties comme le toit, la porte et le pont. Dans [LPV*10], les auteurs intègrent des conditions limites et des forces physiques sur un maillage avant de lancer des calculs de simulations numériques. La modélisation sémantique répond bien aux problèmes liés à la géométrie mais ne permet pas d'intégrer d'informations qui n'y seraient pas attachées, comme du texte ou des valeurs de constantes physiques.

Figure 2.5 – Exemple de modèle sémantique — extrait de [DLBS*11]

2.2.3 Modélisation déclarative

La modélisation déclarative se positionne à un niveau d'abstraction plus élevé. Définie par Lucas dans [LMMP89], elle permet de générer un objet en s'appuyant uniquement sur un ensemble de propriétés. Elle possède ainsi de nombreux degrés de liberté quant aux géométries qu'il est possible d'engendrer à partir d'un unique modèle déclaratif. Ceci peut être un avantage dans le cas de la conception d'un objet physique par un utilisateur de logiciel CAO. Le concepteur décrit ce qu'il souhaite réaliser au travers d'un ensemble de règles. Il se décharge ainsi de la tâche de génération de l'objet et peut se focaliser sur la partie création ([MD04]). De cette manière, ce type de modélisation offre une aide lors de la phase de conception d'un produit. Parmi ses nombreux domaines d'application, il est possible de citer

la modélisation de scènes pour la synthèse d'images ([Ple91, Ruc01, RP02]), la génération de courbes ([Dan06, DL97]) et surfaces ([LG05]). Dans notre cas, la multiplicité des solutions possibles est un inconvénient car nous ne nous plaçons pas dans la phase de conception d'un produit. Toutes les dimensions sont connues et nous souhaitons disposer d'une géométrie complète unique.

2.2.4 Modélisation par entités

Le principe de la modélisation par entités (*FBM, Feature-Based Modeling*) est de définir un objet physique par un ensemble d'entités, chacune possédant un certain nombre de caractéristiques qui la définissent, comme le matériau utilisé ou la couleur. Il s'agit d'une modélisation de haut-niveau qui permet d'outrepasser les limitations des modélisations géométriques classiques ([Gar03]) car elle consiste à représenter une pièce ou un assemblage [6] au travers de plusieurs entités.

Entités

Une entité (*feature*) peut être définie ([SM95]) comme étant une partie physique d'un objet, étant applicable à une forme générale (*e.g.* il ne s'agit pas d'une forme spécifique), ayant une signification en ingénierie et ayant des propriétés prévisibles. Elle est, en réalité, constituée d'un groupement de plusieurs informations, appelées *caractéristiques*, qu'il est possible de classer en plusieurs groupes ([SR88]) :

- caractéristiques de formes : parties de la géométrie nominale,
- caractéristiques de tolérance : écarts avec les valeurs nominales,
- caractéristiques des matériaux : compositions, traitements, conditions, ...
- caractéristiques technologiques : informations de performance.

Cette liste est loin d'être exhaustive, il est possible d'en ajouter de nouvelles en fonction des besoins. Toutes ces notions peuvent être retrouvées dans [Sha88, Gar03].

En ce qui concerne la représentation de l'entité, c'est-à-dire son modèle géométrique, il peut être de n'importe quel type mais les plus utilisées sont les représentations par frontières et par primitives.

Conception par les entités

Ces entités sont ensuite utilisées pour construire un objet plus complexe. Ainsi, dans [Gar03], Gardan explique qu'il est possible de représenter cet assemblage d'entités par un arbre binaire où chaque nœud est un ajout ou un retrait de matière. Le parallèle avec les arbres de construction du modèle CSG peut être réalisé. Il présente néanmoins une différence : les opérations booléennes (union, intersection) ne sont pas positionnées au niveau des nœuds de l'arbre mais sont intrinsèques aux entités. Si l'on souhaite réaliser un retrait de matière, il sera nécessaire de mettre en place une entité où l'une des caractéristiques sera de signaler qu'il s'agit d'un retrait de

6. Originellement, cette approche a été mise en place dans le cadre de l'usinage de pièces mécaniques, ce qui explique pourquoi on parle de *pièces* et d'*assemblages* au lieu d'objets.

matière et non d'un ajout. L'absence des opérations booléennes dans l'arbre de construction permet de faciliter sa lecture. Il est alors important d'utiliser des noms d'entités explicites permettant de comprendre aisément qu'il s'agit d'un retrait de matière (*i.e.* trou, rainure), l'ajout étant la valeur par défaut.

2.2.5 Bilan

L'intégration d'informations extérieures à la géométrie impose l'utilisation d'un modèle de données de haut niveau et suffisamment riche pour pouvoir gérer des géométries complexes. Une modélisation géométrique répond bien au second point mais n'offre pas la possibilité d'intégrer d'informations génériques, comme les valeurs de constantes physiques (*e.g.* constante de gravité). Le modèle déclaratif supporte l'ajout de données mais ne permet pas de disposer d'une unique géométrie complète. Enfin, la modélisation par entités apparaît être un bon choix. Toutefois, ce modèle ne définit qu'un type de conception et il est nécessaire de mettre en place les caractéristiques et les entités requises pour répondre aux mieux à nos besoins.

2.3 Formats d'échange

Le modèle commun devra pouvoir être échangé entre différentes personnes et différents systèmes. Une étude des formats est donc nécessaire afin d'en choisir un compatible avec la modélisation par entités que nous pourrions mettre en place.

2.3.1 Formats spécifiques

Le problème de définir un langage, commun à plusieurs applications, pour définir un modèle de représentation et pouvoir l'échanger, est récurrent en informatique. Depuis ces trente dernières années, de nombreuses recherches ont été réalisées dans ce sens et ont abouti à l'émergence de plusieurs normes ([For06]). Au début des années 80, le but principal était uniquement de pouvoir échanger des données géométriques CAO/DAO. Les premiers formats sont présentés succinctement dans ce qui suit.

IGES
Tout d'abord, en 1980, le format IGES (Initial Graphics Exchange Specification) est mis au point par plusieurs entreprises dont Boeing, General Electric, Xerox, et le département de la défense américaine. Normalisé par l'ANSI (American National Standard Institute), il est utilisé principalement dans le domaine de la mécanique et de l'aéronautique. Pour modéliser la géométrie, il impose la représentation par frontières. De plus, il n'est pas possible d'intégrer d'informations extérieures, comme des constantes physiques ou des champs de données. Malgré cela, il reste l'un des plus répandus pour la visualisation ou l'export de modèles CAO 3D.

VDAFS
Quelques années plus tard, les constructeurs automobiles allemands mettent au point un format dédié à leurs besoins : VDAFS (Vereinung Deutsche Automobilindustrie Flächen Schnittstelle, ou Format de l'industrie automobile allemande pour les échanges de surfaces). Ce dernier est normalisé par la DIN (Deutsches Institut für Normung, l'institut de normalisation allemand) en 1983 et de nombreux logiciels de CAO/DAO le supportent également. Il est toujours utilisé mais se limite principalement au domaine de l'industrie automobile allemande et ne permet pas d'intégrer de données autres que des représentations géométriques 3D, des commentaires et des matrices de transformations. Ainsi, il n'est pas possible d'ajouter de valeurs physiques.

2.3.2 Formats génériques

Ces deux précédentes normes mettent avant tout l'accent sur l'échange de données, données purement géométriques pour IGES et données liées à l'industrie automobile pour VDAFS. Cela implique qu'elles ciblent des domaines d'application particuliers, ce qui les rend difficilement utilisables dans d'autres cas. Ainsi, de nouveaux formats génériques ont vu le jour.

SET
Un format d'échange nommé SET (Standard d'Echange et de Transfert) a été défini par l'aérospatiale française avec pour but de faire mieux que ces prédécesseurs. Il représente une solution aux exigences relatives à l'échange de données entre différents systèmes de CFAO (Conception et Fabrication Assistées par Ordinateur) et au besoin d'archiver ces données. Il dispose d'un langage neutre partagé par l'ensemble des domaines d'application. Il a été normalisé par l'AFNOR (Association Française de NORmalisation) en 1983. Ce format est standard mais ne permet pas de prendre en compte des données extérieures à la géométrie, comme des champs d'informations physiques (*e.g.* températures).

STEP
Afin d'unifier l'ensemble des normes existantes au sein d'une unique, l'ISO (International Organization for Standardization) a mis en place en 1990 la norme STEP (STandard for the Exchange of Product model) dont l'objectif est de définir une représentation non ambiguë de l'échange de données de produits, interprétables par tout système informatique, et couvrant le cycle de vie de ces derniers ([Bou95]). STEP dispose du langage de modélisation EXPRESS et propose un ensemble de modèles de référence (protocoles d'applications) explicités dans le langage EXPRESS et spécifiant une vue particulière des modèles de produits selon un métier donné. Ainsi, un document respectant la norme STEP doit toujours être accompagné d'un protocole d'applications correspondant au domaine traité.

Le langage EXPRESS permet de mettre en place les règles de construction des produits utilisés. Pour ce faire, il utilise le concept des entités présenté dans le pa-

ragraphe 2.2.4. Il permet également de spécifier des contraintes sur des valeurs et dispose de l'héritage entre entités. L'IAI [7] (International Alliance for Interoperability), une agence souhaitant mettre en place des échanges d'informations dans le domaine de l'industrie du bâtiment, a mis en place son format standard : l'IFC (Industry Foundation Classes). Ce dernier s'appuie sur STEP et utilise un nouveau protocole d'applications spécifique au domaine de l'IAI.

Ce format d'échange possède de nombreuses qualités indéniables. Il permet de représenter un produit quel que soit le type métier grâce aux protocoles d'applications. Cependant, cet aspect a rendu le format très complexe à utiliser et à mettre en œuvre. Dans notre cas, son emploi nécessiterait de définir un protocole d'application complet et spécifique à nos besoins. Une longue phase de validation (3 ans) par l'ISO serait ensuite nécessaire.

STEP pourrait donc être utilisé comme format d'échange. Pour cela, un groupe de travail devrait se pencher sur le problème pour mettre en place un protocole d'application adapté. Cette tâche sort du cadre de ce doctorat, c'est pourquoi STEP n'est pas retenu.

MED

Les deux modèles précédents ont du mal à s'imposer en raison de leur complexité. Le format MED (Modélisation et Echanges de Données, [MED06]) a été normalisé par EDF et le CEA dans le but de fournir un moyen neutre d'échange de données entre les codes de calcul. Il est développé par EDF et le CEA et permet de stocker des maillages et des champs associés. Son but est de permettre de réaliser des études de simulations multiphysiques faisant collaborer différents logiciels de calcul scientifique. Il s'appuie sur le format binaire compressé HDF5 (Hierarchical Data Format, [HDF5]) pour sa performance et sa portabilité. Sous licence LGPL, MED est libre de droit et peut facilement être intégré à des développements grâce à la bibliothèque informatique mise en place. Avec la volonté d'effectuer des couplages, aussi bien multiphysiques que multiéchelles, le CEA et EDF tendent à faire supporter MED par leurs codes de calcul. Pour cette même raison, la plateforme SALOME intègre la gestion du format. Malheureusement, MED ne permet pas d'intégrer de représentations continues de la géométrie et il n'est pas possible d'ajouter de données autres que des champs sur maillages.

GDML

En 2005, le CERN, l'Organisation européenne pour la recherche nucléaire (anciennement Conseil Européen pour la Recherche Nucléaire) a mis en place un nouveau format appelé GDML (Geometry Description Markup Language, [CMPS06, GDML10]). Son but est d'unifier les applications logicielles utilisées dans le domaine de la physique nucléaire. Il s'appuie sur le langage XML et peut donc facilement évoluer en fonction des besoins. Le format peut contenir une géométrie continue type CSG et des données comme des constantes physiques ou des variables. Il ac-

7. Site de l'IAI : http ://www.buildingsmart.com/

cepte également l'intégration de données non prévues par l'intermédiaire de balises auxiliaires. Malheureusement, le GDML ne permet pas d'intégrer des champs de valeurs s'appuyant sur un maillage.

2.3.3 Bilan

Comme nous venons de le voir, le format STEP offre la possibilité de mettre en place une modélisation par entités. Sa complexité de mise en œuvre est toutefois un frein à son adoption. Le format MED est intéressant car compatible avec certains codes de calcul. L'impossibilité d'intégrer une représentation continue est néanmoins handicapant. Le GDML semble le format le plus adapté. Il permet d'ajouter des données, prévues ou non par le format, à une représentation continue. Il sera malgré tout nécessaire de le faire évoluer pour répondre à nos besoins.

La mise en place d'un modèle commun permet de pouvoir centraliser toutes les informations partagées par un ensemble de tâches. Parmi ces dernières, il peut s'agir du résultat d'un calcul, comme la température d'un fluide ou d'une modification géométrique. Dans ce dernier cas, le transfert de ces déformations nécessite l'utilisation d'une méthode préalable les caractérisant. Les approches possibles sont présentées dans la partie suivante.

2.4 Caractérisation de déformations

Le transfert de déformations tel que présenté dans le paragraphe 1.2 est abordée par une approche en deux temps. La première, la phase de définition de la déformation, impose que le maillage initial et le maillage modifié soient appariés. Elle peut être obtenue de différentes manières, comme une modification manuelle par un utilisateur ou le résultat d'un code de comportement mécanique. Dans ce dernier cas, il faut alors noter que le code peut utiliser un maillage à connectivité constante, c'est-à-dire identique tout au long du calcul. Pour améliorer les résultats, une autre approche possible ([BLCS05, GMCB11]) intègre une étape de remaillage permettant d'avoir des mailles plus fines aux endroits où les modifications géométriques sont importantes. Pour garder l'appariement entre les maillages initial et déformé, il est impératif de leur appliquer exactement les mêmes opérations à réaliser. Ainsi, l'ajout et le retrait de nœuds doivent être effectués de manière identique sur les deux maillages. Dans ce qui suit, les deux possibilités sont prises en compte et la phase de définition est vue comme une boîte noire.

Les méthodes de déformations pouvant être utilisées pour réaliser un transfert de modifications géométriques sont très nombreuses et différentes études permettent d'avoir une vue d'ensemble ([GM97, MM07, BPK*08]). Les plus à même pour répondre à notre problème sont présentées dans les deux premières parties (paragraphe 2.4.1 et paragraphe 2.4.2) en séparant celles modifiant l'espace dans lequel se trouve l'objet à déformer et celles travaillant directement sur l'objet. Les différentes méthodes de transfert sont ensuite exposées dans le paragraphe 2.4.3.

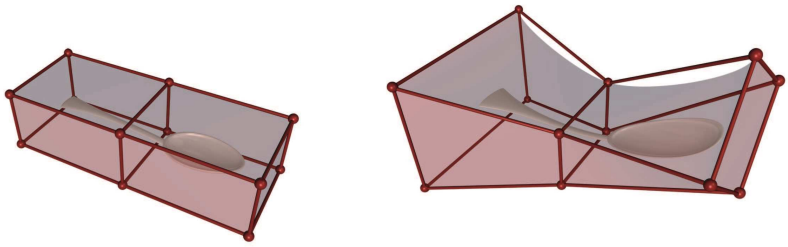
2.4.1 Déformations d'espaces et de sous-espaces

Le principe des méthodes de déformations d'espaces est de plonger un objet dans une zone prédéfinie et connue. En la modifiant, il est possible d'agir sur l'objet et de le déformer. Lorsque la zone n'englobe pas l'intégralité de l'objet, on parle de déformation de sous-espaces. Les principales méthodes sont présentées dans ce qui suit.

2.4.1.1 Déformations de formes

Initialement proposées par Barr ([Bar84]), les déformations de formes s'appuient sur une liste hiérarchique de transformations pour modifier un objet. Les transformations élémentaires comprennent les opérateurs de mise à l'échelle, de pincement, de courbure et de torsion. A chacune d'elles correspond une matrice. Plusieurs peuvent être combinées par multiplications dans le but de créer des modifications plus complexes.

L'idée de la déformation de l'espace dans lequel se trouve un objet a ensuite été reprise dans [SP86]. Les auteurs définissent une boîte englobant un objet à déformer et calculent les coordonnées paramétriques de l'objet pour pouvoir le lier à la boîte. Ils modifient ensuite la forme de la boîte englobante en déplaçant ses points de contrôle. L'objet se trouvant à l'intérieur est alors modifié (figure 2.6). Cette méthode a ensuite fait l'objet de nombreux travaux visant à l'améliorer et à la rendre plus générique. Les études [AS06, RGLG07, GB08] apportent de plus amples détails. Ces différentes approches, regroupées sous le terme de déformations de formes libres (*FFD, Free-Form Deformations*), sont beaucoup utilisées en modélisation géométrique avec interaction de l'utilisateur et la plupart des logiciels graphiques les intègrent. Les FFD souffrent néanmoins de plusieurs faiblesses, la plus importante étant le besoin d'interaction avec un utilisateur. De plus, très peu de méthodes permettent de modifier directement l'objet et donc de lui imposer des contraintes de passage, ce qui les rend difficilement utilisables dans notre cas.



(a) avant déformation (b) après déformation

Figure 2.6 – Principe de fonctionnement des FFD — extrait de [GB08]

2.4.1.2 Déformations utilisant des coordonnées différentielles

Les techniques de déformations utilisant des coordonnées différentielles (*gradient domain deformation techniques*) ont été développées ces dernières années. Au lieu de s'appuyer directement sur les coordonnées des nœuds d'un maillage, ce type d'approche calcule une fonction de déplacements à partir d'un champ de vecteurs d'orientation. Ce dernier est obtenu de manière à respecter des déplacements de nœuds imposés. La fonction et le champ sont liés par une équation faisant intervenir l'opérateur Laplacien et la divergence. L'intérêt est de focaliser la méthode sur le déplacement des points afin de préserver tous les détails. Appliquée aux maillages, il est nécessaire d'utiliser une discrétisation de l'opérateur Laplacien et de la divergence ([XZ09]). Cette étape n'est possible que dans le cas des maillages simplexes, c'est-à-dire constitués uniquement de triangles pour les maillages surfaciques et de tétraèdres pour les volumiques. Les études [BS08] et [XZ09] regroupent les principales méthodes et permettent d'avoir une vue d'ensemble. Les résultats obtenus sont de bonne qualité mais ces approches ont malheureusement l'inconvénient majeur de ne s'appliquer qu'aux maillages simplexes.

2.4.1.3 Fonctions de base radiale

Les fonctions de base radiale (*RBF, Radial Basis Functions*) sont utilisées pour interpoler des données éparses de plusieurs variables attachées à un ensemble de points. Ces données peuvent par exemple correspondre à des précipitations météorologiques ([JCMB01]), des températures pour modéliser un processus physique ([BCZD02]), des couleurs pour le débruitage d'images ([KZ03]), ... Dans le cas de modifications géométriques, elles correspondent aux déplacements des points d'attache, c'est-à-dire ceux dont la nouvelle position est connue et imposée. Ce type d'approches permet de prendre une empreinte de la déformation.

Pour réaliser cela, un réseau de fonctions de base radiale (*radial basis function network*) est utilisé. Il s'agit d'une combinaison linéaire de RBF. Pour un point v de $\mathbb{R}^D$, le réseau est défini ([dSB07]) par :

$$F(v) = \sum_{k=1}^{n} \lambda_k . \phi(\|v - L_k\|) \quad + \mathcal{L}(v) \tag{1}$$

où n est le nombre de points caractéristiques utilisés par le réseau, D la dimension de l'espace (*i.e.* 3 pour des maillages dans $\mathbb{R}^3$), $\|.\|$ une norme (généralement euclidienne) dans $\mathbb{R}^D$, ϕ une fonction de base radiale, $\{L_k\}_{k=1,..,n}$ de $\mathbb{R}^D$ les points caractéristiques, $\{\lambda_k\}_{k=1,..,n}$ de $\mathbb{R}^D$ les vecteurs de coefficients du réseau et $\mathcal{L}$ de $\mathbb{R}^D$ une fonction polynomiale de faible degré, inférieur à celui de la fonction ϕ ([Isk03]). Ainsi, $F(v)$ permet de connaître la nouvelle position du point v après déplacement. Cette méthode n'est pas dépendante d'un type de maillage et peut être utilisée dans un espace de dimension quelconque.

La fonction polynomiale $\mathcal{L}$ correspond aux grands déplacements globaux de la déformation. Généralement, une fonction polynomiale linéaire est utilisée et permet

de représenter globalement la déformation ([JT10]). Dans ce qui suit, elle est négligée pour simplifier les explications, le raisonnement restant le même. A chaque point caractéristique L_k, $k = 1, .., n$, correspond une donnée, notée L'_k, que la fonction F doit interpoler. De cette manière, il est possible d'écrire :

$$F(L_k) = L'_k \quad ; \; k = 1, .., n \tag{2}$$

Les coefficients du réseau sont calculés en résolvant le système d'équations linéaires induit par ces équations. La matrice Ψ de $\mathcal{M}_{n,n}(\mathbb{R})$ est définie par :

$$\Psi = \Big(\phi(\|L_j - L_k\|_2)\Big)_{1 \leq j,k \leq n} \tag{3}$$

En notant $L' = \{L'_k\}_{k=1,..,n}$, le système d'équations linéaires $AX = B$, A de $\mathcal{M}_{n,n}(\mathbb{R})$, à résoudre est défini par :

$$AX = \Psi \; \lambda = L' = B \tag{4}$$

Les RBF ont fait l'objet de nombreuses études ([SW00, BR02, Isk03, dSB07]). Le choix de la fonction ϕ a un impact sur le temps de calcul et sur le résultat final. Il existe deux grandes familles de RBF : celles à support local et celles à support global. Les premières induisent un système d'équations linéaires à matrice creuse. Les secondes sont généralement plus précises ([CBC*01, BK05]) mais demandent un temps de calcul important et une place mémoire conséquente en raison d'une matrice pleine de grande taille. Différentes approches ont donc été proposées pour améliorer ce point ([CBC*01, BCR01, BK05]). Le principe commun de ces méthodes est de faire une approximation du résultat et non plus une interpolation. Dans [BK05], les auteurs proposent également une méthode incrémentale pour effectuer la résolution du système. Leurs points sont déplacés par transformations affines. Il leur est alors possible de décomposer leur matrice en deux sous-matrices plus petites, permettant de pré-calculer les coefficients des RBF dont les centres sont fixes. Cette approche n'est pas réalisable dans notre cas car les déplacements de nos points sont supposés quelconques. De plus, leur méthode itérative de résolution n'est performante que dans le cas d'une approximation de la solution, c'est-à-dire lorsque toutes les RBF ne sont pas utilisées. Certains auteurs ont fait remarquer qu'un type particulier de RBF à support compact, les CS-RBF (Compactly-supported radial basis functions) proposées par Wendland dans [Wen95], aboutissaient à une précision très proche, voire supérieure, à celle des RBF à support global ([BR02, dSB07]).

Les fonctions RBF les plus courantes sont la multiquadrique, la Thin-Plate Spline (TPS), la gaussienne et la CS-RBF. L'utilisation de la RBF gaussienne peut aboutir à de mauvais résultats pour de grands déplacements ([dSB07]). L'ensemble des autres fonctions fournissent de bons résultats ([dSB07]) et ont fait l'objet de nombreux articles : multiquadrique ([FRR96, FSRR00, LXZW10]), TPS ([Boo89, FSRR00, BCZD02, KZ03, HH06]), CS-RBF ([Wen05a, Wen05b, LTK09, RA10]). Il faut également noter que les TPS minimisent une énergie de flexion ([Boo89]) et permettent ainsi d'obtenir un résultat "lisse" (sans forte variation).

Le calcul des coefficients du réseau de RBF nécessite la résolution d'un système d'équations linéaires. Le temps de calcul de cette étape est élevé et dépendant du nombre de nœuds à interpoler. Dans [YBK10], les auteurs mettent en place une approche de résolution parallèle. Elle permet d'améliorer le temps de calcul mais le problème reste présent.

2.4.1.4 Déformations physiques

Les méthodes de déformations physiques (*physics-based modeling methods*) sont des approches faisant intervenir des équations de la physique. Elles résolvent directement ces équations ou approchent itérativement le résultat, permettant ainsi d'obtenir une déformation réaliste. L'étude [NMK*06] les détaille. Dans ce qui suit, seules les plus pertinentes pour répondre à notre problématique sont présentées.

La méthode proposée dans [vFTS06] utilise un champ de vecteurs à divergence nulle pour interpoler le déplacement d'un ensemble de points définis sur le maillage. L'approche permet de réaliser la déformation d'un maillage sans créer d'intersections de mailles, tout en conservant les détails, en préservant le volume et en aboutissant à un résultat C^1-continu. Son principal inconvénient reste néanmoins la difficulté de construction du champ de vecteurs ([BPK*08]).

Dans [Hel03], le déplacement d'un ensemble de nœuds est imposé et un système utilisant l'opérateur laplacien est résolu pour pouvoir calculer la nouvelle position des nœuds restants. Les résultats sont de bonne qualité mais se dégradent à mesure que les déplacements augmentent.

Botsch et *al.* proposent dans [BPWG07] une méthode s'appuyant sur l'énergie élastique pour réaliser des déformations lisses. Ils commencent par intégrer le maillage à déformer dans un ensemble de cellules volumiques. Le déplacement de ces dernières est ensuite gouverné par la minimisation d'une énergie élastique. La position finale des nœuds du maillage est obtenue par interpolation RBF du déplacement de chaque cellule. La figure 2.7 illustre le résultat obtenu sur un maillage. Certaines parties sont des zones de modification imposées et les autres sont déplacées en tenant compte de l'énergie élastique. Le résultat est de bonne qualité mais ce type d'approches souffre de l'impossibilité de fixer des déplacements à des nœuds particuliers. Son application est orientée vers une interaction avec un utilisateur.

Figure 2.7 – Déformations s'appuyant sur des cellules rigides — extrait de [BPWG07] (de gauche à droite : le maillage initial, l'ensemble des cellules volumiques déplacées, le maillage modifié)

Sumner et *al.* ([SSP07]) construisent un graphe pour déformer l'espace dans lequel est plongé un objet. A chaque nœud du graphe est attribué une transformation affine rigide et une fonction d'énergie est associée à la déformation. Après minimisation de cette dernière, l'objet complet est modifié. Les conclusions sont identiques à la précédente approche. L'utilisateur doit interagir avec la méthode et il est impossible de fixer des déplacement précis aux nœuds.

2.4.2 Surfaces déformables

Les surfaces déformables (*surface-based deformation*) sont une famille de méthodes dont la déformation n'agit pas sur un espace contenant l'objet à modifier, mais directement sur l'objet lui-même. Les approches sont nombreuses et les études [MDA01, YSBS06] les détaillent. Les principales sont présentées dans ce qui suit.

2.4.2.1 Multirésolution

Lorsque les déformations à réaliser font intervenir des maillages avec un nombre élevé de nœuds, il est possible d'utiliser une approche par multirésolution. Le principe, illustré par la figure 2.8, est de séparer du maillage complet les hautes et les basses fréquences, c'est-à-dire les détails et la forme globale. La modification géométrique est réalisée sur la forme globale et le maillage final est construit en rajoutant les détails. Les méthodes diffèrent sur la séparation des hautes et des basses

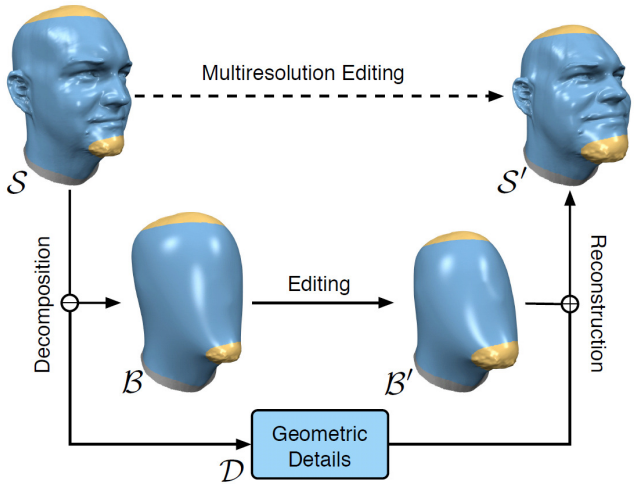


Figure 2.8 – Principe des approches par multirésolution — extrait de [BS08]

fréquences, et sur la manière de réaliser la déformation de la forme globale. L'intégration de contraintes non linéaires dans la déformation multirésolution de courbes et de surfaces lisses est étudiée dans [HBS05, Sau05]. Une méthode permettant de traiter les gros volumes de voxels est présentée dans [HT08]. Ce type d'approches est intéressant pour réduire la complexité des modèles utilisés mais ne permet pas directement de réaliser un transfert de déformations. Elles pourraient cependant être utilisées en pré-traitement pour aboutir à un problème plus simple.

2.4.2.2 Déformations physiques

La déformation de maillages peut également faire intervenir des équations physiques. Parmi ces méthodes ([NMK*06]), seules les plus pertinentes pour répondre à notre problématique sont présentées.

Les modèles masses-ressorts ont été utilisés dans [PB81]. Un maillage est vu sous forme d'un système où un poids est attribué à chaque nœud et les arêtes sont représentées par des ressorts. La position des nœuds dont le déplacement est connu est imposée, initialisant ainsi l'oscillation du système. Le maillage déformé correspond au résultat après stabilisation du système. Cette approche est itérative et souffre d'un temps de calcul augmentant avec le nombre de nœuds dans le maillage et du critère de convergence utilisé.

Le principe des méthodes de déformations par croissance de régions est de positionner un maillage et de le modifier itérativement pour approcher au mieux une forme voulue, comme une surface déformée. Pour cela, deux énergies sont utilisées. La première est une énergie de gonflement favorisant le déplacement du modèle et l'éloignement des points dans la direction de leur normale. La seconde est une énergie de raideur ayant pour but de freiner ce déplacement pour éviter les artefacts. Parmi les plus connues, il est possible de citer les déformations élastiques ([TPBF87]), les contours actifs ([KWT88]) et les maillages simplexes de Delingette ([Del99]). Une étude plus complète sur les contours actifs est disponible dans [Ter03]. Malheureusement, ce type d'approches fait généralement intervenir des paramètres dont les valeurs sont difficiles à déterminer. Tout comme pour les modèles masses-ressorts, elles sont handicapées par le nombre d'itérations nécessaire pour aboutir au résultat. De plus, elles posent le problème du critère de convergence de l'approche itérative.

2.4.3 Méthodes de transfert de déformations

Les méthodes de transfert de déformations proposées sont souvent axées sur la problématique de la modification de formes différentes. L'approche détaillée dans [BVGP09] lie le maillage source et celui de destination par apprentissage automatique. Différentes paires de poses sont nécessaires afin de créer une carte de correspondances qui est ensuite utilisée pour le transfert de nouvelles poses. Dans [SP04], les auteurs réalisent une correspondance entre deux maillages et proposent une méthode de déformation s'appuyant sur les coordonnées différentielles. Des évolutions ont ensuite été proposées ([BCWG09, ZXTD10]) pour permettre de gérer le transfert de déformations à des objets constitués de plusieurs sous-maillages. Afin d'améliorer la préservation des détails, une méthode multirésolution est ajoutée dans [BSPG06]. Ces approches aboutissent à de bons résultats mais se restreignent au cas des maillages simplexes. Le principal inconvénient de ces différentes méthodes est de se focaliser sur le problème du transfert entre des objets différents. Il en résulte qu'elles ne visent pas à interpoler un déplacement mais à l'approcher au mieux pour que la modification paraisse identique à celle sur lequel elle s'appuie (figure 2.9).

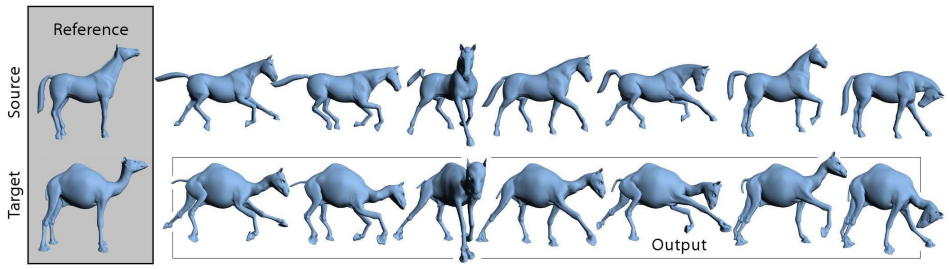


Figure 2.9 – Transfert de déformations entre des objets de formes différentes — extrait de [SP04]

Dans le cas où la déformation doit être transférée entre deux maillages s'appuyant sur une même géométrie, une possibilité est de réaliser une interpolation linéaire du déplacement. De par son côté simpliste, le temps de calcul de cette approche devrait être faible mais le résultat pourrait ne pas être de bonne qualité dans le cas où le maillage sur lequel est appliqué la déformation est plus fin que celui sur lequel elle a été calculée.

Pour améliorer ce point, il est possible de réaliser une interpolation par morceaux utilisant des patchs de surfaces polynomiales ([HB03, YHB05]), comme des Bézier, B-spline ou NURBS. De cette manière, le support de la déformation n'est plus un maillage mais une combinaison de surfaces continues. En réalisant ce même travail sur le maillage initial, *i.e.* avant déformation, il est possible de connaître le déplacement d'un point quelconque. Le transfert de la déformation est donc possible. En revanche, une telle interpolation par morceaux aboutit à un résultat G^1-continu aux raccords entre les patchs. Il est donc visuellement lisse mais ne permet pas de respecter une contrainte de continuité au minimum C^1. Seuls les maillages triangulaires sont supportés et cette méthode n'est pas applicable au cas volumique. Elle ne peut donc être utilisée.

Un autre choix possible est d'utiliser un type d'approches regroupées sous le terme de méthodes sans maillage (*meshless methods*). Elles s'appuient uniquement sur les coordonnées des nœuds et permettent de ne pas être dépendant d'un type de maillages. Elles sont très nombreuses et ont fait l'objet de plusieurs études dont [FM04, VRL07]. Elles sont généralement utilisées pour la simulation de phénomènes physiques, comme l'écoulement d'un fluide. Dans le cas particulier de la déformation de maillages, Wendland indique dans [Wen05b] que les approches qui aboutissent aux meilleurs résultats sont les RBF.

2.4.4 Bilan

La méthode de transfert de déformations à mettre en place doit être générique pour pouvoir prendre en compte n'importe quel type de maillages, permettre de travailler avec un nombre élevé de nœuds et automatique (*i.e.* ne pas nécessiter l'intervention d'un utilisateur). Il doit également être possible de définir des contraintes de passage pour interpoler une modification géométrique.

Les FFD ne respectent pas ces conditions. Les déformations utilisant des coordonnées différentielles impose un maillage simplexe, limitant les possibilités de transfert. Les fonctions de base radiale induisent un temps de calcul élevé pour un nombre important de nœuds à interpoler. Certaines techniques s'appuient sur des équations physiques et aboutissent à des résultats réalistes. Leur calcul peut être direct, auquel cas il est compliqué de définir des contraintes de passage, ou itérative, ce qui entraine obligatoirement l'intervention d'un critère de convergence difficile à déterminer. Les méthodes multirésolution offrent la possibilité d'abaisser la complexité des modèles utilisés. Elles ne peuvent pas directement être utilisées comme technique de transfert mais pourraient lui simplifier la tâche. Enfin, une interpolation linéaire de la déformation peut être envisagée. Des tests sont réalisés dans le chapitre 4 pour vérifier l'intérêt de cette approche en terme de temps de calcul et de qualité du résultat obtenu.

Malgré leur temps de calcul élevé, il semble également pertinent de s'orienter vers les fonctions de base radiale. Une étude préliminaire est réalisée et présentée dans le chapitre 4 afin d'illustrer les résultats obtenus avec les trois RBF retenues, à savoir la multiquadrique, la TPS et la CS-RBF. Ces dernières souffrant d'un temps de calcul élevé, une amélioration possible viserait à baisser la complexité des maillages utilisés. Pour cela, l'approche multirésolution pourrait être utilisée. Dans [BS08], les auteurs précisent qu'elle peut causer des artéfacts dans le cas de déformations importantes. De plus, les problèmes liés à leur mise en œuvre et l'impossibilité de définir clairement le nombre de nœuds à interpoler sont les raisons pour lesquelles elle n'a pas été retenue. Une autre possibilité est l'utilisation d'une méthode de simplification de maillages, assez générique pour répondre au problème. Il faut noter que l'utilisation d'un réseau de RBF nécessite de garder l'appariement entre les maillages avant et après déformation. De plus, pour que l'interpolation réalisée s'appuie sur la déformation qui a été définie, les nœuds des maillages simplifiés doivent être un sous-ensemble de ceux des maillages initiaux (*i.e.* non simplifiés). Une étude des méthodes existantes est proposée dans ce qui suit.

2.5 Méthodes de simplification

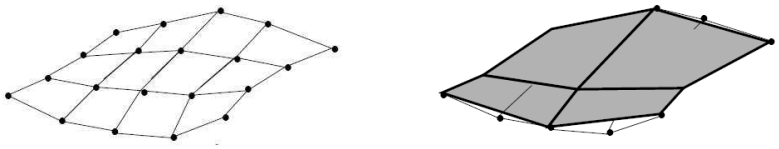
Ces dernières années, de très nombreuses méthodes de simplification ont été développées et ont fait l'objet de plusieurs études ([CMS98, HG97, Gar99, Lue01, Tal04]). Malheureusement, ces différentes approches sont généralement dépendantes de la dimension du maillage (*i.e.* surfacique, volumique) et d'une forme de maille (*i.e.* triangulaire, quadrangulaire). Cette partie s'attache donc à présenter succinctement les différentes méthodes en privilégiant les méthodes surfaciques. Elles ont souvent fait l'objet de travaux particuliers pour être adaptées au cas des maillages volumiques. Les méthodes purement volumiques n'étant pas applicables aux maillages surfaciques, elles ne seront pas abordées.

2.5.1 Fusions de faces

La technique des fusions de faces (*region merging*), présentée dans [DZ91, HH93], vise à fusionner les faces adjacentes coplanaires ou presque coplanaires, c'est-à-dire les faces dont l'écart entre les normales est faible (*i.e.* inférieur à un seuil fixé). Ceci est illustré par la figure 2.10. L'algorithme se décompose alors en quatre étapes principales ([HH93]) :

1. rechercher les ensembles de faces coplanaires et presque coplanaires,
2. créer une liste de segments pour chacun d'eux et supprimer les doublons,
3. recréer les polygones,
4. remailler le résultat obtenu.

Cette approche ne modifie pas la topologie. Elle est relativement rapide avec une complexité en $O(n.\log(n))$. Il faut toutefois signaler qu'elle n'est pas très efficace sur les surfaces contenant de fortes courbures ([HG97]).

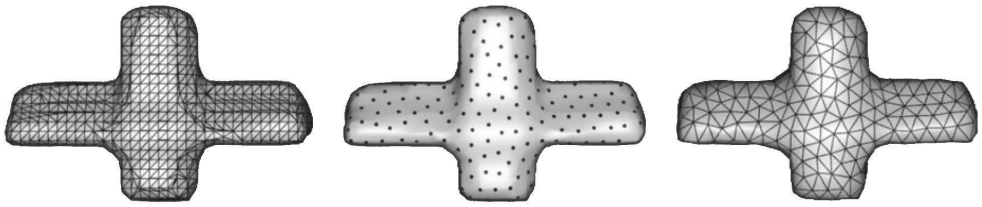


(a) maillage initial (b) maillage simplifié

Figure 2.10 – Simplification de maillages par fusions de faces — extrait de [DZ91]

2.5.2 Ré-échantillonnage de surfaces polygonales

Le principe de la méthode de ré-échantillonnage de surfaces polygonales (*re-tiling polygonal surfaces*) est d'effectuer un remaillage complet. Pour cela, l'auteur ([Tur92]) propose de placer un nombre (défini par l'utilisateur) de sommets de manière aléatoire sur la surface initiale. Ces derniers sont alors soumis à des forces de répulsions (type masse-ressort, *c.f.* paragraphe 2.4.2.2) afin qu'ils se positionnent de manière uniforme, tout en restant sur la surface initiale. Les sommets initiaux sont ensuite supprimés itérativement pour ne garder que ceux nouvellement ajoutés (figure 2.11). Ce procédé est très performant pour la simplification de surfaces lisses. L'auteur a également proposé une modification des forces de répulsions utilisées pour disposer de plus de points dans les zones courbées et ainsi obtenir un maillage plus dense dans ces endroits. Cet algorithme a toutefois tendance à "gommer" les zones saillantes ([HG97]).



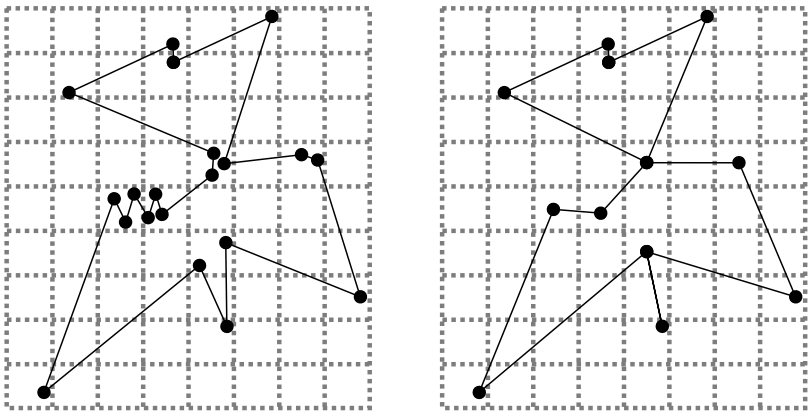
(a) maillage initial (b) nouveaux sommets (c) maillage simplifié

Figure 2.11 – Simplification par ré-échantillonnage — extrait de [Tur92]

2.5.3 Plongement dans une grille régulière (*vertex clustering*)

Une autre grande famille de simplification est le *vertex clustering* ([RB93]). Il s'agit de plonger le maillage initial dans une grille régulière. Chaque sommet se voit alors attaché à une cellule de cette dernière. Enfin, tous les sommets d'une même cellule sont remplacés par un seul point. Ce principe est illustré par la figure 2.12.

Cette technique est très rapide et fonctionne quel que soit le type de maillage utilisé. De plus, son adaptation au cas des maillages volumiques est immédiat. Toutefois, le maillage obtenu est souvent de faible qualité et, comme le montre la figure 2.12, les détails sont supprimés ([HG97, Gar99]).



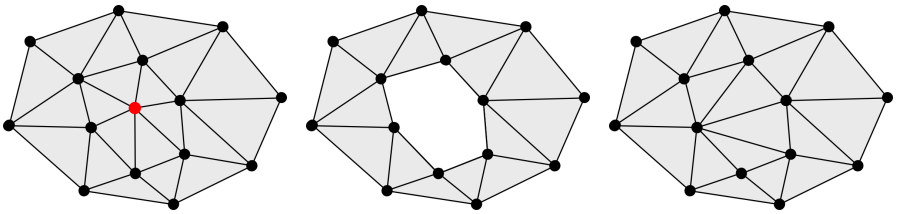
(a) avant (b) après

Figure 2.12 – Plongement d'un maillage dans une grille régulière

Dans le but de combler certains de ces inconvénients, de nombreux travaux ont porté sur l'amélioration de cette technique. On peut citer [LT97] dans lequel les auteurs remplacent la grille régulière par un ensemble de cellules où chacune d'elles peut avoir une forme (cube, sphère) et une taille différentes. Cela permet d'obtenir une simplification aboutissant à un maillage de meilleure qualité ([HG97]). Plus récemment, Kanaya et *al.* ([KTKN05]) ont développé une méthode pour préserver la topologie en ne fusionnant que les sommets liés au sein d'une même cellule.

2.5.4 Décimation

Le principe original de la décimation (*decimation*), proposé dans [SZL92], est de supprimer un sommet ainsi que toutes les faces qui y sont liées, et d'effectuer un remaillage du trou ainsi formé. Cette technique, illustrée figure 2.13, est généralement limitée aux cas des maillages *manifolds*[8] ([Gar99]). Originellement, les auteurs



(a) avant (b) suppresion (c) après

Figure 2.13 – Décimation d'un nœud

utilisent comme critère pour le choix du nœud à simplifier, l'angle ou la distance entre le nœud sélectionné et son voisinage. Ce critère, qui est local, a été modifié dans de nombreux travaux. On peut, par exemple, citer [CCMS97] qui s'appuie sur un critère global pour effectuer sa simplification en calculant l'impact global d'une décimation. Dans [CVM*96], les auteurs utilisent deux enveloppes, l'une externe et l'autre interne, et forcent la simplification à faire évoluer le maillage entre elles. Les deux enveloppes sont obtenues en déplaçant d'une certaine valeur ε chaque nœud dans la direction de sa normale pour l'enveloppe externe ou dans la direction opposée pour l'enveloppe interne. Elles peuvent être assimilées à une érosion et une dilatation de la surface du maillage initial, comme l'illustre la figure 2.14. Cette approche est appelée *"Simplification envelopes"* ([CVM*96]). Dans une autre mé-

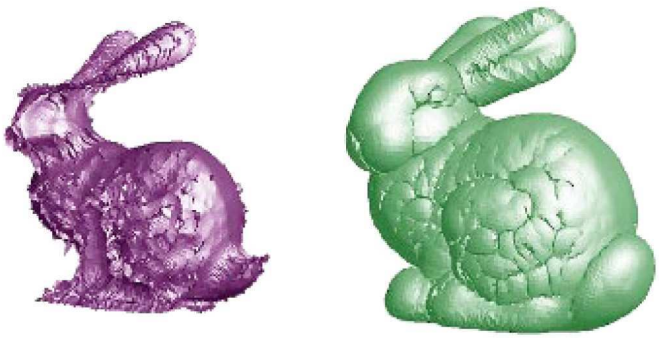


Figure 2.14 – Méthode *"Simplification envelopes"* — extrait de [CVM*96] (à gauche, enveloppe interne et à droite, enveloppe externe)

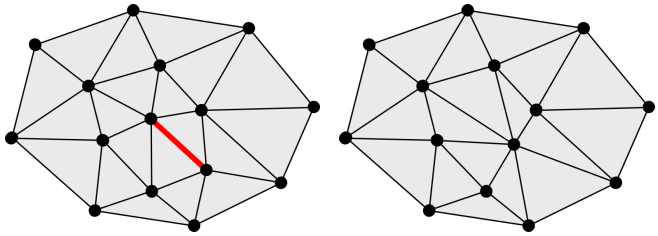
8. Un maillage est dit *manifold* (ou variété) si tout sommet est le centre d'un disque topologique, autrement dit possède un voisinage homéomorphe à un disque dans le cas surfacique, à une sphère dans le cas volumique.

thode ([Hus08]), l'auteur propose de choisir les sommets à décimer en fonction de la déviation du vecteur normal des différentes faces adjacentes. Enfin, récemment dans [CG09], les auteurs détaillent une méthode qui utilise le volume des polyèdres générés par les faces adjacentes à un sommet, comme critère de simplification.

Toutefois, l'ensemble de ces méthodes requièrent une phase de remaillage du trou formé par la suppression d'un sommet qui n'est pas toujours triviale.

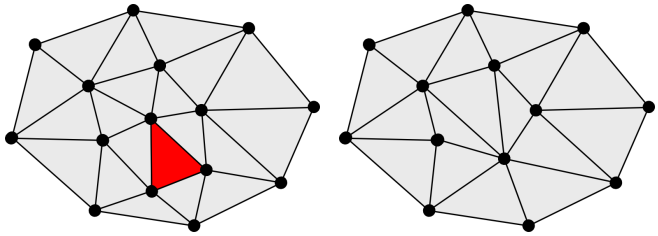
2.5.5 Contraction d'arêtes/faces

Cette technique, proche de la précédente, vise à réunir une arête ou une face en un sommet unique. On parle alors de contraction d'arêtes/faces (*edge/face collapse*). Initialement ([HDD*93]), le nouveau sommet créé se trouve au milieu de l'arête/face (figure 2.15, 2.16). Certaines méthodes ([Hop96, GH97]) utilisent un critère de simplification permettant de calculer la position optimale du nouveau sommet grâce à une minimisation de l'erreur engendrée par la fusion. Enfin, on peut également citer les approches dans lesquelles le nœud résultat est l'un des nœuds de l'arête/face initiale. On parle alors de demi-contraction d'arêtes/faces (*half-edge/face collapse*).



(a) avant (b) après

Figure 2.15 – Contraction d'une arête



(a) avant (b) après

Figure 2.16 – Contraction d'une face

Il existe de nombreuses méthodes de simplification de maillages surfaciques qui utilisent la technique de la contraction d'arêtes. Elles diffèrent principalement par le critère utilisé pour ordonner l'ensemble d'arêtes/faces, ce critère caractérisant un type d'erreur particulier. On peut alors citer les plus connues :

Progressive meshes :
Cette méthode, proposée par Hoppe dans [Hop96], utilise comme critère une énergie à minimiser :

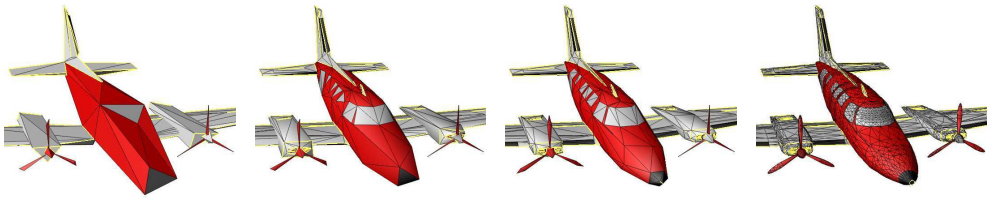
$$E = E_{dist} + E_{spring} + E_{scalar} + E_{disc} \tag{5}$$

où :

- E_{dist} et E_{spring} correspondent respectivement à la distance entre le maillage initial et son simplifié, et à la longueur des arêtes des faces. Ils jouent ainsi sur la précision globale du maillage obtenu,
- E_{scalar} correspond aux erreurs d'approximation faites sur un ensemble de valeurs scalaires attachées au maillage, favorisant ainsi la précision locale du résultat.
- E_{disc} pénalise les modifications géométriques et topologiques.

Ainsi, les maillages simplifiés obtenus à partir d'une telle simplification sont de bonne qualité mais au prix d'un temps de calcul élevé. Il faut ajouter à cela que la topologie n'est pas forcément respectée et que les nœuds du maillage résultat ne constituent pas un sous-ensemble du maillage initial. Néanmoins, cette approche est intéressante car réversible. En effet, en connaissant le maillage de base ainsi que l'ensemble des opérations inverses de celles réalisées lors de la simplification, il est possible de revenir au maillage initial ou tout autre modèle intermédiaire. C'est la raison pour laquelle l'auteur baptise sa méthode ainsi.

Cette méthode est généralisée dans [PH97] au cas général des complexes simpliciaux. Néanmoins, contrairement à [Hop96], la simplification ne permet pas de prendre en compte les erreurs d'approximations faites sur les valeurs associées aux maillages. [SG98] comble cela et nomme sa méthode *progressive tetrahedralizations*.



(a) 150 faces (b) 500 faces (c) 1 000 faces (d) 13 546 faces

Figure 2.17 – Contraction par la méthode *Progressive meshes* — extrait de [Hop96]

Full-range appromixation of triangulated polyhedra ([RR96]) :
Introduit dans [RR96], le critère est issu de l'observation qu'un point sur un maillage est situé à l'intersection des plans dans lesquels se trouvent chacune de ses faces adjacentes. Ainsi, les auteurs associent à chaque sommet une *zone* contenant un ensemble de plans et ils l'initialisent avec les plans de ses faces adjacentes. L'erreur

Δ sur un sommet v est alors calculée comme étant sa distance quadratique maximale avec chacun des plans de sa *zone*.

$$\Delta(v) = \max_{p \in zone(v)} \left[d^2(v,p) \right] \tag{6}$$

A noter qu'au début de la simplification, chaque sommet v possède un coût $\Delta(v)$ nul. Posons v_j et v_k deux sommets d'une arête. Le résultat de la contraction de v_j et v_k est noté $\bar{v}$ et représenté : $(v_j, v_k) \rightarrow \bar{v}$. Lors d'une contraction d'arête, la *zone* du sommet résultat est l'union des *zones* des sommets initiaux :

$$(v_j, v_k) \rightarrow \bar{v} \Rightarrow zone(\bar{v}) = zone(v_j) \cup zone(v_k) \tag{7}$$

Ainsi, $\Delta(\bar{v})$ correspond à l'erreur induite par la contraction des sommets v_j et v_k, avec $\bar{v} = v_j$ ou $\bar{v} = v_k$ de manière à minimiser $\Delta(\bar{v})$.

Quadric Error Metric (QEM) :
Garland et *al.* ([GH97]) modifient l'approche [RR96] en prenant comme erreur la somme des distances quadratiques entre un sommet et chacun des plans de sa *zone*. Ils font alors remarquer que la distance au carré d'un point v à un plan p peut être écrit sous la forme :

$$d^2(v,p) = ({}^t p.v)^2 = ({}^t v.p).({}^t p.v) = {}^t v.(p.{}^t p).v \tag{8}$$

où $v = {}^t\begin{pmatrix} v^{(x)} & v^{(y)} & v^{(z)} & 1 \end{pmatrix}$ et $p = {}^t\begin{pmatrix} a & b & c & d \end{pmatrix}$ avec $a^2 + b^2 + c^2 = 1$. Ainsi, le critère Δ peut s'écrire sous la forme :

$$\begin{aligned}
\Delta(v) &= \sum_{p \in zone(v)} d^2(v,p) && (9)\\
&= \sum_{p \in zone(v)} {}^t v.(p.{}^t p).v && (10)\\
&= {}^t v. \left(\sum_{p \in zone(v)} (p.{}^t p) \right) .v && (11)\\
&= {}^t v. \left(\sum_{p \in zone(v)} K_p \right) .v && (12)
\end{aligned}$$

avec :

$$K_p = p.{}^t p = \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix} \tag{13}$$

Enfin, ils nomment Q la quadrique contenant l'ensemble des plans :

$$Q(v) = \sum_{p \in zone(v)} K_p \tag{14}$$

donc :

$$\Delta(v) = {}^t v.Q(v).v \tag{15}$$

Les auteurs montrent ainsi qu'il est possible de calculer l'erreur induite par une contraction en attachant sur chaque sommet une quadrique Q ne contenant que 10 réels (en double précision) de 8 octets chacun, soit 80 octets par sommet.

Lors de la contraction, les auteurs définissent la quadrique associée au sommet résultat $\bar{v}$ comme étant la somme des quadriques des sommets initiaux. Ainsi, pour une contraction $(v_i, v_j) \rightarrow \bar{v}$, la quadrique de $\bar{v}$ sera : $Q(\bar{v}) = Q(v_i) + Q(v_j)$. Dans [GH97], le sommet résultat de la contraction n'est pas obligatoirement un des deux de l'arête. Ce point n'est pas traité dans cette partie car les nœuds des maillages simplifiés doivent nécessairement être un sous-ensemble de ceux des maillages initiaux.

La complexité de cette méthode est en $O(n.\log(n))$ ([GS02]). Les résultats obtenus sont de bonne qualité grâce à des erreurs locale et globale relativement faibles, ainsi qu'un temps de calcul intéressant ([vP06]). De nombreux travaux s'appuient sur cette approche ([Gar99, GS02, SG05, GZ05, Tra08, LZ08]) pour ces raisons. Cette méthode possède toutefois quelques faiblesses, la principale étant qu'elle ne s'applique qu'aux maillages surfaciques à base de triangles.

Dans [GZ05], Garland et Zhou généralisent l'utilisation des *QEM* à n'importe quelle dimension, offrant alors la possibilité de traiter les maillages volumiques. L'approche reste toutefois restreinte aux mailles simplexes (triangles en 2D, tétraèdres en 3D, ...). De plus, il est impossible de simplifier un maillage dont la dimension est identique à celle de l'espace ; un maillage volumique dans $\mathbb{R}^3$ ne pourra donc pas être pris en compte.

2.5.6 Bilan

Les techniques de simplification sont nombreuses mais souvent dépendantes d'une représentation géométrique particulière. L'approche recherchée devant, au contraire, être la plus générique possible, les méthodes purement volumiques n'ont pas été abordées. De plus, il est important de pouvoir traiter des maillages surfaciques et volumiques, quel que soit le type de mailles. Après simplification, les nœuds doivent être un sous-ensemble des maillages initiaux pour s'assurer que l'interpolation de la modification géométrique s'appuie sur des coordonnées obtenues lors de la phase de définition de la déformation. L'appariement entre les maillages avant et après déformation doit également être conservé et il est nécessaire de pouvoir définir le nombre final de nœuds, *i.e.* après simplification.

Les méthodes de fusions de faces et de décimation effectuent une suppression de nœuds et réalisent ensuite un remaillage du trou formé. Cette dernière étape est complexe et son adaptation au cas volumique n'est pas triviale. Les approches par ré-échantillonnage et par plongement dans une grille régulière dégradent le résultat en supprimant des détails. La contraction d'arêtes/faces apparaît la plus intéressante. Plusieurs critères peuvent être utilisés mais le plus performant est la

métrique de l'erreur quadratique (*QEM, Quadric Error Metric*). Le résultat est obtenu rapidement au regard du temps de calcul de l'approche *Progressive meshes* et l'espace mémoire requis est faible comparativement à la méthode *Full-range appromixation of triangulated polyhedra*. Elle n'est malheureusement pas générique car elle requiert un maillage surfacique triangulaire (limitation outrepassée dans le paragraphe 5.2.2.2).

2.6 Synthèse

Ce chapitre a présenté différentes méthodes existantes liées à la problématique du transfert de déformations. Concernant le couplage de plusieurs codes de calcul, les approches possibles ne permettent pas de prendre en compte les modifications géométriques. Une nouvelle technique de couplage est exposée dans ce qui suit. On retiendra néanmoins que les outils proposés par la plateforme SALOME sont assez génériques pour être utilisable par cette dernière. De plus, bien que l'interface ICoCo n'ait pas été retenue dans le cadre de cette thèse, son intégration future n'est pas pour autant impossible. En effet, elle devra être modifiée afin de permettre un transfert de déformations. Un développement est également nécessaire de la part des codes de calcul afin qu'un plus grand nombre l'implémente. Pour centraliser les informations initiales et les résultats d'une étude, un modèle commun doit être intégré à l'approche proposée. Sa mise en place peut être en partie résolue en adoptant une modélisation par entités. Des développements sont nécessaires afin de définir les caractéristiques et les entités permettant de répondre à nos besoins. L'ensemble de ces points fait l'objet du chapitre 3.

Différentes approches de transfert de modifications géométriques ont également été détaillées. Il en résulte qu'il n'existe pas, à ce jour, de méthodes permettant de répondre à l'ensemble des besoins exposés dans le paragraphe 1.2. Pour cette raison, deux approches possibles ont été proposées. Tout d'abord, l'utilisation d'une interpolation linéaire pourrait aboutir à des résultats intéressants. La seconde porte sur un réseau de fonctions de base radiale. Ces dernières souffrent d'un temps de calcul élevé. Pour le réduire, les méthodes de simplification ont été discutées. Ces différentes fonctions de transfert font l'objet de tests dans le chapitre 4 afin d'illustrer les résultats obtenus et de les confronter. Le chapitre 5 présente finalement les nouvelles fonctions mises en place.

Chapitre 3

Couplage de codes de calcul et modèle commun

Sommaire

3.1 Procédure de couplage de codes 43
3.2 Architecture du modèle commun 45
3.2.1 Intégration de données 45
3.2.2 Définition des caractéristiques 47
3.2.3 Niveau de finesse 48
3.2.4 Définition des entités 49
3.2.4.1 Entité de base 49
3.2.4.2 Entité abstraite 50
3.2.4.3 Entité élémentaire 51
3.2.4.4 Entité complexe 52
3.3 Modèle commun mis en place 56
3.4 Synthèse 56

Le couplage de codes de calcul est une opération qui vise à faire interagir des simulations numériques. Pour cela, différents échanges de données sont nécessaires. Malheureusement, le chapitre précédent a permis de montrer qu'il n'existe pas, à ce jour, de méthode permettant de prendre en compte le transfert de déformations géométriques pour l'ensemble des codes de calcul. Une procédure complète d'échange de données est donc mise en place dans le paragraphe 3.1. Cette dernière fait intervenir un modèle commun. Son architecture est exposée dans le paragraphe 3.2. Le travail effectivement réalisé est présenté dans le paragraphe 3.3.

3.1 Procédure de couplage de codes

La procédure proposée permet de décrire les différentes étapes qui doivent être effectuées pour réaliser un couplage prenant en compte les évolutions de la géométrie. Le coupleur mis en place est une entité dont le but est de lier les différents codes de calcul utilisés. Afin d'être le plus évolutif possible, il doit être possible de remplacer un code par n'importe quel autre d'une même discipline physique sans pour autant impliquer de lourds (re)développements. Ces derniers ne doivent donc pas avoir de dépendances directes entre eux. L'intégration d'un modèle commun et l'utilisation d'une interface générique permettent cette séparation entre les codes.

Pour répondre au besoin du transfert de déformations géométriques lors d'un couplage de codes, nous avons mis en place une méthode d'échange de données. Elle est détaillée dans la figure 3.1 et décrite dans ce qui suit. Programmée en langage Bash, le script développé contient la séquence des opérations à exécuter.

L'étape de définition de la déformation de référence est réalisée par un code spécialisé dans le comportement mécanique. Par abus de langage, on parle de code de déformation. A partir du modèle commun, les géométries sur lesquelles seront effectués les calculs sont récupérées (❶). Elles sont ensuite maillées de façon à répondre aux besoins spécifiques de chacun des codes (❷). Un couplage est une opération itérative. En dehors des deux étapes précédentes qui ne sont réalisées qu'une seule fois, les suivantes sont répétées. Dans le cas de la recherche d'un état permanent, le couplage est stoppé lorsqu'un critère de convergence est atteint. Lors d'un calcul transitoire, le résultat dépend en plus du temps physique simulé.

La définition de la déformation est réalisée par le code de déformation qui nécessite un maillage initial et des champs correspondant aux contraintes physiques. Au démarrage du couplage, des valeurs initiales sont définies ; les étapes ❸ et ❹ ne sont pas traitées. Lorsqu'au moins une itération de couplage a été réalisée, les valeurs utilisées proviennent d'un autre code qui a effectué son calcul sur le maillage déformé. Les champs sont obtenus par l'intermédiaire du modèle commun et sont interpolés (❸) sur le maillage modifié du code de déformation, résultat de son précédent calcul. Le code de déformation réalise son calcul en s'appuyant sur son maillage initial. Il est donc nécessaire de disposer des champs interpolés sur ce dernier. Pour cela, ils sont recopiés nœud à nœud ou maille à maille (❹) sur le maillage initial et sont ainsi exploitables par le code. Cette opération est rendue possible grâce à

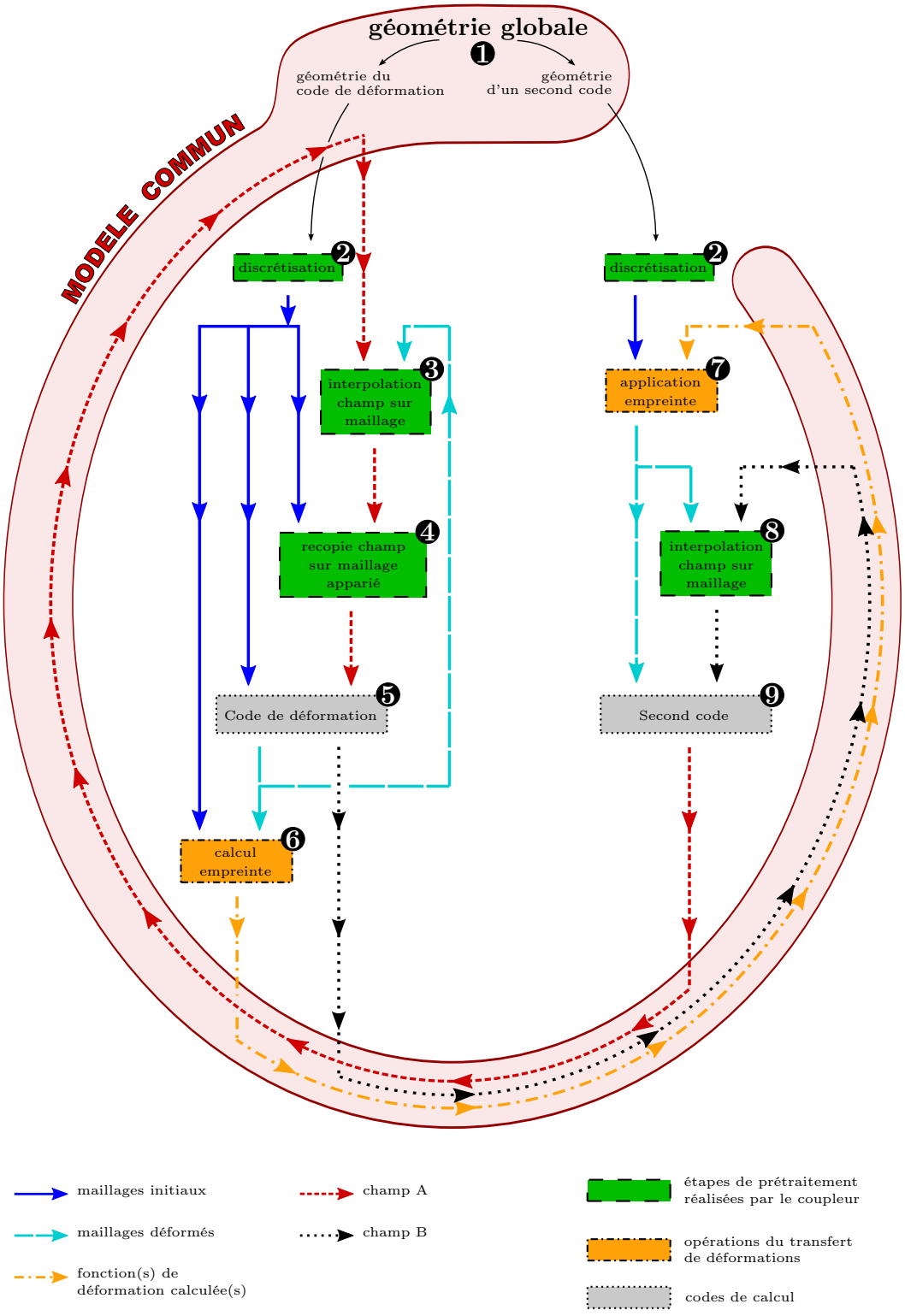


Figure 3.1 – Méthode de couplage permettant un transfert de déformations entre codes couplés

l'appariement entre les maillages initial et déformé utilisés. Une fois le calcul réalisé (❺), les résultats sont de deux ordres : le maillage modifié et un ou plusieurs champs optionnels. Afin de pouvoir transférer la modification géométrique obtenue, une empreinte de la déformation (❻) est calculée à partir du maillage de la géométrie nominale et le résultat du code de déformation. L'ensemble des informations est intégré au modèle commun. Lors du calcul d'un second code, ce dernier dispose d'un maillage initial correspondant à la discrétisation de sa géométrie nominale. Pour en obtenir une version à jour, l'empreinte de la déformation lui est appliquée (❼). Les éventuels champs utilisés sont récupérés et interpolés sur ce dernier (❽). Le code est finalement exécuté (❾) et ses résultats intégrés au modèle commun.

L'ensemble de ces différentes étapes sont effectuées itérativement jusqu'à convergence du résultat. Le critère utilisé peut être positionné à différents endroits dans cette méthode. Le test doit être réalisé sur l'évolution d'une donnée entre deux calculs d'un des deux codes, cette dernière pouvant porter sur la déformation du maillage ou d'un champ physique comme la température en tout point.

Sur la figure 3.1, les boîtes dont les bords sont représentés sous la forme de tirets correspondent aux différentes étapes de pré-traitement réalisées par le coupleur, celles en pointillés font référence aux codes de calcul et celles en traits mixtes sont les opérations du transfert de déformations. Les flèches en trait plein fin illustrent l'extraction de la géométrie utilisée par un code de calcul à partir de celle contenue dans le modèle commun. Les traits pleins font référence aux maillages initiaux, c'est-à-dire correspondant à la discrétisation de la géométrie nominale. Les maillages modifiés sont représentés par les tirets longs. Les pointillés correspondent aux champs de valeurs physiques et les traits mixtes à l'empreinte de la déformation calculées. Ces deux dernières informations sont intégrées au modèle commun.

La méthode complète mise en place permet de réaliser un couplage entre différents codes de calcul tout en permettant un transfert des déformations. Les possibilités d'application sont nombreuses, comme par exemple la simulation du comportement d'un moteur thermique, l'étude du refroidissement d'un système, l'interaction fluide-structure, ... Ces approches sont rendues envisageables grâce à l'intégration des informations d'évolutions géométriques au sein d'un modèle commun.

3.2 Architecture du modèle commun

Le modèle commun défini s'appuie sur une modélisation par entités qui permet d'apporter une réponse au besoin de faire coexister une représentation continue d'une géométrie et les paramètres d'une étude. Ce modèle ayant également le but d'être générique et évolutif, des caractéristiques et des entités doivent être mises en place et adaptées en fonction des échanges de données.

3.2.1 Intégration de données

Les données susceptibles d'être stockées dans le modèle commun sont de plusieurs ordres. Il peut s'agir d'un texte descriptif ou spécifiant une entité, de valeurs

correspondant à des constantes physiques, de champs de valeurs comme des conditions initiales, de résultats de codes de calcul, de géométries, de déformations, ... Les différents types définis sont détaillés dans ce qui suit. Pour chacun d'eux sont spécifiés ses paramètres ainsi que le mode d'écriture utilisé dans la suite du chapitre.

Texte
Tout d'abord, il doit être possible de stocker une information, un descriptif et un nom pour n'importe quel objet géométrique. Les commentaires des utilisateurs du modèle commun doivent également être pris en charge. Pour cela, il est nécessaire de stocker des textes.
Dans la suite du document, la notation utilisée sera du type "*Nom : valeur*".
Exemple : *Désignation : texte explicatif définissant une entité.*

Valeur
Une géométrie s'appuie sur des primitives définies grâce à des paramètres, comme le rayon d'une sphère ou la hauteur d'un cylindre. Pour effectuer des calculs physiques, des constantes et des variables doivent être définies. Il est donc impératif de prendre en compte les valeurs numériques au sein du modèle commun.
La notation utilisée sera "*Nom (type) : valeur*", où *type* doit être remplacé par le mot-clé "***entier***" ou "***réel***".
Exemple : *PI (réel) : 3.14*

Unité
La cohérence des unités des valeurs utilisées dans le modèle commun est très importante. Pour respecter cela, le système international (SI, [BIPM08]) est imposé. Les utilisateurs du modèle commun sont toutefois attachés à leurs unités, généralement parce qu'elles sont plus adaptées à leurs besoins. Pour rester le plus souple possible, de nouvelles unités peuvent alors être définies. Dans ce cas, une équation est requise pour connaître la conversion entre l'unité et celle présente dans le SI. Toute lettre ou groupement de lettres présent dans l'équation doit impérativement faire référence à une unité existante et précédemment définie (ceci afin d'éviter toutes boucles).
La notation sera "*Nom (équation) : description*".
Exemple 1 : *bar (*100000) : unité de pression*
Exemple 2 : *mbar (*0.001 bar) : sous-unité de pression*

Champ
L'intégration des champs est primordiale car ils doivent pouvoir être disponibles à l'ensemble des tâches s'appuyant sur le modèle commun. Dans le cas des codes de calcul, il peut s'agir de conditions initiales et de leurs résultats. Un couplage de codes nécessite obligatoirement un échange de ces données. Pour garantir la qualité du stockage et éviter les recopies, les champs sont contenus dans un fichier MED (format spécifique, performant et optimisé pour le stockage de maillages et de champs associés, *c.f.* paragraphe 2.3.2).

La notation sera : "*Nom (**champ**) : fichier_MED (nom_du_champ : pas_de_temps)*", où ***champ*** est ici un mot-clé.
Exemple : *VELOCITY (champ) : ./fichier_a_lire.med (VITESSE : 0).*

Géométrie
Les simulations numériques effectuent leurs calculs en s'appuyant sur des maillages spécifiques. Ces discrétisations sont réalisées sur une géométrie unique devant être intégrée au sein du modèle commun. Pour des raisons de précision numérique et d'évolutivité, cette dernière doit être définie de manière paramétrique et continue. Le modèle retenu est le CSG pour sa simplicité et il est stocké dans un fichier GDML (*c.f.* paragraphe 2.3.2). Des approches plus complexes et plus élaborées peuvent être envisagées, comme les *BlobTree* (*c.f.* paragraphe 2.2.1). Dans la suite du document, les géométries utilisées seront schématisées par l'arbre de construction CSG.

Déformation géométrique
De même que les champs, les modifications géométriques doivent être centralisées dans le modèle commun. Leur mise à disposition permet d'effectuer des transferts de déformations sur des maillages issus d'une géométrie identique ou adjacence à celle sur laquelle elle a été calculée. Cela ouvre ainsi la possibilité aux différentes tâches s'appuyant sur le modèle commun d'utiliser une géométrie à jour, c'est-à-dire prenant en compte les modifications. Les fonctions calculées sont stockées dans un fichier spécifique dont les données sont détaillées dans le chapitre 5.
Notation : "*Nom (**déformation**) : fichier*", où ***déformation*** est un mot-clé.
Exemple : *DILATATION (déformation) : ./fichier_a_lire.def.*

L'ensemble de ces informations doivent être intégrées au modèle commun. Pour cela, des caractéristiques sont définies.

3.2.2 Définition des caractéristiques

Les caractéristiques sont des briques d'information permettant de définir une entité. Elles sont toutes optionnelles et peuvent intervenir plusieurs fois pour une même entité. Les grandes familles de caractéristiques sont présentées :

- caractéristique paramétrique : paramètre géométrique (*e.g.* rayon, épaisseur),
- caractéristique de composition : entités utilisées (uniquement pour les entités élémentaires et complexes, *c.f.* paragraphe 3.2.4.3 et paragraphe 3.2.4.4),
- caractéristique de construction : géométrie nominale utilisant les paramètres,
- caractéristique de contrainte : contrainte sur l'entité et les paramètres (*e.g.* $rayon > 0$),
- caractéristique technique : données initiales regroupant matériau (*e.g.* composition, traitement), constante (*e.g.* valeur physique), unité utilisée (*e.g.* bar), condition limite (*e.g.* flux thermique), condition initiale (*e.g.* température), ...

- caractéristique d'évolution : donnée évoluant au cours du temps (*e.g.* déformation géométrique, résultat d'un code de calcul). Ces caractéristiques seront accompagnées d'un temps pour connaître l'ordre chronologique d'obtention.
- caractéristique informative : texte (*e.g.* nom, référence, description),
- caractéristique diverse : toute autre information jugée pertinente.

Les quatre premières caractéristiques correspondent à la géométrie nominale d'une entité. Les caractéristiques techniques comprennent les informations nominales (en dehors de la géométrie). Les matériaux utilisés sont identifiés par un code et un pourcentage de présence, le code faisant référence à des bases de données des matériaux disponibles et connus. L'accent n'est pas mis sur ce point car cette étape sort du cadre de la thèse. Il faut néanmoins signaler qu'elle centralise les informations techniques sur les matériaux et permet ainsi de s'assurer de la cohérence du modèle.

Les caractéristiques utilisées sont identifiées comme étant "optionnelle" ou "indispensable", notées "<O>" ou "<I>" respectivement ; la valeur par défaut étant "indispensable". Elles peuvent également être identifiée "unique" ("<U>"). Dans ce cas, la caractéristique sera également "indispensable" et deux entités ne pourront avoir la même valeur pour cette caractéristique.

3.2.3 Niveau de finesse

Le modèle mis en place doit pouvoir gérer différents niveaux de finesse de représentation de la géométrie. Pour s'en convaincre, prenons un exemple. Nous disposons d'un modèle complet d'une maison sur lequel s'appuient deux traitements. Le premier est un calcul du volume d'eau dans les canalisations et le second un rendu virtuel permettant à un architecte de visualiser l'intérieur de la maison (et donc les canalisations apparentes). Dans la première tâche, seuls les éléments correspondant aux canalisations sont importants. Pour effectuer son calcul, le modèle commun devra être en mesure de fournir une représentation géométrique épurée de toutes les représentations des entités autres que les canalisations. Dans le second traitement, toutes les entités doivent apparaître. Toutefois, afin d'accélérer les calculs de rendu, la représentation des canalisations pourra être sommaire. Elle pourra correspondre à une primitive cylindrique plutôt qu'au résultat de l'intersection de deux cylindres.

Pour répondre à ce besoin, les entités doivent pouvoir disposer de plusieurs représentations de leur géométrie, et donc de plusieurs caractéristiques de construction. Ces représentations doivent être rangées par ordre croissant de finesse, allant de la plus précise à la plus grossière, chaque description étant identifiée par une échelle caractéristique. Les différents arbres de construction géométrique utilisent des primitives s'appuyant impérativement sur tout ou partie des paramètres définis pour garantir la cohérence du modèle. Les primitives utilisées dans la construction peuvent être des entités précédemment créées (ce point est détaillé dans le paragraphe 3.2.4). Dans ce cas, leur niveau de finesse souhaité est spécifié ; le niveau minimum (*i.e.* le plus précis) étant la valeur par défaut. Le résultat d'une entité à plusieurs niveaux de finesse est illustré par la figure 3.10.

3.2.4 Définition des entités

La définition des entités est hiérarchique. Quatre niveaux sont définis, du plus abstrait au plus concret : entité de base, abstraite, élémentaire et complexe. Elles peuvent interagir de manière à hériter d'une autre entité de même niveau d'abstraction ou de niveau inférieur, c'est-à-dire s'appuyer sur une entité afin de disposer de l'ensemble de ses caractéristiques, de pouvoir les modifier et d'en ajouter de nouvelles. Dans ce cas, une caractéristique de contrainte est ajoutée avec comme mot-clé "ORIGINE" suivi du nom de l'entité à hériter.

3.2.4.1 Entité de base

Les entités de base sont les atomes de construction du modèle commun. Elles sont au nombre de deux. La première et la plus importante est l'entité *E_ BASE* héritée par toutes les autres (figure 3.2). Elle impose un squelette de construction, définit les caractéristiques minimales obligatoires ainsi que les données du système SI (Système International d'unités), rendant ainsi possible de spécifier les unités des dimensions renseignées pour les caractéristiques paramétriques.

Caractéristique de construction	**Caractéristiques techniques**
finesse 1 <I>	*m (*1) : unité SI de longueur*
	*kg (*1) : unité SI de masse*
Caractéristiques informatives	*s (*1) : unité SI de temps*
Nom <U>	...
Description <O>	

Figure 3.2 – Entité de base E_BASE

La seconde entité de base permet de disposer d'une manière de regrouper plusieurs entités similaires. Cela permet de définir une représentation particulière d'un groupe d'entités, agglomérant ainsi l'ensemble des détails. Nommée *E_ GROUPE* (figure 3.3), elle nécessite seulement de disposer d'au moins deux niveaux de finesse. Le premier correspond obligatoirement au niveau le plus fin de l'ensemble des entités regroupées. Les autres niveaux sont plus grossiers et peuvent reprendre l'ensemble des niveaux de chaque entité du groupement. L'héritage à l'entité *E_ BASE* étant automatique, il ne sera plus spécifié dans ce qui suit.

Caractéristiques de construction	**Caractéristique technique**
finesse 1<I>	ORIGINE E_BASE
finesse 2<I>	

Figure 3.3 – Entité de base E_GROUPE

3.2.4.2 Entité abstraite

Cette entité définit l'ensemble des propriétés d'un élément. Elle est abstraite car ne correspond pas à un objet physique en particulier, comme l'illustre l'exemple du figure 3.4 où l'entité abstraite *EA_ TUBE* pourrait aussi bien être utilisée pour définir une gaine qu'une conduite de gaz ou hydraulique. Elle dispose de l'ensemble des caractéristiques fondamentales la définissant sans pour autant connaître leur valeur. Il est donc impossible de générer une représentation de la géométrie de ce type d'entités. En ce sens, les entités de base détaillées précédemment sont des entités abstraites. Leur caractère fondamental justifie néanmoins de les mettre à part. Une entité abstraite peut se servir d'une ou plusieurs autres entités abstraites dans son arbre de construction CSG.

L'entité abstraite *EA_ TUBE* détaillée dans la figure 3.4 est générique et peut être utilisée comme une brique pour la définition de n'importe quel tube. Par exemple, il est possible de définir une nouvelle entité abstraite correspondant à une canalisation en cuivre (symbole Cu) en héritant de *EA_ TUBE*. Comme cela est illustré par la figure 3.5, des caractéristiques ont été ajoutées et d'autres ont été modifiées. Un second niveau de finesse a été ajouté de manière à pouvoir représenter une canalisation comme un tube ou comme un tube plein (*i.e.* un cylindre), en fonction des besoins ultérieurs. Bien que certaines données indispensables soient

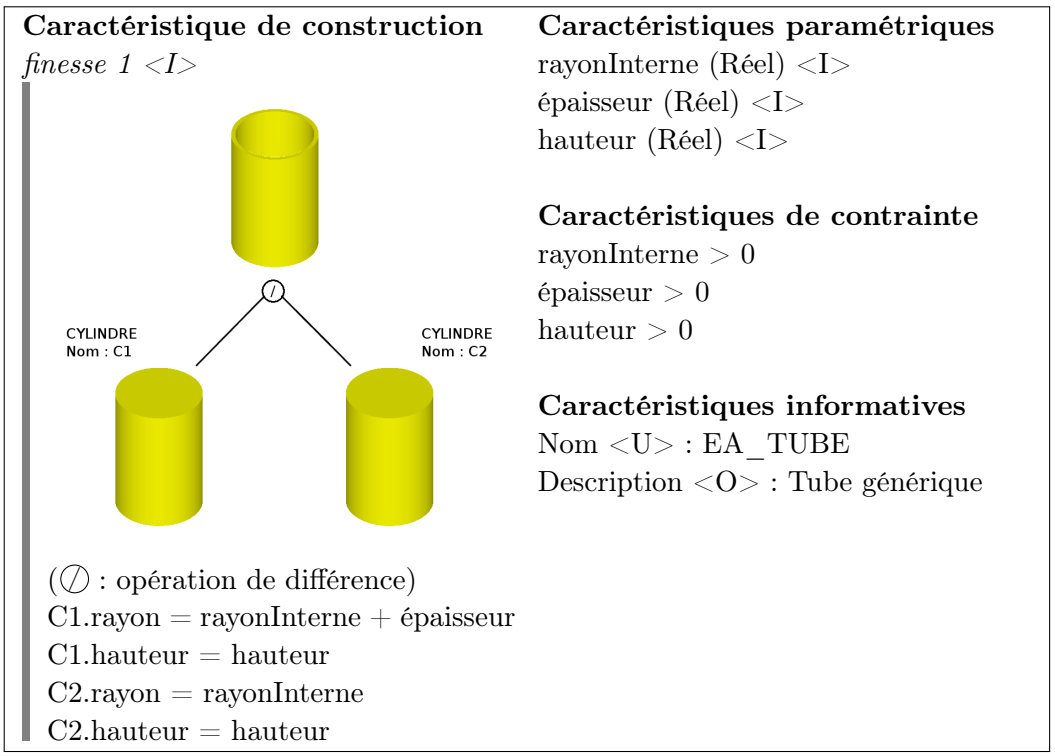
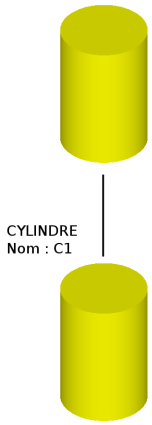


Figure 3.4 – Exemple d'entité abstraite

Caractéristique de construction
finesse 2



C1.rayon = rayonInterne
C1.hauteur = hauteur

Caractéristiques de contrainte
ORIGINE EA_TUBE
rayonInterne = $14mm$
épaisseur = $1mm$

Caractéristiques techniques
mm (/1000 m) : millimètre
Matériau : Cu (100%)

Caractéristiques informatives
Nom <U> : EA_CANALISATION
Type : CAN_XXXX
Description <O> : Canalisation en cuivre

Caractéristique diverse
Note : Pression interne pouvant dépasser 3b. Attention aux soudures.

Figure 3.5 – Exemple d'entité abstraite héritant de l'entité EA_TUBE

renseignées, comme le rayon interne et l'épaisseur, toutes ne le sont pas (*e.g.* hauteur). Cette entité est donc abstraite.

3.2.4.3 Entité élémentaire

Les entités élémentaires sont très proches des entités abstraites, à la différence que toutes les caractéristiques indispensables sont renseignées. Il est alors possible d'instancier cette entité, c'est-à-dire d'en créer une représentation physique et géométrique. Dans les différents niveaux de finesse, une entité élémentaire peut utiliser une ou plusieurs entités abstraites ou élémentaires, la seule contrainte étant de renseigner toutes les caractéristiques indispensables. En reprenant l'exemple de la canalisation en cuivre, une entité élémentaire possible est détaillée dans la figure 3.6.

Caractéristiques de contrainte
ORIGINE EA_CANALISATION
hauteur = $1m$

Caractéristique diverse
Utilisation : convient pour faire circuler de l'eau.

Caractéristiques informatives
Nom <U> : EE_CAN_100_14
Type : CAN_100_14
Référence produit : REF001

Figure 3.6 – Exemple d'entité élémentaire héritant de EA_CANALISATION

Caractéristiques de contrainte	**Caractéristiques informatives**
ORIGINE EE_CAN_100_14	Nom <U> : EE_CAN_200_14
hauteur = $2m$	Type : CAN_200_14
	Référence produit : REF002

Figure 3.7 – Exemple d'entité élémentaire héritant de l'entité EE_CAN_100_14

Toutes les caractéristiques indispensables, pré-existantes ou ajoutées, sont renseignées. L'entité est donc instanciable. Elle peut également servir de base à d'autres entités élémentaires, comme l'illustre la figure 3.7.

Les caractéristiques de composition peuvent être utilisées par les entités élémentaires. Comme l'illustre la figure 3.8, elles permettent de connaître les entités instanciées utilisées par l'entité courante. Il est ainsi possible de disposer de leurs caractéristiques et de séparer leurs géométries.

3.2.4.4 Entité complexe

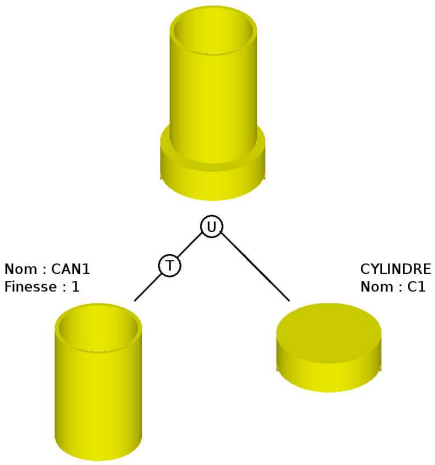
Une entité complexe est une entité élémentaire particulière. Elle correspond au résultat d'une construction faisant intervenir uniquement des entités instanciables, qu'elles soient complexes ou élémentaires. Un modèle commun est finalement une entité complexe contenant l'ensemble des parties à modéliser. Reprenons l'exemple de la maison et de ses canalisations.

La figure 3.9 illustre le groupement de toutes les entités de canalisation présentes dans le futur modèle commun. Les différentes entités élémentaires sont ainsi réunies au sein d'une nouvelle héritant de l'entité de base E_GROUPE. De cette manière, il peut être plus facile de gérer de manière globale un ensemble d'entités qui peuvent avoir une signification proche.

On suppose disposer d'une entité complexe nommée *EC_MAISON* regroupant toutes les entités définissant la maison : murs, toits, fenêtres, sols, ... Le modèle commun, l'entité complexe finale, répondant aux besoins exposés dans le paragraphe 3.2.3 pourrait alors être celui présenté dans la figure 3.10.

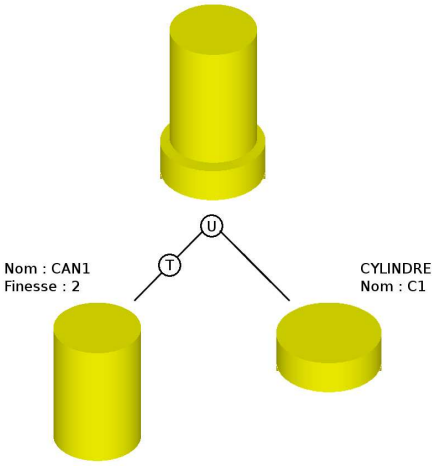
Grâce à une telle entité complexe, il est possible de demander différentes représentations de la scène en fonction des besoins des traitements à effectuer. Le calcul du volume d'eau dans les canalisations utilisera la représentation géométrique de l'entité en spécifiant la finesse 3, celle correspondant uniquement aux canalisations définis précisément. Le second traitement effectuant le rendu de la scène sera réalisé avec la finesse 2. La maison entière est modélisée et les éléments de faible taille, comme les canalisations, seront dans ce cas représentés de manière grossière pour accélérer la phase de rendu. Le modèle ainsi défini permet de gérer différentes représentations pour un modèle commun. Il permet également d'y lier des informations de manière à pouvoir les localiser dans l'espace en les attachant aux entités et dans le temps en spécifiant un pas de temps.

Caractéristiques de construction
finesse 1 <I>



(Ⓤ : opération d'union)
(Ⓣ : opération de translation)
C1.rayon = CAN1.rayonInterne + $10mm$
C1.hauteur = $5mm$

finesse 2 <I>



C1.rayon = CAN1.rayonInterne + $10mm$
C1.hauteur = $5mm$

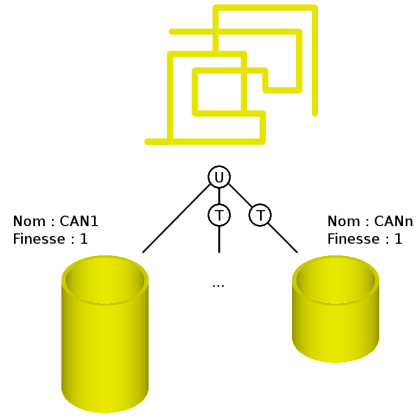
Caractéristique de composition
EE_CAN_200_14 CAN1

Caractéristiques informatives
Nom <U> : EE_CAN_BOUCHEE
Type : CAN_200_14B
Référence produit : REF003

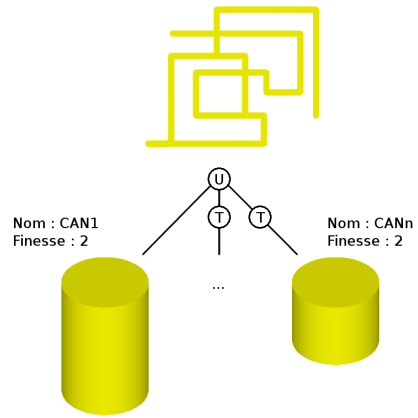
Figure 3.8 – Exemple d'entité élémentaire utilisant une autre entité élémentaire

Caractéristiques de construction

finesse 1 <I>



finesse 2 <I>



Caractéristique de contrainte

ORIGINE E_GROUPE

Caractéristiques de composition

EE_CAN_200_14 CAN1

...

EE_CAN_100_14 CANn

Caractéristiques informatives

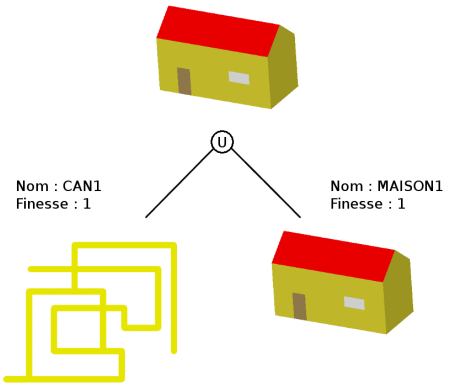
Nom <U> : EC_CANALISATIONS

Description <O> : Groupement de l'ensemble des entités de canalisations du modèle

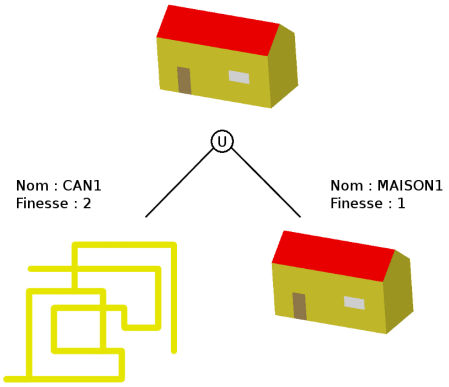
Figure 3.9 – Exemple d'entité complexe de groupement

Caractéristiques de construction

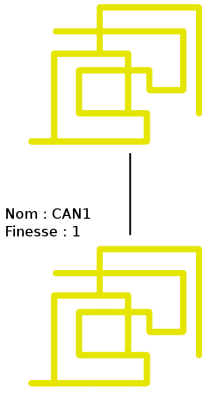
finesse 1 <I>



finesse 2 <I>



finesse 3



Caractéristiques de composition

EC_MAISON MAISON1
EC_CANALISATIONS CAN1

Caractéristiques informatives

Nom <U> : SCENE
Description <O> : Modélisation complète de la scène

Figure 3.10 – Exemple d'entité complexe correspondant au modèle commun final

3.3 Modèle commun mis en place

L'implémentation du modèle commun est une tâche lourde et complexe qui n'a pu être effectuée par manque de temps. Afin de valider la méthode de couplage permettant le transfert de déformations, un premier travail est réalisé. Le modèle commun est spécifié en langage python et s'appuie sur la plateforme SALOME (*c.f.* paragraphe 2.1.3). La géométrie commune aux différents codes couplés (❶ sur la figure 3.1) est définie avec l'aide du composant GEOM. Elle s'appuie sur une construction CSG qui peut s'apparenter à celle proposée précédemment ; la principale différence étant de ne pas pouvoir gérer différents niveaux de finesse. La géométrie complète est ensuite décomposée en fonction des calculs à réaliser. Deux maillages différents sont générés en utilisant le composant SMESH, soit à partir d'une même géométrie, soit à partir de géométries adjacentes (❷ sur la figure 3.1). De cette manière, on s'assure de la cohérence des résultats et on certifie que les maillages sont bien issus d'une géométrie commune. En l'état actuel, il n'est toutefois pas possible d'intégrer les champs de données et les déformations dans le modèle commun. Afin de permettre des échanges entre codes, l'approche choisie est d'utiliser des transferts par fichiers. Le coupleur impose aux différents codes un format spécifique, le format standard MED. MED est supporté par de divers codes de calcul et permet d'intégrer des champs de différents types. Les empreintes des modifications géométriques sont quant à elles regroupées dans un format simplifié (un fichier texte) visant à permettre le couplage. De cette manière, le coupleur mis en place est à même de réaliser les différents échanges de données (représentés par les trois flèches intégrées au modèle commun sur la figure 3.1). Il est ainsi possible de réaliser un couplage fonctionnel où les déformations géométriques sont prises en compte.

3.4 Synthèse

Un couplage de codes faisant intervenir des transferts de déformations géométriques impose la mise en place d'une nouvelle méthode d'échange de données. Cette dernière s'appuie sur un modèle commun qui permet d'assurer la cohérence des calculs et des informations partagées. Il offre également la possibilité de mettre à disposition et de centraliser les résultats obtenus. Pour réaliser cela, le formalisme nécessaire au développement et à l'échange d'un modèle par entités a été mis en place. Des caractéristiques et des entités ont été définies afin de permettre l'intégration de données variées. Les tâches s'appuyant sur le modèle commun pouvant être de nature très différentes, l'approche mise en œuvre permet d'obtenir des géométries particulières, répondant à un besoin spécifique. Le format de stockage choisi pour stocker le modèle commun s'appuie sur le format GDML (*c.f.* paragraphe 2.3). Ce dernier étant spécifique à un cas d'application, il doit être étendu pour permettre la sauvegarde de l'ensemble des informations.

Un premier travail de développement du modèle commun a été réalisé. Bien que sommaire, il permet de valider les choix réalisés concernant la méthode de couplage

mise en place. Il est ainsi possible d'effectuer un couplage de codes de calcul tout en prenant en compte les évolutions de la géométrie. Il reste toutefois à détailler les étapes liées au transfert des déformations (❻ et ❼ sur la figure 3.1). Des études préliminaires sont réalisées et détaillées dans le chapitre suivant.

Chapitre 4

Etudes préliminaires

Sommaire

4.1 Protocole d'évaluation mis en place 61
4.2 Interpolation linéaire 62
4.3 Fonctions de base radiale 66
4.3.1 RBF multiquadrique 67
4.3.2 RBF Thin-Plate Spline 71
4.3.3 CS-RBF 73
4.4 Synthèse 76

Dans le chapitre 2, nous avons souligné que des méthodes existantes de transfert de déformations permettent de répondre partiellement à nos attentes. Les approches retenues sont l'interpolation linéaire et les fonctions de base radiale multiquadrique, TPS et CS-RBF. Nous proposons dans ce chapitre de les étudier afin d'avoir une première idée du résultat que l'on pourrait obtenir en les utilisant. Les tests sont effectués sur des maillages surfaciques.

4.1 Protocole d'évaluation mis en place

Afin de qualifier les résultats présentés dans ce chapitre, nous proposons dans ce paragraphe un protocole d'évaluation schématisé par la figure 4.1. On se place dans le cas du transfert de déformations entre des maillages s'appuyant sur une même géométrie. On note M_{1i} (i pour *initial*) le maillage initial sur lequel est

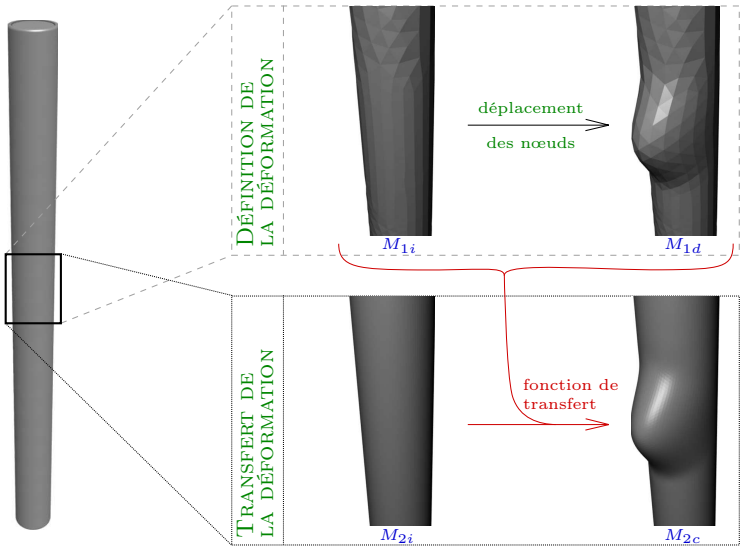
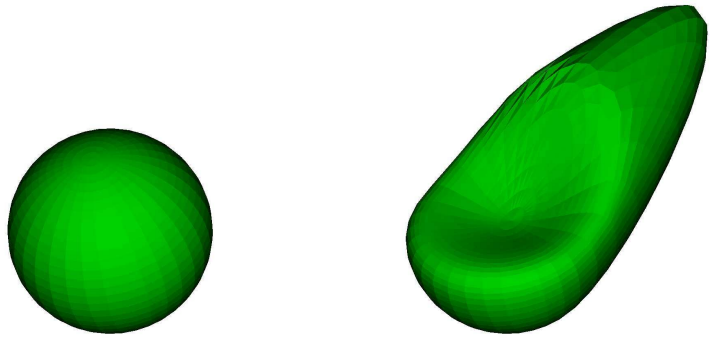


Figure 4.1 – Protocole d'évaluation des résultats des études préliminaires

calculé la modification géométrique. M_{1d} (d pour *déformé*) correspond au maillage M_{1i} modifié, où certains nœuds ont été déplacés lors de la phase de définition de la déformation, typiquement, le résultat d'un code de comportement mécanique. Les deux maillages M_{1i} et M_{1d} sont appariés. Le maillage initial que l'on souhaite modifier est noté M_{2i} et s'appuie sur la même géométrie que M_{1i}. Le maillage résultat de l'application sur M_{2i} de l'empreinte prise sur M_{1i} et M_{1d} est noté M_{2c} (c pour *calculé*). Cette notation est utilisée dans l'ensemble du document.

Dans ce qui suit, les résultats de chaque méthode étudiée sont illustrés par des schémas. Les calculs sont réalisés sur un ordinateur PC $64bits$ disposant de $4Go$ de mémoire vive, un seul cœur cadencé à $1.6GHz$ est utilisé. Les tests sont effectués sur le maillage d'une simple sphère et l'étape de définition de la déformation a été faite manuellement. Les nœuds de M_{1i} ont été déplacés en s'appuyant sur l'outil Blender ([Blender]) pour aboutir à M_{1d}. Ces deux maillages contiennent 1 000 nœuds et

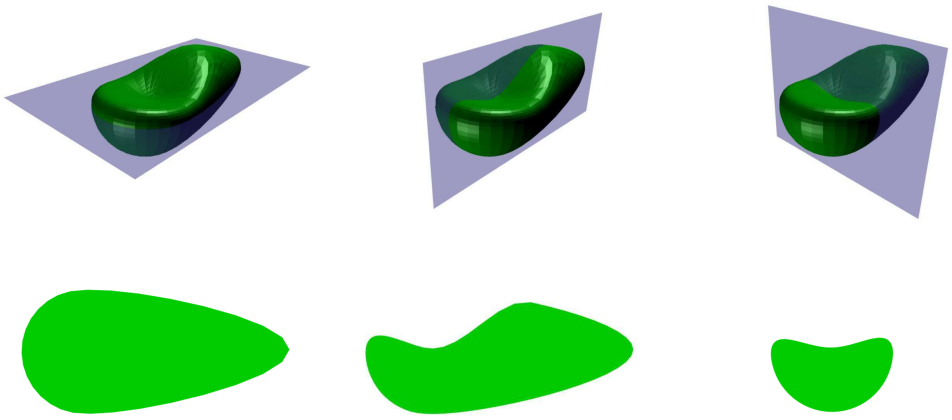
2 000 faces triangulaires chacun, et le résultat est illustré par la figure 4.2. Le



(a) M_{1i} (b) M_{1d}

Figure 4.2 – Maillages M_{1i} et M_{1d} de la phase de définition de la déformation

maillage M_{1i} est une discrétisation d'une sphère. Il a été allongé globalement pour prendre la forme d'un œuf et modifié localement (*i.e.* le creux). Les illustrations regroupées dans la figure 4.3 montrent la forme du résultat obtenu, à savoir le maillage M_{1d}.



(a) sur le plan XY (b) sur le plan YZ (c) sur le plan XZ

Figure 4.3 – Maillage M_{1d} en fonction de trois plans de coupe

4.2 Interpolation linéaire

Le principe de la méthode d'interpolation linéaire se décompose en trois étapes (figure 4.4). Elles sont réalisées pour l'ensemble des nœuds du maillage initial à modifier M_{2i}. L'ensemble de ces étapes repose sur le fait que les maillages M_{1i} et

M_{1d} sont constitués uniquement de triangles. Dans le cas où ce ne serait pas le cas, les maillages sont préalablement triangulés. Détaillons la méthode en prenant un point $v \in M_{2i}$ et en cherchant à connaître sa nouvelle position dans le maillage calculé M_{2c}.

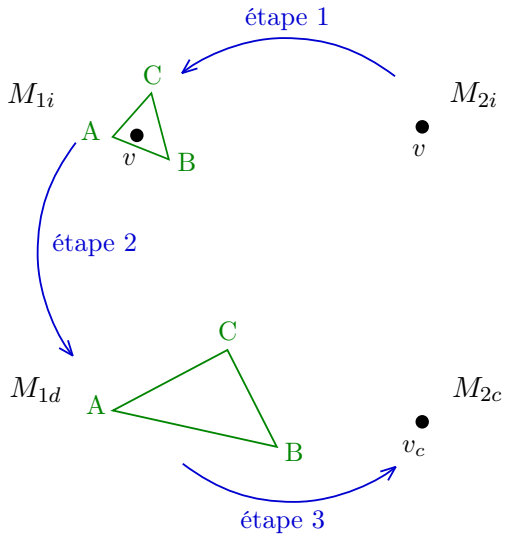


Figure 4.4 – Schéma de la méthode d'interpolation linéaire

Etape 1
Tout d'abord, le point $v \in M_{2i}$ est localisé dans M_{1i}. Les maillages M_{1i} et M_{2i} sont tous deux des discrétisations de la même géométrie. En faisant une recherche d'appartenance du point v aux différentes faces de M_{1i}, il est possible de localiser v dans M_{1i}. Toutefois, il peut arriver qu'aucune face ne contienne ce point lorsque M_{1i} et M_{2i} discrétisent une géométrie courbe avec une finesse différente. Dans ce cas, le projeté de v sur M_{1i} est recherché et la face dans laquelle il se trouve localisée.

Etape 2
Connaissant cette face dans M_{1i}, elle est recherchée dans le maillage déformé M_{1d}. Les nœuds des maillages M_{1i} et M_{1d} étant appariés, ce travail consiste à sélectionner, dans M_{1d}, les nœuds du maillage M_{1i} qui composent la face précédemment trouvée. De cette manière, la même face est bien sélectionnée mais la position de ses sommets peut avoir été modifiée.

Etape 3
Pour connaître la position finale du point v après modification, il faut pouvoir le positionner correctement dans la face de M_{1d}. Pour cela, les coordonnées barycentriques du point v (ou de son projeté) dans la face du maillage non déformé M_{1i} sont calculées, de manière à définir sa position par rapport aux trois sommets qui composent le triangle. Le calcul de la position du point v_c est réalisé en utilisant

ces mêmes coordonnées barycentriques appliquées aux points du triangle dans le maillage M_{1d}.

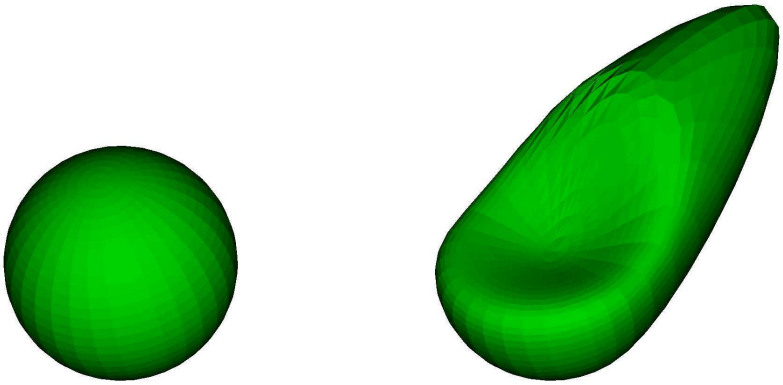
Afin d'accélérer la recherche de la face lors de l'étape 1, un octree a été mis en place pour partitionner l'espace et restreindre les faces pouvant contenir le point traité. Ce choix est motivé par sa simplicité de mise en place et le gain de temps apporté. De cette manière, la méthode est performante en temps de calcul et, en dehors de l'octree, ne nécessite pas d'espace en mémoire. Le tableau 4.1 détaille le temps de calcul de la méthode en fonction du nombre de nœuds dans le maillage M_{2i}. La phase de précalcul correspond à la création et au remplissage de l'octree. Les valeurs de cette colonne sont toutes identiques car, pour un maillage M_{1i} donné, cette étape est la même quel que soit le maillage M_{2i}. La phase de déplacement effectue la recherche de la nouvelle position de l'ensemble des points de M_{2i}. Il faut alors noter que le temps de calcul dépend faiblement du nombre de nœuds dans le maillage M_{1i}. Ceci est dû à l'utilisation de l'octree qui restreint l'espace de recherche à une zone plus petite, réduisant ainsi les calculs à effectuer. Le temps moyen de calcul du déplacement d'un point varie en fonction du nombre de nœuds dans le maillage M_{2i}. Il est toutefois possible d'estimer, dans le cas où M_{1i} possède 2 000 nœuds, que la moyenne est de $\simeq 1.0ms$ par nœud. En extrapolant, le temps de calcul total pour un maillage M_{2i} contenant un million de nœuds serait alors proche de $\simeq 15min$.

Nb nœuds dans M_{1i}	Nb nœuds dans M_{2i}	Temps de calcul		
		total	précalcul	déplacement
1 000	90	0,5s	0,4s	0,1s
1 000	380	0,7s	0,2s	0,5s
1 000	1 000	1,0s	0,2s	0,8s
1 000	9 900	8,6s	0,2s	8,4s
1 000	39 800	34,0s	0,2s	33,8s
2 000	90	0,8s	0,6s	0,2s
2 000	380	0,8s	0,4s	0,4s
2 000	1 000	2,0s	0,5s	1,5s
2 000	9 900	12,3s	0,6s	11,7s
2 000	39 800	38,0s	0,5s	37,5s

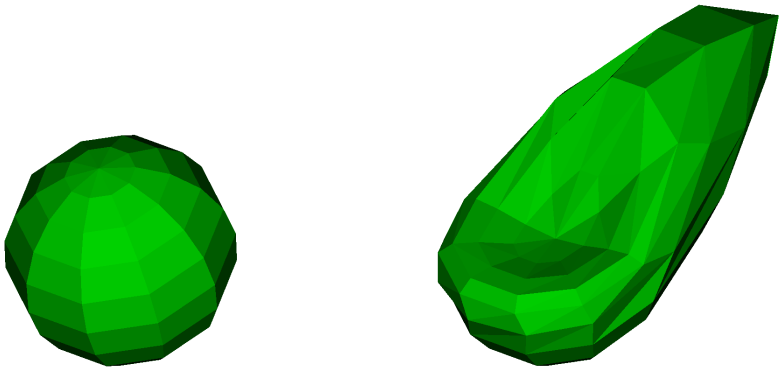
Table 4.1 – Détail des temps de calcul de la méthode d'interpolation linéaire

Les temps de calcul obtenus pourraient être améliorés en définissant plus finement le pas de l'octree. De plus, la méthode peut être facilement parallélisable.

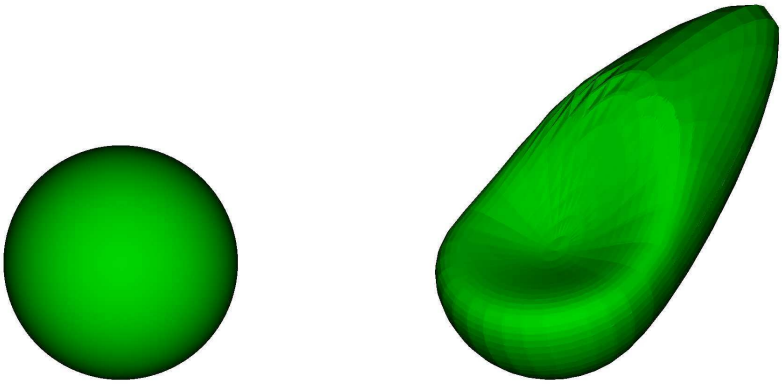
Les résultats obtenus sont regroupés dans la figure 4.5. Comme on peut le constater, le résultat est visuellement de bonne qualité lorsque le maillage M_{2i} utilisé est plus grossier que le maillage M_{1i}. Aucun artefact n'apparait et la déformation transférée correspond à celle survenue sur M_{1i} pour aboutir à M_{1d} (figure 4.5(b)). Dans le cas où le maillage M_{2i} utilisé est plus fin que le maillage M_{1i}, le résultat obtenu est moins convainquant. Le principe de la méthode étant de projeter tous les nœuds



(a) M_1 contenant 1 000 nœuds (initial à gauche, déformé à droite)



(b) M_2 plus grossier que M_1 contenant 90 nœuds (initial à gauche, déformé par interpolation linéaire à droite)



(c) M_2 plus fin que M_1 contenant 39 800 nœuds (initial à gauche, déformé par interpolation linéaire à droite)

Figure 4.5 – Résultats de la méthode d'interpolation linéaire

de M_{2i} sur les triangles du M_{1i}, le transfert de la déformation a pour effet d'aboutir à un maillage M_{2c} de même qualité que M_{1d} (figure 4.5(c)).

Cette méthode est donc performante en terme de temps de calcul et de place mémoire utilisée. D'après les résultats obtenus, elle est intéressante lorsque le maillage sur lequel transférer la déformation est plus grossier que celui sur lequel la modification a eu lieu, et décevante dans le cas contraire. Il faut également ajouter qu'il n'est pas possible de calculer une fonction caractérisant la déformation. Son intégration au sein du modèle commun oblige donc de stocker l'ensemble des maillages M_{1i} et M_{1d}. Pour toutes ces raisons, cette approche est mise de côté.

4.3 Fonctions de base radiale

La seconde approche envisagée pour effectuer un transfert de déformations s'appuie sur un réseau de RBF. Rappelons que la forme d'une telle fonction est :

$$F(v) = \sum_{k=1}^{n} \lambda_k \ \phi(\|v - L_k\|_2) \quad + \mathcal{L}(v) \tag{16}$$

Cette partie s'attache à montrer les différences de résultat entre les trois RBF ϕ retenues dans le paragraphe 2.4, à savoir la multiquadrique, la TPS et la CS-RBF.

L'intégration d'un réseau de RBF dans le modèle commun, quelle que soit la RBF utilisée, oblige à stocker l'ensemble des paramètres de cette fonction. Pour le cas général, il faut mémoriser l'ensemble des points caractéristiques L ainsi que les coefficients λ du réseau.

Le principe de la méthode utilisant un réseau de RBF se décompose en deux étapes (figure 4.6). La première, réalisée une seule fois, permet de caractériser la déformation en définissant la fonction RBF F. La seconde calcule la nouvelle position d'un point par application de F. Elle est effectuée autant de fois qu'il y a de nœuds dans le maillage initial à modifier M_{2i}. Détaillons la méthode en prenant un point $v \in M_{2i}$ et en cherchant à connaître sa nouvelle position dans le maillage calculé M_{2c}.

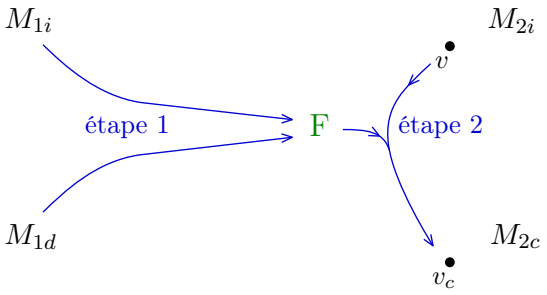


Figure 4.6 – Schéma de la méthode utilisant un réseau de RBF

Etape 1

On impose au réseau de RBF la contrainte d'interpolation suivante (avec n le

nombre de nœuds dans le maillage M_{1i}) :

$$\forall j \in [0; n], v_j \in M_{1i}, v'_j \in M_{1d}, F(v_j) = v'_j \tag{17}$$

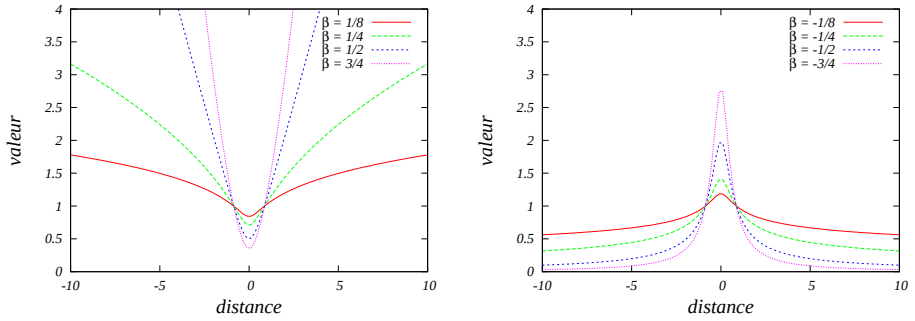
De cette manière, il est possible de calculer les coefficients λ du réseau en résolvant le système d'équations linéaires induit. La fonction F correspond ainsi à une empreinte de la déformation entre M_{1i} et M_{1d}.

Etape 2
F est ensuite utilisée pour connaître la position finale d'un point $v \in M_{2i}$ après modification. Ce dernier est noté $v_c \in M_{2c}$ et s'obtient de la façon suivante : $F(v) = v_c$.

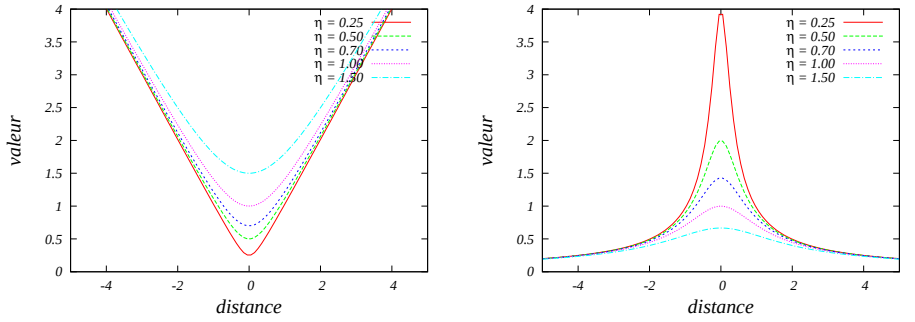
Les résultats diffèrent en fonction de la RBF choisie. La multiquadrique, la TPS et la CS-RBF font donc l'objet de tests dans les parties qui suivent.

4.3.1 RBF multiquadrique

La RBF multiquadrique est une fonction de base radiale d'équation $\phi(r) = (r^2 + \eta^2)^\beta$. Les deux paramètres, η et β, influencent le résultat. β est généralement fixé à $1/2$ ou $-1/2$ mais le choix de la valeur de η est toujours un problème ouvert. Leur influence est mise en évidence par la figure 4.7 (figure 4.7(a)



(a) $\eta = 0.5$ ($\beta > 0$ à gauche, $\beta < 0$ à droite)



(b) β constant ($\beta = 1/2$ à gauche, $\beta = -1/2$ à droite)

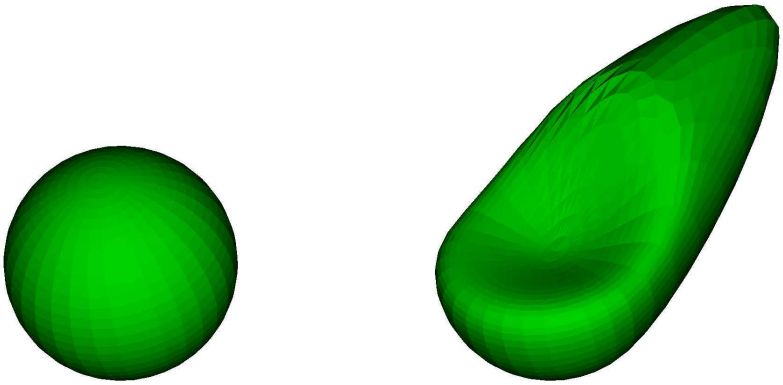
Figure 4.7 – Influence des paramètres η et β sur la fonction multiquadrique

pour η constant, figure 4.7(b) pour β constant). Ces graphiques montrent que les valeurs des deux paramètres modifient grandement la forme de la fonction multiquadrique utilisée. Plus β est proche de 0, plus la courbe est plate. Lorsque $\beta > 0$, plus la valeur est grande, plus la courbe tend rapidement vers $+\infty$. Lorsque $\beta < 0$, plus la valeur est faible, plus la courbe forme un pic. Le paramètre β permet ainsi de gérer le comportement de la déformation autour d'un point caractéristique. Pour η, plus sa valeur est faible, plus la courbe devient pointue (pour $\beta > 0$) ou forme un pic raide (pour $\beta < 0$). Ce paramètre permet de gérer la zone d'influence d'un point caractéristique. Il est possible de supposer que le résultat sera de meilleure qualité avec une zone d'influence plus grande pour chaque point caractéristique. De même, un comportement autour de ces points de type gaussien ou parabolique devrait améliorer la qualité. D'après cette première analyse, on peut estimer aboutir à un résultat lisse avec η grand et β proche de 0 si $\beta < 0$, ou avec η et β grands si $\beta > 0$. Afin de montrer les différences de résultats obtenus en fonction de ces deux paramètres, des tests ont été effectués avec un maillage M_{2i} plus fin que M_{1i}. Les valeurs choisies pour β sont celles couramment utilisées, à savoir $1/2$, $1/4$, $-1/2$ et $-1/4$ (figure 4.8). Celles concernant η ont été définies dans le but de mettre en évidence son influence (figure 4.9). Dans tous les cas, la forme est globalement respectée car les déformations de premier ordre sont contenues dans la fonction polynomiale $\mathcal{L}$ du réseau de RBF. Toutefois, on peut constater l'influence des paramètres β et η sur le résultat. Comme attendu, lorsque η est faible, le résultat contient de nombreux pics. Il en est de même lorque β s'éloigne de 0 pour $\beta < 0$, ou s'en rapproche pour $\beta > 0$. Ces effets s'estompent lorsque la valeur du paramètre η augmente et que β se rapproche de 0 si $\beta < 0$, ou s'en éloigne sinon. Dans le cas de l'application de cette méthode sur un maillage M_{2i} plus grossier que M_{1i}, les conclusions sont identiques. Le résultat varie en fonction des paramètres : globalement, la déformation sera respectée mais localement, la position des nœuds ne sera pas toujours la même.

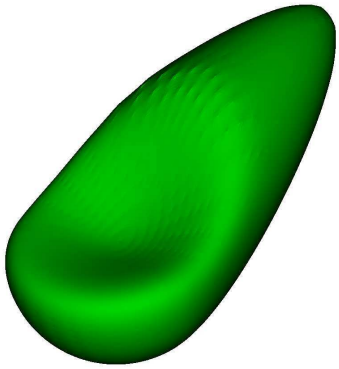
Le tableau 4.2 regroupe les temps de calcul. La phase de précalcul correspond

Nb nœuds dans M_{1i}	Nb nœuds dans M_{2i}	Temps de calcul		
		total	précalcul	déplacement
1 000	90	5,8s	5,6s	0,2s
1 000	380	6,4s	5,7s	0,7s
1 000	1 000	7,6s	5,9s	1,7s
1 000	9 900	22,5s	5,5s	17,0s
1 000	39 800	1min 13,5s	5,7s	1min 07,8s
2 000	90	31,5s	31,0s	0,5s
2 000	380	32,0s	31,0s	1,0s
2 000	1 000	34,7s	31,3s	3,4s
2 000	9 900	1min 04,0s	31,4s	32,6s
2 000	39 800	2min 41,0s	31,3s	2min 10,7s

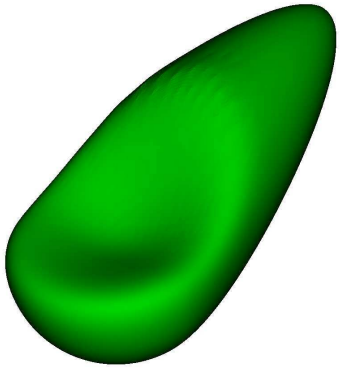
Table 4.2 – Détail des temps de calcul de la méthode utilisant une multiquadrique



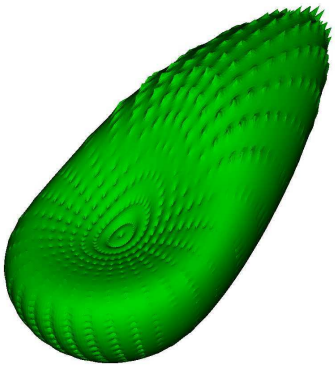
(a) M_1 contenant 1 000 nœuds (initial à gauche, déformé à droite)



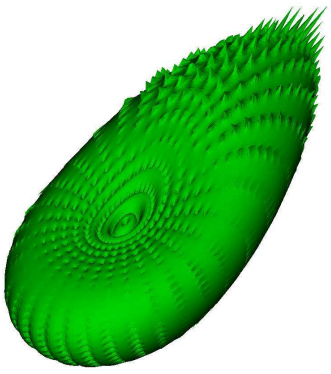
(b) $\beta = 0.25$



(c) $\beta = 0.5$

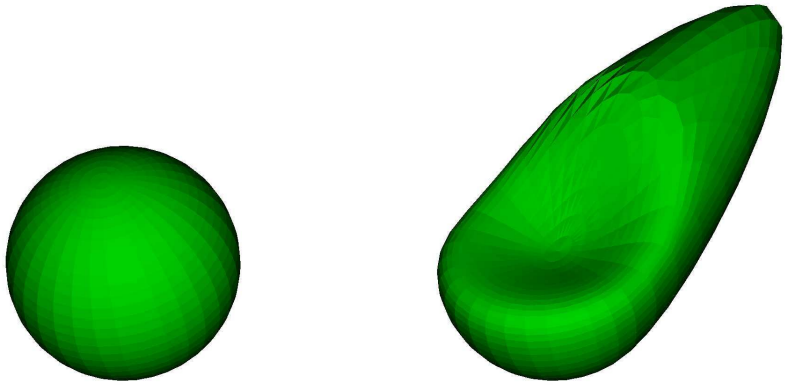


(d) $\beta = -0.25$

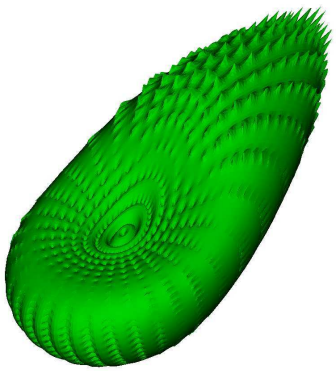


(e) $\beta = -0.5$

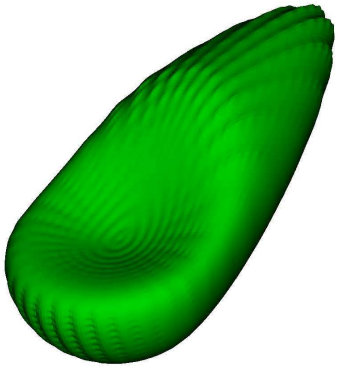
Figure 4.8 – Résultats de la méthode RBF multiquadrique en fonction du paramètre β ($\eta = 0.0062$ constant, M_{2i} plus fin que M_1 contenant 39 800 nœuds)



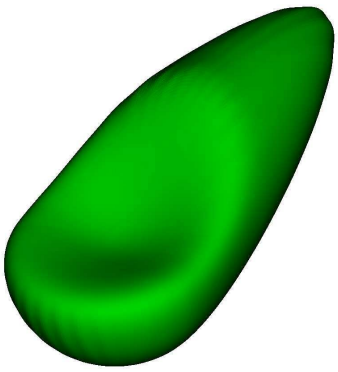
(a) M_1 contenant 1 000 nœuds (initial à gauche, déformé à droite)



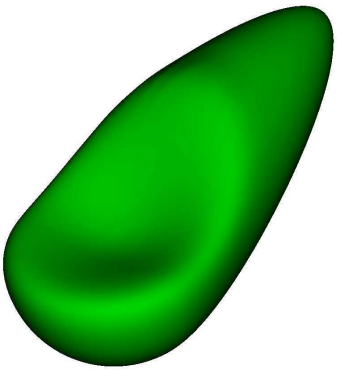
(b) $\eta = 0.0125$



(c) $\eta = 0.05$



(d) $\eta = 0.1$



(e) $\eta = 0.2$

Figure 4.9 – Résultats de la méthode RBF multiquadrique en fonction du paramètre η ($\beta = -0.5$ constant, M_{2i} plus fin que M_1 contenant 39 800 nœuds)

à la résolution du système d'équations linéaires dans le but d'obtenir les coefficients λ du réseau de RBF. Elle est réalisée par une élimination directe de Gauss. La seconde étape calcule le déplacement de l'ensemble des nœuds du maillage M_{2i}. Comme cela a été détaillé dans le paragraphe 2.4.1.3, on constate que les temps de calcul sont très dépendants du maillage M_{1i}. Durant la phase de précalcul, la place mémoire requise augmente en $O(n^2)$ et le temps de calcul en $O(n^3)$ en fonction de n, le nombre de points dans le maillage M_{1i}. La seconde étape est linéaire, avec une moyenne de $3.3ms$ par nœud pour un maillage M_{1i} contenant 2 000 nœuds. Ainsi, un maillage M_{2i} contenant un million de points serait modifié en une heure.

Les paramètres β et η sont difficiles à définir de manière à obtenir un résultat lisse. β peut être arbitrairement imposé en choisissant une valeur usuelle, toutefois, la valeur de η est très liée au maillage M_{1i}. Enfin, les temps de calcul de la méthode sont conséquents et augmentent de façon importante en fonction du nombre de points dans les maillages utilisés. Cette approche est écartée pour toutes ces raisons.

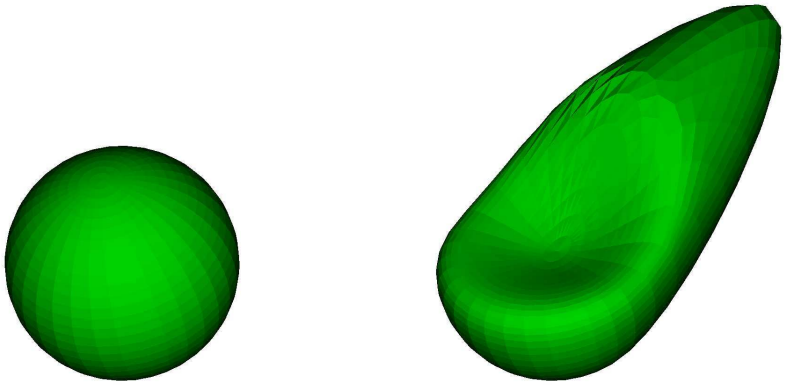
4.3.2 RBF Thin-Plate Spline

La fonction de base radiale TPS est de la forme : $\phi(r) = r^{2k}.log(r)$. D'après [BCZD02], le paramètre k est utilisé pour spécifier la continuité de la fonction ϕ qui est C^{2k-1} continue. Si $k = 1$, la RBF est C^1-continue, si $k = 2$, elle est C^3-continue. Dans ce qui suit, nous imposons que la TPS soit C^2, ce qui induit comme forme de fonction $\phi(r) = r^4.log(r)$. La fonction ne possède donc aucun paramètre.

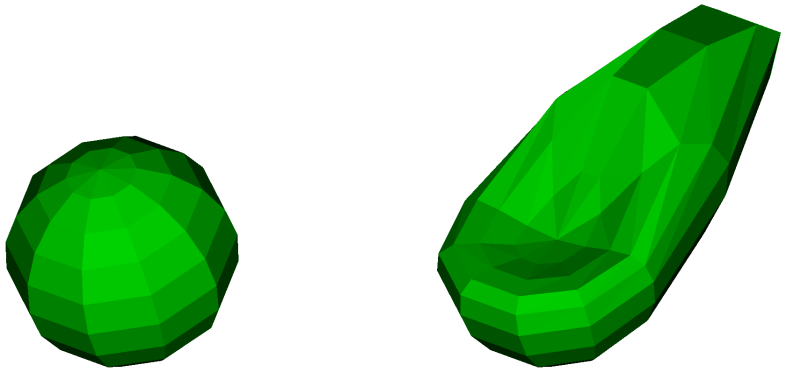
Les maillages obtenus en utilisant un réseau de TPS sont regroupés dans la figure 4.10. On constate alors les bons résultats de la méthode, autant sur M_{2i} plus grossier que plus fin que M_{1i}. Les temps de calcul sont détaillés dans le tableau 4.3. La première phase correspond à la résolution du système d'équations linéaires réalisée par une élimination directe de Gauss, la seconde calcule le déplacement de l'ensemble des nœuds de M_{2i}. Comme précédemment, durant la phase de précalcul, la place mémoire requise augmente en $O(n^2)$ et le temps de calcul en $O(n^3)$ en

Nb nœuds dans M_{1i}	Nb nœuds dans M_{2i}	Temps de calcul		
		total	précalcul	déplacement
1 000	90	2,7s	2,6s	0,1s
1 000	380	3,0s	2,6s	0,4s
1 000	1 000	3,3s	2,3s	1,0s
1 000	9 900	10,7s	2,3s	8,3s
1 000	39 800	41,6s	2,6s	39,0s
2 000	90	16,3s	16,1s	0,2s
2 000	380	17,0s	16,2s	0,8s
2 000	1 000	18,0s	16,4s	1,6s
2 000	9 900	34,0s	16,4s	17,6s
2 000	39 800	1min 23,5s	16,2s	1min 07,3s

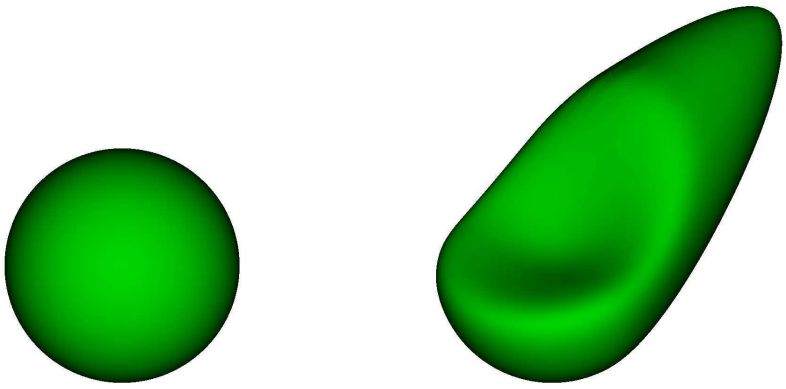
Table 4.3 – Détail des temps de calcul de la méthode utilisant une TPS



(a) M_1 contenant 1 000 nœuds (initial à gauche, déformé à droite)



(b) M_2 plus grossier que M_1 contenant 90 nœuds (initial à gauche, déformé par TPS à droite)



(c) M_2 plus fin que M_1 contenant 39 800 nœuds (initial à gauche, déformé par TPS à droite)

Figure 4.10 – Résultats de la méthode utilisant les TPS

fonction de n, le nombre de points dans M_{1i}. La seconde étape est linéaire, avec une moyenne de $1.7ms$ par nœud pour un maillage M_{1i} contenant 2 000 nœuds. Ainsi, un maillage M_{2i} contenant un million de points serait modifié en $30min$.

La forte augmentation du temps de la phase de précalcul s'explique par le fait que le calcul implique de travailler avec une matrice dense, symétrique mais à diagonale nulle. Ces inconvénients sont dus à deux facteurs inhérents aux TPS. La fonction TPS ϕ utilisée est nulle en 0 (par prolongement par continuité en zéro) et 1. Elle est négative sur $]0;1[$ et croissante sur $[1;+\infty[$. La matrice Ψ ne contient que très peu de zéros : la diagonale est nulle car $\phi(\|L_j - L_j\|_2) = \phi(0) = 0$ et il est très peu probable que $r = \|L_j - L_k\|_2$ soit exactement égal à 1 ($\phi(1) = 0$). Il faut également signaler que Ψ est symétrique, car :

$$\forall\ 1 \leq j,k \leq n,\ \phi(\|L_j - L_k\|_2) = \phi(\|L_k - L_j\|_2) \tag{18}$$

où n est le nombre de points dans le maillage M_{1i}. Toutefois, elle n'est pas définie positive car les éléments de sa diagonale sont nuls. Ainsi, la matrice est dense et symétrique non définie positive à diagonale nulle. Il est donc difficile de trouver une méthode performante permettant de résoudre le système d'équations linéaires.

En utilisant un réseau de TPS, aucun paramètre n'est à spécifier. Le maillage final est lisse mais le temps de calcul est non négligeable et augmente considérablement en fonction du nombre de nœuds dans le maillage M_{1i}.

4.3.3 CS-RBF

La CS-RBF est une fonction de base radiale à support compact. Il est alors possible d'utiliser des solveurs itératifs performants permettant de réduire les temps de calcul. La forme de la CS-RBF dépend de la dimension de l'espace dans lequel se trouvent les maillages. Pour $\mathbb{R}^3$:

$$\phi(r) = \begin{cases} (1-r)^4(4r+1) & \text{si} \quad r < 1 \\ 0 & \text{sinon} \end{cases} \tag{19}$$

Pour adapter la CS-RBF aux maillages utilisés, il est courant de redéfinir le support de la fonction en utilisant $\phi^s(r) = \phi(r/s)$, où s est la nouvelle taille. Ce paramètre permet de définir la zone d'influence autour des points caractéristiques. Pour illustrer cet aspect, on note d_{max} la distance maximale entre deux points caractéristiques. En fixant $s = d_{max}$, on s'assure que la zone d'influence d'un point caractéristique en contienne au moins un autre. Pour démontrer son incidence, des tests sont réalisés en faisant varier le paramètre s de d_{max} à $10d_{max}$, par pas de $0.5d_{max}$. Différents résultats sont illustrés par la figure 4.11. On constate qu'ils sont de plus en plus lisses à mesure que la taille du support augmente. Sur cet exemple, le résultat devient bon lorsque $s = 3.5d_{max}$ et varie très peu pour $s > 3.5d_{max}$. La taille du support a un impact direct sur le temps de calcul, comme l'illustrent les tableaux ci-après où la première phase correspond à la résolution du système d'équations linéaires et la seconde calcule le déplacement de l'ensemble des nœuds de M_{2i}. Le tableau 4.4 correspond aux temps avec $s = 3.5d_{max}$ et le tableau 4.5 avec

$s = 10d_{max}$. Durant la phase de précalcul, la place mémoire requise augmente en $O(n^2)$, avec n le nombre de points dans M_{1i}. A noter qu'il est possible de la réduire en adoptant un mode de stockage optimisé ne mémorisant pas les zéros contenus dans Ψ. L'augmentation du temps de calcul est plus difficile à déterminer. Il dépend de n et de s. La seconde étape est linéaire avec une moyenne de $0.18ms$ par nœud pour un maillage M_{1i} contenant 2 000 nœuds et une taille de support $s = 3.5d_{max}$. Ainsi, un maillage M_{2i} contenant un million de points serait modifié en un peu plus de $3min$. Avec $s = 10d_{max}$, ce temps passerait à $15min$.

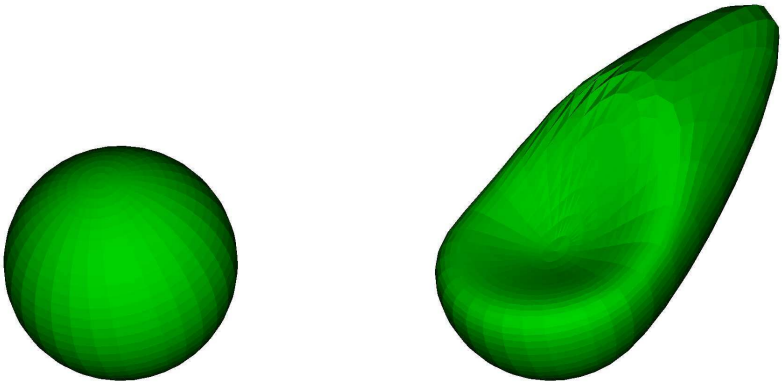
En utilisant un réseau de CS-RBF, il est donc possible d'obtenir un résultat intéressant. Le maillage final est lisse et le temps de calcul relativement faible. Le choix de la taille du support reste néanmoins un problème à résoudre.

Nb nœuds dans M_{1i}	Nb nœuds dans M_{2i}	Temps de calcul		
		total	précalcul	déplacement
1 000	90	0,8s	0,8s	$<$ 0,1s
1 000	380	0,8s	0,8s	$<$ 0,1s
1 000	1 000	0,9s	0,8s	0,1s
1 000	9 900	1,8s	0,8s	0,9s
1 000	39 800	4,5s	0,8s	3,7s
2 000	90	4,5s	4,5s	$<$ 0,1s
2 000	380	4,6s	4,5s	0,1s
2 000	1 000	4,7s	4,5s	0,2s
2 000	9 900	6,0s	4,5s	1,5s
2 000	39 800	10,6s	4,5s	6,1s

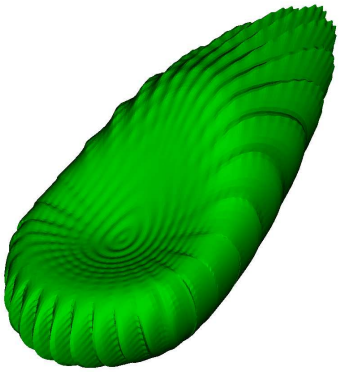
Table 4.4 – Détail des temps de calcul de la méthode utilisant une CS-RBF avec une taille de support $s = 3.5d_{max}$

Nb nœuds dans M_{1i}	Nb nœuds dans M_{2i}	Temps de calcul		
		total	précalcul	déplacement
1 000	90	2,5s	2,4s	0,1s
1 000	380	2,6s	2,4s	0,2s
1 000	1 000	3,0s	2,5s	0,5s
1 000	9 900	7,0s	2,4s	4,6s
1 000	39 800	21,0s	2,4s	18,6s
2 000	90	16,6s	16,5s	0,1s
2 000	380	17,0s	16,6s	0,4s
2 000	1 000	17,5s	16,5s	1,0s
2 000	9 900	26,0s	16,4s	9,6s
2 000	39 800	55,0s	16,6s	38,4s

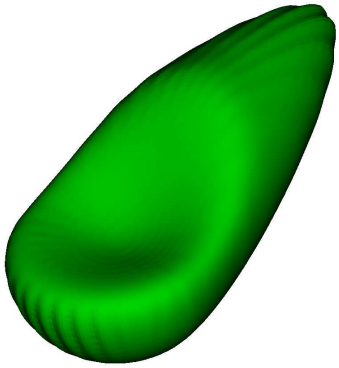
Table 4.5 – Détail des temps de calcul de la méthode utilisant une CS-RBF avec une taille de support $s = 10d_{max}$



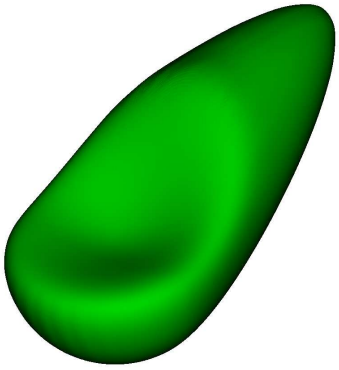
(a) M_1 contenant 1 000 nœuds (initial à gauche, déformé à droite)



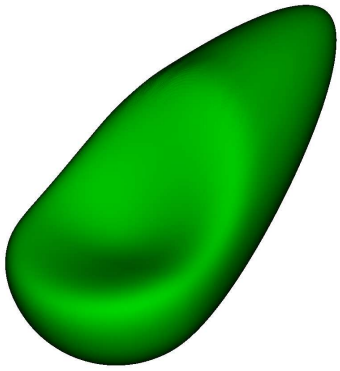
(b) $s = d_{max}$



(c) $s = 1.5d_{max}$



(d) $s = 2d_{max}$



(e) $s = 3.5d_{max}$

Figure 4.11 – Résultats de la méthode utilisant la CS-RBF multiquadrique en fonction de la taille du support s (M_{2i} plus fin que M_1 contenant 39 800 nœuds)

4.4 Synthèse

Ces études préliminaires ont permis de montrer qu'une approche naïve, l'interpolation linéaire, pouvait être utilisée. D'après les résultats obtenus, elle est intéressante lorsque le maillage sur lequel transférer la déformation est plus grossier que celui sur lequel la modification a eu lieu. Elle est décevante dans le cas contraire, situation fréquente lors du transfert de déformations géométriques calculées par un code de comportement mécanique. Cette méthode n'est donc pas retenue.

Les réseaux de RBF donnent quant à eux de bons résultats. La RBF multiquadrique nécessite néanmoins de fixer deux coefficients. Ce travail est délicat et difficile à réaliser de manière automatique. La RBF TPS a l'avantage de ne pas avoir de paramètre. Le support de la fonction étant toutefois global, le temps de calcul et la place mémoire requise augmentent très fortement en fonction du nombre de nœuds dans le maillage utilisé pour la définition de la déformation. La CS-RBF est à support local. Il est donc possible de réduire le temps de calcul et la place mémoire nécessaires. Il est néanmoins indispensable de définir la taille du support des fonctions.

En dehors de l'interpolation linéaire, toutes ces méthodes souffrent d'un problème de temps de calcul élevé, dépendant du nombre de points dans le maillage utilisé pour la définition de la déformation. Une technique visant à localiser la déformation pour réduire le nombre de nœuds intervenant dans le calcul de la fonction de transfert est donc proposée dans ce qui suit et l'évolution de deux fonctions existantes est proposée.

Chapitre 5

Transfert de déformations

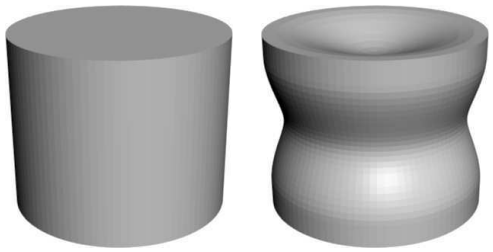
Sommaire

5.1 Localisation des déformations géométriques 79
5.2 Définition de nouvelles fonctions 84
5.2.1 Méthode de validation mise en place 84
5.2.2 Fonction à support global F_α 86
5.2.2.1 Présentation 87
5.2.2.2 Etape de simplification et paramètre α 87
5.2.2.3 Etape d'interpolation 90
5.2.2.4 F_α et modèle commun 90
5.2.2.5 Résultats 90
5.2.3 Fonction à support local F_α^σ 93
5.2.3.1 Présentation 94
5.2.3.2 Etape de simplification et paramètre α 94
5.2.3.3 Taille de support et paramètre σ 94
5.2.3.4 F_α^σ et modèle commun 95
5.2.3.5 Résultats 95
5.2.4 Bilan 98
5.3 Application 99
5.3.1 Entre des représentations d'une même géométrie 99
5.3.2 Entre des représentations de géométries adjacentes 107
5.4 Synthèse 111

Le principal inconvénient des réseaux de fonctions de base radiale est d'avoir un temps de calcul qui augmente considérablement en fonction du nombre de points dans le maillage sur lequel ils s'appuient (M_{1i} dans notre cas). Une méthode de localisation des déformations permet d'améliorer ce problème. Elle est présentée dans le paragraphe paragraphe 5.1. Son but est de limiter la zone de calcul du réseau de RBF afin d'éviter qu'il n'intervienne dans les parties non déformées. Deux nouvelles fonctions sont ensuite présentées dans le paragraphe 5.2. Elles permettent d'accélérer le temps de calcul et de gérer des maillages contenant un grand nombre de nœuds grâce à l'utilisation d'une étape de simplification. Le paragraphe 5.3 permet d'illustrer les résultats obtenus dans le cas d'un transfert réalisé entre des maillages différents issus d'une même géométrie ou de géométries adjacentes.

5.1 Localisation des déformations géométriques

Pour limiter la zone de calcul d'un réseau de RBF, une méthode de localisation des déformations est mise en place. Elle vise à définir différentes boîtes englobantes contenant chacune une ou plusieurs zone(s) modifiée(s). Finalement, un réseau de RBF est utilisé par boîte englobante, permettant ainsi de restreindre le calcul d'une fonction et de son application sur un nuage de points. Les cinq étapes de la méthode proposées sont explicitées ci-après. Les maillages M_{1i} et M_{1d} utilisés sont représentés sur la figure 5.1.

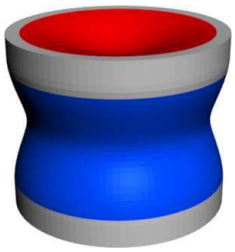


(a) M_{1i} (b) M_{1d}

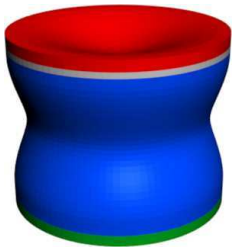
Figure 5.1 – Maillages utilisés pour illustrer la méthode de localisation des déformations

Etape 1

Tout d'abord, on parcourt l'ensemble des points du maillage M_{1i}. M_{1i} et M_{1d} étant appariés, il est possible de connaître l'ensemble des nœuds ayant été déplacés. Ces derniers sont regroupés dans des zones distinctes, chacune d'elle contenant des points reliés entre eux par l'intermédiaire des arêtes du maillage. Dans l'exemple ci-contre, trois zones sont définies : en rouge sur le dessus, en bleu sur le tour et en vert sur le dessous (partie cachée).

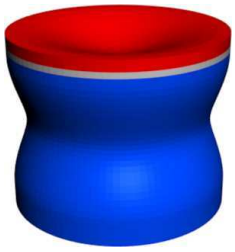


Etape 2



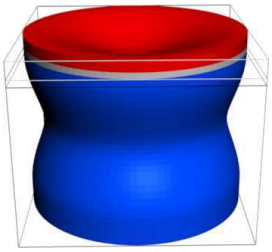
Chaque zone se voit augmentée de quelques voisins. Le but est d'obtenir un bon raccord encore les parties avec et sans modification. Pour cela, il est nécessaire de prendre en compte des points d'attache sur la partie non déformée. Leur nombre dépend de la taille de la zone modifiée. On appelle *n rang(s) de voisins* l'ensemble des points qui n'appartiennent pas à une zone et qui se trouvent à une distance n de l'une d'elles, cette distance correspondant au nombre d'arêtes qu'il est nécessaire de parcourir pour l'atteindre. Un minimum de 3 rangs de voisins a permis de montrer expérimentalement de bons résultats. Avec un tel choix, dans l'exemple ci-contre, la zone verte devient alors visible.

Etape 3



Il est possible que l'ajout des voisins implique un contact entre différentes zones. Toutes les zones connexes sont alors fusionnées. Dans l'exemple ci-contre, les anciennes zones bleue et verte sont fusionnées en une plus grande (en bleu). Celle en rouge reste distincte de la bleue, même après l'étape d'extension.

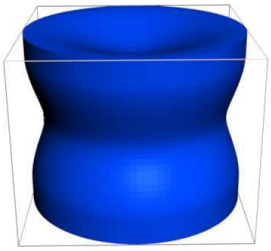
Etape 4



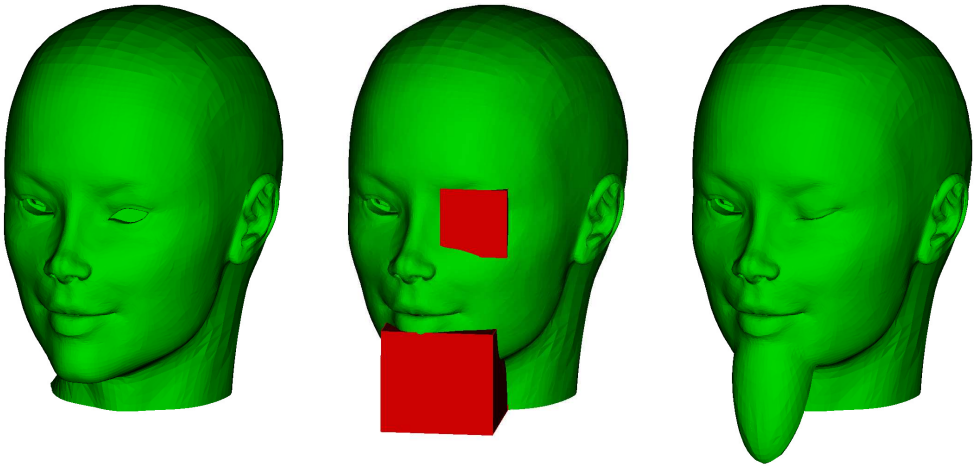
Le but de cette étape est d'éviter que différentes fonctions de transfert n'agissent sur une même zone. Les réseaux de RBF étant des fonctions de déformation d'espace, toutes les zones partageant le même support dans sa globalité ou en partie, doivent être traitées par une seule fonction de transfert. Afin de les localiser, une boîte englobante est mise en place pour chacune d'elles. Par soucis de simplicité, elle est définie en fonction des axes utilisés. Toutefois, dans le cas des maillages complexes, une amélioration peut être apportée afin de ne pas englober des parties non-connexes et donc des nœuds étrangers à la déformation. Dans ce cas, la boîte mise en place doit être définie en fonction du nuage de points utilisé, par exemple avec une approche par analyse en composantes principales (ACP, [DR06]).

Etape 5

Les boîtes englobantes connexes sont réunies par union pour définir une boîte plus grande. Tous les nœuds contenus dans cette dernière sont finalement utilisés lors du calcul du réseau de fonction de base radiale. Dans l'exemple ci-contre, en raison de la cavité de la zone du dessus (rouge), les deux boîtes se chevauchent. Elles sont donc réunies en une dernière, plus grande.



Grâce à cette approche, il est possible de réduire le nombre de nœuds utilisés pour caractériser la déformation. Elle est localisée avec une boîte englobante, facilement utilisable pour connaître les nœuds à déplacer par application de la fonction de transfert mise en place. Dans ce qui suit, un transfert de déformations locales est réalisé afin de montrer l'impact de cette technique. Le maillage utilisé est constitué de 10 000 nœuds (figure 5.2(a)). Deux modifications locales ont été appliquées, la première pour fermer l'œil gauche et la seconde pour faire apparaître un bouc au niveau du menton. Elles sont réalisées avec l'outil Blender ([Blender]) afin d'obtenir un maillage M_{1d} (figure 5.2(c)). Elles permettent de mettre en application la méthode de localisation des déformations. Les zones modifiées sont mises en évidence sur le maillage initial par les boîtes englobantes illustrées sur la figure 5.2(b).

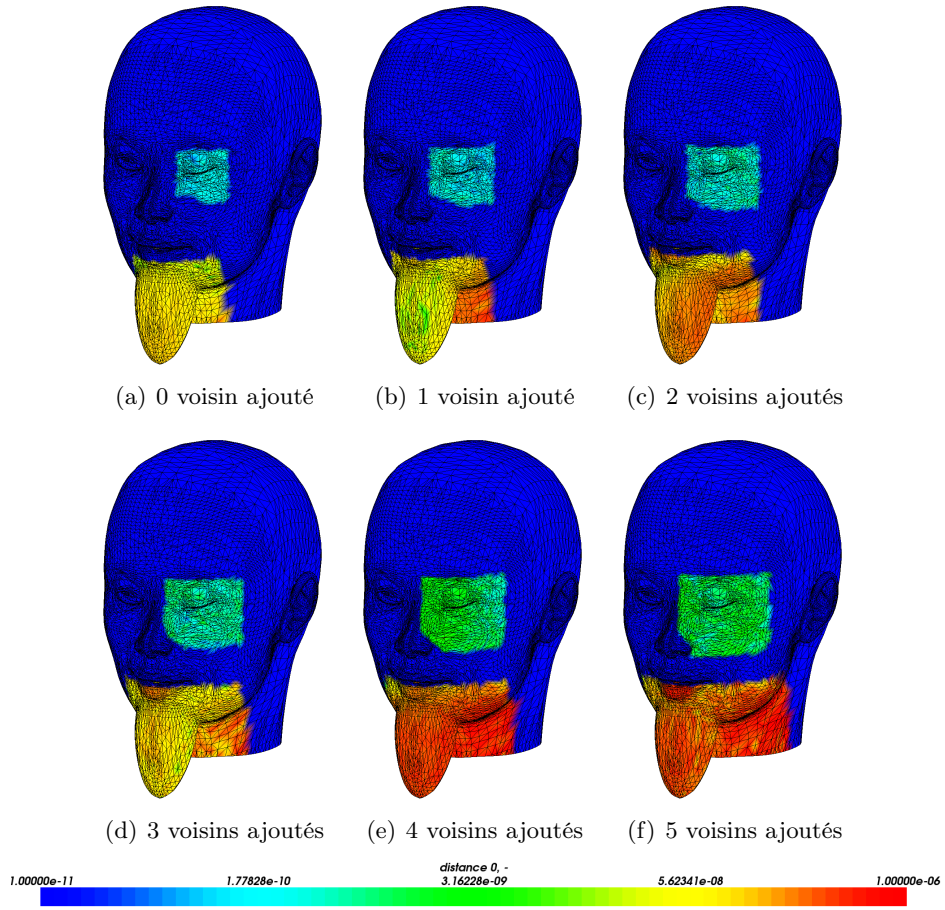


(a) M_{1i} (b) M_{1i} avec les boîtes englobantes sur les zones déformées (c) M_{1d}

Figure 5.2 – Maillages initial et déformé localement de la tête

Le transfert est réalisé par la RBF TPS sur le maillage M_{1i} à partir de la déformation survenue sur M_{1i} pour aboutir à M_{1d}. Le maillage ainsi obtenu et M_{1d} étant appariés, il est possible de connaître la distance euclidienne entre eux, point à point. Plus cette distance est faible, plus le transfert a été effectué de manière précise. Comme le montre la figure 5.3, plus on ajoute de voisins, plus

on détériore la qualité du résultat global. Ceci est dû au fait que la résolution du système d'équations linéaires induit par le réseau de RBF utilise une matrice de grande taille qui implique des erreurs d'arrondis lors des calculs numériques. Il apparaît donc clairement qu'il est préférable de limiter au maximum le nombre de voisins ajoutés.



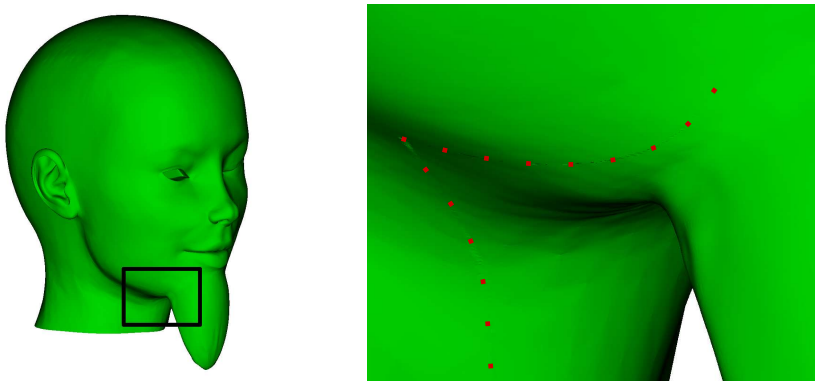
(a) 0 voisin ajouté (b) 1 voisin ajouté (c) 2 voisins ajoutés

(d) 3 voisins ajoutés (e) 4 voisins ajoutés (f) 5 voisins ajoutés

Figure 5.3 – Distance euclidienne entre le maillage calculé et M_{1d}

Toutefois, si la fonction de transfert n'est obtenue qu'à partir des points déplacés, sans tenir compte des autres nœuds proches, des discontinuités entre les zones calculées et non modifiées peuvent survenir lors de l'application de la fonction sur un maillage différent. Pour s'en convaincre, le transfert de ces deux modifications locales est réalisé sur un maillage plus fin, contenant 900 000 points, sans ajouter de point de la partie non modifiée. Le résultat obtenu est illustré sur la figure 5.4. Le trait en pointillés met en évidence la discontinuité observée. Sur cet exemple, le problème est corrigé lorsque trois voisins, au minimum, sont pris en compte.

Les différents temps de calcul sont contenus dans le tableau 5.1. La première



(a) maillage calculé (la zone zoomée est indiquée par le rectangle)

(b) aucun voisin ajouté (la discontinuité est mise en évidence par les pointillés)

Figure 5.4 – Mise en évidence d'une discontinuité au bord d'une zone modifiée

colonne indique le nombre de voisins ajoutés entre les zones déformées et non modifiées. La seconde contient le temps de calcul total du transfert des deux déformations avec, entre parenthèses, le ratio d'amélioration par rapport à un transfert s'appliquant sur la totalité du maillage. La suivante permet d'informer sur le détail des temps de calcul et d'application de la fonction de transfert pour chacune des deux zones, la première correspondant au menton, la seconde à l'œil. Comme on pouvait s'y attendre, plus on ajoute de voisins, plus le temps de calcul augmente. Dans tous les cas, il faut noter l'important gain de temps. Lorsque la méthode de localisation des déformations n'est pas utilisée, la fonction de transfert est calculée sur l'ensemble du maillage. Cette étape est réalisée en un peu moins de $40min$,

nb voisins	temps total	zone	calcul		application	
			# nœuds	temps	# nœuds	temps
0	3min 08s	1	1 299	5,5s	108 147	2min 35,5s
	(92×)	2	490	0,5s	45 547	26,5s
1	3min 46s	1	1 422	7,0s	116 376	3min 00,0s
	(77×)	2	585	0,7s	54 656	38,3s
2	5min 40s	1	1 601	10,0s	129 798	4min 26,0s
	(51×)	2	655	0,9s	61 667	1min 03,1s
3	8min 14s	1	1 770	17,0s	143 924	6min 39,0s
	(35×)	2	725	1,7s	68 214	1min 16,3s
4	10min 06s	1	1 877	22,5s	153 380	8min 00,0s
	(28×)	2	807	2,5s	76 039	1min 41,0s
5	11min 47s	1	2 042	25,5s	167 653	9min 09,5s
	(24×)	2	918	3,0s	86 364	2min 09,0s

Table 5.1 – Evolution du temps de calcul en fonction de nombre de voisins utilisés

M_{1i} contenant 10 000 points. Son application sur le maillage fin de 900 000 nœuds est effectuée en un peu plus de $4h$ $10min$. Ainsi, le temps de calcul total est de $4h$ $50min$. En comparant avec les temps obtenus avec la méthode proposée, il est possible de constater que le gain obtenu sur cet exemple est conséquent. A titre de comparaison, le transfert complet de la déformation, incluant le calcul de la fonction et son application sur le maillage fin, est réalisé en un peu plus de $8min$ si on adopte un choix de 3 voisins ajoutés, soit un résultat obtenu 35 fois plus vite.

5.2 Définition de nouvelles fonctions

En s'appuyant sur la technique de localisation des déformations, il est possible d'accélérer les calculs des fonctions de transfert. Cependant, cette approche ne peut être réellement intéressante que dans le cas de modifications locales, restreintes à de petites zones du maillage total. Dans le cas d'une déformation globale à tout le maillage, la boîte englobante n'a plus pour effet d'accélérer les calculs de la fonction de transfert mais sert à localiser spatialement les modifications géométriques dans le modèle commun. Dans le chapitre 4, les méthodes de transfert étudiées et jugées intéressantes ne permettent pas de prendre une empreinte d'une déformation faisant intervenir un grand nombre de points en un temps acceptable pour l'opération de couplage à réaliser au CEA. Il est donc impératif de mettre en place une fonction de transfert capable de traiter toutes les déformations, qu'elles soient locales ou globales, et pouvant faire intervenir de nombreux points. Pour cela, deux nouvelles fonctions sont définies, l'une à support global, l'autre à support local. La première s'appuie sur la fonction de base radiale TPS et la seconde sur la CS-RBF. Une méthode de simplification préalable est utilisée, rendant possible la spécification le nombre de nœuds qui serviront de support pour la phase d'interpolation. Elle permet ainsi de maîtriser les temps de calcul et l'espace mémoire utilisé.

5.2.1 Méthode de validation mise en place

La phase de simplification réduit la qualité des maillages définissant la déformation. L'étape d'interpolation risque donc de lisser des détails et de mal reconstruire les déformations localement fortes. Il est donc nécessaire de disposer d'un cas simple permettant de réaliser différents tests en fonction de l'étape de simplification. Le cas utilisé dans cette partie est spécifique à ce besoin. Il s'agit d'un demi-cylindre ouvert en bas regroupant différents cas pouvant poser problème : zone courbe pour la partie cylindrique, zone plane pour la partie médiane et arêtes saillantes. Les maillages sont volumiques, de l'ordre du mètre et comportent 30 500 nœuds et 30 510 tétraèdres. La déformation est réalisée avec l'outil Blender. Elle est de l'ordre du centimètre et globale à l'ensemble du maillage. Le maillage utilisé est représenté sur la figure 5.5. Son état initial est illustré à gauche tandis que la partie déformée est zoomée à droite. Afin de mettre en évidence la modification, la position initiale est entourée d'une ligne en pointillés et celle déformée en ligne pleine.

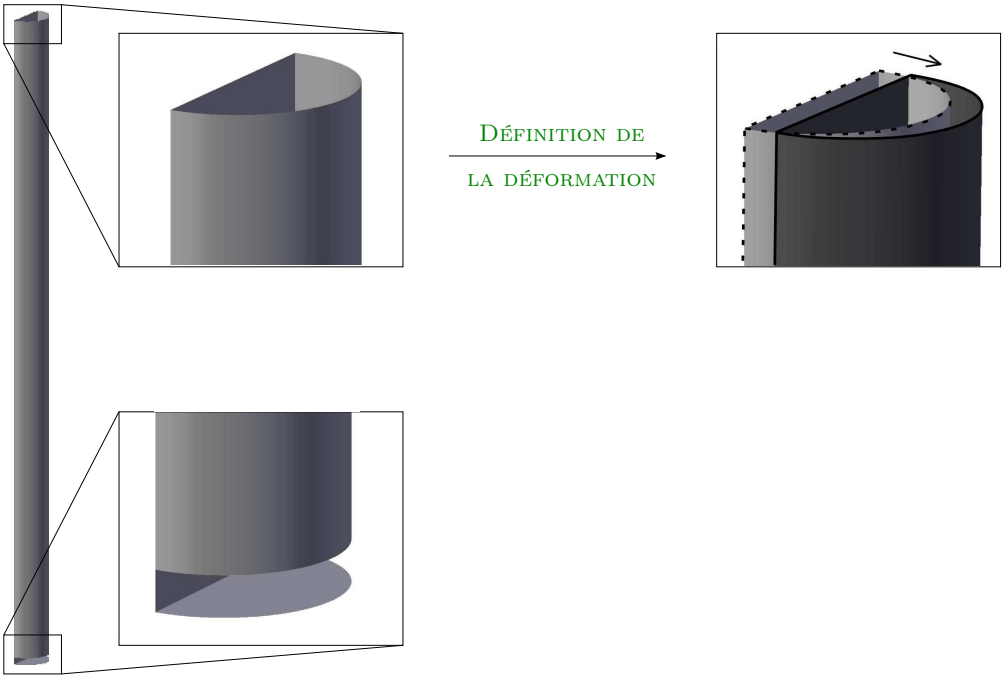


Figure 5.5 – Illustration du maillage utilisé pour la validation des fonctions mises en place

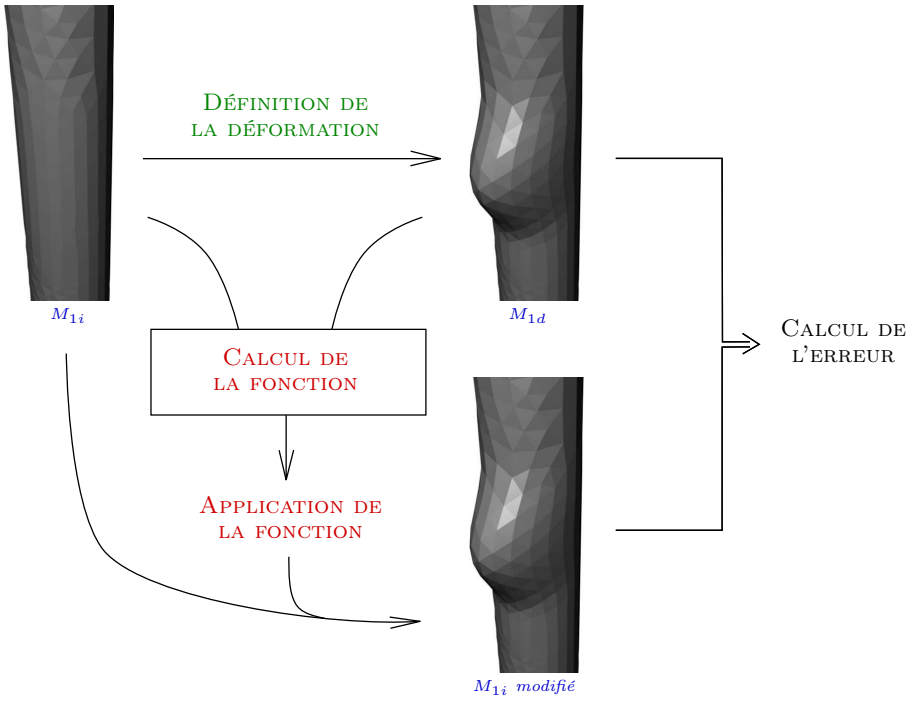


Figure 5.6 – Méthode de validation des fonctions définies

La méthode mise en place pour qualifier les résultats est schématisée par la figure 5.6. Un maillage initial, noté M_{1i}, est utilisé comme source. La déformation est définie sur un second maillage, noté M_{1d}. La modification géométrique est caractérisée par une des deux fonctions qui seront présentées dans le paragraphe 5.2.2 et le paragraphe 5.2.3. Cette dernière est ensuite appliquée sur le maillage source M_{1i} pour obtenir $M_{1i\ \text{modifié}}$. Les maillages M_{1d} et $M_{1i\ \text{modifié}}$ sont ensuite comparés. Pour mesurer la qualité du transfert de déformations, deux fonctions usuelles d'erreur sont introduites. La première, E_{max}, est locale et informe sur la distance euclidienne maximale. La seconde, E_{RMS}, est globale et renseigne sur la qualité moyenne d'interpolation. L'_j fait référence au $j^{ième}$ nœud du maillage M_{1d} et L''_j celui du maillage $M_{1i\ \text{modifié}}$.

$$E_{max} = \max_{j=1}^{n}(\|L''_j - L'_j\|_2) \tag{20}$$

$$E_{RMS} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\|L''_j - L'_j\|_2\right)^2} \tag{21}$$

Tous les calculs sont réalisés en double précision sous Linux Debian Etch, avec un processeur 64-bits, un seul cœur cadencé à $2.6GHz$ et disposant de $8Go$ de mémoire vive. La fonction de transfert utilisée a un impact sur la précision du résultat calculé. D'après [Dau84], le nombre de chiffres significatifs c sur la solution peut être approché par $c \simeq c_{max} - log_{10}(K(\Psi))$, avec $10^{-c_{max}}$ l'ordre de grandeur de l'incertitude sur les données et $K(\Psi)$ le conditionnement de la matrice Ψ (matrice utilisée par le réseau de RBF, *c.f.* paragraphe 2.4.1.3). On remarque d'après cette formule que c'est l'ordre de grandeur du conditionnement et non sa valeur qui est important. Les calculs sont tous réalisés en double précision et les données sont considérées exactes, donc $c_{max} = 16$. La norme 2 est utilisée. Pour $K_2 \simeq 10^{10}$, la précision sera de l'ordre du micron car les valeurs du maillage utilisé sont de l'ordre du mètre. Le conditionnement K_2 est généralement calculé par décomposition en valeurs propres car il correspond à la valeur absolue du rapport de la plus grande valeur propre sur la plus petite.

Les deux fonctions de transfert présentées dans cette partie font l'objet de tests. Tout d'abord, le conditionnement de la matrice Ψ est affiché en fonction des paramètres de chacune des fonctions, ainsi que les erreurs E_{RMS} et E_{max}. Les différents temps de calcul sont également détaillés. Les deux fonctions proposées sont testées sur des maillages de référence provenant de l'université de Stanford et les résultats obtenus sont présentés.

5.2.2 Fonction à support global F_α

La première fonction mise en place permet d'utiliser le principe de minimisation d'énergie offert par les TPS. Il vise également à combler le problème du temps de calcul du réseau de TPS en utilisant une méthode de simplification.

5.2.2.1 Présentation

La fonction F_α combine deux étapes différentes. Le première est une méthode de simplification utilisant la QEM (Quadric Error Metric, *c.f.* paragraphe 2.5.5). Il s'agit d'une approche par contraction d'arêtes qui garde α nœuds et dont les critères permettent de simplifier préférentiellement les zones peu courbées tout en gardant une certaine homogénéité dans la répartition du nuage de points. La simplification se fait de façon identique sur deux maillages afin de pouvoir garder l'appariement entre les maillages M_{1i} et M_{1d}. La seconde étape de la fonction F_α est le calcul d'un réseau de TPS qui interpole les α points gardés après simplification. Il faut alors signaler qu'en raison de l'étape de simplification, la fonction F_α n'interpole pas exactement la déformation. Les nœuds gardés après simplification étant néanmoins les plus importants (au sens du critère de la méthode de simplification), l'erreur induite est relativement faible, comme nous le montrerons dans les résultats. Ces nœuds sont ensuite appelés des points caractéristiques du réseau de TPS et sont notés $\{L_i\}_{i=1,..,\alpha}$. Pour chaque point v de $\mathbb{R}^3$, la fonction $F_\alpha : \mathbb{R}^3 \to \mathbb{R}^3$ est définie par :

$$F_\alpha(v) = \sum_{i=1}^{\alpha} \lambda_i . \phi(\|v - L_i\|_2) \quad + \mathcal{L}(v) \tag{22}$$

où ϕ est une RBF TPS C^2 continue ($\phi(r) = r^4 log(r)$). Chaque TPS utilisée est centrée en un point caractéristique noté L_i de $\mathbb{R}^3$. λ_i de $\mathbb{R}^3$ sont les coefficients du réseau et $\mathcal{L}$ une fonction polynomiale de degré un à coefficients dans $\mathbb{R}^3$.

5.2.2.2 Etape de simplification et paramètre α

L'étape de simplification permet d'extraire une liste de α points caractéristiques des maillages M_{1i} et M_{1d}. La méthode utilisant les QEM n'est malheureusement pas générique et requiert un maillage surfacique triangulaire. Pour combler cela, nous adoptons une approche par décomposition du maillage utilisé. Dans le cas d'un maillage volumique, les mailles de dimension trois sont converties en plusieurs de dimension deux. Ainsi, les tétraèdres sont décomposés en quatre triangles, les hexaèdres en six carrés, ... De cette manière, le critère de simplification associé à un nœud prend bien en compte l'ensemble des faces qui lui sont connexes. Ces dernières sont utilisées pour connaître les équations des plans des mailles associées à chaque nœud pour le calcul de la métrique initiale. Pour cette raison, les mailles de dimensions deux contenant plus que trois sommets sont triangulées. Dans le cas où tous les sommets d'une maille sont coplanaires, différentes approches peuvent être utilisées (voir [Kum96, LZ00, SM09] pour plus de détails). Seules les équations des plans nous intéresse, c'est pourquoi aucune contrainte n'est spécifiée aux triangles formés (*e.g.* pas de critère de forme à respecter). Une décomposition naïve et rapide est utilisée. Dans le cas où les sommets d'une maille ne sont pas coplanaires, il n'est pas possible d'associer un plan à la maille. Un choix de décomposition doit donc être réalisé, faisant alors intervenir un biais. Il faut tout de même noter que les

erreurs engendrées sont faibles car n'influencent que le calcul initial des QEM. Pour cette raison, une décomposition naïve et rapide est également utilisée dans ce cas.

Les α nœuds obtenus après simplification sont utilisés ensuite pour effectuer l'interpolation par le réseau de TPS. Il faut donc simplifier les deux maillages de manière identique afin de garder l'appariement entre leurs nœuds.

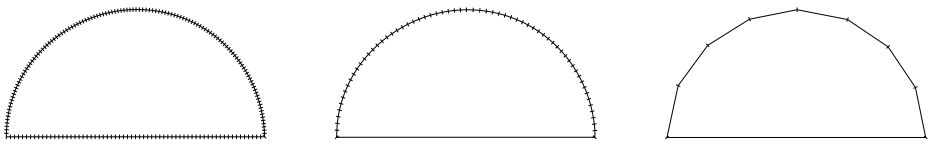
Simplification de deux maillages appariés

Traditionnellement, les méthodes de simplification se concentrent sur un seul maillage. L'idée exposée est d'effectuer ce travail sur deux maillages en même temps afin de garder l'appariement entre eux. M_{1i} pouvant contenir des particularités absentes dans M_{1d} et *vice versa*, il est important de faire intervenir les quadriques des deux maillages afin de piloter la simplification. Pour ce faire, la quadrique $Q(v)$, définie précédemment dans le paragraphe 2.5.5, est redéfinie comme étant la valeur maximale des quadriques des deux maillages :

$$Q(v) = \max(Q_{1i}(v), Q_{1d}(v)) \tag{23}$$

avec Q_{1i} (resp. Q_{1d}) la quadrique associée au maillage M_{1i} (resp. M_{1d}). Ce critère, commun aux deux maillages, permet de sélectionner l'arête qui sera contractée. De cette manière, les opérations de simplification sont identiques pour les deux maillages et l'appariement entre les nœuds est conservé.

Le point fort du critère utilisé par Garland est de simplifier préférentiellement les zones de faible courbure où la contraction d'une arête entraîne la plus petite erreur possible. Toutefois, dans notre cas, ce point peut également être un problème. En effet, les zones variant très peu se voient fortement simplifiées. Ce phénomène est schématisé sur la figure 5.7(b), où la droite du demi-cercle est très simplifiée contrairement au demi-cercle lui-même. Cela crée un déséquilibre important dans la répartition des points sur le maillage. Dans le cas d'une simplification plus poussée, cet aspect est progressivement gommé (figure 5.7(c)).

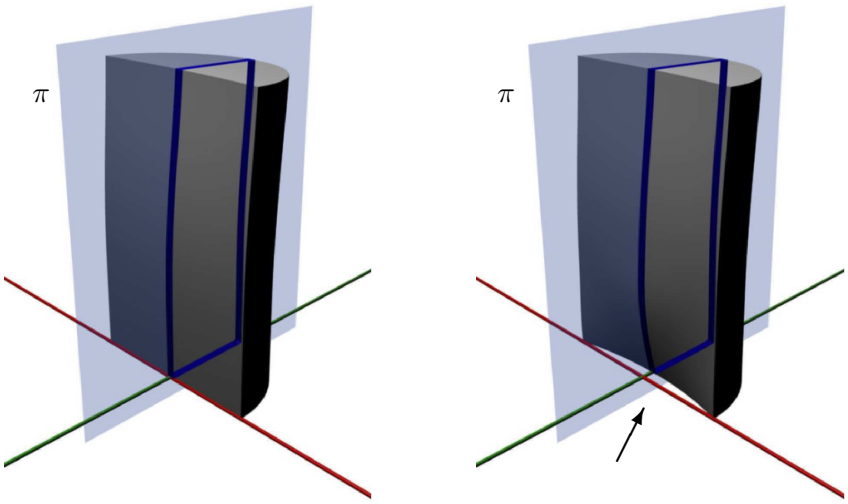


(a) avant la simplification (b) première simplification (c) simplification importante

Figure 5.7 – Etape de simplification utilisant les QEM

Bien que la TPS ne nécessite pas que les points caractéristiques soient répartis de manière homogène, ce fort déséquilibre aboutit à un manque d'informations. Dans ces zones particulières, le réseau de fonctions TPS ainsi calculé, n'ayant pas assez de points d'attache, peut aboutir à de mauvais résultats. La figure 5.8 illustre ce point sur un exemple où la fonction F_α est utilisée avec et sans l'étape de simplification. Un demi-cylindre est déformé par application d'une force sur sa partie supérieure. La modification géométrique est transférée sur un second maillage, aboutissant au

problème pointé par la flèche. La disparité dans le nuage de points aboutit à une fonction calculée qui conduit à une mauvaise reconstruction de la déformation.



(a) sans simplification (b) avec simplification

Figure 5.8 – Impact de l'étape de simplification sur la qualité du transfert de déformations (le plan π est utilisé pour montrer le bord du demi-cylindre)

Critère d'homogénéité

Pour pallier ce problème, nous avons choisi de modifier le critère de simplification utilisé. La quadrique de Garland ayant pour but de minimiser l'erreur engendrée par la contraction d'une arête, le nouveau critère ajouté se devait de prendre en compte l'homogénéité dans la position des points. Le choix retenu est la somme des aires des faces auxquelles un nœud appartient. De la même façon que précédemment, ce critère est calculé sur les deux maillages M_{1i} et M_{1d} et le critère H est défini par :

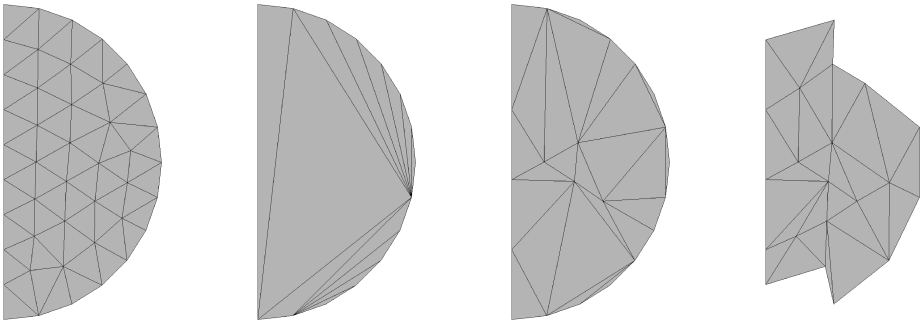
$$H(v) = \sum_{F \in faces_{1i}(v)} aire(F) + \sum_{F \in faces_{1d}(v)} aire(F) \tag{24}$$

avec $faces_{1i}(v)$ (resp. $faces_{1d}(v)$) l'ensemble des faces du maillage M_{1i} (resp. M_{1d}) auxquelles le point v appartient.

Ainsi, au début de la phase de simplification, les aires étant faibles, le critère H intervient peu dans le choix de l'arête à contracter. Au fur et à mesure, ce dernier joue un rôle de plus en plus important, permettant ainsi de préserver un minimum d'homogénéité dans la disposition des points. Afin de régler l'importance des deux critères utilisés, Q et H, pour pouvoir privilégier la qualité de la surface du maillage ou son homogénéité, le paramètre ω est introduit. La fonction de coût utilisée dans la méthode de simplification sera alors :

$$C(v) = \omega.\Delta(v) + (1-\omega).H(v), \quad \omega \in [0,1] \in \mathbb{R} \tag{25}$$

Au-delà du fait que les critères Q et H fassent tous deux intervenir des distances au carré, il faut signaler que Q n'est pas une aire, contrairement à H. De plus, la valeur du paramètre ω est dépendante des maillages utilisés. Malgré cela, les tests réalisés ont montré que le problème, illustré par la figure 5.8, est résolu lorsque ω est fixé expérimentalement 0.5.



(a) initial (b) $\omega = 1.0$ (c) $\omega = 0.5$ (d) $\omega = 0.0$

Figure 5.9 – Simplification d'un demi-cercle en fonction du paramètre ω

5.2.2.3 Etape d'interpolation

Les maillages M_{1i} et M_{1d} obtenus après la phase de simplification contiennent tous deux α nœuds, appelés points caractéristiques et notés $\{L_j\}_{j=1,..,\alpha}$ et $\{L'_j\}_{j=1,..,\alpha}$ pour les M_{1i} et M_{1d} respectivement. La condition d'interpolation du réseau de TPS est définie en imposant $F_\alpha(L_j) = L'_j$, pour $j = 1, .., \alpha$. De cette manière, il est possible de définir un système d'équations linéaires à résoudre pour connaître les valeurs des paramètres λ du réseau.

5.2.2.4 F_α et modèle commun

L'intégration de la fonction F_α dans le modèle commun nécessitera de stocker les paramètres du réseau de TPS précédemment calculés. Ces dernières seront les coefficients $\{\lambda_k\}_{k=1,..,\alpha+4}$ ainsi que les points caractéristiques $\{L_j\}_{j=1,..,\alpha}$. La taille nécessaire sera donc de $2\alpha + 4$ valeurs dans $\mathbb{R}^3$, soit $6\alpha + 12$ réels.

5.2.2.5 Résultats

Pour illustrer le conditionnement de la matrice Ψ induite par le réseau de TPS, l'approche traditionnelle par décomposition en valeurs propres a été testée. Malheureusement, en raison de la forme de Ψ, cette dernière ne permet pas d'obtenir d'informations exploitables. Le système d'équations linéaires à résoudre est de la forme $\Psi\ \lambda = L'$. L'approche choisie est de calculer l'impact de la variation de L' sur la valeur des coefficients λ. Dans le cas où la norme 2 est utilisée, une approximation

inférieure du conditionnement peut être obtenue par ([Dau84]) :

$$\frac{\|\Delta\lambda\|}{\|\lambda\|} \leq K_2(\Psi)\ \frac{\|\Delta L'\|}{\|L'\|} \quad \Rightarrow \quad K_2(\Psi) > \frac{\|\Delta\lambda\|.\|L'\|}{\|\lambda\|.\|\Delta L'\|} \tag{26}$$

L'influence du paramètre α sur le conditionnement de Ψ est illustrée sur le graphique 5.10. L'ordonnée correspond à la valeur $K_2(\Psi)$ et l'abscisse à la valeur du paramètre α en pourcentage de n, le nombre de points dans le maillage initial. Nous pouvons alors remarquer que, plus la valeur du paramètre α est faible, donc plus le maillage est simplifié, meilleur est le conditionnement.

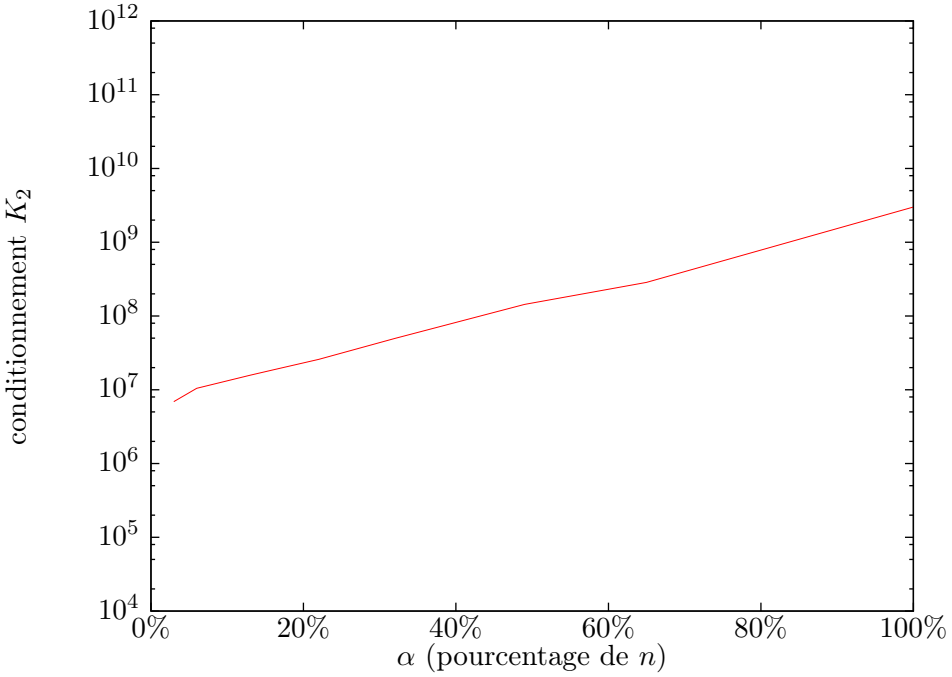
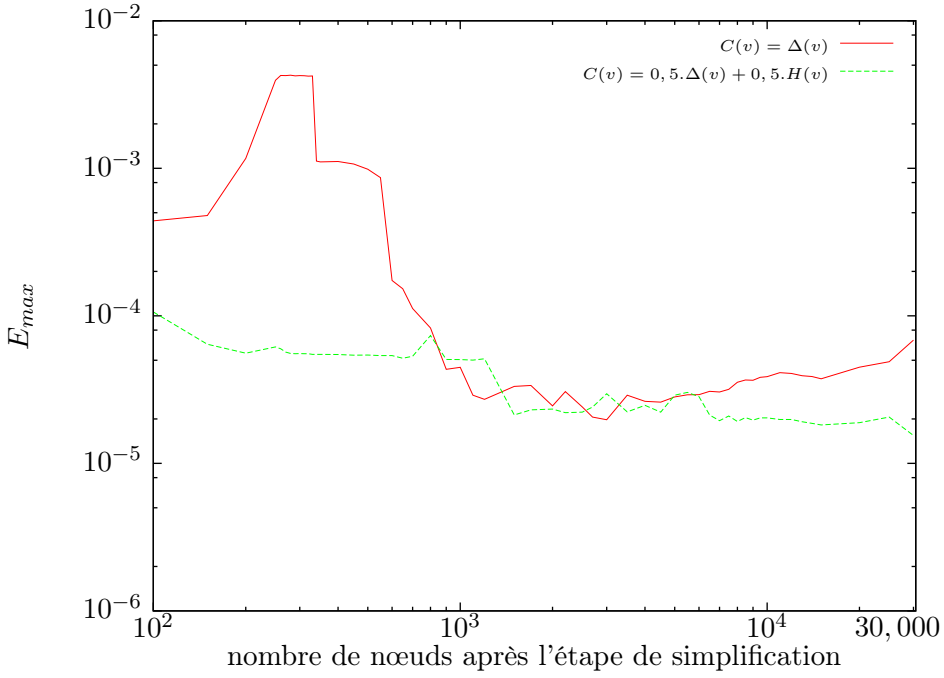


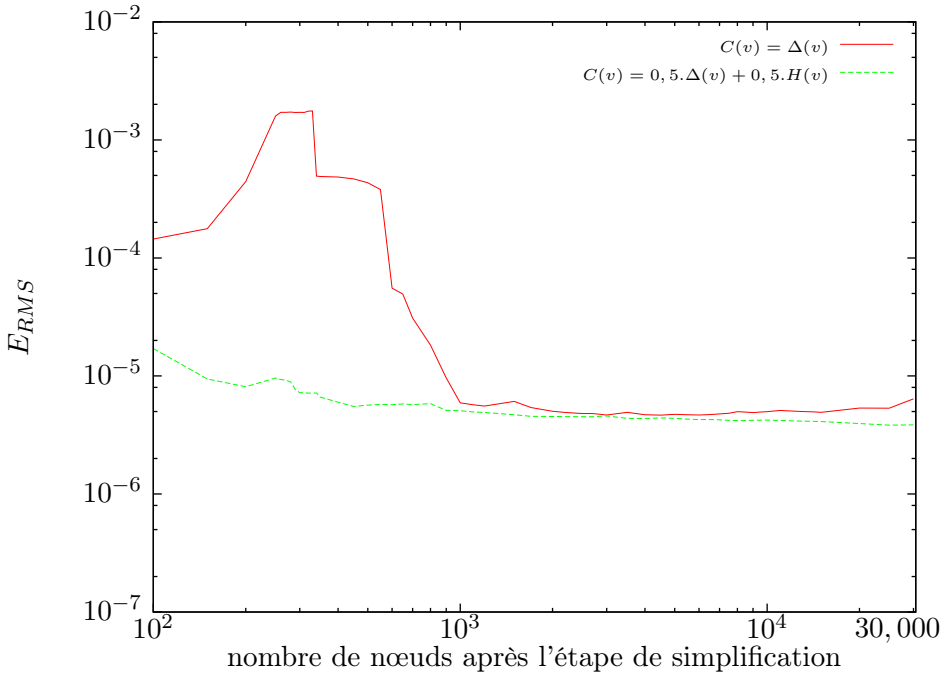
Figure 5.10 – Conditionnement K_2 de la matrice Ψ de F_α pour $n = 30\ 500$

L'évolution des erreurs dues à la phase de simplification est illustrée par la figure 5.11, où les erreurs E_{RMS} et E_{max} sont affichées en fonction de la valeur du paramètre α. Le pic sur les deux graphiques entre les simplifications à 100 nœuds et à 1 000 nœuds est l'illustration de l'effet pointé par la figure 5.8 qui met en évidence le problème d'interpolation suite à une simplification aboutissant à une répartition déséquilibrée du nuage de points. Il faut alors noter que l'ajout du critère H dans la fonction de simplification aide à le réduire. Dans cet exemple, les erreurs sont identiques avec et sans l'ajout de ce critère (en dehors du pic sur chacune des figures). Il faut néanmoins rappeler que le paramètre ω est dépendant du maillage.

La phase de simplification a également un impact sur le temps de calcul, comme l'illustre le tableau 5.2. Ce dernier contient le temps de calcul de l'ensemble de la méthode, le pic de mémoire utilisée et affiche les erreurs E_{RMS} et E_{max} avec le paramètre $\omega = 0$ et $\omega = 0.5$. Toutes ces données sont fonction du nombre α de



(a) E_{max}



(b) E_{RMS}

Figure 5.11 – Erreurs en fonction du paramètre α

nœuds gardés par la méthode de simplification. Le système d'équations linéaires est résolu par une élimination directe de Gauss, les méthodes de résolution performantes à notre disposition [9] ne permettant pas de traiter la matrice Ψ.

	Temps de	Pic de	E_{RMS} (10^{-8})		E_{max} (10^{-8})	
α	calcul	mémoire	$\omega = 0.5$	$\omega = 1.0$	$\omega = 0.5$	$\omega = 1.0$
500	6s	52Mo	566	43 387	5 420	98 638
1 000	12s	52Mo	508	582	5 048	4 473
3 000	1min 25s	109Mo	459	467	2 962	1 975
6 000	7min 51s	318Mo	427	467	2 829	2 923
10 000	34min 34s	808Mo	423	498	2 032	3 868
20 000	4h 31min 55s	3174Mo	395	535	1 888	4 482
30 000	15h 20min 30s	6973Mo	385	639	1 545	6 840

Table 5.2 – Influence du paramètre α sur les performances

Le temps de calcul des principales étapes de la méthode est détaillé dans le tableau 5.3. Cette méthode est en $O(\alpha^3)$ pour la résolution du système et en $O(\alpha^2)$ pour le calcul des nouvelles positions.

α	Etape de simplification	Temps de calculs de F_α	Temps d'application de F_α
500	$\sim$ 1s	$<$ 1s	5s
1 000	$\sim$ 1s	2s	9s
3 000	$<$ 1s	54s	27s
6 000	$<$ 1s	6min 55s	55s
10 000	$<<$ 0s	32min 55s	1min 38s
20 000	$<<$ 0s	4h 28min 48s	3min 07s
30 000	$<<$ 0s	15h 15min 29s	5min 01s

Table 5.3 – Détails du temps de calcul en fonction du paramètre α

5.2.3 Fonction à support local F_α^σ

La seconde fonction mise en place vise à accélérer les temps de calcul en utilisant une fonction à support local. De cette manière, il est possible d'utiliser un plus grand nombre de points caractéristiques. Les bons résultats apportés par la fonction F_α ont motivé le choix d'une fonction RBF aux résultats proches de celui des TPS.

9. Les calculs sont réalisés en s'appuyant sur la librairie PETSc (Portable, Extensible Toolkit for Scientific Computation, http://www.mcs.anl.gov/petsc/).

5.2.3.1 Présentation

La fonction F_α^σ est identique à F_α, à la différence que la RBF utilisée n'est pas une TPS mais une CS-RBF. L'inconvénient majeur de ce choix est la perte de la propriété mathématique de minimisation de l'énergie de variation des TPS ([Boo89]). La forme des CS-RBF dépend de la dimension du maillage. Dans notre cas, les maillages sont dans $\mathbb{R}^3$, ce qui implique comme expression de ϕ :

$$\phi(r) = \begin{cases} (1-r)^4(4r+1) & ; \text{si r} < 1 \\ 0 & ; \text{sinon} \end{cases} \tag{27}$$

Le support de cette fonction est la boule unité, avec $\phi(0) = 1$ et $\forall r \in]0;1[,\ 0 < \phi(r) < 1$. La matrice Ψ est ainsi symétrique définie positive, creuse et à diagonale non nulle ([Wen05b]).

5.2.3.2 Etape de simplification et paramètre α

La phase de simplification se fait de manière identique à la fonction F_α. La forme de la matrice permettant d'accélérer les temps de calcul, il est toutefois possible d'utiliser une valeur α plus grande. Cela permet de garder plus de points, donc plus d'informations sur la déformation.

5.2.3.3 Taille de support et paramètre σ

Afin de pouvoir prendre en compte les différentes répartitions du nuage de points des maillages, la taille du support de la fonction ϕ doit être modifiable. Bien que des travaux portent sur la résolution de ce problème ([OBS05]), ce dernier n'est pas encore résolu. La principale difficulté réside dans la manière de fixer le nouveau support. Une valeur trop faible induirait un manque d'information entre les nœuds pour la phase d'interpolation et donc un mauvais résultat. Dans le cas contraire, une valeur trop élevée ferait perdre le bénéfice d'utiliser une RBF à support compact et la matrice du système à résoudre se remplirait de manière importante. Le temps de calcul en pâtirait dans le cas de l'utilisation d'une méthode de résolution itérative.

Nous introduisons le paramètre σ de la fonction permettant de redéfinir la taille du support des CS-RBF utilisées. Nous notons alors $\phi_i^{s_i}(r) = \phi_i(r/s_i)$. Pour une fonction donnée ϕ_i, s_i est calculé comme étant le rayon minimal de la boule contenant σ centres $\{L\}$ de CS-RBF. La taille finale du support S est obtenue de manière à s'assurer que toutes les CS-RBF font intervenir au moins σ centres :

$$S = \max_{1 \leq i \leq \alpha} (s_i) \tag{28}$$

La figure 5.12 illustre ce point avec $\sigma = 3$. Chaque centre L_i se voit attribuer une sphère (illustrée en deux dimensions par un cercle noté C_i) dont le rayon augmente jusqu'à ce que la condition détaillée précédemment soit respectée. Dans cet exemple, la taille du support global sera celle du rayon de la plus grande sphère, à savoir celle en pointillés et notée C_0.

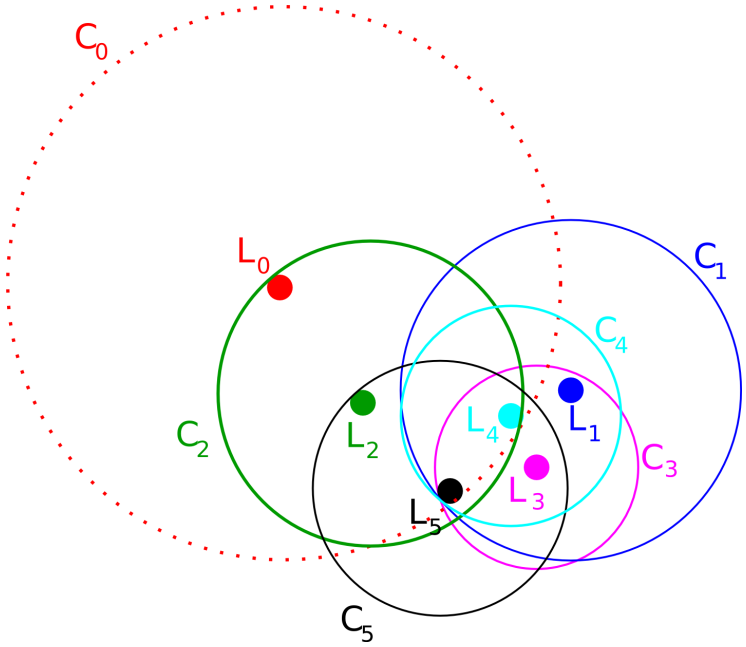


Figure 5.12 – Schéma illustrant la taille du support d'une CS-RBF en fonction du paramètre $\sigma = 3$ (ici $\alpha = 6$)

Le calcul de la taille du support global n'est réalisé qu'une fois pour une fonction F_α^σ donnée. De cette manière, il est possible d'accélérer les calculs et de réduire la place mémoire utilisée. De plus, cette approche permet également de gérer les maillages non uniformes par l'intermédiaire du paramètre σ. Plus sa valeur sera importante, plus le rayon d'influence des CS-RBF sera grand et donc plus de centres interviendront dans le calcul d'un point. Il faut toutefois signaler que les avantages de l'utilisation d'une fonction à support compact décroissent avec l'augmentation de σ.

La technique qui vient d'être détaillée est une première approche visant à apporter une réponse au problème de la taille du support du réseau de CS-RBF. D'autres méthodes doivent être testées pour pouvoir les confronter au choix réalisé.

5.2.3.4 F_α^σ et modèle commun

L'intégration de la fonction F_α^σ dans le modèle commun nécessitera de stocker les paramètres du réseau de CS-RBF précédemment calculé. Ces dernières seront les coefficients $\{\lambda_k\}_{k=1,..,\alpha+4}$, les points caractéristiques $\{L_j\}_{j=1,..,\alpha}$ ainsi que la taille du support des fonctions. La taille mémoire nécessaire sera donc de $2\alpha + 4$ valeurs dans $\mathbb{R}^3$ et d'un réel (la taille du support), donc $6\alpha + 13$ réels.

5.2.3.5 Résultats

L'influence des paramètres α et σ sur le conditionnement de Ψ est illustrée sur le graphique 5.13, avec $n = 30\ 500$ le nombre de points dans le maillage initial.

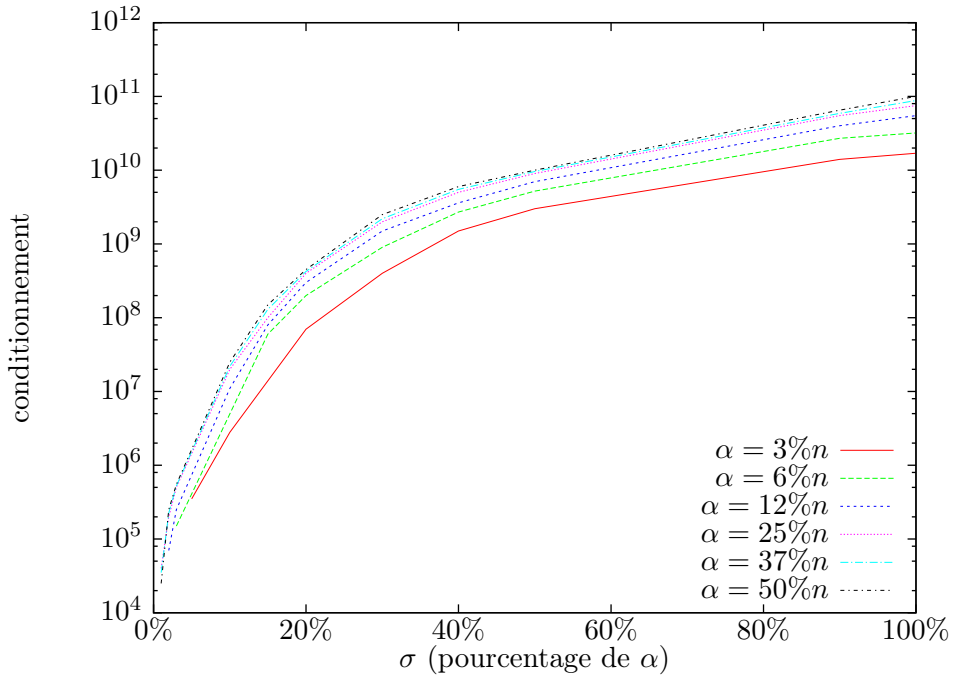
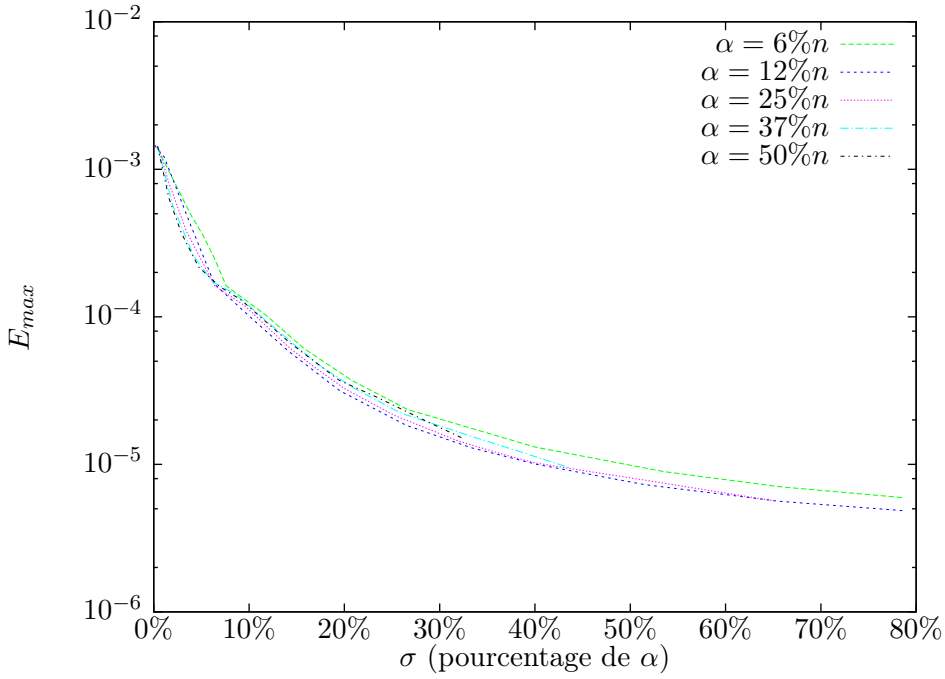


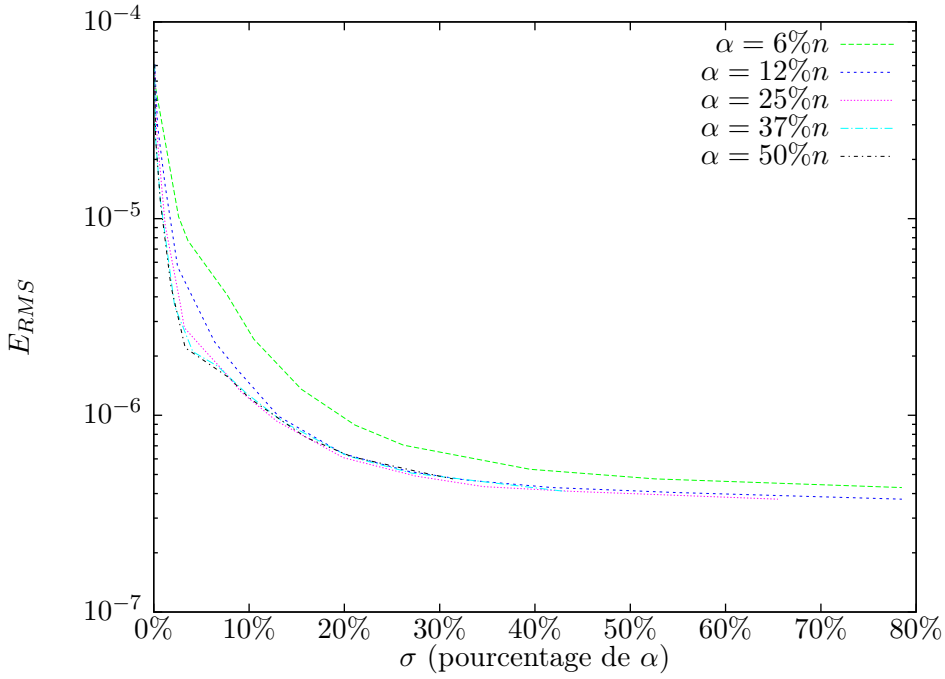
Figure 5.13 – Conditionnement K_2 de la matrice Ψ de F_α^σ pour $n = 30\ 500$

Dans ce cas, une approche par décomposition en valeurs propres a pu être utilisée. Sur le graphique, le conditionnement K_2 est affiché en fonction du paramètre σ, pourcentage du paramètre α. Chacune des courbes correspond à un choix de valeur pour α en pourcentage du nombre de nœuds n dans le maillage initial. Pour un σ donné, on remarque que l'ensemble des courbes sont proches, variant au maximum d'un ordre de grandeur entre $\alpha = 3\%\ n$ et $\alpha = 50\%\ n$. Le paramètre α n'a donc pas un fort impact sur le conditionnement K_2 de la matrice Ψ. La valeur de σ est plus déterminante. Lorsque σ est compris entre 40% α et 100% α, la variation de K_2 est faible. Lorsque $\sigma < 40\%\ \alpha$, on constate une décroissance importante de la valeur du conditionnement à mesure que σ diminue. Nous pouvons finalement remarquer que, plus les valeurs des paramètres sont faibles, meilleur est le conditionnement.

Les graphiques 5.14(a) et 5.14(b) illustrent les erreurs obtenues. Les calculs ont été réalisés en utilisant différentes valeurs de α, variant de 6% à 50% de la taille totale n du maillage. Lorsque α diminue, les erreurs ont tendance à augmenter en raison du manque d'information pour la phase d'interpolation. Elles restent néanmoins très similaires, ce qui permet d'indiquer que la valeur du paramètre α n'a pas un fort impact sur la qualité du résultat. Celle du paramètre σ est plus déterminante. Lorsque $\sigma \geq 20\%\ \alpha$, les erreurs sont similaires. Toutefois, lorsque $\sigma < 20\%\ \alpha$, elles croissent de façon importante. Les temps de calcul associés aux résultats sont détaillés dans le tableau 5.4 où seuls les temps avec σ faible sont affichés car l'augmentation se poursuit dans les mêmes proportions. Le choix des valeurs de α et σ doit être un compromis entre qualité de l'interpolation et temps de calcul, ce que nous pourrions évaluer sur notre cas test par $\alpha = 6\%\ n$ et $\sigma = 20\%\ \alpha$.



(a) E_{max}



(b) E_{RMS}

Figure 5.14 – Erreurs en fonction des paramètres α et σ

α (% n)	σ (% α)	Temps de calcul	Temps d'application
6%	10%	9s	12s
6%	20%	20s	24s
6%	30%	33s	33s
12%	10%	42s	27s
12%	20%	1m 57s	50s
12%	30%	3m 25s	1m 07s
25%	10%	3m 43s	50s
25%	20%	11m 48s	2m 06s
25%	30%	31m 18s	3m 15s
50%	10%	38m 43s	55s
50%	20%	1h 42m 35s	1m 42s
50%	30%	2h 59m 40s	3m 16s

Table 5.4 – Temps de calcul en fonction des valeurs de α et σ

Avec ces valeurs, 8 chiffres sont significatifs. Le résultat est tout à fait compatible avec nos besoins (*c.f.* paragraphe 5.2.1) car l'erreur moyenne est $E_{RMS} < 10^{-6}$ et l'erreur maximale est $E_{max} < 3 \times 10^{-5}$, pour des déplacements de l'ordre du centimètre. Il faut toutefois signaler que ces valeurs dépendent évidemment des maillages utilisés et du type des déformations qui interviennent.

5.2.4 Bilan

Dans ce paragraphe, deux fonctions de transfert ont été présentées.

La première, F_α, s'appuie sur la RBF TPS. Elle a donc l'avantage de minimiser une énergie de flexion, ce qui lui permet d'aboutir à des résultats "lisses". Elle possède néanmoins un support global lui imposant une valeur α qui doit être fixée par un utilisateur en tenant compte de deux aspects. Tout d'abord, si le temps de calcul est un facteur important, il est nécessaire de définir α en fonction de la puissance de calcul à disposition. Enfin, la taille de la matrice du système d'équations linéaires à résoudre est étroitement liée au paramètre α. La valeur de α doit donc permettre de stocker la matrice dans la mémoire disponible de l'ordinateur. La fonction F_α a fait l'objet de trois publications : [DDGP10a, DDGP10b, DGDP11b].

La seconde, F_α^σ, utilise la CS-RBF à support local. Elle ne minimise pas d'énergie de flexion mais aboutit à des résultats visuellement proches des TPS. L'utilisation d'un support local permet d'accélérer les temps de calcul. Toutefois, un paramètre supplémentaire doit être fixé pour définir la taille du support utilisé. Une trop petite valeur induit de mauvais résultats car trop peu d'informations sont utilisées lors de la phase d'interpolation. Une trop grande remplit exagérément la matrice du système d'équations linéaires ce qui handicape les méthodes de résolution numérique. De plus, cette valeur est dépendante du maillage utilisé, ce qui la rend difficile à choisir. Pour cela, le paramètre σ est introduit. Il permet de définir de manière automatique la taille du support en fonction d'une donnée quantifiable par l'utilisateur, à savoir

le nombre de nœuds minimum requis lors de l'étape d'interpolation. La fonction F_α^σ a été publiée dans : [DGDP10, DGDP11a].

Bien que pour un problème donné, F_α et F_α^σ soient toutes deux utilisables, le choix de l'utilisation de l'une ou l'autre doit être réalisé au cas par cas. La fonction F_α est préférentiellement conseillée dans le cas où la propriété mathématique de minimisation de l'énergie de variation est nécessaire. Pour un temps de calcul donné, la fonction F_α^σ permet de spécifier une valeur α plus grande que pour F_α. Elle est donc particulièrement intéressante dans les cas faisant intervenir un grand nombre de nœuds et de nombreuses modifications géométriques. Si le paramètre σ n'est pas souhaité ou que sa valeur est difficile à fixer, l'utilisation de la fonction F_α est conseillée.

5.3 Application

Le but de ce paragraphe est d'illustrer les résultats obtenus sur des maillages complexes. F_α et F_α^σ permettent toutes deux de prendre une empreinte d'une déformation géométrique. Cette dernière est ensuite utilisée pour calculer le déplacement des nœuds d'un second maillage. En se replaçant dans la méthode de couplage illustrée par la figure 3.1, ce travail correspond respectivement aux étapes ❻ et ❼. Comme le détaille le problème exposé dans le paragraphe 1.2.2, deux types de transfert de déformations sont possibles, soit entre des maillages issus d'une même géométrie, soit entre des géométries adjacentes.

5.3.1 Entre des représentations d'une même géométrie

Dans le premier cas, les maillages utilisés discrétisent une même géométrie. Ils sont néanmoins susceptibles d'être très différents. Une méthode adaptée a donc été développée afin de permettre un transfert de déformations entre eux.

Deux traitements sont réalisés sur une même géométrie (l'objet A sur la figure 5.15). Le maillage pour la première tâche est obtenu après discrétisation de la géométrie ($1a$). Le travail réalisé correspond à la définition de la déformation géométrique ($1b$). Une ou plusieurs fonctions de transfert permettent d'interpoler les modifications géométriques intervenues sur le maillage initial ($1c$). Sur cet exemple, une seule fonction sera nécessaire car la déformation est globale à l'ensemble du maillage.
Le maillage utilisé pour le second traitement est obtenu après discrétisation de la même géométrie ($2a$). Avant de pouvoir lancer le second traitement, le maillage doit être mis à jour. Les parties à modifier sont alors déplacées par l'intermédiaire des fonctions précédemment calculées. Une fois appliquées à la totalité des nœuds, le maillage de l'objet A est complètement modifié ($2b$).

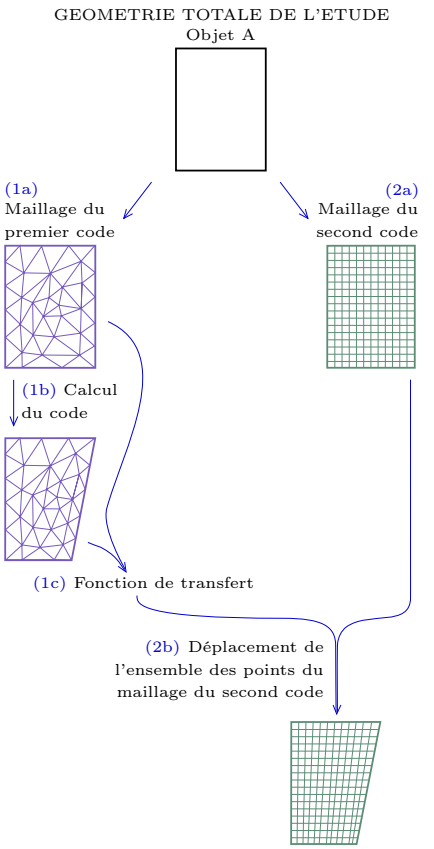
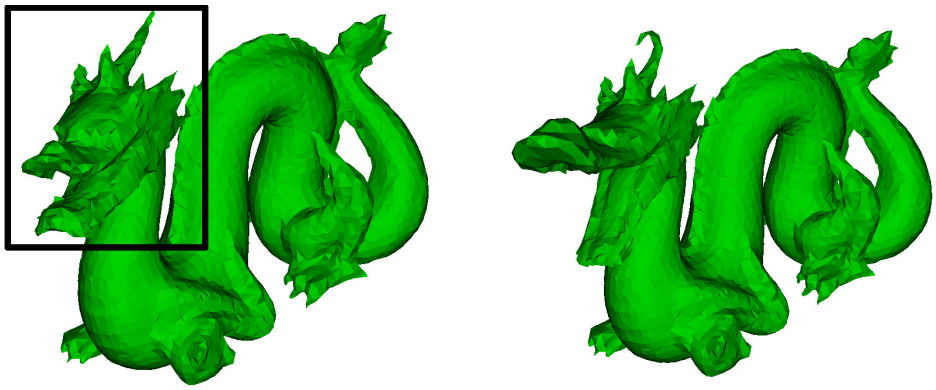


Figure 5.15 – Méthode proposée pour gérer des géométries identiques

Cette méthode est appliquée sur différents maillages. Dans ce qui suit, la technique de localisation des déformations présentée dans le paragraphe 5.1 n'est pas utilisée, le but étant d'illustrer les résultats obtenus avec les fonctions F_α et F_α^σ sur des maillages contenant un grand nombre de nœuds. Trois maillages sont utilisés.

Le premier est le *dragon* de l'université de Stanford (figure 5.16). Le maillage initial est surfacique, contient 5 205 nœuds et une déformation locale est réalisée au niveau de la tête. La modification est transférée sur un second maillage du *dragon*, plus fin, contenant 437 645 points. Les valeurs pour les paramètres α et σ sont fixées respectivement à 1000 et 100. Ils permettent d'aboutir à des résultats lisses en un temps de calcul acceptable, comme nous le verrons par la suite dans ce paragraphe. Ils pourraient être définis plus finement de manière expérimentale mais l'objectif de ces illustrations n'est pas de trouver leur meilleure valeur pour un maillage précis. Les résultats obtenus par les fonctions F_α et F_α^σ sont illustrés sur la figure 5.18 et la figure 5.19. On remarque alors qu'ils sont globalement très similaires. Toutefois, deux différences apparaissent. La première se trouve au dessus de la tête où la pointe n'est pas reconstruite de la même façon. Il n'est toutefois pas possible d'indiquer laquelle est la plus proche du résultat attendu, les deux étant acceptables. La seconde différence porte sur le temps de calcul pour réaliser



(a) M_{1i} (la zone déformée est mise en évidence par le rectangle)

(b) M_{1d}

Figure 5.16 – Maillages initial et déformé du *dragon* de Stanford

le transfert complet. F_α nécessite $8min\ 34,5s$ tandis que F_α^σ $2min\ 16,5s$, soit un temps $3,8$ fois plus faible.

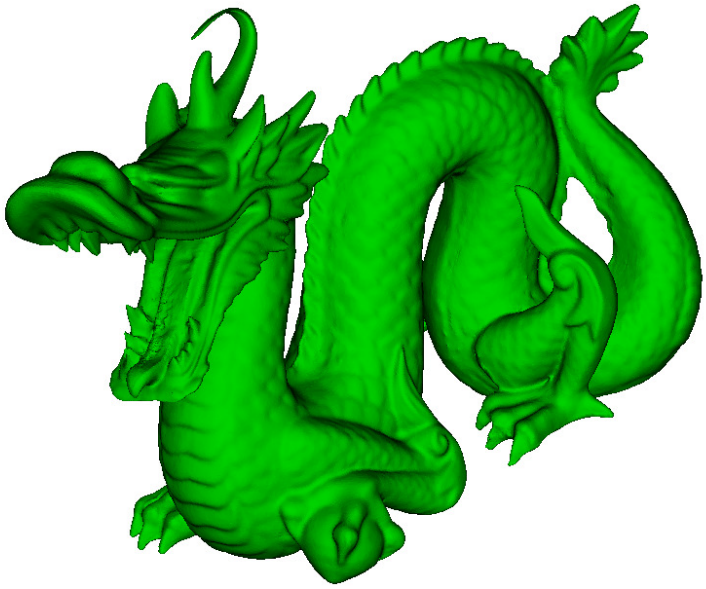
Le second transfert de déformations réalisé porte sur le maillage *Armadillo* de Stanford. Le maillage initial est surfacique et contient 10 000 nœuds. Diverses modifications lui ont été appliquées et la déformation finale est globale à l'ensemble du maillage. Le modèle a été globalement aminci et la main droite, l'oreille gauche et la carapace (au niveau du dos) agrandi. Les maillages M_{1i} et M_d sont illustrés sur la figure 5.17.

(a) M_{1i}

(b) M_{1d}

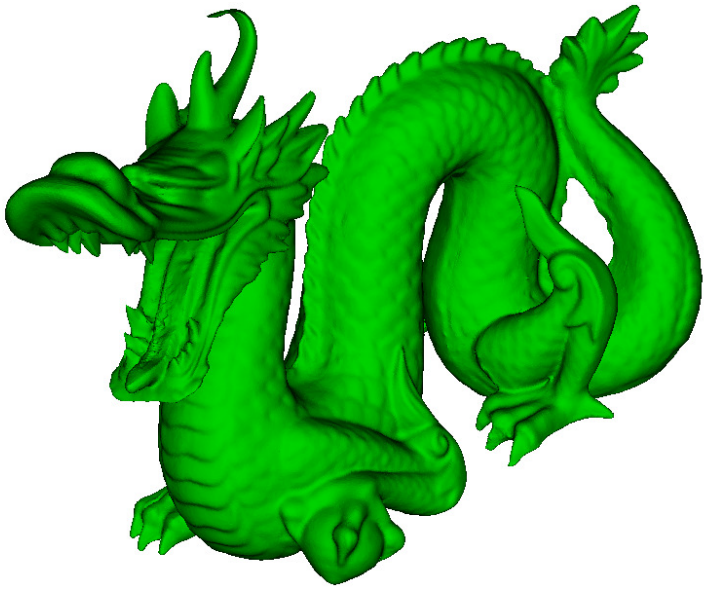
Figure 5.17 – Maillages initial et déformé *Armadillo* de Stanford

Le transfert est réalisé sur un second maillage contenant 172 974 nœuds. Pour les mêmes raisons que précédemment, les paramètres α et σ sont fixées respectivement à 1000 et 100. Le résultat est illustré sur la figure 5.21 pour F_α et la figure 5.22 pour F_α^σ. On remarque alors que les calculs aboutissent à des maillages très proches. La



Opération	Nombre de nœuds	Temps de calcul
Calcul de F_{1000}	dans M_{1i} : 5 205	4,0s
Application de F_{1000}	à déplacer : 437 645	8min 30,5s

Figure 5.18 – Fonction F_α sur le maillage *dragon* de Stanford

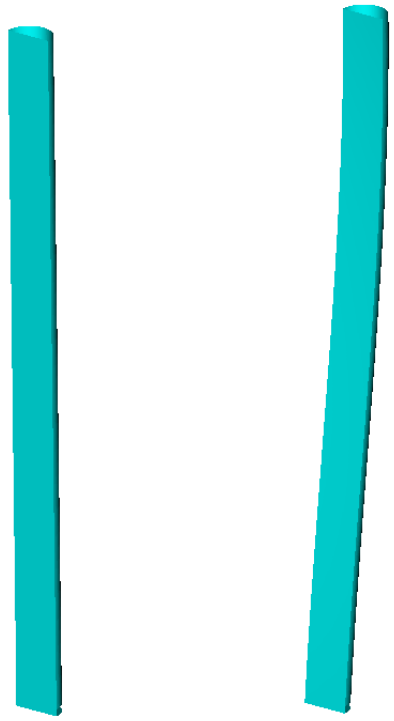


Opération	Nombre de nœuds	Temps de calcul
Calcul de F_{1000}^{100}	dans M_{1i} : 5 205	6,0s
Application de F_{1000}^{100}	à déplacer : 437 645	2min 10,5s

Figure 5.19 – Fonction F_α^σ sur le maillage *dragon* de Stanford

principale différence porte sur le temps de calcul total, avec $2min\ 56,0s$ pour F_α et $1min\ 18,5s$ pour F_α^σ, soit un temps $2,25$ fois plus faible. La gain de temps moins important pour la fonction F_α^σ par rapport au maillage précédent s'explique par le fait que la fonction F_α^σ est surtout plus rapide que F_α pour l'étape de calcul de la nouvelle position des points de M_{2i}. Le second maillage *Armadillo* contenant moins de points que le second du *dragon*, le gain est plus faible.

Le troisième maillage utilisé pour tester les fonctions est moins complexe que les deux premiers mais correspond à un modèle plus orienté CAO. Il s'agit d'un demi-tube maillé finement (354 854 nœuds). Ce dernier est modifié de manière à ce que la déformation réalisée fasse intervenir un grand nombre de points. Les maillages utilisés sont représentés sur la figure 5.20. Le transfert est réalisé sur le maillage



(a) M_{1i} (b) M_{1d}

Figure 5.20 – Maillages initial et déformé d'un demi-tube CAO

M_{1i}. Il est donc possible de connaître la qualité du résultat en comparant point à point le maillage obtenu et M_{1d}. Pour illustrer l'influence des paramètres, quatre tests sont réalisés en utilisant chacun avec une fonction de transfert particulière. Afin d'utiliser un grand nombre de points caractéristiques et montrer l'impact sur le temps de calcul, la première est une F_α, où $\alpha = 5000$. Les trois suivantes sont toutes des F_α^σ avec des paramètres différents : $F_{\alpha=5000}^{\sigma=500}$ pour comparer avec F_α, $F_{\alpha=5000}^{\sigma=3000}$ et $F_{\alpha=10000}^{\sigma=1800}$ pour montrer la précision obtenue par F_α^σ lorsque le temps de calcul est le même que celui de F_α.

Opération	Nombre de nœuds	Temps de calcul
Calcul de F_{1000}	dans M_{1i} : 10 000	3,0s
Application de F_{1000}	à déplacer : 172 974	2min 53,0s

Figure 5.21 – Fonction F_α sur le maillage *Armadillo* de Stanford

Opération	Nombre de nœuds	Temps de calcul
Calcul de F_{1000}^{100}	dans M_{1i} : 10 000	6,0s
Application de F_{1000}^{100}	à déplacer : 172 974	1min 12,5s

Figure 5.22 – Fonction F_α^σ sur le maillage *Armadillo* de Stanford

Les temps de calcul globaux sont détaillés dans le tableau 5.5. $F_{\alpha=5000}$ nécessite $17min\ 16s$ alors que $F^{\sigma=500}_{\alpha=5000}$ seulement $4min\ 38s$, soit un temps $3,7$ fois plus faible. Comme souhaité, les deux autres fonctions F^{σ}_{α} ont des temps de calcul similaires à $F_{\alpha=5000}$.

	Opération	Nombre de nœuds	Temps de calcul
$F_{\alpha=5000}$	Calcul	dans M_{1i} : 354 854	$2min\ 35,0s$
	Application	à déplacer : 354 854	$14min\ 41,0s$
			Total : $17min\ 16,0s$
$F^{\sigma=500}_{\alpha=5000}$	Calcul	dans M_{1i} : 354 854	$1min\ 00,0s$
	Application	à déplacer : 354 854	$3min\ 38,0s$
			Total : $4min\ 38,0s$
$F^{\sigma=3000}_{\alpha=5000}$	Calcul	dans M_{1i} : 354 854	$4min\ 13,0s$
	Application	à déplacer : 354 854	$13min\ 34,0s$
			Total : $17min\ 47,0s$
$F^{\sigma=1800}_{\alpha=10000}$	Calcul	dans M_{1i} : 354 854	$6min\ 42,0s$
	Application	à déplacer : 354 854	$11min\ 09,0s$
			Total : $17min\ 51,0s$

Table 5.5 – Temps de calcul des quatre fonctions utilisées pour le transfert sur le maillage du demi-tube CAO

Les distances euclidiennes obtenues en comparant M_{1d} avec le résultat des quatre fonctions sont illustrés sur la figure 5.23. $F_{\alpha=5000}$ a une erreur maximale de l'ordre de 10^{-5}. On remarque alors que le résultat obtenu par la fonction $F^{\sigma=500}_{\alpha=5000}$ est moins précis, la différence étant de deux ordres de grandeur. Pour les deux autres fonctions F^{σ}_{α}, la qualité du résultat est proche de celui avec $F_{\alpha=5000}$.

Dans le cas où une précision de 10^{-3} est suffisante pour l'utilisateur, la fonction $F^{\sigma=500}_{\alpha=5000}$ doit être préférée car elle permet d'obtenir un résultat plus rapidement. Sinon, l'une ou l'autre des fonctions F_{α} ou F^{σ}_{α} peut être utilisée. L'avantage de F_{α} est de ne pas avoir de second paramètre à déterminer, celui de F^{σ}_{α} est de pouvoir réduire le temps de calcul en acceptant un résultat moins précis.

Dans l'ensemble de ces trois transferts, les deux fonctions ont permis de montrer les bons résultats qu'il est possible d'obtenir en un temps faible au regard du nombre de points utilisés. De plus, les résultats sont lisses sans montrer de discontinuités ou d'artéfacts. Nous allons maintenant détailler le second mode de transfert de déformations portant sur des maillages issus de géométries adjacentes.

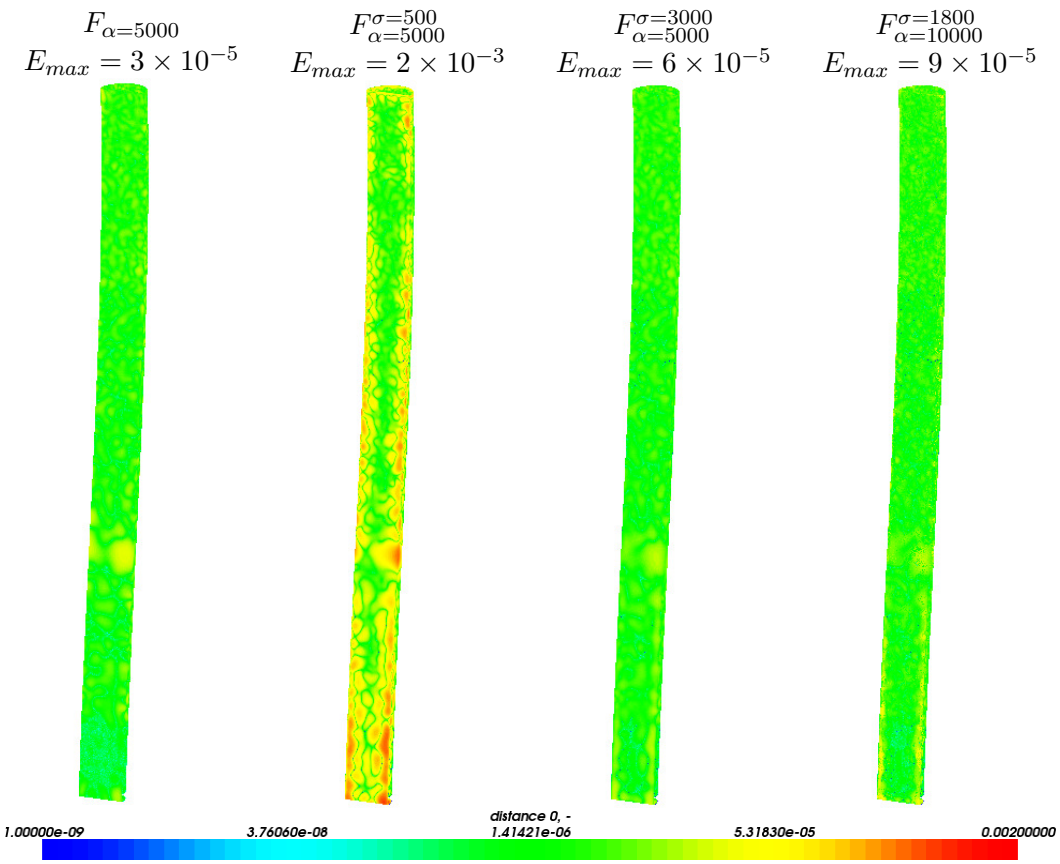


Figure 5.23 – Qualité du transfert sur le maillage du demi-tube CAO

5.3.2 Entre des représentations de géométries adjacentes

Le but de ce transfert est de modifier les bords communs aux géométries adjacentes et d'imposer une contrainte de non déplacement aux autres. Pour cela, nous avons mis en place une méthode qui est détaillée dans ce qui suit.

La géométrie complète de l'étude proposée par la figure 5.24 fait intervenir deux objets A et B adjacents. Le premier traitement s'appuie sur l'objet A. Ce dernier est donc discrétisé ($1a$) et le maillage obtenu est utilisé pour le calcul. Dans le cas présent, il s'agit de la définition de la déformation géométrique ($1b$). Une ou plusieurs fonctions de transfert permettent d'interpoler les modifications géométriques intervenues sur le maillage initial ($1c$).
Le maillage utilisé par la seconde tâche est obtenu après discrétisation de l'objet B ($2a$). Avant de pouvoir lancer le second traitement, le maillage doit être mis à jour. Pour cela, ses frontières sont extraites ($2b$). Cette opération est réalisée en ne gardant que les faces qui ont précédemment été définies comme "frontière" lors de la discrétisation de la géométrie. Les frontières communes au bord et au maillage de l'étape $1a$ sont ensuite extraites et déplacées par l'intermédiaire des fonctions précédemment calculées ($2c$). Il est ainsi possible de connaître le déplacement de l'ensemble des frontières du maillage de l'objet B ($2d$). Une dernière fonction de transfert est calculée de manière à interpoler l'évolution de la position des bords de ce maillage ($2e$). Une fois appliquée à la totalité des nœuds, le maillage de l'objet B est complètement modifié ($2f$).

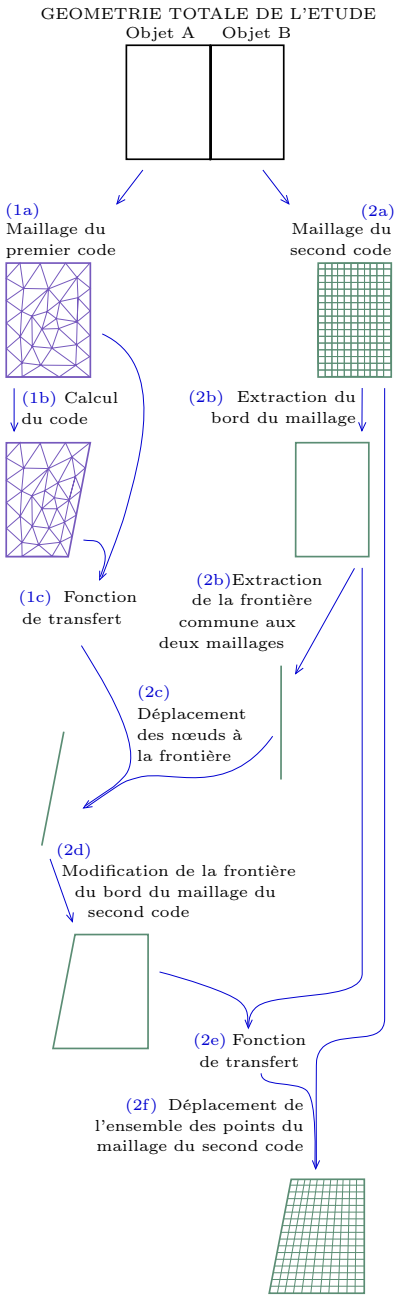


Figure 5.24 – Méthode proposée pour gérer des géométries adjacentes

Pour illustrer la méthode, un exemple est présenté dans ce qui suit. Il permet de mettre en application l'approche présentée et de la valider sur un cas simple. Elle fait l'objet d'une étude plus complète ensuite dans le chapitre 6 où sa mise en application est détaillée sur un cas pratique proposé par le CEA.

L'exemple sommaire correspond au transfert de déformations entre deux maillages issus de géométries adjacentes (figure 5.25). Chacun discrétise un demi-cube dont une des faces est partagée par les deux. Une déformation est appliquée au maillage de droite, en vert, puis est transférée sur le maillage de gauche, en violet (figure 5.26). Pour cela, différentes étapes sont nécessaires. Tout d'abord, une première fonction est calculée pour prendre l'empreinte de la modification géométrique survenue sur le maillage de droite, entre M_{1i} et M_{1d}. Les paramètres α et σ sont fixés respectivement à 2 000 et 500. La fonction est utilisée pour déplacer le bord de M_{2i} commun à M_{1i}. Pour connaître le déplacement des nœuds internes au maillage M_{2i}, une seconde fonction de transfert est calculée. Elle prend une empreinte de la déformation des bords du maillage. Les paramètres utilisés sont $\alpha = 15\ 000$ pour la fonction F_α et $\alpha = 5\ 000$ et $\sigma = 1\ 000$ pour F_α^σ. La valeur de α n'est pas la même pour les deux fonctions car F_α nécessite plus de points d'accroche que F_α^σ pour obtenir un résultat similaire. La déformation étant de grande amplitude, F_α doit être contrainte par plus de points caractéristiques en raison de son support global. Les temps de calcul de ces différentes étapes sont regroupés dans le tableau 5.6 pour F_α et le tableau 5.7 pour F_α^σ. Ces deux fonctions présentent une nette différence de temps de calcul qui s'explique par le choix du paramètre α pour la seconde fonction de transfert utilisée. Les résultats obtenus avec les deux fonctions sont très similaires. Dans les deux cas, ils correspondent à la figure 5.26(a). Pour illustrer

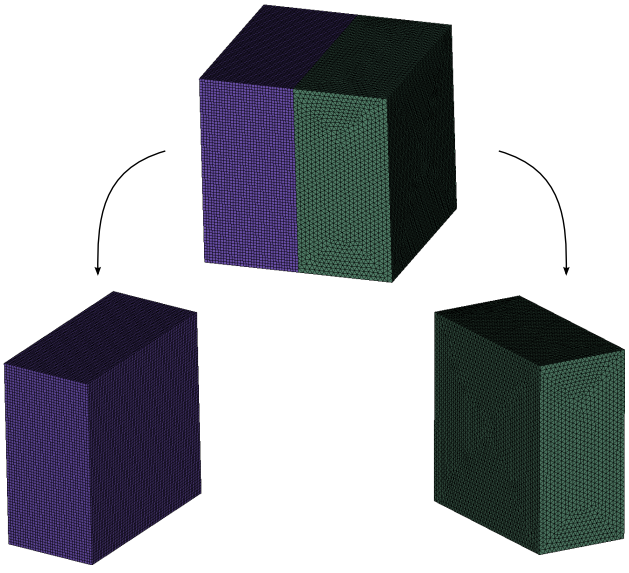


Figure 5.25 – Maillages issus de géométries adjacentes (M_{1i} à droite et M_{2i} à gauche)

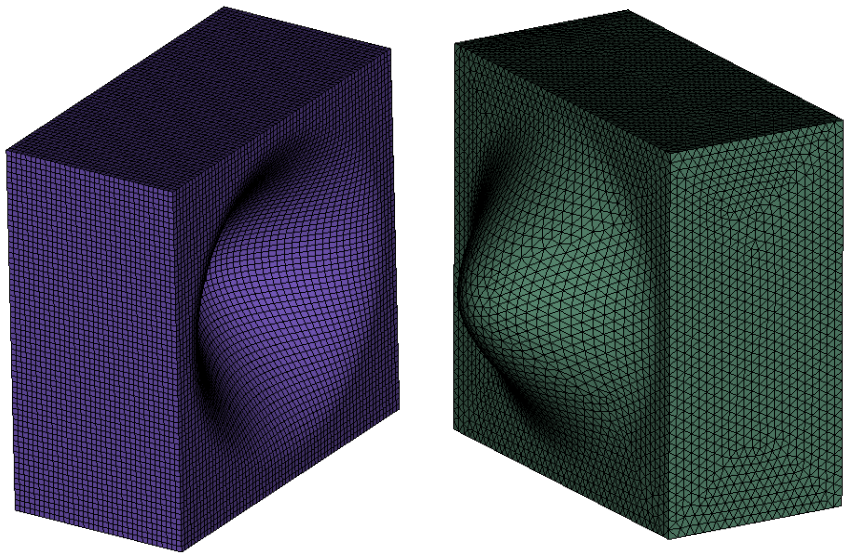
Opération	Nombre de nœuds	Temps de calcul
Calcul de F_{2000}	dans M_{1i} : 1 959	$21,0s$
Application de F_{2000}	à déplacer : 4 624	$1min\ 38,5s$
Calcul de F_{15000}	dans M_{2i} : 18 092	$56min\ 13,5s$
Application de F_{15000}	à déplacer : 143 748	$43min\ 46,5s$

Table 5.6 – Temps de calcul du transfert de déformations avec F_α entre deux cubes adjacents

Opération	Nombre de nœuds	Temps de calcul
Calcul de F_{2000}^{500}	dans M_{1i} : 1 959	$42,0s$
Application de F_{2000}^{500}	à déplacer : 4 624	$1min\ 42,0s$
Calcul de F_{5000}^{1000}	dans M_{2i} : 18 092	$10min\ 14,5s$
Application de F_{5000}^{1000}	à déplacer : 143 748	$11min\ 13,0s$

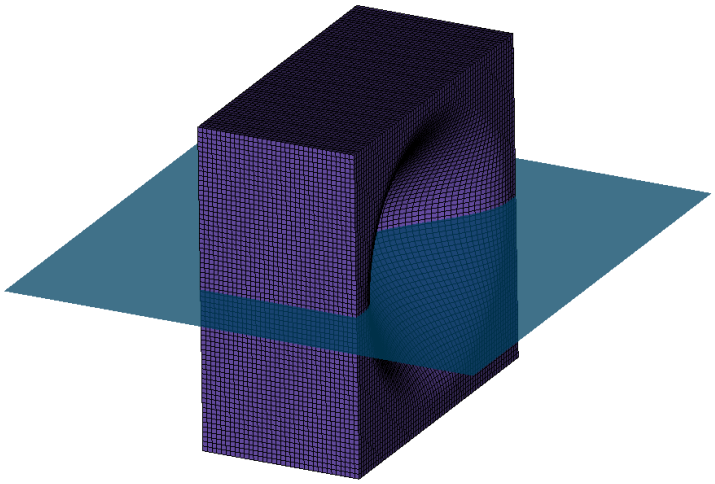
Table 5.7 – Temps de calcul du transfert de déformations avec F_α^σ entre deux cubes adjacents

la répartition des nœuds internes aux maillages, une coupe axiale est effectuée. Le résultat est illustré sur la figure 5.27. Comme on peut le constater, la fonction F_α permet d'obtenir une répartition très homogène du nuage de points alors que F_α^σ a tendance à peu déplacer les points éloignés des frontières modifiées. Cet effet est lié à la taille du support utilisé pour la fonction. Une valeur de σ faible induira un calcul avec de petites zones d'influence autour des points caractéristiques. Les

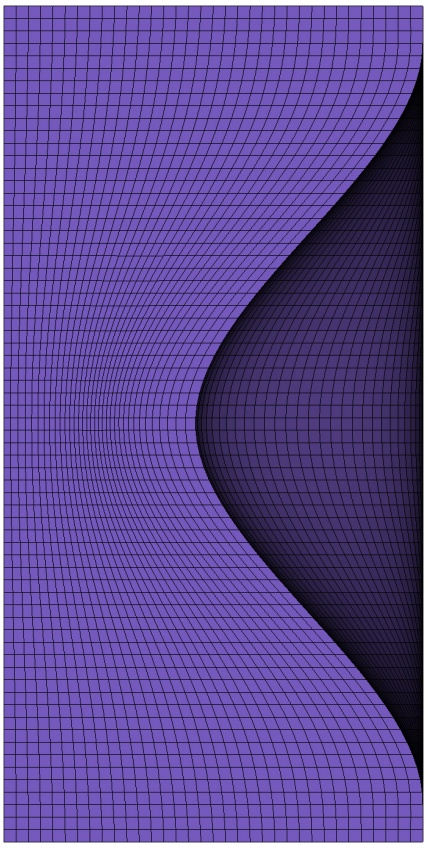


(a) M_{2c} calculé par la fonction F_α^σ (b) M_{1d}

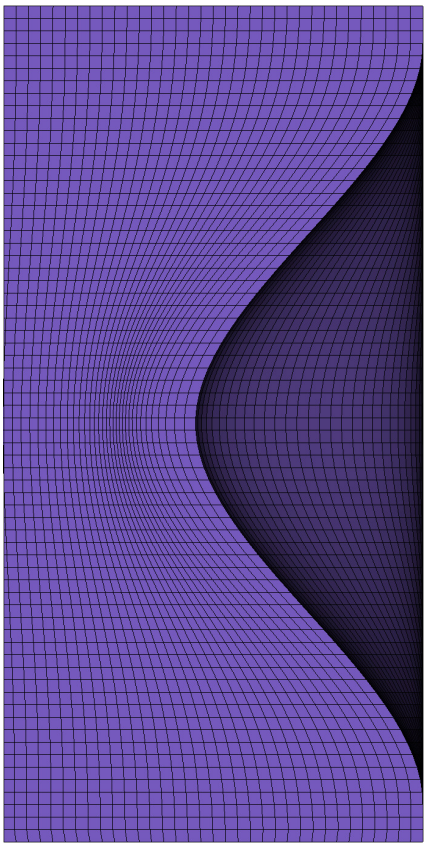
Figure 5.26 – Maillages déformés des deux demi-cubes adjacents



(a) coupe axiale réalisée



(b) résultat de F_α



(c) résultat de F_α^σ

Figure 5.27 – Coupe axiale du résultat du transfert d'une déformation entre deux cubes adjacents (vue de dessus)

points se trouvant loin des parties déformées seront alors tassés par rapport au reste du maillage. Pour une valeur de σ plus importante, les points caractéristiques interviennent dans le calcul de plus de points et le comportement global de la fonction se rapproche de F_α.

5.4 Synthèse

Les méthodes de transfert de déformations peuvent s'appuyer sur un très grand nombre de points. Afin d'éviter qu'interviennent les parties non déformées, les zones de calcul sont limitées par la méthode de localisation des déformations présentée dans le paragraphe 5.1. Deux nouvelles fonctions, F_α et F_α^σ, sont ensuite définies dans le paragraphe 5.2 pour permettre de gérer des maillages volumineux. Pour cela, elles s'appuient sur une méthode de simplification qui vise à restreindre le support de l'interpolation en ne gardant que les α nœuds importants. Une étape d'interpolation est ensuite utilisée pour prendre une empreinte de la déformation. Les deux fonctions définies peuvent être utilisées indépendamment. Les résultats illustrés dans le paragraphe 5.3 montrent qu'il est possible d'utiliser l'une ou l'autre des deux fonctions pour réaliser un transfert de déformations géométriques en un temps raisonnable en comparaison du temps de calcul d'un code s'appuyant sur les mêmes maillages.

L'ensemble de notre travail est appliqué sur un exemple concret. Le chapitre 6 le détaille et montre l'impact de la prise en compte des déformations géométriques dans un couplage de codes de calcul.

Chapitre 6

Mise en application

Sommaire

6.1 Cas d'application 115
6.1.1 Réacteur Jules Horowitz 115
6.1.2 Dispositifs expérimentaux 116
6.1.3 Présentation du cas d'application 117
6.2 Couplage réalisé 117
6.2.1 Calcul thermo-mécanique 121
6.2.2 Première étape intermédiaire 123
6.2.3 Calcul thermo-hydraulique 124
6.2.4 Seconde étape intermédiaire 126
6.2.5 Arrêt du couplage 126
6.3 Apport du transfert des déformations géométriques 127
6.4 Résultats obtenus 127
6.4.1 Maillage du fluide 127
6.4.2 Temps de calcul 129
6.4.3 Température du fluide 130
6.4.4 Simulation de l'écoulement du fluide 131
6.5 Synthèse 132

Les précédents chapitres ont apporté une solution au problème du transfert de déformations entre des maillages issus d'une même géométrie ou de géométries adjacentes. Le travail présenté dans ce qui suit vise à la mettre en application sur un cas concret. Tout d'abord, le contexte est exposé succinctement dans le paragraphe 6.1. Il porte sur les dispositifs expérimentaux du futur réacteur de recherche Jules Horowitz. L'étude réalisée est la simulation de l'écoulement d'un fluide entre deux solides. Ces derniers sont sujets à des modifications de leur géométrie, impliquant des effets sur l'écoulement. Ce calcul est réalisé par un couplage de codes qui est détaillé dans le paragraphe 6.2. Afin d'illustrer l'impact du transfert des déformations, un second couplage est effectué. Les différences avec le premier sont présentées dans le paragraphe 6.3. Les résultats obtenus avec chacun de ces deux calculs sont exposés dans le paragraphe 6.4.

6.1 Cas d'application

Les réacteurs de recherche (*MTR, Material Testing Reactor*) sont des outils indispensables pour la sûreté des installations actuelles, comme pour l'allongement de la durée de vie des centrales ou l'étude d'incidents et d'accidents ([BG04, Bac05]). Ils sont également utilisés pour la conception des futurs réacteurs de puissance afin de valider les choix des matériaux et des combustibles.

En raison du vieillissement de l'actuel réacteur de recherche français OSIRIS implanté sur son site de Saclay, le CEA a fait le choix d'en construire un nouveau d'envergure européenne, le Réacteur Jules Horowitz [10] (RJH), sur le site de Cadarache.

6.1.1 Réacteur Jules Horowitz

Le RJH ([Ira06]) est un réacteur de recherche de type piscine, c'est-à-dire qu'il est plongé dans une grande quantité d'eau pour le refroidir. Présenté sur la figure 6.1, il est constitué d'un cœur où se déroulent les fissions nucléaires. Autour de ce dernier est placé un réflecteur dont le rôle est de limiter les fuites de neutrons à l'extérieur du cœur. Le cœur est contenu dans un caisson à l'intérieur duquel circule de l'eau sous pression qui constitue le circuit primaire du réacteur. Le réflecteur est, quant à lui, baigné dans l'eau de la piscine qui correspond au circuit secondaire entourant le caisson.

Comme l'illustre la figure 6.2, le RJH permet de produire des isotopes radioactifs pour la médecine et d'étudier de nouveaux combustibles et matériaux. Ces expérimentations sont réalisées dans des dispositifs expérimentaux qui peuvent être placés dans le cœur du réacteur ou dans son réflecteur, en fonction des effets recherchés. En réflecteur, il existe des systèmes à déplacement dans les rues d'eau qui permettent de régler l'énergie reçue par un dispositif en changeant sa position sans avoir à modifier la puissance du cœur.

10. Jules Horowitz (1921-1995), physicien français, fut un des pionniers du CEA pour le développement des réacteurs nucléaires.

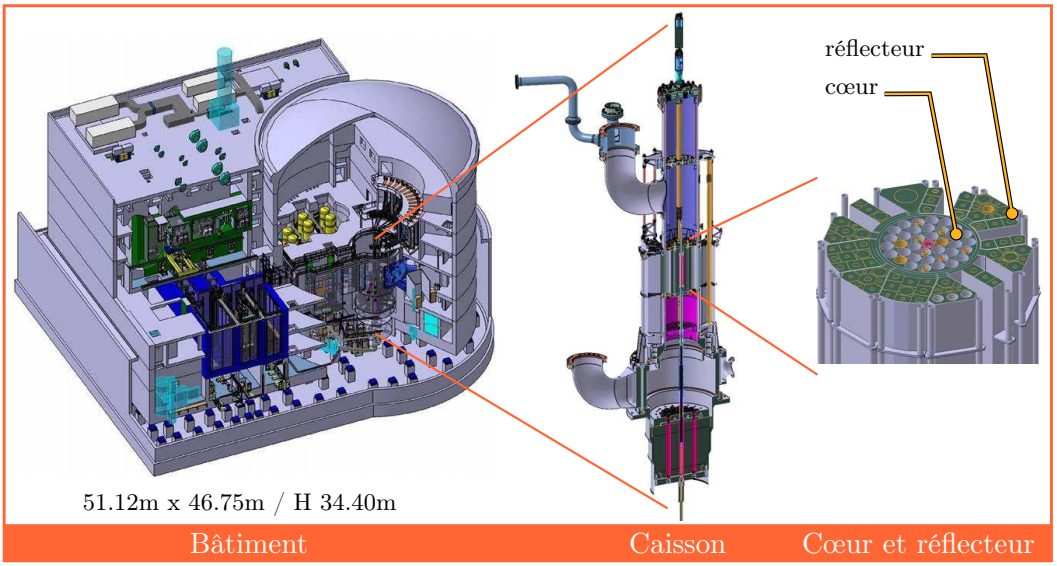


Figure 6.1 – Présentation du Réacteur Jules Horowitz — images extraites du site www.cea.fr

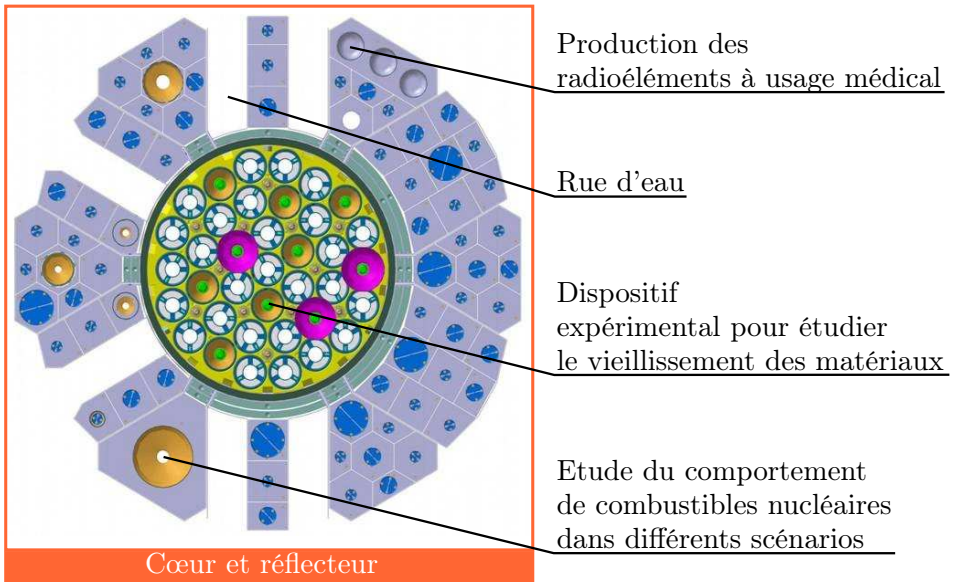


Figure 6.2 – Zoom sur le cœur et le réflecteur du RJH — image extraite du site www.cea.fr

6.1.2 Dispositifs expérimentaux

Bien que les dispositifs expérimentaux puissent être très différents les uns des autres, ils partagent un point commun. Ils sont tous placés dans un objet, appelé tube de force, dont le but est d'isoler l'expérimentation du reste du réacteur. Le but est de confiner la matière nucléaire et de permettre d'avoir une pression et une

température internes différentes de celles externes. Pour cela, le tube de force est constitué de deux parties séparées par une fine lame d'Hélium faisant office d'isolant thermique. Pour cette raison, la partie intérieure est volontairement négligée dans ce qui suit.

La figure 6.3 illustre la configuration générale pouvant contenir un dispositif expérimental en réflecteur. Le tube de force est inséré dans une pièce cylindrique, appelée fourreau, dont le but est de guider sa descente et d'assurer son refroidissement. Il est vissé sur la partie haute. Le fourreau est quant à lui maintenu en haut et en bas. Il est plongé dans l'eau de la piscine du réacteur. Des orifices en haut permettent à cette eau de passer entre le tube de force et le fourreau pour pouvoir les refroidir.

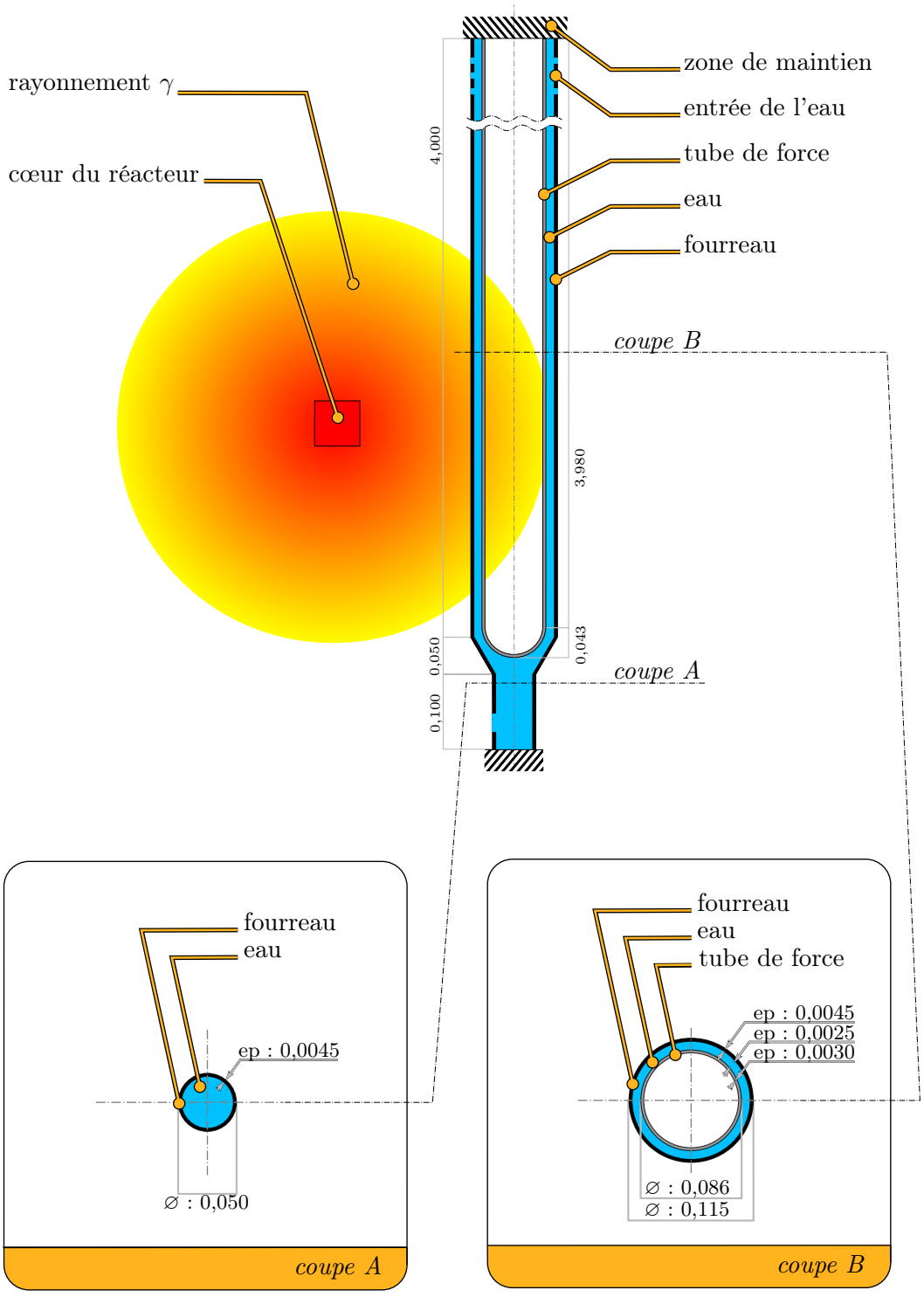
Parmi les rayonnements émis par le cœur du réacteur, une petite partie correspond à une émission de type γ qui se dépose dans les matériaux et les échauffe. Il est donc nécessaire de les refroidir. L'augmentation de la température du tube de force et du fourreau induit une dilatation et donc une déformation de ces pièces. La section de passage de l'eau se voit alors modifiée. L'importance du rayonnement γ observé s'atténue avec la distance ce qui induit des différences de modification des objets en fonction de leur position. Les éléments les plus proches du réacteur ont tendance à plus se dilater que ceux qui sont plus éloignés. Une étude de l'impact de ces déformations sur le refroidissement des dispositifs expérimentaux est donc nécessaire.

6.1.3 Présentation du cas d'application

Pour réaliser cette étude, il est nécessaire de prendre en compte les deux phénomènes liés à la température des solides. D'une part, ils sont chauffés par le rayonnement γ, d'autre part, ils sont refroidis par l'eau en mouvement les séparant. Un code de thermo-mécanique est utilisé pour calculer la température et le déplacement du tube de force et du fourreau. En revanche, il ne modélise pas les échanges thermiques qui se produisent entre les parties solide et fluide. Un code de thermo-hydraulique intervient pour calculer la température, la pression et la vitesse de l'eau en mouvement. La température aux parois avec les solides doit être transmise au code de thermo-mécanique. Après calcul, celui-ci doit renvoyer, au code de thermo-hydraulique, le flux de chaleur à travers les parois ainsi que le déplacement des objets. L'interaction des deux disciplines physiques implique la mise en place d'un couplage des deux codes dans le cadre d'un processus itératif.

6.2 Couplage réalisé

Le calcul réalisé par le couplage mis en place recherche un état permanent. Il s'appuie sur le modèle commun présenté dans le chapitre 3 et modélise la géométrie de l'étude détaillée dans la figure 6.3. On remarque que le tube de force et le fourreau ont tous deux une hauteur voisine de 4 m. Pour éviter des calculs inutiles, on souhaite limiter la partie étudiée à la zone subissant des modifications géométriques.



∅ : diamètre extérieur *ep* : épaisseur

Figure 6.3 – Schéma du cas d'application (l'unité utilisée est le mètre)

La déformation induite par la dilatation est différente selon l'objet. Concernant le tube de force, ce dernier est maintenu en haut et libre de tout mouvement en bas. Il est échauffé sur les derniers 80 cm et ne se dilate qu'à cet endroit. Sa déformation est donc localisée sur sa partie basse, c'est pourquoi seul le dernier mètre est discrétisé. De même, le fourreau est uniquement échauffé sur sa partie basse. La dilatation est donc localisée dans la même zone. Cependant, le fourreau est fixé en haut et en bas. Ainsi, la déformation est globale et il est nécessaire d'utiliser un maillage complet de sa géométrie.

Concernant les deux objets, la dilatation est due à l'énergie déposée dans les matériaux par le rayonnement γ. Comme l'illustre la figure 6.4, où une vue de dessus est proposée, le problème présente un plan de symétrie vertical, c'est pourquoi seule la moitié de la géométrie est modélisée (la partie supérieure sur la figure 6.4).

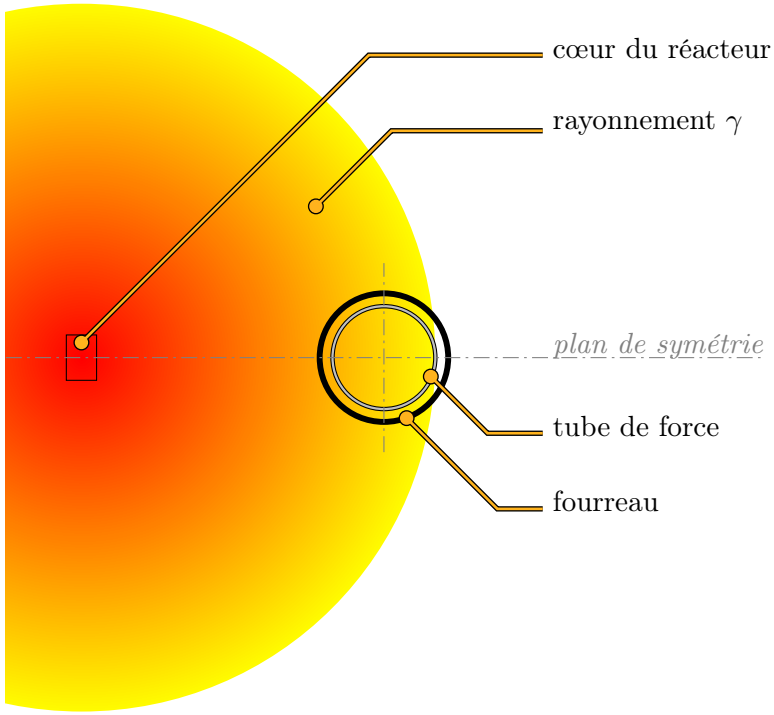


Figure 6.4 – Schéma du cas d'application (coupe vue de dessus)

Les déformations obtenues sont schématisées sur la figure 6.5 où seul le dernier mètre est illustré. Elles ont été amplifiées pour rendre les phénomènes visibles. La géométrie nominale du tube de force et du fourreau est représentée en partie pleine en gris et leur position après déformation est illustrée par le contour rouge. La largeur de la lame d'eau à l'état initial entre les deux solides est de 2.5 mm. Suite aux modifications géométriques induites par leur dilatation, cette largeur change. Elle vaut, sur la figure 6.5, 3.2 mm ($+0.7$ mm) en Ⓐ, 1.8 mm (-0.7 mm) en Ⓑ, 4.0 mm ($+1.5$ mm) en Ⓒ et 1.0 mm (-1.5 mm) en Ⓓ. Comme le montrent les flèches, la partie basse du tube de force se déplace vers le côté opposé du cœur du

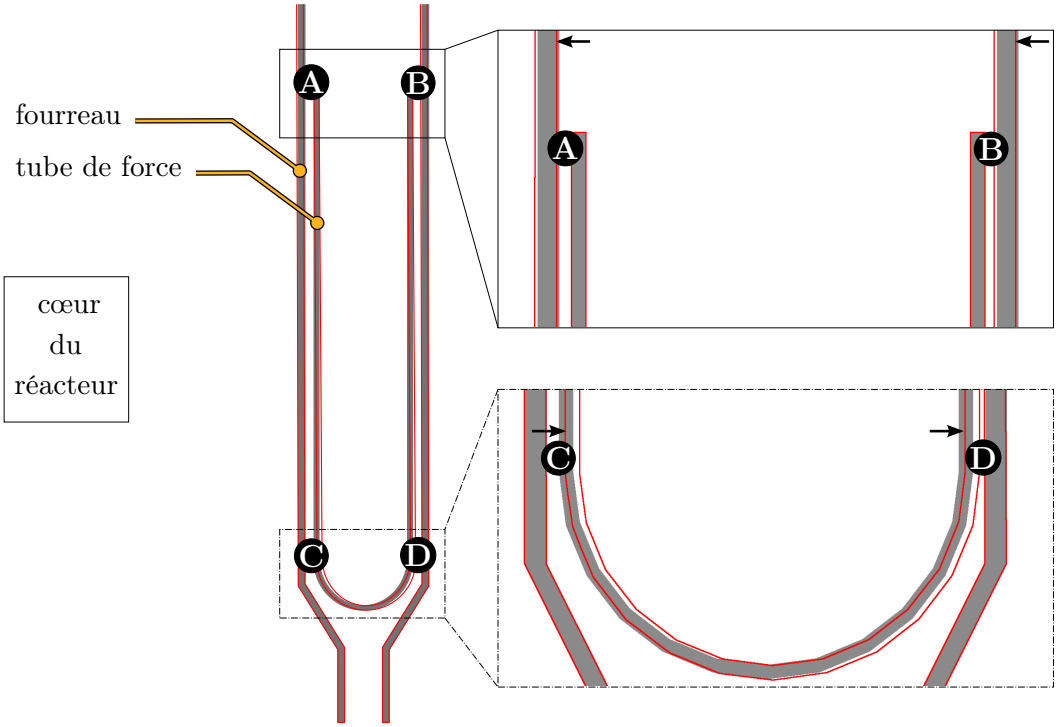


Figure 6.5 – Schéma du cas test avec et sans prise en compte des déformations

réacteur. La variation de la position du fourreau est globale et l'objet a tendance à fléchir du côté du cœur. Les conséquences sont un éloignement des parois des deux solides dans la partie proche du cœur et un rapprochement à l'opposé.

Pour réaliser cette simulation, la méthode de couplage permettant un transfert de déformations entre codes, présentée dans la paragraphe 3.1, est utilisée. Le code de comportement thermo-mécanique (❺ sur la figure 3.1) utilisé est Cast3m ([VMCC89]), celui de thermo-hydraulique (❾ sur la figure 3.1) est Trio_U ([BH00, CCE02]). Il s'agit de codes déterministes du CEA qui n'avaient jamais été couplé avant. Les nombreuses difficultés rencontrées avec ces codes ont été résolues grâce à la contribution des équipes de développement. Les données échangées dans ce couplage sont localisées aux interfaces entre le fluide et les deux solides, c'est-à-dire aux parois intérieure du fourreau et extérieure du tube de force. La modification géométrique calculée par Cast3m doit être transférée sur le maillage utilisé par Trio_U. Pour cela, la méthode relative au transfert de déformations entre des représentations géométriques adjacentes (paragraphe 5.3.2) est utilisée.

Afin de pouvoir contrôler les évolutions des simulations numériques, un critère de convergence commun est défini. Il s'appuie sur un champ C, un vecteur de données attaché à des positions sur un maillage (nœuds, barycentre des éléments, ...),

contenant n valeurs réelles. On note C_j sa $j^{ième}$ valeur, et $\|C\|_2$ sa norme L_2 :

$$\|C\|_2 = \sqrt{\sum_{j=1}^{n} C_j^2} \tag{29}$$

On note $C^{[k]}$ l'état du champ C à l'itération k. Le critère de convergence utilisé, noté $\|\Delta C\|_2$, est la différence normalisée en norme L_2 entre deux états successifs de ce champ. Il correspond à :

$$\|\Delta C\|_2^{[k]} = \frac{\|C^{[k]} - C^{[k-1]}\|_2}{\|C^{[k]}\|_2} \tag{30}$$

Le couplage réalisé suit le schéma présenté dans la figure 3.1.

Les étapes initiales correspondent à la création des maillages initiaux (❷) à partir de la géométrie (❶) du modèle commun. Ce travail est rendu possible en s'appuyant sur la plateforme SALOME (*c.f.* paragraphe 2.1.3). Grâce au composant GEOM, la géométrie complète est créée de manière paramétrique. En s'appuyant dessus, deux géométries spécifiques sont définies, une par code de calcul. La première modélise les deux solides où les trois mètres supérieurs du tube de force sont supprimés. La seconde représente le mètre inférieur de la partie fluide. Les deux maillages sont réalisés grâce au composant SMESH de SALOME en utilisant un mailleur interne (Hexahedron) pour la partie solide et le mailleur NetGen pour la partie fluide.

Les étapes suivantes peuvent être regroupées en quatre phases principales : calcul thermo-mécanique (❺), intégration des résultats dans le modèle commun et préparation du calcul Trio_U (❻, ❼, ❽), calcul thermo-hydraulique (❾), intégration des résultats dans le modèle commun et préparation du nouveau calcul Cast3m (❸, ❹).

6.2.1 Calcul thermo-mécanique

La quantité d'énergie déposée dans les matériaux par le rayonnement γ étant dégressive en fonction de la distance au cœur (figure 6.3), l'échauffement du tube de force et du fourreau dépend de leur localisation. Pour une température et une position des solides données, un déplacement de ces objets est calculé. Cette modification induit un changement de l'échauffement des solides, et donc de leur température. Le travail réalisé par Cast3m est le résultat d'un couplage interne itératif entre le calcul de la température et celui des déformations géométriques. La résolution de ce problème implique un système non linéaire dont le critère d'arrêt est celui présenté dans le paragraphe 6.2, où le champ utilisé est la température T_{Cast3m} en tout point du maillage des solides. La convergence est atteinte à une itération k telle que $\|\Delta T_{Cast3m}\|_2^{[k]} \leq 10^{-6}$. Cette valeur est proposée par les experts du CEA qui estiment que le système est suffisamment stable pour conclure à un résultat. Cela signifie que la variation de la température est devenue négligeable et n'entraine plus de changement significatif dans la position des solides.

L'énergie déposée, due au rayonnement γ, varie linéairement avec la distance au cœur. A hauteur du cœur, la source est de 4 $W.g^{-1}$ sur le côté extérieur du tube de force au plus proche du cœur et nulle au plus éloigné. Cette énergie déposée dans les matériaux implique une élévation de leur température qui est calculée par le code de comportement thermo-mécanique en tout point de son maillage. En raison de la forme des solides, un maillage hexaédrique est utilisé. Il est plus fin dans la direction des gradients de la température dont sont issus les flux thermiques.

Pour réaliser ce calcul, des conditions limites doivent être fixées. La paroi extérieure du fourreau est en contact direct avec l'eau de la piscine du réacteur à 38 $°C$. Le côté intérieur du fourreau et le côté extérieur du tube de force sont séparés par une lame d'eau provenant de la piscine du réacteur. Initialement, cette donnée n'est pas connue. Elle est donc imposée à 38 $°C$, la température de la piscine, et sera mise à jour au fur et à mesure du couplage par le résultat du code de comportement thermo-hydraulique Trio_U. La paroi interne du tube de force est en contact avec la lame d'Hélium faisant office d'isolant thermique. Le flux thermique sur cette paroi est donc spécifié comme étant nul (aucun échange d'énergie). L'ensemble des conditions limites sont regroupées dans la figure 6.6 et le maillage des deux solides y est schématisé.

itération	*température*	*température*	*température*	*flux*
première	38°C	38°C	38°C	*nul*
suivantes	38°C	*résultat Trio_U*	*résultat Trio_U*	*nul*

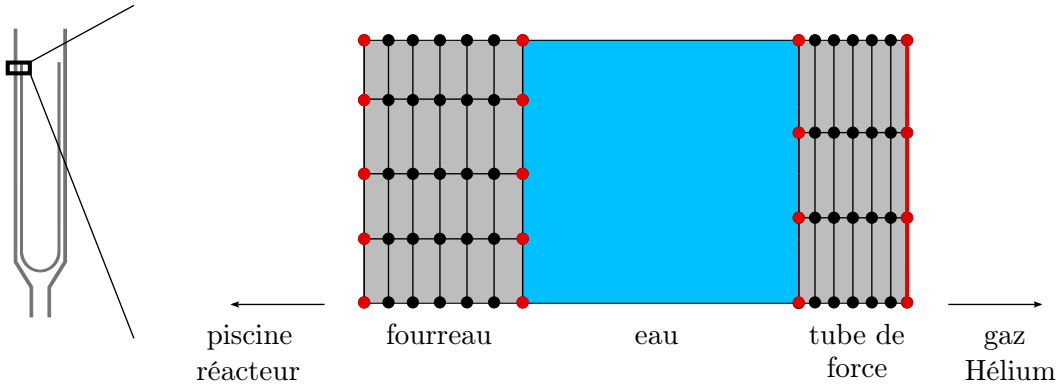


Figure 6.6 – Conditions limites du calcul de comportement thermo-mécanique

A partir de ces informations, Cast3m est à même de réaliser son calcul. Pour cela, il s'appuie sur le maillage des deux objets discrétisant la géométrie nominale de l'étude. Ce dernier est volumique, constitué de 7 956 hexaèdres et possède 10 206 nœuds. Les résultats obtenus sont la température interne des matériaux, à partir de laquelle est déduit le flux thermique aux parois, et le maillage déformé correspondant aux déplacements de certains nœuds du maillage initial. Ces deux dernières

informations doivent alors être intégrées au modèle commun pour être disponibles pour le second code de calcul.

6.2.2 Première étape intermédiaire

Le travail du coupleur peut être décomposé en deux phases. La première correspond à l'intégration des résultats obtenus par le premier code dans le modèle commun. En l'état actuel des développements, cette étape est remplacée par un stockage des données dans des fichiers d'échange MED. Le format de stockage de Cast3m est spécifique au code. Un programme particulier, développé par les équipes du CEA, permet de le convertir au format MED. Les informations non géométriques (*i.e.* autres que les déformations) y sont regroupées. Concernant la déformation géométrique, une empreinte de la modification survenue sur le maillage initial doit être calculée. Le choix de la fonction de transfert s'est porté sur F_α^σ décrite au paragraphe 5.2.3. Pour caractériser la déformation, deux fonctions F_α^σ sont utilisées, une par solide. La première correspond au fourreau dont le support contient 7 917 nœuds, la seconde au tube de force regroupant 2 289 points. Les paramètres de la fonction sont fixés à $\alpha = 2\ 000$ et $\sigma = 200$. Des valeurs plus faibles impliquent une mauvaise reconstruction lors du déplacement des parois fluides. Des valeurs plus élevées aboutissent à un résultat très similaire tout en augmentant le temps de calcul. Une fois ces calculs terminés, les deux fonctions sont stockées dans des fichiers textes contenant l'ensemble des paramètres (*c.f.* paragraphe 5.2.3.4).

La seconde phase permettant le couplage de codes porte sur la fourniture, par le coupleur, des données utilisées par le second programme de simulation numérique. Pour être utilisables, les informations doivent être interpolées sur le maillage utilisé par Trio_U. Un point est important : un champ de valeurs repose sur un maillage donné. Afin de pouvoir réaliser le transfert de champs sur un autre maillage, il est nécessaire que tous deux correspondent à la même géométrie. Dans le cas présent, cela signifie qu'il faut réaliser le transfert des déformations sur le maillage du code Trio_U avant de pouvoir lui interpoler les différents champs utilisés. En effet, le code de déformation Cast3m a précédemment calculé ses valeurs sur son maillage résultat, c'est-à-dire celui modifié. Comme le détaille la figure 5.24, le travail de transfert des déformations est réalisé en deux temps. Tout d'abord, les fonctions F_α^σ calculées sont récupérées et appliquées sur le maillage utilisé par Trio_U aux différents nœuds de leur support respectif. La surface en contact avec le côté intérieur du fourreau contient 74 293 nœuds et celle en contact avec le côté extérieur du tube de force 61 536. Pour déplacer les nœuds restants, entre ces deux parois, une dernière fonction de transfert F_α^σ est calculée. Ses paramètres sont fixés à $\alpha = 2\ 000$ et $\sigma = 200$. Ils aboutissent à une répartition homogène des nœuds à l'intérieur du maillage volumique. Elle fait intervenir 135 829 points, correspondant à la somme du nombre de nœuds sur chacune des deux parois. La fonction obtenue est appliquée sur 200 790 points.

Sur le résultat de ce transfert, il est ensuite nécessaire de réaliser les interpolations des différents champs de valeurs requis pour le calcul du second code.

Dans le cas présent, il s'agit du flux thermique aux parois communes entre les maillages utilisés par Cast3m et Trio_U. Cette information est contenue dans un fichier MED. Pour pouvoir réaliser l'interpolation, il est nécessaire de disposer de la partie du maillage du code Trio_U correspondant au support du champ. Après l'avoir extraite, la plateforme SALOME est utilisée afin d'interpoler les différentes valeurs. L'interpolateur utilisé est conservatif et permet de traiter des maillages non-coplanaires.

6.2.3 Calcul thermo-hydraulique

L'eau entre en haut du dispositif et descend sous l'effet de l'aspiration par une pompe placée en bas. Le débit, constant, est tel que la vitesse du fluide soit de 2 $m.s^{-1}$ à l'entrée (aux points Ⓐ et Ⓑ sur la figure 6.5) sur la géométrie nominale. En raison des déformations, la largeur d'eau est modifiée (figure 6.5), la vitesse du fluide en entrée a changé et il est nécessaire de la mettre à jour (*c.f.* annexe A.3 qui détaille son calcul). La modification de cette vitesse ainsi que celle de la section de passage de l'eau impliquent un changement d'autres valeurs pour le calcul, comme celles concernant l'énergie cinétique de la turbulence et de sa dissipation (*c.f.* annexe A.5). Pour que Trio_U puisse effectuer son calcul, la température d'entrée du fluide doit être imposée. L'eau, qui entre dans l'interstice entre le fourreau et le tube de force, provient de la piscine du réacteur dont la température est de 38 $°C$. Durant les trois premiers mètres de sa descente, le fluide n'est sujet à aucun échange thermique car les deux parois sont à température ambiante. Elle est donc imposée à 38 $°C$ en entrée de la zone de calcul. L'ensemble des conditions limites du calcul Trio_U et le maillage utilisé sont schématisés sur la figure 6.7. Le nombre de Reynolds de l'écoulement est proche de 14 000 (*c.f.* annexe A.4). La turbulence est prise en compte par un modèle $k - \varepsilon$ (*c.f.* annexe A.5).

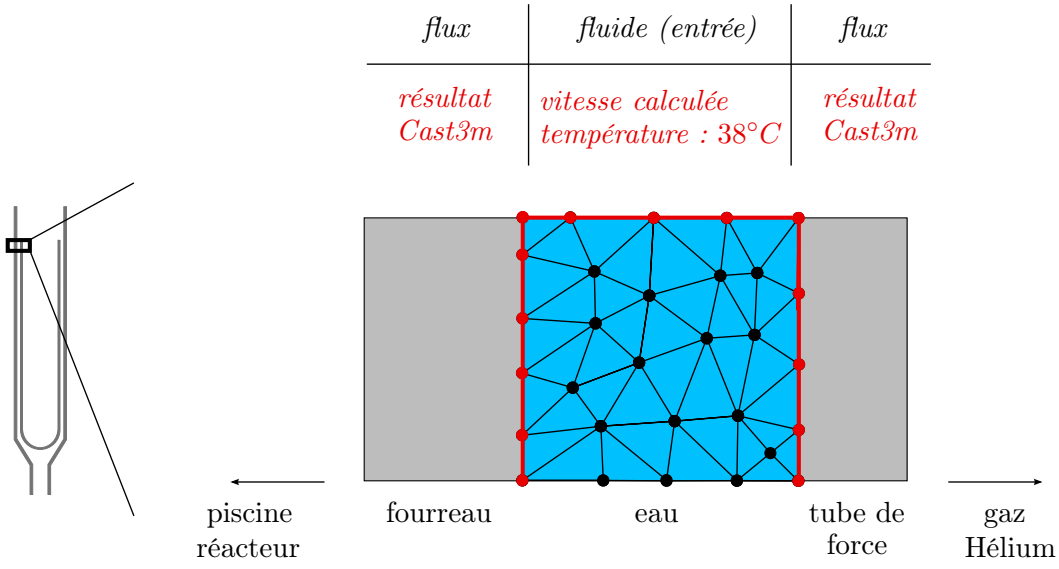


Figure 6.7 – Conditions limites du calcul de comportement thermo-hydraulique

A partir de ces données, le code Trio_U est à même de simuler le comportement du fluide. Il calcule sa température et permet ainsi de renseigner sur les valeurs aux parois des deux solides. Le code de calcul Trio_U nécessite un maillage volumique tétraédrique. Pour la même raison que pour Cast3m, il doit être fin horizontalement, c'est-à-dire disposer d'un minimum de 6 mailles en largeur. Pour réaliser une simulation numérique avec précision, Trio_U nécessite que les tétraèdres soient le plus réguliers possible. Combiné au besoin de finesse horizontale, cela induit un maillage très volumineux (dépassant les 60 millions de tétraèdres). Un calcul avec un tel maillage n'est pas envisageable. Un nouvel opérateur de diffusion a été mis en place par l'équipe de développement de Trio_U afin de permettre l'utilisation de tétraèdres étirés, sans dégradation excessive du résultat. L'écoulement du fluide dans la partie annulaire est relativement basique en comparaison du reste de la géométrie. Les mailles utilisées sont donc étirées d'un facteur 16 dans la partie supérieure et assez régulières dans la partie inférieure, la frontière étant positionnée au dessus de la calotte sphérique du tube de force à 170 mm (figure 6.8). De cette manière, le maillage utilisé contient 336 619 nœuds et 1 564 446 tétraèdres.

Le code Trio_U permet de calculer l'écoulement du fluide en fonction du temps physique simulé. Le calcul est effectué jusqu'à atteindre un état stationnaire, ce qui se traduit par une faible variation de la température du fluide. Le critère utilisé est celui présenté dans le paragraphe 6.2, où le champ utilisé est la température T_{Trio_U} en tout point du maillage du fluide. La convergence est atteinte à une itération k telle que $\|\Delta T_{Trio_U}\|_2^{[k]} \leq 10^{-6}$. Cette valeur est fournie par les experts du CEA. De même que pour Cast3m, cela signifie que les effets sur le calcul des déformations

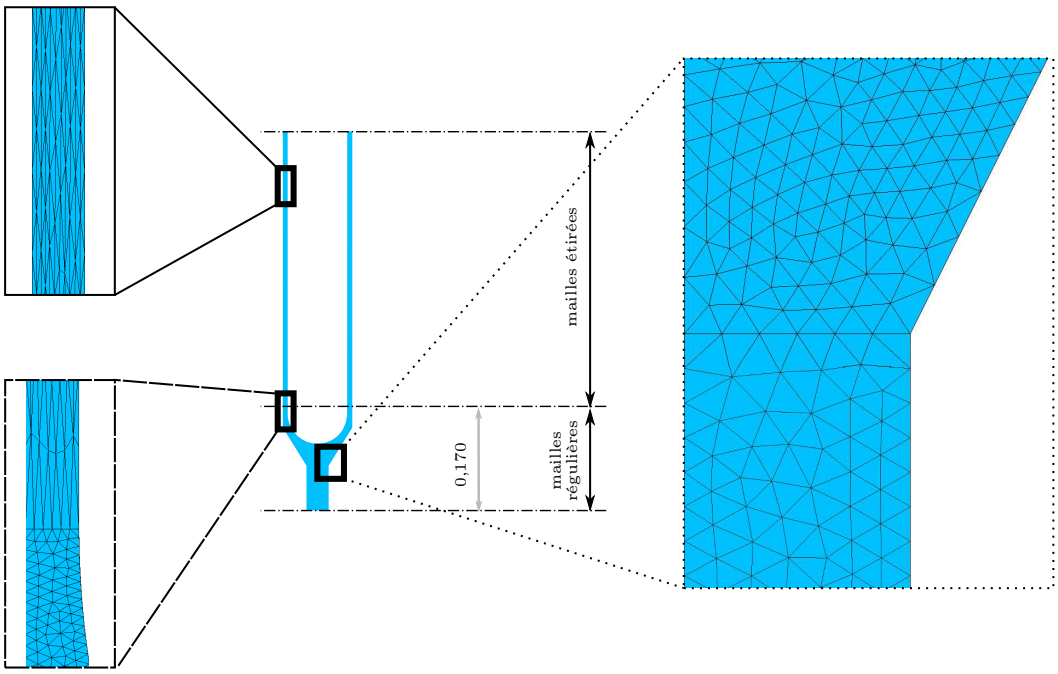


Figure 6.8 – Maillage utilisé par le code Trio_U (l'unité utilisée est le mètre)

des solides sont négligeables. Le résultat de la simulation numérique est un champ de températures aux faces du maillage fluide qu'il est nécessaire de fournir au code Cast3m.

L'état stationnaire est atteint entre 1 s (sans déformation) et 2 s (avec déformation) physiques simulées, ce qui correspond au temps de parcours moyen d'une particule fluide.

6.2.4 Seconde étape intermédiaire

Le code Trio_U permet de stocker le champ de températures directement dans le format de fichier d'échange MED. Aucun travail de conversion n'est donc requis. Avant de pouvoir relancer un nouveau calcul du code de comportement thermo-mécanique Cast3m, le champ de températures doit être interpolé sur son maillage spécifique. Cast3m réalise son calcul en s'appuyant sur le maillage initial, c'est-à-dire issu de la géométrie nominale. Le code Trio_U a effectué son travail sur le maillage modifié. L'interpolation du champ de températures sur le maillage initial Cast3m n'est donc pas directement possible. Il est tout d'abord interpolé sur le maillage modifié Cast3m, résultat de son précédent calcul. Ce maillage et l'initial étant appariés, le champ de températures est ensuite recopié nœud à nœud. De cette manière, le code Cast3m dispose du champ de températures calculé par Trio_U. Il est à même d'effectuer son nouveau calcul en prenant en compte le refroidissement des deux solides par l'eau les séparant.

Ce couplage est donc itératif et le critère d'arrêt est défini dans ce qui suit.

6.2.5 Arrêt du couplage

Le couplage mis en place a pour but d'étudier l'écoulement du fluide séparant les deux solides. Un état permanent est recherché car aucune des données d'entrée ne varie en fonction du temps. Nous avons choisi de faire porter le critère de convergence sur le champ de températures calculé par Cast3m (❺ sur la figure 3.1). En effet, un calcul Cast3m aboutit à un flux thermique aux parois et à une déformation. Tout comme le flux, cette modification géométrique dépend de la température des solides. Si le champ de températures entre deux calculs Cast3m est constant (*i.e.* la variation est inférieure à un seuil fixé par les experts du CEA), il est possible d'en conclure que les nouveaux résultats transmis au code de comportement thermo-hydraulique sont identiques aux précédents. Les deux codes utilisés étant déterministes, à une entrée donnée correspond un unique résultat. Cela induit qu'il n'y a plus de variations dans le résultat Trio_U entre deux calculs successifs. Le couplage est stoppé lorsque $\|\Delta T_{couplage}\|_2^{[k]} \leq 10^{-4}$, avec $T_{couplage}$ le champ de températures d'une itération de couplage après exécution du code Cast3m.

6.3 Apport du transfert des déformations géométriques

Pour illustrer l'apport du transfert de déformations, un second couplage est réalisé sans prendre en compte les modifications géométriques.

Lors de la première étape intermédiaire, le calcul de l'empreinte (❻ sur la figure 3.1) n'est plus réalisé. Le support du champ de flux thermiques étant le maillage modifié, une recopie des valeurs est opérée sur le maillage initial. Ceci est rendu possible en raison du fait que ces deux maillages sont appariés. Le champ obtenu est ensuite stocké dans un fichier MED. Le maillage utilisé par Trio_U est celui issu de la géométrie nominale. L'étape d'application des fonctions de transfert (❼ sur la figure 3.1) n'est donc plus requise. De même que précédemment, le futur support du champ sur le maillage Trio_U est extrait et les données y sont interpolées (❽ sur la figure 3.1 où le maillage utilisé est issu de la géométrie initiale). Le code effectue ensuite son calcul.

Lors de la seconde étape intermédiaire, le champ de températures obtenu par Trio_U s'appuie sur le maillage initial. Il est utilisé pour pouvoir modifier les conditions limites du calcul réalisé par le code Cast3m. Ce dernier effectue son travail sur le maillage initial. Le champ de températures peut donc être interpolé sur les parties communes entre son support et ce maillage (❸ sur la figure 3.1 où le maillage utilisé est issu de la géométrie initiale) et l'étape de recopie des valeurs du champ n'est plus nécessaire (❹ sur la figure 3.1).

6.4 Résultats obtenus

Les résultats obtenus après exécution des deux couplages présentés sont illustrés dans ce qui suit.

6.4.1 Maillage du fluide

Le maillage du fluide utilisé par le code de comportement thermo-hydraulique Trio_U est présenté dans ce paragraphe. Pour illustrer le résultat du transfert des déformations, une coupe est réalisée à hauteur du pincement de la lame d'eau et une vue de dessus est représentée. Le maillage en condition initiale (*i.e.* sans déformation) est illustré sur la figure 6.9. Une fois les fonctions de transfert appliquées sur ce dernier, le maillage est modifié. Représenté sur la figure 6.10, on remarque aisément le pincement de la partie basse, au niveau de la calotte sphérique du tube de force. Comme l'illustre la coupe présentée, la répartition des nœuds internes est conservée par les fonctions F_α^σ utilisées.

La déformation de la géométrie a un impact sur le temps de calcul du code thermo-hydraulique. Les différents temps sont détaillés dans le paragraphe suivant.

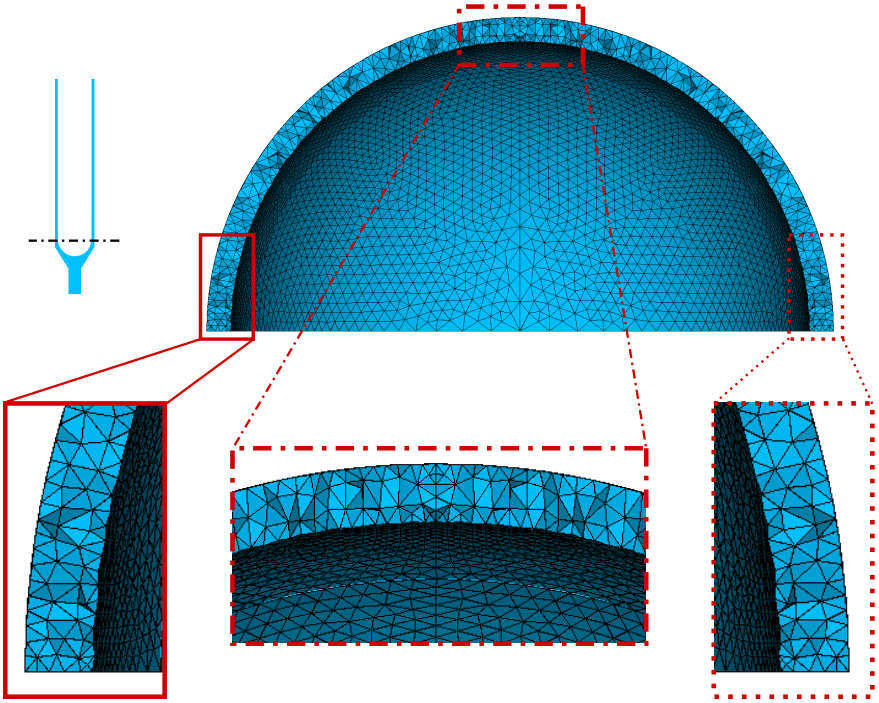


Figure 6.9 – Vue de dessus d'une coupe du maillage nominal utilisé par Trio_U

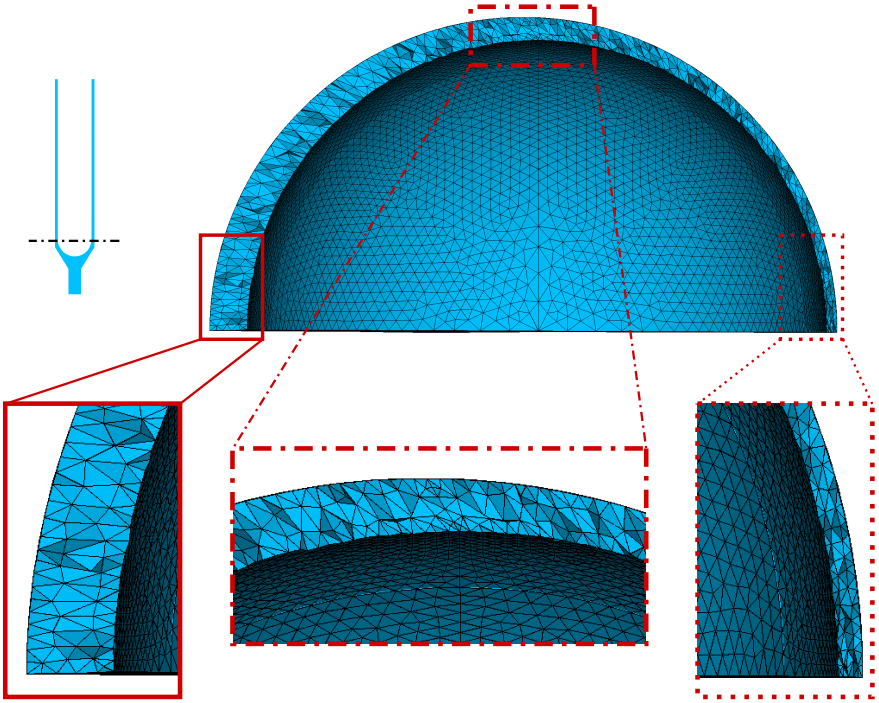


Figure 6.10 – Vue de dessus d'une coupe du maillage modifié utilisé par Trio_U

6.4.2 Temps de calcul

Les deux couplages sont effectués sur les mêmes ordinateurs afin de pouvoir comparer leurs temps de calcul. Pour un couplage donné, deux machines sont utilisées. La première est un PC 64 *bits* sous Linux Debian Etch, où seul un cœur cadencé à 2.6 GHz est utilisé, et disposant de 8 Go de mémoire vive. Elle exécute le coupleur et le code de comportement thermo-mécanique Cast3m. La seconde est utilisée pour le travail du code Trio_U. Il s'agit d'une machine 64 *bits* parallèle du CEA disposant de 64 Go de mémoire vive. Le calcul est réalisé sur 64 nœuds cadencés à 1.8 GHz.

Les deux couplages ont nécessité quatre itérations Cast3m-Trio_U pour converger. Les temps de calcul des différentes étapes sont présentés dans le tableau 6.1 pour le couplage sans la prise en compte des déformations, dans le tableau 6.2 avec. Ils montrent tous deux qu'en dehors du code Trio_U, les temps de calculs sont constants et ne dépendent pas de l'itération du couplage entre Cast3m et Trio_U.

Les différences entre les tableaux 6.1 et 6.2 portent sur la première étape intermédiaire et sur le calcul du code Trio_U.

Etapes	Itération			
	#1	#2	#3	#4
Cast3m	8min	8min	8min	8min
intermédiaire 1	4min	4min	4min	4min
Trio_U	22h 30min	21h 10min	16h 00min	17h 00min
intermédiaire 2	1min	1min	1min	1min
Total	22h 38min	21h 23min	16h 13min	17h 13min
Temps de calcul complet du couplage : 3j 05h 27min				

Table 6.1 – Détail du temps de calcul des différentes itérations du couplage ne prenant pas en compte les déformations

Etapes	Itération			
	#1	#2	#3	#4
Cast3m	8min	8min	8min	8min
intermédiaire 1	7min	6min	7min	6min
Trio_U	32h 10min	40h 00min	31h 50min	35h 55min
intermédiaire 2	1min	1min	1min	1min
Total	32h 26min	40h 15min	32h 07min	36h 10min
Temps de calcul complet du couplage : 5j 20h 58min				

Table 6.2 – Détail du temps de calcul des différentes itérations du couplage prenant en compte les déformations

Dans le premier cas, il s'agit de la prise en compte des déformations géométriques calculées par Cast3m : calcul des deux fonctions de transfert correspondant à la déformation des solides, application sur les parois communes aux maillages utilisés par Cast3m et Trio_U, calcul de la fonction de transfert de la déformation des bords du maillage fluide et application sur les nœuds internes. Les temps de calculs sont détaillés dans le tableau 6.3.

Dans le second cas, on remarque que le temps de calcul du code Trio_U augmente d'un facteur deux lorsque les déformations sont prises en compte (presque $6j$ contre un peu plus de $3j$).

	Calcul de F_α^σ			Application de F_α^σ		
	paramètres	nb nœuds	temps	zone	nb nœuds	temps
tube de force	$\alpha =$ 2 000 $\sigma =$ 200	2 289	5s	paroi commune	61 536	16s
fourreau	$\alpha =$ 2 000 $\sigma =$ 200	7 917	5s	paroi commune	74 293	18s
fluide	$\alpha =$ 2 000 $\sigma =$ 200	135 829	6s	nœuds internes	200 790	54s

Table 6.3 – Détail du temps de calcul du transfert des déformations géométriques

6.4.3 Température du fluide

Les deux solides sont refroidis par l'eau en mouvement les séparant. Pour montrer l'apport de notre approche, nous montrons les températures finales (*i.e.* après convergence du couplage) que nous obtenons avec (figure 6.11) et sans (figure 6.12) la prise en compte des déformations géométriques. On remarque alors deux choses importantes.

La première concerne la température maximale obtenue : 43.68 $°C$ sans la prise en compte des déformations et 42.22 $°C$ avec. On obtient donc un écart de 1.46 $°C$ entre les deux calculs sur une variation totale de 5.68 $°C$ des températures extrémales, ce qui correspond à un échauffement maximal 25 % plus faible. La prise en compte des déformations géométriques permet donc de modifier la solution, la différence de la valeur maximale étant notable. Cette différence s'explique par le fait que la portion qui s'échauffe le plus par conduction thermique avec les parois des solides voit sa largeur augmenter suite aux déformations géométriques. Plus de fluide circule, permettant ainsi de mieux refroidir le tube de force et le fourreau.

Le second point important porte sur le comportement du fluide. Dans la figure 6.11, on constate que le fluide se mélange très peu entre la zone proche du cœur et celle éloignée. Sous la calotte sphérique du tube de force, on obtient une ligne médiane séparant le fluide à température maximale (proche du cœur) et le fluide à température minimale (éloigné du cœur). Dans la figure 6.12, le fluide est bien plus brassé en position basse (sous la calotte sphérique du tube de force). Ceci

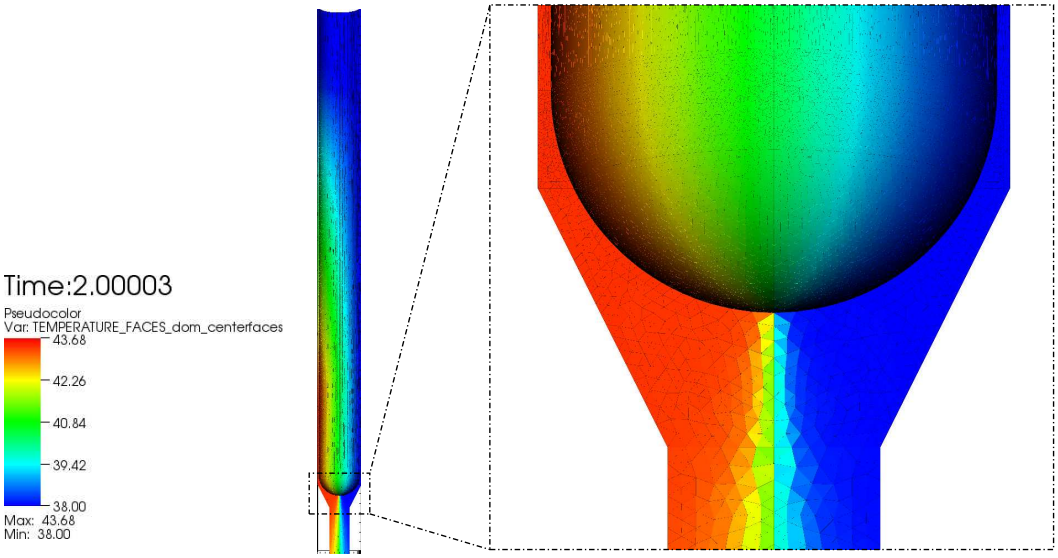


Figure 6.11 – Température du fluide obtenue sans la prise en compte des déformations géométriques

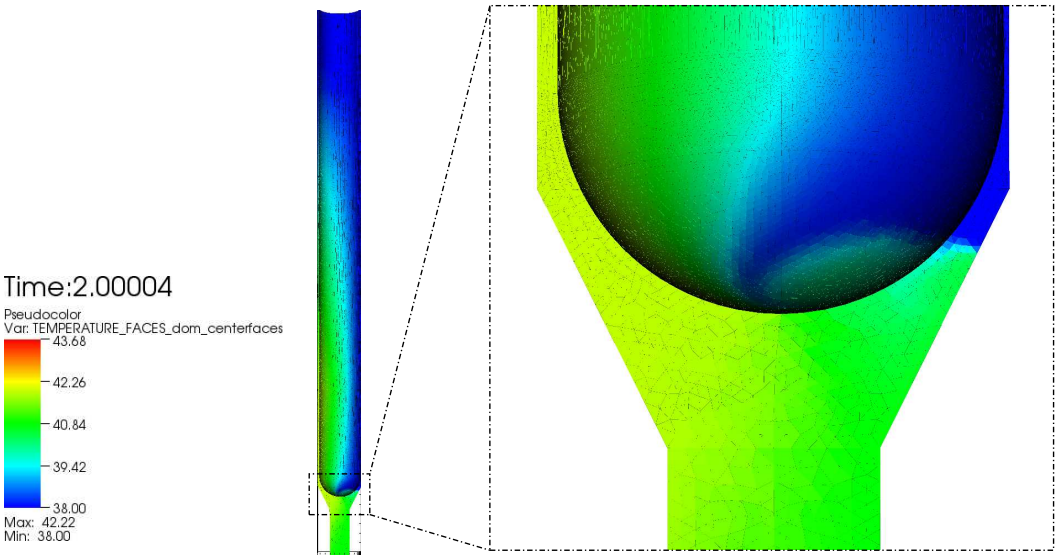


Figure 6.12 – Température du fluide obtenue avec la prise en compte des déformations géométriques

est dû au pincement de la lame d'eau, obligeant le fluide à se répartir ailleurs. Pour confirmer cet effet, l'écoulement du fluide est illustré dans le paragraphe suivant.

6.4.4 Simulation de l'écoulement du fluide

Pour mettre en avant l'écoulement du fluide dans les deux couplages réalisés, une particule est suivie tout au long de sa descente dans le maillage utilisé par le

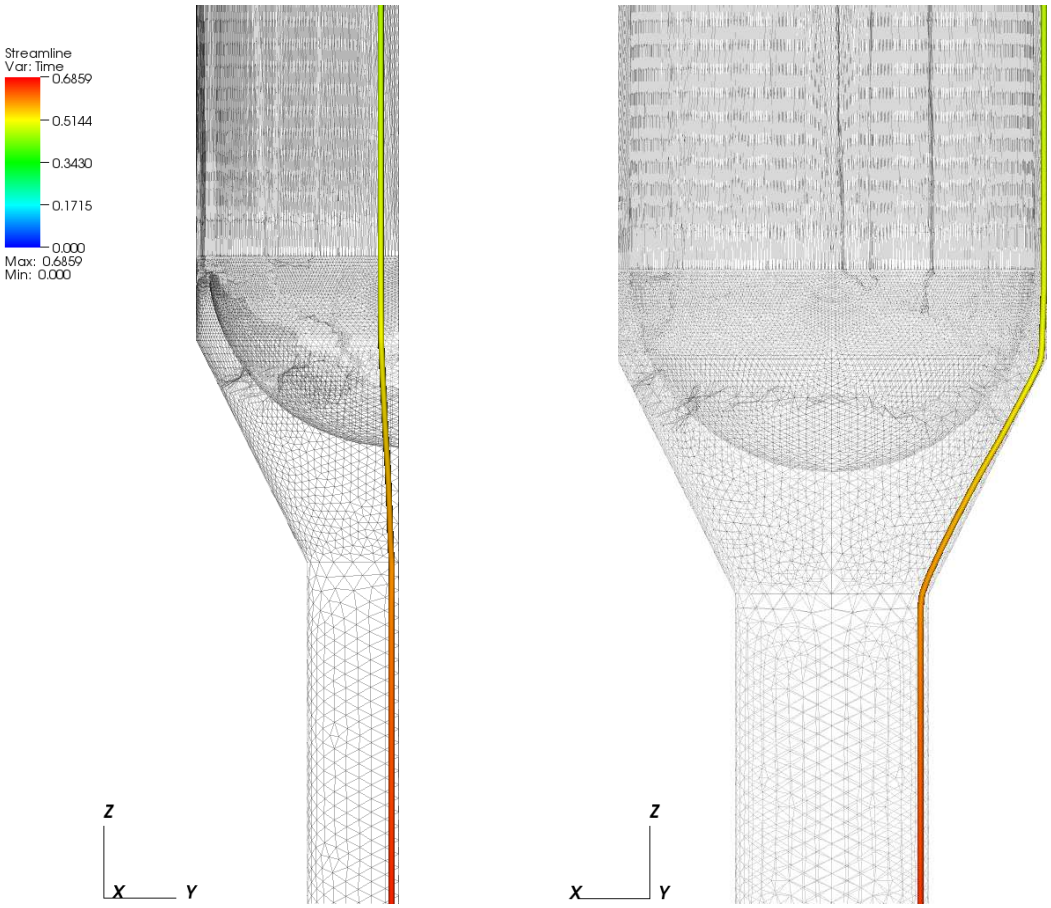
code Trio_U. Le brassage du fluide étant dû au pincement de la lame d'eau en partie basse du dispositif (au niveau de la calotte sphérique du tube de force), la particule est positionnée initialement du côté éloigné du cœur, en haut. Son déplacement est illustré pour les deux couplages : sans la prise en compte des déformations (figure 6.13) et avec (figure 6.14). La couleur utilisée schématise le temps physique nécessaire pour que la particule atteigne une position donnée. Dans le premier cas, l'écoulement est direct et la particule reste du côté éloigné du cœur. Elle atteint le bas du dispositif en presque 0.7 s de temps physique simulé. Dans le second cas, où les déformations géométriques sont prises en compte, l'écoulement est plus complexe. La trajectoire de la particule s'éloigne petit à petit de la zone pincée et rencontre une région perturbée sous la calotte sphérique du tube de force. C'est la raison pour laquelle le fluide est brassé et les températures plus homogènes dans la partie basse. Le bas du dispositif est atteint en presque 1.3 s de temps physique simulé, soit presque le double de celui du calcul sans la prise en compte des déformations.

6.5 Synthèse

Les deux couplages mis en place dans ce chapitre ont permis de mettre en application les différents points traités dans ce travail de thèse. Les codes de calcul utilisés sont développés par le CEA. Etant validés dans leur domaine d'application, il est possible de supposer que le résultat obtenu par un couplage de ces derniers aboutit à une simulation plus fine de la réalité. Les deux couplages réalisés ont abouti à des solutions nettement différentes, ce qui tend à signifier que le transfert des déformations doit être pris en compte. Les experts en thermo-hydraulique du CEA confirment ce point car l'écoulement final du fluide correspond à leurs attentes. Le résultat obtenu permet également de pouvoir prédire qu'une vibration du tube de force en partie basse (calotte sphérique) est possible du fait de la perturbation de l'écoulement. Toutefois, il n'existe pas, à ce jour, d'expérimentation permettant de corroborer ces hypothèses.

vue de côté

vue de face



vue de dessus

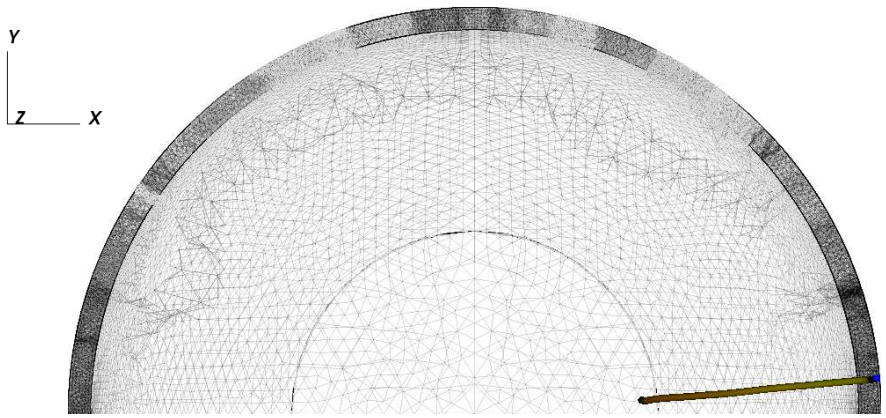
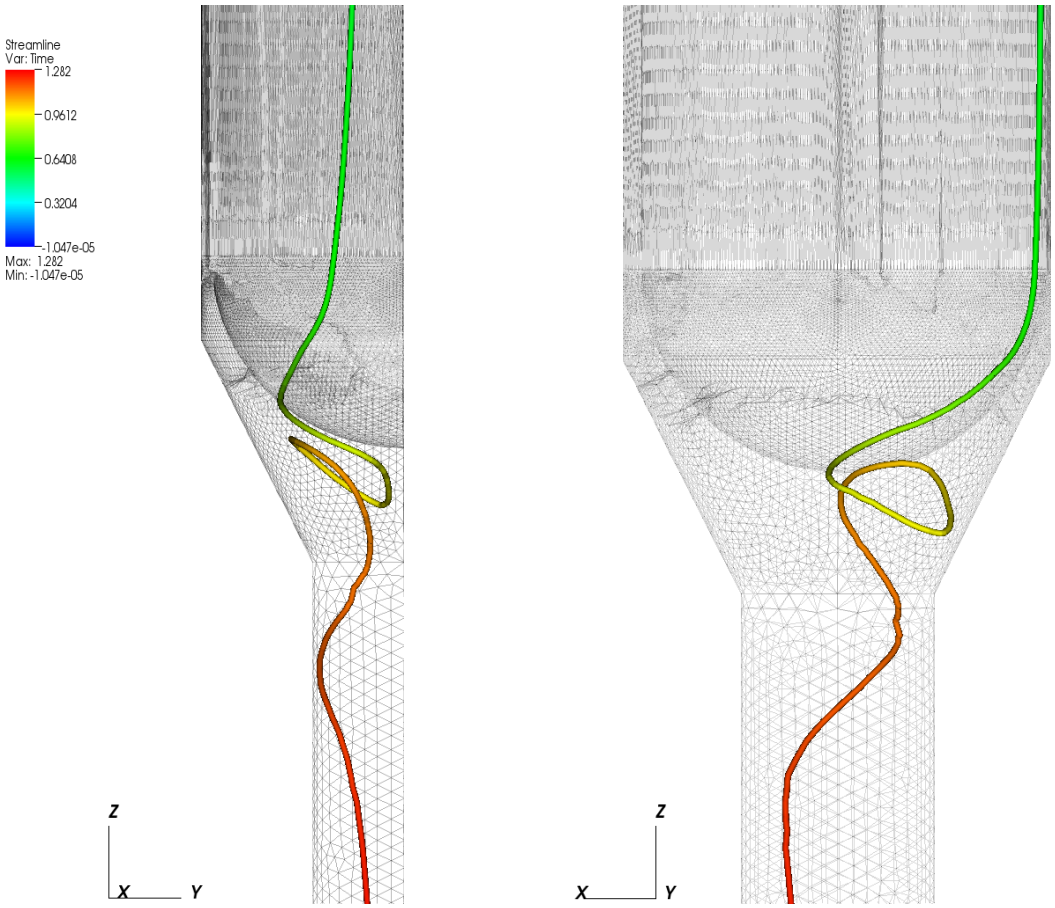


Figure 6.13 – Ecoulement d'une particule fluide dans la géométrie initiale

vue de côté

vue de face



vue de dessus

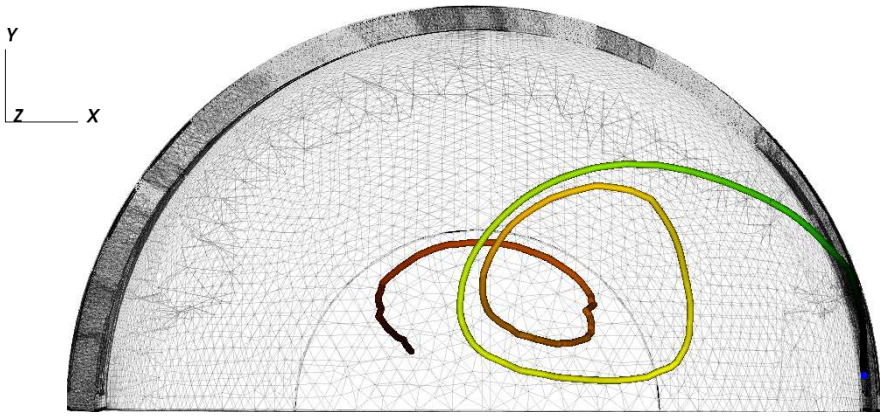


Figure 6.14 – Ecoulement d'une particule fluide dans la géométrie déformée

Chapitre 7

Synthèse et perspectives

Les simulations numériques sont réalisées par des programmes informatiques spécialisés dans un domaine physique et sur un type de problèmes donné. Lors d'un couplage entre plusieurs codes de calcul, des données sont échangées entre eux afin de leur permettre de prendre en compte les résultats obtenus par les autres codes. Cet aspect est vrai pour les champs de valeurs, comme la température, mais également pour les modifications géométriques. Le travail présenté dans ce document concerne le transfert de déformations entre codes de calcul.

Comme cela a été présenté dans le chapitre 1, une telle approche peut être traitée par un travail en deux temps. Tout d'abord, il est important que les différents codes couplés s'appuient sur des données physiques communes et sur une même géométrie globale. Pour cela, un modèle commun, même sommaire, est nécessaire. Il permet de fédérer les codes et de rendre disponibles les conditions initiales des calculs et les résultats des simulations numériques. Ensuite, des fonctions particulières doivent être utilisées pour réaliser le transfert des modifications géométriques obtenues après une phase préalable de définition des déformations. Les deux cas que nous avons proposé de traiter concernent le transfert entre des maillages issus d'une même géométrie ou de géométries adjacentes.

Le chapitre 2 a présenté un état de l'art des méthodes existantes pour répondre aux différents besoins que sont la mise en place d'un couplage, d'un modèle commun et de fonctions de transfert dans le cas particulier de la prise en compte des déformations. Malheureusement, les approches existantes ne permettent pas de résoudre les problèmes liés aux modifications géométriques. Des fonctions de transfert de déformations ont cependant été retenues mais souffrent d'un temps de calcul élevé. Pour améliorer ce point, des méthodes de simplification de maillages ont été exposées afin de réduire leur complexité.

Une nouvelle technique de couplage et un modèle commun ont été proposés dans le chapitre 3. Les différentes étapes de la méthode de couplage mise en place ont été détaillées et permettent de gérer les différents échanges de données. L'architecture du modèle commun utilisé dans cette méthode a ensuite été définie. Le

travail d'implémentation n'a pu être effectué dans le cadre de ce doctorat. Un modèle commun sommaire a tout de même été mis en place dans le but de tester et de valider l'approche proposée. Ce travail devra être achevé pour étendre les cas d'application possibles et nécessitera probablement de compléter l'architecture mise en place pour répondre aux nouvelles spécificités des cas à traiter. En l'état actuel des développements, la méthode de couplage proposée est fonctionnelle. Toutefois, une amélioration sera d'y intégrer l'interface de couplage ICoCo. Pour cela, elle doit évoluer de manière à permettre le transfert de déformations géométriques et doit être supportée par l'ensemble des codes de calcul pouvant être couplés.

Une perspective à plus long terme est la mise en œuvre d'un modèle commun pour la physique des réacteurs regroupant diverses disciplines comme la neutronique, la thermo-hydraulique, la mécanique, le comportement du combustible, ... Pour réussir un tel travail, il sera nécessaire de définir les cas d'études de manière générale et de spécifier les différents problèmes qui s'y portent, comme les problèmes de thermo-mécanique et de thermo-hydraulique pour réaliser l'étude du refroidissement d'un dispositif expérimental. *In fine*, les problèmes correspondent aux codes de calculs qui pourront être utilisés. Une telle approche permettrait de remplacer un code de calcul par un autre de manière simple. Il faut cependant indiquer qu'il sera nécessaire de supprimer les jeux de données, qui sont spécifiques aux codes de calcul, et de les remplacer par des échanges de données et d'informations avec le coupleur via l'interface ICoCo.

Concernant la problématique du transfert des déformations, l'état de l'art a permis de retenir quelques méthodes répondant partiellement à nos attentes. Ces dernières ont fait l'objet de tests dans le chapitre 4 afin de les étudier pour avoir une première idée du résultat que l'on pourrait obtenir en les utilisant. Le chapitre 5 a tout d'abord présenté notre méthode de localisation des déformations dont le but est de limiter la taille de la zone à traiter. Ensuite, les deux fonctions effectivement mises en place ont été définies : F_α et F_α^σ. Elles combinent toutes deux une méthode de simplification visant à abaisser la complexité des maillages utilisés pour ne garder que α nœuds, et un réseau de fonctions de base radiale permettant d'interpoler le déplacement constaté de ces α points. La première fonction, F_α, est à support global car elle s'appuie sur la fonction TPS (Thin-Plate Spline). Elle minimise une énergie de flexion permettant de réduire les oscillations de la surface ou du volume obtenu par application sur un second maillage. La seconde fonction qui a été définie, F_α^σ, est à support local. Elle utilise la fonction CS-RBF (Compactly-Supported Radial Basis Function). Le choix que nous avons réalisé concernant la taille du support de la fonction est de la définir de manière à s'assurer que toutes les CS-RBF du réseau fassent intervenir au moins σ points caractéristiques. La taille du support est constante à toutes les CS-RBF du réseau. Une perspective intéressante serait de tester d'autres possibilités, comme un support adaptatif et local à chaque CS-RBF. Une autre proposition d'évolution repose sur le fait que la forme de la CS-RBF, de type gaussien, laisse présager qu'il serait possible de remplacer les RBF par des fonctions de même forme, comme les fonctions de densité utilisées

pour la représentation d'objets implicites ([Ges00], [Mar02]). Les possibilités sont très nombreuses et une étude devra être effectuée pour tester et valider une telle approche. Enfin, le choix de la méthode de simplification utilisée s'est porté sur la contraction d'arêtes utilisant le critère de la *QEM* (*Quadric Error Metric*). Il serait intéressant de confronter les résultats obtenus avec d'autres techniques de simplification.

Afin d'accélérer les temps de calcul, le travail réalisé devra être parallélisé. La technique de localisation des déformations mise en place implique différentes zones à traiter, chacune nécessitant une fonction de transfert. Ces fonctions étant indépendantes, elles peuvent être calculées de manière parallèle. Enfin, lors de l'application des fonctions sur un maillage, l'étape du calcul de la nouvelle position des points peut également être rendue parallèle.

Le travail qui a été détaillé dans les chapitres 3 et 5 a été mis en application dans le chapitre 6. Le cas d'étude a porté sur les dispositifs expérimentaux du réacteur de recherche Jules Horowitz. Un couplage a été mis en place afin de valider l'approche proposée concernant le transfert des déformations géométriques lors du couplage de codes de calcul. Il a également permis d'illustrer son impact sur la simulation physique. Le résultat obtenu est convainquant car il montre des différences notables, tant du point de vue de la thermique (*i.e.* température maximale calculée) que de l'écoulement du fluide (*i.e.* trajectoire d'une particule). La précision de la simulation des phénomènes physiques est améliorée au prix, dans le cas présent, d'un temps de calcul plus élevé. Il nous semble toutefois important de réaliser de plus amples études et de les confronter à des expérimentations pour connaître l'amélioration exacte apportée par la prise en compte du transfert des déformations lors de couplages de codes de calcul.

Le cas traité dans ce travail de doctorat s'est limité à la problématique du transfert de déformations utilisant des maillages. Son extension à d'autres modèles géométriques est un axe de recherche intéressant qu'il sera nécessaire de prendre en compte. Il sera ainsi possible de réaliser des couplages faisant intervenir des codes de MonteCarlo, ces derniers représentant la géométrie en utilisant les CSG.

Pour améliorer la qualité des résultats, il est intéressant de pouvoir quantifier les erreurs réalisées. Pour obtenir ces informations, il est nécessaire de prendre en compte les incertitudes qui interviennent aussi bien au niveau de la géométrie, des données et des modèles physiques utilisés que des simulations numériques en raison de la précision des ordinateurs. La géométrie nominale est définie sans tenir compte des tolérances mécaniques des pièces. Les maillages utilisés sont une discrétisation de la géométrie, ce qui induit un biais (*e.g.* discrétisation d'une sphère). Les valeurs physiques sont toujours associées à un niveau d'incertitude. Enfin, les ordinateurs manipulent des valeurs discrètes. Ainsi, les sources d'incertitude sont nombreuses et cette problématique doit faire l'objet d'une étude complète.

Afin de rendre pérenne le travail réalisé, il devra être intégré dans la plateforme SALOME. Pour cela, le développement du modèle commun doit tout d'abord être finalisé et supporté par SALOME. Son format d'échange doit être spécifié et les modifications par rapport au format GDML sur lequel il s'appuie doivent être définies. Par ailleurs, le composant de couplage YACS de SALOME doit évoluer pour inclure les deux fonctions de transfert F_α et F_α^σ. De cette manière, la méthode de couplage présentée dans le chapitre 3 pourra être réalisée au travers des différents outils proposés par SALOME. Et pour être générique et rester dans l'optique d'un couplage simple à mettre en œuvre, où les codes de calcul peuvent être remplacés sans induire de lourds développements, l'interface de couplage ICoCo doit aussi être prise en compte. Une évolution préalable de cette interface sera nécessaire pour permettre le transfert de déformations géométriques.

Les développements réalisés dans ce travail de doctorat ont montré la faisabilité de l'approche proposée et son efficacité pour effectuer un transfert de déformations géométriques lors d'un couplage de deux codes de calcul. Notre travail pourrait aisément se généraliser à d'autres calculs comme l'analyse des interactions fluide-structure ou lors de couplages faisant intervenir plus de deux codes. Il pourrait également être applicable à d'autres domaines que les simulations numériques, où plusieurs tâches sont réalisées autour d'une géométrie globale commune qui est soumise à des déformations.

Annexe A

Calculs complémentaires liés au cas pratique

Sommaire

A.1 Epaisseur moyenne de la lame d'eau 141
A.2 Diamètre hydraulique 142
A.3 Vitesse d'entrée du fluide 142
A.4 Nombre de Reynolds 143
A.5 Valeurs d'énergie cinétique turbulente et de sa dissipation 143
A.6 Puissance thermique 144
A.6.1 Puissance déposée dans les tubes 144
A.6.2 Puissance thermique évacuée par l'eau 144
A.6.3 Bilan possible et prévision des résultats 145

Cette annexe porte sur le cas d'application présenté dans le chapitre 6. Elle rassemble les calculs effectués pendant la première étape intermédiaire du couplage avec transfert des déformations géométriques, entre l'exécution du code de comportement thermo-mécanique Cast3m et celle du code de comportement thermo-hydraulique Trio_U. Le transfert des déformations calculées par Cast3m implique une variation de la largeur de la lame d'eau. Pour en tenir compte, Trio_U nécessite que certaines données soient mises à jour. Les différents calculs sont présentés dans ce qui suit.

A.1 Epaisseur moyenne de la lame d'eau

La section de passage de la lame d'eau dans la partie haute du cas d'application (Ⓐ et Ⓑ sur la figure 6.5 page 120) peut être obtenue nominalement de façon simple. Il s'agit de la différence des aires des deux disques constituant la section des parties hautes cylindriques (*c.f.* coupe B sur la figure 6.3 page 118). La figure A.1 représente cette lame d'eau au niveau de l'entrée de l'étude par l'intermédiaire d'une coupe axiale.

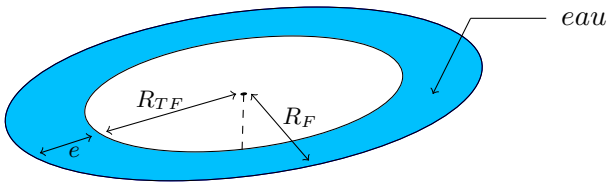


Figure A.1 – Schéma représentant la section de passage de la lame d'eau à l'entrée du cas test.

Initialement, le rayon interne du fourreau (R_F) est strictement égal au rayon externe du tube de force (R_{TF}) auquel est ajouté la largeur nominale de la lame d'eau, notée e. Ainsi, $R_F = R_{TF} + e$. L'aire de la section de passage, notée S, vérifie donc l'équation :

$$S = \pi(R_F^2 - R_{TF}^2) = \pi(e^2 + 2R_{TF}e) \tag{31}$$

En raison de l'échauffement γ qui survient dans cette étude, les deux solides se déforment. Le tube de force étant maintenu en haut, son rayon externe reste inchangé à cet endroit, c'est-à-dire égal à R_{TF}. Le fourreau est déformé globalement, entraînant un changement de son rayon interne en entrée du cas d'application.

Le code de calcul Cast3m spécialisé dans le comportement thermo-mécanique nous renseigne sur l'aire de la nouvelle section de passage de l'eau après déformations (S^*). Connaissant cette information, il est possible d'en déduire l'épaisseur moyenne de la lame d'eau (e^*) en résolvant l'équation du second degré :

$$e^{*2} + 2R_{TF}e^* - \frac{S^*}{\pi} = 0 \tag{32}$$

On en déduit que :

$$\Delta = 4\left(R_{TF}^2 + \frac{S^*}{\pi}\right) \tag{33}$$

et :

$$e^* = -R_{TF} \pm \frac{\sqrt{\Delta}}{2} \tag{34}$$

La valeur recherchée e^* est celle qui est positive.

A.2 Diamètre hydraulique

Le diamètre hydraulique est directement lié à l'épaisseur moyenne de la lame d'eau. Noté D_h, il s'agit d'une grandeur caractéristique utilisée pour les calculs d'écoulement dans un tube ou une conduite hydraulique. Elle se définie par :

$$D_h = \frac{4S}{P} \tag{35}$$

avec :

D_h	(m)	diamètre hydraulique
S	(m^2)	aire de la section de la lame d'eau
P	(m)	périmètre mouillé de la section

En reprenant les notations utilisées précédemment dans le paragraphe A.1, ce diamètre hydraulique s'exprime par :

$$D_h = \frac{4S}{P} = \frac{4 \times \pi(R_F^2 - R_{TF}^2)}{2\pi(R_F + R_{TF})} = 2(R_F - R_{TF}) = 2e \tag{36}$$

A.3 Vitesse d'entrée du fluide

Le débit massique Q du fluide est constant et imposé par une pompe placée en bas du fourreau. Il peut être calculé grâce à l'équation suivante :

$$Q = U\rho S \tag{37}$$

avec :

Q	$(kg.s^{-1})$	débit massique du fluide
U	$(m.s^{-1})$	vitesse moyenne du fluide
ρ	$(kg.m^{-3})$	masse volumique du fluide
S	(m^2)	aire de la section de la lame d'eau

En condition initiale, le débit massique est tel que la vitesse U est égale à $2\ m.s^{-1}$ à l'entrée du dispositif (Ⓐ et Ⓑ sur la figure 6.5 page 120). La déformation du

fourreau et du tube de force induit une modification de la section de passage de l'eau. Afin de connaître la vitesse U^* d'entrée du fluide, il est nécessaire de calculer :

$$U^* \rho S^* = U \rho S \Rightarrow U^* = \frac{US}{S*} \tag{38}$$

avec :

U^*	$(m.s^{-1})$	vitesse moyenne du fluide pour une itération (après déformation)
S^*	(m^2)	aire moyenne de la section de la lame d'eau après déformation
U	$(m.s^{-1})$	vitesse moyenne du fluide pour la géométrie nominale
S	(m^2)	aire de la section de la lame d'eau pour la géométrie nominale

A.4 Nombre de Reynolds

Le nombre de Reynolds est une valeur adimensionnelle utilisée pour caractériser la nature du régime d'un écoulement. Noté Re, il correspond au rapport entre les forces d'inertie et les forces visqueuses :

$$Re = \frac{\rho U L}{\mu} \tag{39}$$

avec :

Re		nombre de Reynolds de l'écoulement
ρ	$(kg.m^{-3})$	masse volumique du fluide
U	$(m.s^{-1})$	vitesse moyenne du fluide
L	(m)	distance caractéristique
μ	$(Pa.s)$	viscosité du fluide

Dans notre cas, la distance caractéristique utilisée sera le diamètre hydraulique D_h détaillé dans le paragraphe A.2.

A.5 Valeurs d'énergie cinétique turbulente et de sa dissipation

Les formules qui suivent s'appuient sur le manuel utilisateur du code de calcul Trio_U, le support du code ESI CFD (`http://support.esi-cfd.com/esi-users/turb_parameters/`) et le site CFD online (`http://www.cfd-online.com`). Les valeurs de l'énergie cinétique turbulente k et de sa dissipation ε utilisées dans la résolution du modèle de turbulence ont un impact sur la qualité du résultat obtenu ainsi que sur la rapidité de la résolution complète du calcul.

$$k = \frac{3}{2}(tU)^2 \tag{40}$$

avec :

k	$(m.s^{-1})$	énergie cinétique turbulente
t		taux de turbulence de l'écoulement
U	$(m.s^{-1})$	vitesse moyenne du fluide

Le taux de turbulence peut être obtenu de manière approchée à partir du nombre de Reynolds Re de l'écoulement par $t = 0.16 \times Re^{-1/8}$.

$$\varepsilon = \frac{k^{3/2}}{l} \tag{41}$$

avec :

ε	$(m^2.s^{-1})$	taux de dissipation de k
k	$(m.s^{-1})$	coefficient k précédemment calculé
l	(m)	échelle de longueur de turbulence

L'échelle de longueur de turbulence, notée l, est une valeur physique décrivant la taille des grands tourbillons dans un flux turbulent. Cette valeur est estimée à $l = 0.07 \times D_h$, où D_h est le diamètre hydraulique (*c.f.* paragraphe A.2).

A.6 Puissance thermique

Le rayonnement γ issu du cœur du réacteur induit un échauffement des matériaux de notre étude. Le fourreau et le tube de force voient donc leur température augmenter. La chaleur générée est évacuée par convection thermique par la lame d'eau circulant entre les deux solides. Afin de s'assurer de la cohérence des calculs, les bilans de puissance sont réalisés.

A.6.1 Puissance déposée dans les tubes

La puissance déposée dans les matériaux en raison de l'échauffement γ peut être calculée comme étant :

$$P_{solides} = \iiint_V \rho E_\gamma dv \tag{42}$$

avec :

$P_{solides}$	(W)	puissance thermique accumulée dans les solides
V	(m^3)	volume soumis à l'échauffement γ
ρ	$(kg.m^{-3})$	masse volumique des matériaux
E_γ	$(W.g^{-1})$	échauffement γ

Il faut alors distinguer le fourreau du tube de force. Deux puissances sont donc présentes dans l'étude pratique mise en place. La puissance totale accumulée est finalement la somme de ces deux dernières.

A.6.2 Puissance thermique évacuée par l'eau

La puissance évacuée par l'eau peut être obtenue par l'intermédiaire de la formule suivante :

$$P_{fluide} = QC_p\Delta T \tag{43}$$

avec :

P_{fluide}	(W)	puissance thermique évacuée par l'eau
Q	$(kg.s^{-1})$	débit massique du fluide
C_p	$(J.kg^{-1}.K^{-1})$	chaleur massique du fluide
ΔT	(K)	différence de la température en entrée et la température moyenne en sortie

Le débit massique s'exprime de la façon suivante :

$$Q = \iint_S \rho U ds \tag{44}$$

avec :

Q	$(kg.s^{-1})$	débit massique du fluide
ρ	$(kg.m^{-3})$	masse volumique du fluide
U	$(m.s^{-1})$	vitesse moyenne du fluide
S	(m^2)	aire de la section de la lame d'eau

A.6.3 Bilan possible et prévision des résultats

Dans le script des données du code Cast3m, on calcule la puissance $P_{solides}$ déposée dans les solides à partir d'une fonction représentant la source γ issue du cœur. Le code Trio_U calcule et fournit P_{fluide}. De l'égalité des puissances, on peut prédire la variation moyenne de la température du fluide entre l'entrée et la sortie de l'étude.

Liste des auteurs

— **A** —

Agarwal, P. . . . . ix, 35
Akkouche, S. . . . . ix, 18
Allègre, R. . . . . ix, 18
Allen, C. B. 27
Alliez, P. 24, 28
Angelidis, A. 25
Aragonés, J.-M. 16
Ayache, N. 29

— **B** —

Bacher, P. 115
Bade, A. 29
Badler, N. 30
Ballagny, A. 115
Baran, I. 30
Barba, L. 28
Barr, A. H. 25, 30
Baudron, A.-M. 16
Beatson, R. K. 27
Bechmann, D. . . . . ix, 25
Belliard, M. 14
Belyaev, A. 94
Ben-Chen, M. 30
Bergeaud, V. 16
Bernardi, C. 120
Bestion, D. 16
Beswick, A. R. 26
Bijl, H. 26, 27
Bischo, S. 24, 28
Blocken, B. 13
Bonaccorsi, T. . . . . ix, 4, 14, 15
Bonneau, G. 29, 31
Bookstein, F. L. 27, 94
Borouchaki, H. 24
Borrel, P. 34
Botsch, M. . . . . ix, 24, 26–30, 32
Bouazza, M. 22
Boubée de Gramont, T. 14
Boxerman, E. 28, 30
Boztosun, I. 26, 27, 71
Bozzini, M. 27
Brooks, F. . . . . ix, 35
Busayarat, C. . . . . ix, 19

— **C** —

Cacuci, D. G. 16
Calvin, C. 120
Cambi, G. 14
Carlson, M. 28, 30
Carr, J. C. 27
Carter, J. O. 26
Chaine, R. . . . . ix, 18
Charafi, A. 26, 27, 71
Charras, T. 120
Chauliac, C. 16
Cherouat, A. 24
Cherrie, J. B. 27
Cheutet, V. 17
Chuon, C. 36
Chytracek, R. 23
Ciampalini, A. 35
Cignoni, P. 32, 35
Cohen, J. . . . . ix, 35
Combescure, A. 120
Crouzet, N. 16
Cueto, O. 120

— **D** —

Daniel, M. 17, 19, 20, 98, 99
Daubisse, J. 86, 91
De Luca, L. . . . . ix, 19
deBoer, A. 26, 27
DeHaemer, M. . . . . ix, 33
Delingette, H. 29, 30
DeRose, T. 36
Desbrun, M. 30
Djidjeli, K. 26, 27, 71

Döderlein, C. 16
Duby, C. 80
Duchamp, T. 36
Duplex, B. 98, 99

— **E** —
Emonot, P. 120
Evans, T. R. 27

— **F** —
Falcidieno, B. 19
Fang, S. 27
Fillion, P. 14
Fleischer, K. 30
Florenzano, M. . . . . ix, 19
Forax, G. 21
Fries, T. 31
Fright, W. R. 27

— **G** —
Gain, J. . . . . ix, 25
Galin, E. . . . . ix, 18
Gardan, Y. 17, 18, 20
Garland, M. 32–36, 38, 39
Geay, A. 16
Gera, B. 13
Gesquière, G. 25, 98, 99, 137
Giannimi, F. 19
Gibson, S. F. F. 24
Giraud-Moreau, L. 24
Goldbronn, P. 14
Gotsman, C. 30
Gousseau, P. 13
Grandotto, M. 14
Grinspun, E. 30
Gross, M. . . . . ix, 28, 30, 37
Guha, S. 36
Guigon, B. 115
Gulden, W. 14
Guy, A. 18

— **H** —
Hahmann, S. 17, 29, 31
Hancock, P. 27
Hansen, C. 33
Hecht, F. 120
Heckbert, P. S. 32–34, 36, 38, 39
Helenbrook, B. T. 28
Heurtebise, X. 29
Hinker, P. 33
Hoppe, H. . . . . ix, 36, 37
Hussain, M. 36
Hutchinson, M. 27

— **I** —
Iracane, D. 115
Iske, A. 26, 27

— **J** —
Jasak, H. 27
Jeffrey, S. J. 26
Joppich, W. 16
Ju, E. 5

— **K** —
Kanaya, T. 34
Kang, Z. 27
Kaspar, R. 26, 27
Kass, M. 30
Kataoka, H. 13
Keiser, R. 28, 30
Knepley, M. G. 28
Kobbelt, L. 24, 27, 28
Kobori, K.-I. 34
Kumar, S. 87
Kürschner, M. 16

— **L** —
La Greca, R. 17, 20, 25
Lai, H. C. 31
Lamot, M. 87
Laug, P. 24
Lautard, J.-J. 16
Lee, J. 5
Lefebvre, V. 16
Léon, J. 17
Levy, B. 24, 28
Li, T. 27
Li, Y. 39
Lorensen, W. E. 35

Lou, R. 19
Low, K.-L. 34
Lucas, M. 19, 20
Luebke, D. P. 32
Luo, Z. 27

— **M** —
Ma, L. 87
Maculet, R. 17, 19
Manocha, D. . . . . ix, 35
Mantyla, M. 20
Marc, R. 19
Mari, J.-L. 137
Martin, D. 19
Martin, P. 19
Matthies, H. 31
McCallum, B. C. 27
McCormick, J. 23
McDonald, J. 36
Ménegaux, D. 17
Mikchevitch, A. 19
Millard, A. 120
Mirtich, B. 24
Mitchell, T. J. 27
Mochida, A. 13
Molloy, D. 24
Montagnat, J. 29
Montani, C. 32, 35
Moodie, K. B. 26
Moore, P. 24
Müller, M. 28, 30

— **N** —
Nealen, A. 28, 30
Nisan, S. 14
Nishio, K. 34
Nozu, T. 13

— **O** —
Ohtake, Y. 94

— **P** —
Parry, S. R. 25
Pauly, M. . . . . ix, 24, 28–30
Pedrini, H. 39
Perdu, F. 16, 17, 98, 99
Pernot, J.-P. 19
Platt, J. 30
Platt, S. 30
Plemenos, D. 19, 20
Pokorski, W. 23
Popović, J. 37
Porfiri, M.-T. 14

— **R** —
Raffin, R. 25
Raghavan, R. 27
Ragozin, D. L. 27
Rendall, T. C. S. 27
Richebois, E. 16
Richtsmeier, J. T. 27
Robin, S. 80
Rodger, D. 31
Rogers, T. M. 20
Ronfard, R. 37, 38
Rossignac, J. 34, 37, 38
Rossini, M. 27
Rossl, C. 24, 28
Royer, E. 16
Ruchaud, W. 20

— **S** —
Saanouni, K. 24
Santin, G. 23
Sauvage, B. 17, 29
Schaback, R. 27
Schmid, J. 29
Schoot, M. S. v. 26, 27
Schroeder, W. J. 35
Scopigno, R. 32, 35
Sederberg, T. W. 25
Seidel, H.-P. 28, 94
Shaffer, E. 39
Shah, J. J. 20
Shamsuddin, S. 29
Sharma, P. 13
Shirasawa, T. 13
Singh, K. 25
Singh, R. 13

Siréta, P. 16
Sorkine, O. . . . . ix, 26, 29, 32
Srinivasan, R. 27
Staadt, O. G. 37
Stathopoulos, T. 13
Stefani, C. . . . . ix, 19
Stuetzle, W. 36
Suaib, N. 29
Sumner, R. W. . . . . ix, 5, 29–31
Sun, H. 87

— **T** —

Talton, J. O. 32
Tan, T.-S. 34
Terzopoulos, D. 30
Teshima, Y. 34
Theisel, H. 28
Thon, S. 29
Tominaga, Y. 13
Tong, L. 27
Tong, Y. 30
Toumi, I. 14
Tran, N.-V. 39
Tuković, Z. 27
Turk, G. . . . . ix, 33–35

— **V** —

van Heijst, G. 13
Vandroux, S. 17
vanKaick, O. M. 39
Varshney, A. . . . . ix, 35
Véron, P. . . . . ix, 19
Verpeaux, P. 120
Viana, S. A. 31
Vlasic, D. 30
von Funck, W. 28

— **W** —

Weber, H. . . . . ix, 35
Weber, O. 30
Wei, Y. 27
Weiss, F.-P. 16
Wendland, H. 27, 31, 94
Wicke, M. . . . . ix, 28
Witkin, A. 30
Wright, W. . . . . ix, 35
Wyvill, B. 18

— **X** —

Xie, M. 27
Xu, W.-W. 26, 30

— **Y** —

Yahya, S. 29
Yokota, R. 28
Yoshie, R. 13
Yoshikawa, M. 13
Yvart, A. 31

— **Z** —

Zalik, B. 87
Zarge, J. A. 35
Zerroukat, M. 26, 27, 71
Zhao, W. 27
Zhou, K. 26, 30
Zhou, Y. 39
Zhu, Q. 39
Zimmermann, M. A. 16
Zitová, B. 26, 27
Zyda, M. J. . . . . ix, 33

Bibliographie

[AGCA06] Allègre R., Galin E., Chaine R., Akkouche S. : The hybrid-tree : Mixing skeletal implicit surfaces, triangle meshes, and point sets in a free-form modeling system. *Graphical models 68*, 1 (2006), 42–64. (ix et 18)

[AS06] Angelidis A., Singh K. : Space deformations and their application to shape modeling. In *SIGGRAPH '06* (New York, USA, 2006), pp. 10–29. (25)

[Bac05] Bacher P. : *Réacteurs nucléaires. Généralités.* Dossier BN3020, Techniques de l'Ingénieur, 2005. (115)

[Bar84] Barr A. H. : Global and local deformations of solid primitives. *SIGGRAPH 18*, 3 (1984), 21–30. (25)

[BCD*08] Baudron A.-M., Crouzet N., Döderlein C., Geay A., Lautard J.-J., Richebois E., Royer E., Siréta P. : Unstructured 3D MINOS/FLICA4 coupling in SALOME - application to JHR transient analysis. In *Physor* (septembre 2008). (16)

[BCR01] Beatson R. K., Cherrie J. B., Ragozin D. L. : Fast evaluation of radial basis functions : methods for four-dimensional polyharmonic splines. *SIAM journal on mathematical analysis 32*, 6 (2001), 1272–1310. (27)

[BCWG09] Ben-Chen M., Weber O., Gotsman C. : Spatial deformation transfer. In *SCA '09 : Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, USA, 2009), ACM, pp. 67–74. (30)

[BCZD02] Boztosun I., Charafi A., Zerroukat M., Djidjeli K. : Thin-plate spline radial basis function scheme for advection-diffusion problems. *Electronic Journal of Boundary Elements BETEQ 2001 (2)* (2002), 267–282. (26, 27 et 71)

[BG02] Belliard M., Grandotto M. : Computation of two-phase flow in steam generator using domain decomposition and local zoom methods. *Nuclear Engineering and Design 213* (2002), 223–239. (14)

[BG04] Ballagny A., Guigon B. : *Réacteurs de recherche et d'irradiation de matériaux.* Dossier BN3030, Techniques de l'Ingénieur, 2004. (115)

[BH00] Bernardi C., Hecht F. : More pressure in the finite element discretization of the Stokes problem. *Mathematical Modelling and Numerical Analysis 34*, 5 (2000), 953–980. (120)

[BIPM08] Le système international d'unités (SI). Organisation intergouvernementale de la Convention du Mètre, Bureau international des poids

et mesures, 2008. 8^e édition, http://www.bipm.org/utils/common/pdf/si_brochure_8_fr.pdf. (46)

[BK05] Botsch M., Kobbelt L. : Real-time shape editing using radial basis functions. In *Computer Graphics Forum* (2005), pp. 611–621. (27)

[BL10] Bergeaud V., Lefebvre V. : SALOME : a software integration platform for multi-physics, pre-processing and visualisation. *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2010 (SNA-MC'2010)* (2010). (16)

[BLCS05] Borouchaki H., Laug P., Cherouat A., Saanouni K. : Adaptive remeshing in large plastic strain with damage. *International Journal for Numerical Methods in Engineering 63*, 1 (2005), 1–36. (24)

[Blender] Blender. A free open source 3D content creation suite under the GNU General Public License, http://www.blender.org/. (7, 61 et 81)

[Bon06] Bonaccorsi T. : *Modélisation pluridisciplinaire d'expériences d'irradiation dans un réacteur d'irradiation technologique*. Université de la Méditerranée, Septembre 2006. Thèse de doctorat. (ix, 4, 14 et 15)

[Boo89] Bookstein F. L. : Principal warps : Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 6 (1989), 567–585. (27 et 94)

[Bou95] Bouazza M. : *La norme STEP*. Hermes, Paris, France, 1995. (22)

[BPK*08] Botsch M., Pauly M., Kobbelt L., Alliez P., Levy B., Bischo S., Rossl C. : Geometric modeling based on polygonal meshes. In *Eurographics Tutorial* (2008). (24 et 28)

[BPWG07] Botsch M., Pauly M., Wicke M., Gross M. : Adaptive space deformations based on rigid cells. In *Computer Graphics Forum* (2007), vol. 3–26, John Wiley & Sons, pp. 339–347. (ix et 28)

[BR02] Bozzini M., Rossini M. : Testing methods for 3D scattered data interpolation. In *Multivariate Approximation and Interpolation with Applications*, vol. 20 of *Monografías de la academia de ciencias exactas*. Físicas químicas y Naturales de Zaragova, 2002. (27)

[BS08] Botsch M., Sorkine O. : On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on 14*, 1 (2008), 213–230. (ix, 26, 29 et 32)

[BSPG06] Botsch M., Sumner R. W., Pauly M., Gross M. : Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization* (2006), pp. 357–364. (30)

[BVGP09] Baran I., Vlasic D., Grinspun E., Popović J. : Semantic deformation transfer. *ACM Trans. Graph. 28*, 3 (2009), 1–6. (30)

[CAB*11] Chauliac C., Aragonés J.-M., Bestion D., Cacuci D. G., Crouzet N., Weiss F.-P., Zimmermann M. A. : NURESIM -

a european simulation platform for nuclear reactor safety : Multi-scale and multi-physics calculations, sensitivity and uncertainty analysis. *Nuclear Engineering and Design 241*, 9 (2011), 3416–3426. (16)

[CBC*01] Carr J. C., Beatson R. K., Cherrie J. B., Mitchell T. J., Fright W. R., McCallum B. C., Evans T. R. : Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, USA, 2001), ACM, pp. 67–76. (27)

[CCE02] Calvin C., Cueto O., Emonot P. : An objet-oriented approach to the design of fluid mechanics software. *Mathematical Modelling and Numerical Analysis 36*, 5 (2002), 907–921. (120)

[CCMS97] Ciampalini A., Cignoni P., Montani C., Scopigno R. : Multiresolution decimation based on global error. *The Visual Computer 13*, 5 (1997), 228–246. (35)

[CDH*07] Cheutet V., Daniel M., Hahmann S., La Greca R., Léon J., Maculet R., Ménegaux D., Sauvage B. : Constraint modeling for curves and surfaces in CAGD : A survey. *International Journal of Shape Modeling (IJSM) 13*, 2 (2007), 159–199. (17)

[CG09] Chuon C., Guha S. : Volume cost based mesh simplification. *Computer Graphics, Imaging and Visualization, International Conference on* (2009), 164–169. (36)

[CMPS06] Chytracek R., McCormick J., Pokorski W., Santin G. : Geometry description markup language for physics simulation and analysis applications. *IEEE Transactions on Nuclear Science 53*, 5 (2006), 2892–2896. (23)

[CMS98] Cignoni P., Montani C., Scopigno R. : A comparison of mesh simplification algorithms. *Computers & Graphics 22*, 1 (1998), 37–54. (32)

[CVM*96] Cohen J., Varshney A., Manocha D., Turk G., Weber H., Agarwal P., Brooks F., Wright W. : Simplification envelopes. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, USA, 1996), ACM, pp. 119–128. (ix et 35)

[Dan06] Daniel M. : Declarative approach of curve modeling. In *Third International Conference Curves and Surfaces* (Chamonix, France, 2006). (20)

[Dau84] Daubisse J. : *Sur quelques méthodes numériques spécifiques de l'hydrodynamique navale*. Université de Nantes, 1984. Thèse de doctorat. (86 et 91)

[DDGP10a] Duplex B., Daniel M., Gesquière G., Perdu F. : Déformation de maillages et codes de calcul. In *Journées du Groupe de Travail*

Modélisation Géométrique (GTMG'2010) (March 2010), pp. 57–66. (98)

[DDGP10b] Duplex B., Daniel M., Gesquière G., Perdu F. : Transfer of mesh deformations between physical codes. In *Curves and Surfaces conference* (June 2010). (98)

[Del99] Delingette H. : General object reconstruction based on simplex meshes. *Int. J. Comput. Vision 32*, 2 (1999), 111–146. (30)

[DGDP10] Duplex B., Gesquière G., Daniel M., Perdu F. : Echange de déformations entre des codes de calcul utilisant des géométries différentes. In *23èmes journées de l'Association Française d'Informatique Graphique (AFIG)* (November 2010), pp. 117–126. (99)

[DGDP11a] Duplex B., Gesquière G., Daniel M., Perdu F. : Deformation exchange between adjacent physical code geometries. In *The 19-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2011 (WSCG'2011)* (Plzen, Czech Republic, April 2011), pp. 21–24. ISBN 978-80-86943-81-7. (99)

[DGDP11b] Duplex B., Gesquière G., Daniel M., Perdu F. : Deformation exchange between large meshes. *The IADIS Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP)* (July 2011), 239–244. ISBN 978-972-8939-48-9. (98)

[DL97] Daniel M., Lucas M. : Towards declarative geometric modeling in mechanics. In *Integrated design and manufacturing in Mechanical Engineering* (1997), P. Chedmail J.C. Bocquet D. D., (Ed.), Kluwer Academic Publisher, pp. 427–436. (20)

[DLBS*11] De Luca L., Busayarat C., Stefani C., Véron P., Florenzano M. : A semantic-based platform for the digital analysis of architectural heritage. *Computers & Graphics 35*, 2 (2011), 227–241. (ix et 19)

[DR06] Duby C., Robin S. : Analyse en composantes principales. *Institut National Agronomique, Paris-Grignon* (2006). (80)

[dSB07] deBoer A., Schoot M. S. v., Bijl H. : Mesh deformation based on radial basis function interpolation. *Computers & Structures 85*, 11-14 (2007), 784–795. Fourth MIT Conference on Computational Fluid and Solid Mechanics. (26 et 27)

[DZ91] DeHaemer M., Zyda M. J. : Simplification of objects rendered by polygonal approximations. *Computers & Graphics 15* (1991), 175–184. (ix et 33)

[FM04] Fries T., Matthies H. : Classification and overview of meshfree methods. *Department of Mathematics and Computer Science, Technical University of Braunschweig* (2004), 1–64. (31)

[For06] Forax G. : *Les standards d'échange du PLM : Version moderne de la tour de BABEL !* magazine GDT/PLM 132, CAD-magazine, juin-juillet 2006. http://cad-magazine.com/article/gdt_plm/. (21)

[FRR96] Fang S., Raghavan R., Richtsmeier J. T. : Volume morphing methods for landmark based 3D image deformation. In *SPIE International Symposium on Medical Imaging* (1996), pp. 404–415. (27)

[FSRR00] Fang S., Srinivasan R., Raghavan R., Richtsmeier J. T. : Volume morphing and rendering - an integrated approach. *Comput. Aided Geom. Des. 17*, 1 (2000), 59–81. (27)

[Gar99] Garland M. : Multiresolution modeling : Survey & future opportunities. In *Eurographics 1999, State of the Art Report, EG-STAR* (1999), Eurographics Association. (32, 34, 35 et 39)

[Gar02] Gardan Y. : *CAO : modélisation géométrique.* Dossier H3750, Techniques de l'Ingénieur, 2002. (17 et 18)

[Gar03] Gardan Y. : *CAO : vers la modélisation fonctionnelle.* Dossier H3752, Techniques de l'Ingénieur, 2003. (20)

[GB08] Gain J., Bechmann D. : A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics (TOG) 27*, 4 (2008), 107–143. (ix et 25)

[GBSvH11] Gousseau P., Blocken B., Stathopoulos T., van Heijst G. : CFD simulation of near-field pollutant dispersion on a high-resolution grid : A case study by LES and RANS for a building group in downtown montreal. *Atmospheric Environment 45*, 2 (2011), 428–438. (13)

[GDML10] GDML user's guide, 2010. version 2.1. (23)

[Ges00] Gesquière G. : *Calcul et contrôle du volume d'objets implicites.* Université de la Méditerranée, Aix-Marseille II, 2000. Thèse de doctorat. (137)

[GGTF98] Grandotto M., Goldbronn P., Toumi I., Fillion P. : Comparison of finite element and finite volume algorithms for thermohydraulic studies of steam generators and nuclear reactors cores. In *Proceedings of the 10th International Conference on Finite Elements in Fluids* (Tucson, USA, 5-8 January 1998), pp. 94–99. (14)

[GH97] Garland M., Heckbert P. S. : Surface simplification using quadric error metrics. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 209–216. (36, 38 et 39)

[GM97] Gibson S. F. F., Mirtich B. : *A survey of deformable modeling in computer graphics.* Technical report TR97-19, MERL-A Mitsubishi Electric Research Lab, 1997. (24)

[GMCB11] Giraud-Moreau L., Cherouat A., Borouchaki H. : Simulation de mise en forme de structures minces avec remaillage adaptatif couplé à une méthode d'optimisation. *Association Calcul des Structures et Modélisation (CSMA)* (2011). (24)

[GNP*97] Gulden W., Nisan S., Porfiri M.-T., Toumi I., Boubée de Gramont T. : ITER safety analyses with ISAS. *Journal of Fusion Energy 16* (1997), 75–83. (14)

[GS02] Garland M., Shaffer E. : A multiphase approach to efficient surface simplification. In *VIS '02 : Proceedings of the conference on Visualization '02* (Washington, USA, 2002), IEEE Computer Society, pp. 117–124. (39)

[GZ05] Garland M., Zhou Y. : Quadric-based simplification in any dimension. *ACM Trans. Graph. 24*, 2 (2005), 209–239. (39)

[HB03] Hahmann S., Bonneau G. : Polynomial surfaces interpolating arbitrary triangulations. *IEEE Transactions on Visualization and Computer Graphics* (2003), 99–109. (31)

[HBS05] Hahmann S., Bonneau G., Sauvage B. : Déformation de courbes multirésolution sous contraintes. *Revue internationale d'ingénierire numérique 1*, 1 (2005). (29)

[HDD*93] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. : Mesh optimization. In *Proc. of SIGGRAPH-93 : Computer Graphics* (Anaheim, USA, 1993), pp. 19–26. (36)

[HDF5] Hierarchical data format. version 5-1.8.7, `http://www.hdfgroup.org/`. (23)

[Hel03] Helenbrook B. T. : Mesh deformation using the biharmonic operator. *International journal for numerical methods in engineering 56*, 7 (2003), 1007–1021. (28)

[HG97] Heckbert P. S., Garland M. : *Survey of Polygonal Surface Simplification Algorithms*. Tech. rep., CS Dept., Carnegie Mellon U., 1997. (32, 33 et 34)

[HH93] Hinker P., Hansen C. : Geometric optimization. In *VIS '93 : Proceedings of the 4th conference on Visualization '93* (Washington, USA, 1993), IEEE Computer Society, pp. 189–195. (33)

[HH06] Hancock P., Hutchinson M. : Spatial interpolation of large climate data sets using bivariate thin plate smoothing splines. *Environmental Modelling & Software 21*, 12 (2006), 1684–1694. (27)

[Hop96] Hoppe H. : Progressive meshes. In *SIGGRAPH96* (New York, USA, 1996), Rushmeier H., (Ed.), CGPACS, ACM Press/ACM SIGGRAPH, pp. 99–108. (ix, 36 et 37)

[HT08] Heurtebise X., Thon S. : Chapter 6 : Multiresolution representation and deformation of very large volume datasets based on Haar wavelets. In *3rd International Conference on Geometric Modeling and Imaging (GMAI 2008)* (july 2008), pp. 34–40. (29)

[Hus08] Hussain M. : Fast decimation of polygonal models. In *ISVC '08 : Proceedings of the 4th International Symposium on Advances in Vi-*

sual Computing (Berlin, Germany, 2008), Springer-Verlag, pp. 119–128. (36)

[Ira06] Iracane D. : The JHR, a new material testing reactor in Europe. *Nuclear Engineering and Technology (NET'06) 38*, 5 (2006), 437–442. (115)

[Isk03] Iske A. : Radial basis functions : Basics, advanced topics, and meshfree methods for transport equations. In *Rendiconti del Seminario Matematico, Università e Politecnico Torino* (2003), pp. 247–286. (26 et 27)

[JCMB01] Jeffrey S. J., Carter J. O., Moodie K. B., Beswick A. R. : Using spatial interpolation to construct a comprehensive archive of australian climate data. *Environmental Modelling & Software 16*, 4 (2001), 309–330. (26)

[JK06] Joppich W., Kürschner M. : MpCCI – a tool for the simulation of coupled applications. *Concurrency and Computation : Practice and Experience 18*, 2 (2006), 183–192. (16)

[JL08] Ju E., Lee J. : Expressive facial gestures from motion capture data. *Computer Graphics Forum 27*, 2 (2008), 381–388. (5)

[JT10] Jasak H., Tuković Z. : Dynamic mesh handling in OpenFOAM applied to fluid-structure interaction simulations. In *Proceedings of the V European Conference on Computational Fluid Dynamics (ECCOMAS CFD 2010)* (Lisbon, Portugal, 14-17 June 2010), J. C. F. Pereira A. S., Pereira J. M. C., (Eds.). (27)

[KTKN05] Kanaya T., Teshima Y., Kobori K.-I., Nishio K. : A topology-preserving polygonal simplification using vertex clustering. In *GRAPHITE '05 : Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (New York, USA, 2005), ACM, pp. 117–120. (34)

[Kum96] Kumar S. : *Surface Triangulation : A Survey.* Tech. rep., Department of Computer Science, University of North Carolina, July 1996. (87)

[KWT88] Kass M., Witkin A., Terzopoulos D. : Snakes : Active contour models. *International journal of computer vision 1*, 4 (1988), 321–331. (30)

[KZ03] Kaspar R., Zitová B. : Weighted thin-plate spline image denoising. *Pattern Recognition 36*, 12 (2003), 3027–3030. (26 et 27)

[LG05] La Greca R. : *Approche déclarative de la modélisation de surfaces.* Université de la méditerranée (Aix-Marseille), octobre 2005. Thèse de doctorat. (20)

[LMMP89] Lucas M., Martin D., Martin P., Plemenos D. : Le projet ExploFormes, quelques pas vers la modélisation déclarative de formes. *Journées GROPLAN, Strasbourg* (1989), 35–49. (19)

[LPV*10] Lou R., Pernot J.-P., Véron P., Giannimi F., Falcidieno B., Mikchevitch A., Marc R. : Semantic-preserving mesh direct drilling. In *Proceedings of the IEEE Shape Modelling International Conference (SMI'10)* (2010). Aix-en-Provence, France. (19)

[LT97] Low K.-L., Tan T.-S. : Model simplification using vertex-clustering. In *SI3D '97 : Proceedings of the 1997 symposium on Interactive 3D graphics* (New York, USA, 1997), ACM, pp. 75–82. (34)

[LTK09] Luo Z., Tong L., Kang Z. : A level set method for structural shape and topology optimization using radial basis functions. *Computers & Structures 87*, 7-8 (2009), 425–434. (27)

[Lue01] Luebke D. P. : A developer's survey of polygonal simplification algorithms. *IEEE Comput. Graph. Appl. 21*, 3 (2001), 24–35. (32)

[LXZW10] Li T., Xie M., Zhao W., Wei Y. : Shear-warp rendering algorithm based on radial basis functions interpolation. *Computer Modeling and Simulation, International Conference on 2* (2010), 425–429. (27)

[LZ00] Lamot M., Zalik B. : Algorithms for triangulating simple polygons. In *Proceedings of the 22nd International Conference on Information Technology Interfaces (ITI)* (june 2000), pp. 429–436. (87)

[LZ08] Li Y., Zhu Q. : A new mesh simplification algorithm based on quadric error metrics. *Advanced Computer Theory and Engineering, International Conference on 0* (2008), 528–532. (39)

[Mar02] Mari J.-L. : *Modélisation de formes complexes intégrant leurs caractéristiques globales et leurs spécificités locales.* Université de la Méditerranée, Aix-Marseille II, 2002. Thèse de doctorat. (137)

[MD04] Maculet R., Daniel M. : Conception, modélisation géométrique et contraintes en CAO : Une synthèse. *Revue d'Intelligence Artificielle 18*, 5-6 (2004), 619–645. (19)

[MDA01] Montagnat J., Delingette H., Ayache N. : A review of deformable surfaces : Topology, geometry and deformation. *Image and Vision Computing 19*, 14 (2001), 1023–1040. (29)

[MED06] Modélisation et échanges de données, 2006. version 2.3, `http://www.code-aster.org/outils/med/`. (23)

[MM07] Moore P., Molloy D. : A survey of computer-based deformable models. In *Machine Vision and Image Processing Conference, 2007. IMVIP 2007. International* (5-7 2007), pp. 55–66. (24)

[NMK*06] Nealen A., Müller M., Keiser R., Boxerman E., Carlson M. : Physically based deformable models in computer graphics. In *Computer Graphics Forum* (2006), vol. 25, Wiley Online Library, pp. 809–836. (28 et 30)

[OBS05] Ohtake Y., Belyaev A., Seidel H.-P. : 3D scattered data interpolation and approximation with multilevel compactly supported RBFs.

Graphical Models 67, 3 (2005), 150–165. Special Issue on SMI 2003. (94)

[PB81] Platt S., Badler N. : Animating facial expressions. *ACM SIGGRAPH computer graphics 15*, 3 (1981), 245–252. (30)

[PC00] Porfiri M.-T., Cambi G. : Integrated safety analysis code system (ISAS) application for accident sequence analyses. *Fusion Engineering and Design 51-52* (2000), 587–591. (14)

[Per06] Perdu F. : *Proposition d'une interface de couplage en vue des couplages de codes dans Neptune version 1.* Note technique, CEA DEN/DER/SSTH/LMDL, 2006. (16)

[PH97] Popović J., Hoppe H. : Progressive simplicial complexes. *Computer Graphics 31*, Annual Conference Series (1997), 217–224. (37)

[Ple91] Plemenos D. : *Contribution à l'étude et au développement de techniques de modélisation, génération et visualisation de scènes : le projet MultiFormes.* Université de Nantes, 1991. Thèse d'Etat. (20)

[RA10] Rendall T. C. S., Allen C. B. : Reduced surface point selection options for efficient mesh deformation using radial basis functions. *Journal of Computational Physics 229*, 8 (2010), 2810–2820. (27)

[RB93] Rossignac J., Borrel P. : Multi-resolution 3D approximations for rendering complex scenes. In *Geometric Modeling in Computer Graphics* (1993), Springer Verlag, pp. 455–465. (34)

[RGLG07] Raffin R., Gesquière G., La Greca R. : *Déformations de modèles géométriques.* Tech. Rep. LSIS.RR.2007.001, LSIS, Janvier 2007. (25)

[RP02] Ruchaud W., Plemenos D. : Multiformes : modélisation déclarative et moteurs de génération de scènes. In *Journées du Groupe de Travail Modélisation Géométrique (GTMG'2002)* (2002), GTMG. (20)

[RR96] Ronfard R., Rossignac J. : Full-range approximation of triangulated polyhedra. In *Proceeding of Eurographics, Computer Graphics Forum* (August 1996), Rossignac J., Sillon F., (Eds.), vol. 15(3), Eurographics, Blackwell, pp. 67–76. (37 et 38)

[Ruc01] Ruchaud W. : *Étude et réalisation d'un moteur de résolution de contraintes géométriques pour la modélisation déclarative.* Université de Limoges, 2001. Thèse de doctorat. (20)

[Sau05] Sauvage B. : *Déformation de courbes et surfaces multirésolution sous contraintes.* Institut National Polytechnique de Grenoble, 2005. Thèse de doctorat. (29)

[SG98] Staadt O. G., Gross M. : Progressive tetrahedralizations. In *VIS '98 : Proceedings of the conference on Visualization '98* (Los Alamitos, USA, 1998), IEEE Computer Society Press, pp. 397–402. (37)

[SG05] Shaffer E., Garland M. : A multiresolution representation for massive meshes. *IEEE Transaction on Visualization and Computer Graphics 11* (2005), 139–148. (39)

[SGS11] Sharma P., Gera B., Singh R. : A CFD validation of fire dynamic simulator for corner fire. *CFD Letters 2*, 4 (2011), 137–148. (13)

[Sha88] Shah J. J. : Feature transformations between application-specific feature spaces. *Computer-Aided Engineering Journal 5*, 6 (Dec 1988), 247–255. (20)

[SM95] Shah J. J., Mantyla M. : *Parametric and Feature Based CAD/CAM : Concepts, Techniques, and Applications.* John Wiley & Sons, Inc., New York, USA, 1995. (20)

[SM09] Sun H., Ma L. : A new triangulation algorithm based on the determination of the polygon's diagonals. In *International Conference on Computational Intelligence and Software Engineering (CiSE)* (2009), pp. 1–3. (87)

[SP86] Sederberg T. W., Parry S. R. : Free-form deformation of solid geometric models. In *SIGGRAPH '86* (New York, USA, 1986), pp. 151–160. (25)

[SP04] Sumner R. W., Popović J. : Deformation transfer for triangle meshes. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers* (New York, USA, 2004), ACM, pp. 399–405. (ix, 5, 30 et 31)

[SR88] Shah J. J., Rogers T. M. : Functional requirements and conceptual design of the feature-based, modelling system. *Computer-Aided Engineering Journal 5*, 1 (Feb 1988), 9–15. (20)

[SSP07] Sumner R. W., Schmid J., Pauly M. : Embedded deformation for shape manipulation. In *SIGGRAPH '07* (New York, USA, 2007), pp. 80–88. (29)

[SW00] Schaback R., Wendland H. : Characterization and construction of radial basis functions. In *Multivariate Approximation and Applications* (2000), Cambridge University Press, pp. 1–24. (27)

[SZL92] Schroeder W. J., Zarge J. A., Lorensen W. E. : Decimation of triangle meshes. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 65–70. (35)

[Tal04] Talton J. O. : *A Short Survey of Mesh Simplification Algorithms.* Rapport technique, University of Illinois at Urbana-Champaign, 2004. (32)

[Ter03] Terzopoulos D. : Deformable models : classic, topology-adaptive and generalized formulations. *Geometric level set methods in imaging, vision, and graphics* (2003), 21–40. (30)

[TMY*08] Tominaga Y., Mochida A., Yoshie R., Kataoka H., Nozu T., Yoshikawa M., Shirasawa T. : AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. *Journal of Wind Engineering and Industrial Aerodynamics 96*, 10-11 (2008), 1749–1761. 4th International Symposium on Computational Wind Engineering (CWE2006). (13)

[TPBF87] Terzopoulos D., Platt J., Barr A. H., Fleischer K. : Elastically deformable models. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, USA, 1987), ACM, pp. 205–214. (30)

[Tra08] Tran N.-V. : *Traitement de surfaces triangulées pour la construction de modèles géologiques structuraux.* Université de la Méditerranée, 2008. Thèse de doctorat. (39)

[Tur92] Turk G. : Re-tiling polygonal surfaces. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 55–64. (ix, 33 et 34)

[vFTS06] von Funck W., Theisel H., Seidel H.-P. : Vector field based shape deformations. *ACM Trans. Graph. 25*, 3 (2006), 1118–1125. (28)

[VMCC89] Verpeaux P., Millard A., Charras T., Combescure A. : A modern approach of large computer code for structural analysis. In *The* 10^{th} *SMIRT* (Los Angeles, USA, 1989). (120)

[vP06] vanKaick O. M., Pedrini H. : A comparative evaluation of metrics for fast mesh simplification. *Computer Graphics Forum 25*, 2 (2006), 197–210. (39)

[VP10] Vandroux S., Perdu F. : *Couplage Système/CFD monophasique utilisant les codes CATHARE et TRIO_U : cas tests de validation sur un circuit en hélium.* Note technique NT/2010-020/A, CEA/DEN/CAD/DER/SSTH/LDAL, 2010. (17)

[VRL07] Viana S. A., Rodger D., Lai H. C. : Overview of meshless methods. *ICS Newsletter 14*, 2 (november 2007), 1–4. (31)

[Wen95] Wendland H. : Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math. 4*, 1 (1995), 389–396. (27)

[Wen05a] Wendland H. : Computational aspects of radial basis function approximation. In *Topics in Multivariate Approximation and Interpolation, Volume 12 (Studies in Computational Mathematics)*. Elsevier Science Inc., New York, USA, 2005, pp. 231–256. (27)

[Wen05b] Wendland H. : *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics (No. 17). Cambridge University Press, 2005. (27, 31 et 94)

[WGG99] Wyvill B., Guy A., Galin E. : Extending the CSG tree. Warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum 18*, 2 (1999), 149–158. (18)

[XZ09] Xu W.-W., Zhou K. : Gradient domain mesh deformation – a survey. *Journal of Computer Science and Technology 24* (2009), 6–18. (26)

[YBK10] Yokota R., Barba L., Knepley M. G. : PetRBF – a parallel O(N) algorithm for radial basis function interpolation with gaussians.

Computer Methods in Applied Mechanics and Engineering 199, 25-28 (2010), 1793–1804. (28)

[YHB05] Yvart A., Hahmann S., Bonneau G. : Hierarchical triangular splines. *ACM Transactions on Graphics (TOG) 24*, 4 (2005), 1374–1391. (31)

[YSBS06] Yahya S., Suaib N., Bade A., Shamsuddin S. : Solid deformation modeling techniques. In *Computing Informatics, 2006. ICOCI '06. International Conference on* (june 2006), pp. 1–14. (29)

[ZXTD10] Zhou K., Xu W.-W., Tong Y., Desbrun M. : Deformation transfer to multi-component objects. In *Computer Graphics Forum* (2010), vol. 2–29, John Wiley & Sons, pp. 319–325. (30)

Résumé : Le CEA développe et utilise des logiciels de calcul, également appelés codes de calcul, dans différentes disciplines physiques pour optimiser les coûts de ses installations et de ses expérimentations. Lors d'une étude, plusieurs phénomènes physiques interagissent. Un couplage et des échanges de données entre plusieurs codes sont nécessaires.
Chaque code réalise ses calculs sur une géométrie, généralement représentée sous forme d'un maillage contenant des milliers voire des millions de mailles. Cette thèse se focalise sur le transfert de déformations géométriques entre les maillages spécifiques de chacun des codes de calcul couplés. Pour cela, elle présente une méthode de couplage de plusieurs codes, dont le calcul des déformations est réalisé par l'un d'entre eux. Elle traite également de la mise en place d'un modèle commun aux différents codes de l'étude regroupant l'ensemble des données partagées. Enfin, elle porte sur les transferts de déformations entre des maillages représentant une même géométrie ou des géométries adjacentes. Les modifications géométriques sont de nature discrète car elles s'appuient sur un maillage. Afin de les rendre accessible à l'ensemble des codes de l'étude et pour permettre leur transfert, une représentation continue est calculée. Pour cela, deux fonctions sont développées : l'une à support global, l'autre à support local. Toutes deux combinent une méthode de simplification et un réseau de fonctions de base radiale. Un cas d'application complet est traité dans le cadre du réacteur Jules Horowitz. L'effet des dilatations différentielles sur le refroidissement d'un dispositif expérimental est étudié.

Mots-clés : transfert de déformations, fonction de base radiale, simplification de maillage, métrique d'erreur quadratique, modèle géométrique, couplage de codes de calcul, dispositif expérimental

Abstract: The CEA develops and uses scientific software, called physical codes, in various physical disciplines to optimize installation and experimentation costs. During a study, several physical phenomena interact, so a code coupling and some data exchanges between different physical codes are required.
Each physical code computes on a particular geometry, usually represented by a mesh composed of thousands to millions of elements. This PhD Thesis focuses on the geometrical modification transfer between specific meshes of each coupled physical code. First, it presents a physical code coupling method where deformations are computed by one of these codes. Next, it discusses the establishment of a model, common to different physical codes, grouping all the shared data. Finally, it covers the deformation transfers between meshes of the same geometry or adjacent geometries. Geometrical modifications are discrete data because they are based on a mesh. In order to permit every code to access deformations and to transfer them, a continuous representation is computed. Two functions are developed, one with a global support, and the other with a local support. Both functions combine a simplification method and a radial basis function network. A whole use case is dedicated to the Jules Horowitz reactor. The effect of differential dilatations on experimental device cooling is studied.

Keywords: deformation transfer, radial basis function, mesh simplification, quadric error metric, geometrical model, physical code coupling, experimental device