Interaction Fluide-Structure pour les corps élancés

Guillaume de Nayer

▶ **To cite this version:**

Guillaume de Nayer. Interaction Fluide-Structure pour les corps élancés. Mécanique des fluides [physics.class-ph]. Ecole Centrale de Nantes (ECN), 2008. Français. NNT : . tel-00820472

HAL Id: tel-00820472

https://theses.hal.science/tel-00820472

Submitted on 5 May 2013

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.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

École Centrale de Nantes

École Doctorale

Sciences Pour l'Ingénieur, Géosciences, Architecture

Année 2008 N° B.U. :

Thèse de Doctorat

Spécialité : Dynamique des fluides et des transferts

Présentée et soutenue publiquement par :

Guillaume De Nayer

le 17 Décembre 2008
à l'Ecole Centrale Nantes (Nantes)

Interaction Fluide-Structure pour les corps élancés

Jury

Président :	Nicolas MOES	Professeur, École Centrale Nantes
Rapporteurs :	Serge HUBERSON	Professeur, Université de Poitiers
	Bernard MOLIN	Enseignant-chercheur HDR, École Centrale Marseille
Examinateurs :	Frédéric BOYER	Maître-assistant HDR, École des Mines de Nantes
	Alban LEROYER	Maître de conférence, École Centrale Nantes
	Élisabeth LONGATTE	Ingénieur-chercheur, EDF-R&D, Département MFTT
	Mhamed SOULI	Professeur, Université de Lille
	Michel VISONNEAU	Directeur de recherche CNRS, École Centrale Nantes

Directeur de Thèse : Michel VISONNEAU
Laboratoire de Mécanique des Fluides CNRS UMR 6598 N° ED 498-025

Remerciements

Ce travail de thèse a été réalisé au sein de l'Équipe Modélisation Numérique du Laboratoire de Mécanique des Fluides de l'École Centrale Nantes.

Je tiens d'abord à remercier Messieurs Gérard DELHOMMEAU et Alain CLEMENT, directeurs du Laboratoire, pour m'avoir accueilli au sein du LMF, où j'ai pu préparer cette thèse dans de très bonnes conditions.

J'exprime la plus grande gratitude à mon directeur de thèse Michel VISONNEAU. Après m'avoir enseigné la mécanique des fluides numérique en école d'ingénieur, puis épaulé en Master, il m'a permis d'entreprendre cette thèse en m'accueillant au sein de l'Équipe Modélisation Numérique (EMN). Le sujet qu'il m'a proposé est actuel et je l'ai trouvé très intéressant.

Mes remerciements vont également à Alban LEROYER, qui m'a supporté tout au long de ce travail. Son rôle de co-encadrant n'a pas toujours été facile. Pourtant, il a su répondre rapidement à mes incessants problèmes et interrogations. Il m'a aussi orienté dans les bonnes directions pendant les moments de doute et de perdition.

J'exprime toute ma reconnaissance à Messieurs Bernard MOLIN et Serge HUBERSON pour avoir accepté d'être mes rapporteurs. Leurs conseils et critiques m'ont été très utiles pour la rédaction de ce manuscrit.

Je remercie vivement Monsieur Nicolas MOES d'avoir été président de mon jury.

J'exprime aussi tout particulièrement ma gratitude à un autre membre du jury, Monsieur Frédéric BOYER. Sans son aide précieuse, la compréhension des poutres de Cosserat ainsi que celle de l'approche géométriquement exacte aurait été impossible. De plus, son humour et sa décontraction permettent des échanges aisés et agréables.

Je remercie également Monsieur Mhamed SOULI et Madame Élisabeth LONGATTE d'avoir participé à mon jury.

Les membres permanents de l'EMN ont évidemment énormément contribué à ce travail de thèse, je tiens donc à les remercier chacun personnellement :

- Monsieur Patrick QUEUTEY, maître incontesté du sed, awk et du script bash, qui m'a constamment aidé au cours de ces trois années ; c'est sans doute grâce à lui que m'est venue la passion du monde opensource et en particulier Debian GNU/Linux[1].

[1] http://www.debian.org

- Monsieur Émmanuel GUILMINEAU, gourou LaTeX de l'équipe, qui a accompagné mes premiers pas dans ce langage de programmation formidable.
- Monsieur Gan Bo DENG, qui a la faculté de faire disparaître les problèmes informatiques et ce même à deux étages de distance !
- Monsieur Jean PIQUET pour m'avoir fait découvrir et aimer la mécanique des fluides numérique et la modélisation de la turbulence.
- Monsieur Guy CAPDEVILLE pour m'avoir enseigné la mécanique des fluides compressibles.

Je tiens aussi à remercier chaleureusement les membres non-permanents de l'équipe :
- Davy DAUBY, ancien doctorant, qui a été mon partenaire de babyfoot pendant les pauses et aussi un ami.
- Oussama CHICKAOUI et Prio avec lesquels j'ai passé de très bons moments.
- Mathieu DURAND, doctorant, avec lequel les discussions sont toujours constructives.

Une pensée va aussi aux illustres inconnus des listes de diffusion suivantes :
- Ceux de la liste debian-user-french, pour leur aide inestimable dans la configuration, l'utilisation de la distribution Debian et des commandes Unix en général.
- Ceux de la liste Gutenberg, qui ont permis grâce à leurs compétences en LaTeX la rédaction de ce manuscrit.

Vient ensuite ma famille à laquelle je voudrais adresser un grand merci ; en particulier, merci à mes parents pour m'avoir soutenu tout au long de cette thèse, pour l'avoir relue, en bref pour toute l'aide apportée (sans oublier les fameux gâteaux au chocolat, source d'inspiration et de réconfort !).

Évidemment, rien n'aurait été possible sans le soutien de mes amis. Dans le désordre, je remercie affectueusement Julien et Arnaud CADO, qui sont à mes côtés depuis des temps immémoriaux, Baptiste FARA, Julien BARRETEAU, Alexandre KTOURZA, Grégory PUJALS, Laurine CHEVALIER, Marie GRANGER, Aurélien HÉNON et tous les autres.

Enfin, je tiens à exprimer une reconnaissance toute particulière à Déborah JUNGBLUTH, ma compagne, pour m'avoir supporté (aux sens multiples du terme). Les voyages et promenades sont, avec elle, toujours un plaisir et une source d'aventures et de découvertes.

À mes parents...

Table des matières

Table des matières i

Table des figures vii

Liste des tableaux xiii

Nomenclature xv

Introduction 1

1 Interactions Fluide-Structure (IFS) 5
1.1 Généralités sur une discipline très vaste . . . 5
1.1.1 Problèmes d'IFS avec fluide incompressible . . . 5
1.1.1.1 Génie civil . . . 7
1.1.1.2 Industrie nucléaire . . . 7
1.1.1.3 Industrie maritime et navale . . . 10
1.1.1.4 Domaine biomédical . . . 11
1.1.1.5 Génie industriel . . . 11
1.1.2 Problèmes d'IFS avec fluide peu compressible . . . 13
1.1.3 Problèmes d'IFS avec fluide compressible . . . 13
1.1.4 Problèmes d'IFS et ondes acoustiques . . . 14
1.2 IFS d'un point de vue numérique . . . 15
1.2.1 Stratégies et algorithmes de couplage . . . 15
1.2.1.1 Algorithme de couplage en temps explicite synchrone et décalé . . . 17
1.2.1.2 Algorithme de couplage en temps parallèle . . . 18
1.2.1.3 Algorithme de couplage en temps itéré . . . 18
1.2.1.4 Algorithmes appliqués à un problème simple . . . 19
1.2.2 Formulations et discrétisations . . . 21
1.3 Conclusion . . . 22

2 Solveur de dynamique des fluides : ISIS 25
2.1 Des lois de conservation aux équations de Navier-Stokes . . . 25
2.1.1 Examen du point de vue adopté . . . 25
2.1.2 Dérivation selon un champ de vitesse quelconque . . . 26
2.1.3 Formulation Volumes-Finis des équations de Navier-Stokes . . . 27
2.1.4 Évaluation des intégrales . . . 28

2.1.4.1 Intégrales de volume . . . 28
2.1.4.2 Intégrales de surface . . . 28
2.2 Résolution numérique des équations de Navier-Stokes dans ISIS . . . 28
2.2.1 Équations de Navier-Stokes en moyenne de Reynolds . . . 29
2.2.2 Discrétisation des équations de transport . . . 30
2.2.3 Reconstruction sur les faces . . . 32
2.2.3.1 Reconstruction centrée . . . 32
2.2.3.2 Reconstruction décentrée . . . 34
2.2.4 Calcul des gradients sur une cellule . . . 38
2.2.4.1 Méthode des moindres carrés pondérée par les distances . 38
2.2.4.2 Méthode de Gauss . . . 40
2.2.5 Algorithme de couplage . . . 40
2.2.5.1 Équation de pression . . . 40
2.2.5.2 Algorithme de résolution . . . 42
2.2.6 Résolution des systèmes linéaires . . . 43
2.3 Modélisation de la turbulence . . . 44
2.3.1 Modèles à viscosité turbulente . . . 44
2.3.1.1 Modèle de Spalart-Allmaras . . . 44
2.3.1.2 Modèle $k-\omega$ de Wilcox . . . 46
2.3.1.3 Modèle $k-\omega$ SST de Menter . . . 46
2.4 Conclusion . . . 48

3 Solveur de dynamique des structures : PTAH 49
3.1 Arrière-plan théorique . . . 49
3.1.1 Présentation, modélisation et hypothèses . . . 49
3.1.2 Cinématique de base des poutres fines . . . 51
3.1.3 Paramétrisation du champ de rotation . . . 53
3.1.3.1 Description lagrangienne totale . . . 55
3.1.3.2 Description lagrangienne réactualisée . . . 55
3.1.3.3 Description eulérienne . . . 55
3.1.4 Mise à jour de la configuration de la poutre . . . 55
3.1.4.1 Réactualisation du champ de position . . . 55
3.1.4.2 Réactualisation du champ de rotation . . . 56
3.1.5 Formulation variationnelle faible du modèle d'Euler-Bernoulli . . . 58
3.1.6 Linéarisation et discrétisation du modèle d'Euler-Bernoulli . . . 60
3.1.6.1 Linéarisation de la forme faible . . . 60
3.1.6.2 Discrétisation : méthode des éléments finis . . . 62
3.1.6.3 Intégration temporelle : schéma de Newmark . . . 63
3.1.7 Modèle de Rayleigh . . . 64
3.1.8 Mise à jour de la forme discrétisée des modèles . . . 66
3.1.8.1 Réactualisation du champ de position . . . 67
3.1.8.2 Réactualisation du champ de tangente de l'espace matériel 67
3.1.8.3 Réactualisation du champ de courbure de l'espace matériel 67
3.1.8.4 Réactualisation du champ de vitesse linéaire . . . 67
3.1.8.5 Réactualisation du champ d'accélération linéaire . . . 67
3.1.8.6 Réactualisation du champ de rotation . . . 68

3.1.8.7 Réactualisation du champ de taux de torsion 68
3.1.8.8 Réactualisation du champ de vitesse et d'accélération angulaire de torsion 68
3.1.9 Problème des mouvements imposés aux extrémités de la poutre . . 69
3.2 Validation du solveur poutre 71
3.2.1 Validation en 2D 71
3.2.1.1 Cas des petits déplacements 71
3.2.1.2 Cas des grands déplacements 78
3.2.2 Validation en 3D 80
3.2.2.1 Validation du couplage flexion-torsion 83
3.2.2.2 Validation du modèle de Rayleigh en 3D 88
3.3 Conclusion 100

4 Couplage en espace : remaillage 101
4.1 Méthodes de remaillage déjà présentes dans **ISIS** 101
4.1.1 Remaillage par mouvement en bloc 102
4.1.2 Méthode des ressorts 102
4.1.3 Méthode de remaillage par pondération analytique 103
4.2 Module de remaillage pseudo-structure 103
4.2.1 Arrière-plan théorique 103
4.2.1.1 Rappels sur les contraintes et les déformations 104
4.2.1.2 Élasticité linéaire 106
4.2.2 Discrétisations 107
4.2.2.1 Discrétisation isotrope 107
4.2.2.2 Discrétisation avec coefficient de cisaillement variable . . . 108
4.2.2.3 Discrétisation avec forte implicitation 112
4.2.3 Intégration dans **ISIS** 114
4.2.3.1 Calcul des coefficients nécessaires à la résolution du système 114
4.2.3.2 Calcul des déplacements aux frontières 116
4.2.3.3 Résolution du problème 117
4.2.3.4 Imposition des conditions aux limites dans la résolution . 117
4.2.3.5 Interpolation de la solution du centre des cellules aux nœuds 117
4.2.3.6 Méthode de correction des erreurs d'interpolation 118
4.2.3.7 Imposition des conditions initiales aux nœuds 121
4.2.3.8 Algorithme de résolution rapide 122
4.2.4 Validation du module de remaillage 123
4.2.5 Exemples de remaillage 124
4.2.5.1 Déformations de poutres (maillage GRIDGEN ™) 124
4.2.5.2 Le robot anguille : *ROBEA/RAAMO* (maillage GRIDGEN ™) 128
4.2.5.3 La configuration "batteur-caisse" (maillage GRIDGEN ™) . 129
4.2.5.4 Corps sphérique en translation imposée (maillage HEXPRESS ™) 129
4.2.5.5 Projet *VIRTUE* : optimisation d'une coque de bateau (maillage HEXPRESS ™) 137
4.2.5.6 Le requin (maillage HEXPRESS ™) 137

4.3 Couplage des solutions de remaillage : la solution idéale ? . . . 144
4.4 Conclusion . . . 150

5 Interface fluide-structure et algorithme de couplage temporel 151
5.1 Couplage en espace . . . 151
5.1.1 État de l'art . . . 152
5.1.1.1 Les différentes méthodes d'interpolation entre des maillages non-conformes . . . 152
5.1.1.2 Comparaison et performances de ces méthodes . . . 153
5.1.2 Choix de l'interpolation . . . 153
5.2 Couplage en temps . . . 158
5.3 Détails sur le couplage IFS . . . 160
5.4 Conclusion . . . 160

6 Applications d'IFS réalisées 163
6.1 Applications 2D . . . 163
6.1.1 Plaque dans une conduite . . . 163
6.1.2 Filament fixé derrière un bloc dans un écoulement . . . 170
6.2 Cas-tests 3D . . . 184
6.2.1 Câble fixé à une extrémité et soumis à un courant . . . 184
6.2.2 Câble tracté dans un environnement multifluide . . . 191
6.2.3 Riser : VIV . . . 195
6.3 Conclusion . . . 210

Conclusion générale et perspectives 211

A Calcul des coefficients de la matrice orthotrope en 3D pour le remailleur (MAPLE ™) 215

B Quaternions et composition de rotations 249
B.1 Un peu d'histoire . . . 249
B.2 L'algèbre des quaternions $\mathbb{H}$ . . . 249
B.2.1 Définition . . . 249
B.2.2 Opérations dans l'algèbre des quaternions $\mathbb{H}$ . . . 250
B.2.2.1 Conjugaison . . . 250
B.2.2.2 Norme . . . 251
B.2.3 L'ensemble $\mathbb{P}$ des quaternions purs . . . 251
B.2.3.1 Définition . . . 251
B.2.3.2 Propriétés . . . 251
B.2.3.3 Notations . . . 251
B.2.4 L'ensemble $\mathbb{S}$ des quaternions de norme 1 . . . 252
B.2.4.1 Isomorphisme avec $GL(\mathbb{R}^3)$ . . . 252
B.2.4.2 Lien entre quaternion et matrice de passage . . . 252
B.2.5 La dérivation temporelle d'un quaternion . . . 253
B.2.5.1 Propriété (utile pour la suite) . . . 253
B.2.6 Représentation matricielle de $(\mathbf{e}, \mathbf{i}, \mathbf{j}, \mathbf{k})$ . . . 253

B.3 Relation entre rotation instantanée et quaternions . . . 254
B.3.1 Relation entre $\mathbf{Q}$ et $\mathbf{\Omega_0}$ . . . 255
B.3.1.1 Conclusion . . . 256
B.3.2 Relation entre $\mathbf{Q}$ et $\mathbf{\Omega_S}$ . . . 256
B.4 Rotation, matrices de passage et changement de base . . . 257
B.4.1 Matrices de passage . . . 257
B.4.1.1 Définition . . . 257
B.4.1.2 Propriétés . . . 257
B.4.2 Changement de base . . . 257
B.4.3 Expression des vecteurs images de $R_{0\to1}$ . . . 258
B.5 Composition de rotations successives . . . 258
B.5.1 Position du problème . . . 258
B.5.2 Expression en terme de matrices de passage . . . 259
B.5.2.1 Rotations successives s'appuyant sur la base courante . . . 259
B.5.2.2 Rotations successives s'appuyant sur la base de référence . 259
B.5.2.3 Relation entre $P_{\mathcal{B}_0\to\mathcal{B}_3}$ et $P_{\mathcal{B}_1\to\mathcal{B}_2}$ . . . 260
B.5.3 Expression en terme de quaternions . . . 260
B.5.3.1 Généralisation . . . 261
B.5.3.2 Exemple . . . 262

C Résolution de l'équation d'orientation par les quaternions 263
C.1 Avant-propos . . . 263
C.2 Problème . . . 263
C.3 Cas de l'évolution temporelle du quaternion d'orientation . . . 264
C.3.1 Calcul de l'exponentielle de $\mathbb{A}$ . . . 264
C.4 Cas de l'évolution spatiale du quaternion d'orientation . . . 265
C.4.1 Calcul de l'exponentielle de $\mathbb{K}$ . . . 266

D Communication au congrès FIV2008 267
D.1 Description . . . 267
D.2 Long abstract . . . 267
D.3 Errata . . . 274

Bibliographie 275

Table des figures

1.1 Diverses études d'ouverture et d'écoulement d'interactions fluide-structure autour de parachutes . . . 6
1.2 Test en soufflerie d'un modèle du pont de Tacoma de 1940 . . . 7
1.3 Diverses études numériques de simulation d'écoulement autour de profils de pont . . . 8
1.4 Effondrement du pont de Tacoma sous l'action du vent en 1940 . . . 8
1.5 Simulations numériques et expérimentales autour d'un faisceau de tubes rencontrés dans l'industrie nucléaire . . . 9
1.6 Variétés des structures sous-marines et off-shore rencontrées dans le domaine de l'industrie maritime . . . 10
1.7 Simulations numériques d'écoulements de sang . . . 11
1.8 Modélisation complexe d'un cœur immergé dans un fluide et de l'aorte . . . 12
1.9 Modélisation et simulation numérique du phénomène de sloshing dans un camion-citerne . . . 12
1.10 Simulation numérique du phénomène de sloshing dans des cuves de l'industrie aérospatiale . . . 13
1.11 Modélisation et simulation numérique d'un F-16 (maillage de la structure à l'interface et maillage fluide à l'interface) . . . 13
1.12 Modélisation et simulation numérique de l'ouverture d'un airbag . . . 14
1.13 Modélisation et simulation numérique de l'acoustique d'une guitare en fonction du pincement initial de la corde . . . 14
1.14 Schémas de couplage en temps explicite . . . 18
1.15 Schéma de couplage en temps parallèle . . . 18
1.16 Schéma de couplage en temps itéré ou implicite . . . 19

2.1 Reconstruction sur une face . . . 32
2.2 Variation de Q dans la direction de l'écoulement . . . 35
2.3 Critères de monotonie en variables normalisées . . . 36
2.4 Définition du point imaginaire U . . . 37
2.5 Schéma GDS en variables normalisées . . . 38
2.6 Cellule C_0 et son voisinage . . . 39

3.1 Cinématique de base . . . 52
3.2 La transformation finie compatible avec la cinématique de base . . . 54
3.3 Mise à jour sur SO3 de l'angle itératif θ . . . 57

3.4 Déformée théorique et déformée calculée numériquement ($q = 1000$) (unités SI) . . . 73
3.5 Poutre encastrée en A dans un fluide au repos . . . 73
3.6 Oscillations du point B dans le vide (modèle Euler-Bernoulli) . . . 74
3.7 Fréquences propres des oscillations de la poutre (modèle Euler-Bernoulli) . . . 75
3.8 Oscillations du point B dans le vide . . . 76
3.9 Fréquences propres des oscillations de la poutre . . . 76
3.10 Oscillations amorties du point B dans le vide . . . 77
3.11 Déformée finale après amortissement (unités SI) . . . 77
3.12 Oscillations amorties du point B dans le vide . . . 78
3.13 Déformée finale après amortissement . . . 79
3.14 Déformée calculée numériquement ($F = 14\,SI$) . . . 80
3.15 Oscillations du point B dans le vide imposées par la première loi horaire . . . 81
3.16 Oscillations du point B dans le vide imposées par la seconde loi horaire . . . 82
3.17 Déformée 3D projetée dans le plan $(\overrightarrow{E_1}, \overrightarrow{E_2})$ . . . 84
3.18 Déformée 3D projetée dans le plan $(\overrightarrow{E_2}, \overrightarrow{E_3})$ . . . 85
3.19 Déformée 3D projetée dans le plan $(\overrightarrow{E_1}, \overrightarrow{E_3})$ . . . 86
3.20 Poutre au repos et déformée 3D . . . 87
3.21 Déformée 3D projetée dans le plan $(\overrightarrow{x}, \overrightarrow{y})$ . . . 89
3.22 Déformée 3D projetée dans le plan $(\overrightarrow{y}, \overrightarrow{z})$ . . . 90
3.23 Déformée 3D projetée dans le plan $(\overrightarrow{x}, \overrightarrow{z})$ . . . 91
3.24 Déformée 3D de la poutre pour $\alpha = 5$ . . . 92
3.25 Configuration initiale de la poutre volante . . . 93
3.26 Projection des nœuds de la poutre dans le plan $(\overrightarrow{x}, \overrightarrow{y})$ . . . 94
3.27 Projection des nœuds de la poutre dans le plan $(\overrightarrow{y}, \overrightarrow{z})$ . . . 95
3.28 Projection des nœuds de la poutre dans le plan $(\overrightarrow{x}, \overrightarrow{z})$ . . . 96
3.29 Trajectoires des extrémités de la poutre . . . 97
3.30 Déplacement de l'extrémité $X = 0$ . . . 98
3.31 Déplacement de l'extrémité $X = L$ . . . 99

4.1 Ressorts de compression et de torsion en 2D . . . 102
4.2 Coefficient de pondération . . . 103
4.3 Bases locales créées sur les lignes isométriques du coefficient de pondération 109
4.4 Comparaison entre le maillage généré avec les différentes discrétisations sur un mouvement de rotation pure . . . 112
4.5 Schéma de la procédure de remaillage . . . 115
4.6 X_{ref} = maillage intérieur + nœuds frontières du maillage initial N_1, N_2 et N_3
$X_{ref_{def}}$ = maillage intérieur + nœuds frontières $N_1^{'}$, $N_2^{'}$ et $N_3^{'}$, déplacés suivant la nouvelle position du corps . . . 116
4.7 Déplacements aux frontières . . . 117
4.8 Données avant interpolation : maillages X_{ref}, $X_{ref_{def}}$ et vecteurs déplacements aux centres des cellules $\overrightarrow{U}_{cell}$ . . . 118
4.9 Cas géométriques posant problème pour l'interpolation aux nœuds . . . 119
4.10 Réinterpolation aux nœuds de l'erreur faite aux centres des cellules . . . 120

4.11 Influence de la méthode de correction d'erreur d'interpolation dans le cas des cellules très étirées . . . 121
4.12 Configuration problématique pour le calcul parallèle . . . 122
4.13 Ordre de l'erreur sur un maillage non structuré . . . 124
4.14 Déformation d'une poutre (cf. cas-test IFS à la section 6.1.1) . . . 125
4.15 Remaillage autour d'une poutre (cf. cas-test IFS à la section § 6.1.2) . . . 126
4.16 Remaillage autour d'une poutre subissant une charge fixe imposée (maillage fin) . . . 127
4.17 Remaillage autour de l'anguille sur un maillage 3D avec un cisaillement variable localement . . . 128
4.18 Présentation globale du cas-test du "batteur-caisse" . . . 130
4.19 Test de remaillage avec un mouvement de translation pure de 5 fois la longueur de la caisse . . . 131
4.20 Test de remaillage avec un mouvement de rotation pure de 45° . . . 132
4.21 Passage de la seconde vague au niveau de la caisse . . . 133
4.22 Détail de la déformation du maillage autour de la caisse au passage de la seconde vague . . . 134
4.23 Maillage HEXPRESS™ autour d'un corps sphérique . . . 135
4.24 Maillage HEXPRESS™ autour du corps sphérique translaté . . . 136
4.25 Forme initiale de la coque du tanker . . . 138
4.26 Deux déformations différentes de la coque vues de l'arrière du tanker . . . 139
4.27 Deux autres différentes déformations de la coque vues du dessous du tanker 140
4.28 Maillage déformé (en rouge, à gauche) ; maillage de référence (en noir, à droite) . . . 141
4.29 Modélisation d'un requin . . . 141
4.30 Maillage HEXPRESS™ autour du requin (Vue générale de côté) . . . 142
4.31 Maillage HEXPRESS™ autour du requin (Vue générale de dessus) . . . 142
4.32 Maillage HEXPRESS™ déformé autour du requin qui nage (Vue générale de dessus) . . . 143
4.33 Maillage HEXPRESS™ déformé autour du requin qui nage (Vue générale de côté) . . . 143
4.34 Maillage HEXPRESS™ déformé autour du requin qui nage (Vue générale de dessus) . . . 144
4.35 Maillage HEXPRESS™ déformé autour de la queue du requin . . . 145
4.36 Maillage HEXPRESS™ déformé autour de l'extrémité de la queue du requin 145
4.37 Maillage HEXPRESS™ déformé autour de la nageoire dorsale du requin . . . 146
4.38 Maillage HEXPRESS™ déformé autour d'une des nageoires pectorales du requin . . . 146
4.39 Maillage initial autour d'un losange comportant deux coins avec un angle très aigu . . . 147
4.40 Le maillage déformé autour du losange après une rotation de 90° . . . 148
4.41 Couplage de l'algorithme de remaillage par mouvement en bloc (actif dans la zone grisée) avec celui pseudo-structure afin de traiter la rotation du losange . . . 149

5.1 Exemple de maillages non-conformes . . . 152

5.2 Couplage à l'interface fluide-structure : découpage en faces virtuelles 154
5.3 Couplage à l'interface fluide-structure : transfert des efforts 155
5.4 Déplacements des nœuds du maillage fluide à partir de la position de la poutre déformée 155
5.5 Déplacements des nœuds du maillage fluide à partir de la position de la poutre déformée en 3D 156
5.6 Détails sur les déplacements des nœuds du maillage fluide à partir de la position de la poutre déformée en 3D 158
5.7 Algorithme de couplage en temps des solveurs fluide et structure 159

6.1 Plaque encastrée dans une conduite soumise à un écoulement transverse . . 164
6.2 Maillage fluide de la conduite et autour de la poutre 165
6.3 Comparaison du coefficient C_p pour le calcul stationnaire (isolignes rouges) et instationnaire (isolignes vertes) 166
6.4 Lignes de courant de l'écoulement stationnaire 166
6.5 Évolution temporelle de l'abscisse du point B : calcul statique et dynamique 167
6.6 Comparaison du coefficient C_p pour le calcul stationnaire sans IFS (isolignes rouges) et instationnaire avec IFS (isolignes bleues en pointillé) 168
6.7 Lignes de courant de l'écoulement calculé avec couplage IFS 169
6.8 Filament déformable fixé derrière un carré et plongé au sein d'un écoulement incompressible 170
6.9 Maillages fluides autour du filament 172
6.10 Étude des efforts F_y dans le cas du filament indéformable 174
6.11 Lâcher de tourbillons autour du filament indéformable 175
6.12 Déformation du maillage grossier 175
6.13 Évolution temporelle de l'extrémité libre du filament 176
6.14 Transformée de Fourier des efforts F_y 177
6.15 Résultats obtenus pour le filament déformable avec le maillage fin 178
6.16 Champ de pression autour du filament qui se déforme pendant une demi-période $\frac{T}{2} \approx 0.16\,s$ (calcul réalisé sur le maillage fin) 179
6.17 Résultats obtenus pour le filament déformable avec le schéma de Newmark amorti ($\alpha_{amort} = 0.035$) 181
6.18 Évolution temporelle de l'extrémité libre du filament en utilisant la méthode de Bossak 183
6.19 Comparaison des efforts F_y obtenus avec le schéma standard de Newmark (en rouge et en trait continu) et avec celui de Bossak (en noir et en trait-pointillé) 183
6.20 Câble fixé à une extrémité et plongé en partie dans l'air et en partie dans l'eau 184
6.21 Maillage HEXPRESS™ léger 186
6.22 Câble soumis à un courant (avec en bleu-vert la surface libre) 187
6.23 Étude des déplacements réduits ($\frac{A}{D}$) de l'extrémité libre du câble soumis à un courant 188
6.24 Maillage déformé autour du câble à l'instant $t = 15\,s$ 189
6.25 Comparaison des déplacements adimensionnels entre le calcul monoprocesseur (en rouge) et le calcul parallèle (en noir) 190

6.26 Câble tracté dans un milieu multifluide (sondages tous les 0.5 s) 192
6.27 Mouvement en bloc du domaine de calcul (domaine initial en noir ; domaine final en rouge) 192
6.28 Exemple de gestion du remaillage et des mouvements rigides 193
6.29 Comparaison des déplacements adimensionnels suivant $\overrightarrow{y}$ et $\overrightarrow{z}$ (en rouge) avec ceux obtenus dans la section §6.2.1 (en noir) 194
6.30 Présentation de l'expérience (images extraites de [22]) 196
6.31 Schéma du domaine de calcul 198
6.32 Maillage du domaine de calcul pour le cas-test de J.R. Chaplin et de son équipe 200
6.33 Comparaison de la déformée moyenne du riser dans la direction de l'écoulement (extrémités encastrées) (modèle Euler-Bernoulli en rouge ; modèle de Rayleigh en bleu ; mesures expérimentales en noir) 201
6.34 Maillage du domaine de calcul pour les liaisons rotule et double pivot . . . 203
6.35 Comparaison de la déformée moyenne du riser dans la direction de l'écoulement (liaisons double pivot) (modèle Euler-Bernoulli en rouge ; modèle de Rayleigh en bleu ; mesures expérimentales en noir) 204
6.36 Comparaison de la prédiction numérique avec l'expérience dans le cas de la modélisation avec des liaisons double pivot 205
6.37 Comparaison de la déformée moyenne du riser dans la direction de l'écoulement entre les différents codes numériques et les données expérimentales . 207
6.38 Comparaison des différents codes numériques testés sur le cas-test 1 de Chaplin pour l'enveloppe des déplacements du riser dans le plan $(\overrightarrow{x},\overrightarrow{z})$ . . 208
6.39 Comparaison des différents codes numériques testés sur le cas-test 1 de Chaplin pour l'enveloppe des déplacements du riser dans le plan $(\overrightarrow{y},\overrightarrow{z})$ . . 209

B.1 Tableau récapitulatif 252
B.2 1/4 de tour d'axe successif x, y et z par rapport à la base $\mathcal{B}_0$ 262
B.3 1/4 de tour d'axe successif x, y et z par rapport aux bases courantes . . . 262

Liste des tableaux

2.1 Reconstruction par le schéma GDS 37

3.1 Validation statique pour Euler-Bernoulli (unités SI) 72
3.2 Validation dynamique pour le modèle d'Euler-Bernoulli (unités SI) 74
3.3 Validation dynamique pour le modèle de Rayleigh (unités SI) 75
3.4 Conditions du cas-test proposé par D. Primault (unités SI) 79
3.5 Conditions du cas-test proposé par D. Primault (unités SI) 83
3.6 Conditions du cas-test proposé par D. Primault (unités SI) 83
3.7 Mouvements imposés aux extrémités (unités SI) 87
3.8 Comparaison du taux de torsion moyen obtenu sur ce cas-test avec celui de Le Tallec (en gras) (unités SI) 88
3.9 Conditions du cas-test proposé par D. Primault (unités SI) 92

6.1 Conditions du cas-test pour une plaque indéformable (unités SI) 164
6.2 Conditions du cas-test avec plaque déformable (unités SI) 167
6.3 Conditions du cas-test (unités SI) 171
6.4 Conditions du cas-test du câble soumis à un courant (unités SI) 185
6.5 Conditions du cas-test du câble tracté (unités SI) 191
6.6 Propriétés physiques du riser utilisé par J.R. Chaplin et son équipe (unités SI) 195
6.7 Liste des cas-tests réalisés par J.R. Chaplin et son équipe [21] 197

B.1 Table de multiplication "colonne × ligne" 250

Nomenclature

Les notations employées dans ce mémoire sont explicitées ci-dessous, avec leur unité. Lorsque des notations n'apparaissent pas sur cette page, elles sont expliquées dans le contexte du rapport.

Abréviations

CFD	Computational Fluid Dynamics
IFS	Interaction Fluide-Structure
ECN	École Centrale Nantes
LMF	Laboratoire de Mécanique des Fluides
EMN	Équipe Modélisation Numérique
i.e.	c'est-à-dire (id est)
cf.	voir (confer)
fig.	figure
tab.	tableau
VIV	Vortex Induced Vibrations
MPI	Message Passing Interface
DNS	Direct Numerical Simulation
DES	Detached Eddy Simulation
LES	Large Eddy Simulation
HPD	Hypothèse des Petites Déformations

Notation dyadique

$\overrightarrow{U} \cdot \overrightarrow{V}$	Produit scalaire de $\overrightarrow{U}$ par $\overrightarrow{V}$
$\overrightarrow{U} \wedge \overrightarrow{V}$	Produit vectoriel de $\overrightarrow{U}$ par $\overrightarrow{V}$
$\overrightarrow{U} \otimes \overrightarrow{V}$	Produit tensoriel de $\overrightarrow{U}$ par $\overrightarrow{V}$
$\overrightarrow{\nabla}.\overrightarrow{U}$ ou $div(\overrightarrow{U})$	Divergence du vecteur $\overrightarrow{U}$
$\overrightarrow{\nabla}.\overrightarrow{\overrightarrow{\tau}}$ ou $div(\overrightarrow{\overrightarrow{\tau}})$	Divergence du tenseur $\overrightarrow{\overrightarrow{\tau}}$
$\overrightarrow{\nabla} P$ ou $grad(P)$	Gradient du scalaire P
U^T ou U^t	Transposée de U

Solveur Fluide (chapitre 2)

x_j	Coordonnées spatiales	(m)
L	Longueur	(m)
$\overrightarrow{U}$	Vecteur vitesse	$(m.s^{-1})$
p	Pression statique	Pa ou $(kg.m^{-1}.s^{-2})$
ρ	Masse volumique du liquide	$(kg.m^{-3})$
Re	Nombre de Reynolds	
$\overrightarrow{g}$	Intensité de la pesanteur	$(m.s^{-2})$
$\overrightarrow{n}$	Vecteur normal	
μ	Viscosité dynamique	$(kg.m^{-1}.s^{-1})$
ν_t	Viscosité turbulente	$(kg.m^{-1}.s^{-1})$
k	Énergie cinétique turbulente	$(kg.m^{2}.s^{-2})$
ω_t	Fréquence turbulence	(s^{-1})
S_{ij}	Tenseur de déformations	
S	Module du tenseur de déformations	
ϵ	Taux de dissipation de l'énergie turbulente	
U_j	Composante moyenne de la vitesse dans la direction x_j	$(m.s^{-1})$
$u_j^{'}$	Composante fluctuante de la vitesse dans la direction x_j	$(m.s^{-1})$
Ω_{ij}	Tenseur de rotation	
$A \triangleq B$	A est défini comme étant égal à B	

Solveur Structure (chapitre 3)

X	Variable d'espace	(m)
r	Champ de position de la poutre	(m)
R	Champ de rotation de la poutre	
e_i	Base spatiale	
E_i	Base matérielle	
t_i	Base mobile	
t_1	Vecteur unitaire tangent à la ligne de référence de la poutre	
E	Module d'Young	$(kg.m^{-1}.s^{-2})$
A	Section de la poutre	(m^2)
I_a	Moment quadratique axial	(m^4)
I_p	Moment quadratique polaire	(m^4)
Θ	Angle de torsion total	(rad)
ϑ	Angle de torsion actuel	(rad)
θ	Angle de torsion itératif	(rad)

Ω_1	Champ de vitesse de torsion	$(rad.s{-}1)$
K_1	Champ de taux de torsion	
n	Force extérieure exercée sur la poutre	$(kg.m.s^{-2})$
m	Moment de la force extérieure exercée sur la poutre	$(kg.m^2.s^{-2})$
β, γ	Coefficients du schéma de Newmark	
Φ	Champ décrivant les positions et orientations de la poutre	
Υ	Vecteur des contraintes géométriques	
λ	Vecteur des multiplicateurs de Lagrange	
I_3	Matrice identité de taille 3×3	
$0_{m \times n}$	Matrice nulle de taille $m \times n$	
$Q^{'}$	Dérivée de la grandeur Q par rapport à la variable d'espace X	
$\dot{Q}$	Dérivée de la grandeur Q par rapport au temps	
$\hat{V}$ ou $V\hat{}$	Matrice antisymétrique du vecteur $\overrightarrow{V}$	
$\tilde{G}$ ou $G\tilde{}$	Vecteur discret issu du champ continu $\overrightarrow{G}$	

Module de Remaillage (chapitre 4)

$\overrightarrow{U}$	Vecteur déplacement	(m)
$\overrightarrow{\overrightarrow{\sigma}}$	Tenseur des contraintes	$(kg.m^{-1}.s^{-2})$
$\overrightarrow{\overrightarrow{\epsilon}}$	Tenseur des déformations	
E	Module d'Young	$(kg.m^{-1}.s^{-2})$
ν	Coefficient de Poisson	
G	Coefficient de cisaillement	$(kg.m^{-1}.s^{-1})$
μ	Coefficient de Lamé (premier coefficient de viscosité)	$(kg.m^{-1}.s^{-1})$
λ	Coefficient de Lamé (second coefficient de viscosité)	$(kg.m^{-1}.s^{-1})$

Introduction

La mécanique des interactions fluide-structure est née d'un besoin, d'une nécessité de compréhension des phénomènes complexes faisant intervenir plusieurs disciplines scientifiques. En mécanique des solides, les fluides sont considérés comme ayant une influence négligeable ou alors modélisés de manière très grossière, tandis qu'en mécanique des fluides, les solides sont vus uniquement comme des frontières d'analyse. L'expérience a prouvé que ces approximations étaient parfaitement valides dans de nombreux cas, mais elle a montré aussi que dans certaines situations cette vision ne suffit plus : les interactions entre solides et fluides doivent être prises en compte. Il est nécessaire d'utiliser *conjointement* la mécanique des solides et la mécanique des fluides pour obtenir une modélisation correcte de la réalité. On parle alors de couplage, car l'évolution de chacun des deux éléments dépend de celle de l'autre.

Cette rencontre de plusieurs domaines de la mécanique n'a été possible que grâce à l'essor technologique de ces dernières décennies. Il a permis à la mécanique des interactions fluide-structure d'intégrer un nouveau domaine scientifique, celui du calcul numérique. En effet, durant les quarante dernières années, les outils numériques et informatiques n'ont cessé de progresser. Avec la révolution qu'ont connue les ordinateurs dans la seconde moitié du $XX^{ème}$ siècle, les chercheurs peuvent maintenant avoir accès à des moyens de résolution et de visualisation de problèmes complexes. Le monde du calcul numérique connaît ainsi un développement sans précédent. Avec le calcul parallèle, les limites sont repoussées toujours plus loin. La recherche française vient en 2008, par exemple, de se doter d'un nouveau supercalculateur, hébergé à l'Institut du Développement des Ressources en Informatique Scientifique (IDRIS[2]). Il fournit une puissance totale de 207 Téraflops (composé de l'IBM Power6 d'une puissance de 68 Téraflops et du BlueGene/P, d'une puissance nominale de 139 Téraflops), ce qui représente 207000 milliards d'opérations à la seconde.

* * *

C'est dans ce contexte technologique que l'Équipe Modélisation Numérique (EMN) du Laboratoire de Mécanique des Fluides (LMF) de l'École Centrale Nantes (ECN[3]) développe depuis une décennie un code de calcul d'écoulements fluides, nommé **ISIS**. Il est ce qu'on appelle dans le langage du numéricien "un solveur fluide". Il est basé sur la résolution des équations de Navier-Stokes en moyenne de Reynolds (RANSE) sous une forme fortement conservative. Il repose sur une méthode de Volumes-Finis généralisée à des maillages

[2] http://www.idris.fr
[3] http://www.ec-nantes.fr

non-structurés composés de volumes polyédriques arbitraires. Bien que limités aux milieux incompressibles, les domaines d'applications sont diversifiés : en **hydrodynamique**, de nombreux calculs ont été réalisés autour de corps à géométries complexes en mouvement, comme des bateaux avec tous leurs appendices, afin d'améliorer leurs performances. Les écoulements peuvent être turbulents, cavitants, multifluides... Des optimisations de forme sont aussi possibles. Des tests ont été réalisés pour le **monde de l'automobile** (corps simplifié, comme celui de Willy). Depuis quelques années le laboratoire s'intéresse aussi au **biomimétisme**, en particulier à la nage des poissons (projet *ROBEA/RAAMO*). Évidemment cette liste est loin d'être exhaustive et ne cesse de croître. En effet, les industriels font de plus de plus appel à la modélisation numérique en complément des tests expérimentaux, car ceux-ci sont coûteux. Les besoins des ingénieurs devenant de plus en plus complexes, **ISIS** se doit de pouvoir prendre en compte de plus en plus de phénomènes physiques. Ainsi, divers modules peuvent être chargés par le programme, suivant les problèmes à résoudre.

Parmi les difficultés actuelles rencontrées par les hydrodynamiciens, on trouve le problème des risers : ce sont de longs tubes, situés sous les plates-formes off-shore, servant au transport de matières premières du fond de la mer jusqu'à la surface. Ils subissent d'énormes contraintes dues aux courants marins, ils se déforment et peuvent entrer en collision. Les dégâts occasionnés vont parfois jusqu'à la rupture du riser, entraînant pollution et coûts supplémentaires. Cela a poussé l'industrie pétrolière à s'intéresser au problème. C'est le sujet de nombreuses publications. L'EMN a alors décidé de doter **ISIS** d'un **module de résolution de déformations de structures élancées**. C'est dans ce cadre que cette thèse a vu le jour.

* * *

Mon travail de thèse constitue les premiers pas de l'équipe EMN dans le domaine de l'IFS. Le présent manuscrit se compose de cinq grandes parties.

Dans la première, la mécanique des interactions fluide-structure est présentée dans son ensemble (chapitre 1). Cette discipline touche de nombreux domaines et est complexe. Le passage au domaine numérique n'est donc pas trivial. Différentes voies sont possibles et l'état de l'art, fait dans ce mémoire, est loin d'être exhaustif. On distingue pourtant un trait commun entre les codes IFS : ils sont tous composés de quatre éléments. Ces derniers vont servir de base pour la suite de l'étude.

Le chapitre 2 décrit le "solveur fluide" constituant un des piliers du code IFS. On abordera ainsi les techniques de discrétisation et de résolution utilisées, ainsi que les différents modèles mis en œuvre dans **ISIS**.

En IFS, il y a le problème fluide mais aussi celui de la structure. Un "solveur structure" est donc nécessaire. Le chapitre 3 expose la théorie sur laquelle il s'appuie pour intégrer la gestion des déformations des corps élancés en grand déplacement. Une étape de validation a été effectuée en bidimensionnel et tridimensionnel, en petit et grand déplacement.

La quatrième partie regroupe dans les chapitres 4 et 5 la notion de couplage des deux précédents codes. L'échange d'informations entre l'écoulement et le corps est essentiel au niveau spatial et temporel. Le chapitre 4 est entièrement dédié au processus de remaillage, car il a occupé la moitié de ce travail de thèse.

Pour finir, des cas d'applications IFS seront présentés au chapitre 6. Une étude bidimensionnelle stationnaire en petit déplacement a été faite, ainsi qu'une, fortement instationnaire en grand déplacement. Viennent ensuite deux exemples montrant les possibilités du code en 3D et une comparaison tridimensionnelle du code développé avec un cas-test expérimental.

Chapitre 1

Interactions Fluide-Structure (IFS)

Le premier chapitre de la présente thèse a pour but de donner une vision générale de la discipline qu'est la mécanique des interactions fluide-structure. Une recherche non-exhaustive de publications sur le sujet a permis d'étudier les différentes méthodes numériques utilisées dans ce domaine.

1.1 Généralités sur une discipline très vaste

La mécanique des interactions fluide-structure, que nous appellerons par la suite l'interaction fluide-structure (IFS), est le fruit du couplage entre la mécanique des fluides et la mécanique des structures. Elle est donc par sa *nature très complexe*, et touche un *panel d'applications très variées*. C'est pourquoi les travaux réalisés dans cette branche sont très diversifiés et touchent un ensemble de domaines distincts (domaine environnemental, industrie automobile, industrie aéronautique, industrie maritime, industrie nucléaire, domaine de la recherche en biomécanique...). Chacune de ces études met en jeu une structure mobile, rigide ou déformable, et un fluide liquide ou gazeux qui s'écoule autour de la structure [84] [95] (cf. fig. 1.1). La grande variété des applications rencontrées rend un classement rigoureux difficile. On peut toutefois séparer les problèmes instationnaires des problèmes stationnaires (indépendants du temps). On peut aussi les distinguer par les équations régissant les sous-systèmes fluide ou structure. Quelques problèmes classiques d'interactions fluide-structure vont être présentés plus en détail en les classant suivant leur modèle fluide (fluide incompressible, peu compressible, compressible).

1.1.1 Problèmes d'IFS avec fluide incompressible

Tous les fluides sont compressibles. Cependant pour des nombres de Mach faibles l'hypothèse d'incompressibilité peut être faite. Les écoulements avec de l'eau comme fluide sont la plupart du temps incompressibles, comme par exemple l'eau autour d'une coque de bateau. En hydrodynamique l'écoulement est presque toujours incompressible. Dans le cas d'un écoulement avec de l'air il faut faire plus attention. En effet l'air nous entourant peut être considéré comme incompressible, mais dès que sa vitesse augmente ou qu'il est dans un espace confiné, l'hypothèse d'incompressibilité n'est plus valable.

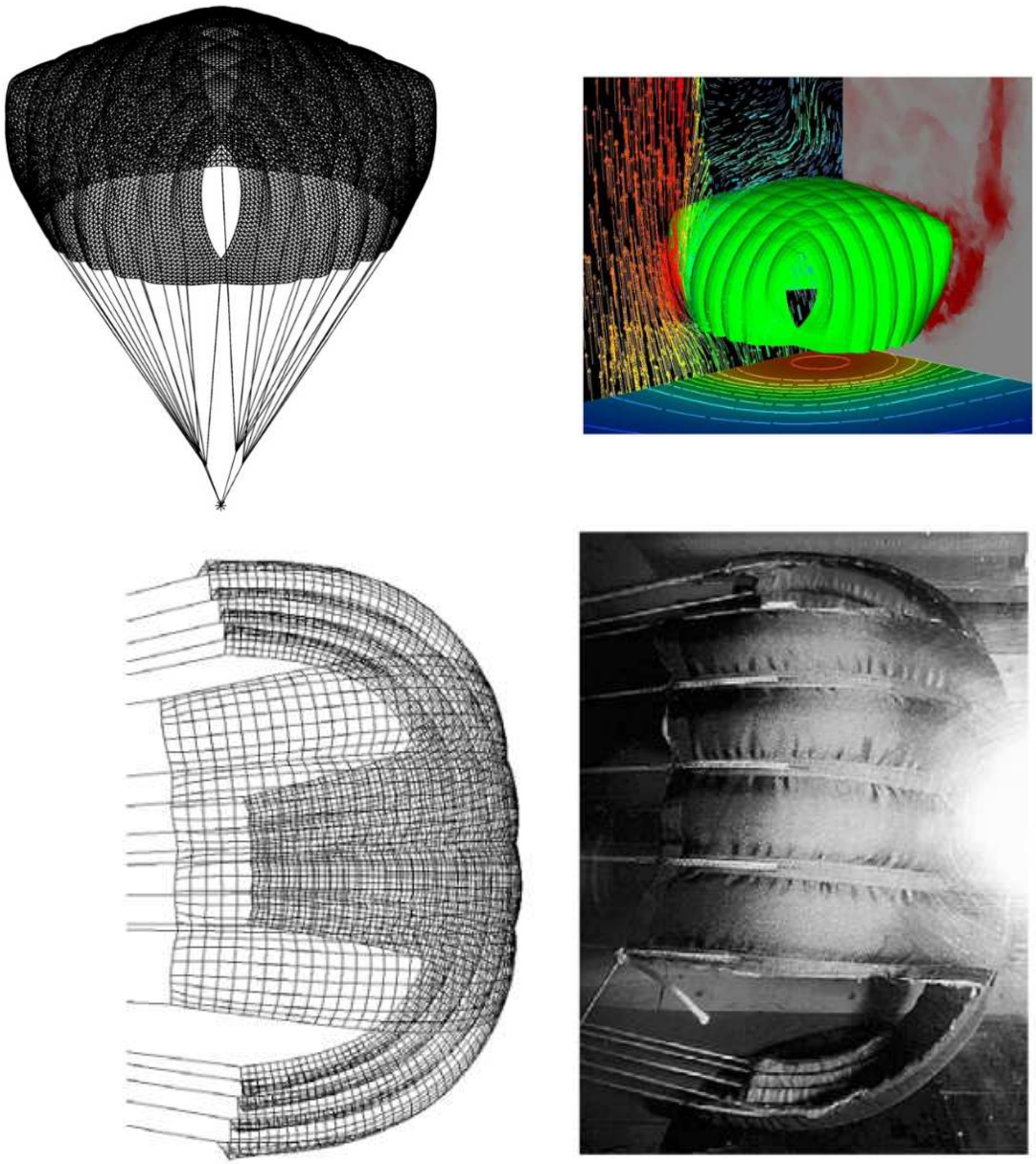


FIG. 1.1: Diverses études d'ouverture et d'écoulement d'interactions fluide-structure autour de parachutes

1.1.1.1 Génie civil

Le génie civil est un des premiers domaines où la mécanique des interactions fluide-structure a été prise en compte par exemple dans la construction de ponts. De nombreuses simulations expérimentales et numériques sont réalisées de nos jours [18] [41] (cf. fig. 1.2 et 1.3) pour éviter d'avoir à subir un nouvel effondrement comme celui du pont de Tacoma Narrows. Le 7 novembre 1940, le premier pont suspendu de Tacoma Narrows s'effondra à cause d'une instabilité aéroélastique de torsion (cf. fig. 1.4). Les vents n'étaient pourtant pas forts ($65\,km.h^{-1}$ environ), mais le pont a commencé à se balancer de plus en plus, tordant le tablier et les câbles. Certains câbles ont alors cédé, laissant tomber des parties entières de tablier.

FIG. 1.2: Test en soufflerie d'un modèle du pont de Tacoma de 1940

Les problèmes d'interactions fluide-structure n'interviennent pas que sur l'ouvrage d'art fini (ici le pont). Pendant les différentes étapes de la construction, l'interaction fluide-structure joue un rôle. Par conséquent les simulations ne portent pas seulement sur l'ouvrage d'art terminé, mais aussi sur les différentes phases de sa réalisation. Cette remarque est bien entendu valable dans d'autres domaines.

1.1.1.2 Industrie nucléaire

Dans ce domaine les problèmes liés à l'interaction fluide-structure peuvent venir de la structure de la centrale elle-même (on rejoint alors le domaine du génie civil) ou bien du processus de fission [82] [112].

Les centrales possèdent toutes de grands hyperboloïdes de révolution qui ont pour mission de refroidir la vapeur d'eau. Ces immenses tours génèrent des vortex sous l'action du vent. Si par malheur la direction du vent est défavorable et que sa vitesse entraîne des lâchers de tourbillons de manière périodique sur une autre tour, cela peut avoir des conséquences destructrices si cette dernière entre en résonance.

L'autre problème classique d'interactions fluide-structure dans le domaine nucléaire est celui des vibrations des tubes générateurs de vapeur [62] (cf. fig. 1.5). Ce faisceau de tubes sert à contrôler la vitesse de la réaction en chaîne et par voie de conséquence la puissance dégagée. Ces tubes sont peu déformables mais sont plongés dans un écoulement très perturbé. Il faut garder une amplitude faible des oscillations des tubes et bien entendu éviter qu'ils ne s'entrechoquent ce qui pourrait entraîner leur rupture.

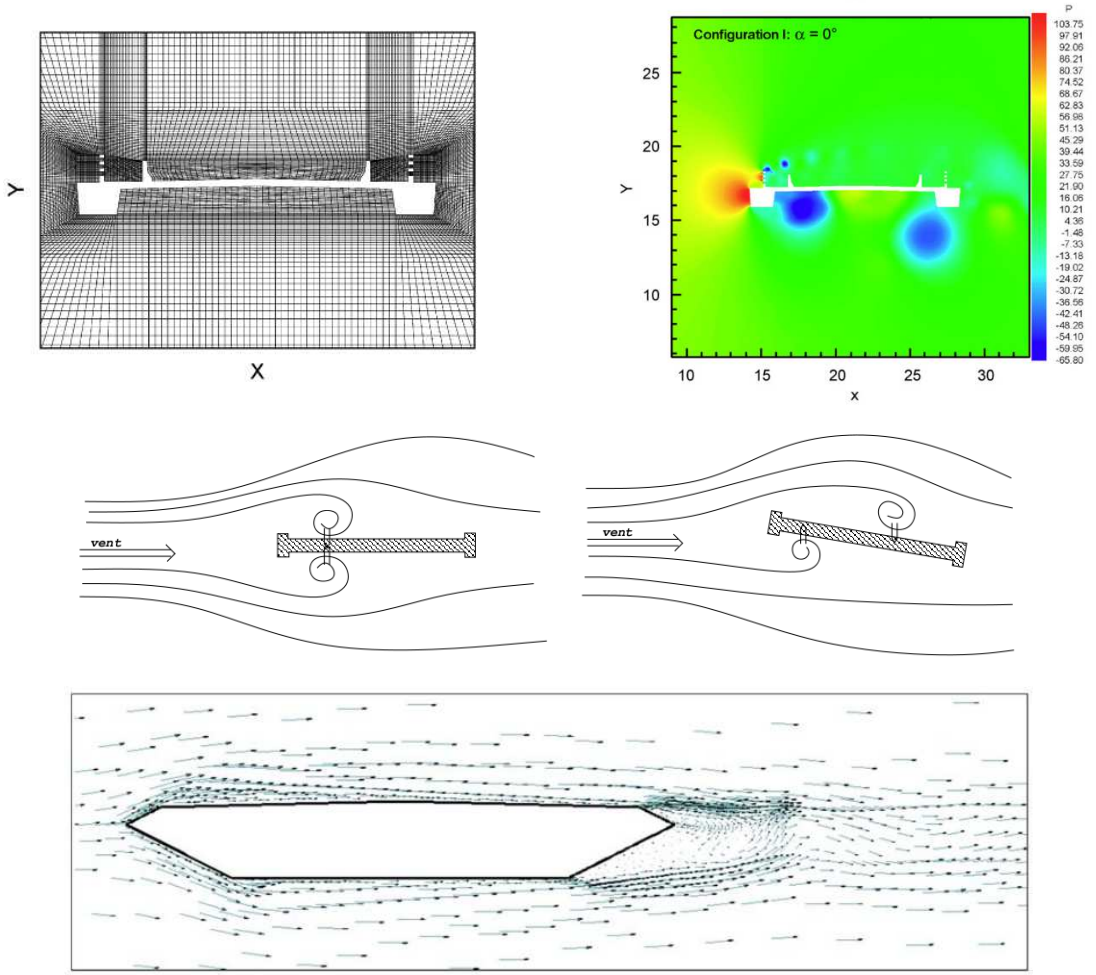


FIG. 1.3: Diverses études numériques de simulation d'écoulement autour de profils de pont

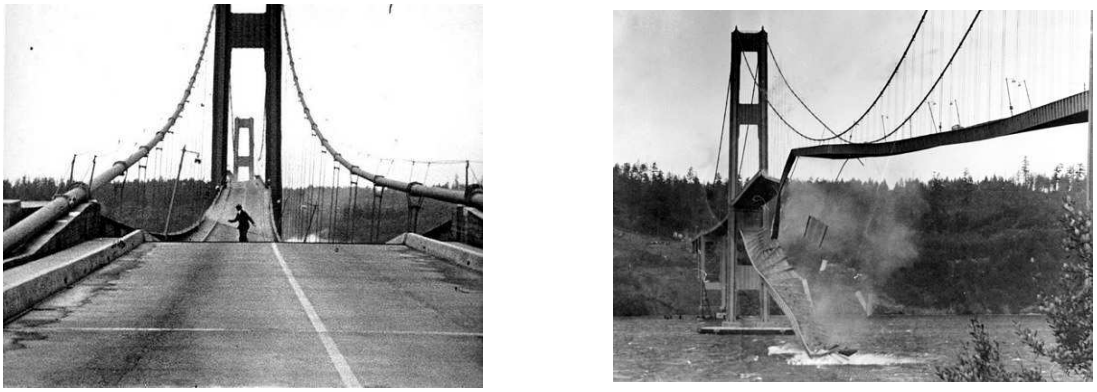


FIG. 1.4: Effondrement du pont de Tacoma sous l'action du vent en 1940

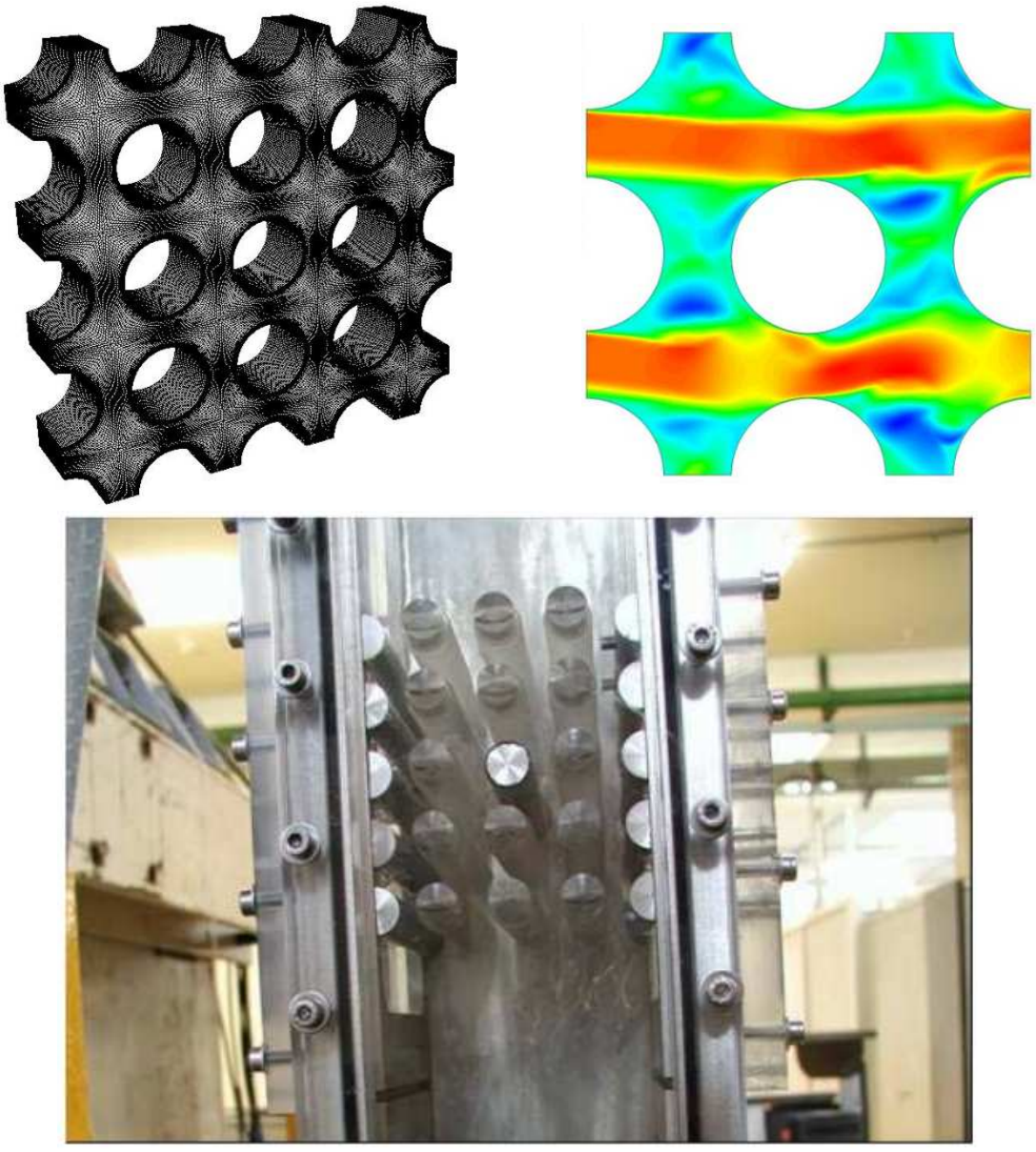


FIG. 1.5: Simulations numériques et expérimentales autour d'un faisceau de tubes rencontrés dans l'industrie nucléaire

1.1.1.3 Industrie maritime et navale

Dans cette catégorie, les exemples ne manquent pas. La houle générée par la mer pose aux hommes de nombreux cas d'interactions fluide-structure : dans les ports où cette houle doit être minimisée ou alors en mer sur les plates-formes off-shore où les infrastructures doivent résister aux impacts de la houle (cf. [69]). L'industrie pétrolière a fait naître de nombreux types d'installations off-shore : des barges flottantes maintenues par des câbles, des pipe-lines verticaux qui remontent le pétrole à la surface. Ces installations doivent résister à la houle qui est un phénomène de surface, mais aussi aux courants marins qui eux aussi ont une action à ne pas négliger (cf. fig. 1.6). Dans le cas de corps isolés, on parle de VIV (Vortex Induced Vibrations) : ce sont des instabilités induites par le détachement tourbillonnaire [73] [74].

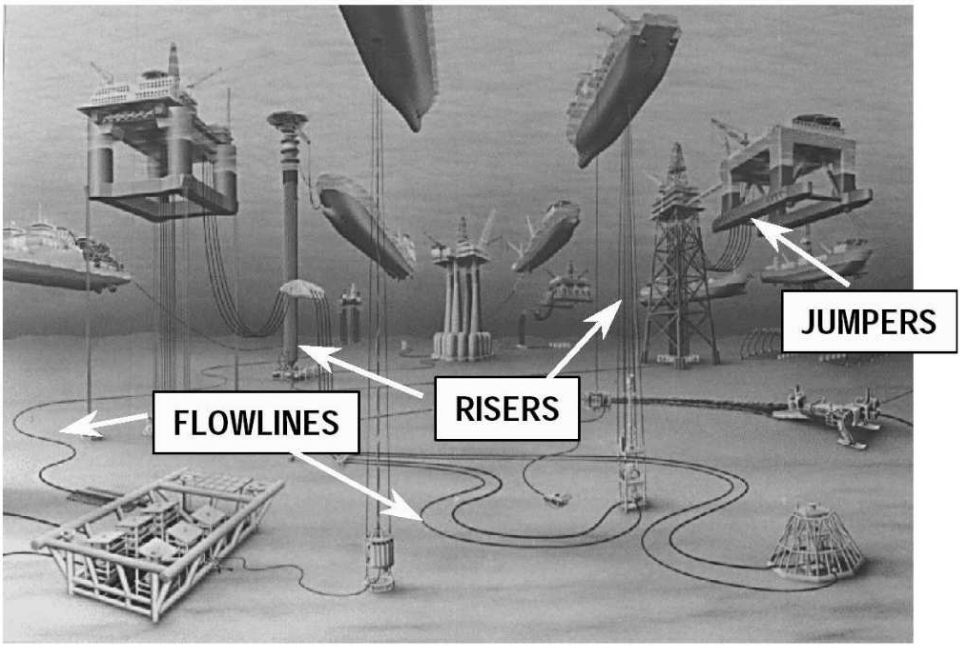


FIG. 1.6: Variétés des structures sous-marines et off-shore rencontrées dans le domaine de l'industrie maritime

En hydrodynamique navale, il faut concevoir puis optimiser les coques et appendices des navires [99]. Dans ce cadre, l'interaction fluide-structure peut paraître limitée car la carène d'un bateau peut apparaître comme rigide. Pourtant elle est bien là, les expériences en bassin avec des coques en giration le prouvent. La résistance à la houle et au déferlement sur les navires [44] est un défi pour les industriels où la mécanique des interactions fluide-structure occupe une place de premier plan.

La nage des poissons intéresse aussi beaucoup les scientifiques [39] [75]. En effet son rendement et les possibilités de virage (virage à 180° sans ralentir et rayon de courbure de l'ordre du dixième de leur longueur) et d'accélération (jusqu'à 20 fois la gravité) qu'elle apporte, sont exceptionnels. Ces données montrent le fossé qui existe entre les constructions humaines et la nature. Pour rappel une voiture doit ralentir de moitié avant de tourner et le rayon de courbure est de l'ordre de 10 fois sa longueur. Pour tenter de percer les mystères de la nature les scientifiques s'intéressent donc de près aux poissons. Citons par

exemple le programme français qui cherche à développer un robot anguille dans le cadre du projet national pluridisciplinaire anciennement ROBEA, et actuellement RAAMO[1] (projet ANR). De nombreux laboratoires travaillent de concert, tant sur le plan numérique qu'expérimental, pour mener à bien ce projet [58] [59] [15].

1.1.1.4 Domaine biomédical

Le génie biomédical s'occupe de la bonne intégration des prothèses ou d'éléments artificiels utilisés à des fins thérapeutiques avec les tissus biologiques. Dans de nombreux cas les fluides du corps humain comme le sang peuvent interagir avec ces tissus [49] [61] (cf. fig. 1.7). Ces derniers étant déformables, on a affaire à des cas naturels complexes d'interactions fluide-structure : formation de glaucome, rupture d'anévrisme... Bien évidemment les prothèses en engendrent elles-aussi.

Ces problèmes comptent parmi les plus difficiles en interactions fluide-structure. Rien que la modélisation de l'espace est un défi (cf. fig. 1.8).

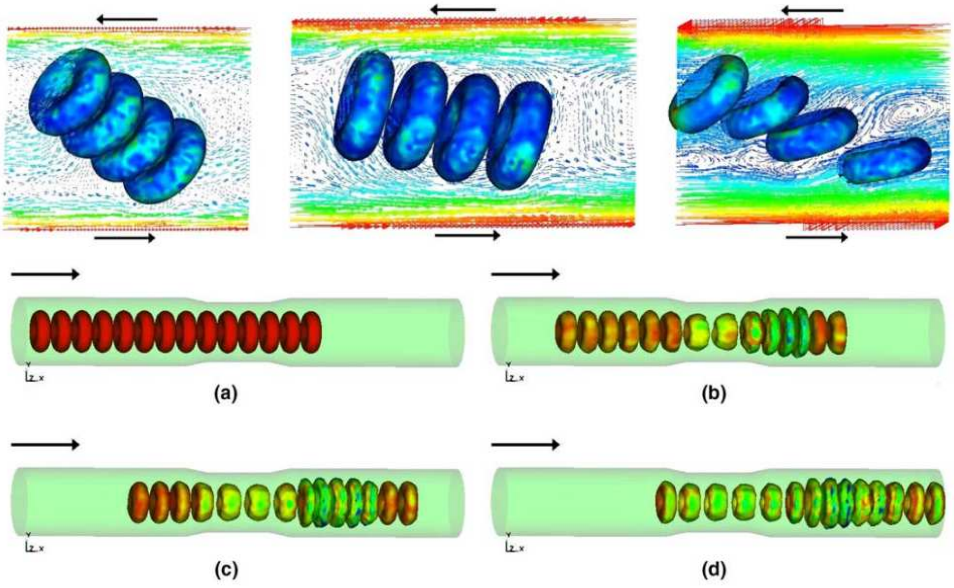


Fig. 1.7: Simulations numériques d'écoulements de sang

1.1.1.5 Génie industriel

Dans ce cadre deux grandes familles d'étude peuvent être distinguées : les études sur les réservoirs [60] et les études sur des structures très fines et très élancées (câbles ou encore tubes (risers)...) [94] (cf. fig. 1.6).

Dans le cas des réservoirs, les scientifiques cherchent à prévoir le comportement du fluide lors des déplacements de la structure. On retrouve ce problème dans les tankers [111], les camions-citernes [5] (cf. fig. 1.9) ou encore les lanceurs de fusées [13] (cf. fig. 1.10).

La problématique des interactions fluide-structure autour de corps élancés sera détaillée par la suite, car c'est l'objet de cette thèse.

[1] http://raamo.irccyn.ec-nantes.fr

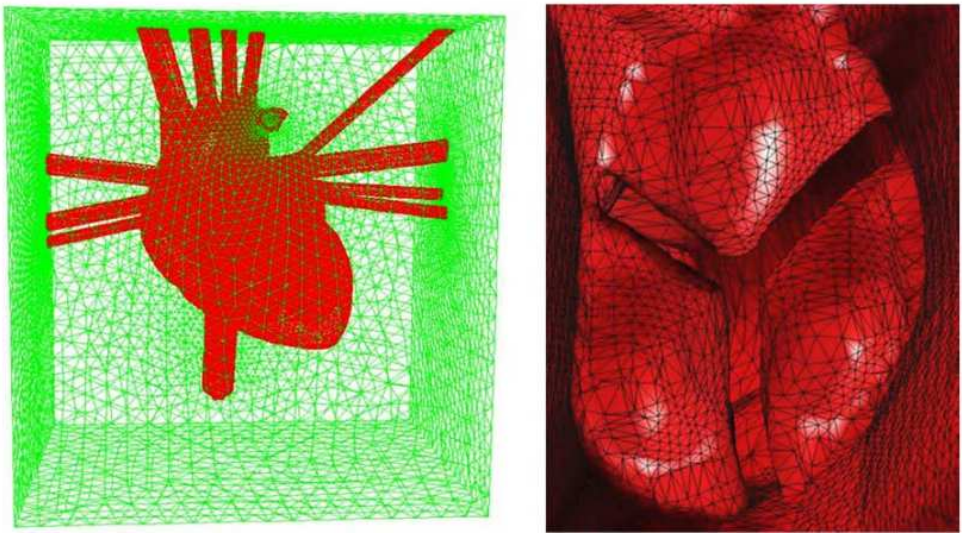


FIG. 1.8: Modélisation complexe d'un cœur immergé dans un fluide et de l'aorte

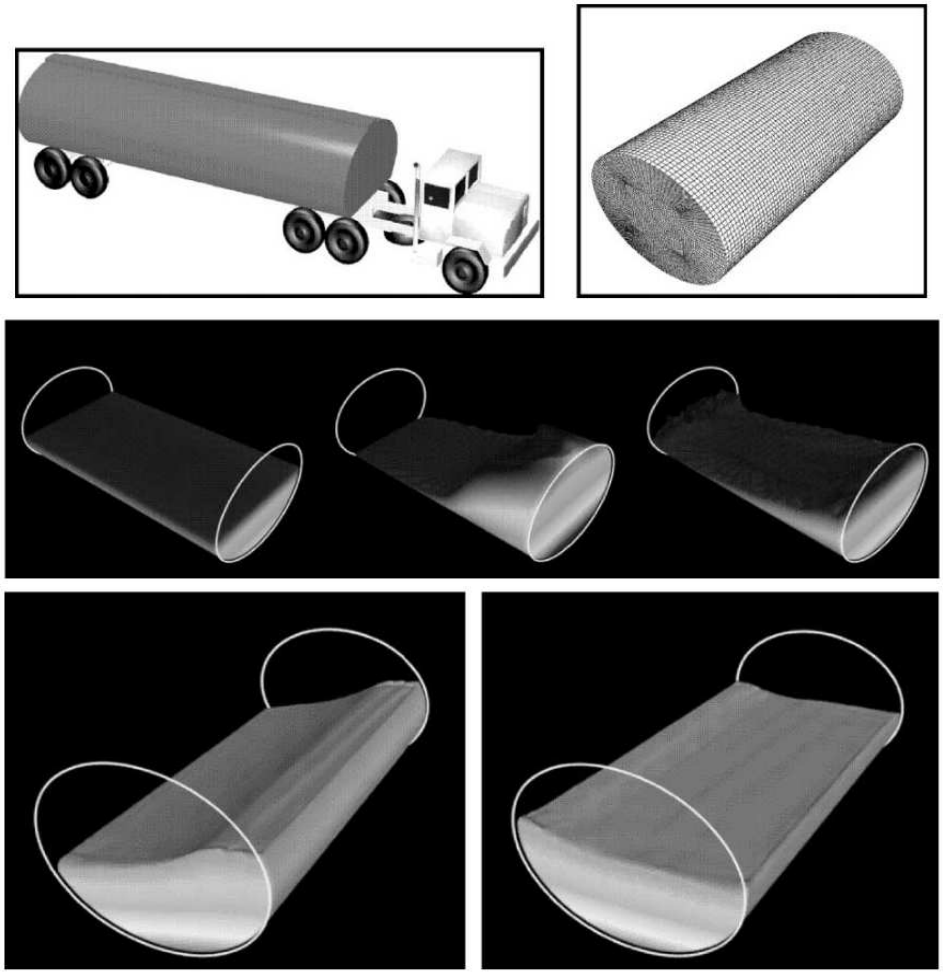


FIG. 1.9: Modélisation et simulation numérique du phénomène de sloshing dans un camion-citerne

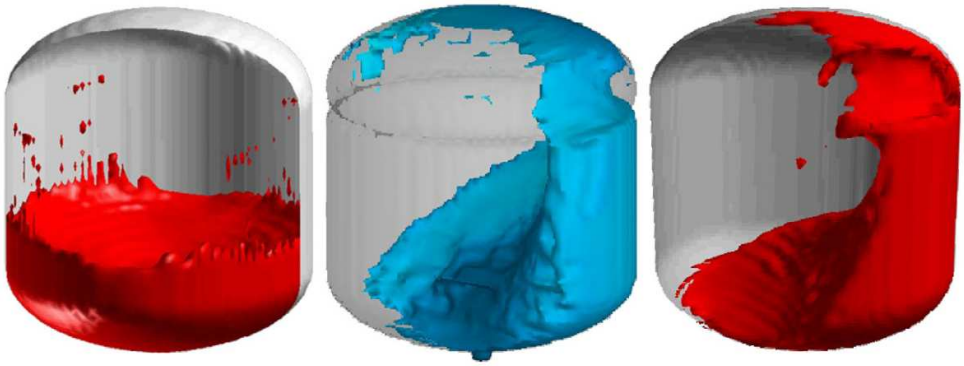


FIG. 1.10: Simulation numérique du phénomène de sloshing dans des cuves de l'industrie aérospatiale

1.1.2 Problèmes d'IFS avec fluide peu compressible

Le nombre d'exemples dans cette catégorie est plus restreint. On peut citer l'impact du vent sur des véhicules terrestres se déplaçant à grande vitesse. Les problèmes liés au croisement des trains, à grande vitesse ou dans un tunnel (milieu confiné), se classent aussi dans cette catégorie.

1.1.3 Problèmes d'IFS avec fluide compressible

Les écoulements compressibles sont généralement rencontrés pour des nombres de Mach assez grands (hors confinement). Dans ce cas là, les problèmes d'interactions fluide-structure sont donc majoritairement issus du domaine aéronautique et aérospatial : calcul autour d'un F-16 [37] (cf. fig. 1.11) ou encore simulations des vibrations des ailes [42] ou des commandes.

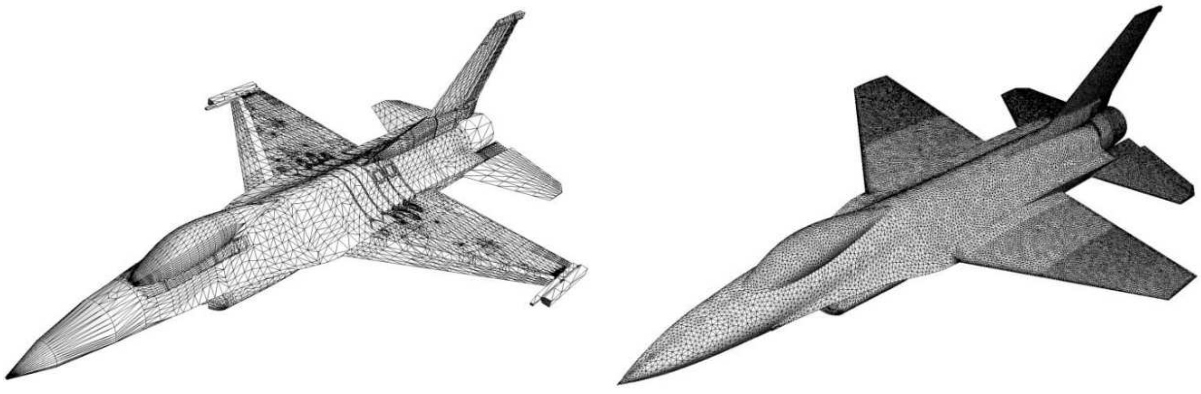


FIG. 1.11: Modélisation et simulation numérique d'un F-16 (maillage de la structure à l'interface et maillage fluide à l'interface)

De nombreux autres domaines d'applications existent : des études sont réalisées sur des machines tournantes [71] [85] (pales d'hélicoptères, turbines et turbocompresseurs). Les explosions appartiennent aussi au domaine compressible. Des recherches sur les impacts

dus à ces déflagrations sur des structures sont en cours (pliage et ouverture d'airbag [25] (cf. fig. 1.12), impact d'explosion aérienne sur des véhicules terrestres).

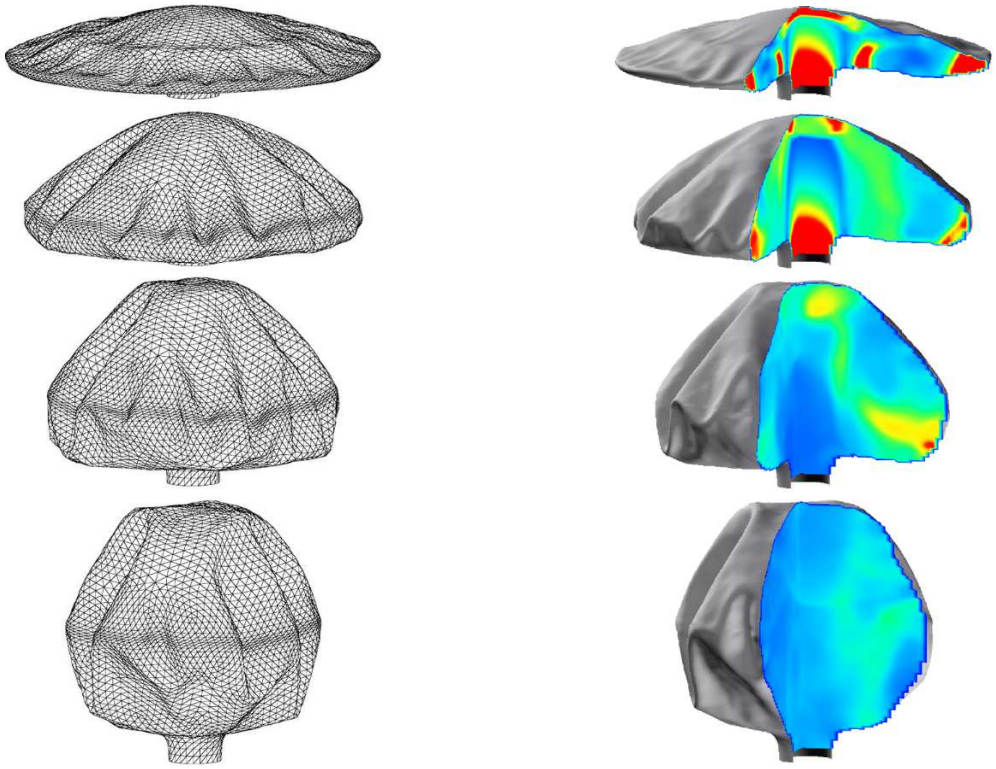


FIG. 1.12: Modélisation et simulation numérique de l'ouverture d'un airbag

1.1.4 Problèmes d'IFS et ondes acoustiques

Ce dernier domaine est un peu à part : dans cette situation le fluide et la structure ne subissent que de faibles perturbations. Dans cette catégorie on trouve par exemple les problématiques environnementales : réduire les signatures acoustiques des voitures et avions. D'autres exercices plus ludiques peuvent être cités comme les études sur les instruments de musique (acoustique d'une guitare [35] [10] (cf. fig. 1.13)).

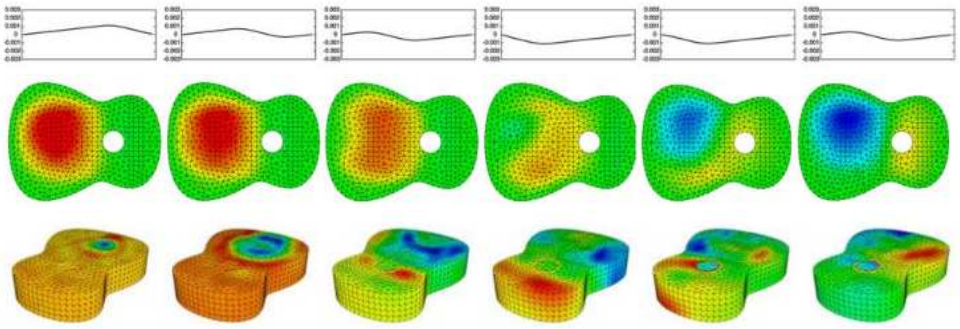


FIG. 1.13: Modélisation et simulation numérique de l'acoustique d'une guitare en fonction du pincement initial de la corde

1.2 IFS d'un point de vue numérique

Tous les exemples précédents nous ont montré le succès rencontré par cette discipline. Celui-ci est dû en partie au développement de la simulation numérique en général et des performances des calculateurs. L'apparition du calcul parallèle dans les années 1970 a rendu possible le couplage de différents codes. L'engouement du domaine scientifique pour cette discipline est aussi dû au fait que la simulation expérimentale coûte très cher. Les exemples donnés précédemment sont très complexes. Les expériences en sont d'autant plus coûteuses. La simulation numérique ne remplace évidemment pas l'expérimentation mais permet souvent de réduire les cas-tests expérimentaux et donc le coût total. L'interaction fluide-structure d'un point de vue numérique a donc connu un fort essor dû à la pression des industriels ces dernières années.

1.2.1 Stratégies et algorithmes de couplage

En présence d'un cas d'interactions fluide-structure un fluide s'écoule autour d'un solide mobile, rigide ou déformable. Il va donc falloir simuler numériquement l'évolution du fluide et de la structure. Le développement et/ou l'utilisation d'un code numérique est nécessaire. De nombreuses stratégies sont possibles pour ces développements. Le choix de la meilleure stratégie dépend de l'existence ou non de codes fluide et/ou structure mais aussi du problème à traiter. Un problème acoustique sera plus simple à traiter car les formulations sont simplifiées. Dans la suite de cette partie les modèles couplés ne seront pas considérés comme des modèles simplifiés.

Le couplage fluide-structure peut être réalisé de deux manières différentes :

- ➢ **Monolithique :** les équations discrétisées pour le fluide et la structure sont résolues de façon simultanée, à l'intérieur d'un même système.
- ➢ **Découplée :** le fluide et la structure sont résolus séquentiellement et avancés dans le temps, éventuellement par différentes méthodes et par différents codes.

À ses débuts les numériciens de l'interaction fluide-structure avaient une préférence pour le couplage monolithique, car celui-ci est *très fort* et assure donc une *meilleure convergence* [50] [103]. Mais de par sa nature il est *très lourd et très difficile à réaliser*, voir totalement impossible pour des problèmes complexes. Le couplage monolithique demande des développements spécifiques qui rendent difficile la construction d'un code fluide-structure généralisé.

La seconde approche est *plus souple*, puisque les deux systèmes fluide et structure sont résolus de manière plus ou moins couplés. *La fréquence des échanges entre le fluide et la structure détermine le degré de force du couplage.* Ce couplage permet l'utilisation de codes déjà existants et la possibilité d'utiliser aussi les dernières versions de ces programmes. Le code fluide-structure a ainsi la capacité d'évoluer au fil des mises à jour de ses sous-programmes et de s'adapter à de nouveaux problèmes. Les échanges de variables sont faits directement entre les deux codes ou alors par l'intermédiaire d'un coupleur prévu à cet effet. Ces communications correspondent aux forces d'action, de réaction de la physique et aux déplacements structurels.

Quatre éléments sont essentiels pour réaliser une simulation numérique d'interaction fluide-structure :

- ➢ Un **solveur de dynamique des fluides**
- ➢ Un **solveur de dynamique des structures**
- ➢ Une **interface de couplage en espace**
- ➢ Une **interface de couplage en temps**

Le **solveur fluide** permet d'*avancer en temps le fluide* : il résout les équations aux dérivées partielles du modèle choisi pour le fluide. Il doit pouvoir lire les informations représentant les déplacements de ses frontières et transmettre les forces exercées sur l'interface fluide-structure dans un format précis. Dans cette thèse le solveur fluide est **ISIS** qui sera présenté en détail au chapitre 2.

Le **solveur structure** permet, de même que le solveur fluide, d'*avancer en temps pour la structure*. Il se base sur les équations aux dérivées partielles du modèle choisi pour la structure. Il doit être capable de lire les informations relatives aux forces physiques générées par le fluide sur l'interface fluide-structure et d'en déduire les déplacements correspondants de la structure. Dans cette thèse le solveur structure est un solveur poutre qui doit être codé.

Chacun de ces flux d'informations est obligatoirement dans un format prédéfini. Ainsi l'**interface de couplage en espace** peut faire son travail. En effet cette interface sert à *traduire les informations transmises par le solveur-serveur dans le format compréhensible par le solveur-client*. Cette partie est loin d'être triviale : les maillages fluide et structure sont en effet la plupart du temps différents à l'interface fluide-structure, ce qui implique l'utilisation d'interpolations pour la transmission des informations [28] [38]. Cette tâche est indépendante du schéma en temps de chaque code. L'interface de couplage en espace inclut aussi le *remaillage fluide autour de la structure déformée*. Ce remaillage peut se baser sur différentes méthodes. Dans cette thèse, une **méthode ALE** (Arbitrary Lagrangian Eulerian) sera utilisée. Quelques mots sur les techniques ALE : elles se fondent comme leur nom l'indique sur une formulation mixte. On évite ainsi les problèmes dus aux approches purement lagrangiennes (description cinématique du fluide très difficile quand l'écoulement est complexe et que le temps de simulation est long) ou purement eulérienne (problème de précision dans la description du fluide le long de l'interface fluide-structure). Dans la méthode ALE l'observateur est lié à un maillage mobile imaginaire. La vitesse de ce dernier n'est ni nulle ni égale à la vitesse du fluide. Elle varie continûment de l'un à l'autre. Les méthodes ALE classiques demandent de réécrire les équations du fluide avec les coordonnées hybrides. Dans notre cas où le solveur fluide est déjà écrit, elles ne sont pas optimales et donc mises de côté. La méthode de remaillage utilisée lors de cette thèse est récente et encore peu développée dans la littérature : on considère le fluide comme une *structure pseudo-solide* et ainsi on peut utiliser les équations de la structure pour déformer le maillage et pour en contrôler la déformation.

Enfin, le dernier élément est l'**algorithme de couplage en temps**. Ce dernier organise l'*échange des informations dans le temps et rend effectif le fonctionnement "parallèle" des deux codes fluide et structure*. Dans la réalité ces échanges sont exacts à tout moment. Ils vérifient le principe d'action et de réaction. Ce ne sera malheureusement pas le cas au sein d'un code fluide-structure à cause de la discrétisation en temps et de l'emploi de

solveurs séparés pour résoudre le fluide et la structure. En fait un algorithme de couplage en temps est d'autant meilleur qu'il vérifie le moins mal le principe d'action et de réaction [78] [77].

Lors d'une simulation d'interaction fluide-structure on note rapidement que la précision et la stabilité en temps du couplage global sont souvent bien moins satisfaisantes que celles de chaque solveur. Le couplage global en temps est donc l'élément déterminant la justesse et la robustesse du programme d'interaction fluide-structure. Parmi ces algorithmes de couplage en temps plusieurs catégories existent :

- ➢ les **algorithmes explicites synchrones** et **décalés** : chaque sous-système (structure et fluide) est *mis à jour en temps de manière successive*. En jouant sur les informations échangées l'énergie artificiellement introduite par l'algorithme de couplage en temps est minimisée le moins mal possible.
- ➢ les **algorithmes parallèles** : les deux sous-systèmes sont *mis à jour en temps au même moment*. Le principe d'action et de réaction est au mieux conservé par le choix des données à échanger.
- ➢ les **algorithmes itérés** (aussi nommés **implicites**) : cette dernière catégorie est un peu *différente*, car pour un pas de temps il peut y avoir plusieurs calculs, plusieurs itérations (d'où son nom). Chacune de ces itérations correspond à un des algorithmes ci-dessus.

1.2.1.1 Algorithme de couplage en temps explicite synchrone et décalé

Voici un exemple d'algorithme décalé pour passer d'un temps t^n à $t^{n+1} = t^n + \Delta t$ (cf. fig. 1.14) :

- 1 : prédiction de la structure à t^{n+1} : $\tilde{X}^{n+1} = X^n + \frac{\Delta t}{2}(3V^n - V^{n-1})$ (où V^n est la vitesse de la structure à l'instant t^n)
- 2 : interprétation de ces mouvements de la structure dans le langage du solveur fluide
- 3 : remaillage fluide pour ces nouvelles frontières
- 4 : calcul fluide : on avance en temps le fluide
- 5 : calcul des forces F_f^{n+1} exercées par le fluide sur la structure à l'instant t^{n+1}
- 6 : interprétation de ces forces dans la formulation du code structure : F_s^{n+1}
- 7 : calcul structure : on avance en temps la structure en sous-relaxant par exemple $F^{n+1} = 2F_s^{n+1} - F^n$

Avec cet algorithme il n'est pas nécessaire de connaître à l'avance (à t fixé) des informations concernant l'autre sous-système. Toutes les données seront connues à la fin du pas de temps considéré : on a donc bien à faire à un algorithme *non monolithique*. On note que cet algorithme est dans la littérature encore divisé en deux sous-classes. En effet la prédiction de la structure peut se faire à t^{n+1} (*schéma explicite synchrone*) ou à $t^{n+\frac{1}{2}}$ (*schéma décalé*).

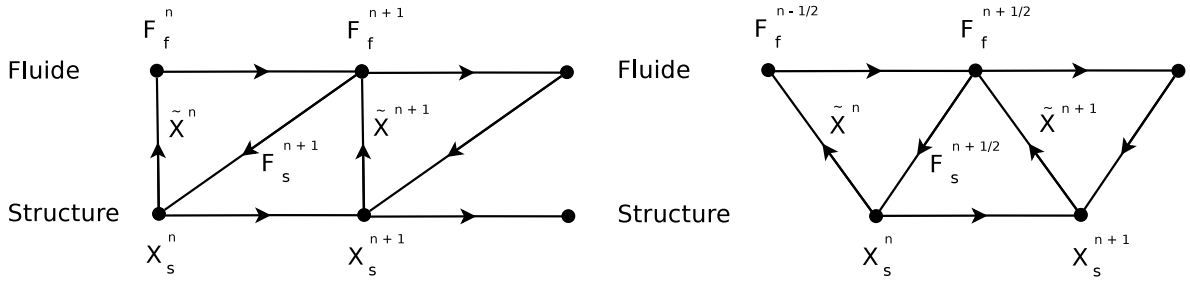


FIG. 1.14: Schémas de couplage en temps explicite

1.2.1.2 Algorithme de couplage en temps parallèle

Voici un exemple d'algorithme parallèle pour passer d'un temps t^n à $t^{n+1} = t^n + \Delta t$ (cf. fig. 1.15) :

- F 1 : prédiction de la structure à t^{n+1} : $\tilde{X}^{n+1} = X^n + \frac{\Delta t}{2}(3V^n - V^{n-1})$
- F 2 : interprétation de ces mouvements de la structure dans le langage du solveur fluide
- F 3 : remaillage fluide pour ces nouvelles frontières
- F 4 : calcul fluide : on avance en temps le fluide
- S 1 : calcul des forces F_f^n exercées par le fluide sur la structure à l'instant t^n
- S 2 : interprétation de ces forces dans la formulation du code structure : F_s^n
- S 3 : calcul structure : on avance en temps la structure en sous-relaxant par exemple $F^{n+1} = 3F_s^n - F_s^{n-1} - F^n$

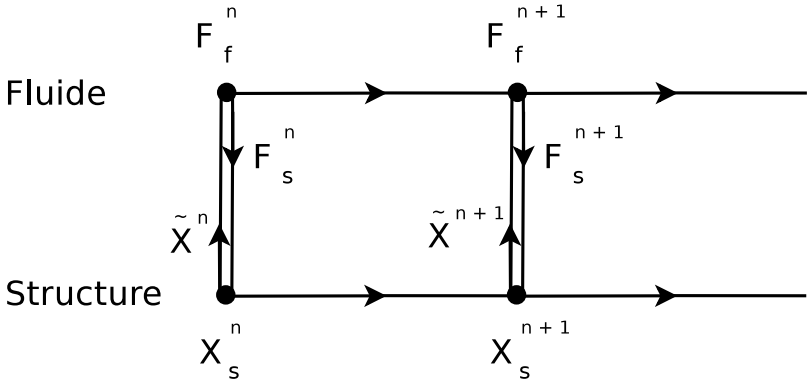


FIG. 1.15: Schéma de couplage en temps parallèle

Cet algorithme *n'est pas monolithique* car les tâches F1-F4 et S1-S3 sont effectuées en parallèle.

1.2.1.3 Algorithme de couplage en temps itéré

Voici un exemple d'algorithme itéré pour passer d'un temps t^n à $t^{n+1} = t^n + \Delta t$ (cf. fig. 1.16) :

- 1 : initialisations : $k = 0$ et $\tilde{X}^{(0)} = X^n$
- 2 : interprétation des mouvements de la structure $\tilde{X}^{(k)}$ dans le langage du solveur fluide
- 3 : remaillage fluide pour ces nouvelles frontières
- 4 : calcul fluide : on avance en temps le fluide
- 5 : calcul des forces $F_f^{n+1(k)}$ exercées par le fluide sur la structure à l'instant t^{n+1}
- 6 : interprétation de ces forces dans la formulation du code structure : $F_s^{n+1(k)}$
- 7 : calcul structure : on avance en temps la structure
- 8 : test de convergence : si l'algorithme a convergé à l'instant considéré on passe au pas de temps suivant, sinon on recommence à l'étape 1 avec $k = k + 1$ et $\tilde{X}^{(k+1)} = X^{n+1(k)}$.

Cet algorithme est *non monolithique*. Il est aussi stable si la convergence est obtenue à chaque pas de temps. Par contre on se rend évidemment compte qu'il est beaucoup plus coûteux que les deux autres. Par souci de pouvoir traiter le plus grand ensemble possible de problèmes un algorithme de ce type sera utilisé dans le couplage fluide-structure présenté dans cette thèse.

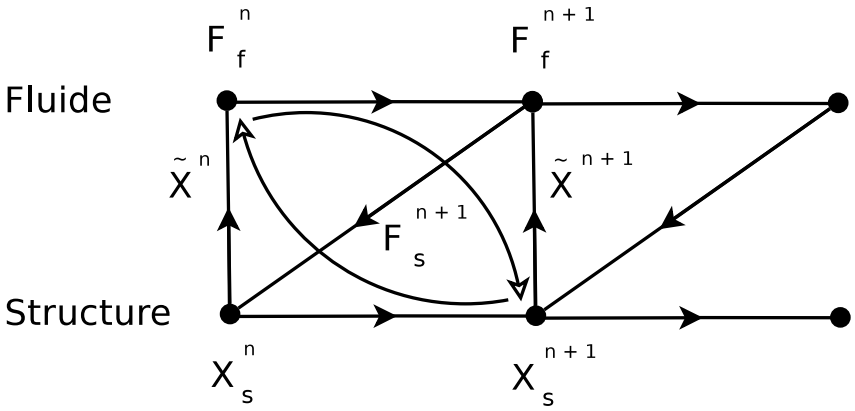


FIG. 1.16: Schéma de couplage en temps itéré ou implicite

1.2.1.4 Algorithmes appliqués à un problème simple

Pour que les différents couplages soient bien clairs ils vont être explicitement écrits sur un problème simple : Considérons *un système composé de deux masses m_x et m_y reliées par un ressort de raideur k*. les abscisses des deux masses, respectivement x et y vérifient les équations suivantes :

$$\begin{cases} m_x\ddot{x} + kx = ky \\ m_y\ddot{y} + ky = kx \end{cases} \tag{1.1}$$

avec pour conditions aux limites :

$$\begin{cases} \dot{x}(0) = 0,\, x(0) = x_0 \\ \dot{y}(0) = 0,\, y(0) = y_0 \end{cases} \tag{1.2}$$

Ce système est isolé et a une énergie qui est conservée au cours du temps. Par ailleurs la solution exacte de ce système est connue :

$$\begin{cases} x(t) = \dfrac{1}{m_x + m_y}(m_x x_0 + m_y y_0 + m_y(x_0 - y_0\cos(\omega t))) \\ y(t) = \dfrac{1}{m_x + m_y}(m_x x_0 + m_y y_0 - m_x(x_0 - y_0\cos(\omega t))) \end{cases} \tag{1.3}$$

avec $\omega^2 = k(\frac{1}{m_x} + \frac{1}{m_y})$.

On utilise *la règle du trapèze* pour construire nos algorithmes de couplage en temps. Pour une simple équation du type $m\ddot{x} + kx = 0$ la règle du trapèze s'écrit :

$$\begin{cases} m\,\dfrac{\dot{x}^{n+1} - \dot{x}^n}{\Delta t} + k\,\dfrac{x^{n+1} + x^n}{2} = 0 \\ x^{n+1} = x^n + \Delta t\dfrac{\dot{x}^{n+1} + \dot{x}^n}{2} \end{cases} \tag{1.4}$$

où $\dot{x}^n$ est une approximation de la vitesse à l'instant t^n et Δt le pas de temps. Ce schéma est d'ordre deux en temps, inconditionnellement stable et conserve l'énergie totale du système.

Revenons à notre système d'équations couplées. On propose les algorithmes suivants :

$$\begin{cases} m_x\,\dfrac{\dot{x}^{n+1} - \dot{x}^n}{\Delta t} + k\,\dfrac{x^{n+1} + x^n}{2} = k\,y^{\clubsuit} \\ x^{n+1} = x^n + \Delta t\dfrac{\dot{x}^{n+1} + \dot{x}^n}{2} \\ m_y\,\dfrac{\dot{y}^{n+1} - \dot{y}^n}{\Delta t} + k\,\dfrac{y^{n+1} + y^n}{2} = k\,x^{\spadesuit} \\ y^{n+1} = y^n + \Delta t\dfrac{\dot{y}^{n+1} + \dot{y}^n}{2} \end{cases} \tag{1.5}$$

où $y^{\clubsuit}$ et $x^{\spadesuit}$ sont à définir suivant les différents algorithmes de couplage.

Schéma de couplage monolithique caché : $y^{\clubsuit} = \frac{y^{n+1}+y^n}{2}$ et $x^{\spadesuit} = \frac{x^{n+1}+x^n}{2}$

Ce schéma est d'ordre deux et inconditionnellement stable. Il est même équivalent à la règle du trapèze pour le système global.

Schéma de couplage décalé d'ordre 1 : $y^{\clubsuit} = y^n$ et $x^{\spadesuit} = \frac{x^{n+1}+x^n}{2}$

Ce schéma est inconditionnellement instable et seulement d'ordre 1.

Schéma de couplage parallèle d'ordre 1 : $y^{\clubsuit} = y^n$ et $x^{\spadesuit} = x^n$

Ce schéma est inconditionnellement instable et d'ordre 1.

Schéma de couplage décalé d'ordre 2 : $y^{\clubsuit} = y^n + \frac{\Delta t}{2}\dot{y}^n$ et $x^{\spadesuit} = \frac{x^{n+1}+x^n}{2}$

Ce schéma est inconditionnellement stable et d'ordre 2.

Schéma de couplage parallèle d'ordre 2 : $y^{\clubsuit} = y^n + \frac{\Delta t}{2}\dot{y}^n$ et $x^{\spadesuit} = x^n + \frac{\Delta t}{2}\dot{x}^n$

Ce schéma est inconditionnellement stable et d'ordre 2.

Schéma de couplage itéré :

$$\begin{cases} y^{\clubsuit^1} = y^n + \dfrac{\Delta t}{2}\dot{y}^n,\ x^{\spadesuit^1} = x^n + \dfrac{\Delta t}{2}\dot{x}^n \\ y^{\clubsuit^{(k+1)}} = \dfrac{y^{n+1(k)} + y^n}{2},\ x^{\spadesuit^{(k+1)}} = \dfrac{x^{n+1(k)} + x^n}{2} \end{cases} \tag{1.6}$$

Cet algorithme s'il converge est équivalent à l'algorithme monolithique.

Notons que la convergence ou l'ordre des schémas dépend des problèmes traités.

1.2.2 Formulations et discrétisations

L'autre point important de l'interaction fluide-structure numérique est le type de discrétisations utilisé. Les méthodes d'**éléments finis** et de **volumes finis** se disputent évidemment la première place. La méthode des différences finies est trop simpliste pour des codes de calcul fluide-structure généralistes.

La communauté de simulation des structures possède une prédilection pour les éléments finis en raison de son *universalité et d'un ordre de précision modulable à volonté*. Cependant des travaux comme ceux de I. Demirdzic et S. Muzaferija (cf. [30] et [32]) sont apparus, traitant les problèmes structure avec des discrétisations volumes finis.

Du côté fluide les deux approches sont aussi possibles. Les solveurs fluides formulés en volumes finis sont très évolués, en particulier dans le domaine de la turbulence.

Le couplage fluide-structure peut mixer ces deux approches, sauf dans le cas monolithique. On rencontrera fréquemment des codes monolithiques basés sur les éléments finis [50] et des codes découplés ayant une discrétisation fluide volumes finis et une discrétisation structure éléments finis [43]. Les codes monolithiques volumes finis sont plus récents et plus rares. Les codes découplés ayant pour discrétisation fluide les éléments finis et pour discrétisation solide les volumes finis sont inexistants. On rencontre aussi des codes découplés qui n'utilisent qu'une formulation : tout "volumes finis" [109] ou tout "éléments finis" [95].

Résumons en deux tableaux les grandes lignes des algorithmes de couplage et les discrétisations :

	Monolithique	**Découplé**
Avantages	☞ Couplage fort ☞ Meilleure convergence	☞ Plus rapide ☞ Modulable et souple
Inconvénients	☛ Gourmand en mémoire ☛ Formulation peu souple (tous les outils doivent être intégrés au code) ☛ Gestion des interfaces en mouvement difficile à coder	☛ Couplage faible à priori ☛ Contraintes et déplacements discontinus ☛ Convergence problématique

	Volumes finis	**Éléments finis**
Avantages	☞ Conservatif localement et globalement ☞ Approche physique (chaque contribution identifiée physiquement) ☞ Traite les phénomènes non-linéaires	☞ Ordre de précision en espace modulable à volonté ☞ Champ d'application universel
Inconvénients	☛ Difficulté de monter en précision	☛ Méthode non-conversative localement

1.3 Conclusion

L'interaction fluide-structure est un pont entre les deux domaines distincts que sont la mécanique des fluides et la mécanique des solides. Cette discipline est par nature très vaste ; elle touche de nombreux corps de métier (le génie civil, le domaine nucléaire et biomédical, les métiers de la mer, l'aéronautique et l'aérospatiale...). Elle a commencé à se développer de plus en plus avec l'apparition du calcul parallèle dans les années 1970. Ces dix dernières années, elle a connu un important essor dans la communauté du calcul numérique. Les équipes de recherche cherchent à étendre les capacités de leur code.

L'équipe numérique du Laboratoire de Mécanique des Fluides (LMF) de l'École Centrale Nantes (ECN) s'est donc intéressée à cette discipline, et plus particulièrement à l'interaction entre les fluides et les corps élancés, c'est-à-dire les structures ayant une dimension beaucoup plus grande devant les deux autres. Cette partie de l'interaction fluide-structure constituera le sujet de cette thèse.

Un programme d'interaction fluide-structure a besoin de quatre éléments pour fonctionner. Le solveur fluide est dans notre cas le solveur RANSE **ISIS** développé depuis dix ans dans l'équipe (EMN). Il fait l'objet du chapitre suivant (2). Le solveur structures élancées a été développé lors de cette thèse à partir des travaux de F. Boyer et de D. Primault. Il est décrit au chapitre 3. Viennent ensuite les problèmes liés au couplage. Le remaillage fluide a pris une place importante dans ce travail. Le chapitre 4 lui est entièrement consacré. Le chapitre 5 traite du couplage en espace et en temps en général. Les interpolations des efforts et des déplacements à l'interface y sont décrites tout comme l'algorithme de couplage temporel. Enfin, le dernier chapitre montre quelques applications IFS réalisées durant cette thèse.

Chapitre 2

Solveur de dynamique des fluides : ISIS

Dans le chapitre 1, nous avons vu qu'un code IFS nécessite deux solveurs. L'objet de ce chapitre est le solveur fluide, **ISIS**, qui est développé depuis une dizaine d'années au Laboratoire de Mécanique des Fluides de l'École Centrale Nantes. Mon travail lors de cette thèse m'a amené à modifier **ISIS** pour y intégrer la gestion de l'interaction fluide-structure. Cependant, je ne suis pas intervenu au sein de la résolution fluide en elle-même. Le contenu de ce chapitre est un résumé succinct du travail effectué par l'Équipe Modélisation Numérique (EMN) de Centrale Nantes durant cette dernière décennie.

Nous nous intéresserons, tout d'abord, à la formulation des équations de Navier-Stokes, puis à leur prise en compte et à leur résolution au sein d'**ISIS**. Nous terminerons ce chapitre par la gestion de la turbulence au sein de ce **code Volumes-Finis**.

2.1 Des lois de conservation aux équations de Navier-Stokes

Cette section décrit comment, à partir des lois de conservation, on forme le système d'équations locales puis intégrées (sur un volume $\mathcal{V}$ du maillage), qui sont codées au sein du solveur fluide **ISIS**.

2.1.1 Examen du point de vue adopté

En mécanique des fluides, l'approche eulérienne est toujours préférée à son homologue lagrangienne. En effet, lors des problèmes ou expériences de mécanique des fluides étudiés, on ne s'intéresse qu'à la cinématique des particules qui passent à un endroit donné sans se préoccuper d'où elles proviennent, ni de ce qu'elles deviendront par la suite. Au début, le point de vue adopté dans **ISIS** a été par conséquent *purement eulérien*. Cela s'est avéré naturel pour le calcul d'écoulements autour de corps fixes.

Cependant, de nos jours, pour pouvoir répondre aux attentes des industriels, le solveur fluide doit aussi pouvoir prendre en compte les mouvements de corps solides et même déformables lors de la résolution de l'écoulement. Il est alors nécessaire d'élargir et d'étendre cette vision purement eulérienne, "trop immobile" pour ces derniers, à un point de vue

qui, sans être matériel, va posséder son propre champ de vitesse. Elle peut être qualifiée d'**ALE** (Arbitrary Lagrangian Eulerian). Une conséquence directe est le fait que le domaine de calcul doit s'adapter aux contours des solides. Les éléments le composant vont devoir se déformer et suivre les corps. Ce processus est appelé "remaillage". Il sera traité en détail dans le chapitre 4.

Le solveur **ISIS** se base sur une discrétisation Volumes-Finis des équations de Navier-Stokes. Le domaine de calcul est découpé en petits volumes élémentaires, qui vont constituer le maillage fluide. Les inconnues du système sont attachées à une des caractéristiques géométriques de ce maillage (noeud du maillage, centre des cellules ou encore centre des faces). Chaque élément de ce maillage, chaque "maille" ou "cellule" peut, comme dit précédemment, se déformer et *possède une vitesse propre*. Pour résoudre notre problème fluide, il est donc nécessaire de modifier les équations de conservation pour en tenir compte. Des *dérivations par rapport à un domaine mobile* apparaissent alors. C'est l'objet de la section suivante.

2.1.2 Dérivation selon un champ de vitesse quelconque

Soit $\overrightarrow{U_d}$ un champ de vitesse quelconque et $g(\overrightarrow{x}, t)$ une fonction scalaire représentant une grandeur matérielle exprimée en coordonnées eulériennes.

On définit alors une nouvelle dérivation appelée dérivation par rapport au temps, en suivant un point ou un domaine animé du champ de vitesse $\overrightarrow{U_d}$. Pour simplifier, on parlera de dérivation temporelle selon le champ de vitesse $\overrightarrow{U_d}$. On notera cette dérivation par $\delta/\delta t$ sans faire apparaître explicitement le symbole $\overrightarrow{U_d}$.

On a la relation ponctuelle :

$$\frac{\delta g}{\delta t} = \frac{\partial g}{\partial t} + \overrightarrow{grad}(g) \cdot \overrightarrow{U_d}$$

Pour un domaine $\mathcal{D}_d$ de frontière $\partial\mathcal{D}_d$ animé du champ de vitesse $\overrightarrow{U_d}$, on montre que :

$$\frac{\delta}{\delta t} \iiint_{\mathcal{D}_d} g dv = \iiint_{\mathcal{D}_d} \frac{\partial g}{\partial t} dv + \iint_{\partial\mathcal{D}_d} g \overrightarrow{U_d} \cdot \overrightarrow{n} dS$$

Cette relation s'interprète de la façon suivante :

Variation de l'intégrale de g dans le domaine mobile $\mathcal{D}_d$ = Intégrale de la variation temporelle de g sur $\mathcal{D}_d$ + Flux convectif de g à travers la frontière $\partial\mathcal{D}_d$

Rappelons ici le théorème établissant l'expression de la dérivée particulaire d'une intégrale de volume. Soit $\mathcal{D}_m$ un domaine matériel (les particules contenues dans $\mathcal{D}_m$ sont animées de la vitesse $\overrightarrow{U}(x,t)$). Sous certaines hypothèses de régularité, on a :

$$\frac{d}{dt} \iiint_{\mathcal{D}_m} g dv = \iiint_{\mathcal{D}_m} \frac{\partial g}{\partial t} dv + \iint_{\partial\mathcal{D}_m} g \overrightarrow{U} \cdot \overrightarrow{n} dS \tag{2.1}$$

Pour le domaine matériel $\mathcal{D}_m$ coïncidant avec $\mathcal{D}_d$ à l'instant t, on a alors la relation :

$$\boxed{\frac{d}{dt}\iiint_{\mathcal{D}_m} g\,dv = \frac{\delta}{\delta t}\iiint_{\mathcal{D}_d} g\,dv + \iint_{\partial\mathcal{D}_d} g\left(\overrightarrow{U} - \overrightarrow{U_d}\right)\cdot\overrightarrow{n}\,dS} \tag{2.2}$$

2.1.3 Formulation Volumes-Finis des équations de Navier-Stokes

Faisons l'hypothèse des **écoulements visqueux isothermes**. Les équations de conservation sont donc réduites à la conservation de la masse et de la quantité de mouvement.

Considérons un domaine $\mathcal{D}_d$ de frontière $\partial\mathcal{D}_d$ animé du champ de vitesse $\overrightarrow{U_d}$ et $\mathcal{D}_m$ le domaine matériel coïncidant à l'instant t considéré.

Conservation de la masse :

$$\frac{d}{dt}\iiint_{\mathcal{D}_m} \rho\,dv = 0$$

Conservation de la quantité de mouvement :

$$\frac{d}{dt}\iiint_{\mathcal{D}_m} \rho\overrightarrow{U}\,dv = \iiint_{\mathcal{D}_m} \rho\overrightarrow{f_v}\,dv + \iint_{\partial\mathcal{D}_m} \overrightarrow{T}\,dS$$

$\overrightarrow{f_v}$: force volumique (dans notre cas, elle se limitera à la gravité)
$\overrightarrow{T}$: vecteur contrainte $\overrightarrow{T} = \overrightarrow{\overrightarrow{\sigma}}\cdot\overrightarrow{n}$, $\overrightarrow{\overrightarrow{\sigma}}$: tenseur des contraintes

Relation de comportement (fluide newtonien) :

$$\overrightarrow{\overrightarrow{\sigma}} = -(p + \frac{2\mu}{3}div\,(\overrightarrow{U}))\,\mathbb{I} + 2\,\mu\,\mathbb{D}$$

p : pression
$\overrightarrow{U}$: vitesse des particules fluides
μ : viscosité dynamique
$\mathbb{D}$: tenseur des taux de déformation

Remarque : la conservation de la quantité de mouvement n'a pas besoin d'être exprimée en moment car celle-ci est équivalente à la symétrie du tenseur des contraintes.

On en déduit alors en utilisant les relations (2.2) :

$$\boxed{\frac{\delta}{\delta t}\iiint_{\mathcal{D}_d} \rho\,dv + \iint_{\partial\mathcal{D}_d} \rho\left[(\overrightarrow{U} - \overrightarrow{U_d})\cdot\overrightarrow{n}\right]dS = 0} \tag{2.3}$$

$$\boxed{\begin{aligned}\frac{\delta}{\delta t}\iiint_{\mathcal{D}_d} \rho\overrightarrow{U}\,dv + \iint_{\partial\mathcal{D}_d} \rho\overrightarrow{U}\left[(\overrightarrow{U} - \overrightarrow{U_d})\cdot\overrightarrow{n}\right]dS = \iiint_{\mathcal{D}_d} -\overrightarrow{grad}(p) + \rho\overrightarrow{g}\,dv \\ + \iiint_{\mathcal{D}_d} -\overrightarrow{grad}(\frac{2\mu}{3}div\,(\overrightarrow{U})) + \overrightarrow{div}\,(2\,\mu\,\mathbb{D})\,dv\end{aligned}} \tag{2.4}$$

Remarque : Les vitesses $\overrightarrow{U}$ et $\overrightarrow{U_d}$ sont la plupart du temps différentes. La première correspond à la vitesse des particules fluides. La seconde est la vitesse de déplacement du domaine $\mathcal{D}_d$. Une fois les équations discrétisées, $\overrightarrow{U_d}$ pourra être assimilée à la vitesse de déplacement du maillage.

2.1.4 Évaluation des intégrales

2.1.4.1 Intégrales de volume

La formulation Volumes-Finis, de manière générale, implique d'évaluer des intégrales de volume. Considérons une cellule $\mathcal{V}$, de volume V et de centre C_c. On approxime l'intégrale d'une fonction f sur un domaine $\mathcal{V}$ par le produit de son volume V avec la valeur de la fonction f au centre du domaine noté C_c :

$$\iiint_{\mathcal{V}} f dv \approx f_{C_c} V$$

Soit $g(\overrightarrow{x}, t)$ une autre fonction. En approximation Volumes-Finis, on peut aussi écrire :

$$\iiint_{\mathcal{V}} f\, g dv \approx f_{C_c} g_{C_c} V$$

$$\iiint_{\mathcal{V}} \frac{f}{g} dv \approx \frac{f_{C_c}}{g_{C_c}} V$$

On montre que *ces trois approximations sont précises au second ordre, si C_c est situé au barycentre géométrique du domaine.*

2.1.4.2 Intégrales de surface

L'écriture des équations de Navier-Stokes avec la formulation Volumes-Finis fait apparaître différents flux à travers les faces du maillage fluide. Ces flux se traduisent par des intégrales sur ces faces, que l'on peut approximer par :

$$\iint_{\mathcal{S}} Q \overrightarrow{dS} \approx Q_f \overrightarrow{S_f}$$

où $\overrightarrow{S_f}$ est la surface de la face et Q_f est la quantité à reconstruire au centre de la face à partir des valeurs des quantités cellulaires (Q_L et Q_R) de part et d'autre de la face.

Nous venons de voir dans cette section 2.1, qu'**ISIS** est un solveur Volumes-Finis. Nous avons vu aussi les contraintes imposées par la prise en compte du mouvement d'un ou plusieurs corps mobiles au sein d'un écoulement. La section suivante traite des méthodes numériques employées dans le code pour résoudre les équations de Navier-Stokes.

2.2 Résolution numérique des équations de Navier-Stokes dans ISIS

L'Équipe Modélisation Numérique (E.M.N.) du laboratoire de Mécanique des Fluides de Centrale Nantes développe, depuis plusieurs années dans le code **ISIS**, des techniques

numériques de pointe, afin de résoudre **les équations de Navier-Stokes**, pour un écoulement *incompressible et turbulent*.

Nous allons tout d'abord aborder les équations de travail et leurs hypothèses. Puis les parties suivantes examineront le traitement et la résolution numérique de ces équations par la méthode des Volumes-Finis pour des maillages non-structurés constitués d'éléments de forme arbitraire.

2.2.1 Équations de Navier-Stokes en moyenne de Reynolds

On considère un écoulement **incompressible** et **isotherme** d'un fluide de masse volumique ρ et viscosité dynamique μ. Sous ces hypothèses, en notant U_i les composantes cartésiennes dans la direction i de la vitesse, f_i celles des forces volumiques et p la pression, l'écoulement est régi par les équations de Navier-Stokes, représentant la conservation de la quantité de mouvement et de la masse. On écrit les équations sous forme intégrale, pour un volume V délimité par la surface S, se déplaçant à la vitesse $\overrightarrow{U_d}$ et ayant $\overrightarrow{n}$ pour vecteur normal unitaire sortant. À partir des équations de continuité et de conservation de quantité de mouvement, il vient :

$$\frac{\partial}{\partial t}\int_V \rho\, U_i\, dV + \int_S \rho\, U_i(\overrightarrow{U} - \overrightarrow{U_d}) \cdot \overrightarrow{n}\, dS = \int_S (\tau_{ij}\, \overrightarrow{I}_j - p\, \overrightarrow{I}_i) \cdot \overrightarrow{n}\, dS + \int_V \rho f_i\, dV \quad (2.5a)$$

$$\frac{\partial}{\partial t}\int_V \rho\, dV + \int_S \rho\, (\overrightarrow{U} - \overrightarrow{U_d}) \cdot \overrightarrow{n}\, dS = 0 \quad (2.5b)$$

On prend pour convention la sommation des termes correspondant à des indices répétés. Le vecteur $\overrightarrow{I}_k$ a pour seule composante non nulle et unitaire la composante k. Le terme τ_{ij} représente le tenseur des contraintes visqueuses qui s'exprime, pour un fluide newtonien, par la relation suivante :

$$\tau_{ij} = 2\mu S_{ij} = \mu(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}) \quad (2.6)$$

Cette équation est dite "instantanée". Pour la résoudre de manière correcte, il faut prendre en compte toutes les échelles de Kolmogorov de la turbulence. C'est réalisé avec la méthode dite "DNS" (Direct Numerical Simulation). Mais ce type de calcul n'est envisageable actuellement que pour des nombres de Reynolds très faibles. Pour des tests plus complexes, le problème de la turbulence doit être simplifié.

Ainsi, on introduit une décomposition de Reynolds, consistant à écrire un champ instantané Φ comme la somme d'un champ moyen $\overline{\Phi}$, en terme d'espace et de temps, et d'un champ fluctuant Φ'. L'application de cette décomposition aux équations de Navier-Stokes permet d'établir les équations régissant les champs moyens. Cette transformation fait apparaître un terme supplémentaire $-\rho\overline{u_i' u_j'}$, appelé **tenseur de Reynolds**, provenant des non-linéarités des équations de Navier-Stokes. Ce terme nécessite l'introduction d'une modélisation pour fermer le système des équations de Navier-Stokes en moyenne de Reynolds. Dans le cas des fermetures à viscosité turbulente, basées sur l'hypothèse de

Boussinesq, le tenseur de Reynolds est lié au tenseur des taux de déformation du champ moyen de vitesse par analogie avec la relation de Newton :

$$-\rho\overline{u_i'u_j'} = 2\mu_t S_{ij} - \frac{2}{3}\rho k\delta_{ij} \tag{2.7}$$

δ_{ij} est le symbole de Kronecker. Les termes μ_t et k représentent, respectivement, le coefficient de viscosité turbulente et l'énergie cinétique turbulente massique. Le traitement de ces termes sera détaillé dans la partie concernant la turbulence (§ 2.3).

Pour la suite, on s'abstient de surligner les champs moyens, pour ne pas alourdir les notations. **Les équations de Navier-Stokes en moyenne de Reynolds** prennent finalement la forme :

$$\frac{\partial}{\partial t}\int_V \rho U_i\, dV + \int_S \rho\, U_i(\overrightarrow{U} - \overrightarrow{U_d})\cdot\overrightarrow{n}\, dS = \int_S (T_{ij}\overrightarrow{I}_j - P\overrightarrow{I}_i)\cdot\overrightarrow{n}\, dS + \int_V \rho f_i dV \tag{2.8a}$$

$$\frac{\partial}{\partial t}\int_V \rho dV + \int_S \rho\,(\overrightarrow{U} - \overrightarrow{U_d})\cdot\overrightarrow{n}\, dS = 0 \tag{2.8b}$$

Le tenseur T_{ij} regroupe les contraintes visqueuses et la partie anisotrope du tenseur de Reynolds, la partie isotrope qui fait intervenir k étant incorporée au terme de pression P.

Des équations de transport supplémentaires peuvent être considérées pour les quantités turbulentes (cf. § 2.3) et dans le cas de calculs multifluides (ce dernier cas, bien que présent dans **ISIS**, ne sera pas abordé dans cette thèse). On s'intéresse maintenant, à la description des différentes méthodes de discrétisation qui sont applicables à toutes les équations de transport rencontrées.

2.2.2 Discrétisation des équations de transport

Comme dit précédemment, la discrétisation Volumes-Finis *cell-centered collocatif* implique que toutes les variables sont disposées au centre géométrique des volumes de contrôle. La forme de ces derniers est **quelconque** : ils sont constitués d'un nombre **arbitraire** de faces. Comme nous allons le voir, les techniques de discrétisation, employées dans cette section, se basent sur les faces. Ainsi le traitement des volumes de contrôle de forme quelconque est naturel. La conséquence directe est la possibilité de traiter des problèmes avec des maillages autour de géométries très complexes. Notons aussi que cela permet le développement et l'utilisation de méthodes d'adaptation locale de maillages [47].

Pour traiter une équation de transport d'une variable générique Q, la méthode des Volumes-Finis transforme les intégrales de volume, sur chaque cellule du maillage, en des intégrales de surface par les *théorèmes de Gauss et de Green-Ostrogradsky*. Les intégrales de surface et de volume sont évaluées selon des approximations précises à l'ordre 2 correspondant à l'intégration d'un polynôme linéaire sur ces éléments. Elles nécessitent la valeur de l'intégrant calculée aux centres des cellules et reconstruite (à l'ordre 2 aussi) aux centres des faces.

L'équation de transport d'une variable générique Q sur une cellule $cell_V$ de centre C, limitée par un nombre quelconque de faces f, peut s'écrire sous forme semi-discrète :

$$\frac{\delta}{\delta\tau}(\rho VQ)_C + \frac{\partial}{\partial\tau}(\rho VQ)_C + \frac{\partial}{\partial t}(\rho VQ)_C + \sum_f (\mathcal{F}c_f - \mathcal{F}d_f) = (S_Q^{cell_V}) + \sum_f (S_Q^f)$$
$$\mathcal{F}c_f = \dot{m}_f\, Q_f \qquad \mathcal{F}d_f = (\Gamma_Q)_f(\overrightarrow{grad}(Q_f)\ \cdot\ \overrightarrow{I}_k)(S_k)_f \tag{2.9}$$

Γ_Q représente un coefficient de diffusion. Les termes $\mathcal{F}c_f$ et $\mathcal{F}d_f$ sont respectivement les flux de convection et de diffusion à travers la face f, $\dot{m}_f$ correspondant au flux de masse à travers cette face. $S_Q^{cell_V}$ et S_Q^f représentent les termes sources volumiques et surfaciques. Dans le cas particulier de l'équation de transport de la quantité de mouvement, le terme de pression est inclus dans le terme source volumique. Le flux de masse à travers les faces s'écrit :

$$\dot{m}_f = \rho\,(\overrightarrow{U} - \overrightarrow{U}_d)_f\ \cdot\ \overrightarrow{S}_f \tag{2.10}$$

Les dérivées temporelles sont évaluées généralement par une approximation décentrée du second ordre, permettant d'écrire une loi de la forme :

$$\frac{\delta A}{\delta t} \approxeq e^c A^c + e^p A^p + e^q A^q \tag{2.11}$$

L'indice c fait référence à l'instant courant, les indices p et q se référant aux pas de temps antérieurs. Par la suite, ces coefficients temporels seront notés indifféremment e^q, e^p, e^c ou e_q, e_p, e_c.

Par ailleurs, le premier terme du membre de gauche de l'équation 2.9 correspond au terme pseudo-stationnaire introduit pour permettre le traitement de problèmes stationnaires. Son rôle est de stabiliser le processus de résolution en renforçant le caractère de dominance diagonale des systèmes linéaires (cf. § 2.2.5.2). La dérivée correspondante est évaluée par :

$$\frac{\delta A}{\delta\tau} = \frac{A^c - A^{c0}}{\Delta\tau} \tag{2.12}$$

A^{c0} est l'estimation précédente de la quantité A^c, du point de vue du processus itératif non-linéaire (c'est-à-dire le champ linéarisé de la méthode de Picard). Finalement, l'équation de transport d'une variable générique s'écrit sous forme discrète :

$$(e^c + \frac{1}{\Delta\tau})(\rho VQ)_C^c + \sum_f (\mathcal{F}c_f - \mathcal{F}d_f) = (S_Q^{cell_V}) + \sum_f (S_Q^f)$$
$$-(e\rho VQ)_C^p - (e\rho VQ)_C^q + \frac{(\rho VQ)_C^{c0}}{\Delta\tau} \tag{2.13}$$

L'équation de conservation de la masse 2.8b peut, quant à elle, s'exprimer de façon discrète de la manière suivante :

$$\sum_f \overrightarrow{U}_f\ \cdot\ \overrightarrow{S}_f = 0 \tag{2.14}$$

En effet, le déplacement de la surface limitant le volume de contrôle vérifie la condition de conservation spatiale :

$$\frac{\partial}{\partial t}\iiint_V dV - \iint_S \overrightarrow{U}_d \cdot \overrightarrow{n} dS = 0 \tag{2.15}$$

Dans les formules présentées ici, les quantités sont aux centres des faces. En Volumes-Finis, les variables sont connues aux centres des cellules. Il est donc nécessaire de reconstituer ces dernières sur les faces. La section suivante traite des différentes techniques de reconstruction existantes.

2.2.3 Reconstruction sur les faces

Pour que la discrétisation soit effectivement d'ordre 2, il est nécessaire que la reconstruction sur les faces ait la même précision. Dans le cadre de maillages non-structurés avec des volumes de contrôle quelconques, il est complexe de connaître en un point des informations provenant de points spatialement éloignés. *Pour la reconstruction des quantités sur une face, on se limite donc à des approximations faisant intervenir les quantités et leurs gradients dans les deux cellules liées à la face considérée.*

2.2.3.1 Reconstruction centrée

À l'exception des termes de convection et des flux de masse, les grandeurs au centre d'une face sont reconstruites par approximations centrées, reposant sur des interpolations linéaires entre les centres des cellules et de la face. On décrit dans cette section les techniques utilisées pour la reconstruction d'une grandeur générique Q, ainsi que son gradient dans la direction normale à la face, nécessaire pour les termes diffusifs.

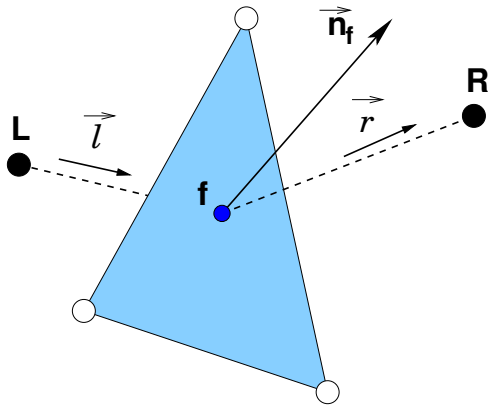


FIG. 2.1: Reconstruction sur une face

La quantité Q_f sur une face peut s'exprimer à partir des valeurs de la quantité et de son gradient dans les cellules adjacentes L et R (cf. fig. 2.1), en utilisant une interpolation linéaire. Deux reconstructions peuvent donc être considérées, qui seront indicées $+$ et $-$:

$$Q_L \simeq Q_f - \overrightarrow{Lf} \cdot \overrightarrow{grad}(Q)_f \qquad Q_R \simeq Q_f + \overrightarrow{fR} \cdot \overrightarrow{grad}(Q)_f \tag{2.16}$$

On introduit une décomposition du gradient dans la direction normale, de manière à faire apparaître les vecteurs $\overrightarrow{l}$ et $\overrightarrow{r}$, unitaires dans les directions des segments [Lf] et [fR] :

$$\forall \beta^+ \qquad \overrightarrow{grad}(Q)_f \cdot \overrightarrow{n} = \beta^+ \overrightarrow{grad}(Q)_f \cdot \overrightarrow{r} + \overrightarrow{e}^+ \cdot \overrightarrow{grad}(Q)_f \quad \text{si} : \overrightarrow{e}^+ \triangleq \overrightarrow{n} - \beta^+ \overrightarrow{r} \tag{2.17}$$

$$\forall \beta^- \qquad \overrightarrow{grad}(Q)_f \cdot \overrightarrow{n} = \beta^- \overrightarrow{grad}(Q)_f \cdot \overrightarrow{l} + \overrightarrow{e}^- \cdot \overrightarrow{grad}(Q)_f \quad \text{si} : \overrightarrow{e}^- \triangleq \overrightarrow{n} - \beta^- \overrightarrow{l} \tag{2.18}$$

Les coefficients $\beta^\pm$ sont choisis de manière à assurer les conditions $\overrightarrow{e}^- \cdot \overrightarrow{n} = 0$ et $\overrightarrow{e}^+ \cdot \overrightarrow{n} = 0$. On peut noter que les vecteurs $\overrightarrow{e}^\pm$ sont nuls dans le cas de maillages orthogonaux. Les reconstructions 2.16 donnent en introduisant cette décomposition :

$$\begin{aligned} Q_L &= Q_f - h^- \left(\overrightarrow{grad}(Q)_f \cdot \overrightarrow{n} - \overrightarrow{grad}(Q)_f \cdot \overrightarrow{e}^- \right) \\ Q_R &= Q_f + h^+ \left(\overrightarrow{grad}(Q)_f \cdot \overrightarrow{n} - \overrightarrow{grad}(Q)_f \cdot \overrightarrow{e}^+ \right) \end{aligned} \tag{2.19a}$$

avec les distances $h^\pm$:

$$h^- \triangleq \overrightarrow{Lf} \cdot \overrightarrow{n} \quad \text{et} \quad h^+ \triangleq \overrightarrow{fR} \cdot \overrightarrow{n} \tag{2.19b}$$

On peut alors montrer qu'une reconstruction au second ordre est obtenue par :

$$Q_f = \frac{h^+}{h} Q_L + \frac{h^-}{h} Q_R + \frac{h^+ h^-}{h} \overrightarrow{grad}(Q)_f \cdot (\overrightarrow{e}^+ - \overrightarrow{e}^-) \tag{2.20a}$$

avec la distance h :

$$h \triangleq h^+ + h^- = (\overrightarrow{Lf} + \overrightarrow{fR}) \cdot \overrightarrow{n} = \overrightarrow{LR} \cdot \overrightarrow{n} \tag{2.20b}$$

Les deux premiers termes de la relation 2.20a fournissent une reconstruction de la quantité au premier ordre. Pour connaître la valeur du gradient sur la face, on utilise cette reconstruction au premier ordre, à partir des gradients de la quantité, connus pour les cellules adjacentes. On a alors la formule de reconstruction :

$$Q_f = \frac{h^+}{h} Q_L + \frac{h^-}{h} Q_R + \boxed{\left(\frac{h^- \overrightarrow{H}^+ - h^+ \overrightarrow{H}^-}{h} \right) \cdot \left(\frac{h^+}{h} \overrightarrow{grad}(Q)_L + \frac{h^-}{h} \overrightarrow{grad}(Q)_R \right)} \tag{2.21}$$

avec :

$$\begin{aligned} \overrightarrow{H}^- \triangleq h^- \overrightarrow{e}^- &= \left(\overrightarrow{Lf} \cdot \overrightarrow{n} \right) \overrightarrow{n} - \overrightarrow{Lf} \\ \overrightarrow{H}^+ \triangleq h^+ \overrightarrow{e}^+ &= \left(\overrightarrow{fR} \cdot \overrightarrow{n} \right) \overrightarrow{n} - \overrightarrow{fR} \end{aligned} \tag{2.22}$$

Il est important de souligner que le terme encadré dans la formule de reconstruction 2.21 est nul pour un maillage orthogonal. Puisqu'il fait intervenir le gradient de la quantité dans les cellules adjacentes à la face, il sera traité de manière explicite lors de la résolution.

Pour reconstruire sur la face la valeur du gradient dans la direction normale, on emploie un raisonnement similaire, permettant d'obtenir à partir de 2.19a la reconstruction :

$$\overrightarrow{grad}(Q)\cdot\vec{n}|_f = \frac{Q_R - Q_L}{h} + \boxed{\left(\vec{n} - \frac{\overrightarrow{LR}}{\overrightarrow{LR}\cdot\vec{n}}\right)\cdot\left(\frac{h^+}{h}\overrightarrow{grad}(Q)_L + \frac{h^-}{h}\overrightarrow{grad}(Q)_R\right)} \tag{2.23}$$

Comme pour la formule de reconstruction de la quantité, le terme encadré tend vers zéro pour un maillage orthogonal et sera traité explicitement lors de la résolution.

2.2.3.2 Reconstruction décentrée

De manière à *renforcer la stabilité du schéma numérique et à éviter l'apparition d'oscillations non-physiques dans la solution*, des reconstructions décentrées plus sophistiquées sont utilisées pour le traitement des termes de convection. On ne peut donner l'ordre de précision de ces différentes reconstructions, car il dépend du maillage utilisé et de la nature physique du problème à résoudre. Nous pouvons juste dire qu'il est compris entre 1 et 2.

Schéma hybride

Le schéma de reconstruction **hybride** inclus dans le code **ISIS** est basé sur *une pondération des valeurs interpolées linéairement à partir des cellules adjacentes*. De plus amples informations sur les fondements de ce type de schéma sont données dans [72] par S.V. Patankar. Contrairement aux approches classiques [31, 40], le coefficient de décentrement n'est pas constant pour l'ensemble du domaine de calcul, mais dépend du nombre de Peclet sur la face :

$$Pe_f = \frac{\dot{m}_f \left\|\overrightarrow{LR}\right\|}{2S\Gamma_Q} \tag{2.24}$$

La précision de la reconstruction n'est donc pas nécessairement uniforme sur l'ensemble du domaine de calcul mais s'adapte localement au mieux au problème traité. En outre, l'orientation relative de la face et de la vitesse est prise en compte. Le coefficient de décentrement d, entre les quantités amont et aval de la face, est calculé à partir d'un schéma exponentiel, assurant une transition lisse :

$$\begin{aligned} Q_f &= d_L Q_L + d_R Q_R + \boxed{d_L\overrightarrow{Lf}\cdot\overrightarrow{grad}(Q)|_L + d_R\overrightarrow{Rf}\cdot\overrightarrow{grad}(Q)|_R} \\ d &= \frac{\exp(Pe_f)}{1+\exp(Pe_f)} \quad d_L \triangleq 1 - d_R \end{aligned} \tag{2.25}$$

Le terme encadré est traité de manière explicite lors de la résolution, comme pour les reconstructions centrées.

Schéma GDS

Le schéma hybride présenté précédemment n'assure cependant pas le caractère borné de la reconstruction. Cette propriété est particulièrement importante dans le contexte d'écoulements multifluides. Le schéma **GDS** (*Gamma Differencing Scheme*) [54] a ainsi été intégré dans ISIS, de manière à *préserver la monotonie de la solution.*

Il repose sur une analyse en terme de variables normalisées [56]. On suppose connue la valeur de la quantité générique Q en trois points, U, C et D positionnés selon la direction de convection. On cherche à reconstruire la quantité sur une face située entre C et D (cf. fig. 2.2). On définit le système de variables normalisées par :

$$\tilde{Q} = \frac{Q - Q_U}{Q_D - Q_U} \tag{2.26}$$

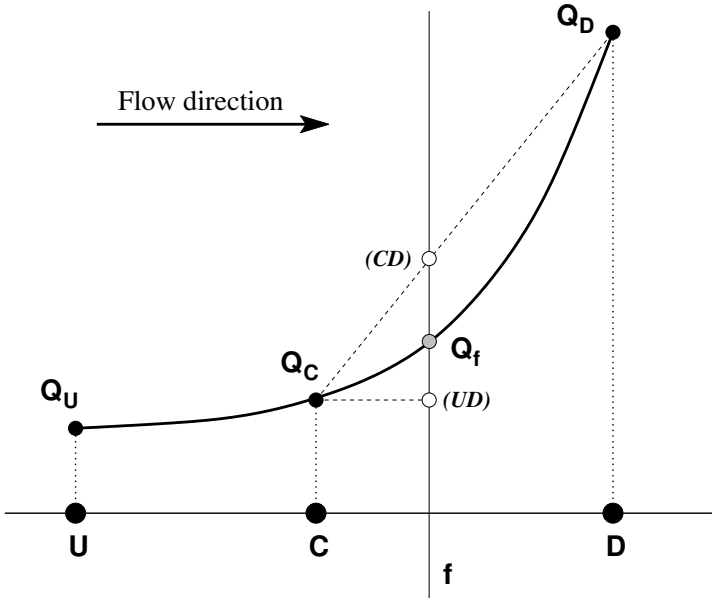


FIG. 2.2: Variation de Q dans la direction de l'écoulement

Le schéma GDS propose une reconstruction s'appuyant sur ces trois points, de la forme :

$$\tilde{Q}_f = f(\tilde{Q}_C) \tag{2.27}$$

Pour éviter les oscillations non physiques, il est nécessaire que Q_C soit bornée, entre les valeurs $Min(Q_U, Q_D)$ et $Max(Q_U, Q_D)$. En terme de variables normalisées, cette condition s'écrit :

$$0 \leqslant \tilde{Q}_C \leqslant 1 \tag{2.28}$$

De manière à établir une reconstruction sur la face décentrée et préservant le caractère borné, le schéma GDS doit vérifier les contraintes suivantes :

- pour $\tilde{Q}_C < 0$, $\tilde{Q}_f = \tilde{Q}_C$
- pour $0 \leqslant \tilde{Q}_C \leqslant 1$, $\tilde{Q}_f$ est bornée par $\tilde{Q}_f \geqslant \tilde{Q}_C$ et par l'unité
- pour $\tilde{Q}_C > 1$, $\tilde{Q}_f = \tilde{Q}_C$

Ces conditions sont vérifiées dans la zone grisée sur la figure 2.3. On constate que le schéma décentré du premier ordre (UDS) satisfait juste ces conditions, tandis que le schéma centré (CDS) ne respecte pas ces critères dans le domaine $\tilde{Q}_C \in]-\infty, 0[\cup]1, +\infty[$.

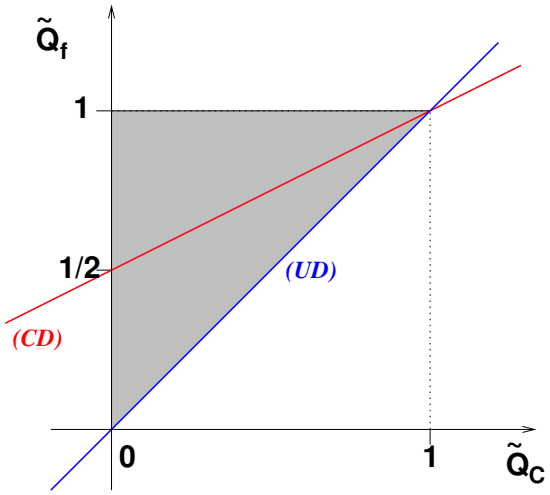


FIG. 2.3: Critères de monotonie en variables normalisées

Pour appliquer cette approche à un environnement multidimensionnel non-structuré, certains aménagements doivent être mis en œuvre. Le point U étant inconnu, la quantité Q_U est évaluée par projection du gradient dans la direction $\overrightarrow{CD}$ (cf. fig. 2.4) :

$$\begin{aligned} Q_U &= Q_C - \overrightarrow{CU} \cdot \overrightarrow{grad}(Q)|_C \\ \text{avec} \quad \overrightarrow{CU} &\triangleq -\overrightarrow{CD} \end{aligned} \tag{2.29}$$

En considérant cette nouvelle définition, la variable normalisée en C s'évalue par :

$$\tilde{Q}_C = 1 - \frac{Q_D - Q_C}{2\overrightarrow{grad}(Q)|_C \cdot \overrightarrow{CD}} \tag{2.30}$$

Cette définition étant mal posée lorsque la quantité Q est constante sur le domaine, on spécifie :

$$\tilde{Q}_C = 0.5 \quad \text{si} \quad |Q_D - Q_C| \leqslant 10^{-6} \quad \text{ou} \quad |\overrightarrow{grad}(Q)|_C \cdot \overrightarrow{CD}| \leqslant 10^{-6} \tag{2.31}$$

Le schéma GDS propose d'établir une transition continue entre le schéma décentré du premier ordre et le schéma centré du deuxième ordre (plus précis) sur l'intervalle $0 < \tilde{Q}_C < 1$. Pour des valeurs faibles de $\tilde{Q}_C$, il faut "rattraper" l'écart entre le schéma

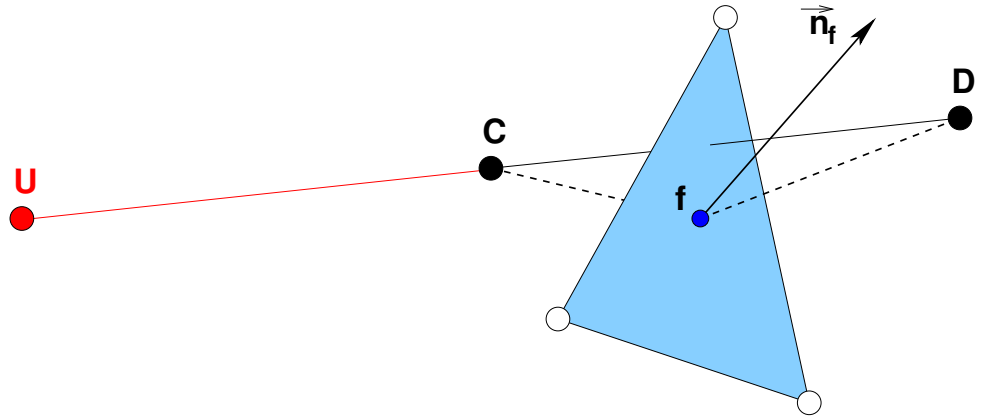


FIG. 2.4: Définition du point imaginaire U

centré et le schéma décentré. Cette transition est effectuée sur l'intervalle $[0, \beta_m]$. Le coefficient β_m prend usuellement la valeur de $\frac{1}{6}$.

$$\begin{cases} \tilde{Q}_C = 0 & \Rightarrow \quad \gamma = 0 \quad \text{Schéma UDS} \\ \tilde{Q}_C = \beta_m & \Rightarrow \quad \gamma = 1 \quad \text{Schéma CDS} \end{cases}$$

Le schéma GDS retient finalement une variation linéaire pour le coefficient de transition, ce qui conduit à une représentation en variables normalisées exposée par la figure 2.5 :

$$\gamma = \frac{1}{\beta_m} \tilde{Q}_C \tag{2.32}$$

En définitive, les caractéristiques du schéma de reconstruction sont résumées dans le tableau 2.1, incluant le facteur d'interpolation :

$$f_x = \frac{\left\| \overrightarrow{fD} \right\|}{\left\| \overrightarrow{CD} \right\|} \tag{2.33}$$

$\tilde{Q}_C$	$\tilde{Q}_f$	Q_f	*Note*
$]-\infty, 0]$	$\tilde{Q}_C$	Q_C	UDS
$]0, \beta_m[$	$-\frac{1}{2\beta_m}\tilde{Q}_C^2 + \left(1 + \frac{1}{2\beta_m}\right)\tilde{Q}_C$	$(1-\gamma(1-f_x))Q_C + \gamma(1-f_x)Q_D$	transition
$[\beta_m, 1[$	$\frac{1}{2} + \frac{1}{2}\tilde{Q}_C$	$f_x Q_C + (1-f_x)Q_D$	CDS
$[1, +\infty[$	$\tilde{Q}_C$	Q_C	UDS

TAB. 2.1: Reconstruction par le schéma GDS

Pour toutes les formules de discrétisation mentionnées précédemment, aucune hypothèse n'est faite concernant la forme ou le nombre de faces des volumes de contrôle. En

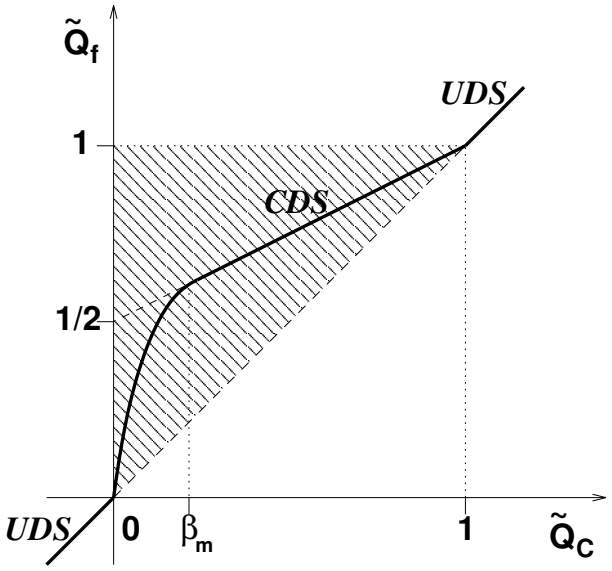


FIG. 2.5: Schéma GDS en variables normalisées

effet, la face est considérée comme l'élément de base des schémas de reconstruction dans le code ISIS. Ainsi, comme annoncé, cette méthodologie basée sur les faces permet le traitement naturel d'un volume de contrôle possédant un nombre de faces arbitraire.

2.2.4 Calcul des gradients sur une cellule

On examine dans cette section les différentes techniques utilisées dans le code ISIS pour évaluer le gradient d'une quantité Q sur une cellule. Deux approches sont proposées : *un calcul par une méthode des moindres carrés* (avec pondération par les distances) et *une technique utilisant le théorème de Gauss.*

2.2.4.1 Méthode des moindres carrés pondérée par les distances

Par mesure de simplicité, la méthode est présentée dans le cas bidimensionnel mais son extension au cas 3D est immédiate. On considère un volume de contrôle C_0 de centre de gravité $\overrightarrow{X_0} = (x_0, y_0)$. On suppose que la quantité q varie linéairement sur celui-ci. Ainsi, la valeur en un point $\overrightarrow{X}$ appartenant à C_0 est donnée par :

$$q(\overrightarrow{X}) = q_{C_0} + \overrightarrow{X_0 X} \cdot (\overrightarrow{grad}(q))_{C_0} \tag{2.34}$$

Dans le cas bidimensionnel, cette relation s'écrit :

$$q(x, y) = q(x_0, y_0) + (x - x_0)q_x + (y - y_0)q_y \tag{2.35}$$

L'évaluation du gradient de q sur la cellule C_0 utilise les n points d'appui qui repèrent les centres des cellules voisines $(C_i)_{i=1,\ldots,n}$ comme illustré sur la figure 2.6. Puisque sont disponibles plus de points que nécessaire, on a un système surdéterminé formé par les n relations :

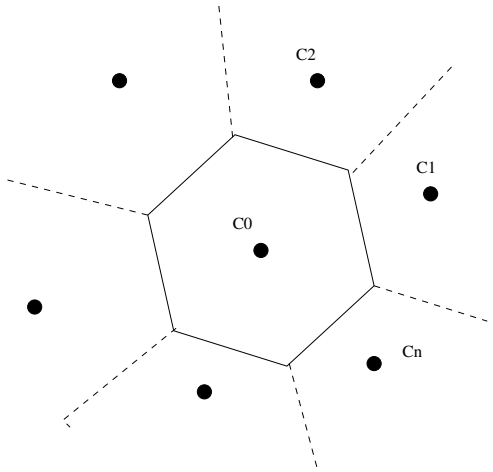


FIG. 2.6: Cellule C_0 et son voisinage

$$q_i = q_0 + (x_i - x_0)q_x + (y_i - y_0)q_y \quad \forall i \in [1, n] \tag{2.36}$$

Le problème est résolu au sens des moindres carrés par minimisation de la fonctionnelle $E_n(q_x, q_y)$:

$$E_n(q_x, q_y) = \sum_{i=1}^{n} \left\{ \Big[\big(q_0 + (x_i - x_0)q_x + (y_i - y_0)q_y\big) - q_i \Big] D_i \right\}^2 \tag{2.37}$$

La quantité D_i correspond au poids donné à un point C_i sur l'ensemble du système. Il va permettre de donner une influence d'autant plus importante à chaque point C_i qu'il se situe à proximité du point de reconstruction C_0. Cette méthode de pondération par les distances est importante pour traiter des maillages étirés [63]. D'après ce qui précède, le poids D_i d'une cellule C_i est défini comme l'inverse de sa distance au point C_0 :

$$D_i = (||\overrightarrow{X_0 X_i}||)^{-1} \quad \forall i \in [1, n]$$

La minimisation de la fonctionnelle, $\dfrac{\partial E_n}{\partial q_x} = 0$ et $\dfrac{\partial E_n}{\partial q_y} = 0$, conduit au gradient de q :

$$\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \frac{1}{S} \begin{pmatrix} S_{yy} & -S_{xy} \\ -S_{xy} & S_{xx} \end{pmatrix} \begin{pmatrix} S_{xq} \\ S_{yq} \end{pmatrix} \tag{2.38a}$$

avec :

$$\begin{aligned} S_{xx} &= \sum_{i=1}^{n} \Big[(x_i - x_0) D_i \Big]^2 \\ S_{yy} &= \sum_{i=1}^{n} \Big[(y_i - y_0) D_i \Big]^2 \\ S_{xy} &= \sum_{i=1}^{n} \Big[(x_i - x_0)(y_i - y_0) D_i^2 \Big] \\ S &= S_{xx} S_{yy} - {S_{xy}}^2 \end{aligned} \tag{2.38b}$$

et :

$$\begin{aligned} S_{xq} &= \sum_{i=1}^{n} \Big[(x_i - x_0)(q_i - q_0) D_i^2 \Big] \\ S_{yq} &= \sum_{i=1}^{n} \Big[(y_i - y_0)(q_i - q_0) D_i^2 \Big] \end{aligned} \tag{2.38c}$$

2.2.4.2 Méthode de Gauss

Une seconde approche possible repose sur le théorème de Gauss. Pour un volume $cell_V$ limité par la surface $face_S$, on a :

$$\iiint_{cell_V} \overrightarrow{grad}(q) dv = \iint_{face_S} q \overrightarrow{n} dS \tag{2.39}$$

En discrétisant la relation précédente 2.39, le gradient sur la cellule peut être approché par :

$$\overrightarrow{grad}(q)\Big|_C = \frac{1}{Vol_C} \sum_f q_f S_f \overrightarrow{n}_f \tag{2.40}$$

La valeur de la quantité sur les faces est reconstruite à partir des cellules adjacentes, par la reconstruction linéaire d'ordre 2 décrite précédemment.

2.2.5 Algorithme de couplage

Une particularité fondamentale des équations de Navier-Stokes pour un écoulement incompressible est le **rôle de la pression**. En effet, cette variable apparaît dans l'équation de quantité de mouvement seulement à travers son gradient et est absente dans l'équation de continuité. Un traitement particulier doit donc être mis en œuvre pour obtenir une équation dont la variable est la pression. C'est l'objet du paragraphe 2.2.5.1. Puis, l'algorithme permettant d'assurer le couplage vitesse-pression est présenté (cf. § 2.2.5.2).

2.2.5.1 Équation de pression

L'établissement d'une équation pour la pression repose sur une transformation de l'équation de continuité. Cette idée est à relier au rôle que joue physiquement la pression dans les équations de Navier-Stokes. En effet, il existe une infinité de champs de vitesse qui satisfont les équations de quantité de mouvement. Et, le gradient de pression va permettre de fixer l'unique solution qui vérifie également la contrainte d'incompressibilité. Dans l'approche adoptée ici, les flux de vitesse aux interfaces sont reconstruits par une approche pseudo-physique, incluant une formulation similaire à l'équation de quantité de mouvement, de manière à faire apparaître les variables de pression.

En utilisant les développements précédents, l'équation de quantité de mouvement semi-discrétisée prend la forme :

$$
\begin{aligned}
(e^c + \frac{1}{\Delta\tau_C})(\rho\overrightarrow{U})^c_C + \frac{a_C\overrightarrow{U}^c_C + \sum_{nb} a_{nb}\overrightarrow{U}^c_{nb} + \overrightarrow{S}}{V^c_C} + \overrightarrow{grad}(P)|_C \\
-\frac{(\rho\overrightarrow{U})^{c0}_C}{\Delta\tau_C} + \frac{(e\rho V\overrightarrow{U})^p_C + (e\rho V\overrightarrow{U})^q_C}{V^c_C} = \overrightarrow{0}
\end{aligned}
\tag{2.41}
$$

$\{a_C, a_{nb}\}$ sont les coefficients diagonaux et extra-diagonaux traités implicitement, correspondant aux termes de convection et de diffusion. $\overrightarrow{S}$ est un terme source contenant toutes les contributions explicites stationnaires. À partir de cette équation, les variables de vitesse aux centres des cellules peuvent s'exprimer de manière compacte par :

$$
\begin{aligned}
\overrightarrow{U}^c_C = &- Cp_C\left(\overrightarrow{\hat{U}}^c_C + \overrightarrow{grad}(P)|_C\right) \\
&+ Cp_C\left(\frac{(\rho\overrightarrow{U})^{c0}_C}{\Delta\tau_C}\right) \\
&- Cp_C\left(\frac{(e\rho V\overrightarrow{U})^p_C + (e\rho V\overrightarrow{U})^q_C}{V^c_C}\right) \\
\text{avec} \quad Cp_C &= \frac{1}{(e^c + \frac{1}{\Delta\tau_C})\rho^c_C + \frac{a_C}{V^c_C}}
\end{aligned}
\tag{2.42}
$$

Le vecteur $\overrightarrow{\hat{U}}$ est homogène à un gradient de pression et contient les termes implicites extra-diagonaux de convection, diffusion et les termes sources :

$$
\overrightarrow{\hat{U}} = \frac{\sum_{nb} a_{nb}\overrightarrow{U}^c_{nb} + \overrightarrow{S}}{V^c_C}
\tag{2.43}
$$

Dans l'esprit de la **méthode de Rhie et Chow** [81], on suppose que les vitesses aux interfaces peuvent s'exprimer selon une formule similaire à 2.42. En indiquant par f chaque quantité ou groupement interpolé aux faces, on a donc :

$$
\begin{aligned}
\overrightarrow{U}^c_f = &- Cp_f\left(\overrightarrow{\hat{U}}_f + \overrightarrow{grad}(P)|_f\right) \\
&+ Cp_f\left(\frac{(\rho\overrightarrow{U})^{c0}_f}{\Delta\tau_f}\right) \\
&- Cp_f\left(\frac{(e\rho V\overrightarrow{U})^p_f + (e\rho V\overrightarrow{U})^q_f}{V^c_f}\right) \\
\text{avec} \quad Cp_f &= \frac{1}{\left(e^c + \frac{1}{\Delta\tau_f}\right)\rho^c_f + \left(\frac{a_C}{V^c}\right)_f}
\end{aligned}
\tag{2.44}
$$

Cette reconstruction est pseudo-physique, dans la mesure où elle est formellement issue de l'équation de quantité de mouvement, mais les termes hors gradient de pression sont simplement interpolés à partir des cellules adjacentes. Une équation de pression peut alors être obtenue en introduisant cette reconstruction dans l'équation de continuité :

$$\begin{aligned}\sum_f Cp_f \overrightarrow{grad}(P)|_f \cdot \overrightarrow{S}_f = &\sum_f \left(-Cp_f \overrightarrow{\hat{U}}_f \right) \cdot \overrightarrow{S}_f \\ &+ \sum_f Cp_f \left(\frac{(\rho \overrightarrow{U})_f^{c0}}{\Delta\tau_f} \right) \cdot \overrightarrow{S}_f \\ &+ \sum_f -Cp_f \left(\frac{(e\rho V \overrightarrow{U})_f^p + (e\rho V \overrightarrow{U})_f^q}{V_f^c} \right) \cdot \overrightarrow{S}_f\end{aligned} \tag{2.45}$$

Le flux du gradient de pression à travers une face f est obtenu par une discrétisation centrée accompagnée d'un terme de correction (pour rester consistant dans le cas de maillages non-orthogonaux) selon 2.23. Lorsque cette équation est résolue, le flux de vitesse $\mathcal{F}(\overrightarrow{U})$ à travers une face, défini par la relation suivante, est nécessairement conservatif :

$$\begin{aligned}\mathcal{F}(\overrightarrow{U}^c) = &- Cp_f \left(\mathcal{F}(\overrightarrow{\hat{U}}) + \mathcal{F}(\overrightarrow{grad}(P)) \right) \\ &+ Cp_f \left(\frac{\mathcal{F}(\rho U^{c0})}{\Delta\tau_f} \right) \\ &- Cp_f \left(\frac{(eV)^p \mathcal{F}(\rho \overrightarrow{U^p}) + (eV)^q \mathcal{F}(\rho \overrightarrow{U^q})}{V_f^c} \right)\end{aligned} \tag{2.46}$$

$$\text{avec :} \qquad \mathcal{F}(\overrightarrow{A}) = \overrightarrow{A}_f \cdot \overrightarrow{S}_f \tag{2.47}$$

2.2.5.2 Algorithme de résolution

Les équations de quantité de mouvement et de pression discrétisées fournissent un système d'équations non-linéaires. Ce système est résolu par une approche découplée, de type **SIMPLE** (*Semi-Implicit Method for Pressure-Linked Equations*) [72, 53]. En voici les principales étapes :

① Initialisation des grandeurs Q^0 à $t = t^0$

② Incrémentation du pas de temps $t = t + \Delta t$

③ Début de la résolution non-linéaire

④ Calcul des grandeurs turbulentes si nécessaire à partir des quantités Q^0

⑤ Résolution de l'équation de quantité de mouvement 2.41 à partir de Q^0 pour obtenir une prédiction du champ de vitesse $U^\star$

⑥ Calcul des grandeurs intermédiaires $\overrightarrow{\hat{U}}$ à partir de $U^\star$

⑦ Résolution de l'équation de pression 2.45 à partir de $\overrightarrow{\hat{U}}$ pour obtenir un nouveau champ de pression P

⑧ Mise à jour des flux de vitesse selon 2.46 et correction du champ de vitesse U selon 2.42 pour tenir compte du nouveau champ de pression P et assurer la conservativité

⑨ Si la réduction des résidus non-linéaires n'est pas suffisante, mise à jour non-linéaire $U^0 \leftarrow U \quad P^0 \leftarrow P$ et retour à l'étape ④

⑩ Mise à jour instationnaire et retour à l'étape ②

Avec cette méthode, *le couplage vitesse-pression est réalisé en même temps que la résolution des non-linéarités* (étapes ③ à ⑨).

Un point important doit être précisé concernant la convergence du couplage vitesse-pression dans le cas de problèmes stationnaires. Pour de tels problèmes, la matrice diagonale formée des coefficients Cp_C (cf. équation 2.42) correspond à un inverse approché D^{-1} de la matrice discrète associée aux équations de quantités de mouvement $M = D + A$ (D est la partie diagonale de M). On utilise alors D^{-1} pour former l'équation de pression à partir de la contrainte d'incompressibilité. Il a été montré que cet inverse approché doit être suffisamment proche de l'inverse exact M^{-1} pour que la procédure de couplage soit convergente [102]. Et, bien entendu, cet inverse est d'autant plus proche de l'inverse exact que M est à dominance diagonale forte. Par conséquent, il faut assurer un certain taux de dominance diagonale pour la matrice M. Cela est effectué en fixant localement le pas de temps fictif $\Delta\tau$ pour renforcer le caractère de dominance diagonale du système linéaire associé à la discrétisation des équations de quantité de mouvement. Notons que cette contrainte sur le terme pseudo-instationnaire est responsable de la différence de vitesse de convergence entre les formulations découplées et fortement couplées.

2.2.6 Résolution des systèmes linéaires

Les systèmes linéaires, issus de l'équation de quantité de mouvement et de l'équation de pression, sont résolus par des *méthodes itératives*.

L'équation de quantité de mouvement est relativement aisée à résoudre, puisque la matrice du système présente une dominance diagonale forte, grâce à l'introduction du pas de temps fictif local. Par conséquent, une simple méthode de type **Gauss-Seidel** est utilisée. *Des conditions aux limites de type Dirichlet ou Neumann peuvent être employées pour la vitesse.*

La résolution du système de l'équation de pression est beaucoup plus délicate. En effet, l'opérateur correspondant est singulier par nature et la matrice du système souffre d'un mauvais conditionnement pour des maillages à fort étirement. Pour que le système puisse être résolu, un critère de compatibilité doit être vérifié, traduisant une condition d'incompressibilité globale sur l'ensemble du domaine de calcul. Les conditions aux limites fixées pour la vitesse doivent donc impérativement vérifier cette condition. Dans ce cas, le système de l'équation de pression admet une infinité de solutions identiques à une constante près. La résolution est effectuée soit par la méthode **GMRES avec un préconditionnement par une méthode de type LU incomplet**, soit par la méthode **PCGSTAB**.

Le processus de résolution a été parallélisé via **MPI** (cf. [70]). On peut, grâce au programme $M\epsilon TIS$ (cf. [67]), diviser le domaine de calcul en sous-espaces. Chaque processeur du parc d'ordinateurs, reliés en parallèle, se charge alors de la résolution sur un sous-espace.

2.3 Modélisation de la turbulence

Cette section présente *différents modèles utilisés pour évaluer le tenseur de Reynolds*. La modélisation a ici pour but d'exprimer les contraintes de Reynolds en fonction des champs moyens, de manière à fermer le système des équations de Navier-Stokes en moyenne de Reynolds, présenté plus haut (cf. § 2.2.1). On décrit les principaux modèles basés sur le concept de viscosité turbulente, à savoir le modèle de Spalart-Allmaras et ceux de Wilcox et Menter.

Tous les modèles implémentés dans le code **ISIS** peuvent être utilisés sous la forme "proche paroi", dans laquelle les équations de transport sont intégrées jusqu'à la paroi. Mis à part le modèle de Spalart-Allmaras, une résolution en mode fonction de paroi, sans contrainte sur la distance minimale du premier point de paroi, peut aussi être utilisée.

2.3.1 Modèles à viscosité turbulente

Par analogie avec le phénomène de diffusion moléculaire, les modèles, s'appuyant sur le concept de viscosité turbulente, introduisent un coefficient de viscosité caractérisé par une échelle de vitesse fluctuante v et une échelle de longueur des structures tourbillonnaires l :

$$\nu_t \sim vl \tag{2.48}$$

La modélisation des contraintes de Reynolds est alors assurée par **l'hypothèse de Boussinesq**, qui relie le tenseur de Reynolds au tenseur des taux de déformation :

$$-\overline{u_i' u_j'} = 2\nu_t S_{ij} - \frac{2}{3} k \delta_{ij} \tag{2.49}$$

où k représente l'énergie cinétique turbulente et δ_{ij} le symbole de Kronecker.

On détermine le coefficient de viscosité turbulente ν_t en résolvant zéro, une ou deux équations de transport, permettant d'évaluer les échelles v et l. Les modèles à zéro équation, pour lesquels les échelles sont données par des relations algébriques, ne seront pas utilisés à cause de leur domaine d'application très limité.

2.3.1.1 Modèle de Spalart-Allmaras

Le modèle de Spalart-Allmaras [93] est un *modèle à une équation*, qui définit la viscosité turbulente par l'intermédiaire d'une variable de viscosité auxiliaire $\tilde{\nu}$ et d'une fonction auxiliaire f_{v1} :

$$\nu_t = \tilde{\nu} f_{v1} \tag{2.50}$$

On résout l'équation de transport de la quantité $\tilde{\nu}$:

$$\begin{aligned}\frac{\partial \rho\tilde{\nu}}{\partial t} + \frac{\partial}{\partial x_j}\Big(\rho\tilde{\nu}U_j\Big) &= c_{b1}(1-f_{v1})\rho\tilde{S}\tilde{\nu} \\ &+ \frac{1}{\sigma}\Big[\frac{\partial}{\partial x_j}\Big(\rho(\nu+\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\Big) + c_{b2}\rho\frac{\partial\tilde{\nu}}{\partial x_j}\frac{\partial\tilde{\nu}}{\partial x_j}\Big] \\ &- \Big[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\Big]\rho\Big[\frac{\tilde{\nu}}{d}\Big]^2 + f_{t1}\rho\Delta U^2\end{aligned} \tag{2.51}$$

Les termes au second membre représentent respectivement, de gauche à droite, les termes de production, de diffusion, de dissipation et de transition. Les coefficients et les fonctions auxiliaires, intervenant dans l'équation de transport, sont déterminés à partir de modèles de base, pour des écoulements cisaillés, de couche limite et avec transition. Les coefficients sont :

$$c_{b1} = 0.1355 \qquad c_{b2} = 0.622 \qquad \sigma = \frac{2}{3} \tag{2.52}$$

$$\begin{aligned}&c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma} \qquad c_{w2} = 0.3 \qquad c_{w3} = 2 \\ &r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \qquad g = r + c_{w2}(r^6 - r) \qquad f_w = g\Big[\frac{1+c_{w3}^6}{g^3+c_{w3}^6}\Big]^{1/6}\end{aligned} \tag{2.53}$$

Les fonctions auxiliaires intervenant pour des écoulements "proche paroi" s'écrivent :

$$\begin{aligned}&\tilde{S} = S + \frac{\tilde{\nu}}{(\kappa d)^2}f_{v2} \qquad S = \sqrt{2\Omega_{ij}\Omega_{ij}} \qquad \chi = \frac{\tilde{\nu}}{\nu} \\ &f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \qquad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}} \qquad c_{v1} = 7.1\end{aligned} \tag{2.54}$$

Pour les zones à écoulement cisaillé et transitionnel, les fonctions auxiliaires sont :

$$\begin{aligned}&f_{t1} = c_{t1}g_t.\exp\Big(-c_{t2}\frac{\omega_t^2}{\Delta U^2}\big[d^2 + (g_t d_t)^2\big]\Big) \qquad f_{t2} = c_{t3}g_t.\exp\big(-c_{t3}\chi^2\big) \\ &g_t = \min\Big\{0.1, \frac{\Delta U}{\omega_t \Delta x_t}\Big\} \qquad c_{t1} = 1 \qquad c_{t2} = 2 \qquad c_{t3} = 1.2 \qquad c_{t4} = 0.5\end{aligned} \tag{2.55}$$

ω_t représente la vorticité, ΔU la norme de l'écart de vitesse à la transition, Δx_t la taille de la maille le long de la paroi au point de transition et d la distance à la paroi.

Le modèle de Spalart-Allmaras permet une amélioration notable des modèles algébriques, tout en fournissant une alternative plus simple que les modèles à deux équations. Néanmoins, son domaine d'application est encore limité, et les modèles à deux équations, décrits plus loin, sont à l'heure actuelle les plus utilisés en sciences de l'ingénieur.

2.3.1.2 Modèle $k-\omega$ de Wilcox

Le modèle de Wilcox [106, 107] est un *modèle à deux équations de transport*, où sont considérées l'énergie cinétique turbulente k, et la quantité ω représentant une fréquence caractéristique de la turbulence :

$$\mu_t = \frac{\rho k}{\omega} \tag{2.56}$$

Ces deux quantités vérifient les équations de transport :

$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_j}\Big(\rho U_j k - (\mu + \sigma^* \mu_t)\frac{\partial k}{\partial x_j}\Big) = \tau_{t_{ij}} S_{ij} - \beta^* \rho \omega k \tag{2.57a}$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial}{\partial x_j}\Big(\rho U_j \omega - (\mu + \sigma \mu_t)\frac{\partial \omega}{\partial x_j}\Big) = \alpha \frac{\omega}{k} \tau_{t_{ij}} S_{ij} - \beta \rho \omega^2 \tag{2.57b}$$

$\tau_{t_{ij}}$ est le tenseur de Reynolds. Les constantes du modèle suggérées par Wilcox sont :

$$\begin{array}{lll} \alpha = \dfrac{5}{9} & \beta = \dfrac{3}{40} & \beta^* = \dfrac{9}{100} \\ \sigma = 0.5 & \sigma^* = 0.5 & \end{array} \tag{2.58}$$

On détermine une condition aux limites pour ω grâce à une analyse asymptotique à proximité de la paroi. Notons Δy la distance entre le premier point et la paroi. Les conditions s'écrivent alors :

$$k = 0 \quad \text{et} \quad \omega = 10\frac{6\mu}{\beta \rho (\Delta y)^2} \tag{2.59}$$

En ce qui concerne les conditions aux limites externes, Wilcox préconise les relations :

$$\omega_\infty = \lambda \frac{U_{ref}}{L_{ref}} \qquad \mu_{t_\infty} = 10^{-3}\mu \qquad k_\infty = \frac{\mu_{t_\infty}\omega_\infty}{\rho} \tag{2.60}$$

L_{ref} et U_{ref} étant des grandeurs de référence, λ pouvant varier entre 1 et 10.

Ce modèle présente des avantages par rapport aux modèles $k-\epsilon$, très utilisés. La simplicité de sa formulation dans la sous-couche visqueuse et le découplage des variables k et ω à proximité de la paroi le rendent plus robuste numériquement. De très bons résultats ont ainsi été rapportés pour des écoulements à fort décollement (cf. [46]). Le point faible de ce modèle reste la condition externe arbitraire pour ω et surtout la sensibilité de la solution vis-à-vis de cette condition.

2.3.1.3 Modèle $k-\omega$ SST de Menter

Pour le modèle $k-\omega$ **BSL** (*BaSeLine*) [65], Menter propose de modifier le modèle de Wilcox, en retenant son expression à proximité de la paroi et en adoptant le comportement d'un modèle $k-\epsilon$ au loin, tout en gardant une formulation avec les variables k et ω. La transition est assurée par une fonction modifiant les coefficients du modèle.

Le modèle $k-\omega$ **SST** [66] (*Shear Stress Transport*) reprend la formulation précédente. De plus, la viscosité turbulente utilisée fait intervenir le transport du cisaillement turbulent :

$$\mu_t = \frac{\frac{\rho k}{\omega}}{\max\{1, \frac{\Omega F_2}{a_1 \omega}\}} \qquad \text{avec} \qquad a_1 = 0.31 \tag{2.61}$$

où Ω représente la vorticité. La fonction auxiliaire F_2 est définie à partir de la distance à la paroi d :

$$F_2 = \tanh\Big(\Big[\max\Big\{2\frac{\sqrt{k}}{0.09 d\omega}, \frac{500\mu}{\rho d^2 \omega}\Big\}\Big]^2\Big) \tag{2.62}$$

Les équations de transport incluent une fonction auxiliaire F_1. Elle permet la transition entre le modèle original et le modèle basé sur la variable ϵ :

$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_j}\Big(\rho U_j k - (\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\Big) = \tau_{t_{ij}} S_{ij} - \beta^* \rho \omega k \tag{2.63a}$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial}{\partial x_j}\Big(\rho U_j \omega - (\mu + \sigma_\omega \mu_t)\frac{\partial \omega}{\partial x_j}\Big) = \gamma \rho \Omega^2 - \beta \rho \omega^2 + 2(1-F_1)\frac{\rho \sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{2.63b}$$

Le rôle de la fonction auxiliaire est de faire varier les coefficients du modèle original, dans la couche limite, vers ceux du modèle $k-\epsilon$ transformé, dans les zones de cisaillement et les zones externes. Elle est définie par :

$$\begin{aligned} F_1 &= \tanh\left(\left[\min\left\{\max\left\{\frac{\sqrt{k}}{0.09 d\omega}, \frac{500\mu}{\rho d^2 \omega}\right\}, \frac{4\rho \sigma_{\omega 2} k}{CD_{k\omega} d^2}\right\}\right]^4\right) \\ \text{avec} \qquad CD_{k\omega} &= \max\left\{\frac{2\rho\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-20}\right\} \end{aligned} \tag{2.64}$$

Les constantes utilisées sont :

$$a_1 = 0.31 \qquad \beta^* = 0.09 \qquad \kappa = 0.41 \tag{2.65}$$

Les coefficients β, γ, σ_k, et σ_ω sont alors définis, comme suit, à partir des coefficients du modèle original, noté 1, et de ceux du modèle $K-\epsilon$ transformé, noté 2 :

$$\phi = F_1 \phi_1 + (1-F_1)\phi_2 \qquad \text{avec} \qquad \phi = \{\beta, \gamma, \sigma_k, \sigma_\omega\} \tag{2.66}$$

Les constantes de chaque modèle sont :

➢ *modèle intérieur* :

$$\begin{aligned} &\sigma_{k1} = 0.85 \qquad \sigma_{\omega 1} = 0.500 \qquad \beta_1 = 0.0750 \\ &\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} = 0.553 \end{aligned} \tag{2.67}$$

➢ *modèle extérieur* :

$$\begin{aligned} &\sigma_{k2} = 1.00 \qquad \sigma_{\omega 2} = 0.856 \qquad \beta_2 = 0.0828 \\ &\gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} = 0.440 \end{aligned} \tag{2.68}$$

Les conditions aux limites sont les mêmes que celles du modèle de Wilcox.

2.4 Conclusion

Ce chapitre a présenté la théorie de base sur laquelle repose **ISIS**. En résumé, c'est un solveur fluide, qui traite des problèmes d'écoulements visqueux turbulents incompressibles à haut nombre de Reynolds. Il permet, grâce à sa formulation Volumes-Finis, de prendre en compte des géométries très complexes. Un soin tout particulier a été porté à la turbulence. De nombreux modèles sont disponibles. Ce code est évidemment parallélisé pour pouvoir venir à bout des problèmes les plus lourds.

ISIS n'étant pas le sujet principal de ce travail, de nombreux aspects ont été occultés : comme par exemple en turbulence, la possibilité d'utiliser une fermeture du second ordre ou encore les approches LES et DES (cf. [23]). **ISIS** est aussi un code multi-phasique. Ainsi, une thèse sur la cavitation a été réalisée (cf. [27]). Enfin, le raffinement de maillage est un des axes actuels de développement (cf. [47]).

Chapitre 3

Solveur de dynamique des structures type poutre : PTAH

L'interaction fluide-structure numérique demande deux codes pour fonctionner : un qui résout les équations discrétisées fluides ; dans notre cas, c'est ISIS, dont nous venons de voir les caractéristiques en détail. Et l'autre qui s'occupe de *calculer les déformations structurelles*. C'est l'objet de ce chapitre. Le cadre de l'interaction fluide-structure a été réduit aux *structures élancées*. Le solveur structure développé est donc un **solveur poutre**.

Pourquoi s'être limité aux structures élancées ? Le laboratoire étant orienté vers l'hydrodynamique, un des principaux problèmes d'interaction fluide-structure rencontré dans ce domaine est celui des risers. Ce sont de gros et très longs tubes plongés dans l'eau. En mer, ces risers subissent de fortes contraintes dues aux courants et parfois se rompent. Des prévisions numériques peuvent être très utiles pour éviter ces dommages. Évidemment le champ d'application final ne sera pas si restreint : un long bateau comme un tanker peut être sous certaines conditions modélisé par une poutre équivalente. Leurs déformations par la houle et les vagues peuvent donc être prises en compte.

Le présent chapitre va examiner les hypothèses et approximations faites au sein de ce solveur poutre qui pour traiter des problèmes comme ceux des risers doit pouvoir prendre en compte les grands déplacements. Puis l'intégration numérique de ce solveur poutre est abordée. Enfin sa validation par comparaison avec des cas-tests théoriques et expérimentaux est présentée.

3.1 Arrière-plan théorique

3.1.1 Présentation, modélisation et hypothèses

La gestion des corps déformables (à déformation *imposée*) et en mouvement a déjà été intégrée à ISIS (cf. [57]). De plus des corps de type poutre avaient déjà été utilisés (*ROBEA/RAAMO*). Cependant, la mise en place du solveur poutre a demandé de nouvelles données sur les corps élancés qui ont dû être codées.

La théorie relative à *ce nouveau solveur structure se base sur les travaux de F. Boyer et de D. Primault* ([16], [17] et [79]). Ces derniers ont développé cette théorie dans le

cadre d'applications en robotique : citons par exemple la prédiction du mouvement des câbles reliant deux parties articulées d'un robot et transportant de l'énergie électrique ou hydraulique.

En robotique la majeure partie des programmes utilise des modèles géométriques (cf. [2] et [1]) ou encore des modèles transmettant l'effort axial seulement (cf. [52]). Ces anciens modèles sont trop simples et ne fonctionnent pas dans tous les cas de figure (cf. [19]). Ainsi les années 90 ont vu naître les programmes de simulation de câble basés sur le modèle des poutres fines. Deux possibilités s'offrent alors pour la modélisation des poutres fines en grandes déformations :

La première traditionnellement utilisée en mécanique des milieux continus tridimensionnels consiste à *procéder aux approximations, conséquences de la faiblesse des rapports des deux dimensions transverses sur la troisième* (cf. [45] et [7]). Les équations d'équilibre sont écrites par intégration des équations tridimensionnelles sur les deux dimensions transverses. Il est alors nécessaire d'intégrer sur l'épaisseur les lois constitutives tridimensionnelles faisant jouer des tenseurs tridimensionnels tels que le tenseur des déformations de Green-Lagrange et celui des contraintes de Piola-Kirchoff pour résoudre le problème en grandes transformations. Pour étayer ces propos, on peut citer les solutions numériques basées sur cette approche telles celles de Bathe & Boulourchi (cf. [8]) consacrées aux poutres de Timoshenko ainsi que celles de Tadjbakhsh & Lagoudas (cf. [97]) et Behdinan & al. (cf. [12]) traitant des poutres d'Euler-Bernoulli.

Le deuxième choix, au contraire du premier, *considère la poutre comme un milieu monodimensionnel réalisé par l'empilement continu de micro-solides rigides*, c'est l'approche directe initiée par Eugène et François **Cosserat** dans leur "Théorie des corps déformables" (cf. [26]).

Ces deux approches sont très différentes dans leur conception et ne sont évidemment pas équivalentes, même si les équations d'équilibre qui en dérivent sont proches.

Le *solveur poutre codé ici se base sur le deuxième choix*. Une modélisation de la poutre type Cosserat est donc envisagée. Notons tout de suite que ce choix impose une première contrainte : la *planéité et la rigidité des sections*. Si aucune autre hypothèse n'est formulée, la cinématique est dite de **Timoshenko Reissner** (cf. [105] et [80]). Dans notre cas, les poutres rencontrées sont dites *fines ou élancées*, car une des trois dimensions est très grande par rapport aux deux autres. On peut alors faire l'*hypothèse d'orthogonalité des sections par rapport à la ligne neutre* : c'est l'**hypothèse de Kirchoff**. La cinématique qui en découle porte aussi ce nom. Deux modèles de poutre en résultent : le **modèle de Rayleigh** et le **modèle d'Euler-Bernoulli** (cf. [64]). Ce dernier suppose en plus que l'énergie cinétique angulaire des sections est négligeable. Cette dernière supposition amène de grandes simplifications dans les équations analytiques, mais pose aussi un problème dans le cas d'une poutre libre de tourner sur elle-même. Cette cinématique est donc à écarter dans le cas de problèmes 3D. Dans le cas 2D, notre module de résolution des déformations des poutres fines peut choisir de se baser sur l'un ou sur l'autre des modèles.

La cinématique des travaux de F. Boyer et D. Primault, détaillée dans § 3.1.2, permet de traiter correctement les problèmes de câbles grâce en particulier aux soins apportés au couplage flexion-torsion. En effet, la solution numérique se base sur une formulation variationnelle de la cinématique de Kirchoff et sur une modélisation exacte de la non-linéarité géométrique la plus complexe : le **couplage flexion-torsion** (cf. [4]). Une notion impor-

tante de ce travail vient d'être évoquée : la **modélisation géométriquement exacte**. On entend par cette expression que *les approximations ne sont faites qu'à la toute fin des développements*. Ce terme fut introduit par Simo dans le cadre de ses travaux sur les poutres de Reissner (cf. [86]) et les coques (cf. [87], [88] et [89]) pour traduire le caractère exact de sa modélisation. Rappelons ici la définition telle que l'énonce son auteur dans [89] : "Une fois la cinématique de base admise, il existe deux niveaux d'approximations dans la résolution numérique d'un problème de dynamique des structures. Le premier concerne la géométrie de la poutre et les équations d'équilibre qui régissent son évolution, le second est relatif à la résolution numérique proprement dite. Dans l'approche géométriquement exacte, acceptant les hypothèses cinématiques qui définissent les transformations admissibles, la géométrie de la poutre ainsi que ses équations d'équilibre sont traitées de manière exacte. Ainsi les seules approximations utilisées concernent le second niveau, c'est-à-dire la discrétisation spatiale et l'intégration numérique temporelle." La modélisation géométriquement exacte permet une meilleure convergence du code. Cette cinématique, tout comme celle d'Euler-Bernoulli ou celle de Reissner, a besoin d'un paramétrage. Celui de la rotation est le plus complexe. Il est détaillé dans § 3.1.3.

L'état de l'art étant fait, les modélisations et hypothèses présentées, les travaux de F. Boyer et D. Primault peuvent être développés plus en détail : les deux prochaines sections exposent la cinématique des poutres fines et la paramétrisation du champ de rotation. Viennent ensuite la présentation du modèle d'Euler-Bernoulli (§ 3.1.5 et § 3.1.6) et celle du modèle de Rayleigh (§ 3.1.7). Enfin, le problème des mouvements imposés aux extrémités de la poutre est abordé (§ 3.1.9).

3.1.2 Cinématique de base des poutres fines

En premier lieu définissons les bases orthonormées qui vont nous être utiles :

- ➢ une **base spatiale** $(e_i)_{i=1,2,3}$ fixée à l'espace dans lequel est plongée la poutre (confondue avec le repère cartésien (x, y, z))
- ➢ une **base matérielle** $(E_i)_{i=1,2,3}$ attachée à la configuration de base de la poutre Σ_0 (les coordonnées dans cette base sont notées (X, Y, Z))
- ➢ une **base mobile** $(t_i(X))_{i=1,2,3}$ attachée à la configuration déformée de la poutre Σ_t (son origine est sur la ligne neutre de la poutre en X et t_1 dirigé par la tangente à la ligne neutre)

Plusieurs remarques immédiates :

- Les flèches au dessus des vecteurs ont été occultées dans la partie théorique de ce chapitre, afin d'alléger l'écriture.
- Les *variables spatiales sont écrites en minuscule*, tandis que *les variables matérielles sont en majuscule*.
- À $t = 0$, la base $(t_i(X))_{i=1,2,3}$ est confondue avec $(E_i)_{i=1,2,3}$.
- Dans ce chapitre, pour des raisons de clarté, la base spatiale $(e_i)_{i=1,2,3}$ est prise confondue avec la base matérielle $(E_i)_{i=1,2,3}$. Ce n'est pas général et dans le code IFS développé au cours de cette thèse, $(e_i)_{i=1,2,3}$ et $(E_i)_{i=1,2,3}$ ne sont pas obligatoirement identiques.

On adopte comme modèle de poutre fine celui d'Euler-Bernoulli. Ainsi on fait l'hypothèse que chaque section de la poutre reste orthogonale à la ligne neutre (hypothèse de Kirchoff). Rappelons que dans le cadre du modèle les sections sont aussi rigides et planes. Par conséquent, toute configuration du câble peut être décomposée en (cf. fig. 3.1) :

➢ un **champ de position** (celui des points de la ligne neutre de la poutre) :

$$\forall X \in [0, L] \mapsto r(X) = r^i(X) e_i \in \mathbb{R}^3 \tag{3.1}$$

➢ un **champ de rotation** modifiant les sections de la configuration initiale en celles de la déformée :

$$\forall X \in [0, L] \mapsto R(X) = \frac{\partial_x r}{\|\partial_x r\|} \otimes E_1 + t_\alpha(X) \otimes E_\alpha \in SO3 \tag{3.2}$$

avec $\alpha \in 2, 3$

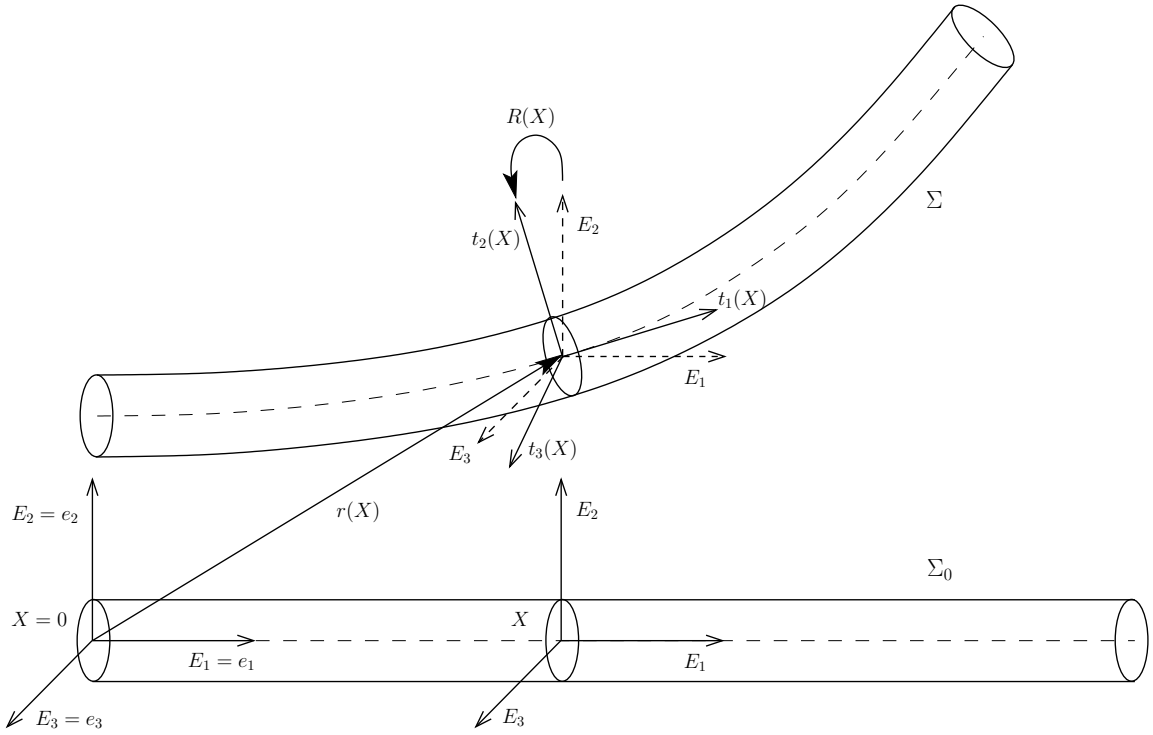


FIG. 3.1: Cinématique de base

Dans le cas de la cinématique de Reissner, la rotation est définie dans le groupe de Lie $SO3$. L'hypothèse de Kirchoff implique que le champ de rotation n'est pas défini par trois vecteurs orthonormés, mais par deux. On va donc pouvoir restreindre $SO3$ à l'espace $S \times SO2$, où S est un sous-ensemble de $SO3$ de dimension 2, paramétré par le champ de position de la ligne de référence, et $SO2$ est le groupe des rotations planes. Pour réaliser cela, le champ de rotation 3.2 est remplacé par la composition de deux rotations : *une correspondant au mouvement de la ligne neutre et l'autre de la matière tournant autour de cette ligne centrale* (cf. fig. 3.2). Après calcul, on montre que le premier champ de rotation (celui correspondant au seul mouvement de la ligne neutre) peut se mettre sous la forme d'une exponentielle de matrice (formule d'Olinde-Rodriguez) :

$$\begin{aligned}\Lambda_{E_1}(t_1(X)) &= exp(\alpha\hat{V}) \\ &= (t_1 \cdot E_1)I_3 + \frac{1}{1+t_1 \cdot E_1}(E_1 \wedge t_1) \otimes (E_1 \wedge t_1) + (E_1 \wedge t_1)^{\hat{}}\end{aligned} \tag{3.3}$$

où

- ➢ t_1 est le vecteur unitaire tangent à la ligne de référence après déformation : $X \mapsto t_1(X) = \frac{\partial_x r}{\|\partial_x r\|}$.
- ➢ V est un vecteur unitaire qui est laissé invariant par une unique rotation d'angle α et qui transforme E_1 en t_1 ($\cos(\alpha) = t_1 \cdot E_1$ et $\sin(\alpha)\, V = E_1 \wedge t_1$).
- ➢ Enfin, $\hat{V}$ (noté aussi $V^{\hat{}}$ est la matrice antisymétrique construite à partir du vecteur V, et définie telle que $\forall x \in \mathbb{R}^3 : V \wedge x = \hat{V}\, x$.
  Par exemple, pour $V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$, on a $\hat{V} = \begin{bmatrix} 0 & -V_3 & V_2 \\ V_3 & 0 & -V_1 \\ -V_2 & V_1 & 0 \end{bmatrix}$.

Quelques Remarques :

- La formule 3.3 est non définie pour $t_1 = -E_1$
- La formule 3.3 est définie sur S. Ce sous-ensemble est paramétré par le champ de tangente unitaire $t_1(X)$, qui est lui-même paramétré par le champ de position 3.1.

La *configuration en espace de la poutre fine de longueur L est alors identifiable à* $\{\varphi : [0, L] \to \mathbb{R}^3 \times SO2\}$, puisqu'on peut écrire le champ de rotation 3.2 sous la forme :

$$\boxed{R = \Lambda_{E_1}(t_1)\, R_{d\tau} = R_{g\tau}\, \Lambda_{E_1}(t_1)} \tag{3.4}$$

où $R_{d\tau}$ et $R_{g\tau}$ sont respectivement les rotations des sections autour de la ligne de référence non déformée et déformée. Ces rotations peuvent être elles-mêmes paramétrées de différentes manières. Nous les détaillerons dans la section suivante (§ 3.1.3).

3.1.3 Paramétrisation du champ de rotation

Le paramétrage de la rotation se fait par l'intermédiaire d'un angle qui permet de décrire l'orientation de chaque section. Par abus de langage nous donnerons à cet angle le titre d'**angle "de torsion"** (il se confond avec le vrai angle de torsion par exemple lorsque l'on est en torsion pure). Ce paramètre "angle de torsion" peut être défini de trois manières différentes et amène trois paramétrages différents :

- ➢ La **description lagrangienne totale** où le champ d'orientation se base sur la configuration de référence.
- ➢ La **description lagrangienne réactualisée** où le champ d'orientation se base sur la configuration de l'instant précédent.
- ➢ Enfin la **description eulérienne** dans laquelle le champ d'orientation n'est pas explicitement paramétré comme précédemment.

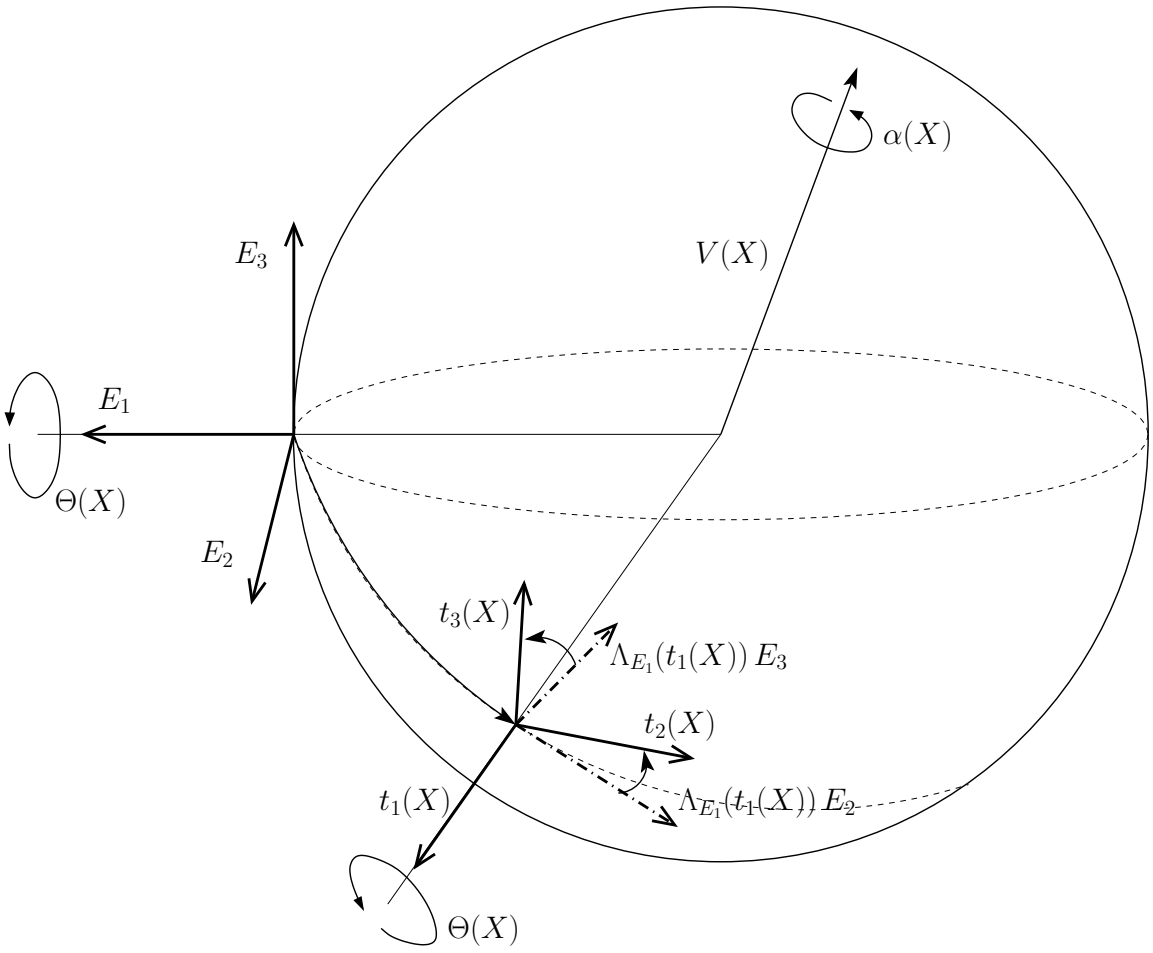


FIG. 3.2: La transformation finie compatible avec la cinématique de base

3.1.3.1 Description lagrangienne totale

On prend comme référence la configuration initiale. Le paramétrage dépend du "chemin" que l'on emprunte de celle-ci à la configuration déformée, car *le groupe SO3 est non commutatif*. On a ainsi (cf. fig. 3.2) :

$$R = R_{g\tau}\,\Lambda_{E_1}(t_1) = exp(\Theta\hat{t_1})\,\Lambda_{E_1}(t_1) = \Lambda_{E_1}(t_1)\,R_{d\tau} = \Lambda_{E_1}(t_1)\,exp(\Theta\hat{E}_1) \tag{3.5}$$

Dans l'équation 3.5, Θ est l'angle "de torsion", et plus précisément il est **"l'angle de torsion total"** par analogie à la terminologie de Reissner (cf. [3]).

3.1.3.2 Description lagrangienne réactualisée

Dans la description lagrangienne réactualisée, la configuration de référence est la configuration d'équilibre à l'instant précédent $t = t_n$. Comme précédemment le paramétrage dépend du "chemin" choisi :

- **Paramétrage spatial** (à gauche) : $R = exp(\vartheta\hat{t_1})\,\Lambda_{R_n\,E_1}(t_1)\,R_n$
- **Paramétrage matériel** (à droite) : $R = R_n\,\Lambda_{E_1}(R_n^T\,t_1)\,exp(\vartheta\hat{E}_1)$

L'angle de torsion ϑ est dit par analogie à la terminologie de Reissner **"angle de torsion actuel"** (cf. [3]).

3.1.3.3 Description eulérienne

La description eulérienne diffère des deux précédentes. En effet, le champ de rotation autour de la ligne de référence est directement décrit par le champ d'opérateur $R_{g\tau}$ ou $R_{d\tau}$. Il n'est pas paramétré. L'espace des configurations de la poutre n'est alors plus $\{\varphi : [0, L] \to \mathbb{R}^3 \times \mathbb{R}\}$ mais $\{\varphi : [0, L] \to \mathbb{R}^3 \times SO2\}$. Nous allons donc devoir trouver un moyen de se déplacer sur cet espace si nous voulons utiliser cette description eulérienne. Ce problème est résolu dans la section qui suit (§ 3.1.4).

3.1.4 Mise à jour de la configuration de la poutre

Pour mettre à jour la poutre au cours du temps, il faut trouver les transformations finies qui, compatibles avec la cinématique de base présentée précédemment, permettent de passer d'une configuration à une autre : c'est-à-dire comment passer d'une configuration Σ_t à l'instant t à $\Sigma_{t+\Delta t}$ (Δt est l'incrément temporel fini) en utilisant des transformations finies sur la configuration de référence (Σ_0 dans le cas lagrangien total ou Σ_t dans le cas lagrangien réactualisé). Puis on passe des transformations finies aux transformations infinitésimales en faisant tendre l'incrément de temps Δt vers 0. Ces dernières sont en effet essentielles pour le calcul des variations des fonctionnelles rencontrées à la section 3.1.6.

3.1.4.1 Réactualisation du champ de position

La mise à jour du champ de position est très simple et se passe d'explication :

$$r_{t+\Delta t}(X) = r_t(X) + \Delta r_t(X) \tag{3.6}$$

Lorsque $\Delta t \rightarrow 0$, on a :

$$r_\epsilon = r + \epsilon \delta r \tag{3.7}$$

où le temps t a été remplacé par le paramètre réel arbitraire ϵ très petit (indépendant de t et de X).

3.1.4.2 Réactualisation du champ de rotation

Pour le champ de rotation, c'est plus complexe. Cela dépend en effet du paramétrage choisi et du "chemin" que l'on suit lorsqu'on réactualise la rotation (à gauche et à droite).

➢ Pour l'approche lagrangienne totale, les deux formules (à gauche et à droite) sont les suivantes :

$$R_{t+\Delta t} = exp((\Theta + \Delta\Theta)(t_1 + \Delta t_1)\hat{}\,)\,\Lambda_{E_1}(t_1 + \Delta t_1) \tag{3.8}$$

$$R_{t+\Delta t} = \Lambda_{E_1}(t_1 + \Delta t_1)\, exp((\Theta + \Delta\Theta)\hat{E}_1) \tag{3.9}$$

où $\Delta\Theta$ est l'incrément de l'angle de torsion total et Δt_1 celui de la tangente unitaire. Lorsque $\Delta t \rightarrow 0$, on trouve R_ϵ en remplaçant Θ par sa perturbation $\Theta_\epsilon = \Theta + \epsilon\delta\Theta$:

$$R_\epsilon = exp(\Theta_\epsilon \hat{t_{1,\epsilon}})\,\Lambda_{E_1}(t_{1,\epsilon}) \tag{3.10}$$

$$R_\epsilon = \Lambda_{E_1}(t_{1,\epsilon})\, exp(\Theta_\epsilon \hat{E}_1) \tag{3.11}$$

où $t_{1,\epsilon}$ est la perturbation de la tangente unitaire :

$$t_{1,\epsilon} = t_1 + \epsilon\delta t_1 = t_1 + \epsilon\delta(\frac{r'}{\|r'\|}) \tag{3.12}$$

➢ Pour l'approche lagrangienne réactualisée, on a :

$$R_{t+\Delta t} = exp((\vartheta + \Delta\vartheta)(t_1 + \Delta t_1)\hat{}\,)\,\Lambda_{R_n\, E_1}(t_1 + \Delta t_1)\, R_n \tag{3.13}$$

$$R_{t+\Delta t} = R_n\, \Lambda_{E_1}(R_n^T\,(t_1 + \Delta t_1))\, exp((\vartheta + \Delta\vartheta)\hat{E}_1) \tag{3.14}$$

où $\Delta\vartheta$ est l'incrément de l'angle de torsion actuel.
La transformation infinitésimale pour cette approche fait intervenir la perturbation de l'angle de torsion actuel $\vartheta_\epsilon = \vartheta + \epsilon\delta\vartheta$:

$$R_\epsilon = exp(\vartheta_\epsilon \hat{t_{1,\epsilon}})\,\Lambda_{R_n\, E_1}(t_{1,\epsilon})\, R_n \tag{3.15}$$

$$R_\epsilon = R_n\, \Lambda_{E_1}(R_n^T\, t_{1,\epsilon})\, exp(\vartheta_\epsilon \hat{E}_1) \tag{3.16}$$

➢ Et enfin, les réactualisations pour l'approche eulérienne (illustrées sur la figure 3.3) :

$$R_{t+\Delta t} = exp(\Delta\theta(t_1 + \Delta t_1)\hat{}\,)\,\Lambda_{t_1}(t_1 + \Delta t_1)\, R_t \tag{3.17}$$

$$R_{t+\Delta t} = R_t\,\Lambda_{E_1}(E_1 + R_n^T\,\Delta t_1)\,exp(\Delta\theta\hat{E}_1) \tag{3.18}$$

où $\Delta\theta$ est un **"angle de torsion itératif"** par analogie à la terminologie trouvée dans [3]. *La somme de tous les incréments de ce dernier n'a pas de sens et ne permet pas de reconstituer le mouvement de la poutre.* En effet, il est l'équivalent discret d'une vitesse angulaire matérielle de torsion non intégrable (non-holonome). Mais par contre *il mène aux algorithmes numériques les plus simples.* C'est pourquoi, par la suite, on présentera les résultats sous l'approche eulérienne. Les calculs en lagrangien total ou en lagrangien réactualisé peuvent être trouvés dans la thèse de D. Primault [79].
Avec cette approche, R_ϵ s'écrit :

$$\begin{aligned} R_\epsilon &= exp(\epsilon\delta\hat{\phi})\,R = exp(\epsilon\delta\theta\hat{t_{1,\epsilon}})\,\Lambda_{t_1}(t_{1,\epsilon})\,R \\ &= R\,exp(\epsilon\hat{\psi}) = R\,\Lambda_{E_1}(E_{1,\epsilon})\,exp(\epsilon\delta\theta\hat{E}_1) \end{aligned} \tag{3.19}$$

où $\delta\phi$ est le champ de variation spatial du champ de rotation de la poutre sur $SO3$ et $\delta\psi$ son homologue matériel. La relation entre ces deux champs est la suivante :

$$\delta\phi = \delta\theta t_1 + t_1 \wedge \delta t_1 = R\,\delta\psi = R\,\left(\delta\theta E_1 + E_1 \wedge \left(R^T\,\delta t_1\right)\right) \tag{3.20}$$

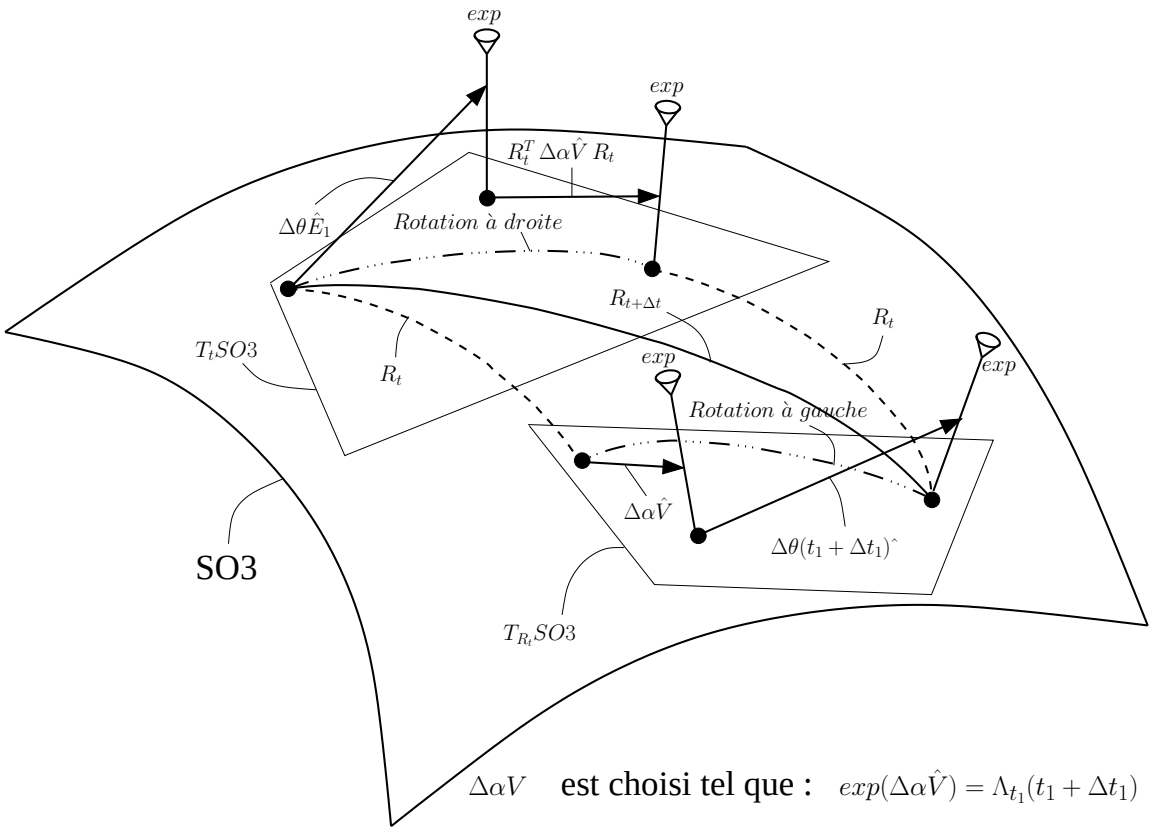


FIG. 3.3: Mise à jour sur SO3 de l'angle itératif θ

3.1.5 Formulation variationnelle faible du modèle d'Euler-Bernoulli

Rappelons que nous considérons le cas d'une poutre mince de section constante A, de moment quadratique axial $I_a = \iint Y^2\, dY\, dZ = \iint Z^2\, dY\, dZ$ et de moment quadratique polaire $I_p = \iint \left(Y^2 + Z^2\right) dY\, dZ = 2I_a$, en petite déformation et grand déplacement. Le matériau qui la constitue est supposé élastique de module d'Young E, de module de cisaillement G et de masse volumique ρ. De plus, l'énergie cinétique de rotation est négligée, nous pouvons alors écrire le **lagrangien d'une poutre d'Euler-Bernoulli** :

$$\begin{aligned} L = & \underbrace{\frac{1}{2}\int_0^L \rho A \left(\frac{\partial r}{\partial t}\right)^2 dX}_{\text{Énergie cinétique de translation}} - \underbrace{\frac{1}{2}\int_0^L EI_a \left(\frac{\partial t_1}{\partial X}\right)^2 dX}_{\text{Énergie de déformation de flexion}} - \\ & \underbrace{\frac{1}{2}\int_0^L GI_p K_1^2 dX}_{\text{Énergie de déformation de torsion}} \underbrace{- \frac{1}{2}\int_0^L EA \left(\frac{\partial r}{\partial X} - t_1\right)^2 dX}_{\text{Énergie de traction-compression}} \end{aligned} \tag{3.21}$$

où K_1 est la première composante de la torsion d'après D. Primault. C'est aussi le premier terme de la courbure de l'espace spatial k (K est la courbure de l'espace matériel). On peut en effet écrire :

$$k = K_1 t_1 + t_1 \wedge t_1{}' = R\left(K_1 E_1 + E_1 \wedge \left(R^T t_1{}'\right)\right) = R\,K \tag{3.22}$$

Pour obtenir la forme faible d'une poutre libre d'Euler-Bernoulli, nous allons à présent appliquer le **principe d'Hamilton** :

$$\delta \int_{t_1}^{t_2} L\, dt = 0, \qquad \forall \delta r, \forall \delta\theta \tag{3.23}$$

Après calculs *en approche eulérienne* :

$$\delta \int_{t_1}^{t_2} L\, dt = -\int_{t_1}^{t_2} \left(\underbrace{\int_0^L \begin{bmatrix} \delta r \\ \delta r{}' \\ \delta\theta \end{bmatrix} f_{dyn}\, dX}_{\text{Composante dynamique}} + \underbrace{\int_0^L \begin{bmatrix} \delta r{}' \\ \delta r{}'' \\ \delta\theta{}' \end{bmatrix} f_{stat}\, dX}_{\text{Composante statique}} \right) dt \tag{3.24}$$

avec les **champs vectoriels des forces d'inertie et internes** suivants :

$$f_{dyn} = \begin{bmatrix} \rho A \ddot{r} \\ 0_{3\times 1} \\ 0 \end{bmatrix} \tag{3.25}$$

$$f_{stat} = \begin{bmatrix} EA\left(1 - \frac{1}{\|r'\|}\right) r' + EI_a \left(2\frac{(r'\cdot r'')^2}{\|r'\|^6} r' - \frac{(r'\cdot r'')r'' + r''^2 r'}{\|r'\|^4}\right) + GI_pK_1 \left(\frac{r'\wedge r''}{\|r'\|^3}\right) \\ EI_a \left(\frac{r''}{r'^2} - \frac{r'\cdot r''}{\|r'\|^4} r'\right) \\ GI_pK_1 \end{bmatrix} \tag{3.26}$$

où les dérivées par rapport au temps sont notées avec des points et les dérivées par rapport à la variable d'espace X par des primes.

Les champs vectoriels des forces sont évidemment différents lorsqu'ils sont écrits en approche lagrangienne totale ou réactualisée. Ils ne sont pas présentés ici pour éviter d'alourdir le chapitre. On peut se reporter à la thèse de D. Primault [79] si besoin est.

Dans le cas d'un câble plongé dans un fluide, celui-ci subit des forces extérieures exercées par le fluide. Il est donc intéressant de pouvoir prendre en compte cette charge extérieure. Pour cela, nous utilisons le **principe d'Hamilton étendu** :

$$\delta \int_{t_1}^{t_2} L\, dt + \int_{t_1}^{t_2} \delta W_{ext} dt = 0, \qquad \forall \delta r, \forall \delta\theta \tag{3.27}$$

où δW_{ext} est le travail virtuel produit par la charge extérieure :

$$\delta W_{ext} = \int_0^L \begin{bmatrix} \delta r \\ \delta r' \\ \delta\theta \end{bmatrix} f_{ext}\, dX \tag{3.28}$$

avec

$$f_{ext} = \begin{bmatrix} n \\ \frac{m\wedge t_1}{\|r'\|} \\ t_1 \cdot m \end{bmatrix} \tag{3.29}$$

et $\begin{bmatrix} n \\ m \end{bmatrix}$ le torseur d'efforts appliqué sur la poutre.

Par la suite, la **méthode des éléments finis** est utilisée pour la discrétisation du modèle. Celle-ci repose sur l'écriture de la forme faible du problème. Avec ce qui précède nous pouvons écrire la **forme faible de la dynamique d'une poutre d'Euler-Bernoulli** :

$$\boxed{G(\Phi, \delta\Phi) = (G_{dyn} + G_{stat} - G_{ext})(\Phi, \delta\Phi) = 0, \quad \forall \delta\Phi} \tag{3.30}$$

où Φ décrit la configuration de la poutre et G_{dyn}, G_{stat} et G_{ext}, respectivement, le travail virtuel des forces d'inertie, internes et extérieures.

3.1.6 Linéarisation et discrétisation du modèle d'Euler-Bernoulli

Les formulations précédentes correspondent au problème continu de la déformation d'une poutre fine d'Euler-Bernoulli. En numérique la notion de continu n'existe pas. Il faut linéariser, discrétiser le problème initial pour obtenir un système soluble par un ordinateur. C'est l'objet de cette section : nous allons en premier lieu présenter les linéarisations nécessaires dans le cadre de l'**approche géométriquement exacte**. Puis le problème sera discrétisé suivant la **méthode des éléments finis** et en utilisant l'**algorithme de Newmark** pour le schéma d'intégration en temps afin d'obtenir un système d'équations non-linéaires.

3.1.6.1 Linéarisation de la forme faible

À partir de la forme faible 3.30, qui doit être satisfaite à tout instant du mouvement, on peut écrire la linéarisation suivante :

$$G(\Phi+\Delta\Phi,\delta\Phi) = (G_{dyn}+G_{stat}-G_{ext})+DG_{dyn}\begin{bmatrix}\Delta r\\ \Delta\theta\end{bmatrix}+DG_{stat}\begin{bmatrix}\Delta r\\ \Delta\theta\end{bmatrix}-DG_{ext}\begin{bmatrix}\Delta r\\ \Delta\theta\end{bmatrix}=0 \tag{3.31}$$

où $\Delta\Phi$ décrit la seconde variation de la configuration de la poutre. Remarquons aussi que cette linéarisation est écrite en termes de champ de forces continu comme l'implique la notion de "calcul géométriquement exact".

L'étape suivante consiste à calculer, en utilisant la **dérivée de Frechet**, les variations des fonctionnelles DG à partir des champs de vecteurs f_{dyn} (cf. 3.25), f_{stat} (cf. 3.26) et f_{ext} (cf. 3.29). On perturbe les champs de position et de rotation de la manière suivante (vision eulérienne) :

$$r_\epsilon = r + \epsilon\Delta r \tag{3.32}$$

$$R_\epsilon = exp(\epsilon\Delta\theta\hat{t_{1,\epsilon}})\,\Lambda_{t_1}(t_{1,\epsilon})\,R = R\,\Lambda_{E_1}(R^T\,t_{1,\epsilon})\,exp(\epsilon\Delta\theta\hat{E}_1) \tag{3.33}$$

On peut ainsi écrire les variations des fonctionnelles dynamique, statique et extérieure sous forme d'opérateur continu en faisant apparaître une matrice de raideur K, dite de **"raideur tangente"**.

À partir de 3.25 on obtient après calculs :

$$DG_{dyn}\begin{bmatrix}\Delta r\\ \Delta\theta\end{bmatrix} = \int_0^L \begin{bmatrix}\delta r\\ \delta r'\\ \delta\theta\end{bmatrix} K_{dyn} \begin{bmatrix}\Delta\ddot{r}\\ \Delta\ddot{r}'\\ \Delta\ddot{\theta}\end{bmatrix} dX \tag{3.34}$$

avec la matrice de raideur dynamique K_{dyn} suivante :

$$K_{dyn} = \begin{bmatrix}\rho A I_3 & 0_{3\times3} & 0_{3\times1}\\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1}\\ 0_{1\times3} & 0_{1\times3} & 0\end{bmatrix} \tag{3.35}$$

À partir de 3.26, après de nombreux calculs, la variation de la fonctionnelle statique se met sous la forme :

$$DG_{stat}\begin{bmatrix}\Delta r\\ \Delta\theta\end{bmatrix} = \int_0^L \begin{bmatrix}\delta r^{'}\\ \delta r^{''}\\ \delta\theta^{'}\end{bmatrix} K_{stat} \begin{bmatrix}\Delta r^{'}\\ \Delta r^{''}\\ \Delta\theta^{'}\end{bmatrix} dX \tag{3.36}$$

avec la matrice de raideur statique K_{stat} suivante :

$$K_{stat} = \begin{bmatrix}K_{stat_{11}} & K_{stat_{12}} & K_{stat_{13}}\\ K_{stat_{21}} & K_{stat_{22}} & 0_{3\times1}\\ K_{stat_{31}} & 0_{1\times3} & K_{stat_{33}}\end{bmatrix} \tag{3.37}$$

avec

$$\begin{aligned}
K_{stat_{11}} =& EA\left(\left(1-\frac{1}{\|r^{'}\|}\right)I_3+\frac{1}{\|r^{'}\|^3}r^{'}\otimes r^{'}\right)+\\
& EI_a\left(-\frac{1}{\|r^{'}\|^4}\left(\left\|r^{''}\right\|^2 I_3+r^{''}\otimes r^{''}\right)+\right.\\
& \frac{4}{\|r^{'}\|^6}\left(\left\|r^{''}\right\|^2 r^{'}\otimes r^{'}+\left(r^{'}\cdot r^{''}\right)\left(r^{'}\otimes r^{''}+r^{''}\otimes r^{'}\right)\right)+\\
& \left.\frac{\left(r^{'}\cdot r^{''}\right)^2}{\|r^{'}\|^6}\left(2I_3-\frac{12}{\|r^{'}\|^2}r^{'}\otimes r^{'}\right)\right)+\\
& GI_p\left(-\frac{K_1}{\left\|r^{'3}\right\|}\hat{r^{''}}-3\frac{K_1}{\left\|r^{'5}\right\|}\left(\left(r^{'}\wedge r^{''}\right)\otimes r^{'}\right)+\right.\\
& \left.\frac{1}{\left\|r^{'6}\right\|}\left(r^{'}\wedge r^{''}\right)\otimes\left(r^{'}\wedge r^{''}\right)\right)\\
K_{stat_{22}} =& EI_a\left(\frac{1}{\|r^{'}\|^2}I_3-\frac{1}{\|r^{'}\|^4}r^{'}\otimes r^{'}\right)\\
K_{stat_{21}} =& EI_a\left(-\frac{1}{\|r^{'}\|^4}\left(r^{'}\otimes r^{''}+2r^{''}\otimes r^{'}\right)-\frac{r^{'}\cdot r^{''}}{\|r^{'}\|^4}I_3+4\frac{r^{'}\cdot r^{''}}{\|r^{'}\|^6}r^{'}\otimes r^{'}\right)\\
K_{stat_{12}} =& K_{stat_{21}}^T+GI_p\left(\frac{K_1}{\|r^{'}\|^3}\hat{r^{'}}\right)\\
K_{stat_{33}} =& GI_p\\
K_{stat_{31}} =& GI_p\left(\frac{\left(r^{'}\wedge r^{''}\right)^T}{\|r^{'}\|^3}\right)\\
K_{stat_{13}} =& K_{stat_{31}}^T
\end{aligned} \tag{3.38}$$

À partir de 3.29, la variation de la fonctionnelle des forces extérieures s'écrit :

$$DG_{ext}\begin{bmatrix}\Delta r\\ \Delta\theta\end{bmatrix} = \int_0^L \begin{bmatrix}\delta r\\ \delta r'\\ \delta\theta\end{bmatrix} K_{ext} \begin{bmatrix}\Delta r\\ \Delta r'\\ \Delta\theta\end{bmatrix} dX \tag{3.39}$$

avec, *dans le cas de forces fixes*, la matrice de raideur extérieure K_{ext} suivante :

$$K_{ext} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & K_{ext_{22}} & 0_{3\times1} \\ 0_{1\times3} & K_{ext_{32}} & 0 \end{bmatrix} \tag{3.40}$$

où

$$\begin{aligned} K_{ext_{22}} &= \frac{1}{\|r'\|^2}\hat{m} - \frac{2}{\|r'\|^4}\left(m \wedge r'\right) \otimes r' \\ K_{ext_{32}} &= \frac{1}{\|r'\|}m^T - \frac{r' \cdot m}{\|r'\|^3}r'^T \end{aligned} \tag{3.41}$$

Dans le cas de forces suiveuses (le vecteur d'une force suiveuse garde constamment la même direction par rapport à la ligne neutre de la poutre durant sa déformation) la raideur extérieure est différente et peut être trouvée dans [79].

3.1.6.2 Discrétisation : méthode des éléments finis

Passons maintenant à la discrétisation de la forme faible 3.30 et de sa linéarisation 3.31 par la méthode des éléments finis. Cette méthode n'est pas présentée. De nombreux ouvrages traitent du sujet.

La poutre, de longueur L, est divisée en N éléments indexés e. Chaque tronçon de poutre a une longueur de l_e et son premier point est noté X_e. On a donc :

$$[0, L] = \bigcup_{e=1}^{N} [X_e, X_e + l_e] \tag{3.42}$$

Des **polynômes d'Hermite** sont utilisés pour interpoler les champs de position variationnels δr et incrémentaux Δr. On les note $(N_{I,e}(X), M_{I,e}(X))_{I=\{1,2\}}$ sur l'élément de poutre e (I est l'indice des nœuds de cet élément). $\forall X \in [X_e, X_e + l_e]$, on a :

$$\begin{aligned} \delta r(X) &= N_{I,e}(X)\,\delta r_{I,e} + M_{I,e}(X)\,\delta r'_{I,e} \\ \Delta r(X) &= N_{I,e}(X)\,\Delta r_{I,e} + M_{I,e}(X)\,\Delta r'_{I,e} \end{aligned} \tag{3.43}$$

où $I = 1, 2$ et où la convention de sommation sur I et e est adoptée.

L'interpolation de la torsion se fait, quant à elle, à l'aide de **polynômes de Lagrange**. Ils sont notés $(L_{I,e}(X))_{I=\{1,2\}}$. $\forall X \in [X_e, X_e + l_e]$ on a :

$$\begin{aligned} \delta\theta(X) &= L_{I,e}(X)\delta\theta_{I,e} \\ \Delta\theta(X) &= L_{I,e}(X)\Delta\theta_{I,e} \end{aligned} \tag{3.44}$$

En utilisant ces interpolations dans 3.24, 3.28, 3.34, 3.36 et 3.39, on écrit la forme faible discrète et sa linéarisation discrète :

$$\boxed{\begin{aligned}\tilde{G}(\Phi,\delta\Phi) &= \delta q^T \left(\tilde{f}_{dyn} + \tilde{f}_{stat} - \tilde{f}_{ext}\right) = 0 \\ (DG(\Phi,\delta\Phi)(\Delta\Phi))\tilde{} &= \delta q^T \left(\tilde{K}_{dyn}\,\Delta\ddot{q} + \left(\tilde{K}_{stat} - \tilde{K}_{ext}\right)\Delta q\right)\end{aligned}} \tag{3.45}$$

où le tilde signifie que l'on a affaire à un vecteur discret issu d'un champ continu ($\tilde{G} = G\tilde{}$)

δq et Δq sont des vecteurs de $7(N+1)$ composantes. Ils correspondent aux variations nodales qui apparaissent dans 3.43 et 3.44. Les matrices $\tilde{K}_{dyn}$, $\tilde{K}_{stat}$ et $\tilde{K}_{ext}$ sont des matrices $7(N+1)\times 7(N+1)$, composées des matrices raideur élémentaires. De même $\tilde{f}_{dyn}$, $\tilde{f}_{stat}$ et $\tilde{f}_{ext}$ sont déduites par assemblage des vecteurs élémentaires correspondants.

3.1.6.3 Intégration temporelle : schéma de Newmark

Pour obtenir un système soluble numériquement il reste à éliminer le terme $\Delta\ddot{q}$. C'est donc le moment de s'intéresser à l'intégration temporelle. Pour résoudre ce problème on fait appel au **schéma en temps standard de Newmark**. On peut alors écrire :

$$\boxed{\begin{aligned}\Delta\ddot{q}^{k+1}_{t+\Delta t} &= \frac{1}{\beta(\Delta t)^2} S\,\Delta q^{k+1}_{t+\Delta t} \\ \Delta\dot{q}^{k+1}_{t+\Delta t} &= \frac{\gamma}{\beta\Delta t} S\,\Delta q^{k+1}_{t+\Delta t}\end{aligned}} \tag{3.46}$$

où β et γ sont les coefficients du schéma standard de Newmark (Dans notre solveur poutre $\beta = \frac{1}{4}$ et $\gamma = \frac{1}{2}$, ce qui correspond à "la règle du trapèze" et l'ordre est de 2). $\Delta q^{k+1}_{t+\Delta t} = q^{k+1}_{t+\Delta t} - q^{k}_{t+\Delta t}$ est le vecteur des incréments nodaux entre deux itérations de l'algorithme de Newton qui sera utilisé pour résoudre le système. L'opérateur S est une matrice permettant de ne sélectionner que les incréments de position et leurs dérivées :

$$S = diag_{N+1 fois}\begin{bmatrix} I_3 & 0_{3\times 3} & 0_{3\times 1} \\ 0_{3\times 3} & I_3 & 0_{3\times 1} \\ 0_{1\times 3} & 0_{1\times 3} & 0 \end{bmatrix} \tag{3.47}$$

On peut finalement former le **système soluble** 3.48 en utilisant 3.46 dans 3.45. Pour résoudre ce système, un algorithme de Newton peut être utilisé. Dans notre cas pour ne pas avoir à coder un sous-programme de résolution de système matriciel, nous utilisons la librairie LAPACK et ses sous-programmes (cf. [6]).

$$\boxed{\left(\tilde{K}^k_{dyn_{t+\Delta t}}\frac{S}{\beta(\Delta t)^2} + \tilde{K}^k_{stat_{t+\Delta t}} - \tilde{K}^k_{ext_{t+\Delta t}}\right)\Delta q^{k+1}_{t+\Delta t} = -\left(\tilde{f}^k_{dyn_{t+\Delta t}} + \tilde{f}^k_{stat_{t+\Delta t}} - \tilde{f}^k_{ext_{t+\Delta t}}\right)} \tag{3.48}$$

Dans le modèle de la poutre fine d'Euler-Bernoulli, l'inertie de rotation des sections est négligée, ce qui a pour conséquence d'annuler les termes dynamiques d'inertie de torsion. On a alors *apparition d'une singularité lorsque la poutre est libre de tourner autour de son axe*. Pour éviter ce problème, le **modèle de Rayleigh**, présenté dans § 3.1.7, a été développé.

3.1.7 Modèle de Rayleigh

Pour éliminer les singularités dynamiques de torsion rencontrées dans le modèle d'Euler-Bernoulli, le champ de vitesse angulaire (de flexion et de torsion) est réintroduit dans le lagrangien du modèle de Rayleigh. Nous remplaçons donc le terme d'énergie cinétique de translation du lagrangien d'Euler-Bernoulli 3.21 par :

$$T = \underbrace{\frac{1}{2}\int_0^L \rho A \dot{r}^2 dX}_{\substack{\text{Énergie cinétique} \\ \text{de translation}}} + \underbrace{\frac{1}{2}\int_0^L \rho I_p \Omega_1^2 dX}_{\substack{\text{Énergie cinétique} \\ \text{angulaire de flexion}}} + \underbrace{\frac{1}{2}\int_0^L \rho I_a \left(\Omega_2^2 + \Omega_3^2\right) dX}_{\substack{\text{Énergie cinétique} \\ \text{angulaire de torsion}}} \tag{3.49}$$

En suivant la même méthode que celle exposée pour le modèle d'Euler-Bernoulli, nous obtenons la fonctionnelle dynamique G_{dyn} :

$$G_{dyn} = \int_0^L \begin{bmatrix} \delta r \\ \delta r' \\ \delta\theta \end{bmatrix} f_{dyn}\, dX \tag{3.50}$$

avec le champ vectoriel des forces d'inertie suivant :

$$f_{dyn} = \begin{bmatrix} \rho A \ddot{r} \\ -\rho I_p \Omega_1 \frac{1}{\|r'\|^3}\left(r' \wedge \dot{r}'\right) + \rho I_a \left(\frac{1}{\|r'\|^2}\ddot{r}' - 2\frac{r'\cdot\dot{r}'}{\|r'\|^4}\dot{r}' - \frac{r'\cdot\ddot{r}'}{\|r'\|^4}r' + 2\frac{\left(r'\cdot\dot{r}'\right)^2}{\|r'\|^6}r' \right) \\ \rho I_p \dot{\Omega}_1 \end{bmatrix} \tag{3.51}$$

De même qu'à la section 3.1.6, la fonctionnelle dynamique linéarisée est calculée à partir de ΔG_{dyn}. Mais à ce stade un important problème se pose : nous avons jusque là travaillé en approche eulérienne, c'est-à-dire avec l'angle de torsion itératif $\Delta\theta$. Mais nous avons vu que cet incrément n'a de sens qu'entre deux itérations de Newton. Il n'est donc pas possible de calculer sa somme entre deux pas de temps et d'obtenir un résultat cohérent avec la physique. *Pour surmonter ce problème nous allons temporairement utiliser l'approche lagrangienne réactualisée et son angle de torsion "actuel"* ϑ. Nous appliquons le **schéma de Newmark détaillé par Simo** dans [90] uniquement pour la dynamique de torsion :

$$\boxed{\begin{aligned} \Delta\Omega_1 &= \frac{\gamma}{\beta\Delta t}\Delta\vartheta \\ \Delta\dot{\Omega}_1 &= \frac{1}{\beta(\Delta t)^2}\Delta\vartheta \end{aligned}} \tag{3.52}$$

En calculant Δf_{dyn} on remarque que les incréments $\Delta\Omega_1$ et $\Delta\dot{\Omega}_1$ interviennent linéairement. Par conséquent via 3.52 $\Delta\vartheta$ aussi. Évidemment nous cherchons à écrire la fonctionnelle dynamique linéarisée ΔG_{dyn} en fonction de $\Delta\theta$ pour avoir une écriture homogène entre les différentes matrices de raideur. En différenciant 3.14 on obtient une relation entre $\Delta\theta$ et $\Delta\vartheta$:

$$\Delta\theta = \Delta\vartheta - u_n(r') \cdot \Delta r' \tag{3.53}$$

avec $u_n(r') = \dfrac{r_n' \wedge r'}{\|r'\| \left(\|r_n'\| \|r'\| + r_n' \cdot r' \right)}$

Donc en utilisant 3.53, le modèle de Rayleigh nous donne la fonctionnelle dynamique linéarisée suivante :

$$DG_{dyn} \begin{bmatrix} \Delta r \\ \Delta\theta \end{bmatrix} = \int_0^L \begin{bmatrix} \delta r \\ \delta r' \\ \delta\theta \end{bmatrix} K_{dyn} \begin{bmatrix} \Delta\ddot{r} \\ \Delta\ddot{r}' \\ \Delta\ddot{\theta} \end{bmatrix} dX \tag{3.54}$$

avec la matrice de raideur dynamique K_{dyn} suivante :

$$K_{dyn} = \begin{bmatrix} K_{dyn_{11}} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & K_{dyn_{22}} & K_{dyn_{23}} \\ 0_{1\times3} & K_{dyn_{32}} & K_{dyn_{33}} \end{bmatrix} \tag{3.55}$$

où

$$\begin{aligned} K_{dyn_{11}} &= \frac{\rho A}{\beta (\Delta t)^2} I_3 \\ K_{dyn_{23}} &= \rho I_p \left(-\frac{\gamma}{\beta \Delta t} \frac{1}{\|r'\|^3} \left(r' \wedge \dot{r}' \right) \right) \\ K_{dyn_{32}} &= \rho I_p \left(\frac{1}{\beta (\Delta t)^2} u_n^T \right) \\ K_{dyn_{33}} &= \frac{\rho I_p}{\beta (\Delta t)^2} \end{aligned} \tag{3.56}$$

$$
\begin{aligned}
K_{dyn_{22}} =& \rho I_p \left(-\frac{\gamma}{\beta \Delta t} \frac{1}{\|r'\|^3} \left(r' \wedge \dot{r}' \right) \otimes u_n + \Omega_1 \frac{1}{\|r'\|^3} \hat{\dot{r}}' - \right. \\
& \left. \frac{\gamma}{\beta \Delta t} \Omega_1 \frac{1}{\|r'\|^3} \hat{r}' + 3\Omega_1 \frac{1}{\|r'\|^5} \left(r' \wedge \dot{r}' \right) \otimes r' \right) + \\
& \rho I_a \left(-\frac{2}{\|r'\|^4} \ddot{r}' \otimes r' - \frac{2}{\|r'\|^4} \dot{r}' \otimes \dot{r}' + \right. \\
& \frac{8}{\|r'\|^6} \left(r' \cdot \dot{r}' \right) \dot{r}' \otimes r' - \frac{1}{\|r'\|^4} r' \otimes \ddot{r}' + \\
& \frac{4}{\|r'\|^6} \left(r' \cdot \ddot{r}' \right) r' \otimes r' - \frac{1}{\|r'\|^4} \left(r' \cdot \ddot{r}' \right) I_3 + \frac{4}{\|r'\|^6} \left(r' \cdot \dot{r}' \right) r' \otimes \dot{r}' + \\
& \frac{2}{\|r'\|^6} \left(r' \cdot \dot{r}' \right)^2 I_3 - \frac{12}{\|r'\|^8} \left(r' \cdot \dot{r}' \right)^2 r' \otimes r' + \\
& \frac{\gamma}{\beta \Delta t} \left(-\frac{2}{\|r'\|^4} \dot{r}' \otimes r' - \frac{2}{\|r'\|^4} \left(r' \cdot \dot{r}' \right) I_3 + \frac{4}{\|r'\|^6} \left(r' \cdot \dot{r}' \right) r' \otimes r' \right) + \\
& \left. \left. \frac{1}{\beta \left(\Delta t \right)^2} \left(\frac{1}{\|r'\|^2} I_3 - \frac{1}{\|r'\|^4} r' \otimes r' \right) \right) \right)
\end{aligned}
$$

De même que pour le modèle d'Euler-Bernoulli, le champ vectoriel des forces d'inertie 3.51 et la matrice de raideur dynamique 3.55 sont discrétisés. On remplace dans 3.48 ceux d'Euler-Bernoulli par ceux calculés précédemment et on obtient le système à résoudre pour le modèle de Rayleigh.

Au cours des sous-chapitres précédents, nous avons vu apparaître des exposants k relatifs aux itérations de l'**algorithme de Newton**. Comment passer de k à $k+1$? En d'autres termes comment mettre à jour les formules à t fixé ? La section suivante (§ 3.1.8) nous l'explique.

3.1.8 Mise à jour de la forme discrétisée des modèles

Que ce soit avec le modèle d'Euler-Bernoulli ou celui de Rayleigh, afin de résoudre l'équilibre sur plusieurs pas de temps, il nous faut réactualiser le **résidu** (la somme de $\tilde{f}_{dyn}$, $\tilde{f}_{stat}$ et $-\tilde{f}_{ext}$) et la **raideur tangente** (la somme de $\tilde{K}_{dyn}$, $\tilde{K}_{stat}$ et $-\tilde{K}_{ext}$) à chaque itération k de Newton (et ce, aux points de Gauss des éléments composant le maillage de la poutre). Cette mise à jour nécessite de connaître les incréments nodaux finis $\Delta q_{t+\Delta t}^{k+1}$. Ceux-ci sont calculés en résolvant le système 3.48.

Pour les deux modèles, les champs suivants doivent être réactualisés :

- ➢ **Champ de position** : $X \in [0, L] \mapsto r(X) \in \mathbb{R}^3$
- ➢ **Champ de tangente de l'espace matériel** : $X \in [0, L] \mapsto r'(X) \in \mathbb{R}^3$
- ➢ **Champ de courbure de l'espace matériel** : $X \in [0, L] \mapsto r''(X) \in \mathbb{R}^3$
- ➢ **Champ de vitesse linéaire** : $X \in [0, L] \mapsto \dot{r}(X) \in \mathbb{R}^3$
- ➢ **Champ d'accélération linéaire** : $X \in [0, L] \mapsto \ddot{r}(X) \in \mathbb{R}^3$

➢ **Champ de taux de torsion** : $X \in [0, L] \mapsto K_1(X) \in \mathbb{R}$

Le modèle de Rayleigh nécessite la réactualisation de deux champs supplémentaires :
➢ **Champ de vitesse angulaire de torsion** : $X \in [0, L] \mapsto \Omega_1(X) \in \mathbb{R}$
➢ **Champ d'accélération angulaire de torsion** : $X \in [0, L] \mapsto \dot{\Omega}_1(X) \in \mathbb{R}$

Voici les formules de mise à jour de ces champs. Elles sont écrites pour tout $X \in (X_e, X_e + l_e)$. Les notations pour les polynômes d'Hermite et de Lagrange sont les mêmes que précédemment (cf. 3.43 et 3.44).

3.1.8.1 Réactualisation du champ de position

$$\boxed{r^{(k+1)}_{(t+\Delta t)} = r^{(k)}_{(t+\Delta t)} + \Delta r} \tag{3.57}$$

avec $\Delta r = \Delta \tilde{r}^{(k+1)}_{(t+\Delta t)}(X) = N_{I,e}(X)\, \Delta r^{(k+1)}_{I,e_{(t+\Delta t)}} + M_{I,e}(X)\, \Delta r'^{(k+1)}_{I,e_{(t+\Delta t)}}$.

3.1.8.2 Réactualisation du champ de tangente de l'espace matériel

$$\boxed{r'^{(k+1)}_{(t+\Delta t)} = r'^{(k)}_{(t+\Delta t)} + \Delta r'} \tag{3.58}$$

avec $\Delta r' = \Delta \tilde{r}'^{(k+1)}_{(t+\Delta t)}(X) = N'_{I,e}(X)\, \Delta r^{(k+1)}_{I,e_{(t+\Delta t)}} + M'_{I,e}(X)\, \Delta r'^{(k+1)}_{I,e_{(t+\Delta t)}}$.

3.1.8.3 Réactualisation du champ de courbure de l'espace matériel

$$\boxed{r''^{(k+1)}_{(t+\Delta t)} = r''^{(k)}_{(t+\Delta t)} + \Delta r''} \tag{3.59}$$

avec $\Delta r'' = \Delta \tilde{r}''^{(k+1)}_{(t+\Delta t)}(X) = N''_{I,e}(X)\, \Delta r^{(k+1)}_{I,e_{(t+\Delta t)}} + M''_{I,e}(X)\, \Delta r'^{(k+1)}_{I,e_{(t+\Delta t)}}$.

3.1.8.4 Réactualisation du champ de vitesse linéaire

$$\boxed{\dot{r}^{(k+1)}_{(t+\Delta t)} = \dot{r}^{(k)}_{(t+\Delta t)} + \frac{\gamma}{\beta \Delta t} \Delta r} \tag{3.60}$$

avec $\Delta \dot{r} = \Delta \tilde{\dot{r}}^{(k+1)}_{(t+\Delta t)}(X) = N_{I,e}(X)\, \Delta \dot{r}^{(k+1)}_{I,e_{(t+\Delta t)}} + M_{I,e}(X)\, \Delta \dot{r}'^{(k+1)}_{I,e_{(t+\Delta t)}}$.

3.1.8.5 Réactualisation du champ d'accélération linéaire

$$\boxed{\ddot{r}^{(k+1)}_{(t+\Delta t)} = \ddot{r}^{(k)}_{(t+\Delta t)} + \frac{1}{\beta\, (\Delta t)^2} \Delta r} \tag{3.61}$$

avec $\Delta \ddot{r} = \Delta \tilde{\ddot{r}}^{(k+1)}_{(t+\Delta t)}(X) = N_{I,e}(X)\, \Delta \ddot{r}^{(k+1)}_{I,e_{(t+\Delta t)}} + M_{I,e}(X)\, \Delta \ddot{r}'^{(k+1)}_{I,e_{(t+\Delta t)}}$.

3.1.8.6 Réactualisation du champ de rotation

La réactualisation du champ de rotation est plus complexe. *Rappelons que nous sommes en approche eulérienne.* La mise à jour du champ de rotation peut alors se faire à droite ou à gauche. Par exemple voici la réactualisation par la gauche :

$$\boxed{R^{(k+1)}_{(t+\Delta t)} = exp\left(\Delta\theta(t_1+\Delta t_1)\hat{}\right)\,\Lambda_{t_1}(t_1+\Delta t_1)\,R^{(k)}_{(t+\Delta t)}} \tag{3.62}$$

où $t_1 = t^{(k)}_{1_{(t+\Delta t)}}(X)$, $\Delta t_1 = t^{(k+1)}_{1_{(t+\Delta t)}}(X) - t^{(k)}_{1_{(t+\Delta t)}}(X)$ avec $t^{(k+1)}_{1_{(t+\Delta t)}}(X) = \frac{r'^{(k+1)}_{(t+\Delta t)}(X)}{\left\|r'^{(k+1)}_{(t+\Delta t)}(X)\right\|}$
et enfin $\Delta\theta = \Delta\tilde{\theta}^{(k+1)}_{(t+\Delta t)}(X) = L_{I,e}(X)\Delta\theta^{(k+1)}_{I,e_{(t+\Delta t)}}$

La formulation nécessite de connaître le champ de taux spatial de torsion K_1 à chaque itération de Newton. Il faut donc aussi le mettre à jour.

3.1.8.7 Réactualisation du champ de taux de torsion

$$\boxed{\begin{aligned} K^{(k+1)}_{1_{(t+\Delta t)}} =& K^{(k)}_{1_{(t+\Delta t)}} - \frac{1}{1+t^{(k)}_{1_{(t+\Delta t)}}\cdot t^{(k+1)}_{1_{(t+\Delta t)}}}\left(\left(t^{(k+1)}_{1_{(t+\Delta t)}} \wedge t_1{}'^{(k+1)}_{(t+\Delta t)}\right)\cdot t^{(k)}_{1_{(t+\Delta t)}} - \right. \\ & \left.\left(t^{(k)}_{1_{(t+\Delta t)}} \wedge t_1{}'^{(k)}_{(t+\Delta t)}\right)\cdot t^{(k+1)}_{1_{(t+\Delta t)}}\right) + \Delta\theta' \end{aligned}} \tag{3.63}$$

où $\Delta\theta' = \Delta\tilde{\theta}'^{(k+1)}_{(t+\Delta t)}(X) = L'_I(X)\Delta\theta^{(k+1)}_{I,e_{(t+\Delta t)}}$

3.1.8.8 Réactualisation du champ de vitesse et d'accélération angulaire de torsion

$$\boxed{\begin{aligned} \Omega^{(k+1)}_{1_{(t+\Delta t)}} &= \Omega^{(k)}_{1_{(t+\Delta t)}} + \frac{\gamma}{\beta\Delta t}\Delta\vartheta \\ \dot{\Omega}^{(k+1)}_{1_{(t+\Delta t)}} &= \dot{\Omega}^{(k)}_{1_{(t+\Delta t)}} + \frac{1}{\beta\left(\Delta t\right)^2}\Delta\vartheta \end{aligned}} \tag{3.64}$$

avec $\Delta\vartheta = \Delta\tilde{\vartheta}^{(k+1)}_{(t+\Delta t)}(X) = L_{I,e}(X)\left(\vartheta^{(k+1)}_{I,e_{(t+\Delta t)}} - \vartheta^{(k)}_{I,e_{(t+\Delta t)}}\right)$.

Le problème est le calcul de la réactualisation de l'angle de torsion actuel $\vartheta^{(k+1)}_{I,e_{(t+\Delta t)}}$. D'après [16] on a :

$$\vartheta^{(k+1)}_{I,e_{(t+\Delta t)}} = exp^{-1}_{\hat{E}_1}\left(\Lambda^T_{E_1}\left(R^T_n\, t^{(k+1)}_{1_{(t+\Delta t)}}\right)\, R^T_n\, R^{(k+1)}_{(t+\Delta t)}\right) \tag{3.65}$$

L'extraction de $\vartheta^{(k+1)}_{I,e_{(t+\Delta t)}}$ est simple, car l'axe de rotation est connu et fixe (E_1). On a donc juste à calculer un arc tangente.

Nous avons, avec ces formules de réactualisation des champs, tout pour construire l'algorithme numérique résolvant le problème d'une poutre soumise à des sollicitations

extérieures et se déformant sous leurs actions. Cependant le problème ainsi formulé ne peut prendre en compte toutes les configurations d'une poutre : par exemple *comment traiter une poutre encastrée à une ou aux deux extrémités* ? Un autre problème doit donc être étudié : *celui d'une poutre subissant des mouvements imposés à ses extrémités.*

3.1.9 Problème des mouvements imposés aux extrémités de la poutre

Plaçons-nous dans le cas d'une poutre qui peut être encastrée ou dont les extrémités peuvent être entraînées par un mécanisme quelconque. Les deux modèles de poutre vus précédemment restent valables et leur utilisation sera transparente. En effet, modéliser ce nouveau problème ne nécessite pas de refaire tous les calculs. Il reste juste à ajouter les efforts de réaction imposés par les nouvelles contraintes aux contributions de l'équilibre précédent. Pour cela, on utilise la **méthode des multiplicateurs de Lagrange** (cf. [20] et [51]).

Les liaisons, connectant les deux nœuds extrémités de la poutre à l'environnement, sont des *contraintes géométriques*. On peut alors les formuler de la manière suivante :

$$\Upsilon(\Phi, t) = \begin{bmatrix} \Upsilon_0(\Phi(0), t) \\ \Upsilon_L(\Phi(L), t) \end{bmatrix} = \begin{bmatrix} 0_{m_1 \times 1} \\ 0_{m_2 \times 1} \end{bmatrix} \tag{3.66}$$

où, comme précédemment, Φ est le champ décrivant les positions et orientations de la poutre et $m = m_1 + m_2$ est le nombre de degrés de liberté contraints par les liaisons. $\Upsilon_X(\Phi, t)$ est défini comme suit :

$$\Upsilon_X(\Phi, t) = \begin{bmatrix} r(X) - r_{imp}(X, t) \\ r'(X) - r'_{imp}(X, t) \\ \vartheta(X) - \vartheta_{imp}(X, t) \end{bmatrix} \tag{3.67}$$

avec $r_{imp}(X, t)$, $r'_{imp}(X, t)$ et $\vartheta_{imp}(X, t)$ étant, respectivement, la position, la tangente et la torsion imposées du point X à t.

Pour résoudre notre nouveau problème, le système doit obéir aux lois de la dynamique (comme dans le précédent problème), mais encore vérifier à tout instant l'égalité 3.66. Pour arriver à nos fins, on force les contraintes imposées par 3.66 et 3.67, via le vecteur des multiplicateurs de Lagrange $\lambda = \begin{bmatrix} \lambda_0 \\ \lambda_L \end{bmatrix} \in \mathbb{R}^m$. Ceux-ci sont des inconnues du problème que l'on va résoudre grâce à un **"lagrangien augmenté"** :

$$\boxed{L_+ = L_{Précédent\ problème} + \lambda^T \Upsilon + p \Upsilon^T \Upsilon} \tag{3.68}$$

Ce "lagrangien augmenté" se compose donc du lagrangien du problème non-contraint (ou "libre"), d'un terme potentiel dû aux forces de réaction et enfin d'un terme de "pénalité" p assurant une meilleure convergence de la méthode. p est un scalaire à fixer. La solution finale est indépendante de cette valeur (le terme $p \Upsilon^T \Upsilon$ s'annule, une fois la

convergence atteinte).

À partir de ce "lagrangien augmenté", nous recommençons une à une les étapes effectuées dans le cas non-contraint :

En appliquant le principe d'Hamilton, on aboutit à la **forme faible "augmentée"** suivante :

$$\boxed{G_+ = G_{libre} + \delta\lambda^T\, r_{contr} + \delta\Phi^T(0)\ (f_{réac}(0) + f_{pen}(0)) + \delta\Phi^T(L)\ (f_{réac}(L) + f_{pen}(L))} \tag{3.69}$$

avec $r_{contr} = \Upsilon$, $f_{réac}$ est la force généralisée de réaction et f_{pen} la force de rappel élastique induite par la pénalité :

$$\begin{aligned} \delta\Phi^T(0)\, f_{réac}(0) &= \delta(\Upsilon_0^T)\, \lambda_0 \\ \delta\Phi^T(L)\, f_{réac}(L) &= \delta(\Upsilon_L^T)\, \lambda_L \\ \delta\Phi^T(0)\, f_{pen}(0) &= \delta(\Upsilon_0^T)\ (2p\Upsilon_0) \\ \delta\Phi^T(L)\, f_{pen}(L) &= \delta(\Upsilon_L^T)\ (2p\Upsilon_L) \end{aligned} \tag{3.70}$$

Vient ensuite l'étape de linéarisation de la "forme faible augmentée" 3.69 :

$$\boxed{\Delta G_+ = \Delta G_{libre} + \Delta\delta\,(\lambda^T\,\Upsilon) + p\,\Delta\delta\,(\Upsilon^T\,\Upsilon)} \tag{3.71}$$

Une fois la linéarisation terminée, on discrétise 3.69 et 3.71 par la méthode des éléments finis. On fait alors apparaître un vecteur **"résidu augmenté"**, $\tilde{r}_+$, composé des résidus discrets élémentaires issus de la discrétisation de 3.69 ; puis la matrice de **"raideur tangente augmentée"**, $\tilde{K}_+$, formée à partir des matrices de raideur discrètes élémentaires venant de la discrétisation de 3.71. En utilisant, comme dans le précédent problème, le schéma d'intégration temporelle de Newmark, on peut écrire le **système "augmenté"** à résoudre :

$$\boxed{\tilde{K}_+\,\Delta\tilde{\Phi}_+ = -\tilde{r}_+} \tag{3.72}$$

avec $\tilde{r}_+ = \begin{bmatrix} \tilde{f}_{dyn} + \tilde{f}_{stat} - \tilde{f}_{ext} + \tilde{f}_{réac} + \tilde{f}_{pen} \\ r_{contr} \end{bmatrix}$ et $\tilde{K}_+ = \tilde{K}_{libre_+} + \tilde{K}_{réac_+} + \tilde{K}_{pen_+}$.

$\tilde{f}_{dyn}$, $\tilde{f}_{stat}$ et $\tilde{f}_{ext}$ sont les mêmes que dans le précédent problème.

$\Delta\tilde{\Phi}_+ = \begin{bmatrix} \Delta\tilde{\Phi} \\ \Delta\lambda \end{bmatrix}$ est le "vecteur nodal des variations augmentées".

Le système est résolu grâce à un algorithme de Newton. Une procédure de mise à jour des champs est évidemment nécessaire. En ce qui concerne les inconnues cinématiques, la procédure de réactualisation a déjà été traitée à la section 3.1.8. Il reste donc à s'intéresser aux multiplicateurs de Lagrange. *Ces derniers évoluant sur un espace vectoriel, la mise à*

jour est simplement additive.

Dans ce sous-chapitre, la méthode des multiplicateurs de Lagrange a été appliquée pour résoudre le problème d'une poutre contrainte aux extrémités. Pour plus de détails sur la résolution de ce problème (en particulier les résultats obtenus avec les différentes approches cinématiques), il faut se reporter à la thèse de D. Primault [79]. Notons que, dans le solveur poutre codé lors de cette thèse, le formalisme eulérien a été utilisé, excepté pour l'écriture des contraintes géométriques aux extrémités.

Dans ce chapitre nous avons vu la théorie sur laquelle se base le solveur poutre programmé lors de cette thèse. Une fois codé, une étape de tests a été nécessaire pour le valider.

3.2 Validation du solveur poutre

La validation du solveur poutre s'est effectuée en deux temps : tout d'abord une **validation bidimensionnelle** nous a permis d'enlever la majeure partie des problèmes de fonctionnement rencontrés. Ensuite, le code a été validé en **tridimensionnel**. Les tests se basent sur des résultats théoriques d'une part et d'autre part sur des données fournies par d'autres solveurs à travers diverses publications.

3.2.1 Validation en 2D

En 2D j'ai décidé, en tout premier lieu, de comparer les résultats numériques, obtenus via le solveur poutre, avec les résultats de la **théorie des petits déplacements, petites déformations** sur des cas très simples. Un fois les petits déplacements validés, deux cas-tests 2D en **grand déplacement** ont été étudiés.

3.2.1.1 Cas des petits déplacements

Pour tester et valider le solveur poutre du point de vue **statique**, un cas bidimensionnel très simple a été choisi : une poutre subit une charge uniforme de valeur q. Sous l'effet de cette charge, la poutre va se courber. En faisant varier le nombre de points constituant le maillage structure, nous allons comparer les résultats obtenus numériquement avec une valeur de référence de la flèche de la poutre.

Comment fixer cette valeur de référence ? Pour une charge q suffisamment faible, on peut comparer la flèche de la poutre déformée calculée avec la valeur théorique obtenue en faisant l'hypothèse des petites déformations (HPD). Cette dernière est donnée par la formule suivante :

$$\boxed{V_{théorique}(L) = \frac{q}{8EI}} \tag{3.73}$$

Pour $q = 1$, la charge est assez petite pour que la valeur de référence soit la valeur donnée par la théorie des petites déformations. Plus la charge augmente, plus l'hypothèse des petites déformations (HPD) devient caduque. La flèche numérique ne tend plus vers

la flèche des petites déformations. Il nous faut, cependant, vérifier l'ordre de convergence de la méthode pour des cas où q est supérieure à 1. On considère alors que pour $n = 500$, la solution du problème structure est atteinte, et nous prenons cette flèche calculée numériquement pour référence.

Charge (q) / Pas d'espace (h)	q=1	q=10	q=100
h=0.1	0.000429999	0.004299938	0.042936363
h=0.02	0.000428628	0.004286225	0.042800024
h=0.01	0.000428585	0.004285796	0.042795759
h=0.002	0.000428571	0.004285702	0.042794395
Valeur de référence	0.000428571	0.004285702	0.042794395
Valeur théorique (HPD)	0.000428571	0.004285714	0.042857143

TAB. 3.1: Validation statique pour Euler-Bernoulli (unités SI)

En comparant les données numériques et les valeurs de références du tableau 3.1, on voit que plus on raffine le maillage structure, plus on tend vers la solution de référence. On vérifie par la même occasion que plus la charge est importante, plus l'écart entre les résultats numériques et la théorie HPD augmente.

Avec le modèle d'Euler-Bernoulli l'ordre de convergence est de 2 et ce quelque soit la charge : 2.01 pour $q = 1$; 2.15 pour $q = 10$; 2.02 pour $q = 100$. En statique, l'équation de Rayleigh revient à l'équation d'Euler-Bernoulli. Donc **l'ordre de la méthode Rayleigh en statique est aussi de 2**.

On peut ensuite comparer la déformée d'une poutre de longueur L, de module d'Young E et de moment quadratique I avec la déformée théorique d'une poutre uniformément chargée (charge q). Cette dernière est donnée par :

$$\boxed{V_{théorique}(x) = \frac{1}{EI}(\frac{q}{24}x^4 - \frac{qL}{6}x^3 + \frac{qL^2}{2}\frac{x^2}{2})} \tag{3.74}$$

Plusieurs charges sont testées. L'erreur relative entre la déformée théorique et la numérique diminue lorsqu'on raffine le maillage. Mais pour un maillage de 10 points avec une charge de $1000\,SI$, la déformée calculée grâce au solveur poutre est déjà très proche de la déformée théorique (cf. fig. 3.4, où la poutre au repos est suivant $\overrightarrow{y}$, donc $\overrightarrow{E_1} = \overrightarrow{e_2} = \overrightarrow{y}$).

Une fois le solveur validé statiquement, une **validation dynamique** s'impose. Une poutre encastrée est chargée pendant une courte période au début du test, puis libérée dans un fluide au repos (cf. fig. 3.5). Dans le vide la poutre va osciller sans amortissement à une fréquence égale à la première fréquence propre théorique (cf. [76]) :

$$\boxed{f_1 = 0.5595\frac{1}{L^2}\sqrt{\frac{EI}{m/L}}} \tag{3.75}$$

Pour ce cas-test une poutre de longueur $L = 1\,m$, d'épaisseur $d = 0.01\,m$, de masse volumique $\rho_{beam} = 1200\,SI$ et de module d'Young $E = 3500\,MPa$ a été choisie. Par

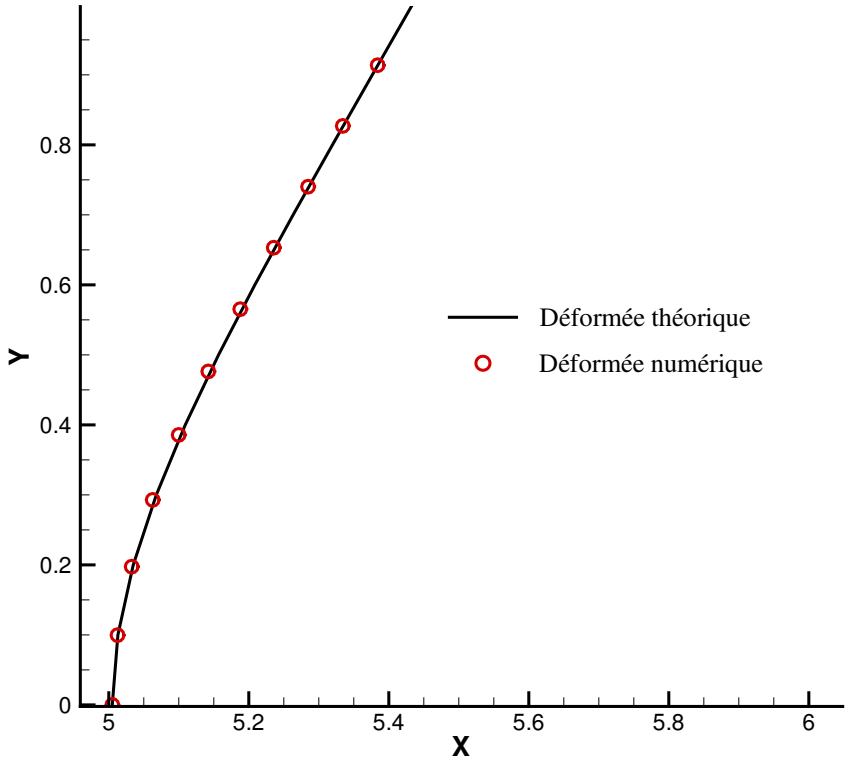


FIG. 3.4: Déformée théorique et déformée calculée numériquement ($q = 1000$) (unités SI)

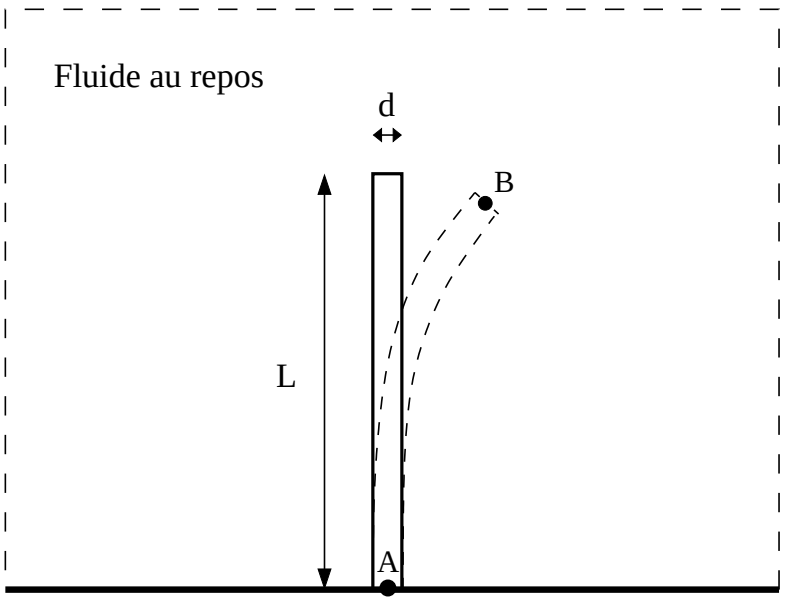


FIG. 3.5: Poutre encastrée en A dans un fluide au repos

conséquent la première fréquence propre théorique est $f_1 = 2.75837\,Hz$. On va, grâce à ce cas-test, **vérifier la convergence du schéma de discrétisation temporelle de Newmark pour le modèle d'Euler-Bernoulli**. En faisant varier le pas de temps, on calcule par transformée de Fourier la première fréquence d'oscillation de la poutre sur $10\,s$ (cf. tab. 3.2). En traçant une régression linéaire sur les cinq premières valeurs ($\Delta t > 0.001$) et en calculant sa pente, on trouve un ordre de convergence en temps de la méthode de Newmark de **1.99** ce qui est très proche de la valeur théorique de 2. Pour des pas de temps $\Delta t \leqslant 0.001$, les erreurs numériques deviennent trop importantes et faussent la valeur de cet ordre de convergence.

Δt (s)	0.09	0.07	0.05	0.01	0.005	0.001	Valeur de référence
f (Hz)	2.32558	2.48447	2.59740	2.75229	2.75344	2.75771	2.75837

TAB. 3.2: Validation dynamique pour le modèle d'Euler-Bernoulli (unités SI)

Sur la figure 3.6 on peut voir l'évolution des oscillations du point B après un chargement statique pendant $0.1\,s$ pour un pas de temps de $0.001\,s$. Les fréquences d'oscillation sont données par transformée de Fourier (cf. fig. 3.7). Si on compare la valeur théorique de la première fréquence d'oscillation avec celle calculée on a une erreur très faible de 0.02%.

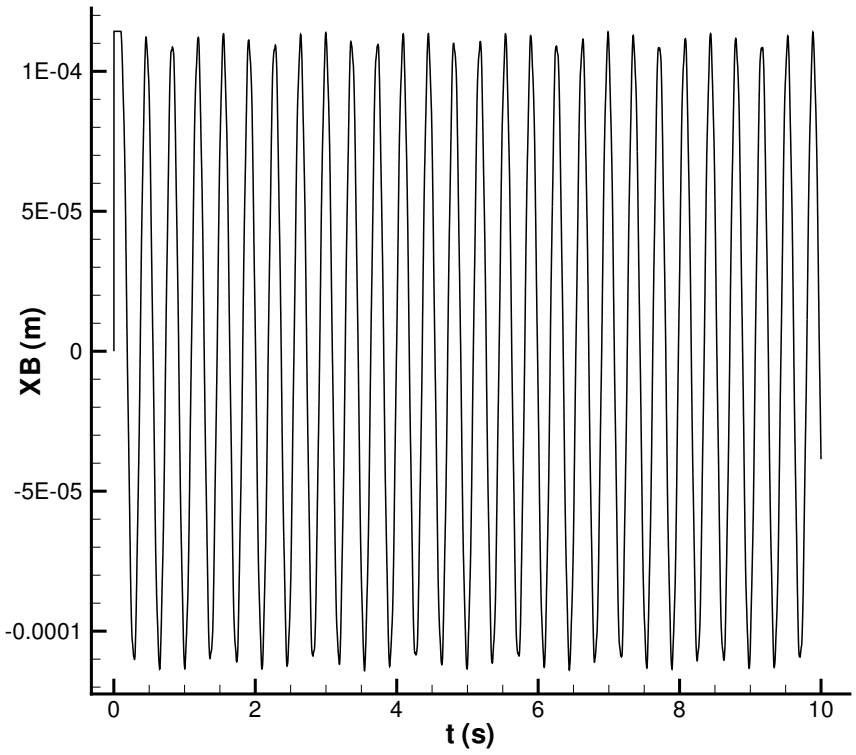


FIG. 3.6: Oscillations du point B dans le vide (modèle Euler-Bernoulli)

On recommence le même cas-test avec cette fois-ci le **modèle de Rayleigh**. En faisant varier le pas de temps, on trouve des fréquences d'oscillation de la poutre qui tendent vers la valeur de référence (cf. tab. 3.3). On obtient un ordre de convergence en temps de **1.97**

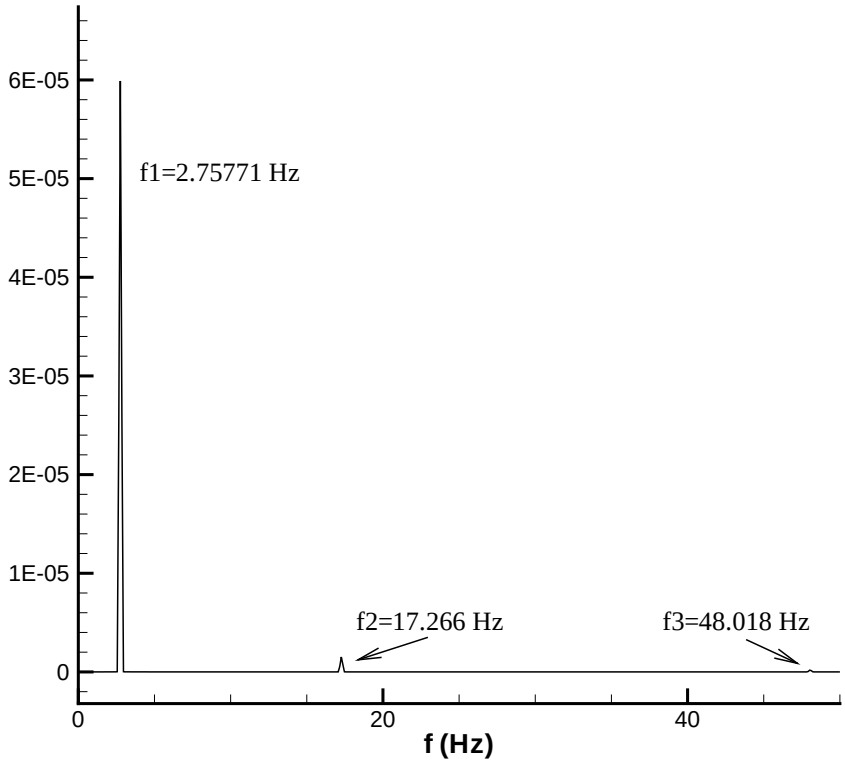


FIG. 3.7: Fréquences propres des oscillations de la poutre (modèle Euler-Bernoulli)

(sur les cinq premiers pas de temps) ce qui, comme précédemment, est du même ordre de grandeur que la valeur théorique du schéma de Newmark 2.

Δt (s)	0.09	0.07	0.05	0.01	0.005	0.001	Valeur de référence
f (Hz)	2.33918	2.47813	2.60274	2.75229	2.75689	2.75827	2.75837

TAB. 3.3: Validation dynamique pour le modèle de Rayleigh (unités SI)

Le même type de validation dynamique a été testé sur une *poutre de longueur différente de* $1\,m$. Au repos, cette dernière est maintenant suivant $\overrightarrow{x}$ ($\overrightarrow{E_1} = \overrightarrow{e_1} = \overrightarrow{x}$). La poutre a les propriétés suivantes : Longueur $L = 0.04\,m$, Épaisseur $d = 0.0006\,m$, Masse volumique $\rho_{beam} = 2000\,SI$ et Module d'Young $E = 0.2\,MPa$. On calcule la première fréquence propre théorique : $f_1 = 0.6057\,Hz$. Sur la figure 3.8 on peut voir l'évolution du point B, i.e. l'extrémité libre de la poutre. La fréquence d'oscillation se situe autour de $0.6012\,Hz$ (cf. fig. 3.9) ce qui nous donne une erreur relative inférieure à 0.7%.

On impose maintenant à la poutre une charge q constante. Le test se fait dans le vide. Pour tendre vers une déformée finale stable on ajoute un amortissement du type $-k\frac{\partial x}{\partial t}$. Quelques secondes après le lancement de la simulation, la déformée de la poutre se stabilise (cf. fig. 3.10). *La déformée finale se confond avec la déformée obtenue statiquement et celle obtenue théoriquement* (cf. fig. 3.11).

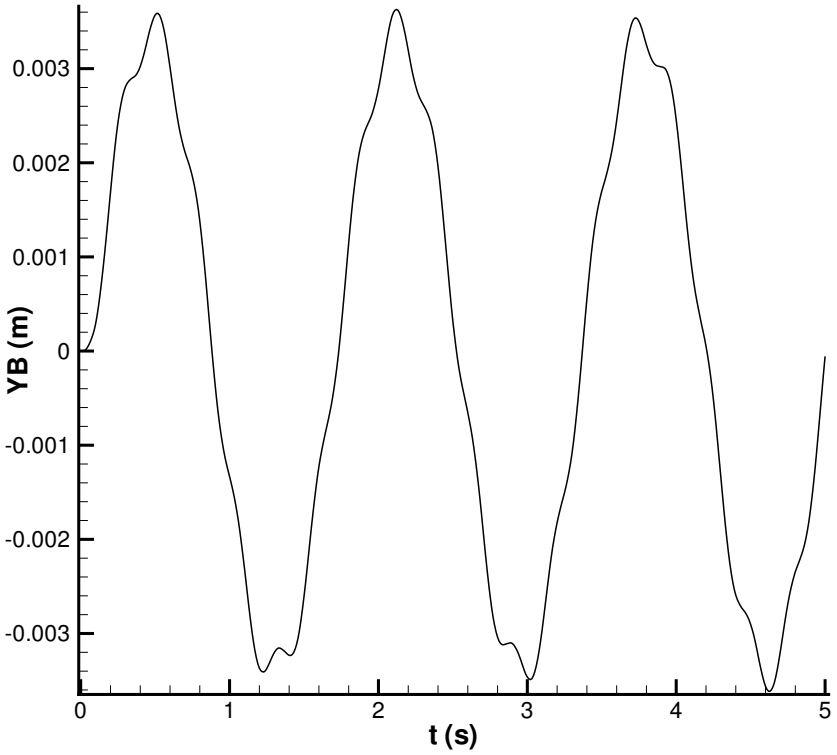


FIG. 3.8: Oscillations du point B dans le vide

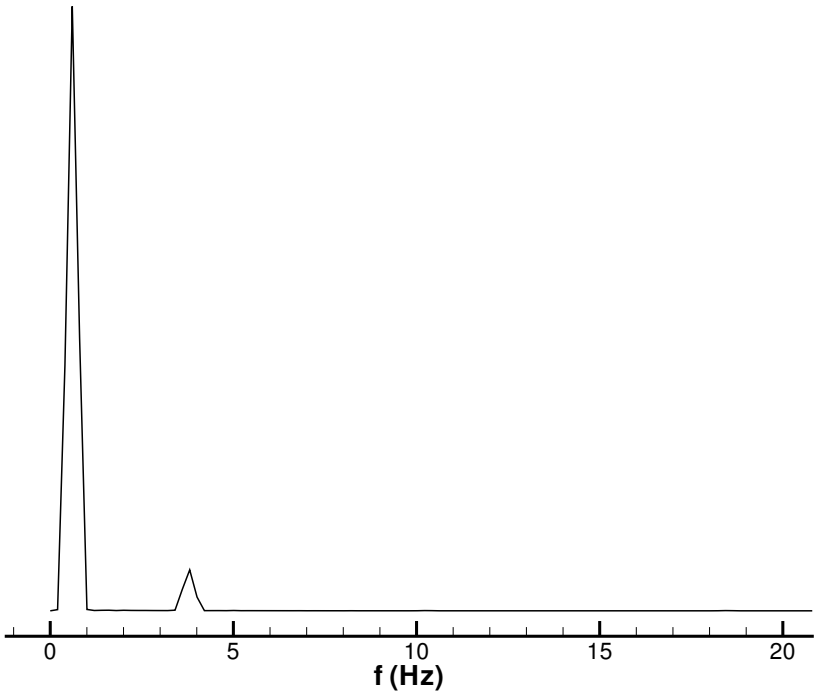


FIG. 3.9: Fréquences propres des oscillations de la poutre

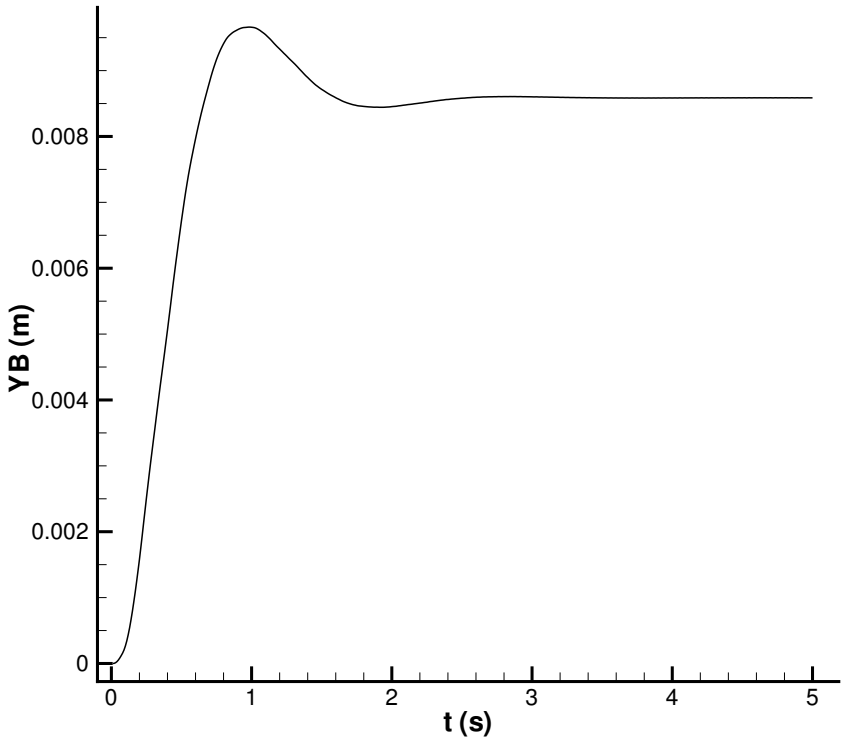


FIG. 3.10: Oscillations amorties du point B dans le vide

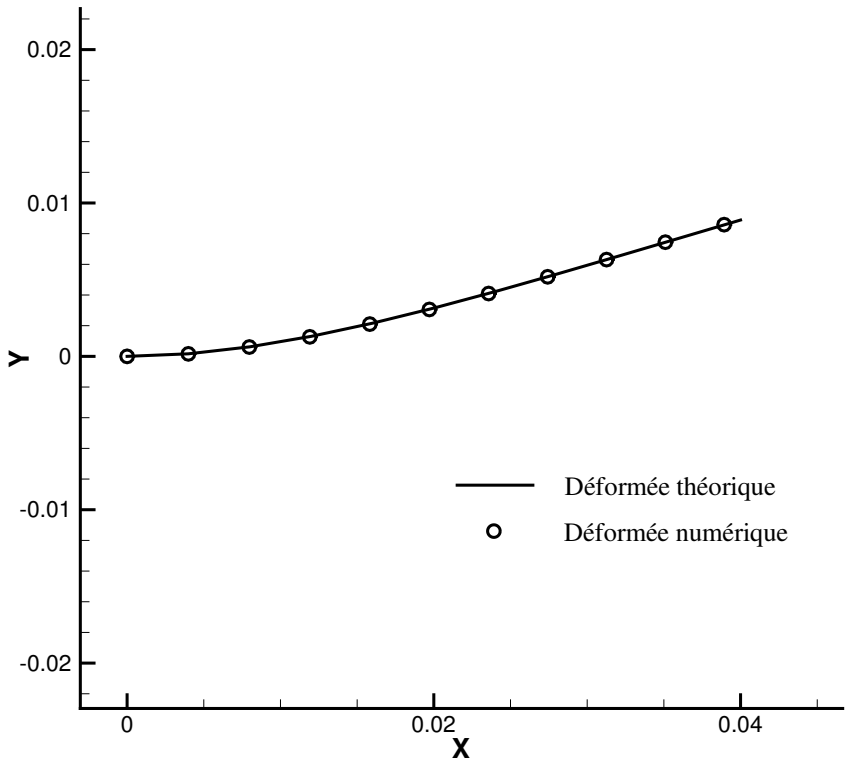


FIG. 3.11: Déformée finale après amortissement (unités SI)

Enfin, une dernière validation dynamique a été effectuée en petits déplacements. Nous avons testé le comportement de la poutre en imposant des forces. Maintenant on reprend la même configuration que précédemment mais *en imposant un couple C au lieu d'une charge q*. Quelques secondes après le lancement du calcul, la déformée de la poutre tend vers une position stable (cf. fig. 3.12). *La déformée finale se confond avec la déformée obtenue statiquement et celle obtenue théoriquement* (cf. fig. 3.13).

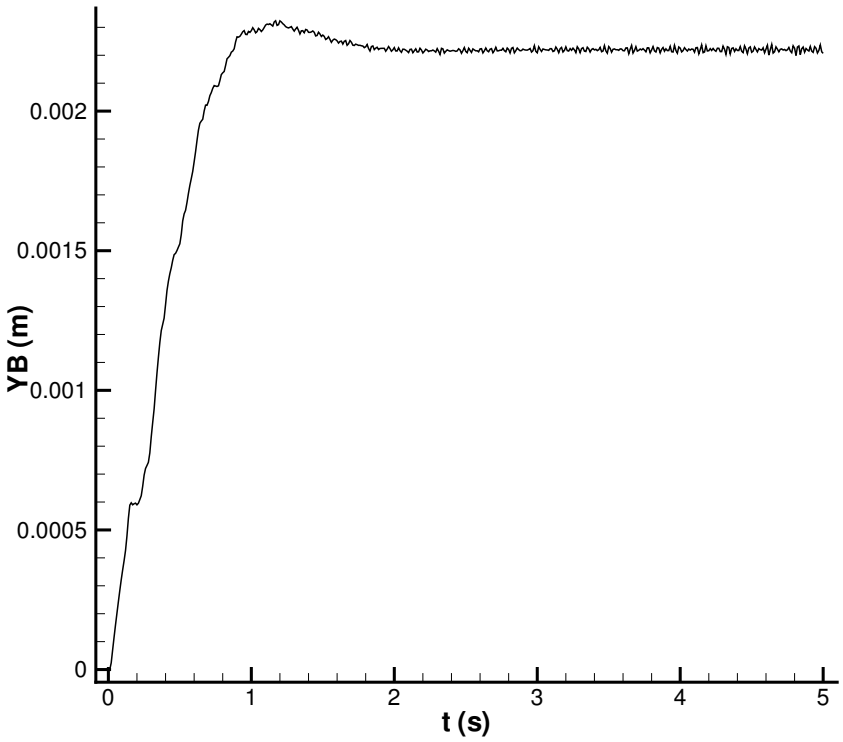


FIG. 3.12: Oscillations amorties du point B dans le vide

Ces cas-tests très simples ont permis de **vérifier l'ordre de convergence des modèles** et de **valider le solveur poutre en petit déplacement et petite déformation**. Mais le but de ce solveur est de traiter le comportement de câbles ou tubes sous-marins. *Ceux-ci doivent être étudiés sous l'hypothèse des grands déplacements.* Il faut donc valider le solveur poutre dans ce cas.

3.2.1.2 Cas des grands déplacements

Ce solveur poutre est entièrement basé sur les travaux de D. Primault et de F. Boyer. Il est donc logique de reprendre certains cas-tests présentés dans sa thèse ou dans ses publications (cf. [79], [16] et [17]).

Un cas statique bidimensionnel, proposé dans la thèse de Dominique Primault, a été testé **pour valider statiquement le solveur structure en grand déplacement** : une poutre encastrée soumise à une force ponctuelle fixe, dont une solution analytique a

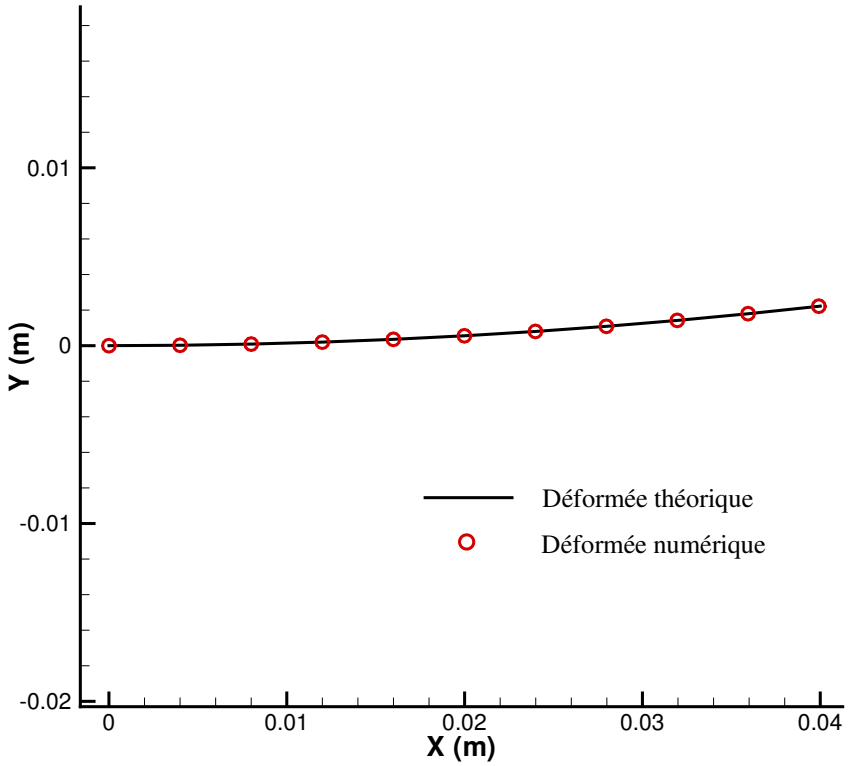


FIG. 3.13: Déformée finale après amortissement

été établie dans [14]. La poutre est encastrée en 0, libre en L et initialement suivant $\overrightarrow{x}$ ($\overrightarrow{E_1} = \overrightarrow{e_1} = \overrightarrow{x}$). Elle a les propriétés données dans le tableau 3.4.

Longueur	$L = 10\,m$
Épaisseur	$d = 0.01\,m$
Module d'Young	$E = 210000\,MPa$
Coefficient de Poisson	$\nu = 0.3$

TAB. 3.4: Conditions du cas-test proposé par D. Primault (unités SI)

Le maillage structure est de 10 points et la force appliquée à l'extrémité libre de la poutre est de $14\,N$ pour correspondre au test présenté dans la thèse de D. Primault. Le résultat (cf. fig. 3.14) concorde en tout point avec celui trouvé dans cette thèse : la déformation est atteinte en 1 pas de temps avec 31 itérations non-linéaires. *L'accord avec la solution analytique donnée dans [14] est très bonne.*

Une **validation dynamique bidimensionnelle en grand déplacement**, faisant intervenir des efforts imposés, a été mise en place à partir des travaux de D. Primault. On prend une poutre encastrée, initialement suivant $\overrightarrow{y}$ ($\overrightarrow{E_1} = \overrightarrow{e_2} = \overrightarrow{y}$), et on lui applique, en son extrémité libre, une force ponctuelle, de direction $\overrightarrow{x} = \overrightarrow{e_1} = -\overrightarrow{E_2}$, dépendant d'une loi horaire. Les caractéristiques de la poutre sont une longueur de $L = 3.048\,m$, une section de $A = 0.0141\,m^2$, une masse volumique $\rho = 48771\,kg.m^{-3}$, un moment quadratique de

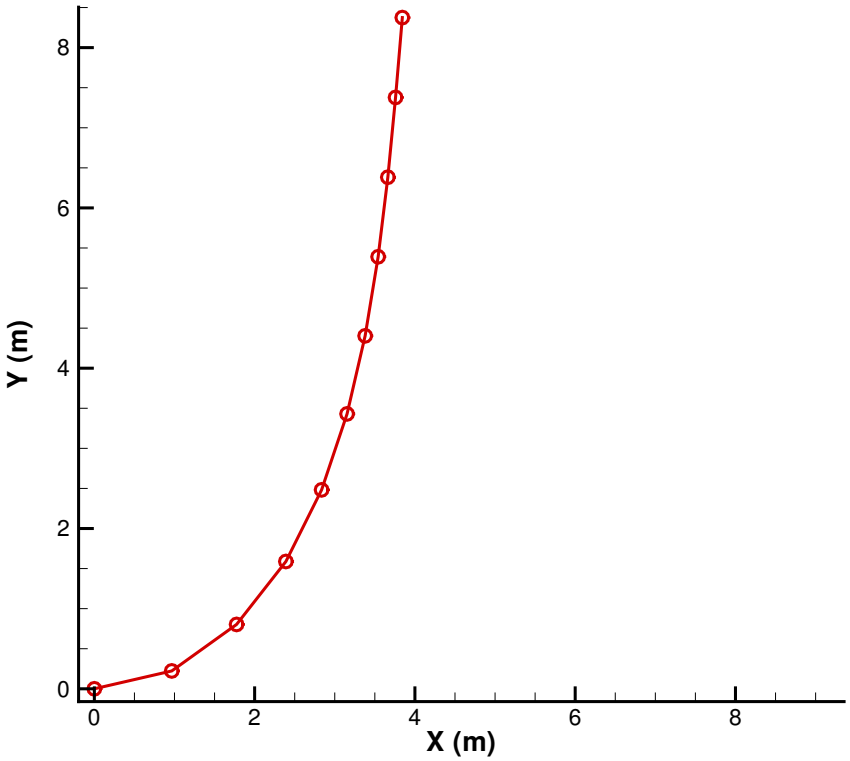


FIG. 3.14: Déformée calculée numériquement ($F = 14\,SI$)

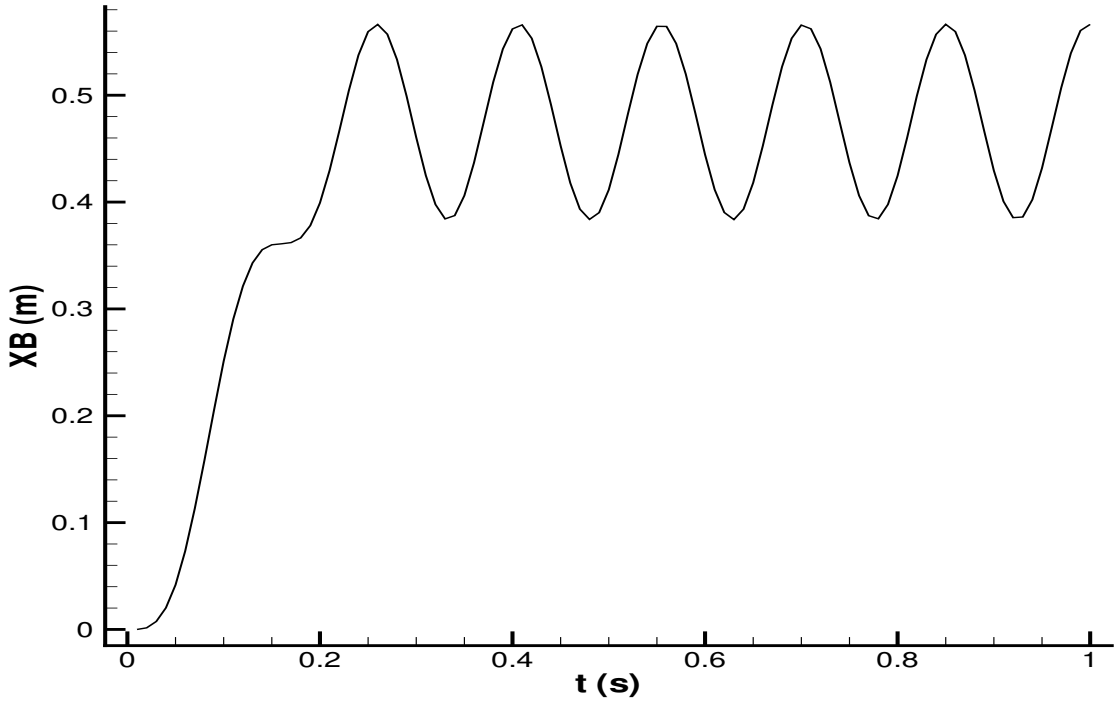
$I_a = 4.162310^{-5}\,m^4$ et un module d'Young de $E = 20.688 \times 10^{10} N\,m^{-2}$. Nous imposons deux lois horaires différentes pour la force appliquée à la poutre :

- ➢ La première loi F_1 est linéairement croissante jusqu'à $t = 0.2\,s$ où elle atteint une valeur de $444900\,N$. Elle reste égale à cette valeur par la suite.
- ➢ La seconde loi F_2 est linéairement croissante jusqu'à $t = 0.2\,s$ où elle atteint une valeur de $444900\,N$, puis linéairement décroissante jusqu'à $t = 0.4\,s$ où elle s'annule. Elle reste égale à zéro par la suite.

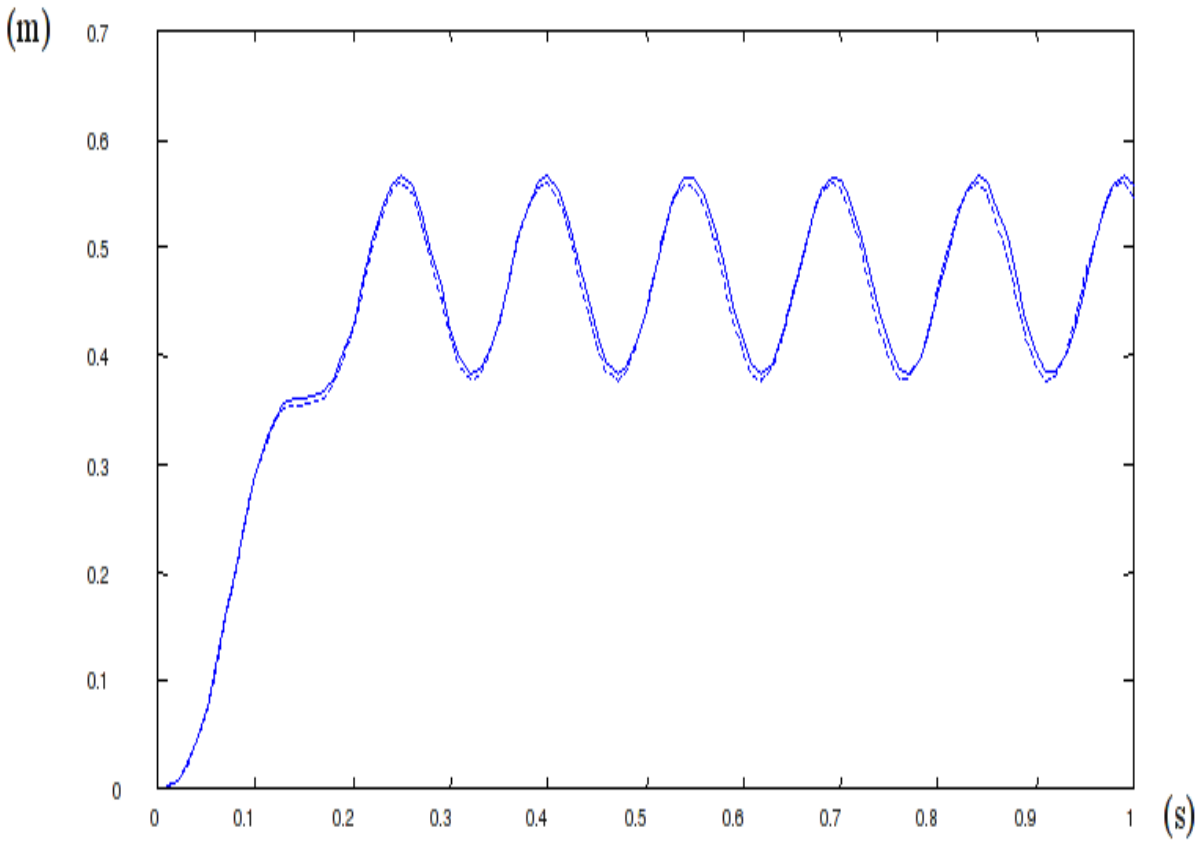
Afin de comparer les résultats avec ceux obtenus dans la thèse de D. Primault, on se place dans les mêmes conditions : le maillage poutre est composé de 4 éléments et le pas de temps est de $0.01\,s$ pour la première loi et de $0.005\,s$ pour la seconde. Les résultats sont présentés sur les figures 3.15(a) et 3.16(a). *Nous obtenons les mêmes courbes que D. Primault, qui les a comparées avec succès avec celles obtenues par Behdinan dans [12]* (cf. fig. 3.15(b) et 3.16(b)).

3.2.2 Validation en 3D

Une fois le solveur structure validé sur des exemples bidimensionnels, il est nécessaire de tester des applications tridimensionnelles. En effet le **phénomène de torsion** a été occulté jusqu'à présent et donc n'a pas été validé. De même que pour la validation 2D en grand déplacement, les cas-tests sont tirés des travaux de F. Boyer et D. Primault.

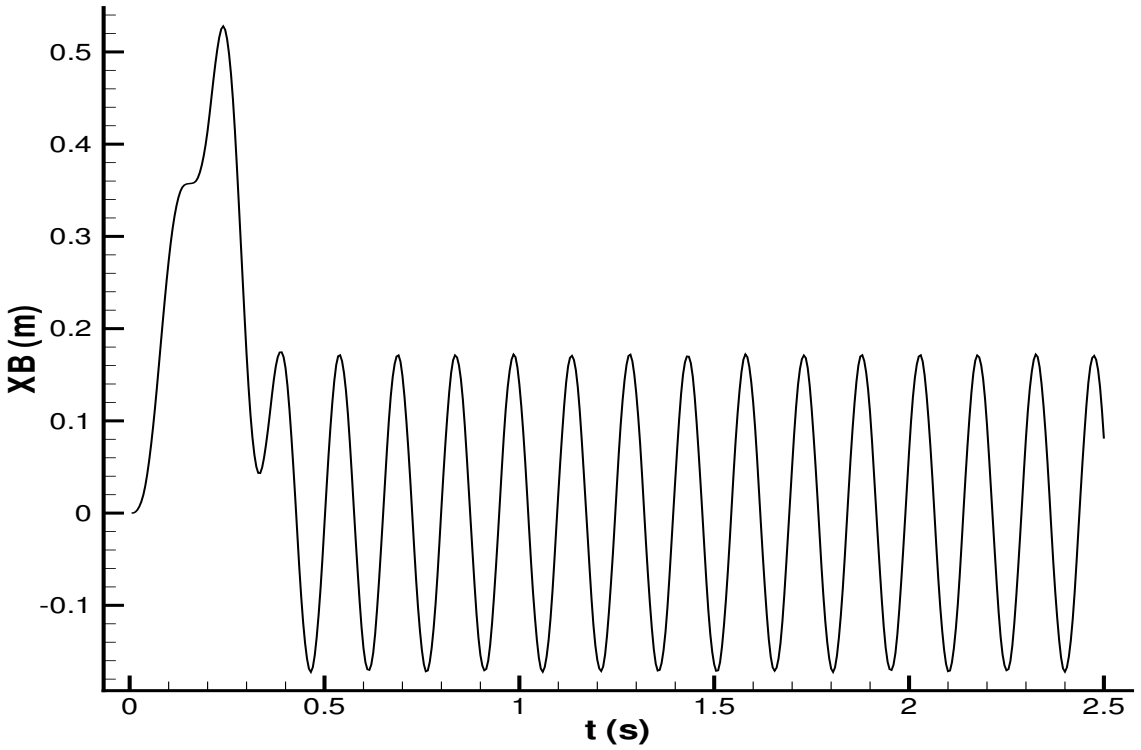


(a) Résultats obtenus avec le solveur poutre

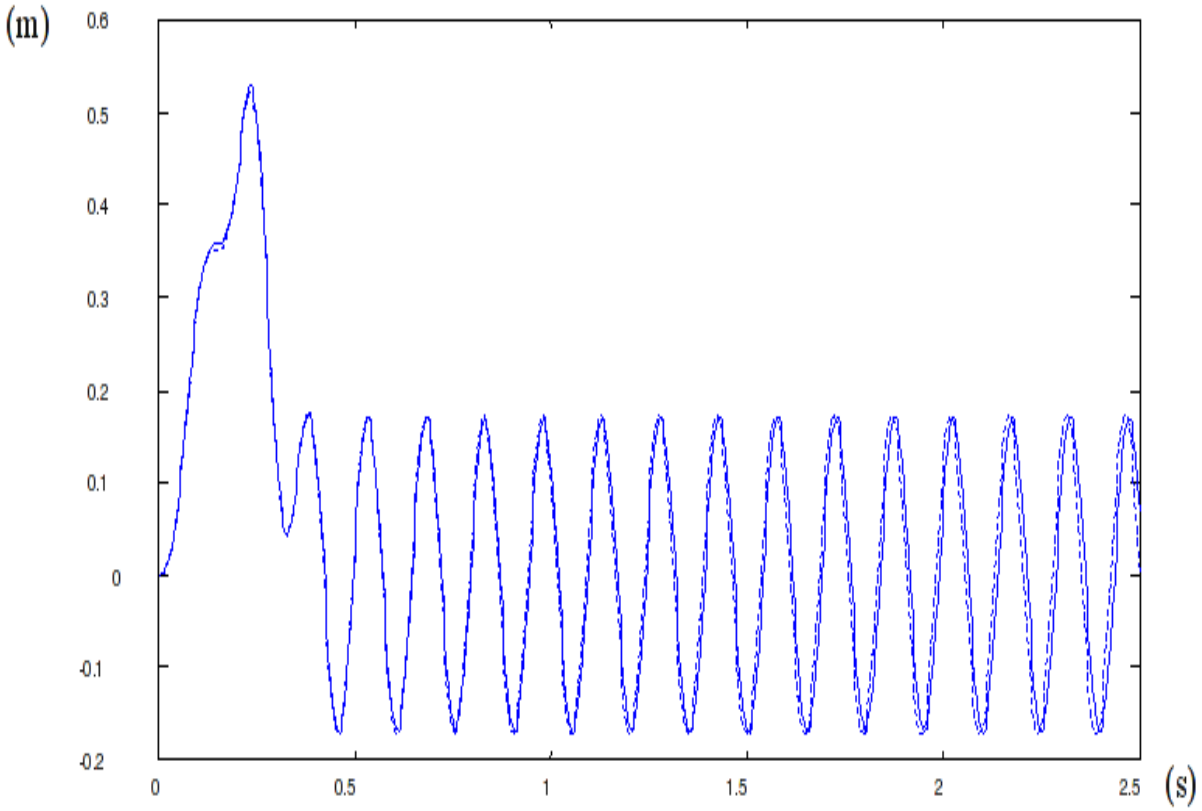


(b) Comparaison des résultats de D. Primault avec ceux de Behdinan

FIG. 3.15: Oscillations du point B dans le vide imposées par la première loi horaire



(a) Résultats obtenus avec le solveur poutre



(b) Comparaison des résultats de D. Primault avec ceux de Behdinan

FIG. 3.16: Oscillations du point B dans le vide imposées par la seconde loi horaire

3.2.2.1 Validation du couplage flexion-torsion

Le **premier test tridimensionnel** *consiste en la déformation d'une poutre encastrée soumise à un fort couple en son extrémité libre.* Au repos, cette dernière est suivant $\overrightarrow{z}$ ($\overrightarrow{E_1} = \overrightarrow{e_3} = \overrightarrow{z}$). Voici les caractéristiques de la poutre prises par D. Primault dans son cas-test (cf. tab. 3.5) :

Longueur	$L = 100\,m$
Raideur de traction-compression	$EA = 4.2056 \times 10^8$
Raideur de torsion	$GI_p = 10^6$
Raideur de flexion	$EI_a = 3.5 \times 10^7$

Tab. 3.5: Conditions du cas-test proposé par D. Primault (unités SI)

Le couple appliqué est le suivant : $\overrightarrow{C} = (3 \times 10^6)\overrightarrow{E_1} + (1 \times 10^6)\overrightarrow{E_2} + (0.5 \times 10^6)\overrightarrow{E_3}$. Pour résoudre le problème, le modèle d'Euler-Bernoulli a été sélectionné de manière à comparer nos résultats avec ceux de D. Primault. Plusieurs maillages structures ont été testés : 10, 20, 30 et 40 éléments finis. La convergence en maillage est atteinte dès 10 éléments. Cependant les résultats présentés ont été obtenus avec un maillage à 20 éléments. Cela permet d'obtenir des images moins hachées de la déformée. Le solveur converge en 10 itérations non-linéaires et donne *exactement* les mêmes déformées que celles présentées dans la thèse de D. Primault (cf. fig. 3.17, 3.18, 3.19 et 3.20).

Après cette validation tridimensionnelle du solveur poutre par imposition d'un couple, intéressons nous au couplage flexion-torsion du modèle en imposant des mouvements aux extrémités de la poutre. Ce **deuxième cas-test** a aussi été traité dans la thèse de D. Primault. Cette dernière obtient de bons résultats qui concordent avec ceux de Le Tallec issus de [55].

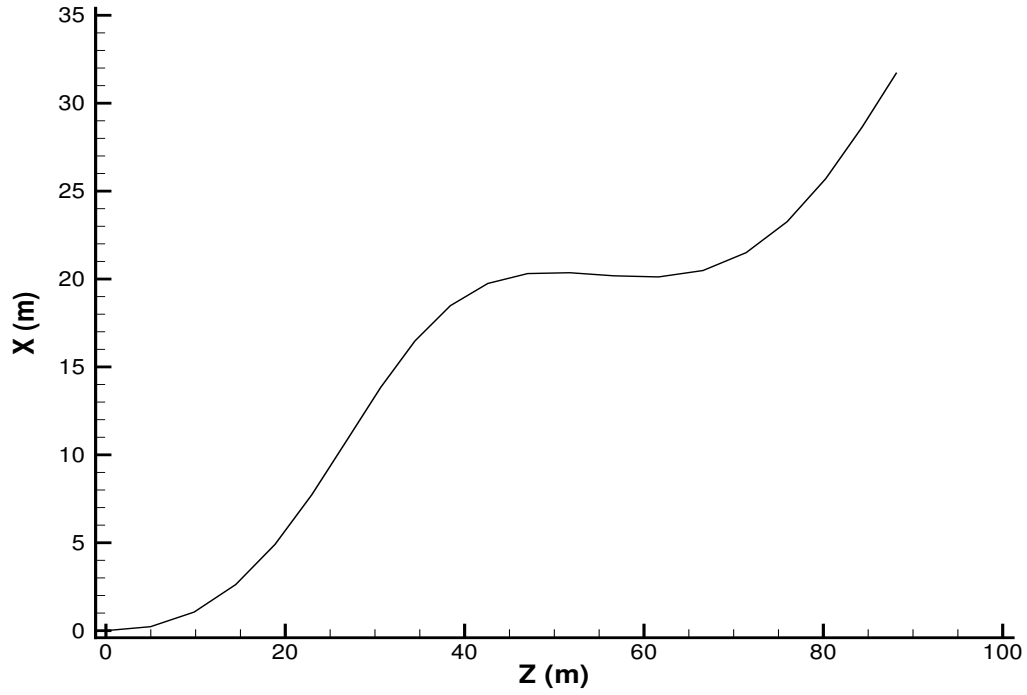
Le test de Le Tallec est destiné à étudier le couplage flexion-torsion, qui est très présent dans la déformation des câbles et tuyaux flexibles. La poutre utilisée possède les propriétés réunies dans le tableau 3.6.

Longueur	$L = 20\,m$
Masse linéique	$\mu = 5.0968\,kg.m^{-1}$
Raideur de traction-compression	$EA = 5.137 \times 10^7$
Raideur de torsion	$GI_p = 1.8 \times 10^3$
Raideur de flexion	$EI_a = 10^3$

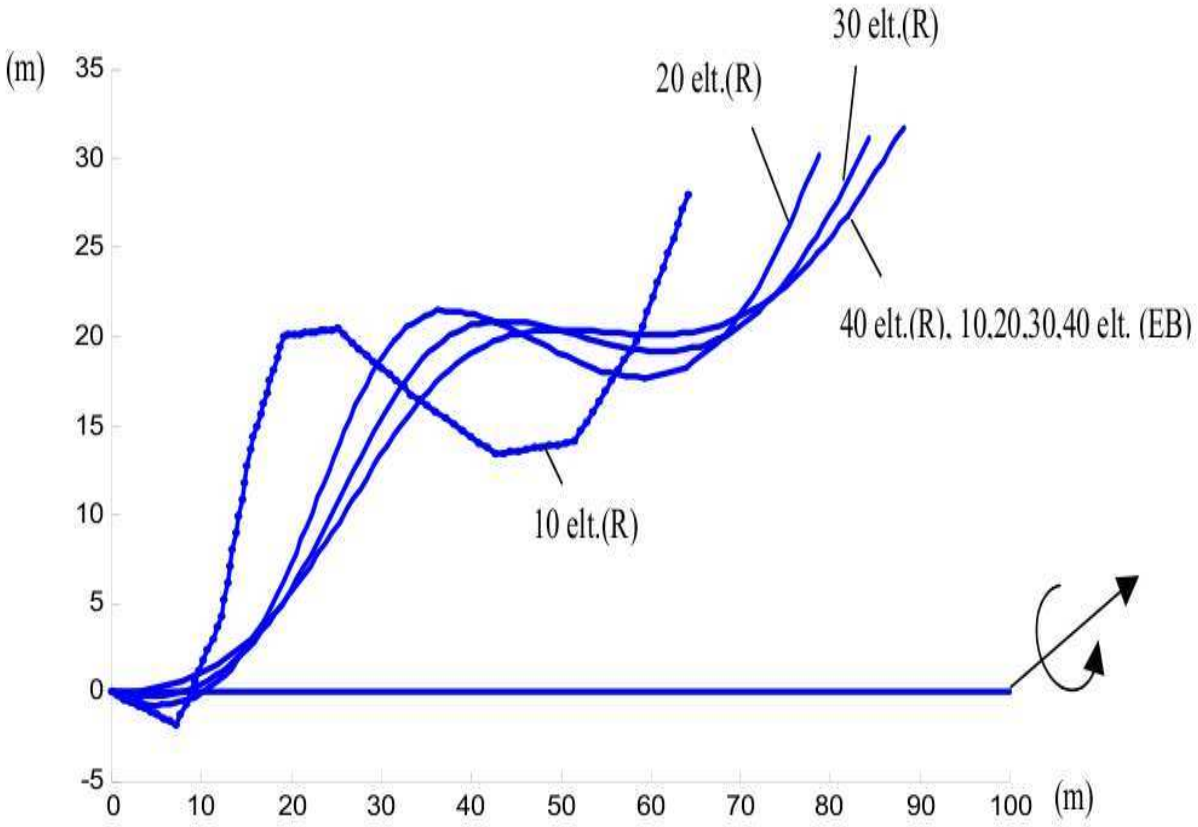
Tab. 3.6: Conditions du cas-test proposé par D. Primault (unités SI)

Cette poutre est initialement au repos et droite suivant $\overrightarrow{x}$ ($\overrightarrow{E_1} = \overrightarrow{e_1} = \overrightarrow{x}$). Elle est plongée dans un champ de gravité vertical. On impose à ses deux extrémités les mouvements quasi-statiques présentés dans le tableau 3.7.

On note que l'extrémité en 0 reste fixe tout au long du test. L'autre extrémité est tout d'abord soumise à une forte rotation de 2π, puis en maintenant cette rotation, elle se rapproche peu à peu de l'extrémité fixe (cf. fig. 3.21, 3.22, 3.23 et 3.24).

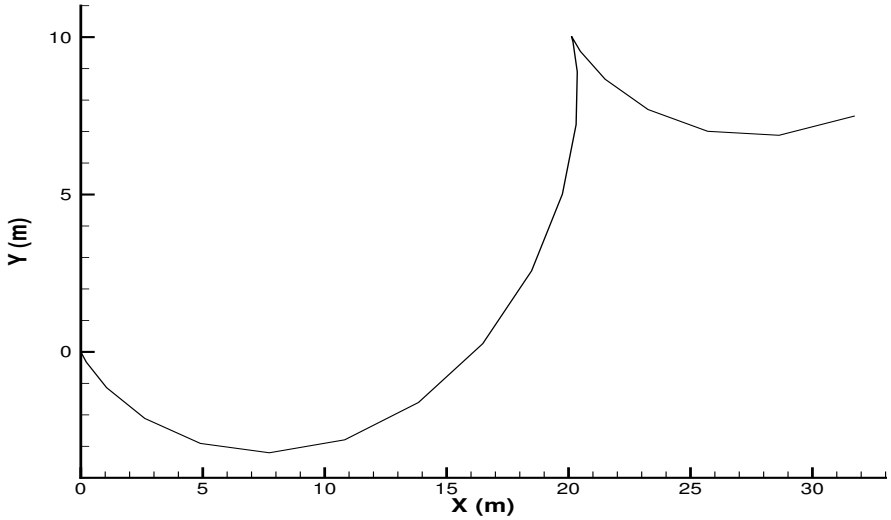


(a) Résultats obtenus avec le solveur poutre

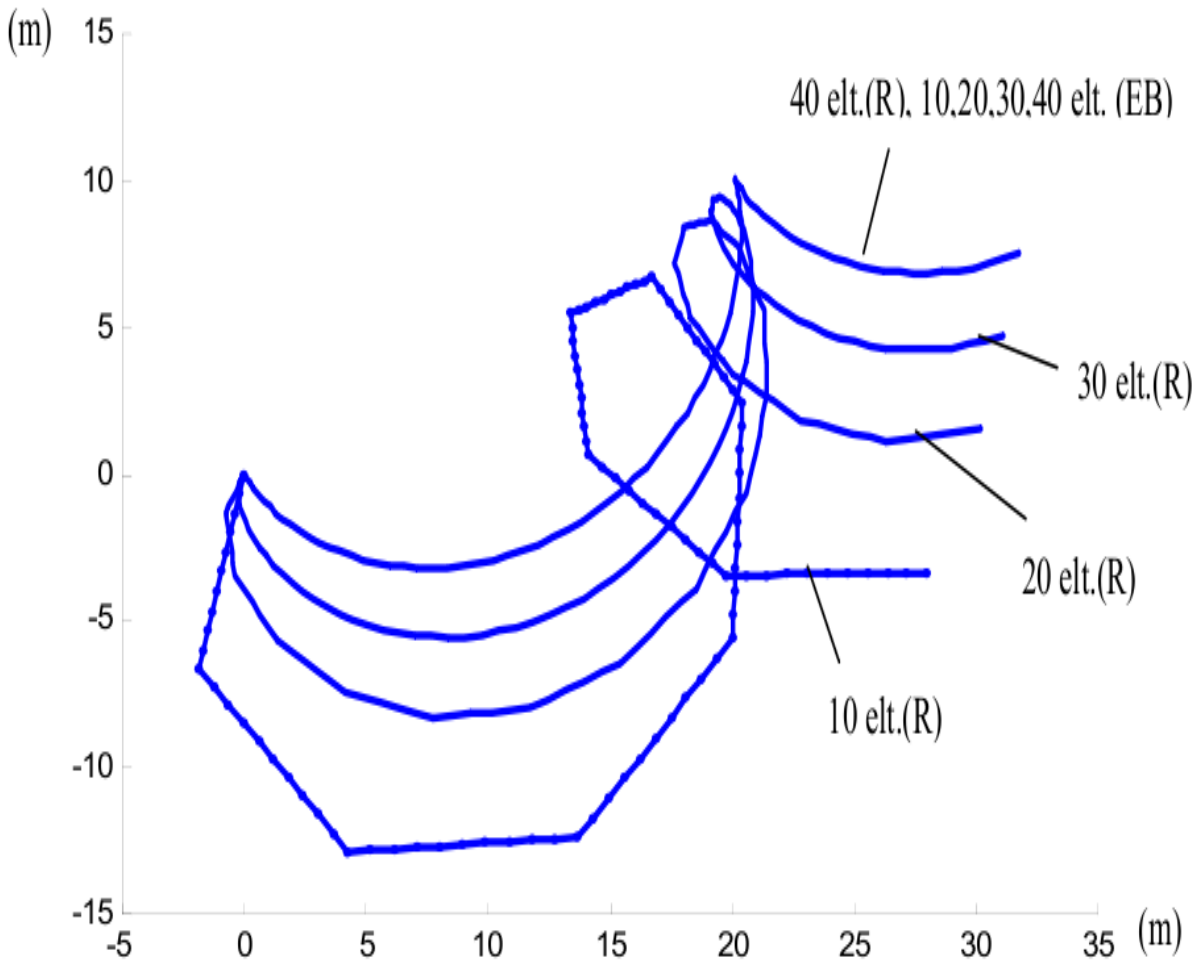


(b) Résultats de D. Primault

FIG. 3.17: Déformée 3D projetée dans le plan $(\overrightarrow{E_1}, \overrightarrow{E_2})$

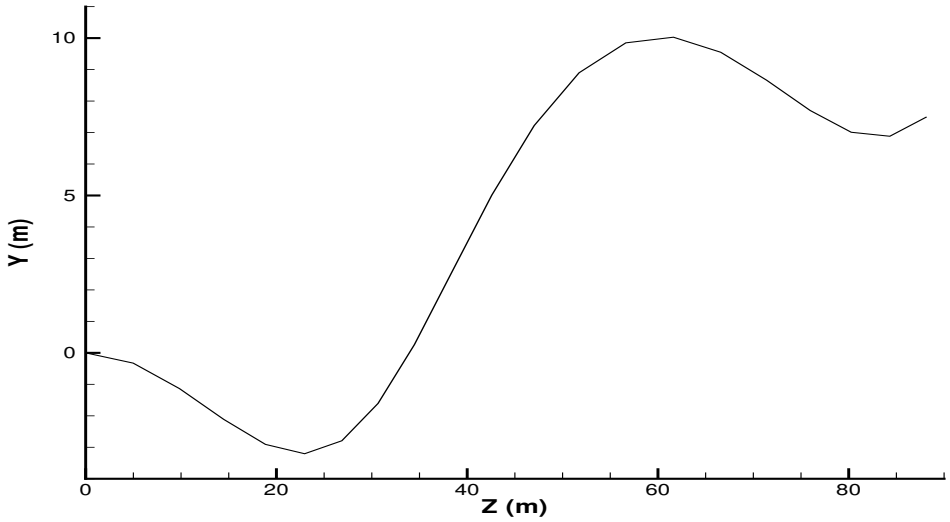


(a) Résultats obtenus avec le solveur poutre

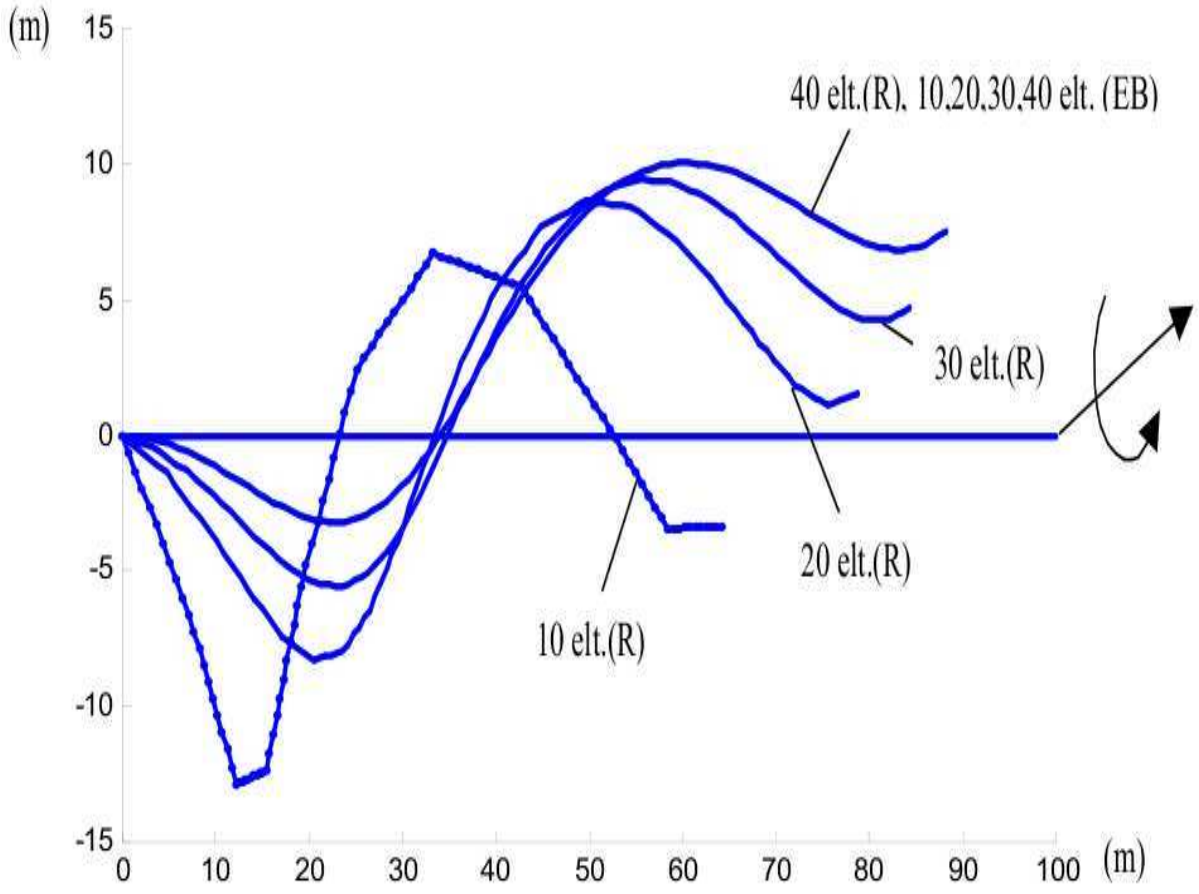


(b) Résultats de D. Primault

FIG. 3.18: Déformée 3D projetée dans le plan $(\overrightarrow{E_2}, \overrightarrow{E_3})$



(a) Résultats obtenus avec le solveur poutre



(b) Résultats de D. Primault

FIG. 3.19: Déformée 3D projetée dans le plan $(\overrightarrow{E_1}, \overrightarrow{E_3})$

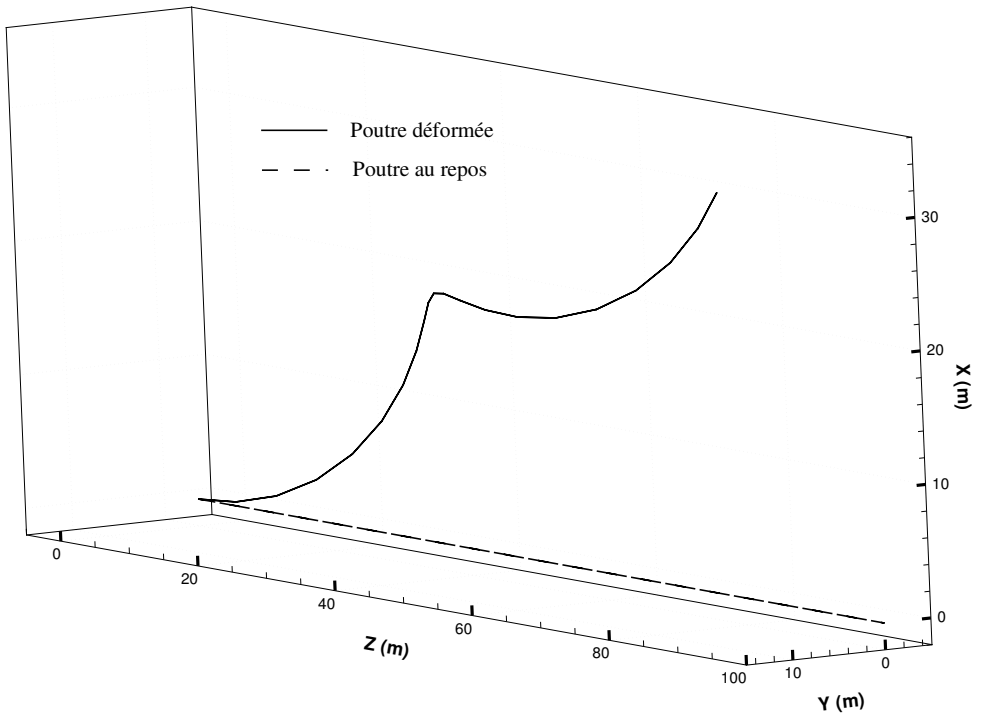


FIG. 3.20: Poutre au repos et déformée 3D

Mouvements imposés aux bords	Valeurs spécifiées
$r^T(0)$	$(0, 0, 0)$
$r^T(L)$	$(\alpha, 0, 0)$ avec $\alpha = 20, 15, 10, 5, 2, 0, -2, -5$
$r'^T(0)$	$(1, 0, 0)$
$r'^T(L)$	$(1, 0, 0)$
$\theta(0)$	0
$\theta(0)$	2π

TAB. 3.7: Mouvements imposés aux extrémités (unités SI)

Pour rester fidèle au cas-test de D. Primault la poutre est divisée en 20 éléments finis de $1\,m$. Le cas-test est statique et utilise le modèle d'Euler-Bernoulli. Pour que l'algorithme de Newton converge le mouvement est imposé sur 10 pas, sauf dans la première configuration où cela n'est pas nécessaire.

Ce qui permet de valider le couplage flexion-torsion est la comparaison de la moyenne du taux de torsion pour les différentes configurations avec les valeurs trouvées par Le Tallec dans [55]. *Nous retrouvons avec notre programme les valeurs présentées par D. Primault dans sa thèse* (cf. tab. 3.8).

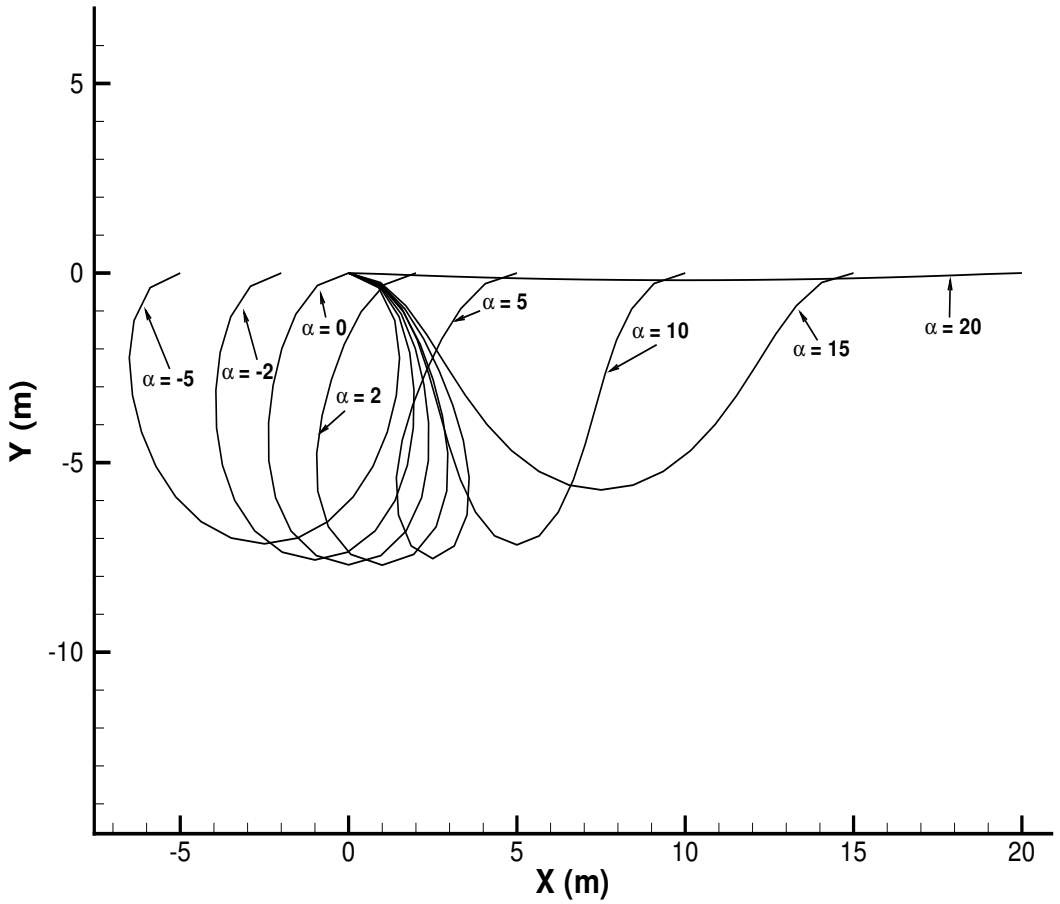
$\alpha(m)$ \ $X(m)$	0	5	10	15	20
20	0.3142	0.3142	0.3142	0.3142	0.3142
	0.3126	**0.3129**	**0.3129**	**0.3129**	**0.3126**
15	0.2809	0.2809	0.2809	0.2809	0.2809
	0.2793	**0.2798**	**0.2781**	**0.2798**	**0.2793**
10	0.2140	0.2140	0.2140	0.2140	0.2140
	0.2132	**0.2139**	**0.2088**	**0.2134**	**0.2133**
5	0.0615	0.0615	0.0615	0.0615	0.0615
	0.1024	**0.0927**	**0.0662**	**0.0882**	**0.1025**
2	0.0077	0.0077	0.0077	0.0077	0.0077
	0.0255	**0.0166**	**-0.0036**	**0.0126**	**0.0255**
0	0.0032	0.0032	0.0032	0.0032	0.0032
	0.0030	**0.0014**	**-0.0018**	**0.0007**	**0.0030**
-2	0.0013	0.0013	0.0013	0.0013	0.0013
	0.0002	**0.0001**	**-0.0002**	**0.0001**	**0.0002**
-5	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	**0.0000**	**0.0000**	**0.0000**	**0.0000**

TAB. 3.8: Comparaison du taux de torsion moyen obtenu sur ce cas-test avec celui de Le Tallec (en gras) (unités SI)

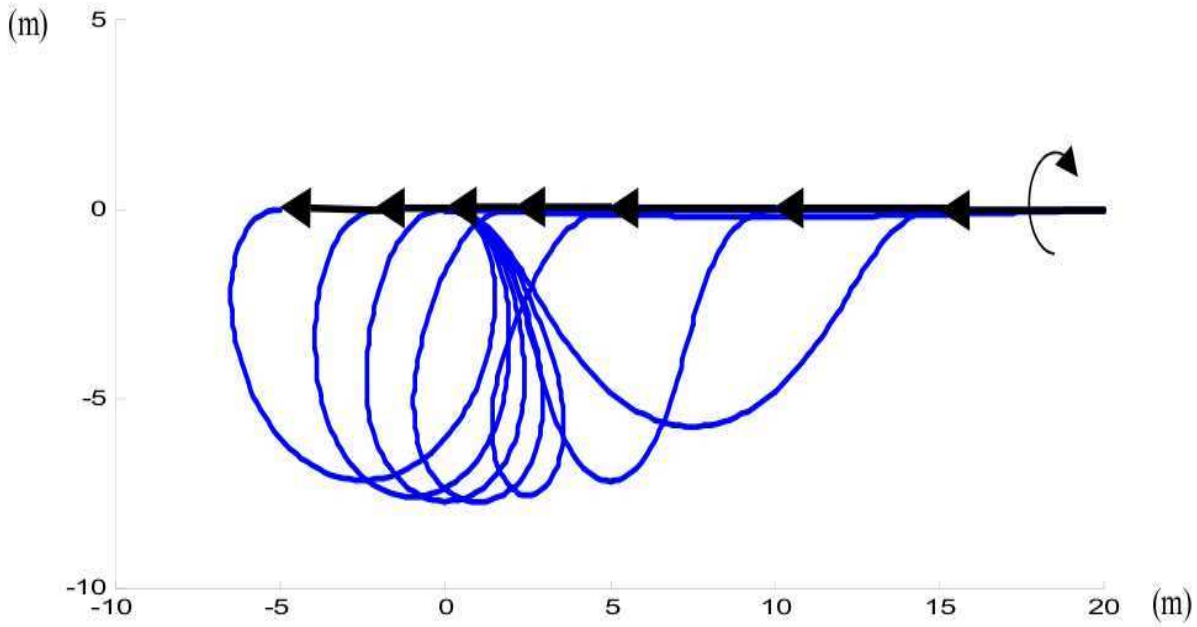
3.2.2.2 Validation du modèle de Rayleigh en 3D

Enfin un **dernier cas-test tridimensionnel** est nécessaire pour valider le modèle de Rayleigh. Pour justifier l'usage de ce modèle, le cas-test doit présenter la singularité dynamique de torsion. En d'autres termes, *la poutre doit être libre de tourner sur elle-même suivant son axe matériel*. Le cas-test est donc une poutre "volante" soumise à des forces et des moments variant au cours du temps.

Les caractéristiques physiques de la poutre sont données dans le tableau 3.9. À $t = 0\,s$ la poutre est droite et inclinée dans le plan $(\overrightarrow{x}, \overrightarrow{y})$. Elle est soumise au cours du temps en son extrémité $X = L$ à une force $\overrightarrow{F(t)}$, portée par $\overrightarrow{x}$, et à deux couples $\overrightarrow{C_2(t)}$, suivant

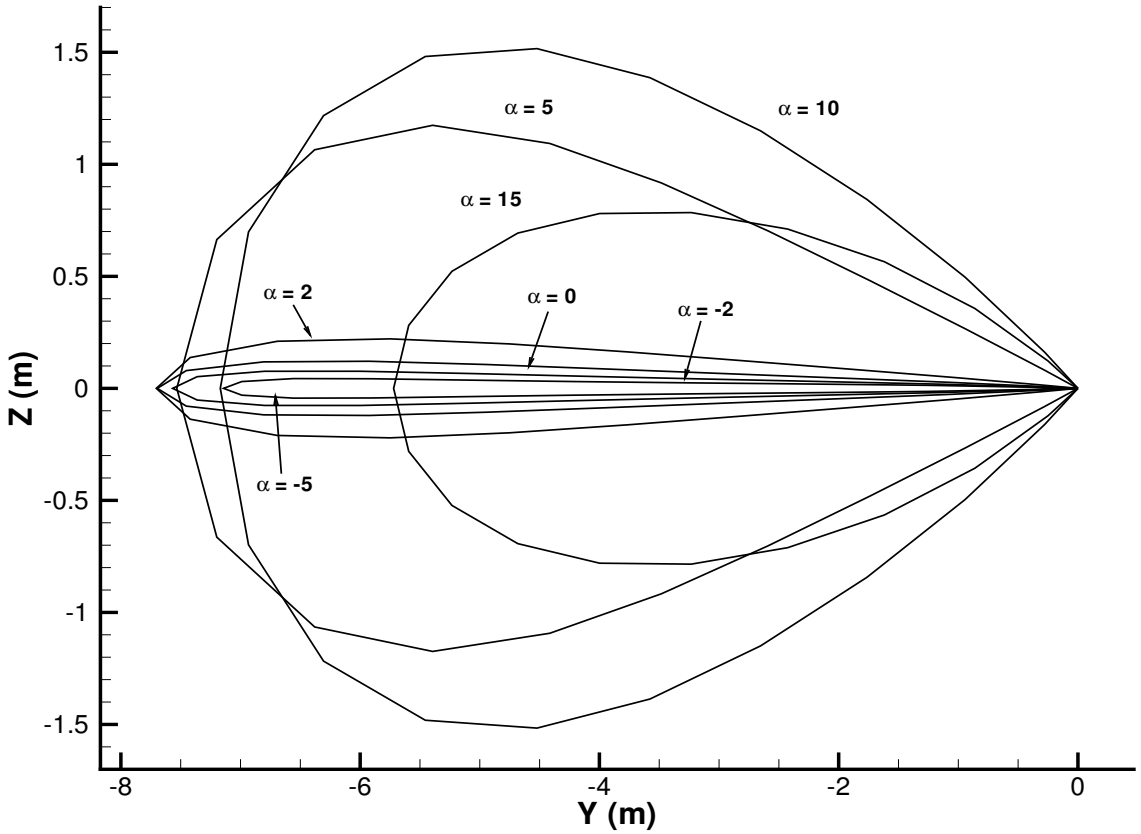


(a) Résultats obtenus avec le solveur poutre

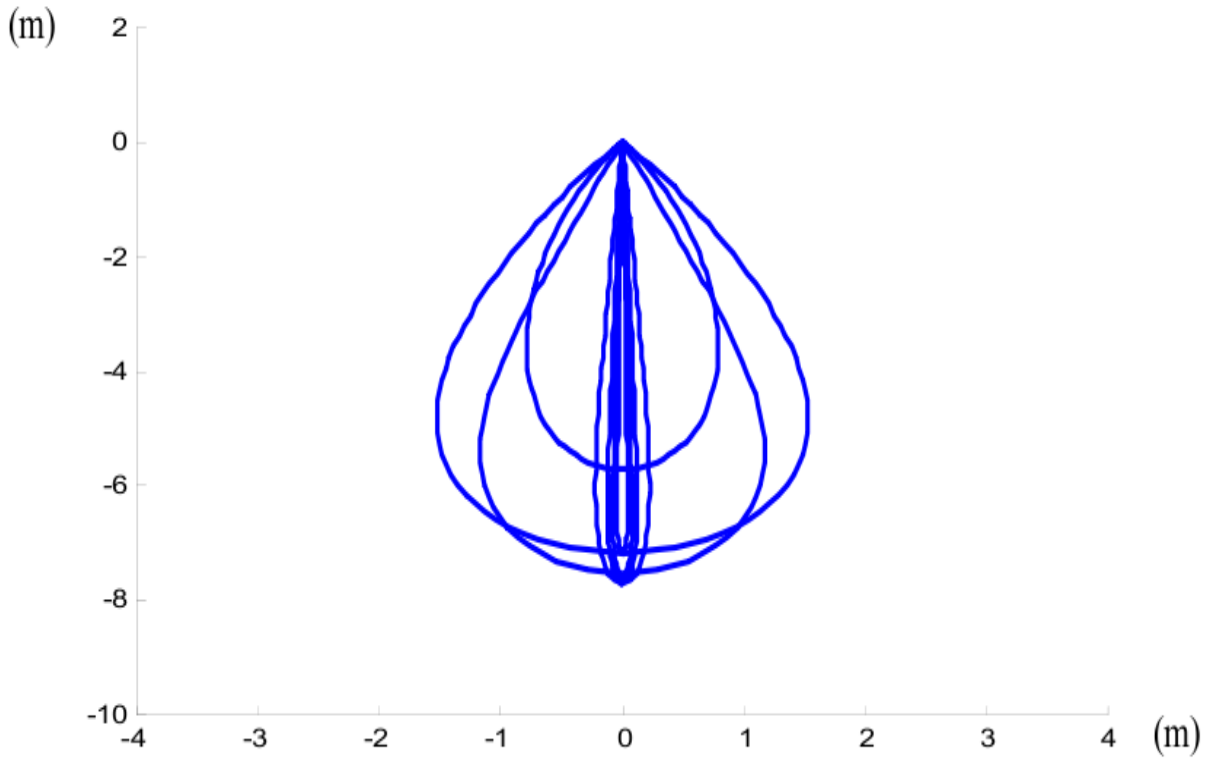


(b) Résultats de D. Primault

FIG. 3.21: Déformée 3D projetée dans le plan $(\overrightarrow{x}, \overrightarrow{y})$

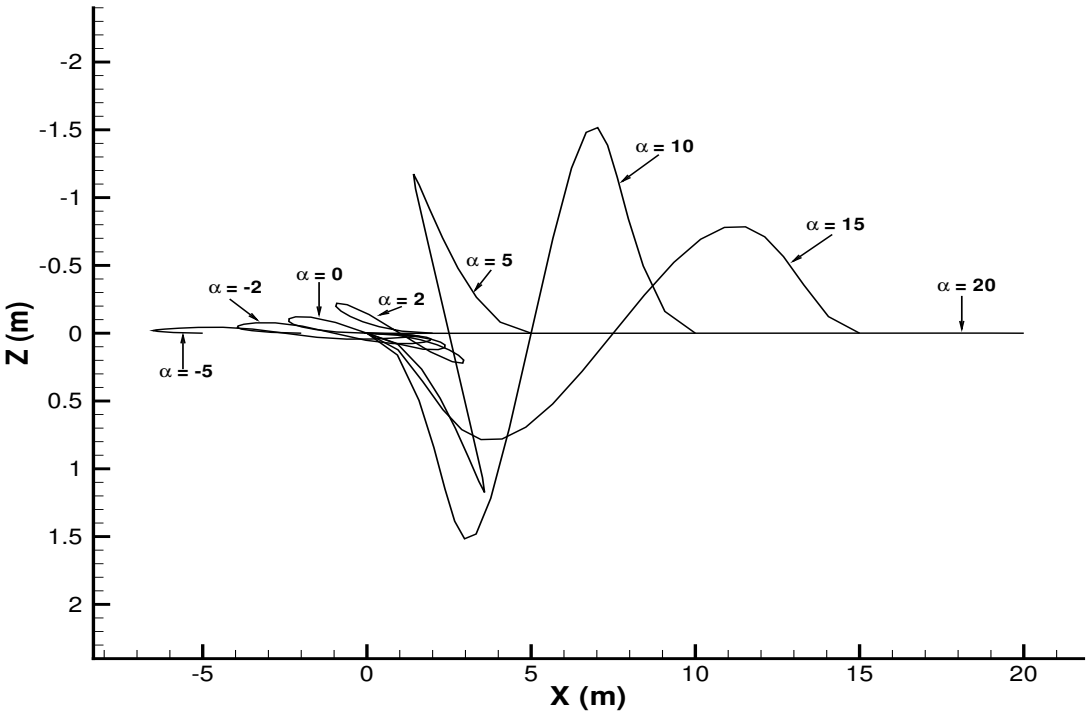


(a) Résultats obtenus avec le solveur poutre

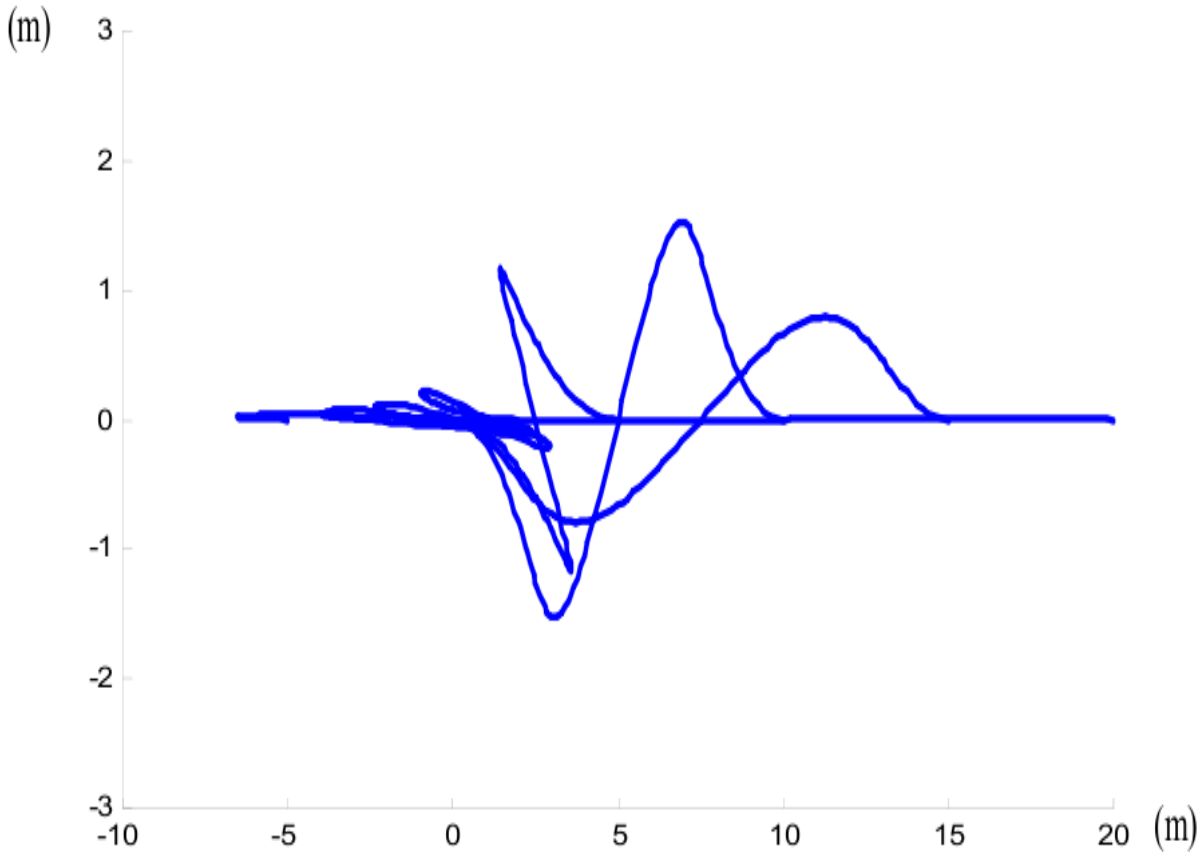


(b) Résultats de D. Primault

FIG. 3.22: Déformée 3D projetée dans le plan $(\overrightarrow{y}, \overrightarrow{z})$



(a) Résultats obtenus avec le solveur poutre



(b) Résultats de D. Primault

FIG. 3.23: Déformée 3D projetée dans le plan $(\vec{x}, \vec{z})$

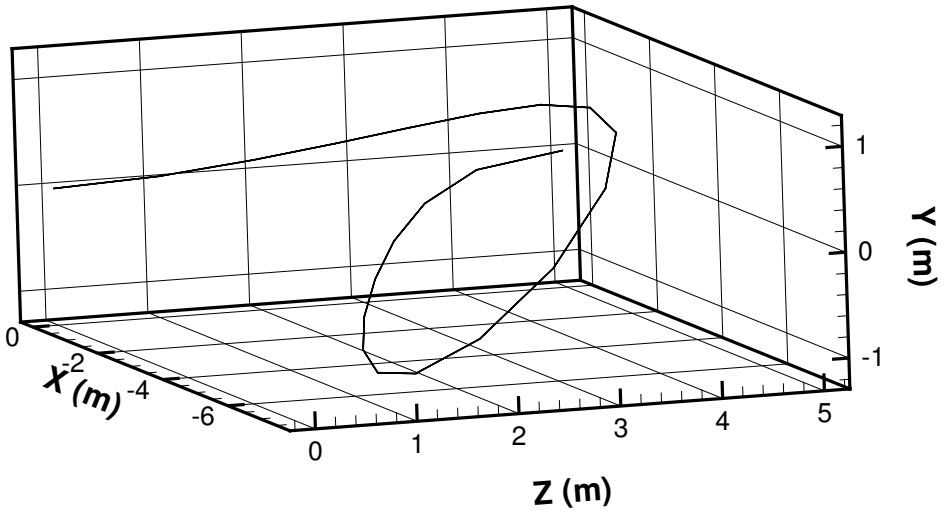


FIG. 3.24: Déformée 3D de la poutre pour $\alpha = 5$

$\overrightarrow{y}$, et $\overrightarrow{C_3(t)}$, suivant $-\overrightarrow{z}$ (cf. fig. 3.25). La poutre est modélisée par 10 éléments finis et la simulation dure $15\,s$ avec un pas de temps de $\Delta t = 0.01\,s$.

Longueur	$L = 10\,m$
Masse linéique	$\mu = 1\,kg.m^{-1}$
Raideur de traction-compression	$EA = 1 \times 10^4$
Raideur de torsion	$GI_p = 5 \times 10^2$
Raideur de flexion	$EI_a = 5 \times 10^2$
Tenseur d'inertie	$J = diag(20, 10, 10)$

TAB. 3.9: Conditions du cas-test proposé par D. Primault (unités SI)

Dans son cas-test D. Primault compare les résultats obtenus avec son modèle de Rayleigh avec celui de Reissner, pour ainsi valider le premier. Les erreurs relatives entre les deux modèles n'excèdent pas 1%. D. Primault valide donc son modèle. Je vais de mon côté comparer mes résultats avec les siens pour valider l'implémentation du solveur poutre dans **ISIS**. Lors de notre test dynamique le solveur poutre a besoin d'au plus 5 itérations non-linéaires pour converger (résidu inférieur à 10^{-5}). En comparant nos résultats avec ceux de D. Primault, nous obtenons le même comportement de la poutre. *En mettant côte à côte les fichiers des positions des nœuds de la poutre, les* 8 *premiers chiffres significatifs sont identiques.* Les projections dans les plans $(\overrightarrow{x}, \overrightarrow{y})$, $(\overrightarrow{y}, \overrightarrow{z})$ et $(\overrightarrow{x}, \overrightarrow{z})$ de la poutre

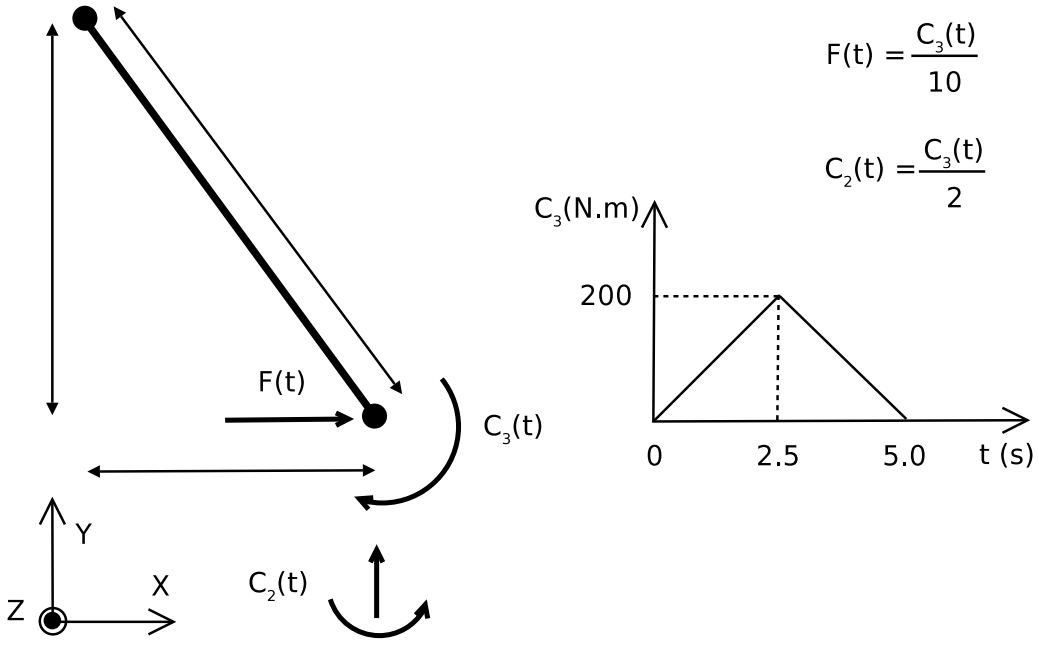
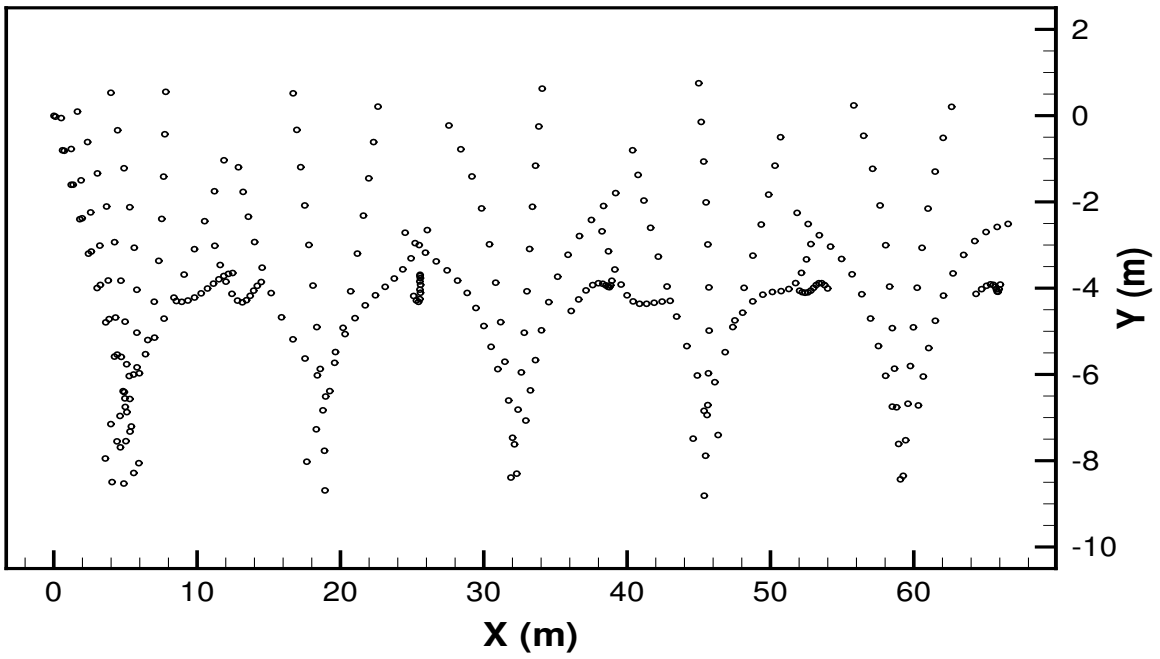
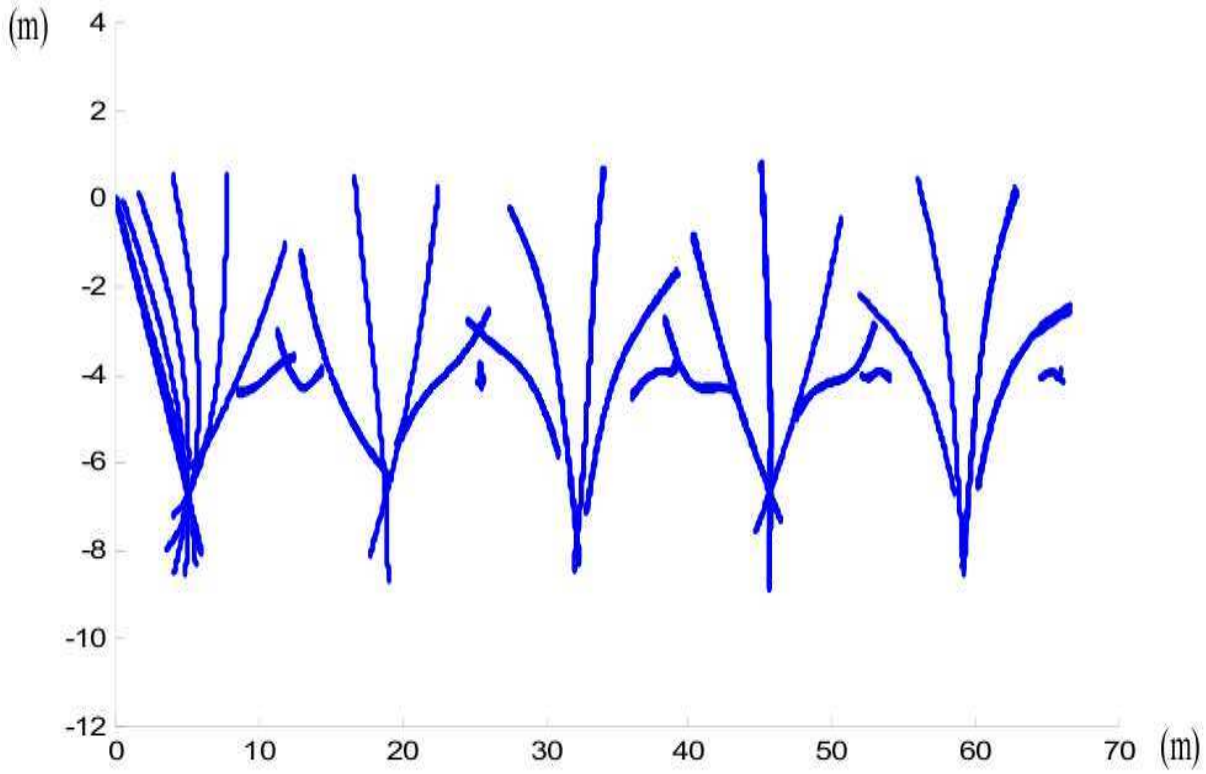


FIG. 3.25: Configuration initiale de la poutre volante

au cours du temps sont tracées ci-dessous (cf. fig. 3.26, 3.27 et 3.28), puis sa trajectoire (cf. fig. 3.29) et le déplacement de ses extrémités (cf. fig. 3.30 et 3.31).

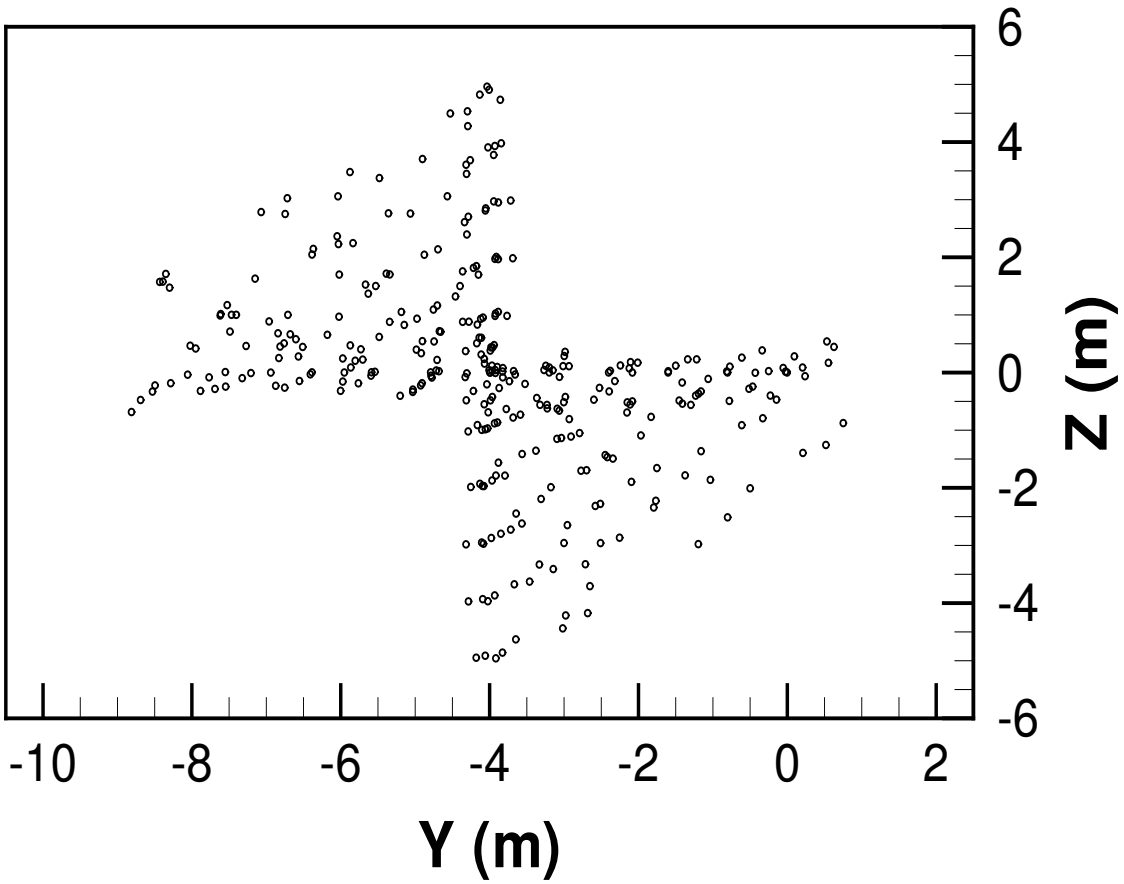


(a) Résultats obtenus avec le solveur poutre

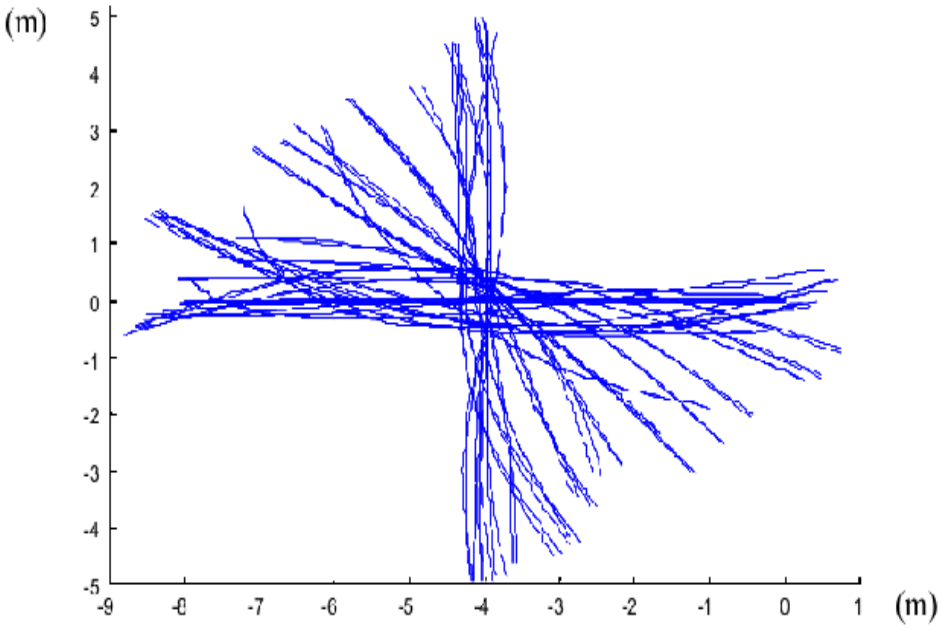


(b) Résultats de D. Primault

FIG. 3.26: Projection des nœuds de la poutre dans le plan $(\overrightarrow{x}, \overrightarrow{y})$

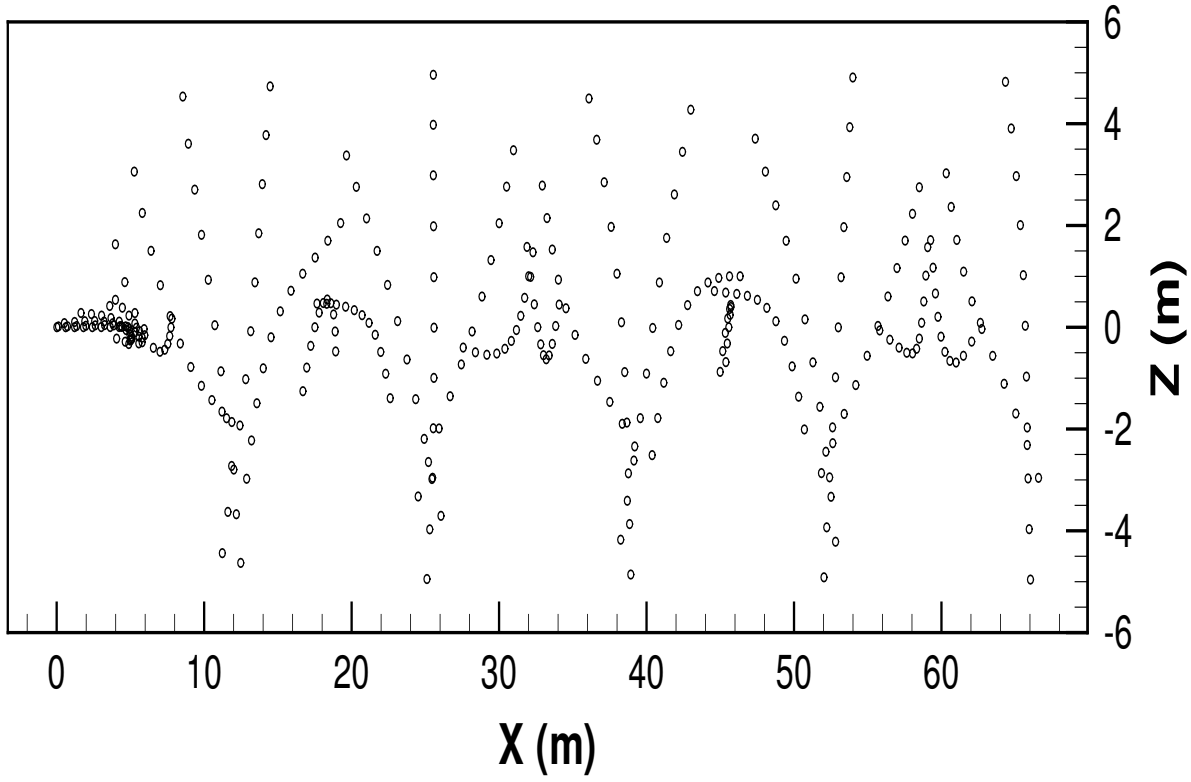


(a) Résultats obtenus avec le solveur poutre

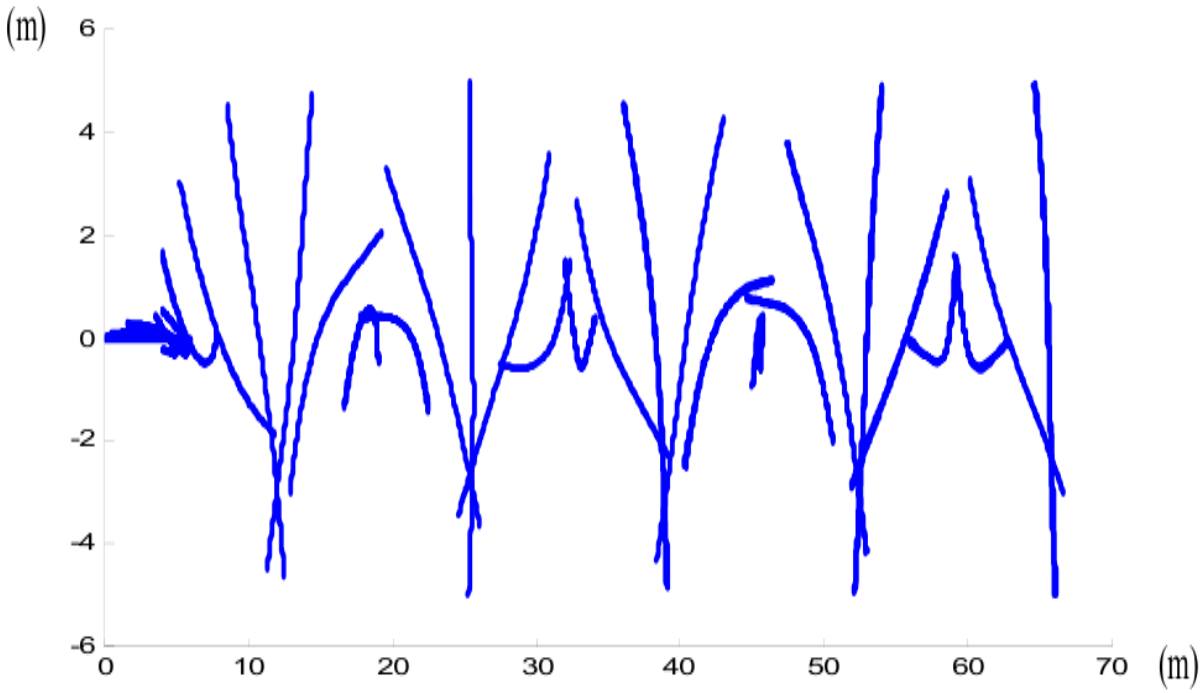


(b) Résultats de D. Primault

FIG. 3.27: Projection des nœuds de la poutre dans le plan $(\overrightarrow{y}, \overrightarrow{z})$

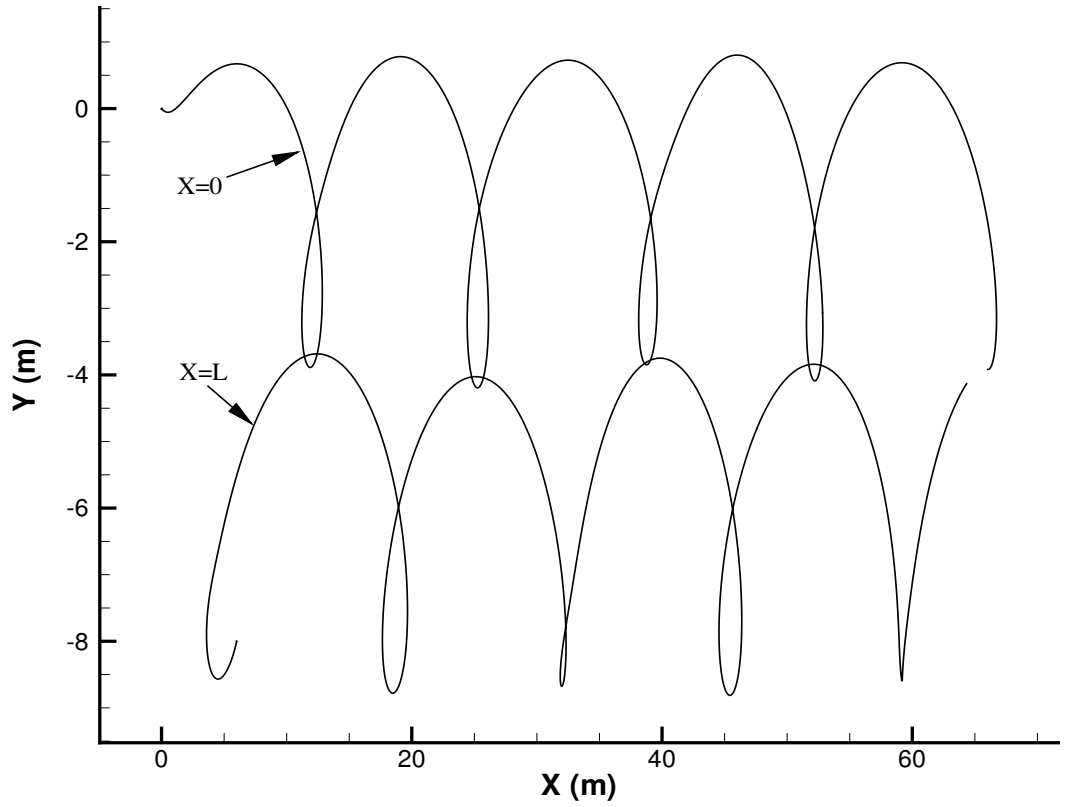


(a) Résultats obtenus avec le solveur poutre

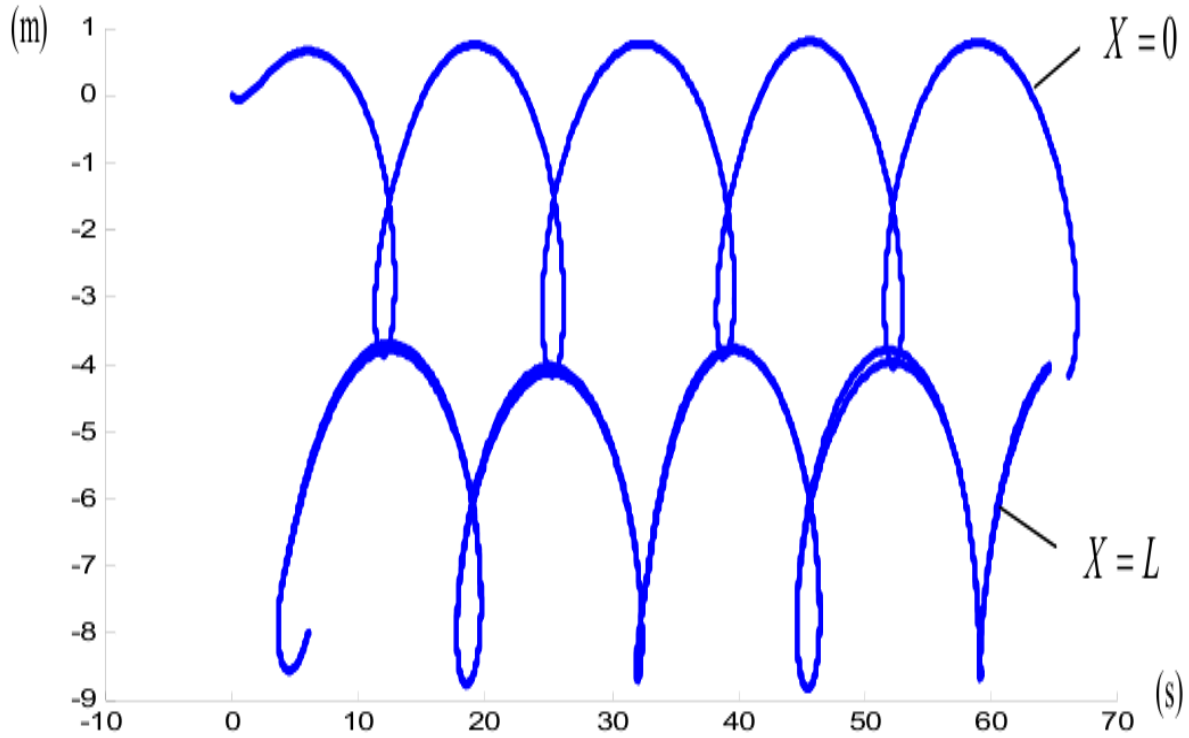


(b) Résultats de D. Primault

FIG. 3.28: Projection des nœuds de la poutre dans le plan $(\overrightarrow{x}, \overrightarrow{z})$

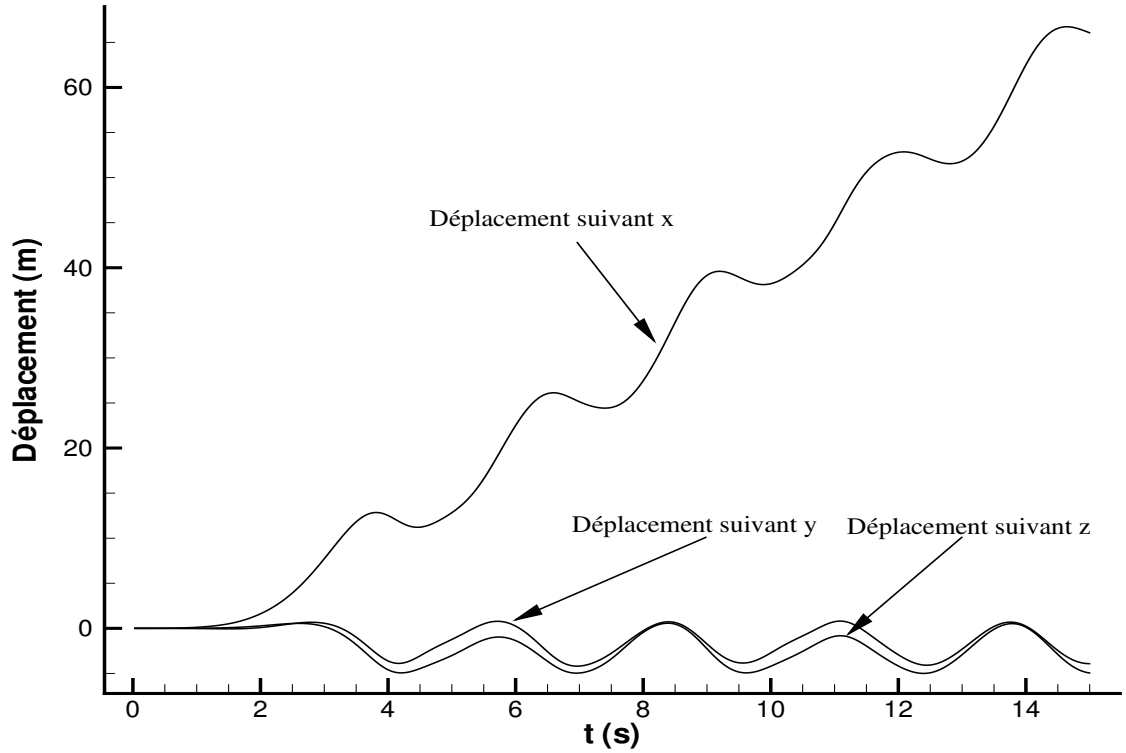


(a) Résultats obtenus avec le solveur poutre

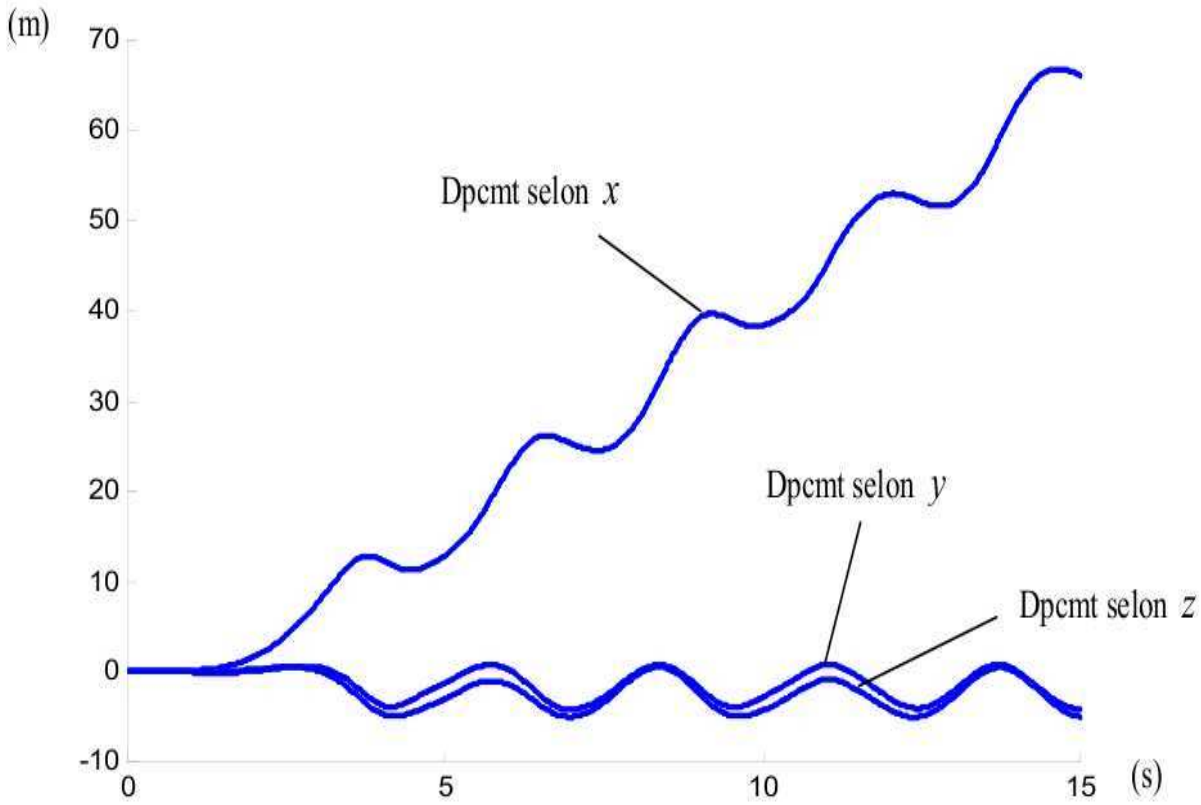


(b) Résultats de D. Primault

FIG. 3.29: Trajectoires des extrémités de la poutre

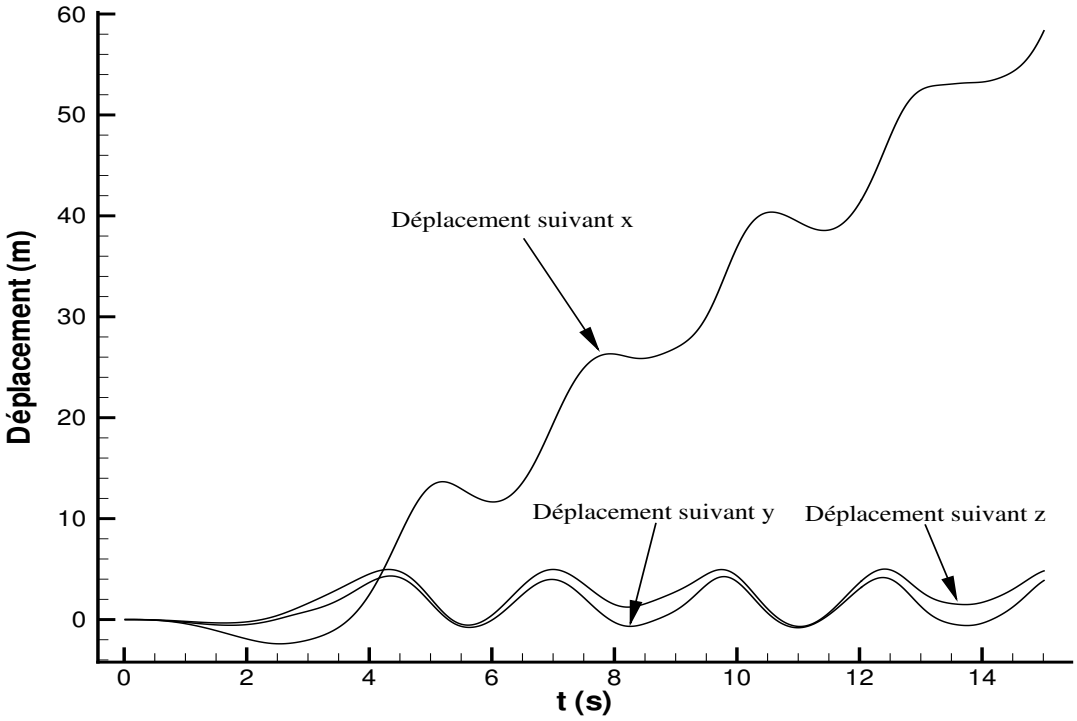


(a) Résultats obtenus avec le solveur poutre

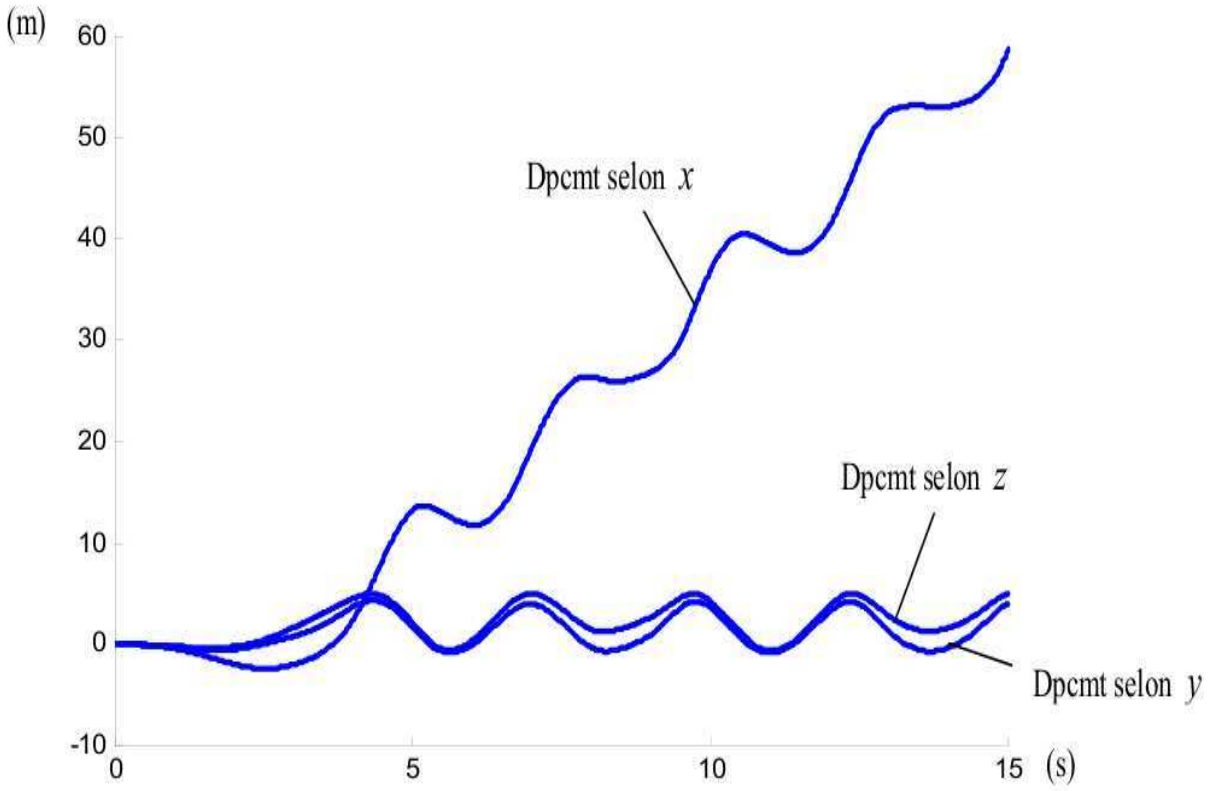


(b) Résultats de D. Primault

FIG. 3.30: Déplacement de l'extrémité $X = 0$



(a) Résultats obtenus avec le solveur poutre



(b) Résultats de D. Primault

FIG. 3.31: Déplacement de l'extrémité $X = L$

3.3 Conclusion

Le code structure, programmé durant cette thèse, se base sur les travaux de F. Boyer et D. Primault sur les poutres fines. Ce solveur poutre utilise une approche de Cosserat. Un soin tout particulier a été porté au couplage flexion-torsion avec l'utilisation d'une approche géométriquement exacte.

Ce solveur poutre a ensuite été validé en bidimensionnel et en tridimensionnel. Des tests 2D en petit déplacement, puis grand déplacement ont été réalisés. En 3D les cas de validation ont porté sur le couplage flexion-torsion et sur le modèle de Rayleigh.

En possession d'un solveur fluide, **ISIS**, présenté au chapitre 2, et du solveur poutre, **PTAH**, nous avons les deux premiers éléments essentiels d'un code IFS. Il reste maintenant à faire dialoguer ces deux programmes via un algorithme de couplage.

Chapitre 4

Couplage en espace : processus de remaillage

Les chapitres 2 et 3 ont présenté les solveurs fluide et structure nécessaires à la réalisation d'un code d'interaction fluide-structure. Mais il manque encore deux éléments essentiels : le couplage en espace et le couplage en temps. Le présent chapitre va faire l'objet d'un des aspects du couplage en espace : le remaillage fluide.

En effet, lorsqu'on simule des écoulements autour de corps en mouvement ou qui se déforment, le maillage doit être modifié à chaque pas de temps voire à chaque itération non-linéaire, pour être cohérent avec la nouvelle configuration du ou des corps en présence. Pour cela des solutions de remaillage ont déjà été implantées dans **ISIS**. Mais celles-ci présentent des limitations qui ont entraîné le développement d'une autre méthode. Sa compréhension, son élaboration et son intégration au sein d'**ISIS** représentent une partie majeure du travail effectué lors de cette thèse. C'est pourquoi un chapitre entier lui est consacré. Notons que, si on se borne dans le reste du manuscrit aux interactions fluide-structure sur des corps élancés, ce n'est pas le cas ici : le module de remaillage doit fonctionner sur des solides déformables (ou non) d'aspect divers.

Le **remailleur pseudo-structure**, objet principal de cette partie, se base sur l'approximation suivante : *le domaine fluide est considéré comme une structure solide élastique*. Les équations de structure peuvent donc être utilisées pour calculer la déformation du maillage. Cette approximation sera dénommée par la suite l'**approximation pseudo-solide**. Tout d'abord, l'approche théorique qui a permis de programmer ce remailleur sera étudiée. Puis, ces propos seront illustrés par quelques exemples de remaillage autour de corps déformables en mouvement. Enfin, ce chapitre se refermera sur les résultats fournis par le couplage de deux méthodes de remaillage.

4.1 Méthodes de remaillage déjà présentes dans ISIS

Différents modules de remaillage existaient déjà dans **ISIS**. Ces méthodes ont chacune leurs avantages et leurs inconvénients, mais elles présentent toutes la même caractéristique : leur utilisation dépend du problème traité [57].

En effet, pour un seul corps solide se déplaçant dans un milieu infini le maillage peut être déplacé en bloc avec le corps. Pour un corps déformable en milieu infini, une méthode

de remaillage est nécessaire : la méthode des ressorts ou la méthode par pondération.

4.1.1 Remaillage par mouvement en bloc

Cette technique de remaillage n'en est pas vraiment une à proprement parler. On la classe pourtant comme telle, car elle aussi fournit un maillage adapté à la nouvelle configuration d'un corps mobile. *Ce corps ne doit pas être déformable.* Il peut juste être en mouvement dans un milieu infini. Chaque point du maillage est alors considéré comme rigidement lié à la structure. Lorsque cette dernière bouge, tout le maillage se déplace avec elle en subissant la même composition de translations et de rotations. Ainsi les frontières ne peuvent pas être indépendantes.

4.1.2 Méthode des ressorts

Pour cette technique le domaine fluide est vu comme une structure mécanique quasi-statique. Le maillage discrétisant ce domaine est assimilé à un assemblage complexe de ressorts. À l'origine de cette méthode, chaque segment composant le maillage était considéré comme un ressort linéaire (cf. [9]). Puis la méthode s'est complexifiée petit à petit, et des ressorts de torsion se sont ajoutés entre les segments permettant ainsi un meilleur contrôle de la déformation du maillage (cf. [36] et [29]) (cf. fig. 4.1).

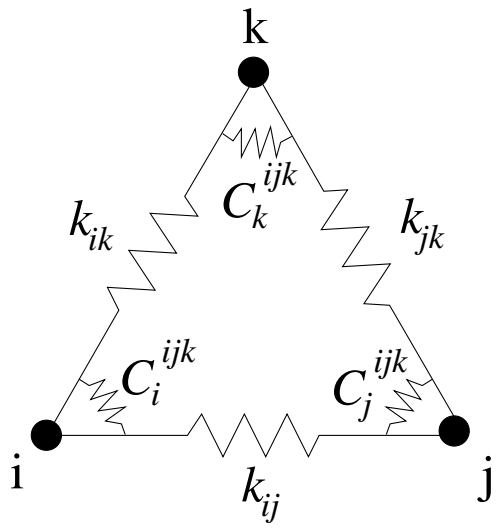


FIG. 4.1: Ressorts de compression et de torsion en 2D

Cette technique, parmi les trois présentes dans **ISIS** initialement, est la seule à pouvoir *gérer les déformations quelconques d'un corps.* Elle donne de bons résultats même en grandes déformations mais pose des *problèmes de temps de calcul en 3D.* En effet, le calcul des coefficients de torsion est très coûteux. De plus, elle demande de nouvelles connectivités (barres-faces et nœuds-nœuds), qui sont très lourdes à générer. La parallélisation en est complexifiée et difficilement réalisable.

4.1.3 Méthode de remaillage par pondération analytique

Cette méthode se base sur un coefficient calculé une fois sur le domaine à partir d'un Laplacien (cf. fig. 4.2). Ce coefficient est égal à 1 sur la frontière avec les corps en mouvement et égal à 0 sur les autres frontières. Dans le cas de corps indéformables, le nouveau maillage est obtenu en pondérant les caractéristiques du mouvement rigidifiant par ce coefficient. La procédure a été étendue aux structures de type poutre. Cependant, dans le cas de grandes déformations, des *problèmes de chevauchements de mailles* apparaissent loin du corps. Un lissage est alors obligatoire rendant cette méthode peu générale. Par contre, elle est *très rapide*, car elle ne demande aucune résolution de système.

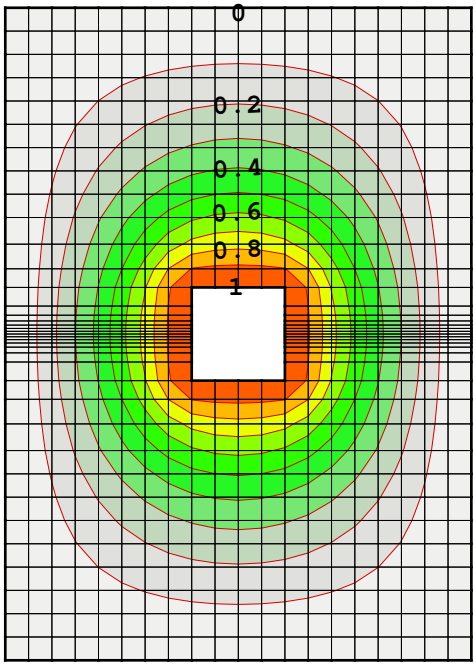


Fig. 4.2: Coefficient de pondération

4.2 Module de remaillage pseudo-structure

Pour lever les problèmes posés par la méthode des ressorts (parallélisation difficile, connectivités complexes), une nouvelle approche a été envisagée : *faire l'analogie entre le domaine fluide et un problème de structure élastique*. Ainsi, si le problème est discrétisé à l'aide de la méthode des Volumes-Finis, les procédures de résolution très efficaces d'**ISIS** peuvent être utilisées. Par conséquent la parallélisation du module de remaillage devient quasi immédiate. Et, avantage appréciable, *la déformation du maillage peut être contrôlée via des caractéristiques du matériau évolutives suivant l'espace.*

4.2.1 Arrière-plan théorique

En premier lieu, revenons sur les hypothèses et les principes de base qui vont nous permettre de travailler avec les équations que fournit l'approximation pseudo-solide.

Les objets, les matériaux peuvent être vus à différentes échelles. Les deux principales sont :

➢ l'**échelle microscopique**
➢ l'**échelle macroscopique**

Ces deux échelles s'opposent. Leurs définitions varient suivant les disciplines (Météorologie, chimie, physique moléculaire...). Dans notre cas l'échelle macroscopique correspond à l'échelle humaine (supérieure à $1\,mm$). Par opposition, l'échelle microscopique correspond à la taille de l'atome. Intuitivement on se rend compte qu'un matériau vu à l'échelle microscopique ou à l'échelle macroscopique ne peut pas être considéré de la même manière. En effet, si l'on regarde de très près un objet (échelle microscopique) la matière paraît granulaire, car constituée d'atomes. Par contre, ce même objet vu de plus loin semble être *continu*. Dans tout ce qui va suivre les corps, les matériaux seront considérés comme continus. Cette hypothèse va nous permettre de dire que ces milieux présentent des propriétés physiques et mécaniques qui peuvent être variables mais qui suivent des lois continues et à dérivées continues en fonction des coordonnées des points.

D'autres hypothèses classiques, lorsqu'on travaille avec des matériaux, considèrent ces derniers comme homogènes (le matériau a les mêmes propriétés en tout point) et/ou isotropes (le matériau a, en un point donné, les mêmes propriétés dans toutes les directions). Ces hypothèses ne nous intéressent pas dans le cadre de la déformation de maillage.

4.2.1.1 Rappels sur les contraintes et les déformations

Sur Wikipedia[1], "la base de la mécanique des milieux continus est définie comme l'étude des déformations et des phénomènes associés à une transformation d'un milieu." Nous voyons apparaître ici un des mots importants en mécanique des milieux continus : **déformation**. La déformation ϵ correspond à l'allongement relatif d'un segment au sein d'un solide, ou encore la variation d'angle entre deux directions. Sa définition sous forme locale s'écrit :

$$\epsilon = \lim_{l \to 0} \frac{\delta l}{l} \tag{4.1}$$

Dans un matériau réel, ces déformations ont lieu dans toutes les directions. On a alors besoin d'un outil plus général : le **tenseur des déformations** $\vec{\vec{\epsilon}}$. Ce tenseur dépend du mode de représentation (eulérienne ou lagrangienne) choisi pour le problème. En représentation lagrangienne, les fonctions décrivant les grandeurs dépendent de la particule considérée (ou de sa position de référence à un temps de référence) et du temps. En représentation eulérienne, les fonctions décrivant les grandeurs dépendent du point géométrique considéré et du temps. Ces deux représentations ont leurs avantages et leurs inconvénients. En lagrangien, les tenseurs de déformations se nomment *tenseur de Cauchy à droite* et *tenseur de Green-Lagrange*. En eulérien, on a le *tenseur de Cauchy à gauche* et le *tenseur d'Almansi-Euler*. Ces tenseurs de déformations ne seront pas détaillés ici. Pour plus de précision on peut se référer aux multiples ouvrages traitant du sujet comme (cf. [83]).

[1] http://fr.wikipedia.org/wiki/Accueil

La deuxième notion importante en mécanique des milieux continus est la notion de **contrainte**. Revenons plus en détail sur cette notion :

Pour un milieu continu, il est d'usage de distinguer, parmi l'ensemble des actions s'exerçant sur ce corps matériel , deux catégories :

- ➢ Les **actions mécaniques** qui se traduisent par des forces et des couples, ou par des densités de forces et de couples.
- ➢ Les **actions non mécaniques**, liées à l'existence de flux ou de sources.

Bien que certaines *actions non mécaniques* puissent engendrer des déformations au sein des milieux continus, *nous n'en tiendrons pas compte dans ce chapitre*, car elles sont inexistantes dans notre cas. Il ne reste donc plus que des actions mécaniques :

- ➢ Les **efforts extérieurs** : actions mécaniques du milieu extérieur sur notre milieu
- ➢ Les **efforts intérieurs** : actions mécaniques développées au sein de notre milieu par les interactions entre les particules qui le composent.

Dans tout ce qui suit, nous ne considérerons que des milieux matériels *non électrisés* et *non magnétisés*. Par conséquent, les actions mécaniques extérieures se réduisent à des densités volumiques ou surfaciques de forces. Les actions mécaniques intérieures nous ramènent à cette notion de contrainte et nous conduisent à définir un vecteur contrainte $\overrightarrow{\sigma}$. Soit une force $\overrightarrow{df}(\overrightarrow{n}, \overrightarrow{x}, t, dS)$ s'exerçant sur une surface matérielle d'aire dS, on appelle **vecteur contrainte** sur le plan de normale $\overrightarrow{n}$ au point P, extrémité du vecteur $\overrightarrow{x}$ et à l'instant t la limite suivante :

$$\overrightarrow{\sigma} = \lim_{dS \to 0} \frac{\overrightarrow{df}(\overrightarrow{n}, \overrightarrow{x}, t, dS)}{dS} \tag{4.2}$$

Donc, par définition une contrainte est l'effort surfacique exercé sur une partie de la pièce en un point par l'extérieur et est homogène à une pression.

Comme pour les déformations, dans un matériau réel, ces contraintes s'exercent dans toutes les directions. On a alors besoin d'un outil plus général : le **tenseur des contraintes** $\overrightarrow{\overrightarrow{\sigma}}$. Ce tenseur dépend aussi du mode de représentation (eulérien ou lagrangien) choisi pour le problème. En eulérien ce tenseur des contraintes est le *tenseur de Cauchy*. En lagrangien deux tenseurs existent : le *tenseur de Boussinesq* et le *tenseur de Piola-Kirchhoff* (cf. [83]). Dans le cadre du remailleur, la représentation eulérienne a été privilégiée pour la simplicité des formulations. Intéressons-nous donc au tenseur des contraintes de Cauchy :

Soit un milieu continu dans une configuration actuelle à l'instant t quelconque mais fixé, soit un repère orthonormal direct $R(O, \overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3})$ supposé fixe dans le référentiel lié à l'observateur et soit un vecteur position $\overrightarrow{x}$ quelconque mais fixé, on peut définir $\overrightarrow{\overrightarrow{\sigma}}$ relatif à R comme :

$$\overrightarrow{\overrightarrow{\sigma}}\,(\overrightarrow{x}, t) = \begin{bmatrix} \sigma_{11}(\overrightarrow{x}, t) & \sigma_{12}(\overrightarrow{x}, t) & \sigma_{13}(\overrightarrow{x}, t) \\ \sigma_{21}(\overrightarrow{x}, t) & \sigma_{22}(\overrightarrow{x}, t) & \sigma_{23}(\overrightarrow{x}, t) \\ \sigma_{31}(\overrightarrow{x}, t) & \sigma_{32}(\overrightarrow{x}, t) & \sigma_{33}(\overrightarrow{x}, t) \end{bmatrix} \tag{4.3}$$

Chaque colonne j de la matrice correspond aux composantes du vecteur contrainte $\overrightarrow{\sigma_j}$ s'exerçant sur le plan de normale $\overrightarrow{e_j}$ au point P extrémité du vecteur $\overrightarrow{x}$ et à l'instant t.

Une fois les deux notions de déformation et de contrainte définies, appliquons au système **le principe fondamental de la dynamique**. Soit $T^e(V_t)$ le torseur des actions mécaniques extérieures s'exerçant sur le volume matériel V à l'instant t, soit $T^d(V_t)$ le torseur dynamique de ce volume matériel au même instant, on a :

$$T^e(V_t) = T^d(V_t) \tag{4.4}$$

On peut en déduire les **équations indéfinies du mouvement en variable d'Euler** :

$$\overrightarrow{div}\ \overrightarrow{\overrightarrow{\sigma}}\ (\overrightarrow{x},t) + \rho(\overrightarrow{x},t)\,\overrightarrow{b}\,(\overrightarrow{x},t) = \rho(\overrightarrow{x},t)\,\overrightarrow{\gamma}(\overrightarrow{x},t) \tag{4.5}$$

où $\rho(\overrightarrow{x},t)$ est le champ des masses volumiques, $\overrightarrow{b}\,(\overrightarrow{x},t)$ le champ des actions mécaniques à distance et $\overrightarrow{\gamma}(\overrightarrow{x},t)$ le champ des accélérations.

Dans notre cas particulier, le *champ des actions mécaniques à distance* $\overrightarrow{\overrightarrow{b}}$ *est nul, aussi bien que le champ des accélérations* γ, puisque l'on cherche un équilibre statique de notre domaine. L'équation du mouvement se simplifie donc :

$$\boxed{\overrightarrow{div}\ \overrightarrow{\overrightarrow{\sigma}} = \overrightarrow{0}} \tag{4.6}$$

Notons tout de suite que cette formule est évidemment *commune à tous les milieux continus fluides ou solides*. On peut par exemple écrire le tenseur des contraintes de Cauchy pour un fluide visqueux incompressible :

$$\overrightarrow{\overrightarrow{\sigma}} = -p\ \overrightarrow{\overrightarrow{\delta}} + 2\mu_f\ \overrightarrow{\overrightarrow{D}} \tag{4.7}$$

où p est la pression, $\overrightarrow{\overrightarrow{\delta}}$ le symbole de Kronecker, μ_f la viscosité dynamique de cisaillement et $\overrightarrow{\overrightarrow{D}}$ le tenseur des taux de déformations. Et ainsi obtenir un système à résoudre pour un milieu continu fluide.

Dans notre cas, le milieu continu est un solide, car, rappelons-le, nous faisons l'hypothèse que *le milieu correspondant au fluide dans le système est assimilable à un pseudo-solide* sur lequel on impose les déplacements des frontières. Nous allons donc chercher un système d'équations à résoudre correspondant au mieux à notre milieu pseudo-solide.

4.2.1.2 Élasticité linéaire

Pour former le système d'équations de notre milieu pseudo-solide, nous devons trouver une formulation du tenseur des contraintes de Cauchy. Intuitivement, l'expérience le montre, les sollicitations et les déformations sont liées. Nous voulons donc une équation reliant $\overrightarrow{\overrightarrow{\sigma}}$ et $\overrightarrow{\overrightarrow{\epsilon}}$. La relation la plus simple est une relation linéaire entre les contraintes et les déformations. Cependant les solides déformables purement élastiques sont rares (le caoutchouc naturel en est un exemple), il existe néanmoins, pour la plupart d'entre eux, un domaine de sollicitations où le comportement est réversible et le plus souvent linéaire (comme pour l'acier). Ce domaine est celui des *petites déformations*.

Faisons alors l'hypothèse que notre pseudo-solide est *linéaire élastique isotrope* et que, par conséquent, nous pouvons utiliser **la loi de Hooke**.

$$\boxed{\overrightarrow{\overrightarrow{\sigma}} = \lambda tr(\epsilon)\ \overrightarrow{\overrightarrow{\delta}} + 2\mu\ \overrightarrow{\overrightarrow{\epsilon}}} \tag{4.8}$$

où μ et λ sont les coefficients de Lamé, $\overrightarrow{\overrightarrow{\delta}}$ le symbole de Kronecker et $\overrightarrow{\overrightarrow{\epsilon}}$ le tenseur des déformations.

D'autre part *en négligeant les termes du second ordre* (hypothèse des petites déformations), le tenseur des déformations $\overrightarrow{\overrightarrow{\epsilon}}$ peut s'écrire en fonction du vecteur déplacement $\overrightarrow{U}$:

$$\boxed{\overrightarrow{\overrightarrow{\epsilon}} = \frac{1}{2}(\overrightarrow{\overrightarrow{grad}}\ \overrightarrow{U} + \overrightarrow{\overrightarrow{grad}}^{\,t}\ \overrightarrow{U})} \tag{4.9}$$

Mais dans ce cas pourquoi utiliser un modèle petites déformations, alors que dans la plupart des cas de remaillage les déplacements des corps sont importants et donc les déformations de notre pseudo-solide aussi ?

La réponse à cette question est simple : **on ne cherche pas à avoir une déformation du maillage qui corresponde exactement à la déformation d'un matériau solide sous les mêmes sollicitations**. Par contre, **on désire obtenir un maillage qui soit régulier dans tout le domaine et surtout que ses mailles aux frontières avec les corps en mouvement restent orthogonales à la paroi**. On peut alors se permettre d'utiliser un modèle petites déformations qui est **léger et rapide**. Pour les mêmes raisons le système est considéré comme **quasi-statique**.

Le système d'équations à résoudre pour notre remaillage utilisant un pseudo-solide de module d'Young E et de coefficient de Poisson ν est par conséquent :

$$\begin{cases} \overrightarrow{div}\ \overrightarrow{\overrightarrow{\sigma}} = \overrightarrow{0} \\ \quad \overrightarrow{\overrightarrow{\sigma}} = \lambda tr(\epsilon)\ \overrightarrow{\overrightarrow{\delta}} + 2\mu\ \overrightarrow{\overrightarrow{\epsilon}} \\ \quad \overrightarrow{\overrightarrow{\epsilon}} = \frac{1}{2}(\overrightarrow{\overrightarrow{grad}}\ \overrightarrow{U} + \overrightarrow{\overrightarrow{grad}}^{\,t}\ \overrightarrow{U}) \end{cases} \tag{4.10}$$

avec

$$\begin{cases} \mu = \dfrac{E}{2(1+\nu)} \\ \lambda = \dfrac{\nu E}{(1+\nu)(1-2\nu)} \end{cases} \tag{4.11}$$

Nous sommes donc partis de ce système simple pour développer une première discrétisation : la *discrétisation isotrope*.

4.2.2 Discrétisations

4.2.2.1 Discrétisation isotrope

À partir de ce système une discrétisation Volumes-Finis a pu être développée. L'équation sous sa forme globale est écrite pour un volume de contrôle V. S est la surface du volume de contrôle V et $\overrightarrow{n}$ la normale à cette surface.

$$\boxed{\iiint_V \overrightarrow{div}(\overrightarrow{\overrightarrow{\sigma}})dV = 0 \Longleftrightarrow \iint_S \left[\mu\ \overrightarrow{\overrightarrow{grad}}\overrightarrow{U} + (\mu+\lambda)div(\overrightarrow{U})\right] \cdot \overrightarrow{n}\,dS = 0} \tag{4.12}$$

Cette discrétisation est la première à avoir été testée. Elle donne des résultats *satisfaisants pour des mouvements de translation* (cf. fig. 4.24). Cependant, dès que des *mouvements de rotation* apparaissent les mailles proches de la paroi se *déforment trop*, même en utilisant une distribution variable du module d'Young E (cf. fig. 4.4(a)). Le maillage généré n'est alors plus acceptable pour un calcul fluide.

Continuant dans l'analogie du fluide avec un pseudo-solide, nous nous sommes dits qu'un cisaillement transverse plus fort pourrait permettre un meilleur contrôle de la déformation du maillage. Une **discrétisation utilisant un coefficient de cisaillement variable** a donc été développée.

4.2.2.2 Discrétisation avec coefficient de cisaillement variable

Dorénavant, la notation matricielle est préférée pour une meilleure visualisation des différents termes composant la discrétisation.

Nous allons utiliser cette approche pseudo-solide pour un matériau **orthotrope**, c'est-à-dire un *"matériau élastique homogène présentant en tout point 2 symétries du comportement mécanique chacune par rapport à 1 plan, les 2 plans étant orthogonaux"*. Il combine donc les propriétés d'un matériau orthogonal et celles d'un matériau anisotrope.

En tridimensionnel, un tel matériau a la loi de comportement suivante dans la base cartésienne en adoptant la notation "chapeau" :

$$\widehat{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = K \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{G_{xy}}{K} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{yz}}{K} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{xz}}{K} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix} \tag{4.13}$$

avec

$$K = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \qquad \text{et} \qquad \epsilon_{xy} = \frac{1}{2}\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right)$$

où G_{ij} est le coefficient de cisaillement relatif aux directions $\overrightarrow{i}$ et $\overrightarrow{j}$. U et V sont les composantes du vecteur déplacement $\overrightarrow{U}$.

Cette matrice symétrique correspond à une discrétisation anisotrope ayant des **directions principales**. En effet, cette matrice est diagonalisable et son repère est alors **le repère des contraintes principales**. Les vecteurs de base de ce repère forment les directions principales de contraintes, et les valeurs propres associées sont les contraintes principales. Notons que l'on peut définir de la même manière des directions principales de déformations et les déformations principales.

Nous cherchons à obtenir une *déformation du maillage la plus isotrope possible*. Si l'on applique tel quel la discrétisation à l'ensemble du domaine, les directions principales de déformations et de contraintes apparaissent. Nous définissons donc cette matrice dans une base locale $(\overrightarrow{n}, \overrightarrow{t_2}, \overrightarrow{t_3})$, ainsi les directions principales seront locales elles aussi. Cette base

n'est pas choisie au hasard dans notre module de remaillage. Elle est liée à chaque maille composant le maillage : $\overrightarrow{n}$ est dirigé par le vecteur gradient du coefficient de pondération du problème, variable pour chaque maille ; $\overrightarrow{t_2}$ et $\overrightarrow{t_3}$ forment un plan orthogonal à $\overrightarrow{n}$. Les lignes isométriques de ce coefficient de pondération ont la propriété de suivre la forme du corps (cf. fig. 4.3). Ainsi, le vecteur $\overrightarrow{n}$ de la base locale sera normal par rapport au corps. Cela permet alors de limiter la déformation due au cisaillement suivant $(\overrightarrow{t_1}, \overrightarrow{t_2})$ et ainsi de conserver une meilleure orthogonalité à la paroi. *Pour chaque maille nous avons donc une base locale associée, dans laquelle la discrétisation du matériau orthotrope est exprimée.*

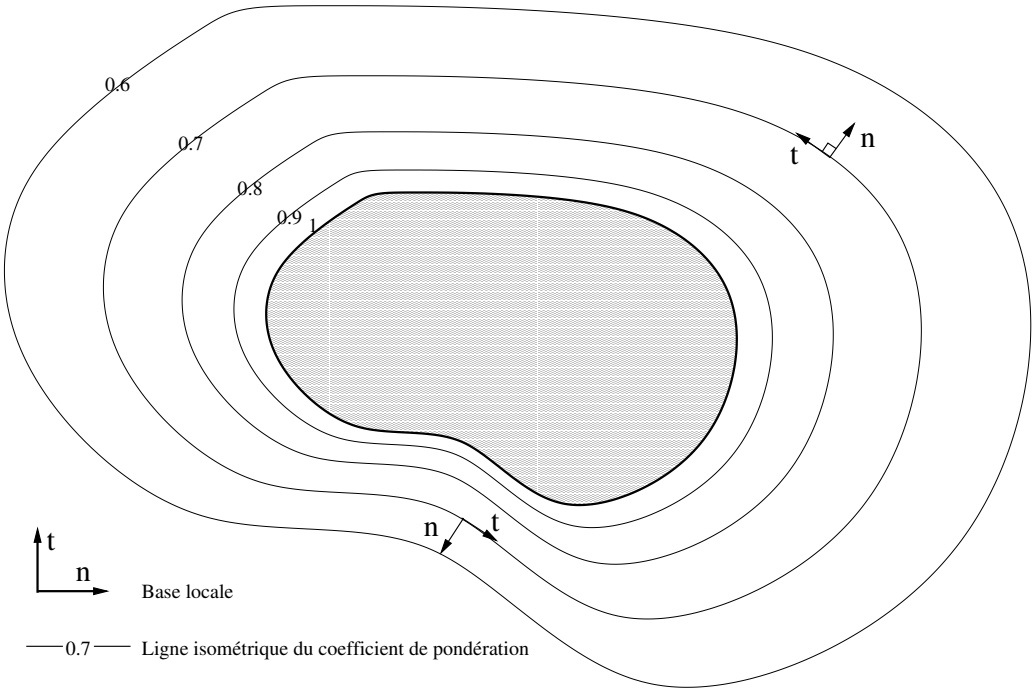


FIG. 4.3: Bases locales créées sur les lignes isométriques du coefficient de pondération

Revenons à notre problème global de remaillage. Il est défini dans la base cartésienne, il faut donc transformer la matrice de la loi de comportement du matériau orthotrope exprimée dans la base locale et l'écrire dans la base cartésienne de référence.

Soit P, la matrice de passage de la base $(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z})$ à $(\overrightarrow{n}, \overrightarrow{t_2}, \overrightarrow{t_3})$ on a :

$$\sigma_{(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z})} = P\, \sigma_{(\overrightarrow{n}, \overrightarrow{t_2}, \overrightarrow{t_3})}\, P^{-1} \tag{4.14}$$

avec

$$\overrightarrow{n} = n_x \overrightarrow{x} + n_y \overrightarrow{y} + n_z \overrightarrow{z} \qquad \text{et} \qquad P = \begin{bmatrix} n_x & P_{12} & P_{13} \\ n_y & P_{22} & P_{23} \\ n_z & P_{32} & P_{33} \end{bmatrix}$$

On obtient ainsi une relation de la forme suivante :

$$\widehat{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = K \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix} \tag{4.15}$$

Les développements étant longs et fastidieux, ils ont été réalisés sous MAPLE™ [2] (cf. annexe A). Les coefficients y sont donnés. Attention toutefois, les calculs sous MAPLE™ ont été réalisés sans la notation "chapeau", par conséquent avec ϵ_{ij} et non γ_{ij}.

On note que les G_{ij} sont indépendants les uns des autres. *On fixe G_{nt_2} et G_{nt_3} égaux et non isotropes, tandis que $G_{t_2t_3}$ est égal à $G_{iso} = \frac{E}{(1+\nu)}$ ce qui entraîne une isotropie suivant cette direction.* On obtient ainsi un comportement semblable à un matériau isotrope transverse.

Au niveau de la discrétisation numérique, il est plus facile de séparer les formes matricielles de l'équation isotrope et celle de la partie anisotrope :

$$\widehat{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = K \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{G_{iso}}{K} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{iso}}{K} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{iso}}{K} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix} +$$

$$K \begin{bmatrix} C_{11}-1 & C_{12}-\frac{\nu}{1-\nu} & C_{13}-\frac{\nu}{1-\nu} & C_{14} & C_{15} & C_{16} \\ C_{12}-\frac{\nu}{1-\nu} & C_{22}-1 & C_{23}-\frac{\nu}{1-\nu} & C_{24} & C_{25} & C_{26} \\ C_{13}-\frac{\nu}{1-\nu} & C_{23}-\frac{\nu}{1-\nu} & C_{33}-1 & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44}-\frac{G_{iso}}{K} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55}-\frac{G_{iso}}{K} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66}-\frac{G_{iso}}{K} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix} \tag{4.16}$$

La première matrice est calculée de la même manière que dans le cas isotrope sans cisaillement, alors que la seconde est uniquement traitée explicitement.

En 2D, les lois de comportement sont différentes dans le cas des déformations planes et des contraintes planes. La loi de comportement 2D isotrope en déformations planes est la suivante :

[2] http://www.maplesoft.com

$$\widehat{\sigma} = \begin{bmatrix} \sigma_n \\ \sigma_t \\ \sigma_{nt} \end{bmatrix} = K \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{1-\nu} \end{bmatrix} \begin{bmatrix} \epsilon_n \\ \epsilon_t \\ \epsilon_{nt} \end{bmatrix} \tag{4.17}$$

C'est la même loi qu'en 3D avec les termes relatifs à la $3^{ième}$ composante annulés. Les développements précédents s'appliquent en annulant les termes relatifs à $\overrightarrow{z}$. Par contre dans le cas des contraintes planes, la loi et donc les développements qui suivent, sont différents.

Écrivons en 2D la loi de comportement en contraintes planes et introduisons G dans l'écriture matricielle réduite "chapeau" :

$$\widehat{\sigma} = \begin{bmatrix} \sigma_n \\ \sigma_t \\ \sigma_{nt} \end{bmatrix} = A \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{G}{A} \end{bmatrix} \begin{bmatrix} \epsilon_n \\ \epsilon_t \\ \epsilon_{nt} \end{bmatrix} \tag{4.18}$$

avec

$$A = \frac{E}{1-\nu^2}$$

De la même manière que dans les cas 3D, on se place dans une base locale puis par changement de base on se ramène à la base cartésienne.
Soit P, la matrice de passage de la base $(\overrightarrow{x}, \overrightarrow{y})$ à $(\overrightarrow{n}, \overrightarrow{t})$ on a :

$$\sigma_{(\overrightarrow{x}, \overrightarrow{y})} = P\, \sigma_{(\overrightarrow{n}, \overrightarrow{t})}\, P^{-1} \tag{4.19}$$

avec

$$\overrightarrow{n} = n_x \overrightarrow{x} + n_y \overrightarrow{y} \qquad \text{et} \qquad P = \begin{bmatrix} n_x & -n_y \\ n_y & n_x \end{bmatrix}$$

On obtient ainsi la relation suivante :

$$\widehat{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} = A \begin{bmatrix} C_{11} & C_{12} & C_{14} \\ C_{12} & C_{22} & C_{24} \\ C_{14} & C_{24} & C_{44} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{bmatrix} \tag{4.20}$$

avec

$$\begin{aligned} C_{11} = C_{22} =& n_x^4 + n_y^4 + 2n_x^2 n_y^2 (\nu + \frac{G}{A}) \\ C_{12} =& \nu(n_x^4 + n_y^4) + 2n_x^2 n_y^2 (1 - \frac{G}{A}) \\ C_{14} = -C_{24} =& 2n_x n_y (n_x^2 - n_y^2)(1 - \nu - \frac{G}{A}) \\ C_{44} =& 4n_x^2 n_y^2 (1-\nu) + (n_x^2 - n_y^2)^2 \frac{G}{A} \end{aligned} \tag{4.21}$$

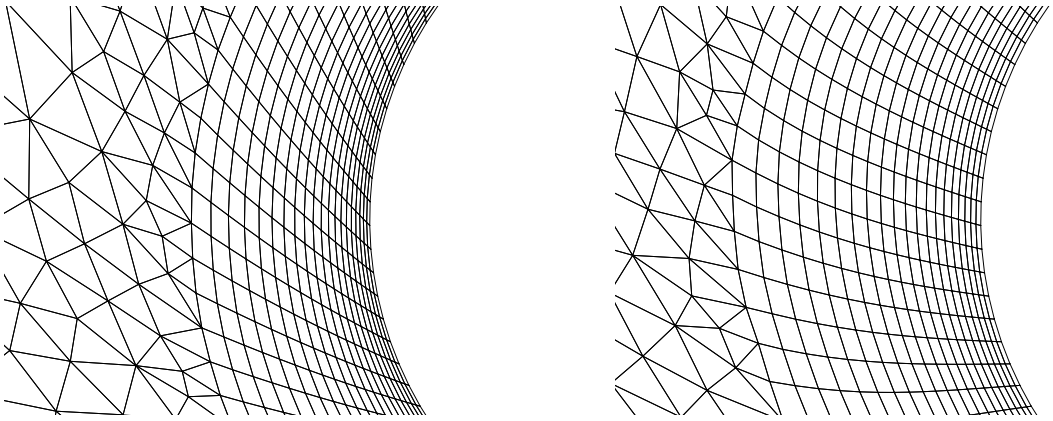
En faisant apparaître la forme matricielle de l'équation isotrope :

$$\widehat{\sigma} = A \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{bmatrix} + A \begin{bmatrix} C_{11}-1 & C_{12}-\nu & C_{14} \\ C_{12}-\nu & C_{22}-1 & C_{24} \\ C_{14} & C_{24} & C_{44}-(1-\nu) \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{bmatrix} \tag{4.22}$$

Comme précédemment dans le cas 3D, la première matrice est calculée de la même manière que dans le cas isotrope sans cisaillement, alors que la seconde est uniquement résolue explicitement.

On note que pour $G = \frac{E}{1+\nu}$, on retrouve bien l'équation isotrope.

Prenons l'exemple d'un corps sphérique en rotation pure. En comparant les résultats obtenus sur la figure 4.4, on remarque que la discrétisation à coefficient de cisaillement variable permet de mieux contrôler la forme des mailles proches de la paroi. Plus on augmente G, plus les cellules restent orthogonales au corps.



(a) Discrétisation isotrope (b) Discrétisation à coefficient de cisaillement variable

FIG. 4.4: Comparaison entre le maillage généré avec les différentes discrétisations sur un mouvement de rotation pure

Mais, dans le cas 2D comme dans le cas 3D, *plus G est grand, plus le poids des termes explicites est important et donc plus la convergence est difficile voire impossible* suivant le problème et les déformations rencontrées. On a donc modifié la formulation discrète pour favoriser la convergence.

4.2.2.3 Discrétisation avec forte implicitation

Le problème 4.23 est *couplé sur les déplacements* :

$$\iint_S \mu \overrightarrow{grad}\,(\overrightarrow{U}) \cdot \overrightarrow{n}\,dS + \iint_S \left[\mu \overrightarrow{grad}^{\,t}\,(\overrightarrow{U}) + \lambda tr(\overrightarrow{\epsilon})\,\overrightarrow{I}\right] \cdot \overrightarrow{n}\,dS = 0 \tag{4.23}$$

En effet, il n'est pas possible de déterminer une des composantes du déplacement $\overrightarrow{U}$ sans connaître les deux autres. Par exemple, si on projette l'équation 4.23 sur $\overrightarrow{x}$, on obtient avec $\overrightarrow{U} = \begin{bmatrix} U \\ V \\ W \end{bmatrix}$ une équation faisant intervenir U, V et W :

$$\begin{aligned} &\iint_S \mu \left(\frac{\partial U}{\partial x} n_x + \frac{\partial U}{\partial y} n_y + \frac{\partial U}{\partial z} n_z \right) dS \quad + \\ &\iint_S \left[\mu \left(\frac{\partial U}{\partial x} n_x + \frac{\partial V}{\partial x} n_y + \frac{\partial W}{\partial x} n_z \right) + \lambda \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} \right) n_x \right] dS = 0 \end{aligned} \tag{4.24}$$

Ce type de système peut être formé de manière **couplée** ou **découplée**. On parle de système couplé, lorsque l'on traite l'ensemble du problème en une fois. Les trois composantes U, V et W du déplacement sont alors les inconnues et sont obtenues directement. Dans la configuration découplée, trois sous-problèmes sont considérés : un par composante du vecteur déplacement. On détermine U en figeant V et W. Puis, V en fixant U et W. Il en va de même pour le calcul de W. Puis, on itère en mettant à jour les différentes composantes. Après quelques passages, on décide que le gain est suffisant et que la convergence est atteinte.

Utiliser un solveur couplé nécessite des connectivités supplémentaires et plus de mémoire. L'objectif étant dans un premier temps de valider la méthode pseudo-solide, il a par conséquent été écarté. Intéressons-nous au problème 4.23 de façon découplée. Dans l'équation 4.23, le second terme est donc traité explicitement et inclus dans le terme source. La première intégrale demande l'évaluation du gradient du déplacement. Ceci est traité grâce aux méthodes présentes dans **ISIS**. Elles sont expliquées à la section §2.2.4 du chapitre sur le solveur fluide. On emploie la méthode de Gauss avec corrections. La partie discrétisée du gradient est alors implicitée pour la résolution, tandis que les corrections non-orthogonales sont stockées dans le terme source.

$$\underbrace{\iint_S \mu \overrightarrow{grad}\,(\overrightarrow{U}) \cdot \overrightarrow{n} dS}_{\substack{\text{traité avec la méthode de} \\ \text{Gauss avec corrections}}} + \underbrace{\iint_S \left[\mu \overrightarrow{grad}^{\,t}\,(\overrightarrow{U}) + \lambda tr(\overrightarrow{\epsilon})\, \overrightarrow{I} \right] \cdot \overrightarrow{n} dS}_{\substack{\text{traité explicitement} \\ \text{(envoyé dans le terme source)}}} = 0 \tag{4.25}$$

Lors de la résolution du système, le poids des termes explicites joue un rôle important sur la convergence. Une modification astucieuse de la discrétisation permet théoriquement une meilleure convergence. Dans le cas isotrope simple au lieu d'employer 4.23, on utilise :

$$\boxed{\begin{aligned} &\iint_S (2\mu + \lambda) \overrightarrow{grad}\,(\overrightarrow{U}) \cdot \overrightarrow{n} dS \quad + \\ &\iint_S \left[\mu \overrightarrow{grad}^{\,t}\,(\overrightarrow{U}) + \lambda tr(\overrightarrow{\epsilon})\, \overrightarrow{I} - (\mu + \lambda) \overrightarrow{grad}\,(\overrightarrow{U}) \right] \cdot \overrightarrow{n} dS = 0 \end{aligned}} \tag{4.26}$$

Le poids du terme implicite est donc augmenté et celui du terme explicite a été réduit. Cela permet une meilleure convergence de la résolution numérique.

Cette méthode est nécessaire dans la discrétisation avec cisaillement variable. En effet, le poids des termes explicites augmente avec la valeur de G. Et, un coefficient de cisaillement G fort donne évidemment de meilleurs résultats. La technique de forte implicitation a donc été modifiée pour la discrétisation avec module de cisaillement variable. Elle prend en compte le poids de G : $2\mu + \lambda$ est remplacé par $\mu + \alpha\mu + \lambda$, où α est défini par $G = \alpha\mu$.

4.2.3 Intégration dans ISIS

Une fois la théorie et les discrétisations développées, nous pouvons aborder l'intégration du module de remaillage au sein du solveur **ISIS**. L'algorithme du module tel qu'il apparaît dans **ISIS** est schématisé sur la figure 4.5.

Le module a besoin de coefficients relatifs à la résolution du système (comme les coefficients de Lamé ou encore les coefficients relatifs aux bases locales). Ceux-ci sont invariables et donc *ne sont calculés qu'une seule fois au début de la simulation, à partir de la configuration de référence*. Il faut ensuite initialiser le vecteur déplacement $\overrightarrow{U}$ avec le vecteur nul, sauf aux frontières où l'on trouve les déplacements imposés par la structure. Cette étape fait intervenir une première interpolation : en effet, *les déplacements imposés par la structure sont donnés aux nœuds, tandis que le vecteur* $\overrightarrow{U}$ *contient les déplacements des centres des cellules*. On initialise ensuite les gradients de déplacements.

Une fois ce travail de préparation terminé, on entre dans la boucle de résolution du système. *C'est une boucle "tant que" qui teste le gain obtenu après résolution*. Elle débute par un nouveau calcul plus précis des gradients de déplacements, car ils sont nécessaires pour obtenir une bonne interpolation. Vient ensuite la discrétisation du problème : dans notre programme, elle est de la forme $AU = Src$ et *toutes les variables sont situées au centre des cellules*, car nous sommes en volumes finis. On calcule le résidu, ce qui va nous permettre de former le système en entier. On résout ce dernier à l'aide des modules d'**ISIS** prévus à cet effet. On termine cette boucle par la réimposition des conditions aux limites, une sous-relaxation éventuelle et un stockage de la solution. Quand un gain suffisant est obtenu, on sort de la boucle "tant que".

Une fois la solution finale de $AU = Src$ obtenue, il nous reste à interpoler la solution du centre des cellules aux nœuds du maillage. Par sécurité, les déplacements exacts des nœuds frontières sont réimposés. Les nouvelles positions des nœuds du maillage déformé sont déduites des déplacements. Le travail du module de remaillage est terminé.

Cet algorithme va servir de plan pour la suite de cette section. Intéressons-nous alors au calcul des coefficients relatifs à la résolution du système.

4.2.3.1 Calcul des coefficients nécessaires à la résolution du système

Les coefficients nécessaires à la résolution du système sont multiples : coefficients des matrices de passage pour la discrétisation avec coefficient de cisaillement, ou encore caractéristiques locales du pseudo-solide.

Le calcul des coefficients des matrices de passage se base sur le calcul du coefficient de pondération utilisé dans le remaillage par pondération. En effet, celui-ci fournit la direction normale voulue pour la base locale des déformations principales.

Une des principales caractéristiques intéressantes de ce module de remaillage pseudo-structure est le contrôle de la déformation de maillage via des paramètres "physiques" : les

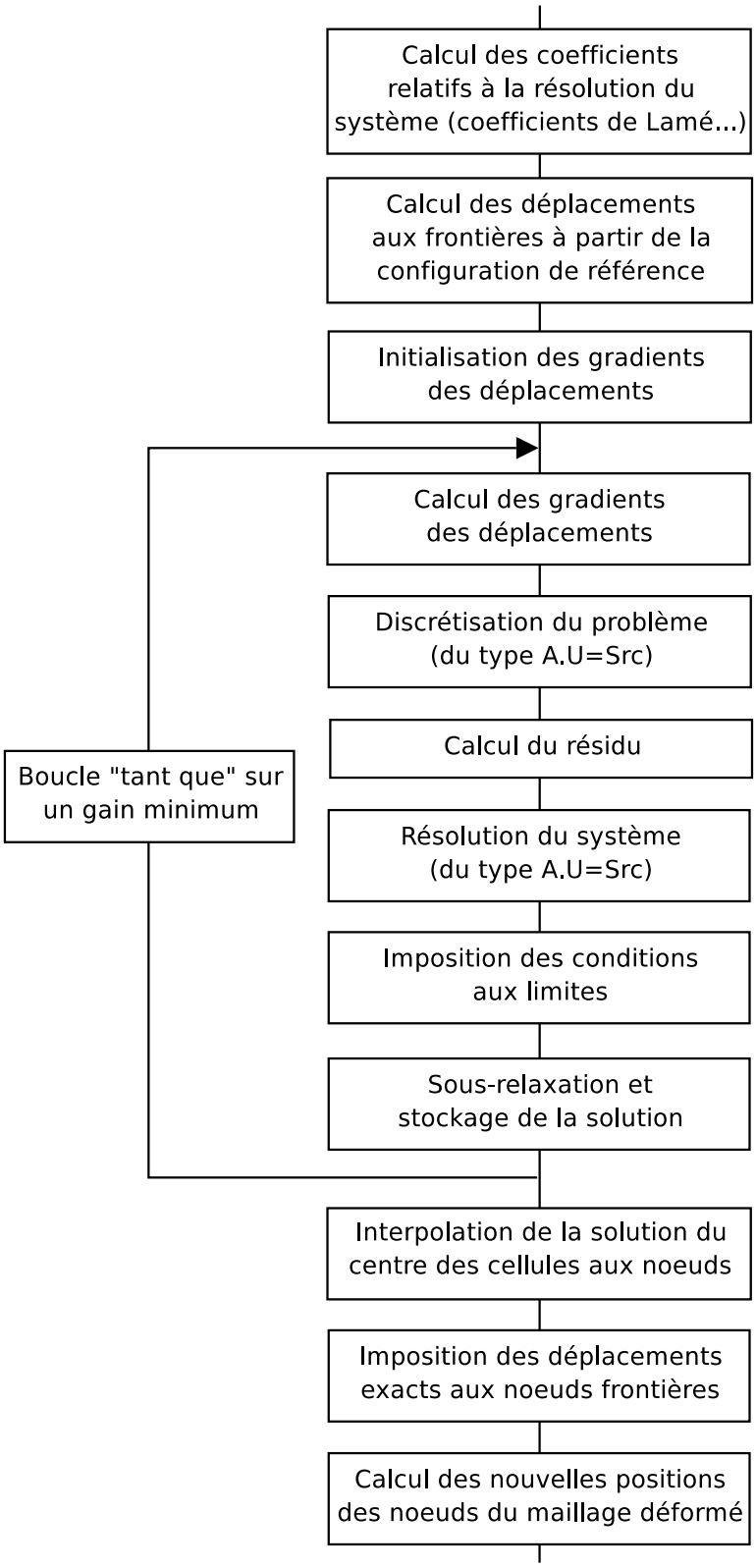


FIG. 4.5: Schéma de la procédure de remaillage

caractéristiques du pseudo-matériau. Ainsi, par exemple en jouant sur le module d'Young E, on fait varier la rigidité du matériau et donc la déformation locale du maillage correspondant.

Diverses fonctions ont été codées au sein du module pour contrôler les variations de ces paramètres "physiques".

Le cisaillement G donne de bons résultats, lorsqu'il est lié au coefficient de Lamé μ (et par conséquent à E). Par défaut dans le code, on emploie : $G = 25\mu$.

Le coefficient de Poisson ν est, quant à lui, fixé à 0.20 sur tout le domaine.

4.2.3.2 Calcul des déplacements aux frontières

Pour le bon fonctionnement du module de remaillage, trois maillages différents doivent être considérés :

- ➢ X_{ref} est le **maillage initial**.
- ➢ X_p est le **maillage au pas de temps précédent**.
- ➢ X_{ref_def} est un **maillage intermédiaire** qui ne sert qu'au calcul des déplacements initiaux. Celui-ci n'est pas un maillage correct : il se compose en effet des nœuds du maillage non déformé (X_{ref} ou X_p suivant le cas) partout, sauf à la frontière des corps, où les nœuds sont ceux du maillage déformé (cf. fig. 4.6).

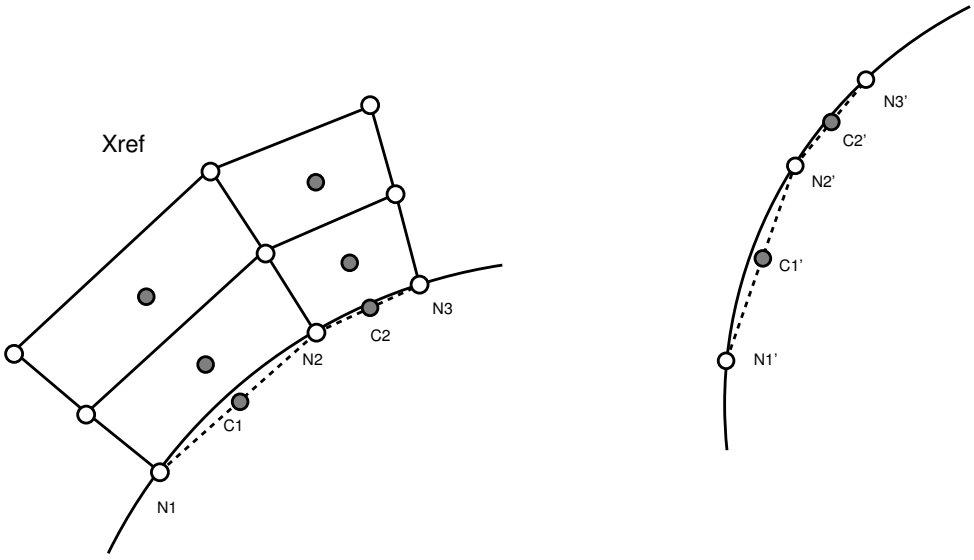


FIG. 4.6: X_{ref} = maillage intérieur + nœuds frontières du maillage initial N_1, N_2 et N_3
$X_{ref_{def}}$ = maillage intérieur + nœuds frontières N_1', N_2' et N_3', déplacés suivant la nouvelle position du corps

Le module de remaillage peut se baser soit sur X_{ref}, soit sur X_p. Dans les deux cas, le calcul des déplacements aux frontières est le même (cf. fig. 4.7) :

① Calcul des centres des "cellules frontières" du maillage déformé X_{ref_def} à partir des nœuds (C_1' et C_2' sur la figure 4.7)

② Calcul du déplacement aux frontières en soustrayant ces nouveaux centres aux anciens ($\overrightarrow{C_1C_1'}$ et $\overrightarrow{C_2C_2'}$ sur la figure 4.7)

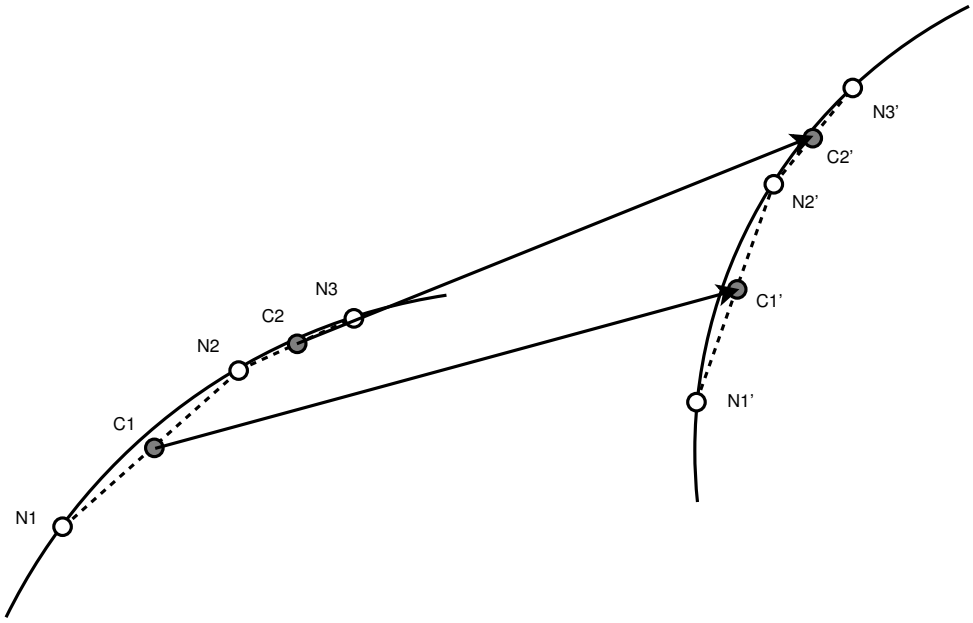


FIG. 4.7: Déplacements aux frontières

Une fois le vecteur déplacement initialisé, le système peut être formé en tenant compte de la discrétisation présentée précédemment.

4.2.3.3 Résolution du problème

Le système est sous la forme $AU = Src$ où U est le vecteur déplacement. On utilise le module de résolution utilisé dans **ISIS** pour la pression. Cela permet d'avoir accès aux mêmes techniques de résolution comme PCGSTAB ou PGMRES.

4.2.3.4 Imposition des conditions aux limites dans la résolution

À la fin de la résolution, les conditions aux limites en déplacement sont réimposées pour être cohérent. On traite les cellules frontières suivant leur marqueur :

- ➢ *Pour les symétries* : on impose à la cellule frontière un déplacement opposé à celui de la cellule adjacente à la paroi.
- ➢ *Pour les corps* : on impose le déplacement exact stocké.

4.2.3.5 Interpolation de la solution du centre des cellules aux nœuds

La résolution du système a été effectuée aux centres des cellules. Les déplacements obtenus doivent par conséquent être interpolés, afin de les avoir aux nœuds du maillage. Diverses interpolations ont été codées pour que l'approximation de la nouvelle position des nœuds du maillage soit la plus exacte possible. Seule la meilleure, celle utilisée dans le programme, est présentée.

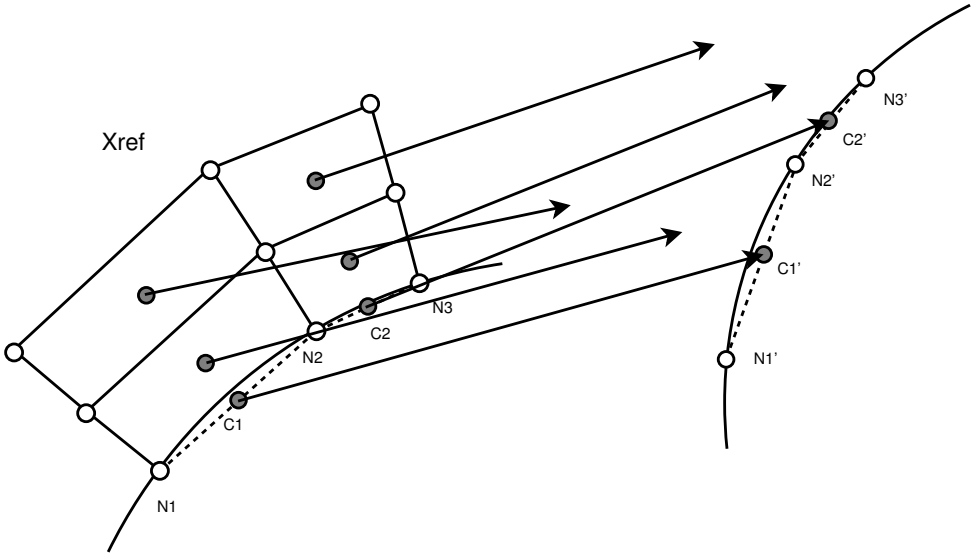


FIG. 4.8: Données avant interpolation : maillages X_{ref}, $X_{ref_{def}}$ et vecteurs déplacements aux centres des cellules $\overrightarrow{U}_{cell}$

L'interpolation 4.27 est une *moyenne pondérée par les distances entre le centre des cellules et les nœuds de la cellule*. Elle est d'ordre 1 et porte sur les coordonnées cartésiennes des vecteurs déplacements aux centres des cellules.

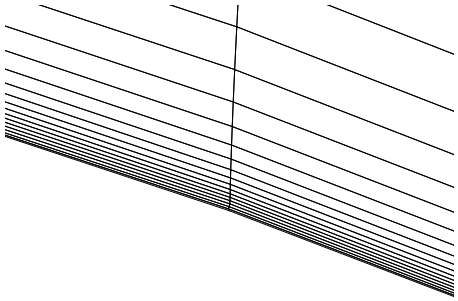
$$\overrightarrow{U}_{node} = \frac{\sum_{\text{Cellules voisines}} \left(\frac{\overrightarrow{U}_{cell} + \vec{\nabla}(\overrightarrow{U_{cell}}) \cdot \overrightarrow{CN}}{\left\| \overrightarrow{CN} \right\|} \right)}{\sum_{\text{Cellules voisines}} \left(\frac{1}{\left\| \overrightarrow{CN} \right\|} \right)} \tag{4.27}$$

Cette méthode est *exacte pour les mouvements de translation*. Elle donne de *très bons résultats pour les mouvements de rotation pure*.

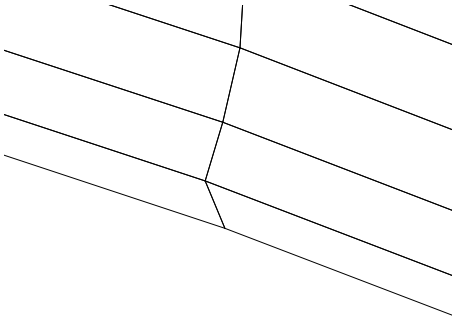
4.2.3.6 Méthode de correction des erreurs d'interpolation

Dans certaines configurations géométriques accompagnées de grandes déformations, des problèmes d'erreur d'interpolation apparaissent aux frontières. Lorsque le remailleur a affaire à des *cellules très étirées* (cf. fig. 4.9(a)), à des *cellules voisines formant un angle important entre elles* (cf. fig. 4.9(c)) ou alors aux deux, d'importantes erreurs d'interpolation s'accumulent. Celles-ci sont aussi présentes près de la paroi du corps. La donnée de départ du processus de remaillage est la nouvelle position du corps, i.e. l'emplacement des nœuds du maillage à la frontière. Pour être cohérent avec cette donnée initiale, on réimpose la position exacte des nœuds aux frontières. À cause des erreurs d'interpolation, un décalage apparaît alors au niveau de la première couche de cellules (cf. fig. 4.9(b)), allant jusqu'à provoquer l'apparition de *volumes négatifs*, et par voie de conséquence l'arrêt du solveur **ISIS**.

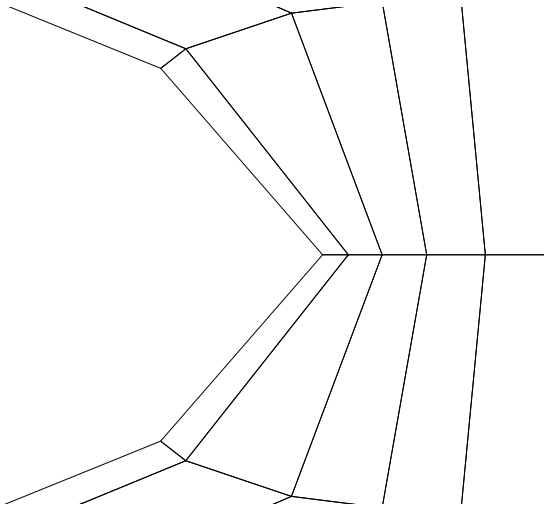
Le passage à des interpolations d'ordre plus élevé a été envisagé, mais très vite écarté.



(a) Cellules très étirées (vue d'ensemble)



(b) Cellules très étirées (zoom sur le problème d'interpolation aux nœuds)



(c) Cellules formant un angle important

FIG. 4.9: Cas géométriques posant problème pour l'interpolation aux nœuds

En effet, leurs mises en place sont longues et fastidieuses pour un résultat incertain. Une autre approche a été décidée : une **réinterpolation aux nœuds de l'erreur faite aux centres des cellules** (cf. fig. 4.10).

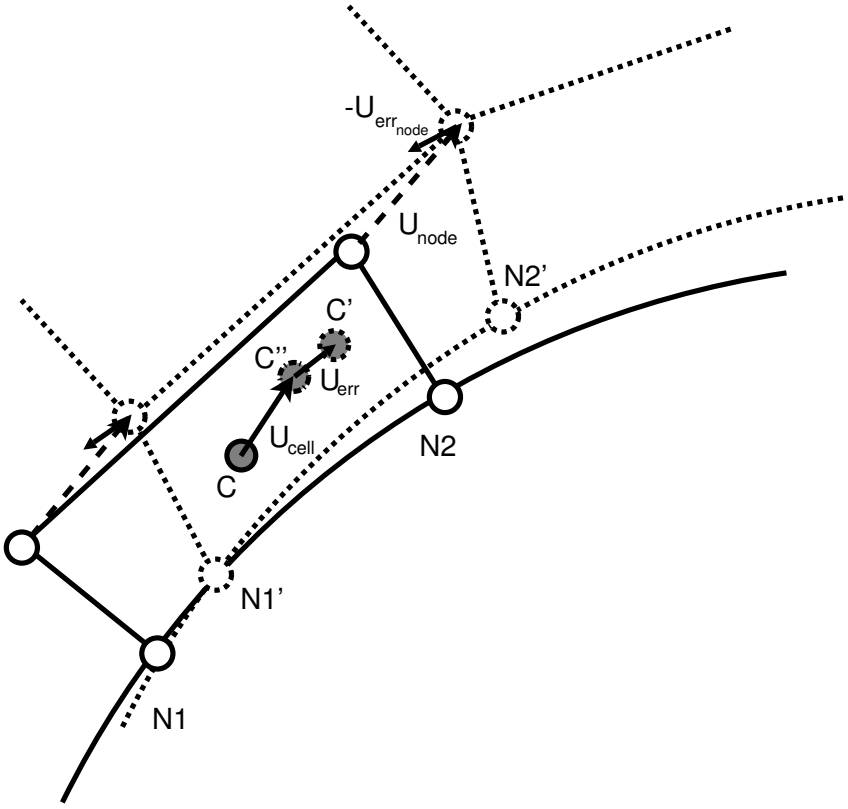


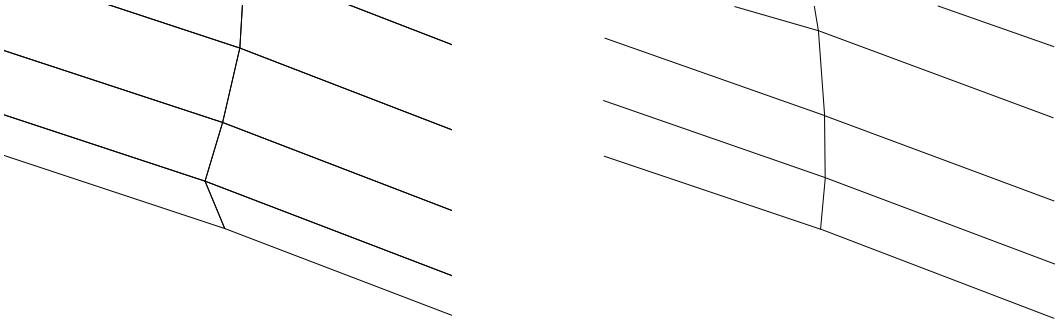
FIG. 4.10: Réinterpolation aux nœuds de l'erreur faite aux centres des cellules

La résolution du problème pseudo-structure donne le déplacement des centres du maillage de manière "exacte" U_{cell}. Une fois interpolé au nœud, ce déplacement U_{node} fournit la nouvelle position des points du maillage. Les outils présents dans **ISIS** permettent de calculer le centre C' des cellules ainsi créées. En comparant ce dernier avec le centre "exact" C'' obtenu par la résolution pseudo-structure, on a accès à l'erreur d'interpolation U_{err} aux centres des cellules. Cette erreur peut être interpolée aux nœuds, donnant $U_{err_{node}}$, permettant ainsi de converger vers la position idéale des nœuds du maillage.
Voici l'algorithme de cette méthode :

1. Initialisation
2. Calcul sans correction des centres des cellules à partir des centres des cellules du maillage de base et du déplacement calculé par le remailleur
3. Boucle sur la correction d'erreur aux centres des cellules
   ① Calcul sans correction des centres des cellules à partir du maillage déformé
   ② Calcul de l'erreur entre ces deux positions du centre des cellules
   ③ Sous-relaxation avec l'erreur de l'itération précédente
   ④ Stockage de l'erreur
   ⑤ Si l'erreur augmente localement on impose une erreur nulle, i.e la position du centre de cellules calculée à partir du maillage de base

⑥ Interpolation de l'erreur aux nœuds
⑦ Calcul du nouveau maillage déformé

Reprenons notre problème de cellules étirées, et appliquons la méthode de correction d'erreur d'interpolation. En observant les résultats présentés sur les figures 4.11(a) et 4.11(b), on note que le "décalage", initialement présent proche de la paroi, disparaît avec l'emploi de la méthode de correction d'erreur d'interpolation. On améliore ainsi la qualité générale du maillage et les cellules ayant des volumes négatifs sont évitées.



(a) Sans la correction d'erreur d'interpolation (b) Avec la correction d'erreur d'interpolation

FIG. 4.11: Influence de la méthode de correction d'erreur d'interpolation dans le cas des cellules très étirées

4.2.3.7 Imposition des conditions initiales aux nœuds

Un traitement spécial de marquage des nœuds frontières est nécessaire. Ceci est dû à certaines configurations multiblocs, apportées par le calcul parallèle. Prenons l'exemple suivant (cf. fig. 4.12) :

Le problème vient du nœud commun aux 3 blocs. En effet vu par le bloc 1 et 3, ce nœud est sur une face frontière ayant le marqueur symZ, il sera donc bien traité avec une approche par les faces frontières. Par contre le bloc 2 ne "voit" pas ce nœud comme un nœud frontière, car pour lui ce nœud appartient à 2 faces INTERFACE qui ne sont pas considérées comme faces frontières. Une approche par les nœuds résout le problème ; le bloc 2 verra ce nœud comme un nœud frontière ayant la condition symZ délivrée par les blocs voisins.

Le marquage des nœuds est réalisé à l'aide d'un marqueur spécial total et de base 2 : un marqueur de type *symétrie suivant X* ajoute 2 au marqueur total ; un marqueur de type *symétrie suivant Y* ajoute 4 ; un marqueur de type *symétrie suivant Z* ajoute 8 ; un marqueur de type *mouvement imposé* ajoute 16. Ainsi, un nœud peut avoir plusieurs marqueurs de types différents venant de leurs faces voisines. Pour l'exemple 4.12, le nœud problématique aura le marqueur spécial 8. Les nœuds ayant leurs bons marqueurs dans le cas multiblocs, on peut leur imposer les différentes conditions initiales.

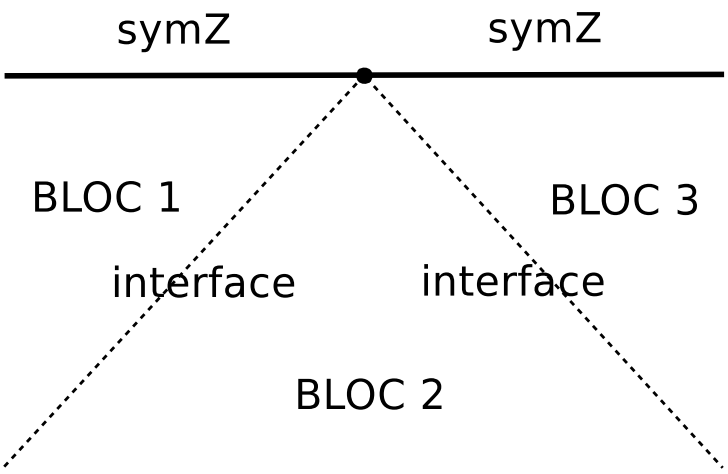


FIG. 4.12: Configuration problématique pour le calcul parallèle

Avec l'imposition des conditions initiales aux nœuds, l'algorithme de remaillage est complet. Mais lors des tests, un problème apparaît rapidement : la consommation en temps CPU est importante. Un algorithme de résolution rapide a donc été développé.

4.2.3.8 Algorithme de résolution rapide

Exposé du problème

Comme expliqué à la section §4.2.2.3, le problème est *couplé sur les déplacements*. Les gradients des déplacements sont en partie traités explicitement lors de la résolution du système. Pour obtenir une solution consistante, on doit itérer. Après plusieurs itérations linéaires, le gain sur le résidu (que l'on a auparavant fixé) est atteint. À ce moment, le système est considéré comme résolu. La solution ainsi obtenue est correcte. Mais ce type de résolution consomme beaucoup de temps CPU (résolution de trois systèmes à chaque itération linéaire). Une autre approche a par conséquent été envisagée et codée pour le cas de l'utilisation du module de remaillage dans un calcul fluide.

Solution mise en place

Dans un calcul fluide le pas de temps est petit. Les variations des déplacements et déformations des corps peuvent donc être considérés comme faibles entre deux pas de temps. On peut donc utiliser les gradients des déplacements de l'itération précédente lors de la résolution du système du remaillage. Cette hypothèse nous permet de traiter notre problème de manière *totalement découplé* et donc d'obtenir, en une seule itération linéaire, un maillage déformé convenable. Notons que cette méthode ne fournit qu'une solution approchée du problème physique couplé, car ce dernier est quasi-statique et les gradients employés sont ceux du pas de temps précédent. Cependant, nous sommes dans le cadre d'une procédure de remaillage, la physique de la déformation importe peu.

Le gain en temps CPU est *très important* et cette méthode est utilisée par défaut

dans ISIS pour le remaillage. La première méthode est tout de même gardée, car elle est consistante. Elle permet par exemple d'utiliser le module de remaillage comme un *outil de modification de maillages*. Prenons un cas simple d'utilisation : un maillage 3D est généré autour d'un corps. Plusieurs calculs doivent être effectués autour de ce corps, mais ce dernier ayant différentes positions. Au lieu de faire un maillage pour chaque configuration, le module de remaillage peut être utilisé sur le maillage initial en déplaçant le corps d'une configuration à l'autre. La méthode de résolution rapide n'est pas applicable dans ce cas. Seule la première méthode est valable.

4.2.4 Validation du module de remaillage

Pour corriger et valider le code du remailleur, de nombreux tests ont été effectués sur diverses formes de corps en mouvement. Des essais 2D ont tout d'abord été réalisés, puis l'extension en 3D a été immédiate. Les corps simples en mouvement sont des carrés, des octaèdres, des triangles et bien sûr des cercles.

Pour valider le remailleur, la comparaison avec une solution exacte est la meilleure solution. Dans le cadre de la validation l'algorithme de résolution rapide est écarté, car il est non-consistant. Un problème faisant intervenir une sphère a été retenu du fait de sa simplicité : Une sphère de rayon R_1 et de centre O est comprise dans une sphère de rayon R_2 ($R_2 > R_1$) et de centre O. La sphère de rayon R_1 subit une expansion de uR_1 suivant $\overrightarrow{e_r}$. La sphère de rayon R_2 est figée. Les coefficients E et ν sont invariants.
On doit donc résoudre :

$$\iiint_V div(\overrightarrow{\overrightarrow{\sigma}})dV = 0$$

avec

$$\overrightarrow{\overrightarrow{\sigma}} = 2\mu \overrightarrow{\overrightarrow{\epsilon}} + \lambda tr(\overrightarrow{\overrightarrow{\epsilon}}) \overrightarrow{\overrightarrow{I}}$$

Vu les symétries du problème on peut écrire $\overrightarrow{U} = U_r(r)\overrightarrow{e_r}$. La conséquence immédiate est une simplification de $\overrightarrow{\overrightarrow{\epsilon}}$:

$$\overrightarrow{\overrightarrow{\epsilon}} = \frac{1}{2}(\overrightarrow{\overrightarrow{grad}}(\overrightarrow{U}) + \overrightarrow{\overrightarrow{grad^t}}(\overrightarrow{U}))$$

$$\overrightarrow{\overrightarrow{grad}}(\overrightarrow{U}) = \begin{bmatrix} \frac{\partial U_r}{\partial r} & 0 & 0 \\ 0 & \frac{U_r}{r} & 0 \\ 0 & 0 & \frac{U_r}{r} \end{bmatrix} = \overrightarrow{\overrightarrow{grad^t}}(\overrightarrow{U}) = \overrightarrow{\overrightarrow{\epsilon}}$$

L'équation générale se réduit alors à :

$$\frac{\partial^2 U_r}{\partial r^2} + 2\frac{1}{r}\frac{\partial U_r}{\partial r} - 2\frac{1}{r^2}U_r = 0$$

L'équation est une équation différentielle du second ordre qui ne dépend pas des coefficients de Lamé. On vérifie ce fait à l'aide du remailleur : les coefficients de Lamé n'ont aucune influence sur la solution.
La solution exacte se calcule simplement :

$$U_r = uR_1 \frac{R_1^2}{(R_1^3 - R_2^3)}(r - \frac{R_2^3}{r^2})$$

La géométrie retenue pour notre validation est $R_1 = 1$, $R_2 = 2$, $uR_1 = 0.1$. Quatre maillages non-structurés tétraédriques ont été générés à l'aide de GRIDGEN™. On a ensuite calculé l'erreur maximale sur les différents maillages. La courbe (cf. fig. 4.13) ainsi tracée montre que l'erreur sur le déplacement diminue d'un facteur 1.2 avec la distance caractéristique du maillage.

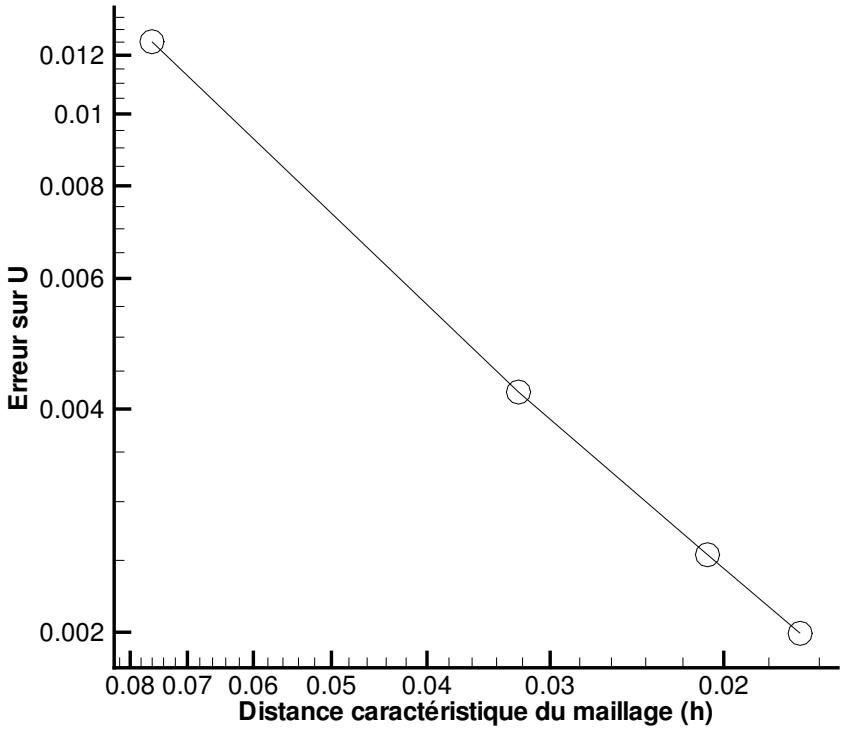


FIG. 4.13: Ordre de l'erreur sur un maillage non structuré

4.2.5 Exemples de remaillage

Le module remaillage étant validé, quelques exemples de maillages obtenus avec celui-ci vont être présentés. Nous verrons tout d'abord des maillages GRIDGEN™ déformés *autour de poutres en grand déplacement*, car l'IFS sur des corps allongés est l'objet de cette thèse. Nous nous intéresserons ensuite au *remaillage autour d'une anguille qui nage* (Projet anciennement ROBEA, et nouvellement RAAMO), à un exemple sur *un corps indéformable, une caisse, en mouvement libre sur l'eau*. Enfin, une étude sur différents corps en mouvement (comme un requin) a été réalisée avec des maillages HEXPRESS™.

4.2.5.1 Déformations de poutres (maillage GRIDGEN™)

Les maillages présentés dans cette section viennent des cas-tests d'application IFS qui seront étudiés au chapitre 6.

Le module de remaillage s'est tout d'abord attelé au problème d'une plaque 2D verticale dans une conduite horizontale. Cette poutre est fixée à la conduite par une de ses extrémités. L'écoulement est lui aussi horizontal, la poutre va donc se plier. Pour que la déformation soit plus importante, une charge continue et réglable est appliquée à la poutre. La modification du maillage est ainsi plus difficile (cf. fig. 4.14). Bien que la déformation soit importante et le maillage fin, le module de remaillage réussit parfaitement sa tâche.

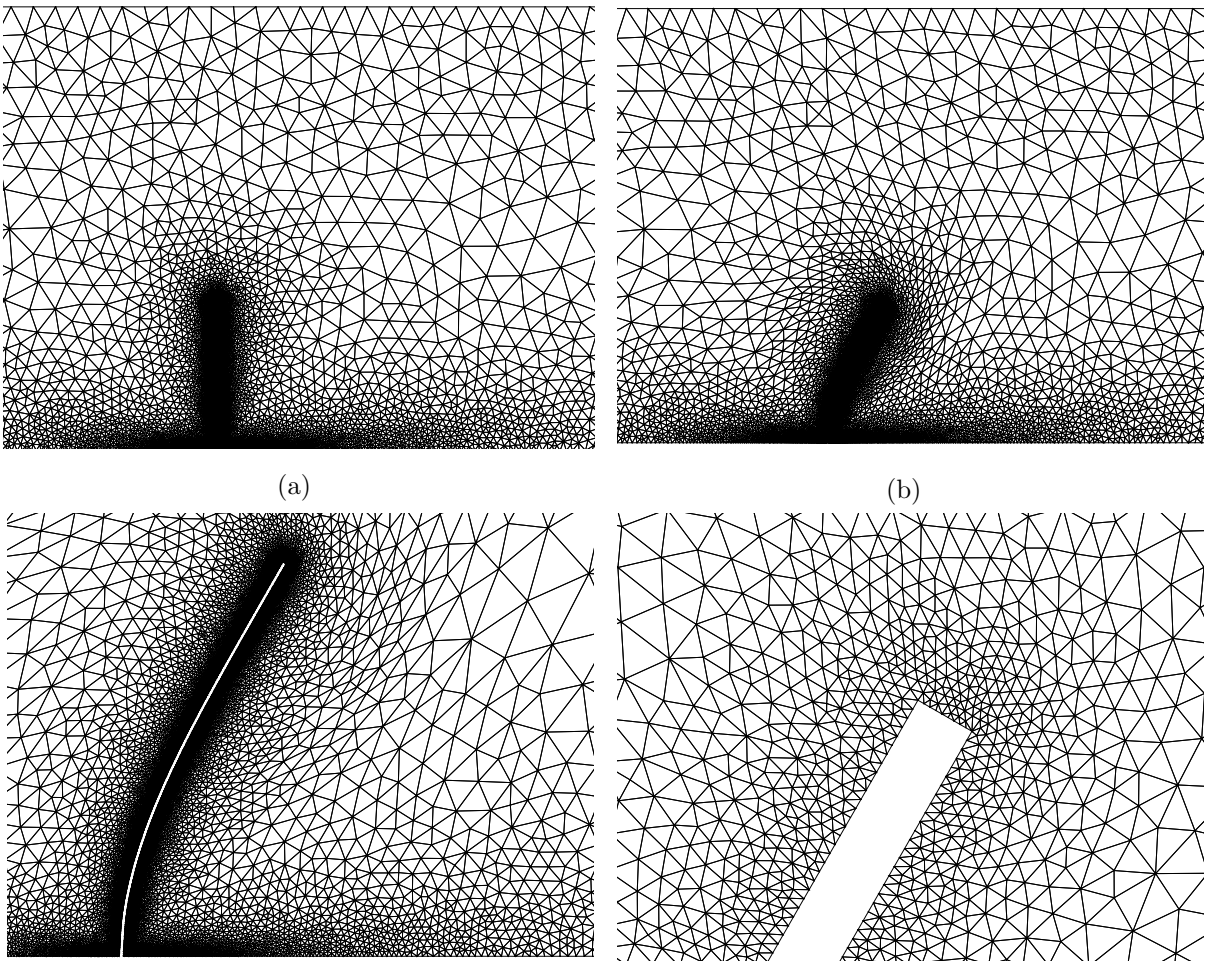
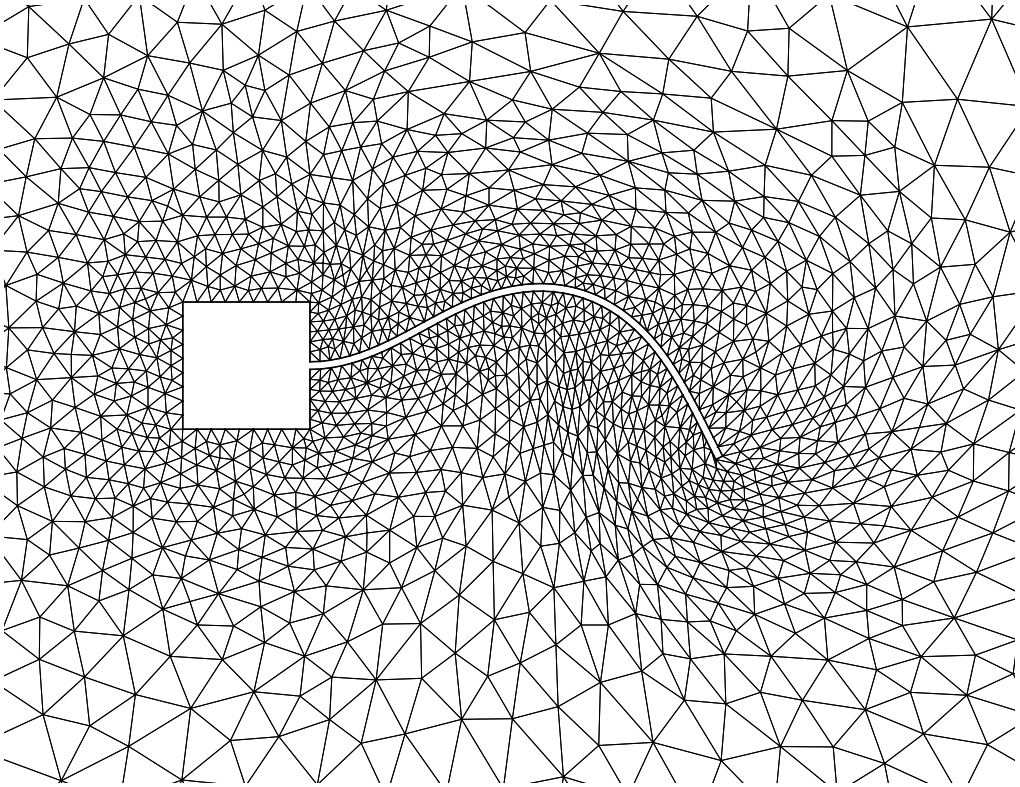


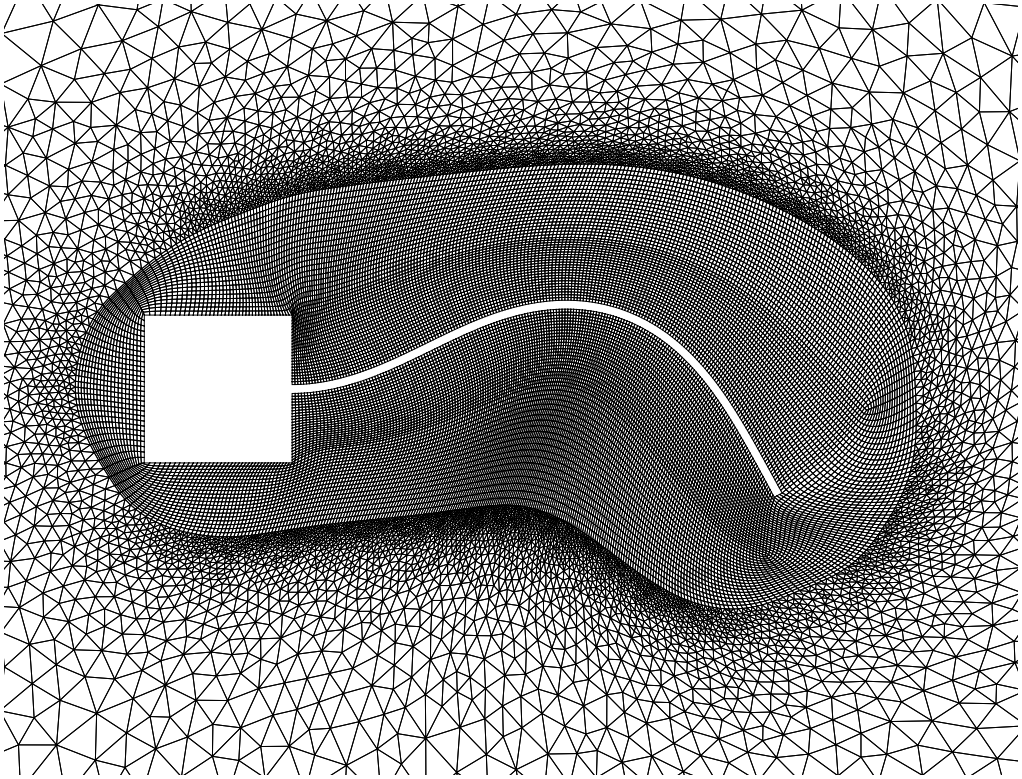
FIG. 4.14: Déformation d'une poutre (cf. cas-test IFS à la section 6.1.1)

Une deuxième configuration a été testée et est présentée : un filament 2D déformable est fixé derrière un corps carré fixe et indéformable. Ce cas-test est présenté en détail au chapitre 6, au paragraphe § 6.1.2. Lors de cet essai, la poutre se déforme de manière importante sous l'action du fluide. Le processus de remaillage est un succès aussi bien sur un maillage GRIDGEN ™ non-structuré grossier (cf. fig. 4.15(a)), que sur un maillage plus fin (cf. fig. 4.15(b)).

Sur le maillage fin, une zone régulière proche du corps a été utilisée pour mieux prendre en compte la déformation. Si l'on applique à la poutre une charge fixe de valeur raisonnable, l'espace régulier proche de la paroi ne perd pas sa forme (cf. fig. 4.16). Par contre



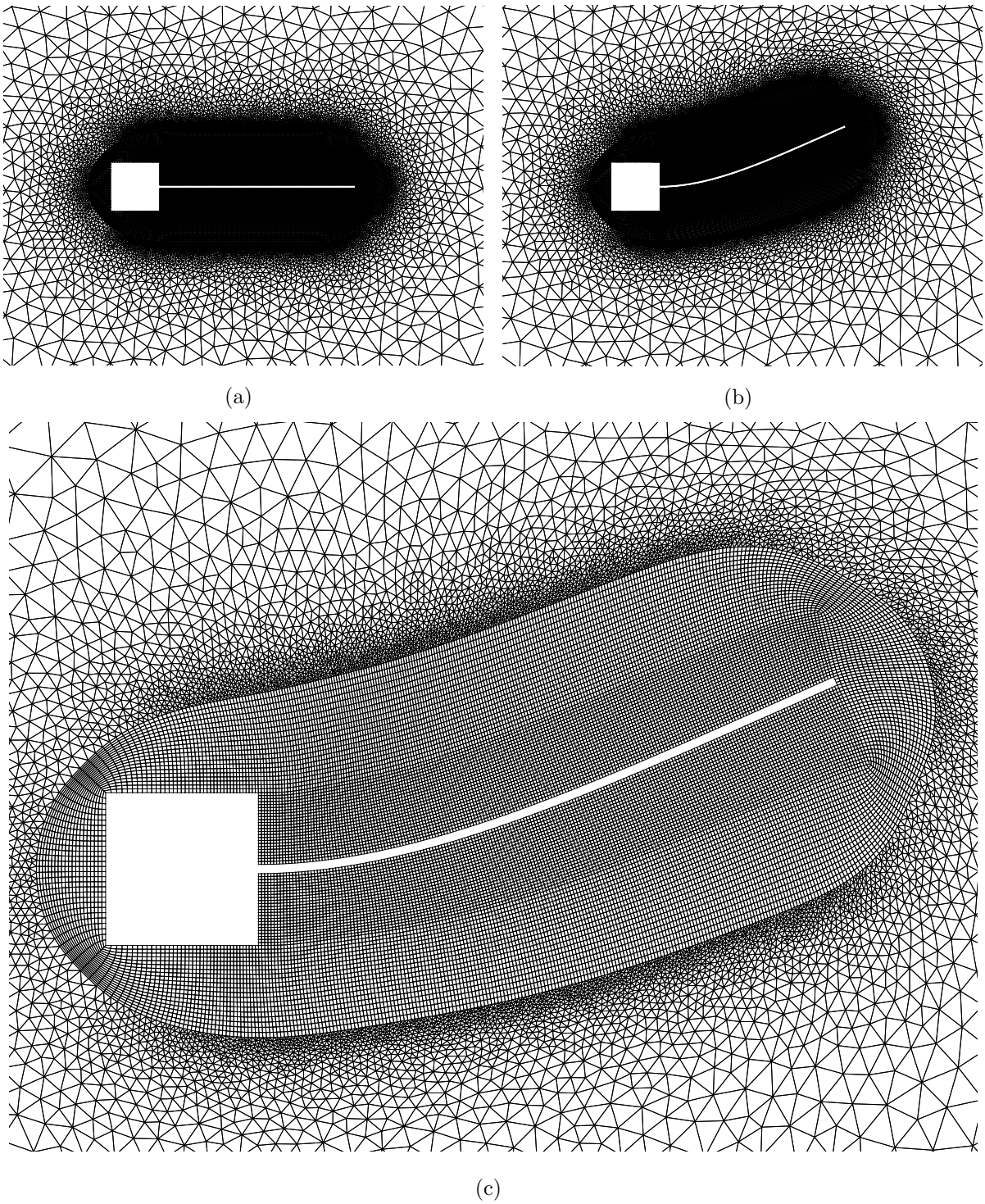
(a) Maillage grossier



(b) Maillage fin

FIG. 4.15: Remaillage autour d'une poutre (cf. cas-test IFS à la section § 6.1.2)

pour des charges plus importantes (2 à 3 fois plus grandes), le processus de remaillage fonctionne toujours, mais la nature régulière du maillage près du corps est perdue peu à peu.



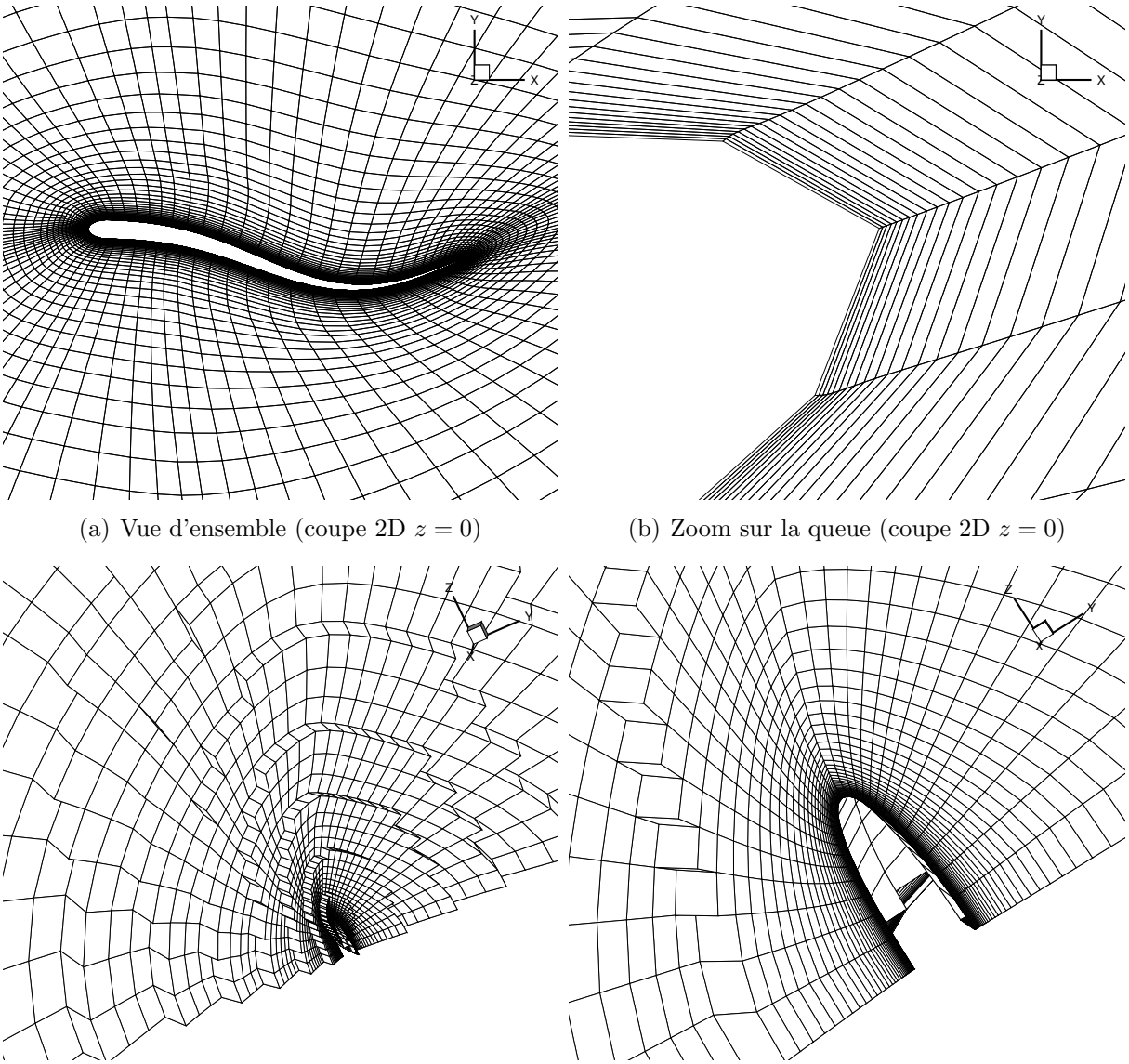
(a) (b)

(c)

FIG. 4.16: Remaillage autour d'une poutre subissant une charge fixe imposée (maillage fin)

4.2.5.2 Le robot anguille : *ROBEA/RAAMO* (maillage GRIDGEN ™)

Notre équipe (EMN) du Laboratoire de Mécanique des Fluides de l'École Centrale Nantes participe depuis le début au projet pluridisciplinaire *ROBEA/RAAMO*. L'intérêt des scientifiques pour l'étude des phénomènes naturels comme la nage des poissons a déjà été évoqué au chapitre 1. Lorsque A. Leroyer [57] a entrepris l'étude numérique de l'écoulement autour d'une anguille, il a fallu qu'il développe des outils rapides de remaillage (remaillage par pondération...). Cependant des problèmes sur les maillages déformés apparaissaient dans le cas de grandes courbures de l'anguille. Le module de remaillage pseudo-structure résout ce problème et permet une déformation propre et contrôlée en 3D (cf. fig. 4.17). On note que les mailles sont correctement déformées suivant l'axe $\overrightarrow{z}$ (cf. fig. 4.17(c) et 4.17(d)).



(a) Vue d'ensemble (coupe 2D $z = 0$)

(b) Zoom sur la queue (coupe 2D $z = 0$)

(c) Vue d'ensemble 3D

(d) Zoom sur la queue

FIG. 4.17: Remaillage autour de l'anguille sur un maillage 3D avec un cisaillement variable localement

4.2.5.3 La configuration “batteur-caisse” (maillage GRIDGEN ™)

Le remailleur ayant été testé sur des corps déformables en mouvement, une étude sur de grandes déformations d'un maillage autour de corps solides s'imposait. Le cas-test du “batteur-caisse” a été retenu : une caisse flotte sur l'eau. Un batteur met en mouvement ce fluide. Les vagues créées viennent heurter la caisse (cf. fig. 4.18). Cela entraîne de grandes déformations autour de la caisse. Cette application s'appuie sur des expériences réalisées en bassin à l'Université Technologique de Berlin (cf. [110]). Elle a été traitée une première fois numériquement au sein de l'EMN dans [57]. Mais, les déformations du maillage n'étaient pas optimales. Le test est donc repris en utilisant un des maillages composé de 21000 cellules (cf. fig 4.18(a)).

Dans un premier temps, pour évaluer les déformations maximales admissibles par le maillage, des mouvements imposés sur la caisse ont été testés. En effet, seule la caisse peut poser problème : le batteur subit un faible mouvement qui est parfaitement géré par le module de remaillage. La caisse, quant à elle, est libre : son mouvement est composé de translations et de rotations.

Une translation pure de 5 fois la longueur de la caisse a été appliquée à celle-ci (cf. fig. 4.19). La comparaison entre le maillage proche de la paroi avant et après translation nous donne entière satisfaction.

Une rotation pure de $\pm 45°$ a été appliquée à la caisse (cf. fig. 4.20). La comparaison entre le maillage près de la paroi avant et après rotation nous donne encore entière satisfaction.

Un calcul multifluide a ensuite été lancé avec les paramètres optimaux pour le remaillage. La simulation, qui s'étale sur $10\,s$, a eu un coût en temps de calcul de 23 heures. Auparavant, en employant la méthode des ressorts, le cas-test durait au moins 1 semaine. Le gain est donc significatif.

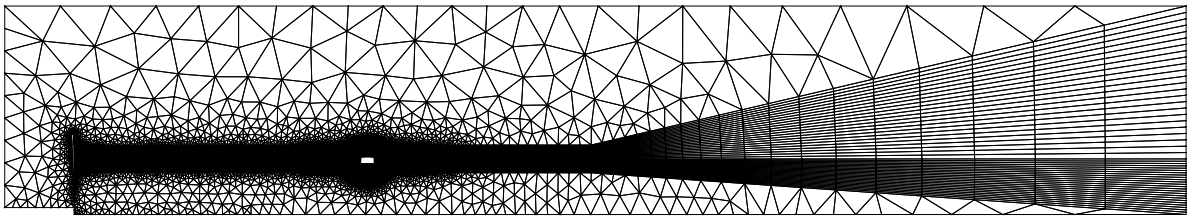
Mais ce qui nous intéresse le plus, c'est la déformation du maillage autour de la caisse. Le passage des deux vagues formées par le mouvement du batteur entraîne un mouvement violent de la caisse (cf. fig. 4.21). Le module de remaillage employé avec la discrétisation à coefficient de cisaillement variable donne de très bons résultats : le maillage ne se déforme pas près de la caisse et les mailles restent orthogonales à la paroi (cf. fig. 4.22).

4.2.5.4 Corps sphérique en translation imposée (maillage HEXPRESS ™)

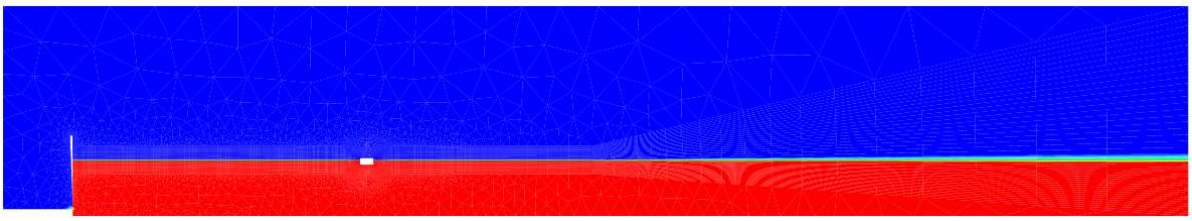
Les exemples de maillages précédents sont tous réalisés avec l'outil GRIDGEN ™. L'équipe numérique du Laboratoire de Mécanique des Fluides de Centrale Nantes travaille maintenant avec un autre outil de génération de maillages développé par NUMECA [3] : HEXPRESS ™. Les maillages obtenus sont très différents. On a donc voulu tester le module de remaillage sur ceux-ci.

Un corps sphérique (cf. fig. 4.23) est translaté de plus de 10 fois son diamètre (cf. fig. 4.24). En comparant les agrandissements faits autour du corps, on observe que le maillage ne s'est pas déformé près du cercle. On voit ici l'influence des paramètres E et G. Notons que les croix apparaissant sur la figure 4.24(a) sont dues à un problème d'interpolation du visualiseur (TECPLOT ™ [4]). Ce n'est pas une modification de la structure du maillage.

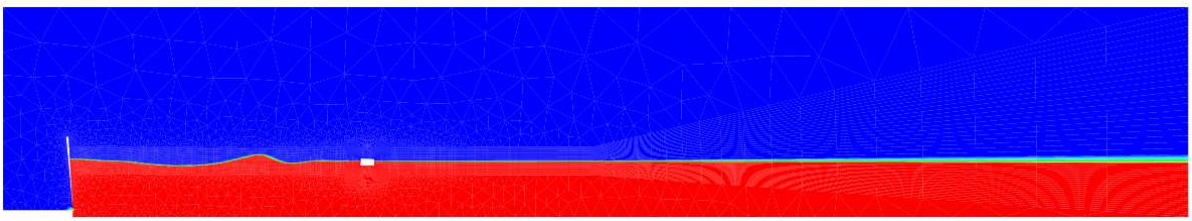
[3] http://www.numeca.com
[4] http://www.tecplot.com



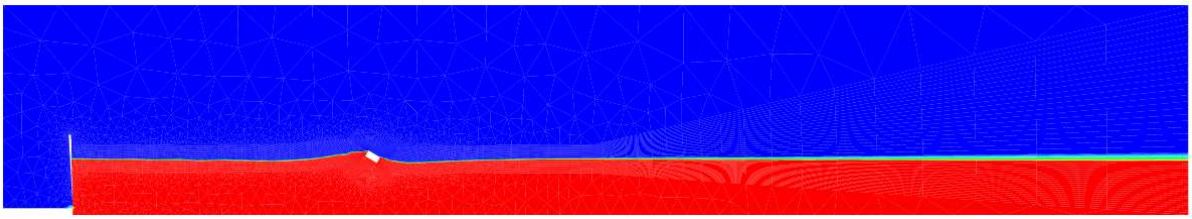
(a) Maillage de référence du “batteur-caisse”



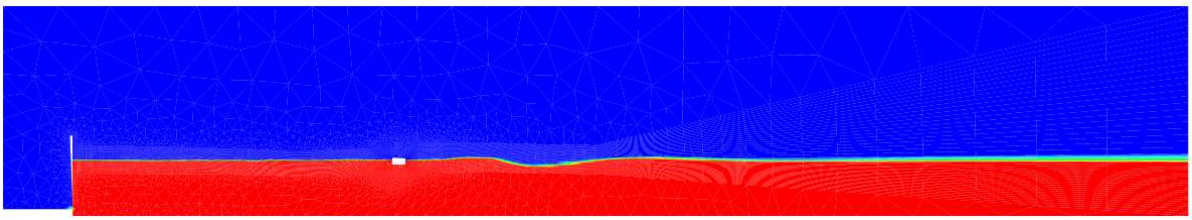
(b) Configuration de départ du “batteur-caisse”



(c) Formation de vagues dues aux oscillations du batteur

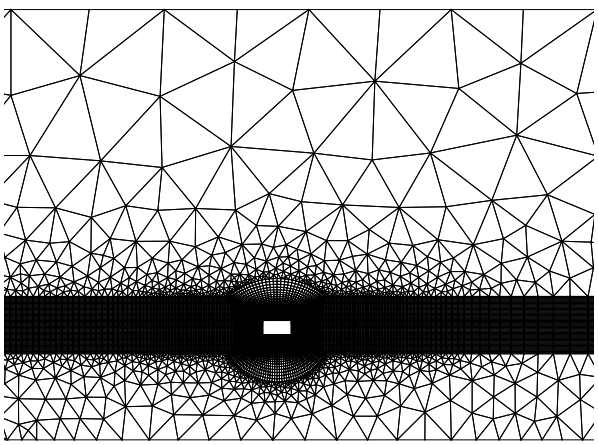


(d) Mouvement de la caisse dû aux vagues

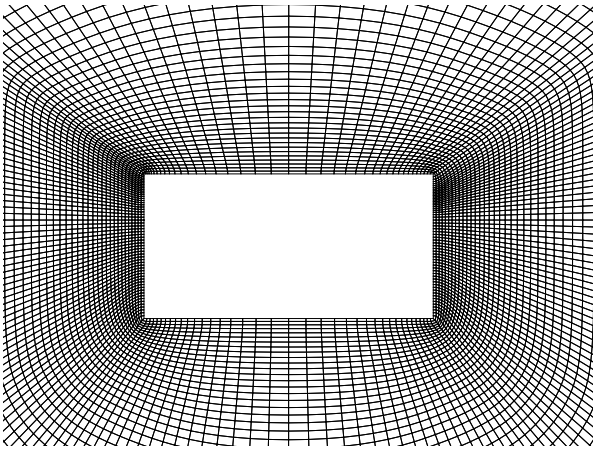


(e) Stabilisation progressive de la caisse après le passage des vagues

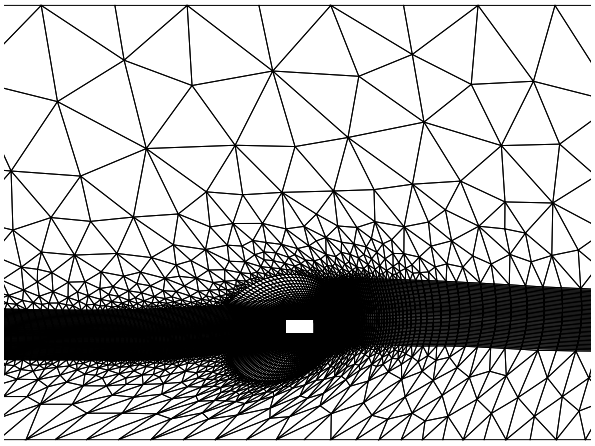
FIG. 4.18: Présentation globale du cas-test du “batteur-caisse”



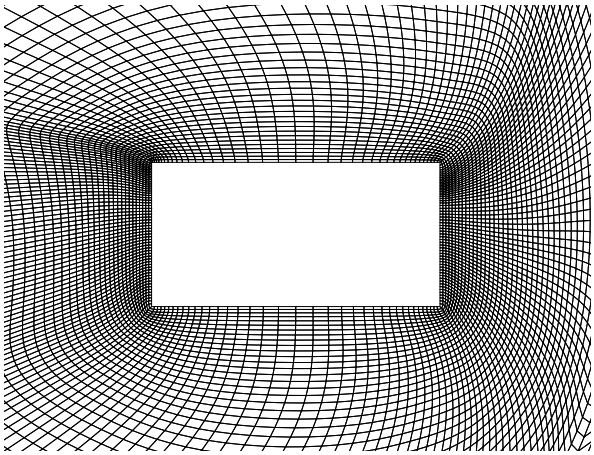
(a) Configuration de départ du “batteur-caisse”



(b) Maillage initial autour de la caisse

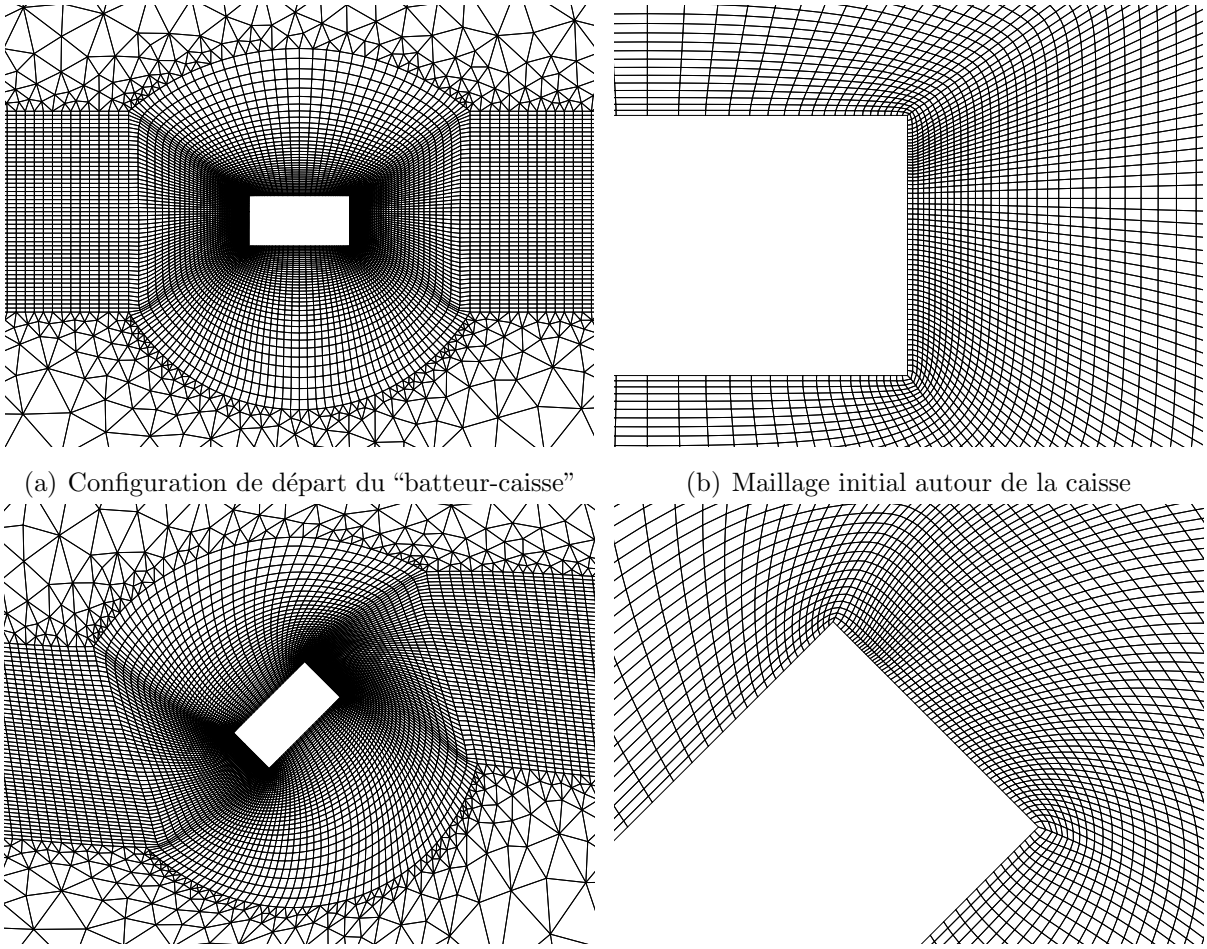


(c) Configuration déformée (translation imposée de 5 fois la longueur de la caisse



(d) Maillage déformé autour de la caisse

Fig. 4.19: Test de remaillage avec un mouvement de translation pure de 5 fois la longueur de la caisse



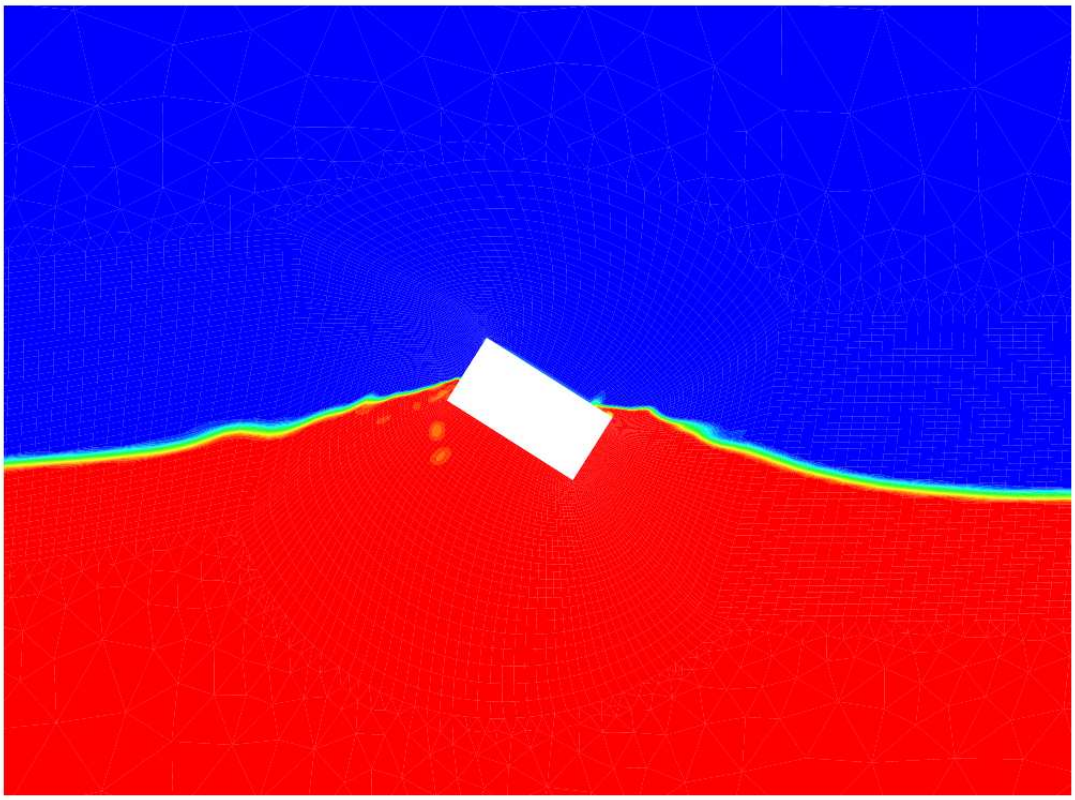
(a) Configuration de départ du “batteur-caisse”

(b) Maillage initial autour de la caisse

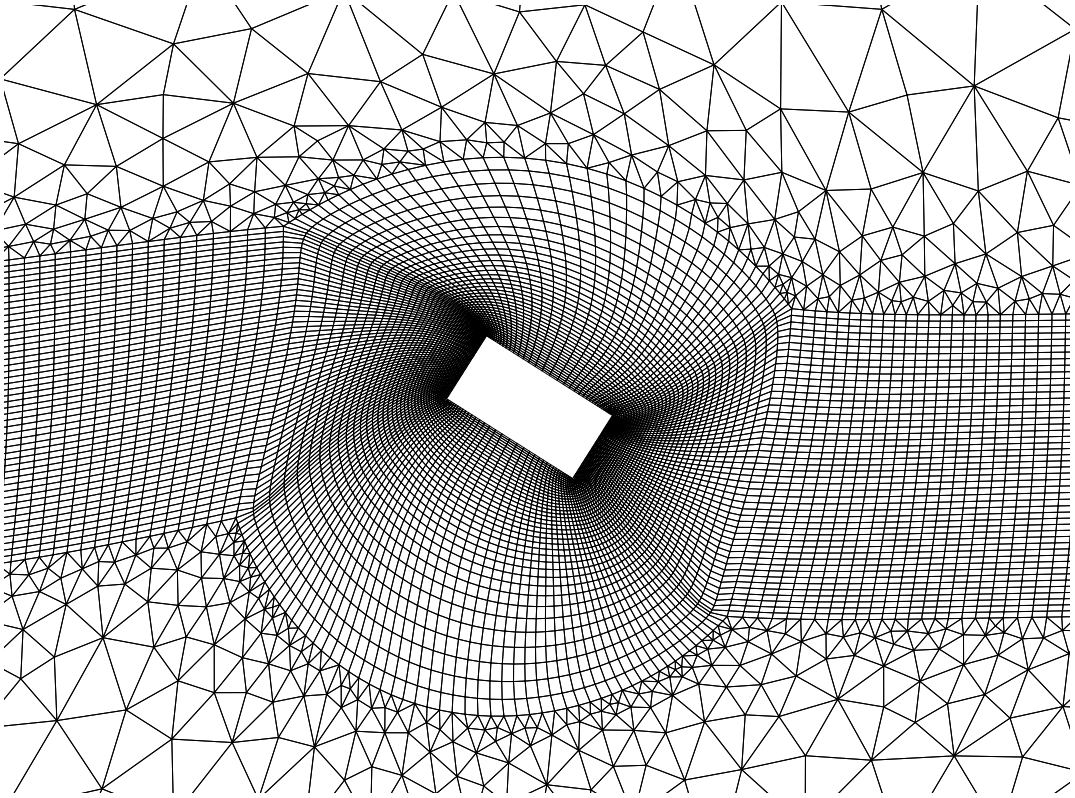
(c) Configuration déformée (rotation imposée de 45°)

(d) Maillage déformé autour de la caisse

FIG. 4.20: Test de remaillage avec un mouvement de rotation pure de 45°

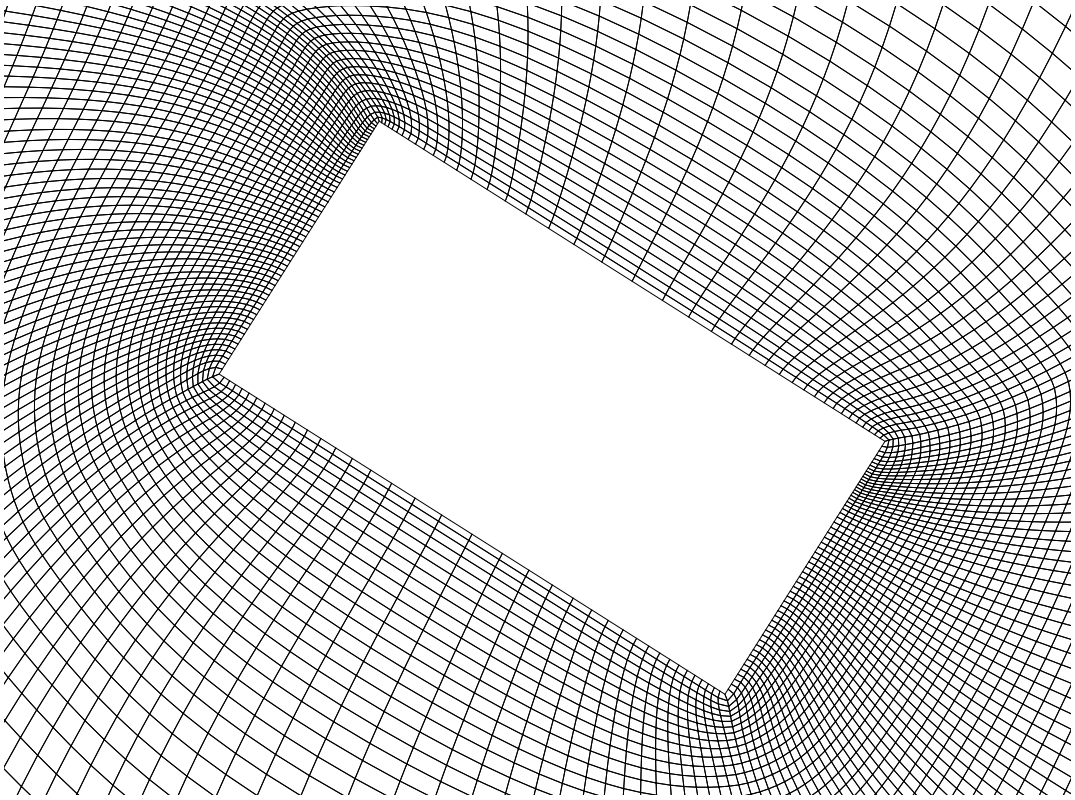


(a) Passage de la seconde vague

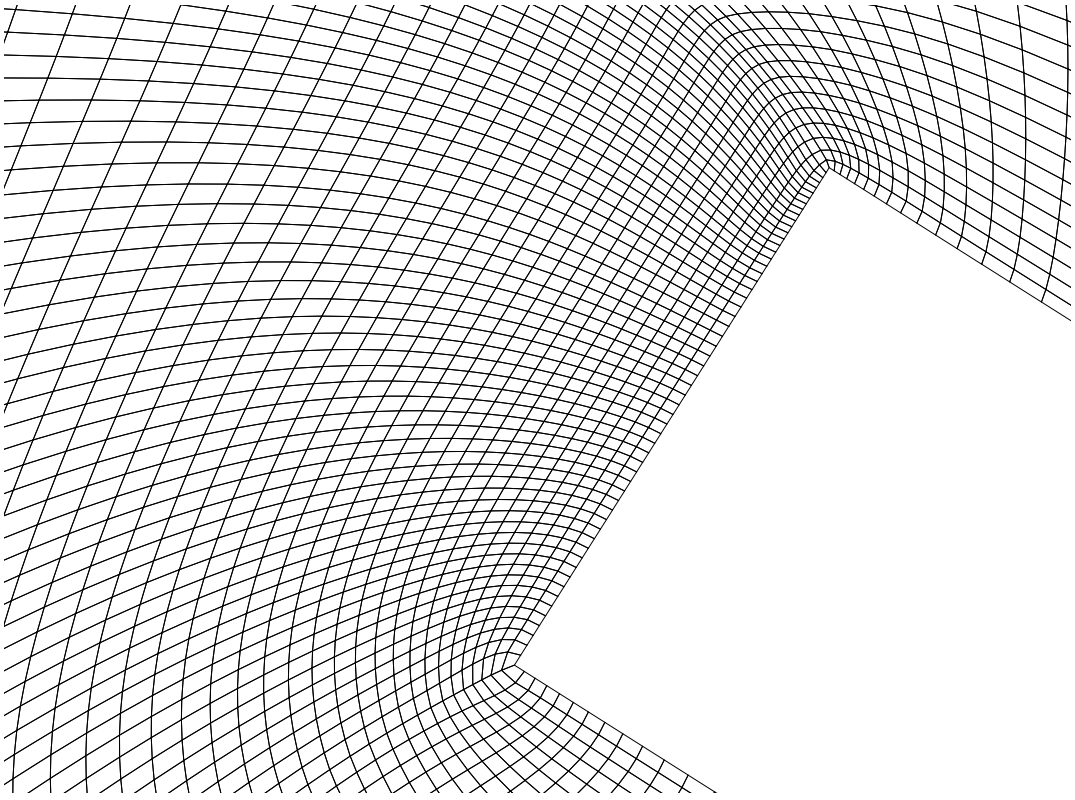


(b) Maillage déformé

FIG. 4.21: Passage de la seconde vague au niveau de la caisse

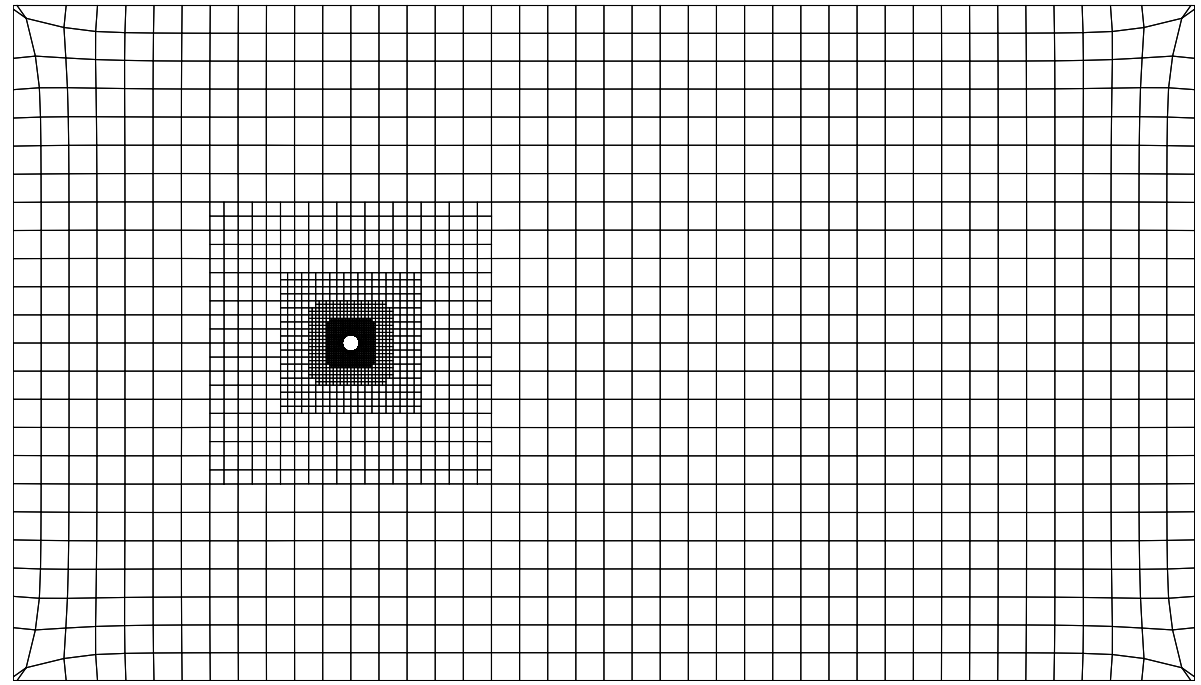


(a) Zoom sur la caisse

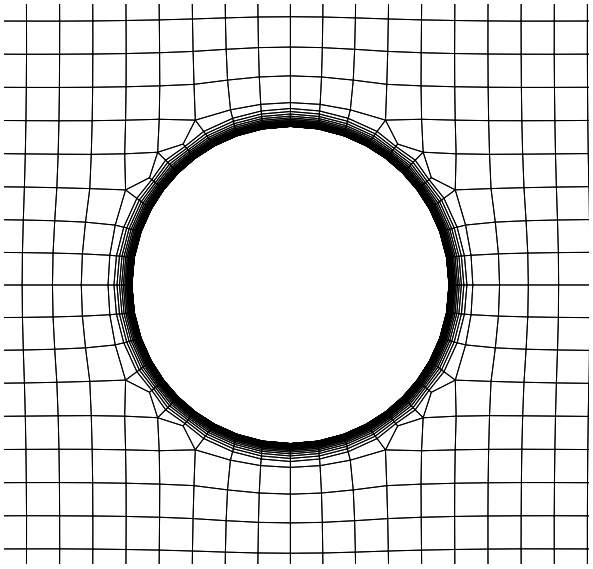


(b) Zoom sur les coins gauches de la caisse

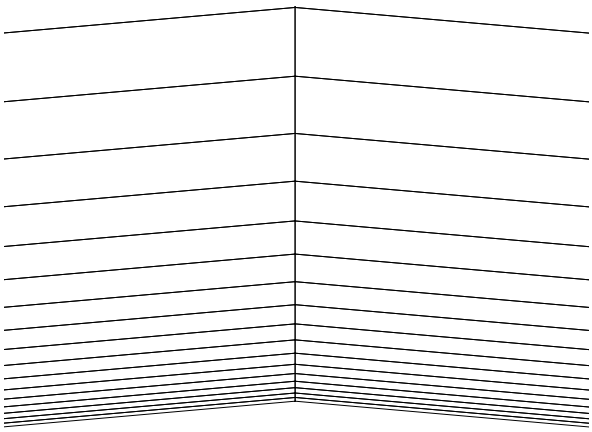
FIG. 4.22: Détail de la déformation du maillage autour de la caisse au passage de la seconde vague



(a)

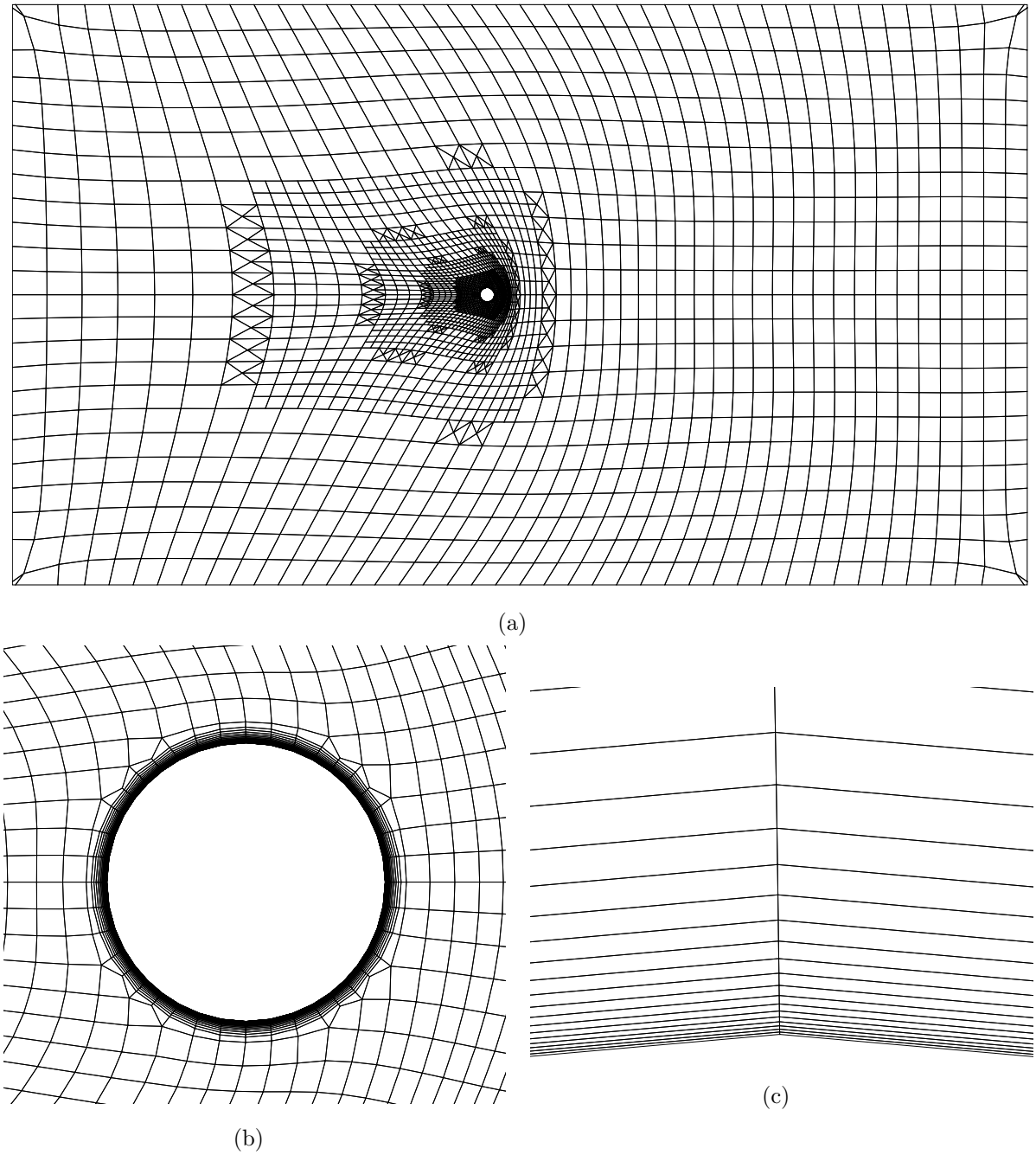


(b)



(c)

FIG. 4.23: Maillage HEXPRESS™ autour d'un corps sphérique



(a)

(b)

(c)

FIG. 4.24: Maillage HEXPRESS™ autour du corps sphérique translaté

4.2.5.5 Projet *VIRTUE* : optimisation d'une coque de bateau (maillage HEXPRESS™)

L'EMN participe depuis son lancement au projet européen *VIRTUE* [5] (The VIRtual Tank Utility in Europe) (cf. [101]). L'objectif est de rassembler des outils de CFD afin de créer un environnement de simulation pour les bateaux.

Dans ce cadre, un travail d'optimisation d'une coque d'un tanker de $320\,m$ de long (cf. fig. 4.25) a été entrepris en partenariat avec FRIENDSHIP-Framework [6] : la résistance à l'avancement et la qualité du sillage doivent être améliorées. Pour cela, **ISIS** est couplé au logiciel d'optimisation de FRIENDSHIP-Framework. Ce dernier génère une nouvelle forme de la coque du tanker à chaque nouveau calcul d'optimisation. **ISIS** doit donc pouvoir générer un maillage ayant pour frontière cette surface. Cette tâche incombe au module de remaillage pseudo-structure. Il déforme le maillage de base pour être cohérent avec la nouvelle forme de la coque du tanker.

Actuellement, l'EMN est en phase de validation de la procédure générale d'optimisation. Diverses formes non-réalistes de carènes ont été testées par P. Queutey pour observer le comportement du module de remaillage sur des cas limites (cf. fig. 4.26, 4.27 et 4.28).

4.2.5.6 Le requin (maillage HEXPRESS™)

Avec le projet *ROBEA/RAAMO*, l'EMN a fait du biomimétisme un de ses axes de recherche. Mais pourquoi ne pas faire des calculs sur des poissons dont la forme est plus complexe que celle d'une anguille ? Le principal problème est le processus de remaillage. En effet, sur un corps de forme complexe en 3D avec de fortes déformations, toutes les difficultés sont réunies. Avant l'apparition du module pseudo-structure dans **ISIS**, ce type d'application était impossible à traiter. Testons dorénavant le programme sur une modélisation de requin.

La forme d'un requin a été récupérée sur internet[7], puis simplifiée (cf. fig. 4.29). Un maillage HEXPRESS™ a ensuite été réalisé (par D. Dauby [27]). Il est composé de 645899 cellules pour 736588 nœuds. Il est raffiné au niveau du corps. Une attention particulière a été portée autour des ailerons et de la queue (cf. fig. 4.30 et 4.31).

Le calcul a été lancé par A. Leroyer [57] en parallèle sur 6 processeurs. Cela permet de valider le module de remaillage pour le calcul MPI. Le mouvement du requin a été généré et commandé de la même manière que pour l'anguille du projet *ROBEA/RAAMO* (cf. [58] [59] [15]). Le requin a nagé pendant $5\,s$, temps où nous avons décidé de stopper le calcul pour analyser les résultats. Le temps CPU consommé pour un tel calcul parallèle est d'une semaine environ.

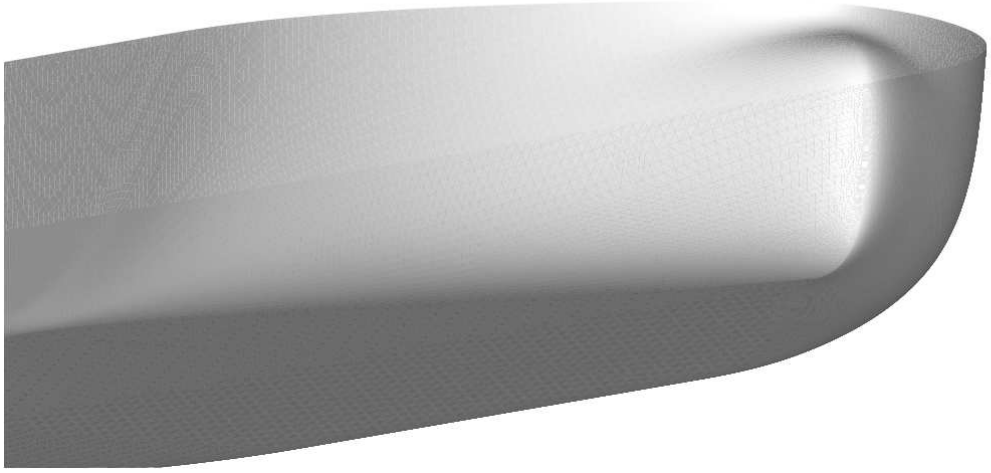
Le maillage déformé produit par le module pseudo-structure est très correct non loin du corps dans les trois directions : les mailles conservent leur forme originelle et s'aplatissent peu (cf. fig. 4.32, 4.33 et 4.34).

Intéressons-nous maintenant aux cellules près de la paroi et tout particulièrement autour de la queue (cf. fig. 4.35) et autour de l'extrémité supérieure de la queue (cf. fig. 4.36). Nous pouvons constater que le maillage reste orthogonal à la paroi et ce sans subir d'alté-

[5] http://www.virtual-basin.org
[6] http://www.friendship-systems.com
[7] http://toucan.web.infoseek.co.jp/3DCG/3ds/FishModelsE.html

(a) Vue de dessous

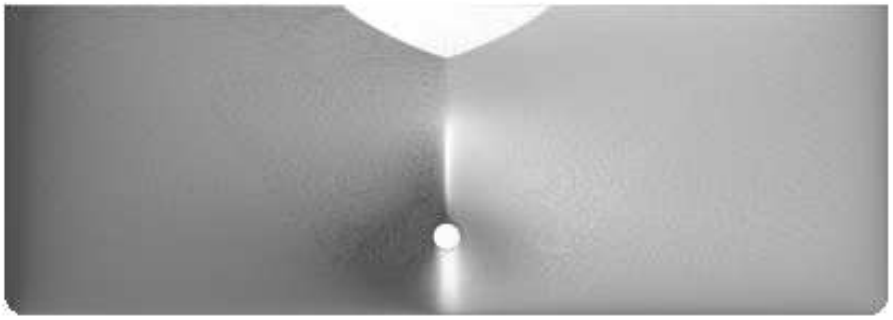
(b) Vue de côté



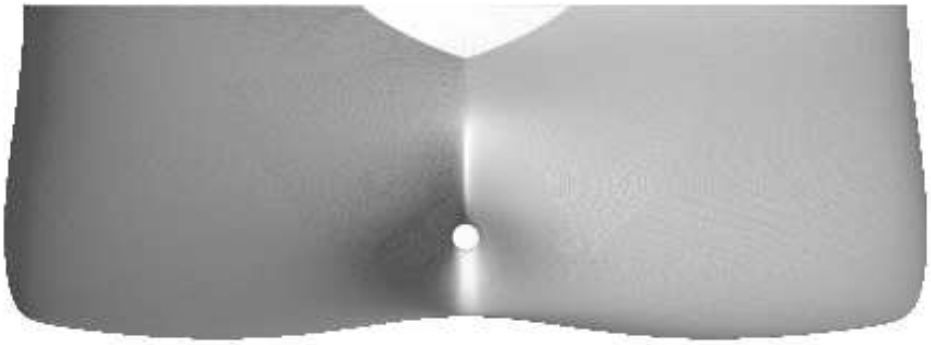
(c) Vue de l'avant

(d) Vue de l'arrière

FIG. 4.25: Forme initiale de la coque du tanker



(a) Coque initiale

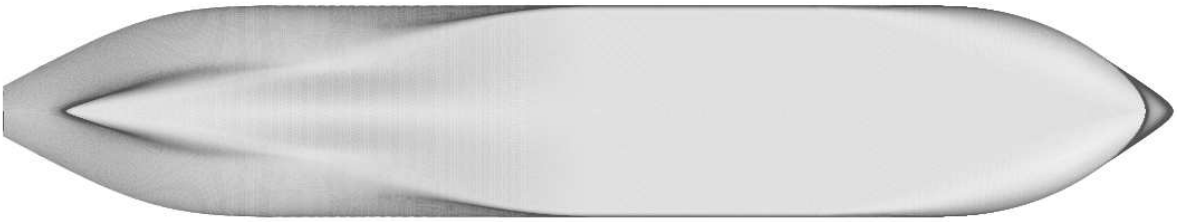


(b) $1^{ère}$ forme de coque déformée

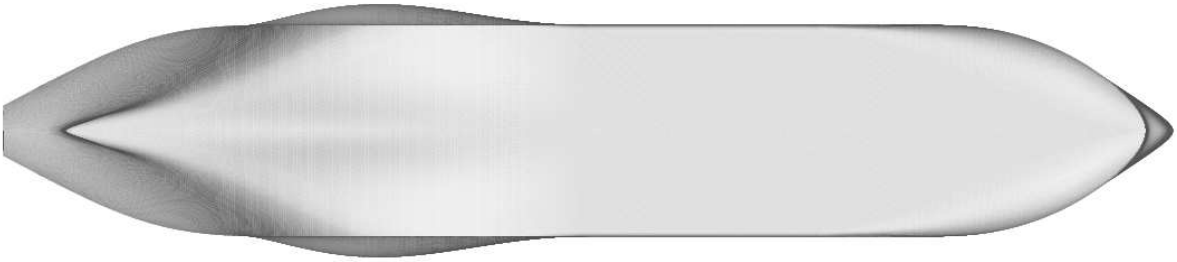
(c) 2^{nde} forme de coque déformée

FIG. 4.26: Deux déformations différentes de la coque vues de l'arrière du tanker

(a) Coque initiale



(b) $3^{ème}$ forme de coque déformée



(c) $4^{ème}$ forme de coque déformée

FIG. 4.27: Deux autres différentes déformations de la coque vues du dessous du tanker

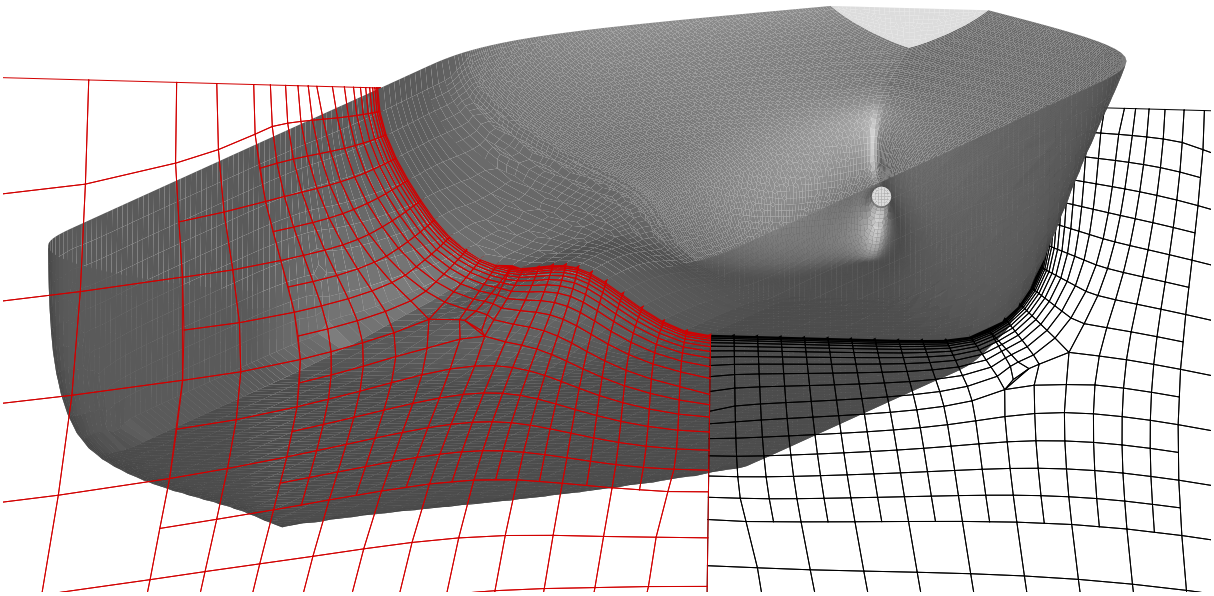


FIG. 4.28: Maillage déformé (en rouge, à gauche) ; maillage de référence (en noir, à droite)

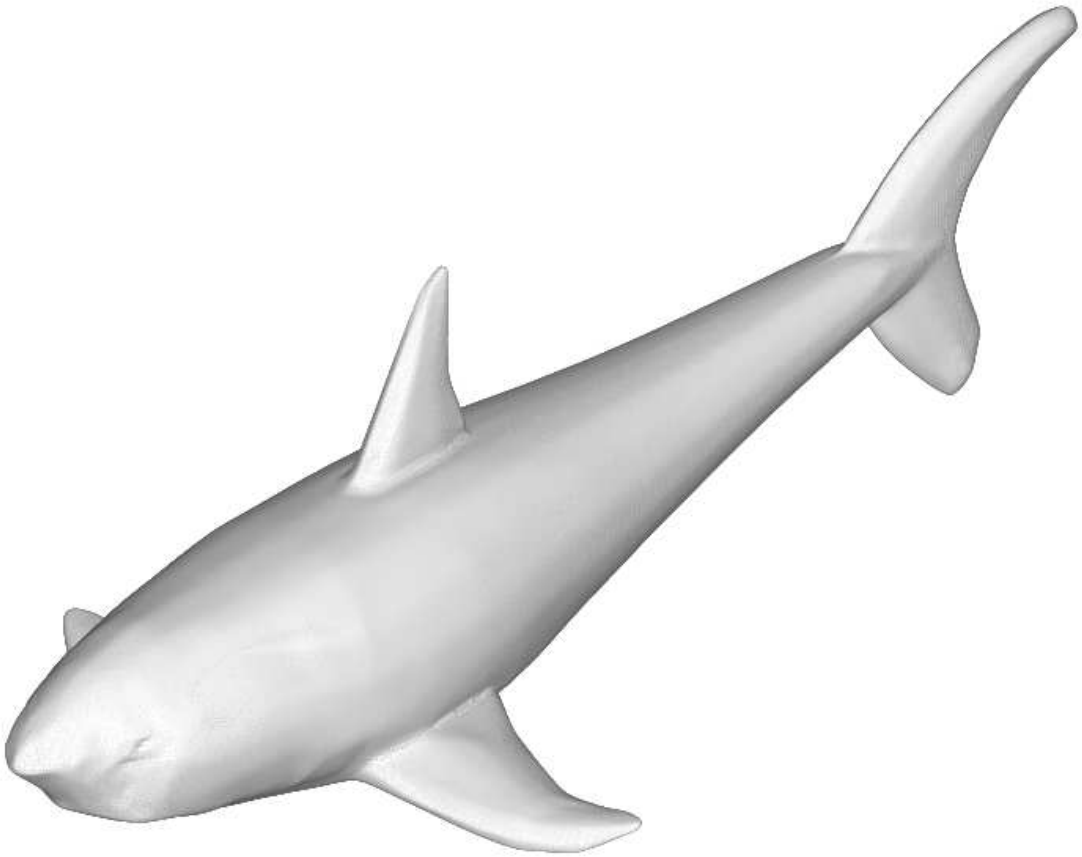


FIG. 4.29: Modélisation d'un requin

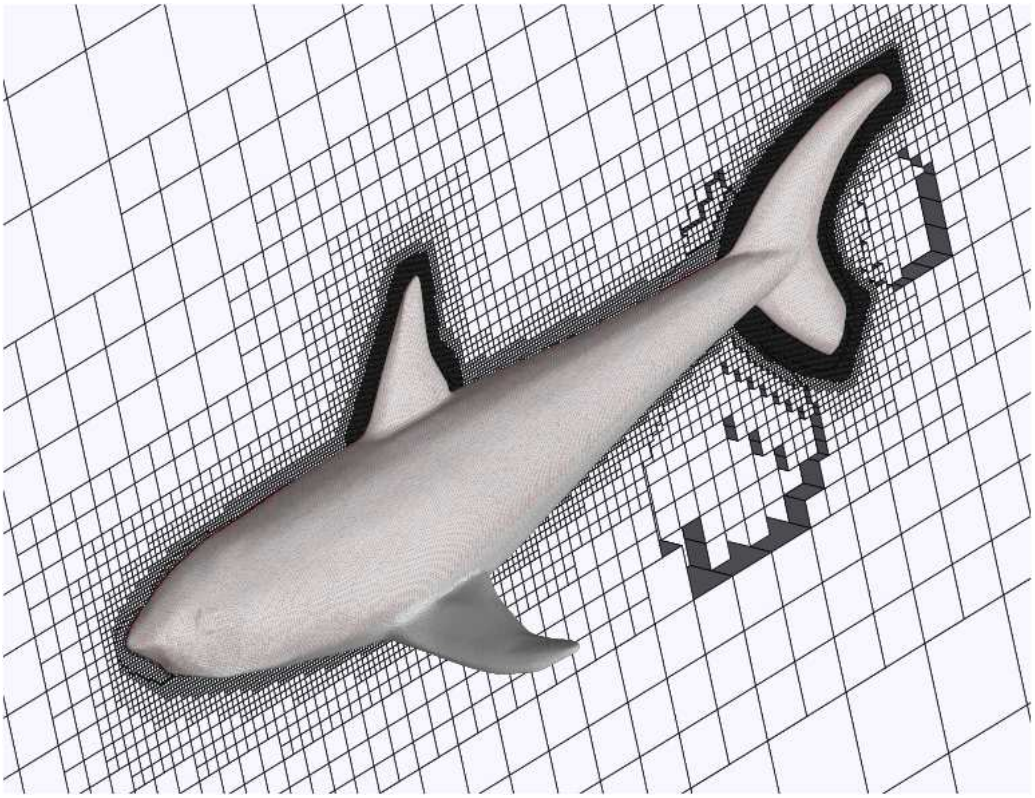


FIG. 4.30: Maillage HEXPRESS™ autour du requin (Vue générale de côté)

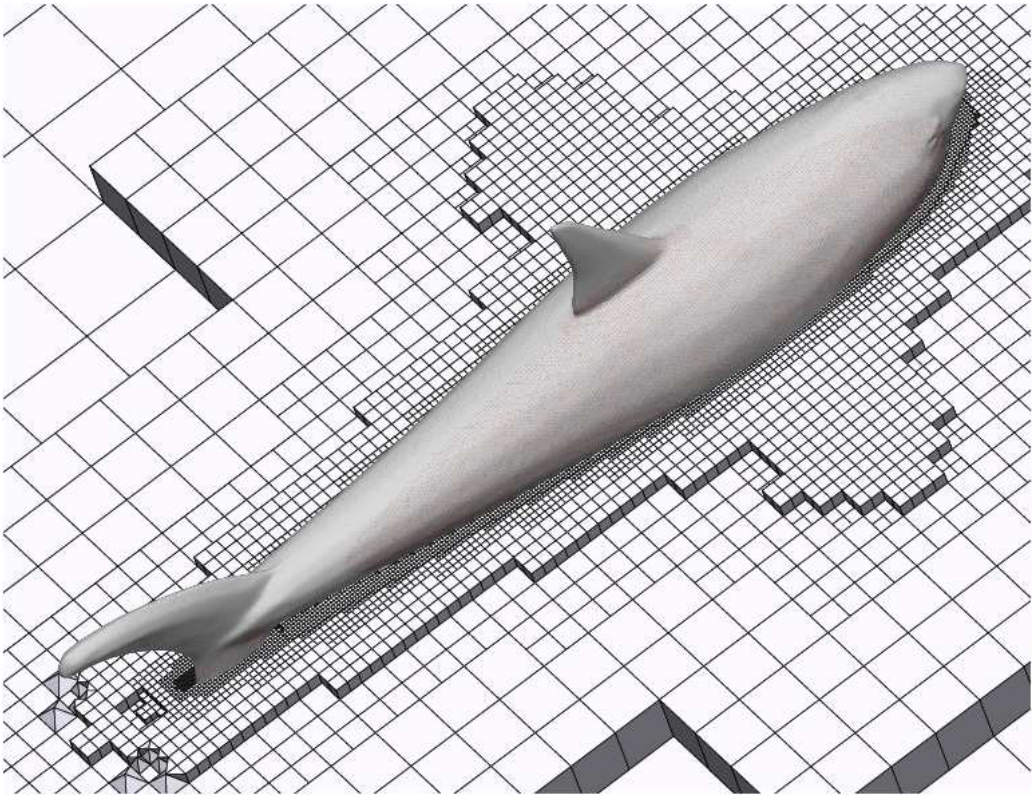


FIG. 4.31: Maillage HEXPRESS™ autour du requin (Vue générale de dessus)

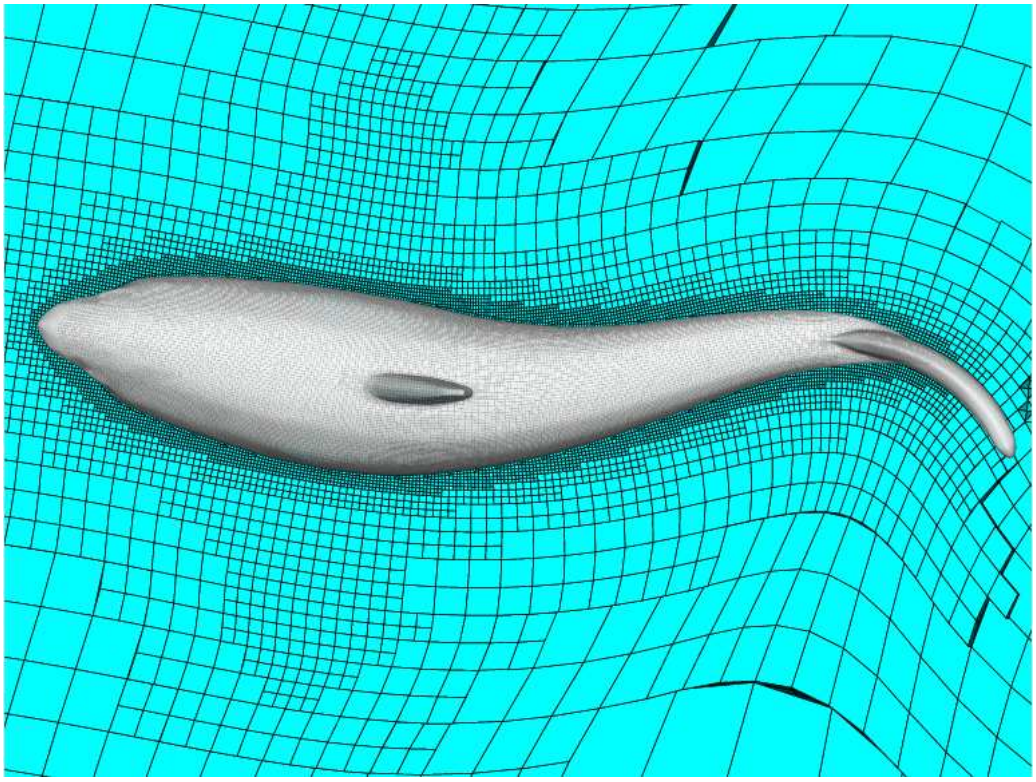


FIG. 4.32: Maillage HEXPRESS™ déformé autour du requin qui nage (Vue générale de dessus)

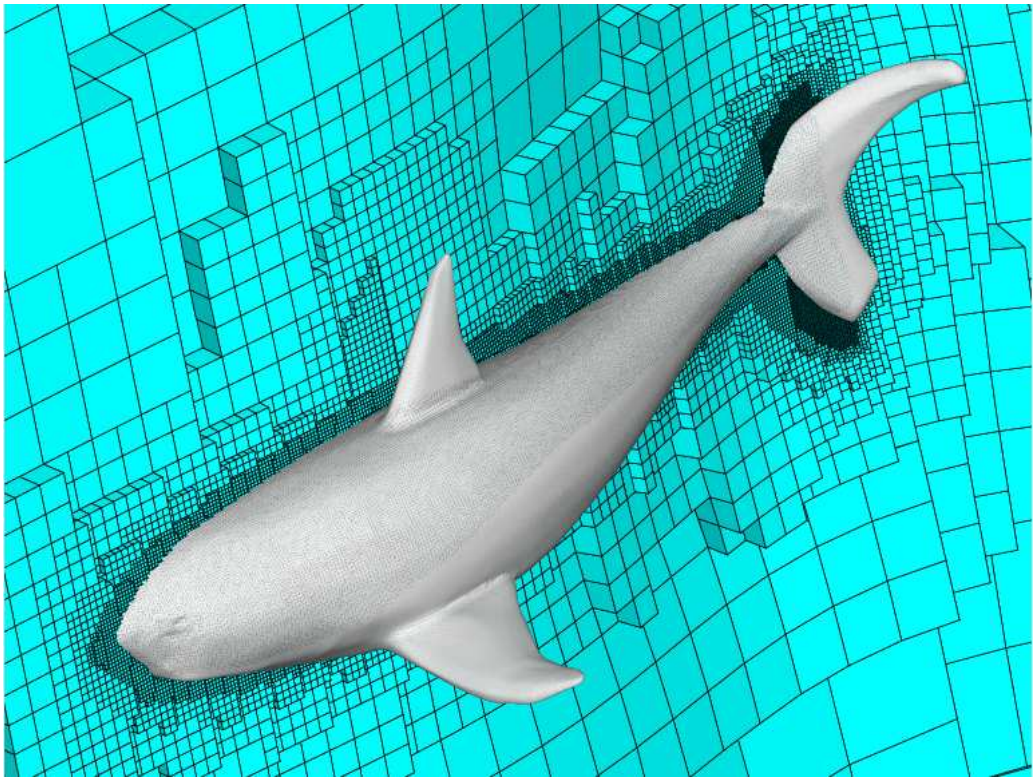


FIG. 4.33: Maillage HEXPRESS™ déformé autour du requin qui nage (Vue générale de côté)

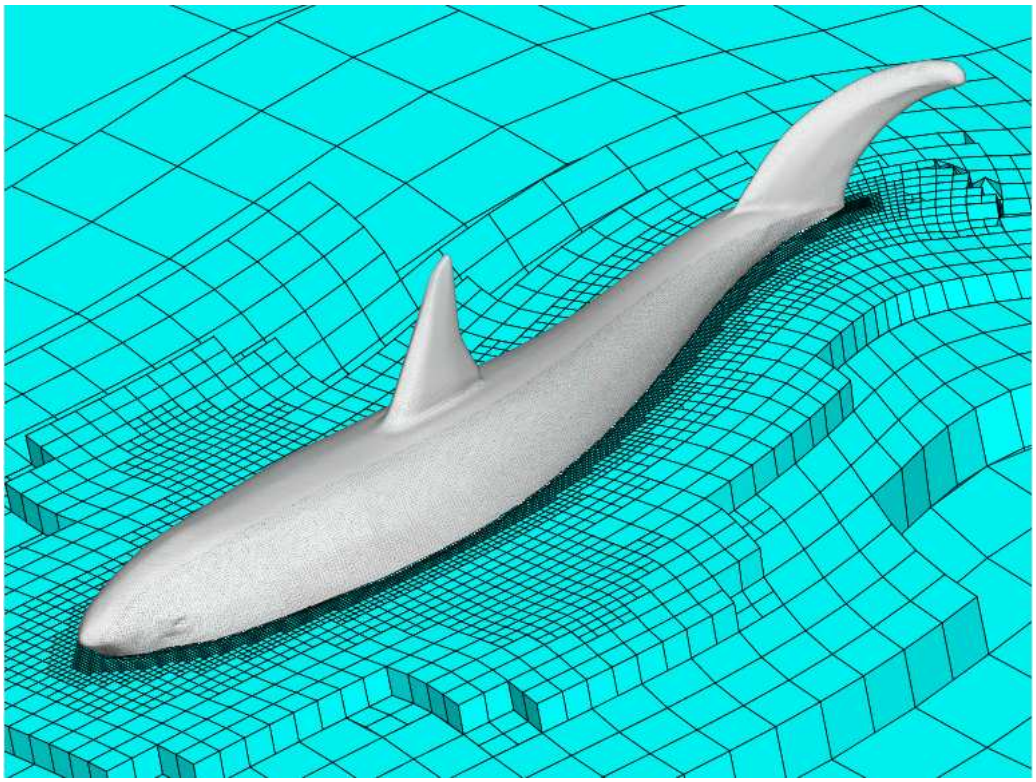


FIG. 4.34: Maillage HEXPRESS ™ déformé autour du requin qui nage (Vue générale de dessus)

ration. La déformation du maillage autour des nageoires dorsale (cf. fig. 4.37) et pectorales (cf. fig. 4.38) est elle aussi très bonne. L'objectif du module pseudo-structure est donc pleinement atteint.

4.3 Couplage des solutions de remaillage : la solution idéale ?

Les exemples présentés jusqu'ici sont le résultat de l'emploi du module pseudo-structure seul. Cependant, ce dernier peut, en présence de corps avec des coins associés à de grands déplacements, ne pas répondre à nos exigences au niveau de la qualité du maillage. En effet, même en augmentant la valeur du pseudo-module d'Young, E, ou encore celui du pseudo-cisaillement, G, des mailles couchées peuvent apparaître à proximité de la paroi (cf. fig. 4.40). On a donc eu l'idée de *combiner les avantages de plusieurs algorithmes de remaillage pour en exclure leurs inconvénients respectifs*. Le module de remaillage par mouvement en bloc ne peut être utilisé que dans un cadre très restreint, mais offre évidemment un maillage aussi satisfaisant que celui de départ. Le module de remaillage par pondération donne des résultats très bons près des corps (sauf en très fort déplacement), mais pose des problèmes de chevauchement au loin. Le remaillage pseudo-solide génère des résultats parfaits au loin mais peut présenter des difficultés autour des corps comportant des angles aigus en grand déplacement.

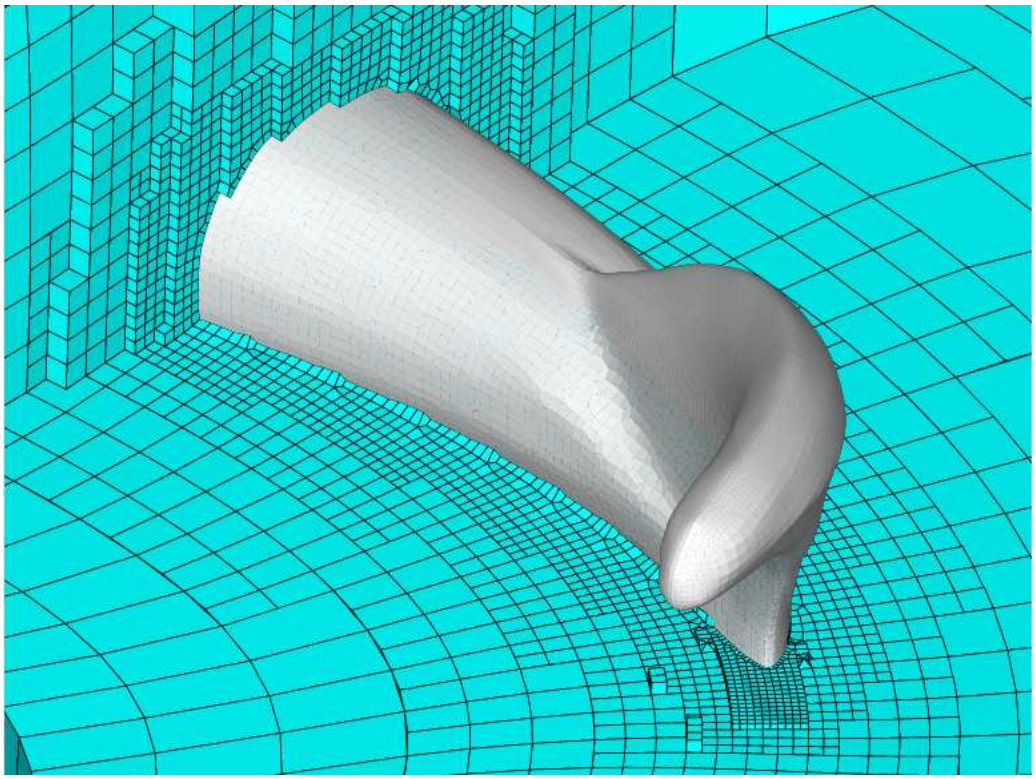


FIG. 4.35: Maillage HEXPRESS™ déformé autour de la queue du requin

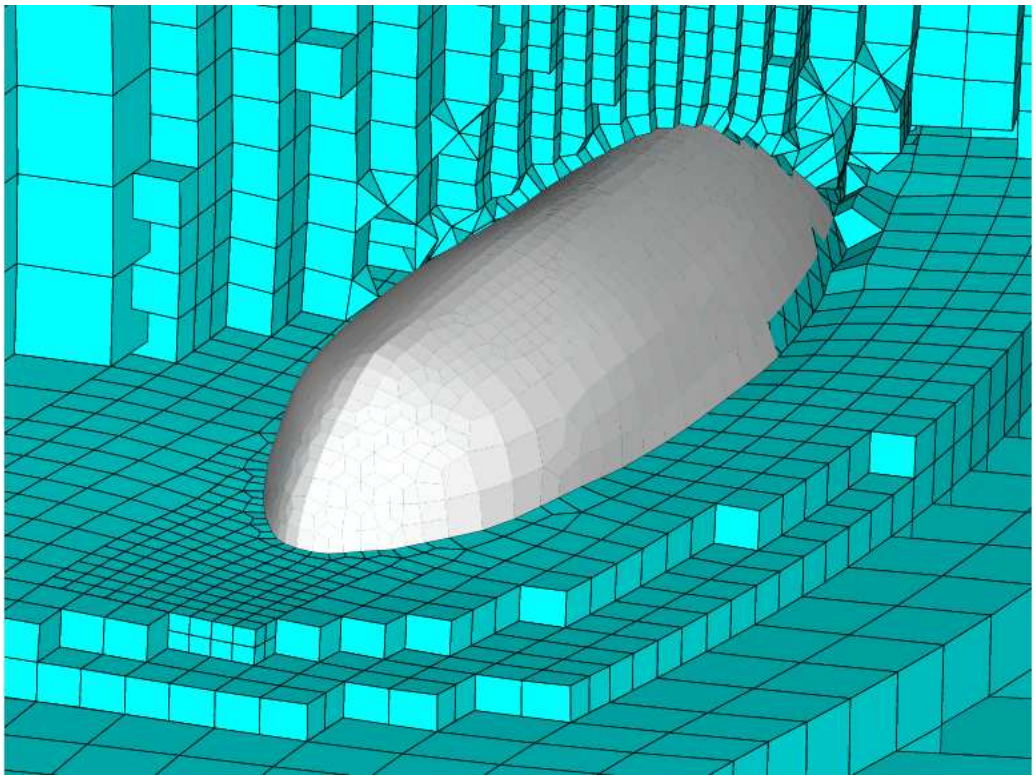


FIG. 4.36: Maillage HEXPRESS™ déformé autour de l'extrémité de la queue du requin

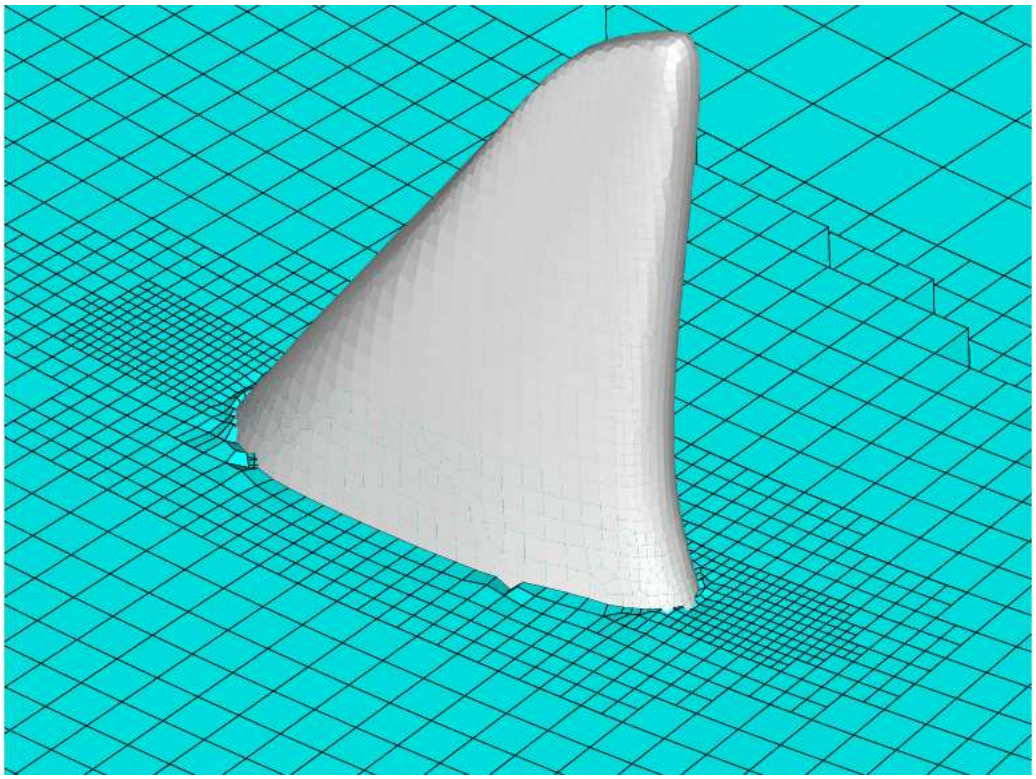


FIG. 4.37: Maillage HEXPRESS™ déformé autour de la nageoire dorsale du requin

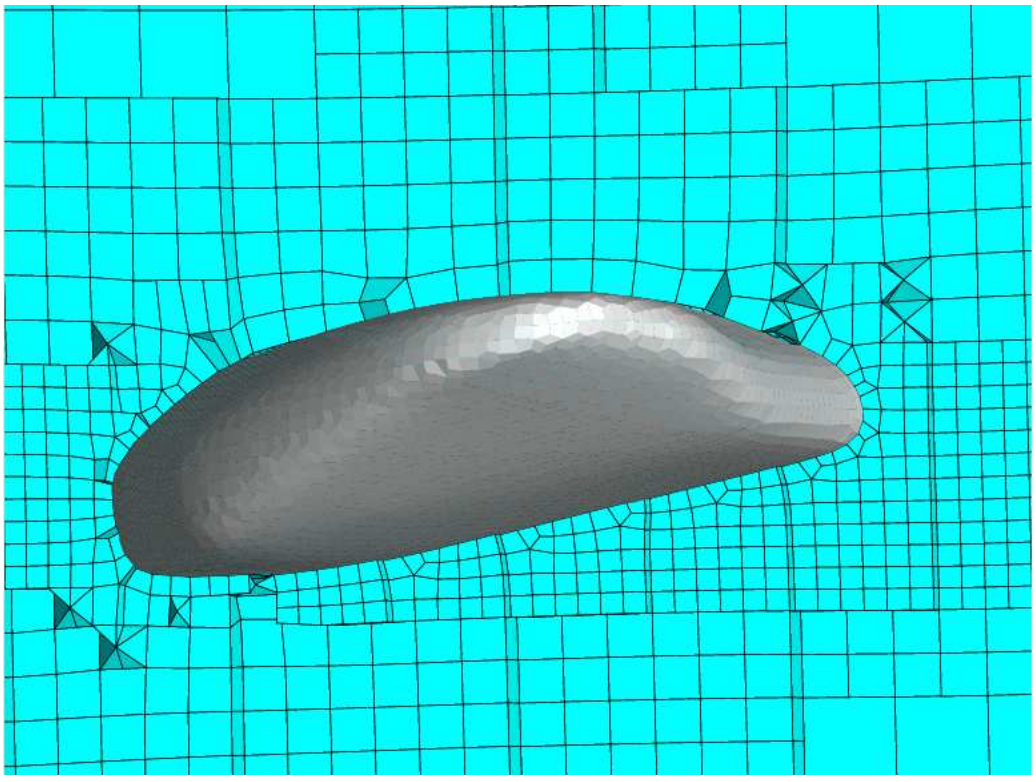


FIG. 4.38: Maillage HEXPRESS™ déformé autour d'une des nageoires pectorales du requin

Ces trois méthodes peuvent être employées en même temps pour une déformation du maillage : trois zones sont contrôlables via des paramètres au sein du fichier de simulation. La zone la plus près du corps contient le remaillage par mouvement en bloc. Vient ensuite celle du remaillage par pondération et enfin celle du module pseudo-structure. Les limites entre ces différentes parties sont données par des critères spatiaux. C'est le coefficient de pondération, calculé pour l'algorithme de remaillage du même nom, qui nous les fournit. Par exemple, le mouvement en bloc a lieu dans l'espace où le coefficient de pondération, $Coef_{Pond}$, est supérieur ou égal à 0.9, et la déformation pseudo-solide pour $Coef_{Pond} < 0.8$. Dans ce cas, la méthode par pondération se situe pour $0.8 \leq Coef_{Pond} < 0.9$.

Pour illustrer ces propos, une forme simple, un losange (cf. fig. 4.39) subit une rotation de 90° . En utilisant le module pseudo-structure seul avec les valeurs de E et de G par défaut, le maillage déformé n'est pas acceptable, car des chevauchements de cellules apparaissent aux extrémités du corps (cf. fig. 4.40(a) et 4.40(b)). En augmentant E et G, ce n'est toujours pas admissible.

On décide donc d'avoir un mouvement en bloc près du corps ($Coef_{Pond} < 0.8$), puis d'utiliser le remaillage pseudo-solide. Le maillage obtenu est très bon et régulier (cf. fig. 4.41(a) et 4.41(b)). Pour obtenir un tel résultat, le coefficient de cisaillement G a été fixé à $G = 1000\mu$.

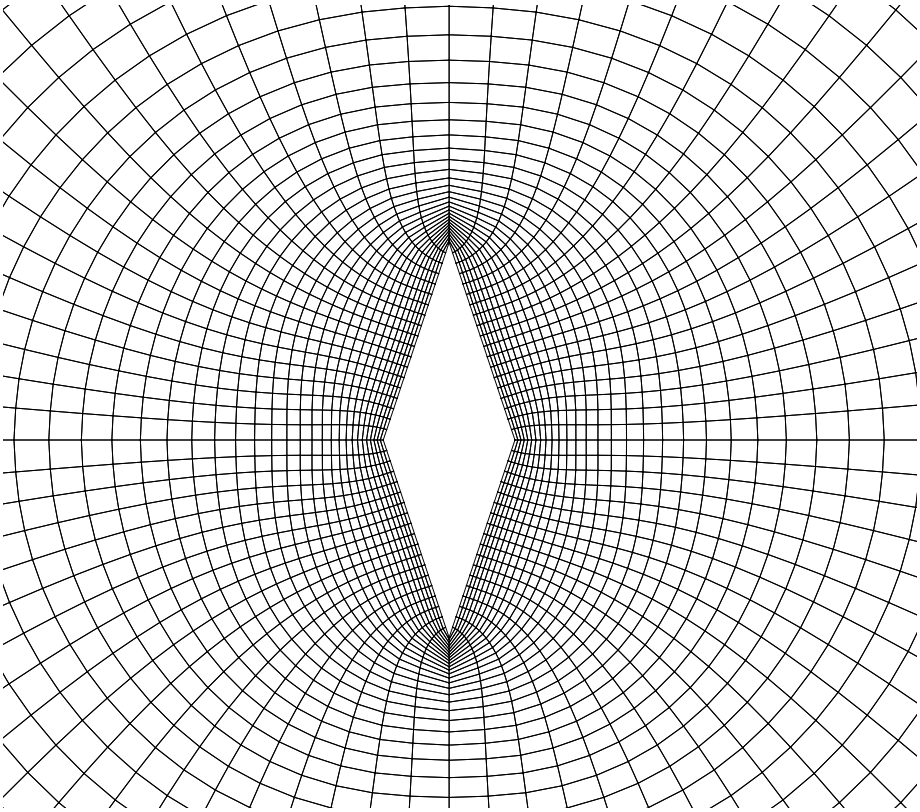
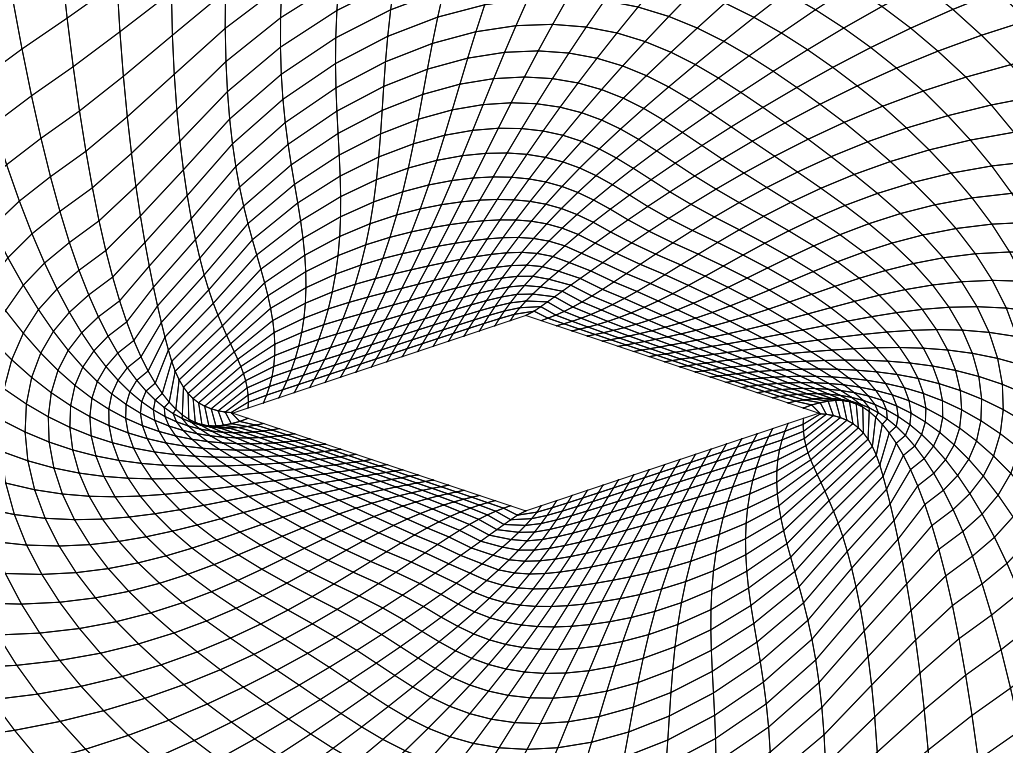
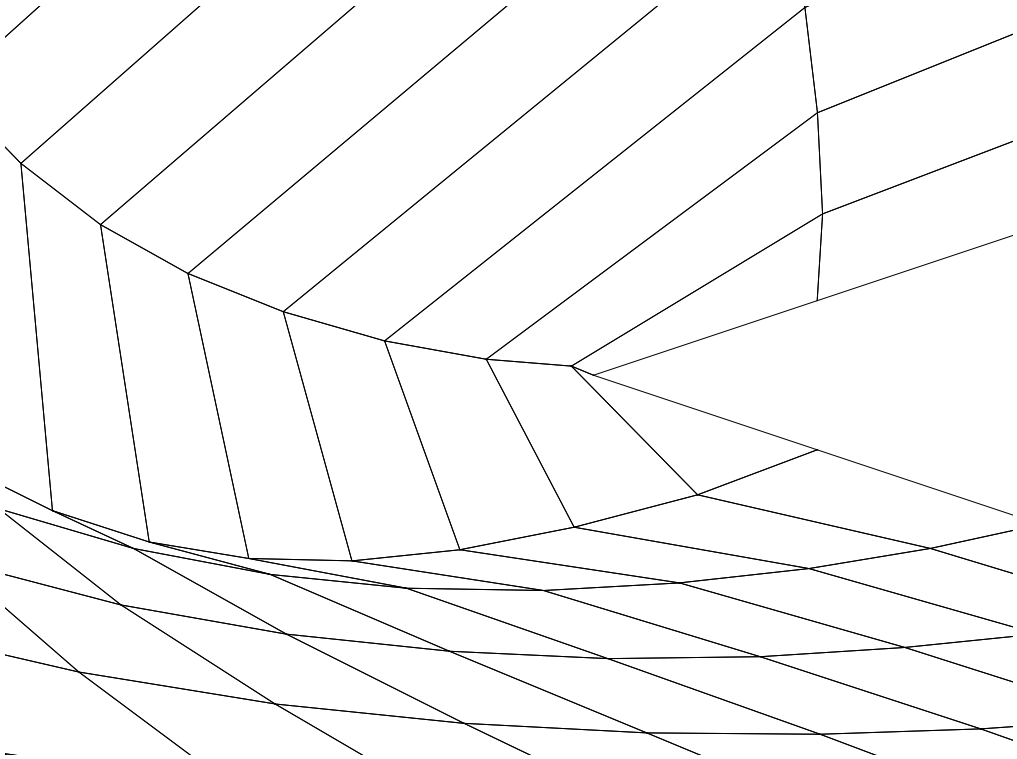


Fig. 4.39: Maillage initial autour d'un losange comportant deux coins avec un angle très aigu

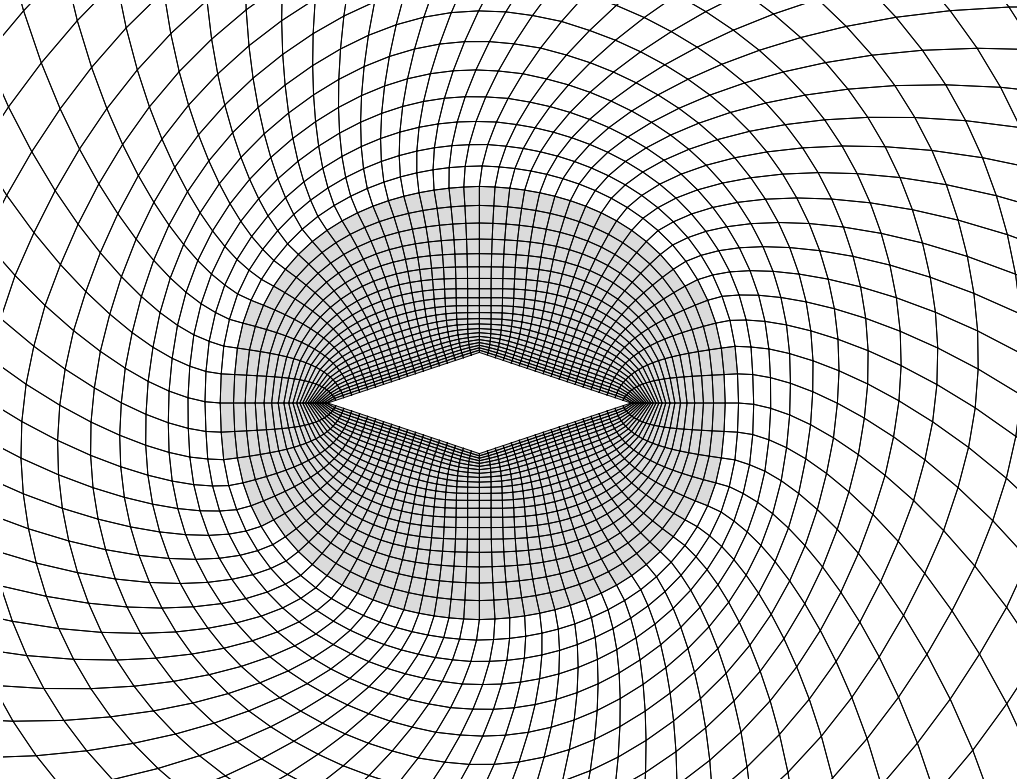


(a) Maillage déformé après rotation de 90° du losange

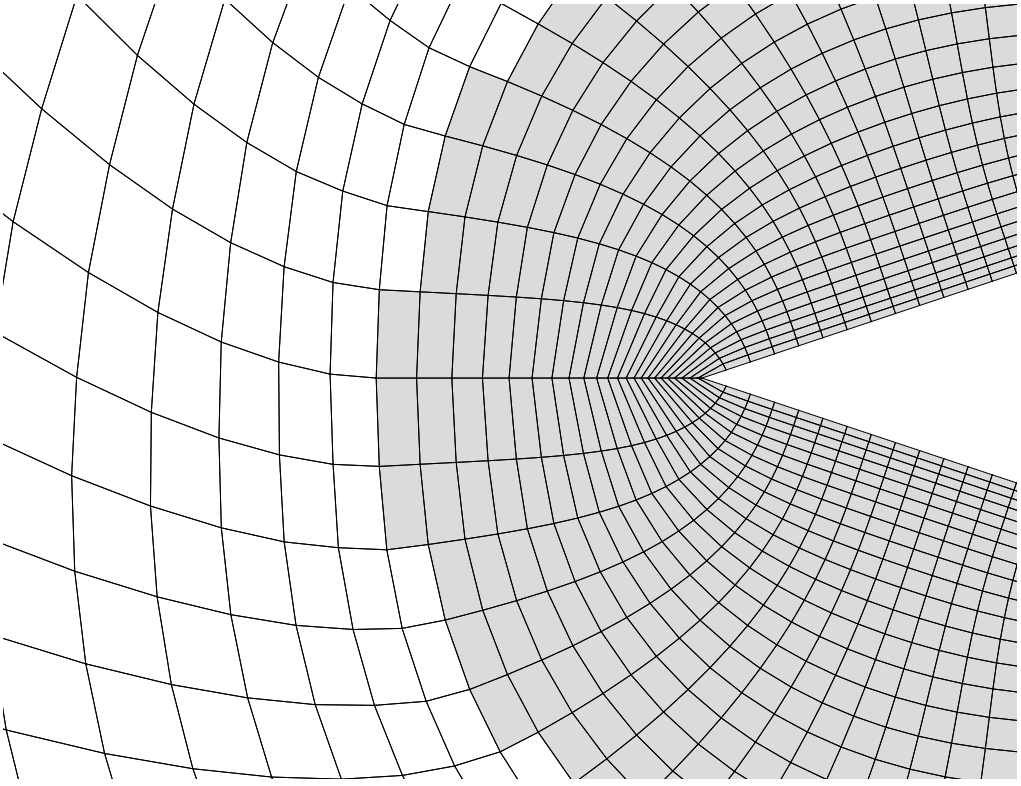


(b) Zoom sur une des extrémités

FIG. 4.40: Le maillage déformé autour du losange après une rotation de 90°



(a) Maillage déformé après rotation de 90° du losange



(b) Zoom sur une des extrémités

Fig. 4.41: Couplage de l'algorithme de remaillage par mouvement en bloc (actif dans la zone grisée) avec celui pseudo-structure afin de traiter la rotation du losange

On voit que grâce au mouvement en bloc, la nouvelle frontière du remaillage pseudo-solide ($Coef_{Pond} = 0.8$) n'a plus d'angle aigu et que le module pseudo-structure peut alors fonctionner correctement au niveau de celle-ci. L'emploi du mouvement en bloc et de la pondération permet donc d'adoucir les coins.

4.4 Conclusion

Au vu des grands déplacements qu'allaient subir les corps élancés déformables, un module de remaillage a été développé en se basant sur une approximation pseudo-structure du domaine fluide. Une fois le développement et l'intégration faits dans **ISIS**, ce sous-programme a été validé, puis testé en bidimensionnel, comme en tridimensionnel. L'utilisation d'une discrétisation évoluée avec un module de cisaillement variable permet d'obtenir de très bons résultats, même sur des corps de forme complexe comme un requin. Toutefois en présence de corps avec des coins en très grand déplacement, le remaillage pseudo-structure peut être mis en défaut. Une solution élégante consistant à coupler plusieurs méthodes de remaillage donne des résultats très satisfaisants.

Ce chapitre 4 a ouvert le débat du couplage fluide-structure en traitant une partie du couplage en espace : le processus de remaillage. Le chapitre 5 répond aux interrogations concernant les autres problèmes de couplage en IFS :

- ➢ La transmission des efforts fluides à la poutre et des déplacements solides au fluide pour le couplage en espace.
- ➢ L'organisation séquentielle des appels aux solveurs pour le couplage en temps.

Chapitre 5

Interface fluide-structure et algorithme de couplage temporel

Le processus de remaillage est un élément nécessaire pour traiter le couplage en interaction fluide-structure. Ce couplage en espace permet le calcul sur la configuration déformée mise à jour. Cependant, nous allons voir que le remaillage n'est pas la seule difficulté à surmonter lorsqu'on couple un solveur fluide et un solveur structure.

Il existe en effet un autre point important dans le couplage en espace : **la transmission des informations échangées par les solveurs à l'interface fluide-structure**. Les maillages solides et les maillages structures sont en général **"non-conformes"** : les nœuds du maillage fluide ne coïncident pas avec ceux du maillage structure à l'interface. Il faut alors prévoir un processus d'interpolation.

Enfin, nous terminerons par **le couplage en temps des solveurs**. Comme nous l'avons entrevu dans le chapitre 1, différents algorithmes de couplage en temps existent. Celui utilisé sera présenté en détail, ainsi que les problèmes rencontrés lors des cas-tests.

5.1 Couplage en espace : transmission des informations à l'interface fluide-structure

En interaction fluide-structure, un solveur fluide et un solveur solide sont nécessaires. Pour que ces deux programmes puissent résoudre les équations discrétisées du fluide ou de la structure, un maillage est obligatoire. Les maillages fluide et structure peuvent être conformes ou non-conformes (cf. fig. 5.1). Construire un maillage fluide-structure conforme implique par définition que les nœuds à l'interface fluide-structure correspondent. Il est alors trivial de transmettre les informations (comme les efforts fluides) entre les deux maillages. Dans le cas non-conforme, *une interpolation des données à communiquer est nécessaire.*

Créer un maillage conforme est possible sur des cas bidimensionnels simples, lorsque l'on a affaire à deux solveurs 2D ou deux solveurs 3D. Dès que le nombre de points augmente et que la géométrie se complique, cela devient très complexe. Dans notre cas, le dilemme ne se pose pas : en effet, notre maillage structure est monodimensionnel et le maillage fluide est bidimensionnel ou tridimensionnel. L'utilisation de maillages conformes

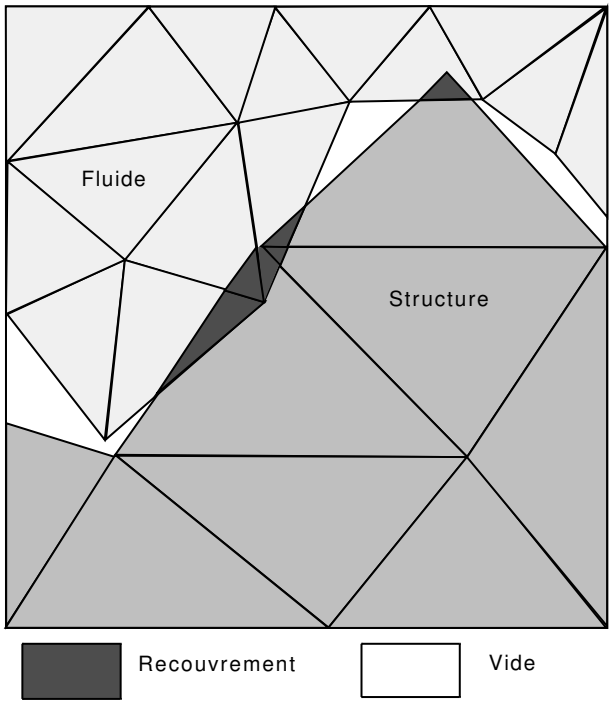


FIG. 5.1: Exemple de maillages non-conformes

n'est donc pas envisageable. Il est par conséquent nécessaire d'avoir recours à une interpolation des données à l'interface fluide-structure.

5.1.1 État de l'art

Dans la littérature, on trouve divers procédés pour transmettre ces informations, comme la méthode d'interpolation **du plus proche voisin**, les méthodes **par projection** et enfin les méthodes **d'interpolations basées sur les splines**. Ces techniques sont toutes très différentes. On peut les comparer suivant plusieurs critères dont les plus importants sont :

- ➢ la **conservation globale de l'énergie** sur l'interface fluide-structure
- ➢ la **conservation globale de la charge** au passage de l'interface
- ➢ la **précision**
- ➢ la **conservation de l'ordre des solveurs couplés**
- ➢ le **rendement** (le rapport entre la précision et le coût numérique)

5.1.1.1 Les différentes méthodes d'interpolation entre des maillages non-conformes

La méthode d'interpolation la plus simple et la plus rapide est sans nul doute celle **du plus proche voisin**. Elle est ainsi appelée, car l'information à transmettre d'un point d'un des maillages est donnée directement au point le plus proche appartenant à l'autre maillage [98].

Les **techniques par projection** sont, quant à elles, un peu plus compliquées à mettre en place. Cependant le principe reste simple : pour obtenir l'information d'un

des maillages, on la projette orthogonalement sur l'autre. L'information peut ainsi être utilisée sur le maillage non originel. Ensuite la technique d'intégration de Gauss peut être utilisée pour terminer le calcul [38]. De même, on peut aussi projeter une face (ou un élément) d'un des maillages sur l'autre et en calculer la surface d'intersection. Cette dernière détermine alors de quelle manière les informations à transmettre vont intervenir dans le maillage cible.

Enfin la troisième et dernière voie pour transmettre des données sur des maillages non-conformes est la **méthode d'interpolation par splines**. Le choix de ces dernières est primordial et influe directement sur les performances de la méthode (cf. [11], [91] et [92].

5.1.1.2 Comparaison et performances de ces méthodes

Ces méthodes de transfert d'information influencent l'ordre de précision et le rendement du code fluide-structure. Il est donc intéressant de les comparer et de choisir celle qui correspond le mieux à nos besoins. Une étude sérieuse et complète a déjà été faite [28]. Cette partie se base sur ses résultats.

Par sa nature, la méthode du plus proche voisin ne donne de bons résultats que sur des maillages conformes. C'est une méthode simple qui peut être appliquée dans des cas simples. Dans le cas de maillages non-conformes, elle doit être évitée.

La méthode par projection pose elle aussi un problème sur certains maillages non-conformes. En effet en présence d'une surface trop concave ou trop convexe une partie des cellules cibles peut ne pas être accessible par la projection orthogonale. Une non-conservation globale des forces en résulte. Une procédure de recherche et de projection doit aussi être implantée pour que cet algorithme puisse fonctionner.

Les interpolations par splines n'ont pas de défaut, excepté leur complexité.

Dans les tests réalisés par A. de Boer dans [28], seule la méthode du plus proche voisin est d'ordre 1. Les méthodes d'interpolation par projection et par splines sont d'ordre égal ou supérieur à 2. Sur le plan du rendement les méthodes par splines sont très performantes par rapport aux méthodes par projection qui consomment plus de CPU au moment de l'utilisation de l'algorithme de recherche. A. de Boer conclut de ses tests numériques que les méthodes par splines sont à l'heure actuelle un très bon choix.

5.1.2 Choix de l'interpolation

Dans [28], A. de Boer prend le cas le plus général de l'interaction fluide-structure : deux maillages 3D non-structurés. Le contexte est ici différent. Notre étude porte sur les structures élancées pouvant être modélisées par des poutres. Le maillage structure est réduit à un maillage monodimensionnel. Le problème est donc considérablement simplifié. Cependant, les nœuds du maillage structure ne correspondent pas à ceux du maillage fluide, car l'intersection du maillage structure (la ligne neutre de la poutre) avec le maillage fluide est vide (sauf parfois aux extrémités). Nos maillages sont donc non-conformes. De ce fait lors du choix de l'interpolation, *l'algorithme du plus proche voisin a été tout de suite écarté. Les méthodes par splines étant complexes, les méthodes par projection ont été préférées.*

Voyons en détail la méthode de projection développée au sein du code interaction fluide-structure. L'hypothèse de Kirchhoff assure que *les sections de la poutre restent*

orthogonales à la ligne neutre en toutes circonstances. De plus *les deux maillages ne glissent pas l'un par rapport à l'autre*. Ainsi, l'algorithme de projection est employé une seule fois en début de calcul. Il va générer les connectivités entre les deux maillages. Celles-ci ne changeront pas au cours des déformations.

Pour le transfert des informations, il faut connaître dans quel tronçon de poutre se situe la projection orthogonale des nœuds du maillage fluide situés à l'interface. Ceci requiert un algorithme de recherche. Le calcul des connectivités n'étant fait qu'une fois on peut se permettre de choisir un algorithme de recherche basique. On utilise la **dichotomie**.

Une fois connu le positionnement des nœuds du maillage fluide par rapport à la structure, on peut s'intéresser au problème de transmission des efforts fluides à la structure. Dans **ISIS**, les variables d'efforts fluides sont situées au centre des cellules. Ces efforts doivent être répartis correctement sur les tronçons de poutre. Les maillages n'étant pas conformes, il est *nécessaire de découper les faces fluides en faces virtuelles plus petites*. Celles-ci porteront en leur centre un effort fluide pondéré par le rapport entre leur aire et l'aire de la face entière contenant l'effort originel. Ainsi, la somme des efforts pondérés des faces virtuelles est égale à l'effort originel de la face fluide mère. Évidemment le découpage en faces virtuelles ne se fait pas au hasard. On trace les plans orthogonaux à la ligne neutre passant par le centre des tronçons de poutre. Si les plans orthogonaux traversent une face fluide, celle-ci est coupée suivant ce plan (cf. fig. 5.2). Ensuite les efforts pondérés sont transmis aux nœuds du maillage structure correspondant (cf. fig. 5.3). On conserve ainsi la charge au passage de l'interface. Notons aussi que vu les hypothèses poutre le problème dû aux maillages trop concaves ou convexes n'est pas à considérer.

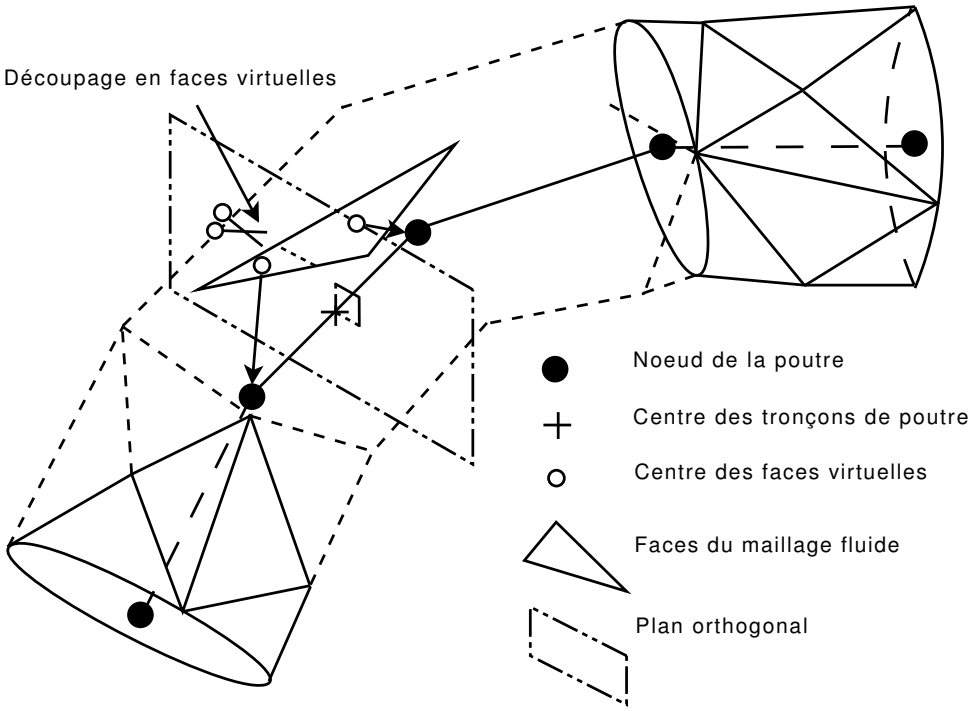


FIG. 5.2: Couplage à l'interface fluide-structure : découpage en faces virtuelles

La transmission des efforts étant assurée, intéressons-nous maintenant au passage des déplacements des nœuds de la structure, calculés par le solveur structure, aux nœuds du

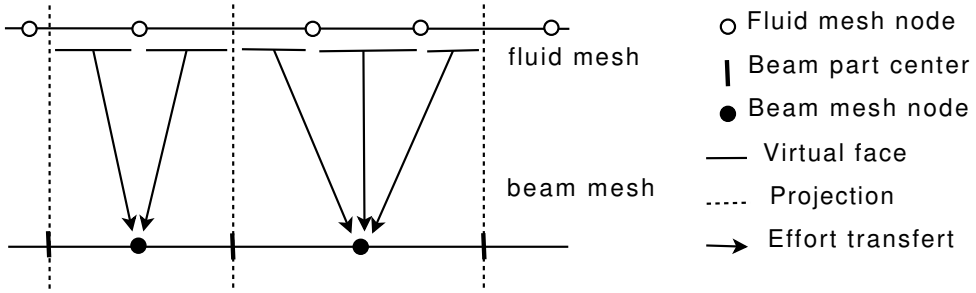


FIG. 5.3: Couplage à l'interface fluide-structure : transfert des efforts

maillage fluide. Nous avons déjà les connectivités entre les nœuds des deux maillages.

Le calcul de l'emplacement des nouveaux nœuds frontières du maillage fluide en 2D est simple (cf. fig. 5.4) : pour chaque nœud du maillage fluide de référence, on calcule son projeté sur le maillage poutre et la distance le séparant du maillage poutre. Vu les hypothèses faites dans le solveur structure (hypothèse de planéité et de rigidité des sections, orthogonalité des sections de poutre par rapport à la ligne neutre), après déformation de la ligne neutre du maillage poutre, un nœud du maillage fluide doit être sur le même plan orthogonal passant par le projeté de ce point calculé à partir du maillage de référence. De plus la distance entre son projeté et lui ne doit pas être différente.

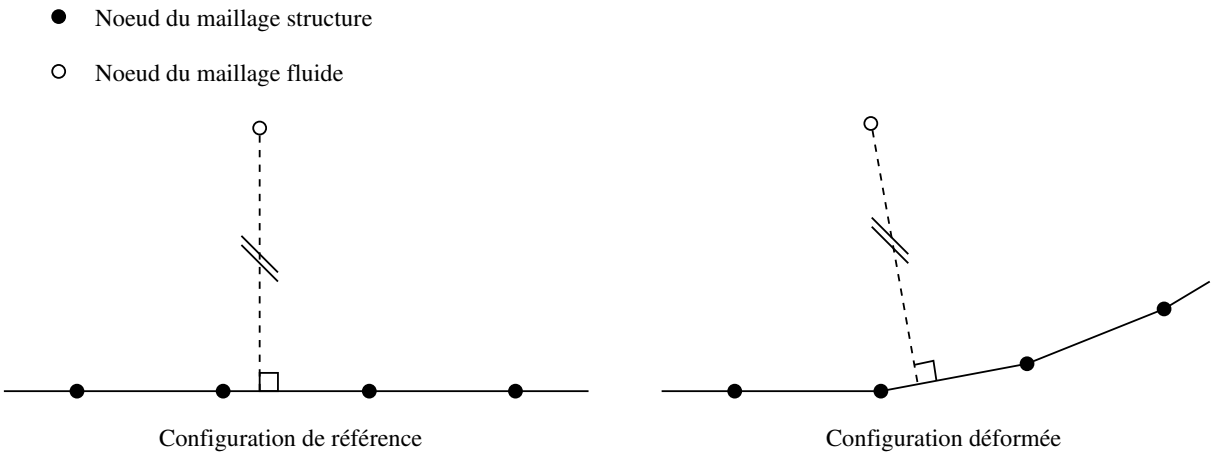


FIG. 5.4: Déplacements des nœuds du maillage fluide à partir de la position de la poutre déformée

En 3D, le calcul des déplacements des nœuds du maillage fluide est similaire, mais plus complexe (cf. fig. 5.5). L'hypothèse de Cosserat impose que la section à chaque nœud de poutre soit rigide. Chaque nœud de l'interface fluide peut alors être repéré dans le plan orthogonal passant par ce nœud, en se rappelant que la distance entre ce point et son projeté sur la ligne neutre de la poutre est fixe. L'orientation de ce plan orthogonal est bien entendu à calculer. En 2D, le cas est simple à traiter : la rotation a pour vecteur fixe $\overrightarrow{z}$ et il est simplement paramétré par un angle. Par contre en tridimensionnel, le vecteur laissé fixe par la rotation n'est pas prédéterminé. Pour résoudre correctement ce problème, nous faisons appel aux **quaternions** (cf. annexes B et C), qui évitent l'apparition de singularités. En effet, si le paramétrage est effectué via 3 rotations, notées (ψ, θ, ϕ), il existe toujours des configurations singulières pour lesquelles le triplet ci-dessus n'est pas

unique. Muni des quaternions, le calcul de l'orientation du plan est possible. Cependant, il est nécessaire de connaître le quaternion au point projeté, ce qui implique une interpolation de quaternions.

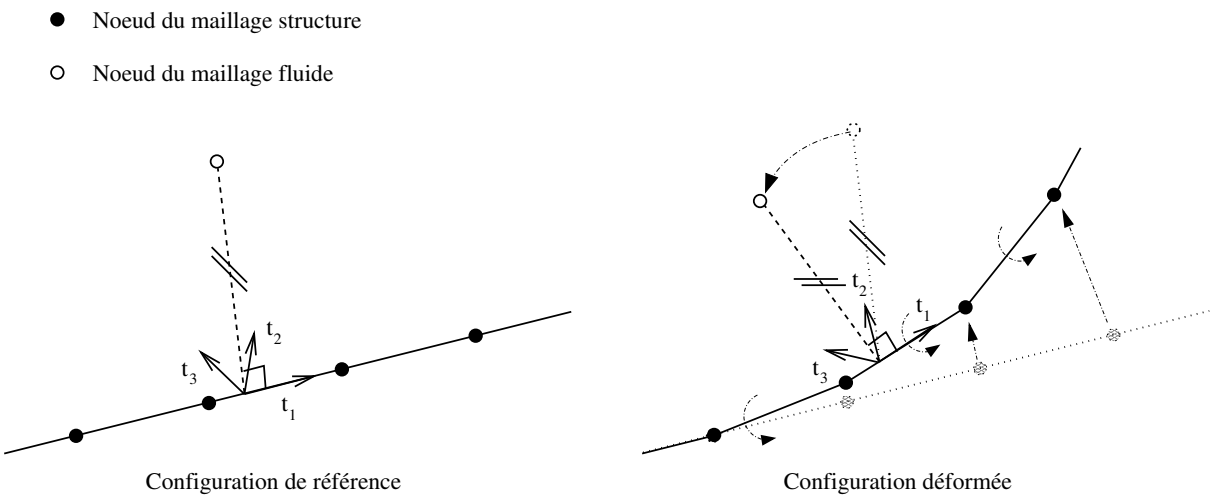


FIG. 5.5: Déplacements des nœuds du maillage fluide à partir de la position de la poutre déformée en 3D

Si le principe de déplacement des nœuds du maillage fluide est relativement simple, le calcul du quaternion au point projeté du maillage structure l'est beaucoup moins. Au cours du développement, deux approches pour le traitement de l'interpolation de quaternions ont été traitées. Les deux sont présentées ici, bien que la première, plus complexe, ait été mise de côté.

Voici la description de la première méthode : le solveur structure nous fournit les rotations des tronçons de poutre. On peut donc calculer directement les quaternions $\overrightarrow{q}$ relatifs à ces tronçons, mais ceux-ci sont situés aux nœuds du maillage structure. Il faut donc estimer la valeur du quaternion au point projeté. Pour cela la courbure $\overrightarrow{c}$ en chaque nœud du maillage poutre est nécessaire. En chaque nœud structure, on peut écrire (cf. annexe C) :

$$\frac{1}{2}\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & q_3 & -q_2 \\ -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \end{bmatrix} \overrightarrow{c} = \frac{\partial \overrightarrow{q}}{\partial s} \qquad \text{où} \qquad \overrightarrow{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Les dérivées spatiales des quaternions sont évaluées au premier ordre. Le système est résolu grâce au module DGELS de la librairie LAPACK [6] : ce programme résout en effet les systèmes de type surdéterminés (comme ici) ou sous-déterminés. Une fois la courbure obtenue en chaque nœud de la structure, nous effectuons une moyenne pour l'avoir au centre des tronçons. Grâce à cette dernière, que nous nommerons $\overrightarrow{k}$ (de composantes k_1, k_2, k_3), et aux données précédemment calculées, le quaternion au point projeté du maillage structure, $\overrightarrow{q_{s_{proj}}}$, peut être évalué au point d'abscisse curviligne s_{proj}, correspondant au projeté (cf. annexe C) :

$$\overrightarrow{q_{s_{proj}}} = exp(\mathbb{K}s_{proj})\ \overrightarrow{q} \tag{5.1}$$

avec

$$\mathbb{K} = \frac{1}{2}\begin{bmatrix} 0 & -k_1 & -k_2 & -k_3 \\ k_1 & 0 & k_3 & -k_2 \\ k_2 & -k_3 & 0 & k_1 \\ k_3 & k_2 & -k_1 & 0 \end{bmatrix} \tag{5.2}$$

En s'appuyant sur l'annexe C, on calcule l'exponentielle de la matrice $\mathbb{K}$ et on obtient : En posant $\theta_k = \frac{s_{proj}}{2}\sqrt{k_1^2 + k_2^2 + k_3^2}$ et $\kappa = \sqrt{k_1^2 + k_2^2 + k_3^2}$, on a :

$$\exp(\mathbb{K}s_{proj}) = \cos(\theta_k)\,\mathbb{I} + \sin(\theta_k)\,\frac{2\,\mathbb{K}}{\kappa} \tag{5.3}$$

Ce qui donne :

$$\begin{aligned}
q_{s_{proj_0}} &= \cos(\theta_k)q_0 + \sin(\theta_k)\left[-\tfrac{k_1}{\kappa}q_1 - \tfrac{k_2}{\kappa}q_2 - \tfrac{k_3}{\kappa}q_3\right] \\
q_{s_{proj_1}} &= \cos(\theta_k)q_1 + \sin(\theta_k)\left[+\tfrac{k_1}{\kappa}q_0 + \tfrac{k_3}{\kappa}q_2 - \tfrac{k_2}{\kappa}q_3\right] \\
q_{s_{proj_2}} &= \cos(\theta_k)q_2 + \sin(\theta_k)\left[+\tfrac{k_2}{\kappa}q_0 - \tfrac{k_3}{\kappa}q_1 + \tfrac{k_1}{\kappa}q_3\right] \\
q_{s_{proj_3}} &= \cos(\theta_k)q_3 + \sin(\theta_k)\left[+\tfrac{k_3}{\kappa}q_0 + \tfrac{k_2}{\kappa}q_1 - \tfrac{k_1}{\kappa}q_2\right]
\end{aligned}$$

$q_{s_{proj}}$ est de norme unitaire.

Voyons maintenant le principe de la seconde méthode : elle se base sur une interpolation linéaire sphérique (connu sous le nom **"slerp"** [34]). Soit 2 quaternions p et q avec l'angle aigu θ entre les deux ($\cos(\theta) = p \cdot q$), le quaternion interpolé au point d'abscisse curviligne s_{proj} (comprise en 0 et 1) s'écrit :

$$\boxed{q_{s_{proj}} = \frac{sin((1 - s_{proj})\theta)p + sin(s_{proj}\theta)q}{sin(\theta)}} \tag{5.4}$$

Notons tout de suite que pour θ très petit cette formule ne s'applique plus. Mais elle dégénère en :

$$\boxed{q_{s_{proj}} = \frac{(1 - s_{proj})p + s_{proj}q}{\|(1 - s_{proj})p + s_{proj}q\|}} \tag{5.5}$$

On vérifie bien que $q_{s_{proj}}$ est de norme 1.

Pour mieux comprendre cette formule, raisonnons d'un point de vue géométrique : si p et q sont vus comme des points placés sur le cercle unité, l'équation précédente est une

paramétrisation du plus court arc entre eux deux.

Une fois obtenue la rotation du tronçon de poutre au point projeté, $R_{ref \to def}$, le calcul de la nouvelle position du nœud du maillage fluide est effectué (cf. fig. 5.6) :

$$\boxed{\overrightarrow{X_{def}} = Mat(R_{ref \to def})\, \overrightarrow{X_{ref}}} \tag{5.6}$$

soit en terme de quaternions (cf. annexe B) :

$$\boxed{\mathbf{X}_{def} = \mathbf{Q}_{R_{ref \to def}}\, \mathbf{X}_{ref}\, \overline{\mathbf{Q}_{R_{ref \to def}}}} \tag{5.7}$$

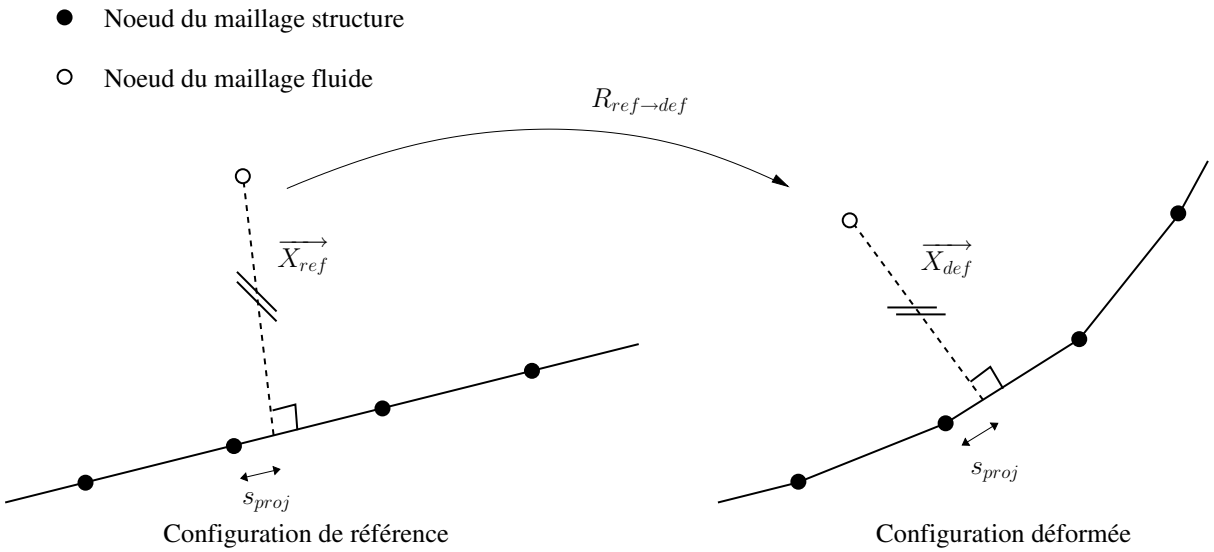


FIG. 5.6: Détails sur les déplacements des nœuds du maillage fluide à partir de la position de la poutre déformée en 3D

5.2 Couplage en temps

Précédemment nous avons vu le développement des codes de calcul fluide et de calcul structure et leurs moyens de communiquer au travers de l'interface fluide-structure. Un dernier point reste à éclaircir : **l'échange des informations dans le temps**. Dans les cas-tests IFS expérimentaux, ces échanges interviennent à tout moment et en continu. C'est évidemment impossible au sein d'un code de calcul. Il faut par conséquent organiser l'échange d'informations "séquentiellement" pour obtenir un résultat le plus proche possible de la réalité. Ce processus est appelé **couplage en temps**.

Le chapitre 1 décrit les différents algorithmes de couplage en temps. Pour rappel, lorsque le code IFS n'est pas monolithique, on trouve :

- ➢ les **algorithmes explicites synchrones et décalés**
- ➢ les **algorithmes parallèles**
- ➢ les **algorithmes itérés (ou implicites)**

L'intégration de l'IFS au sein d'ISIS se fait, soit *avec un algorithme explicite simple*, soit *avec un algorithme itéré*.

L'algorithme explicite est juste utile dans le cadre de la validation du solveur poutre. En effet dans cet algorithme les solveurs sont appelés les uns après les autres en avançant en temps après le passage dans le solveur structure. Pour les tests de vérification structure, le passage dans le solveur fluide est annulé. ISIS se transforme alors en un solveur poutre. C'est ainsi que les cas-tests de validation présentés dans le chapitre 3 sur le solveur poutre ont été réalisés.

L'algorithme explicite ne répond pas à nos attentes sur le plan de la précision et de la stabilité dans le cadre de cas-tests d'application d'interactions fluide-structure. Pour augmenter cette précision, la seule méthode que l'on puisse utiliser est celle proposée par les algorithmes implicites. Le schéma 5.7 présente l'algorithme utilisé dans le code de calcul.

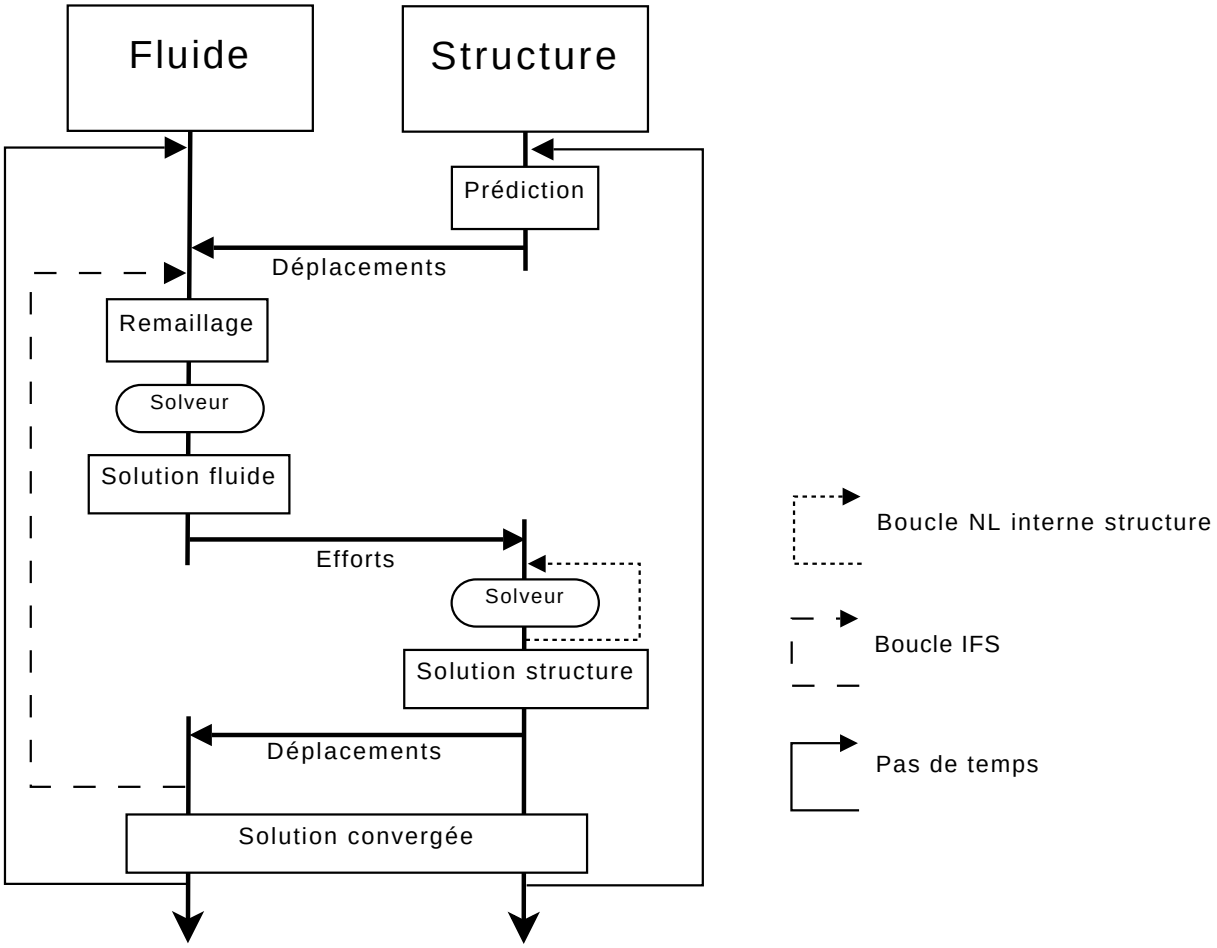


Fig. 5.7: Algorithme de couplage en temps des solveurs fluide et structure

À une configuration et à un instant donnés et fixés le solveur fluide, ISIS, détermine l'écoulement correspondant. Les efforts fluides agissant sur la structure sont transmis au corps via le couplage en espace précédemment expliqué. Le solveur poutre calcule alors les déplacements résultants de cette charge. Il les transmet au solveur fluide et le processus de remaillage entre en jeu. Une fois que le nouveau maillage déformé est prêt, ISIS

calcule de nouveau l'écoulement correspondant à cette configuration. Les efforts fluides sont de nouveau transmis au solveur poutre et ainsi de suite. On appelle cette boucle une itération d'interaction fluide-structure. Après un certain nombre d'itérations de ce type, l'écoulement calculé par le solveur fluide ainsi que la dynamique de la poutre convergent. On considère alors que l'on a atteint la solution finale pour cet instant. On passe ensuite à l'itération temporelle suivante. Et ainsi de suite jusqu'à atteindre la fin de la simulation.

Pour éviter d'avoir un **décalage en temps à la première itération IFS**, une **prédiction de la dynamique de la structure est nécessaire**. Ainsi on ajoute une étape au sein de l'itération IFS : avant d'entrer dans le solveur fluide, la position de la poutre est prédite en utilisant les données stockées précédemment. Ainsi nous sommes cohérents avec la physique. Cette prédiction est d'ordre 1.

En ce qui concerne les vitesses à l'interface fluide-structure, la prise en compte est faite via l'évolution temporelle de la déformation du maillage de surface de la poutre. En effet, le code fluide recalcule chaque fois que nécessaire les flux de vitesse de déplacement aux faces de l'interface (ainsi qu'aux autres faces du maillage, cf. chapitre 9 de [57]). Le couplage est donc physiquement cohérent, même si les vitesses à l'interface fluide-structure diffèrent du fait de l'utilisation de deux schémas temporels distincts (schéma de Newmark pour la structure, schéma décentré d'ordre 2 pour le fluide).

5.3 Détails sur le couplage IFS

Les techniques de couplage IFS mises en œuvre dans cette thèse ont toutes été expliquées. Il reste néanmoins quelques détails à donner sur l'imbrication de celles-ci entre elles.

Le couplage temporel s'effectue via un algorithme itératif. Pour passer au pas de temps suivant, plusieurs itérations IFS doivent être réalisées pour avoir la convergence de l'écoulement. Chacune nécessite de faire appel au couplage spatial. Pour la transmission des informations à l'interface, les connectivités entre les deux maillages n'étant pas modifiées, le processus est rapide et ne consomme pas beaucoup de temps CPU. Mais il en va tout autrement pour le remaillage : l'utilisation du module de remaillage pseudo-structure, à chaque itération IFS, apporte entière satisfaction quant au résultat, mais *le temps CPU dépensé est très important*. L'utilisation de l'algorithme de résolution rapide permet un gain substantiel ; ce n'est cependant pas encore acceptable. C'est pourquoi, encore une fois, *plusieurs techniques de remaillage* vont être mixées : à la **première itération IFS**, le maillage est mis à jour via le module pseudo-structure. En effet, c'est à cette étape que le maillage se déforme le plus, car on change de pas de temps. Pour les itérations IFS suivantes (qui sont à t fixé), le module de remaillage par pondération est appelé, car seul un ajustement des positions des nœuds est nécessaire. Le gain en temps CPU est immédiat.

5.4 Conclusion

Avec ce chapitre les éléments d'un code IFS sont au complet. Les deux solveurs sont en effet connectés entre eux par un couplage spatial et temporel. Le couplage spatial se

fait via une technique par projection et à l'aide des quaternions. Le couplage en temps s'appuie, quant à lui, sur un algorithme itéré. On est donc en possession d'un **code IFS non-monolithique**, mais **couplé fortement**.

L'étape suivante est bien sûr la validation et l'utilisation de ce code IFS sur des corps élancés en grand déplacement.

Chapitre 6

Applications d'IFS réalisées

Le but de cette thèse est de *reproduire numériquement des comportements d'interaction fluide-structure pour des corps élancés*. Quelques cas-tests IFS ont été réalisés à l'aide du code développé. Ils se diviseront en deux grandes catégories : les applications bidimensionnelles et celles tridimensionnelles.

Les applications 2D sont souvent plus aisées à traiter et surtout offrent une visualisation plus facile. Celles-ci ont donc été étudiées en premier. Elles ont permis d'optimiser et de valider le code sur des configurations faciles à mettre en œuvre. Mais ces cas 2D ont aussi fait apparaître les faiblesses du couplage interaction fluide-structure ; en particulier les problèmes liés au schéma de discrétisation en temps de Newmark. Nous verrons tout d'abord un cas-test 2D avec un écoulement stationnaire : une plaque fixée à une extrémité placée dans une conduite. Puis, pour valider le couplage IFS en dynamique, une application plus complexe sera réalisée : un filament retenu derrière un bloc carré dans un écoulement.

Les applications tridimensionnelles présentées dans ce chapitre ont pour but de montrer les capacités du code IFS en 3D. Un long câble est plongé dans un environnement multifluide. Dans un premier temps, il est fixé en son extrémité supérieure et subit l'action des fluides en mouvement. Par la suite, ce même câble est tracté dans un environnement multifluide au repos. Enfin, un code numérique doit être validé par comparaison avec des données expérimentales. Le cas-test réalisé par J.R. Chaplin et son équipe en 2005 [22] a été choisi.

6.1 Applications 2D

6.1.1 Plaque dans une conduite

On considère une plaque flexible de longueur $L = 1\,m$, d'épaisseur $d = 0.01\,m$, encastrée en son extrémité inférieure et placée verticalement dans un écoulement horizontal (cf. fig. 6.1).

Le maillage fluide utilisé pour le calcul numérique est 2D non-structuré (cf. fig. 6.2). Il contient environ 30000 cellules et permet d'approcher la convergence en maillage. Les conditions aux limites choisies sont les suivantes :

- ➢ Vitesse imposée en entrée et sur la frontière supérieure
- ➢ Pression imposée en sortie
- ➢ Condition d'adhérence sur la paroi inférieure et sur la plaque

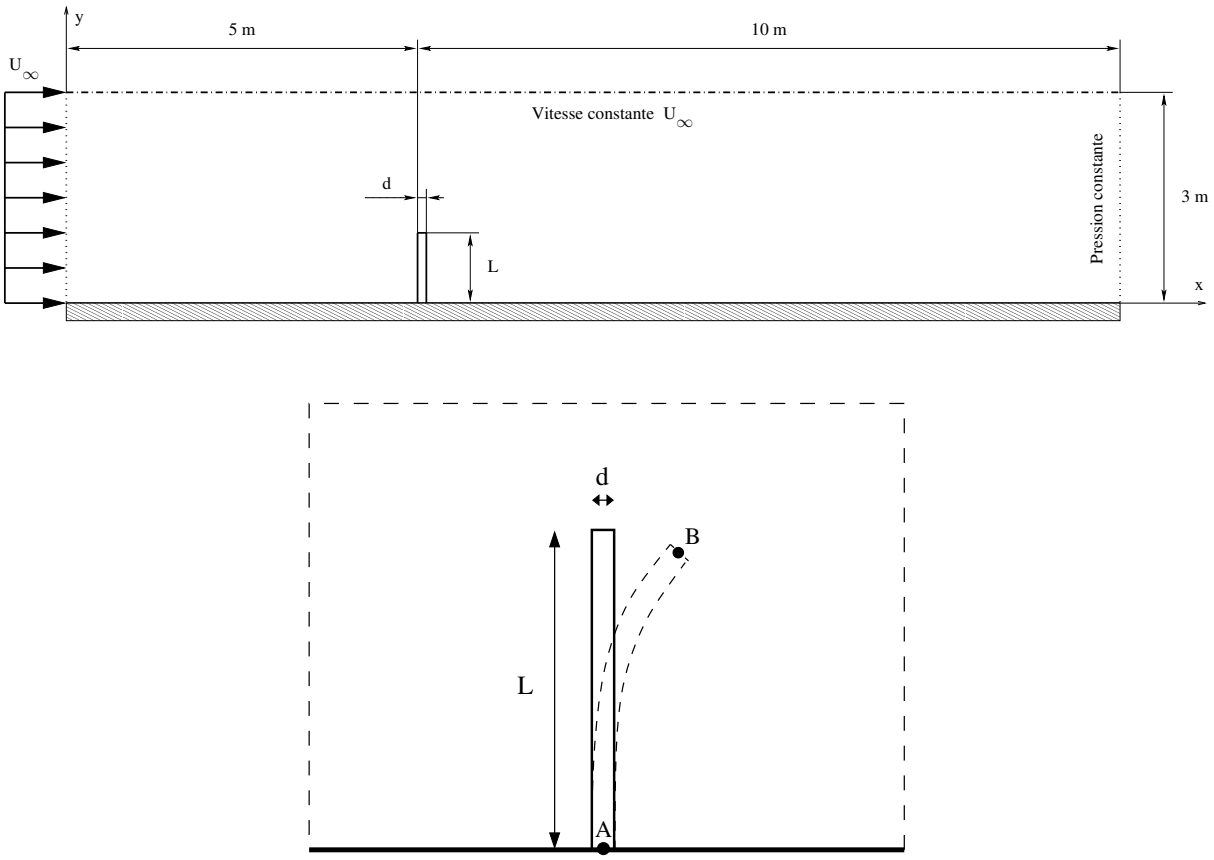


FIG. 6.1: Plaque encastrée dans une conduite soumise à un écoulement transverse

En premier lieu, on choisit de considérer la plaque comme indéformable pour pouvoir comparer l'écoulement stationnaire avec celui généré avec une plaque déformable. Les conditions de cette configuration sont détaillées dans le tableau 6.1. Le nombre de Reynolds étant égal à 50, les calculs s'effectuent en régime laminaire.

Longueur	$L = 1\,m$
Épaisseur	$d = 0.01\,m$
Masse volumique du fluide	$\rho_f = 1\,kg.m^{-3}$
Viscosité dynamique	$\mu_f = 0.2\,Pa.s$
Vitesse d'entrée constante	$U = 10\,m.s^{-1}$

TAB. 6.1: Conditions du cas-test pour une plaque indéformable (unités SI)

La configuration nous permet d'effectuer un calcul stationnaire. En effet un calcul stationnaire et un calcul instationnaire donnent les mêmes résultats après convergence

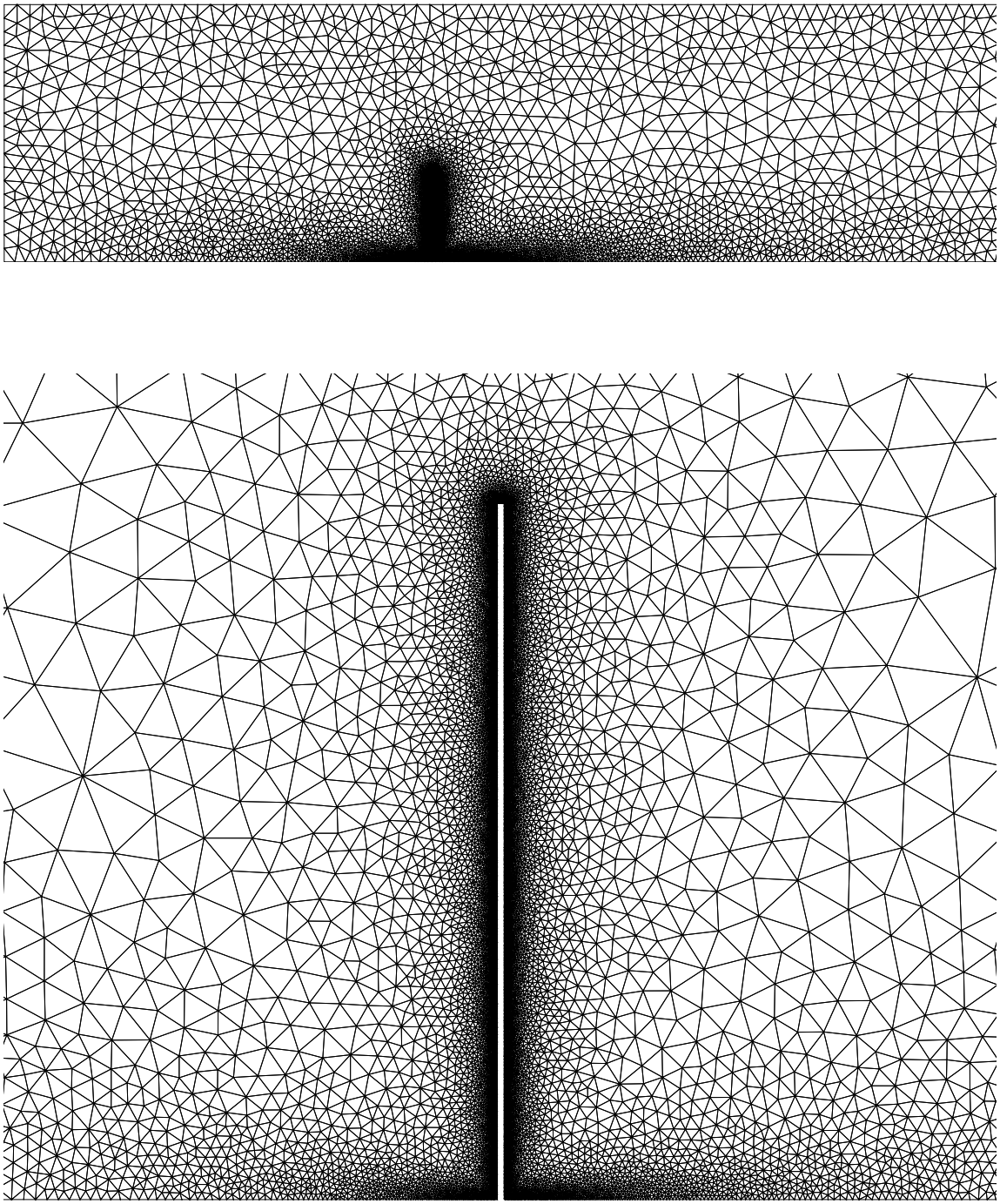


FIG. 6.2: Maillage fluide de la conduite et autour de la poutre

(cf. fig. 6.3). Les isovaleurs du coefficient $C_p = \dfrac{P - P_{ref}}{\frac{1}{2}\rho U_\infty^2}$ sont tracées dans le cas stationnaire en rouge et dans le cas instationnaire en vert. On voit que les iso-C_p rouges sont complètement masquées par les vertes. Les deux écoulements sont identiques. L'avantage est que le calcul stationnaire est plus rapide que le calcul instationnaire. Notons en bas à l'entrée et à la sortie que les iso-C_p sont sensiblement modifiées par la présence des bornes du domaine.

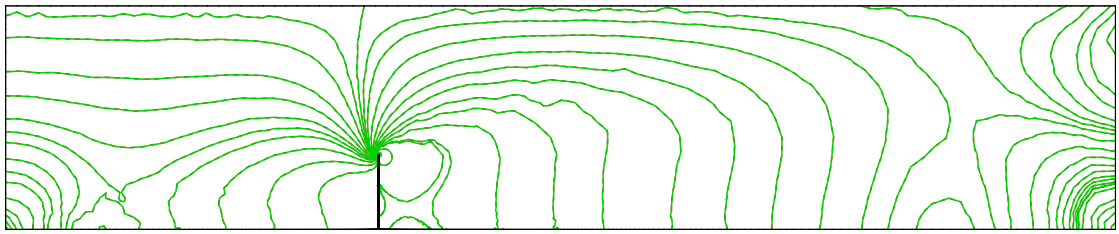


FIG. 6.3: Comparaison du coefficient C_p pour le calcul stationnaire (isolignes rouges) et instationnaire (isolignes vertes)

Une grande recirculation du fluide se crée en aval de la plaque (cf. fig. 6.4). Son centre se situe environ à $X = 4.2\, L_{poutre}$ et à $Y = 1.04\, L_{poutre}$. Sa taille fait $7\, L_{poutre}$ horizontalement et $1.7\, L_{poutre}$ verticalement. Une recirculation apparaît aussi devant l'obstacle. Elle est de taille beaucoup plus réduite.

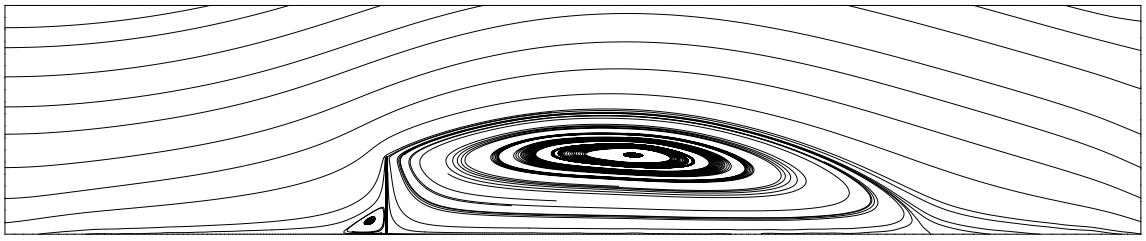


FIG. 6.4: Lignes de courant de l'écoulement stationnaire

La plaque, assimilée à une poutre 2D, devient maintenant déformable. Ses caractéristiques structurelles et celles du cas-test en général sont données dans le tableau 6.2. Cette configuration provient de l'article [43]. Le matériau est un polyester (DIN 16946).

Deux calculs couplés d'interaction fluide-structure sont réalisés : l'un avec un écoulement stationnaire et une poutre résolue statiquement, l'autre calculé avec une poutre résolue dynamiquement dans un écoulement évalué à l'aide d'une approche instationnaire. Dans ce dernier cas, l'écoulement est tout d'abord initialisé avec une poutre indéformable pendant $18.3\, s$. L'écoulement à ce moment-là est le même que celui calculé en stationnaire. Puis, la poutre devient déformable.

Nous nous intéressons tout d'abord à l'évolution de l'extrémité libre de la poutre. Sur la figure 6.5, nous observons la position de cette extrémité au cours du temps. Le calcul

Longueur	$L = 1\,m$
Épaisseur	$d = 0.01\,m$
Module d'Young	$E = 3500\,\text{MPa}$
Masse volumique du solide	$\rho_s = 1200\,kg.m^{-3}$
Masse volumique du fluide	$\rho_f = 1\,kg.m^{-3}$
Viscosité dynamique	$\mu_f = 0.2\,Pa.s$
Vitesse d'entrée constante	$U = 10\,m.s^{-1}$

TAB. 6.2: Conditions du cas-test avec plaque déformable (unités SI)

instationnaire est arrêté à $t = 23.15\,s$, car il est estimé avoir suffisamment convergé. Le calcul stationnaire avec la poutre statique donne bien l'asymptote de la courbe d'évolution temporelle fournie par le calcul instationnaire avec la poutre dynamique. Le déplacement final du point B est d'environ $2.4\,cm$. Le calcul IFS peut donc être fait en écoulement stationnaire avec une structure résolue statiquement. Dans l'approche instationnaire, on peut noter que la première fréquence donnée par transformée de Fourier est égale à $2.71\,Hz$, ce qui est légèrement plus faible que la fréquence théorique $f_1 = 2.76\,Hz$. C'est l'action de l'eau sur la structure qui vient modifier l'oscillation de la poutre et qui la freine.

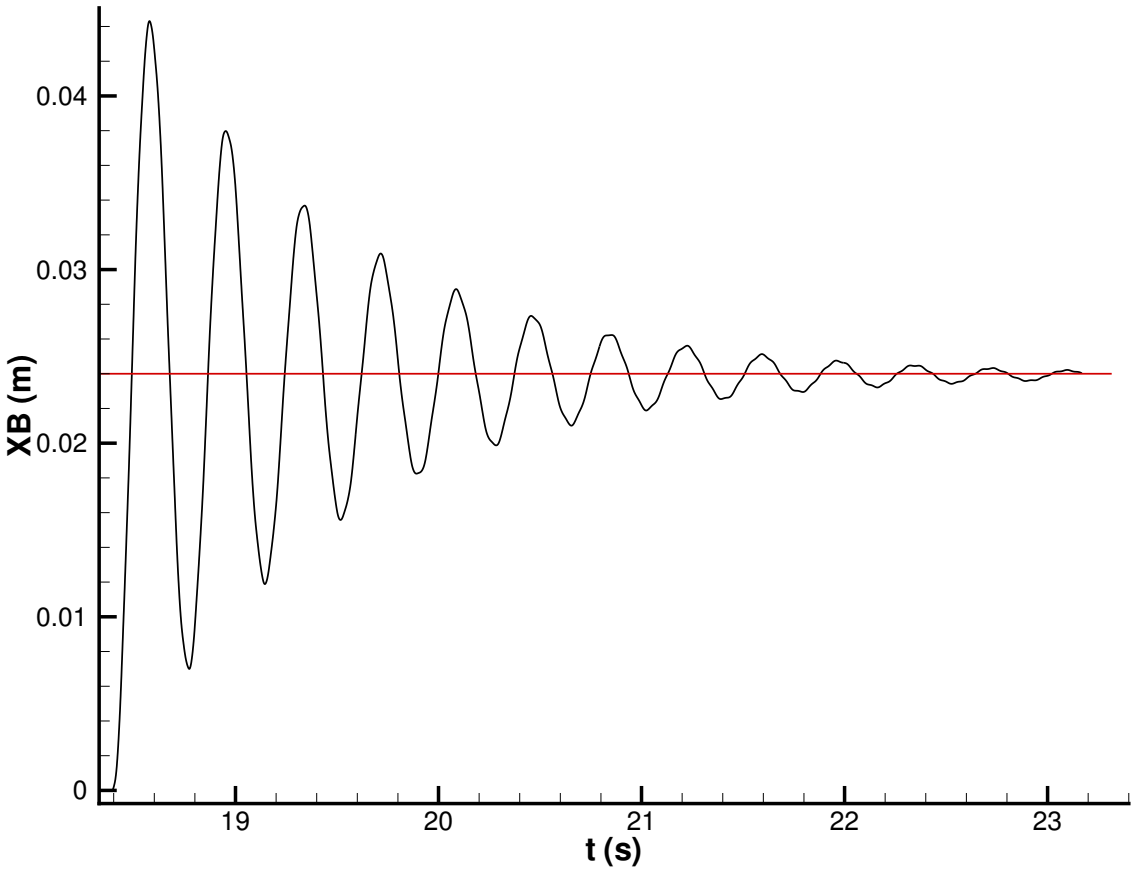


FIG. 6.5: Évolution temporelle de l'abscisse du point B : calcul statique et dynamique

La flèche de la poutre est à comparer avec les résultats obtenus par Glück dans [43] : ce dernier calcule en 3D sur une plaque de même hauteur et épaisseur que nous, mais avec une profondeur de $0.4\,m$. Il trouve un déplacement du point B de l'ordre de $3\,cm$. La différence entre les deux résultats peut s'expliquer par la profondeur finie de la plaque dans son cas.

Comparons maintenant l'écoulement obtenu par le couplage IFS avec celui sans IFS. En traçant les isovaleurs du coefficient C_p, on remarque tout de suite que l'écoulement n'est pas trop modifié lorsqu'on passe à un obstacle déformable (cf. fig. 6.6). Il en va de même si l'on compare les composantes de vitesse V_x et V_y ou encore la vorticité. Cela se voit aussi en traçant les lignes de courant (cf. fig. 6.7). L'écoulement est très semblable à celui de la figure 6.4. Sur le zoom de 6.7, on voit la poutre déformée et au repos.

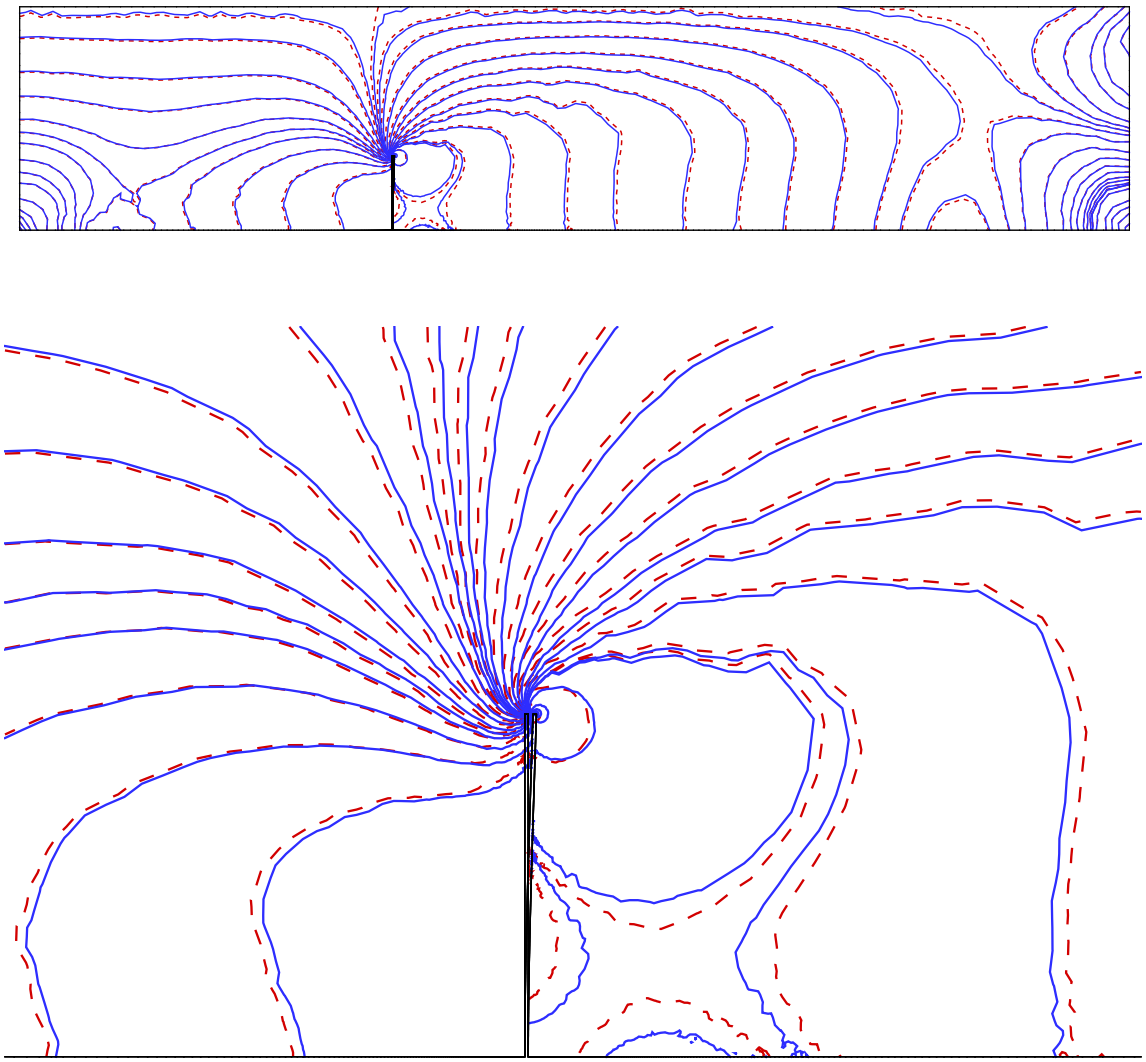


FIG. 6.6: Comparaison du coefficient C_p pour le calcul stationnaire sans IFS (isolignes rouges) et instationnaire avec IFS (isolignes bleues en pointillé)

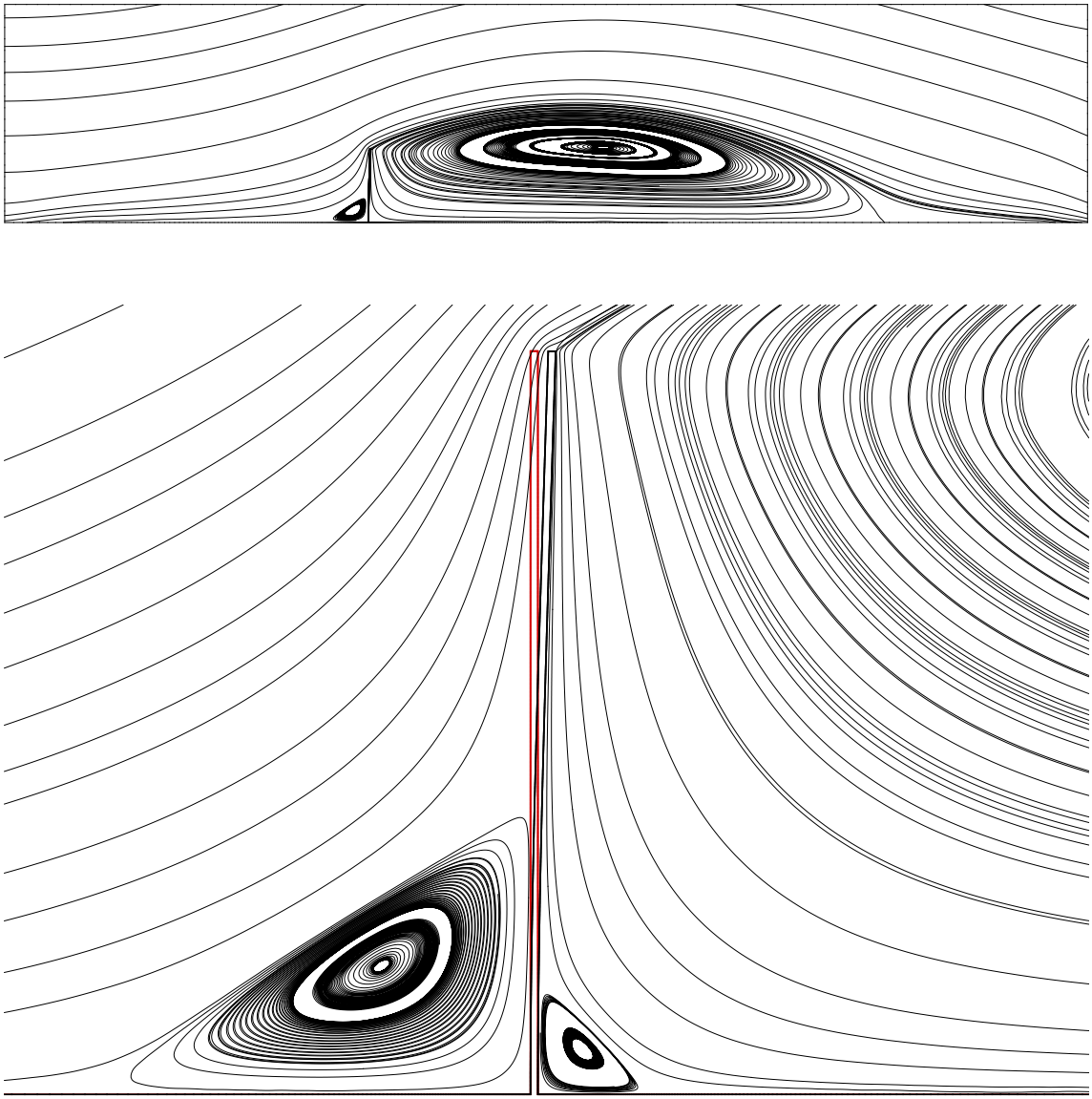


FIG. 6.7: Lignes de courant de l'écoulement calculé avec couplage IFS

Pour conclure sur ce cas-test, on peut dire que même si l'utilisation du couplage IFS ne semble pas vraiment justifiée vu la très faible modification de l'écoulement induite, cette application est très utile pour la validation du programme IFS. En effet, nous avons vérifié que *le code en mode structure dynamique et calcul fluide instationnaire convergeait vers la solution obtenue de manière statique pour la structure et stationnaire pour le fluide.*

6.1.2 Filament fixé derrière un bloc dans un écoulement

Après avoir testé le code IFS sur une application simple 2D avec un écoulement fluide stationnaire, un problème 2D mettant en œuvre un écoulement très instationnaire s'imposait.

Cette application a été proposée à l'origine par Wall et Ramm (cf. [104]) pour démontrer que leur formulation IFS pouvait prendre en compte des phénomènes d'interaction fluide-structure complexes. Plusieurs personnes ont repris ce cas-test par la suite : Steindorf en 2002 (cf. [96]), puis Dettmer en 2004 (cf. [33]). En 2004, Hübner reprend cette configuration en changeant la vitesse du fluide à l'entrée afin de tester son code IFS monolithique.

Le problème se résume à un filament très souple fixé à l'arrière d'un corps carré indéformable. L'ensemble est plongé dans un écoulement fluide incompressible (cf. fig. 6.8). La structure carrée engendre un phénomène très instationnaire : des vortex sont lâchés alternativement de part et d'autre des coins du carré. Ces tourbillons génèrent des forces oscillantes autour du filament. Celui-ci se déforme alors.

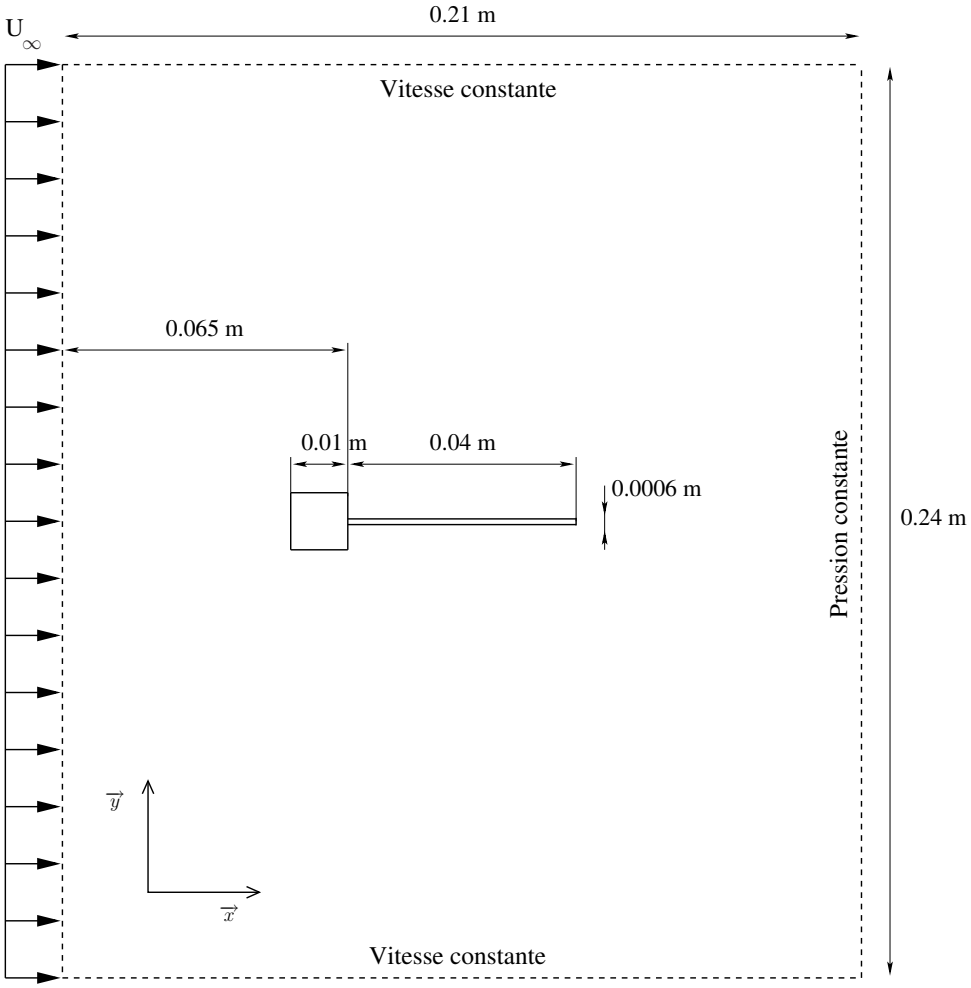


FIG. 6.8: Filament déformable fixé derrière un carré et plongé au sein d'un écoulement incompressible

Ici, la configuration de Hübner a été choisie. Les caractéristiques du fluide et de la structure, ainsi que la géométrie de la poutre sont décrites dans le tableau 6.3. Le nombre de Reynolds est de 204 (avec la longueur du carré prise pour longueur caractéristique de l'écoulement). Les calculs sont par conséquent réalisés en régime laminaire. Les conditions aux limites choisies sont les suivantes :

- ➢ Vitesse imposée en entrée et sur les frontières supérieure et inférieure
- ➢ Pression imposée en sortie
- ➢ Condition d'adhérence sur le filament

Les frontières extérieures du domaine de calcul m'ont semblé trop proches de la structure : même si ce ne sont pas des parois, une frontière trop proche avec une condition de pression ou de vitesse imposée peut modifier l'écoulement. J'ai donc agrandi le domaine en haut et en bas par rapport à l'original de Hübner.

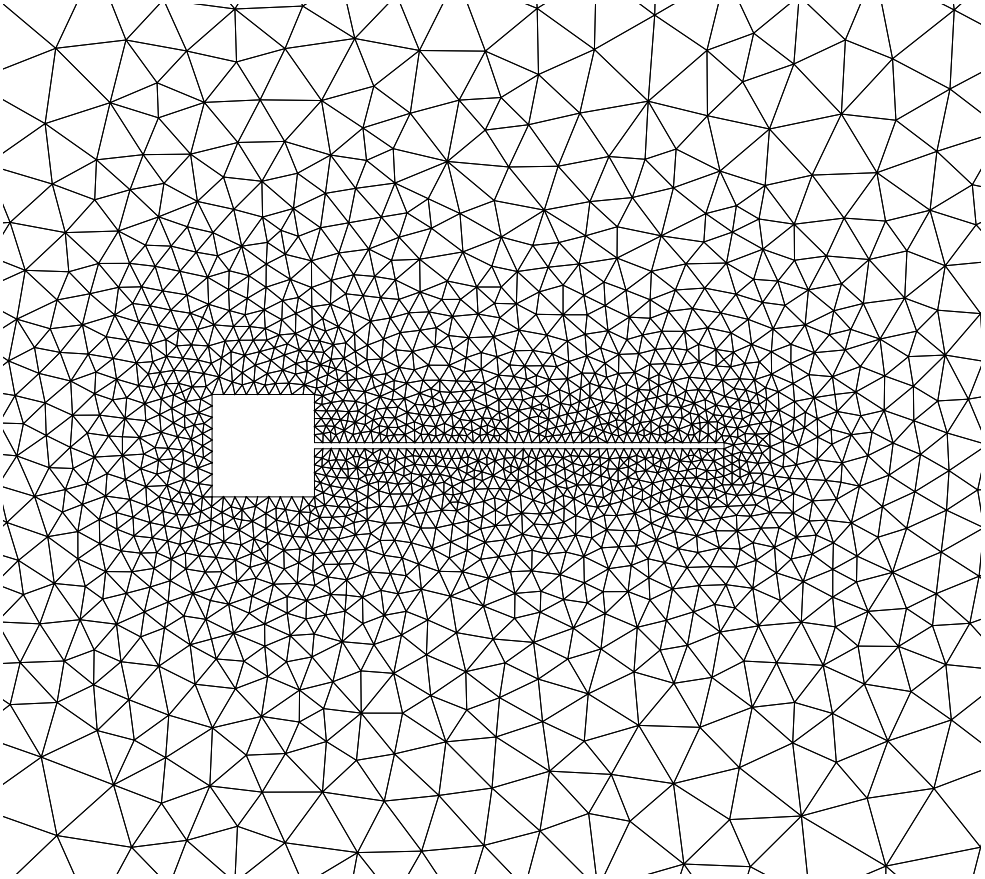
Côté du carré	$a = 0.01\,m$
Longueur de la poutre	$L = 0.04\,m$
Épaisseur de la poutre	$d = 0.0006\,m$
Module d'Young	$E = 2\,\text{MPa}$
Masse volumique du solide	$\rho_s = 2000\,kg.m^{-3}$
Masse volumique du fluide	$\rho_f = 1.18\,kg.m^{-3}$
Viscosité dynamique	$\mu_f = 1.8210^{-5}\,Pa.s$
Vitesse d'entrée	$U_\infty = 0.315\,m.s^{-1}$

Tab. 6.3: Conditions du cas-test (unités SI)

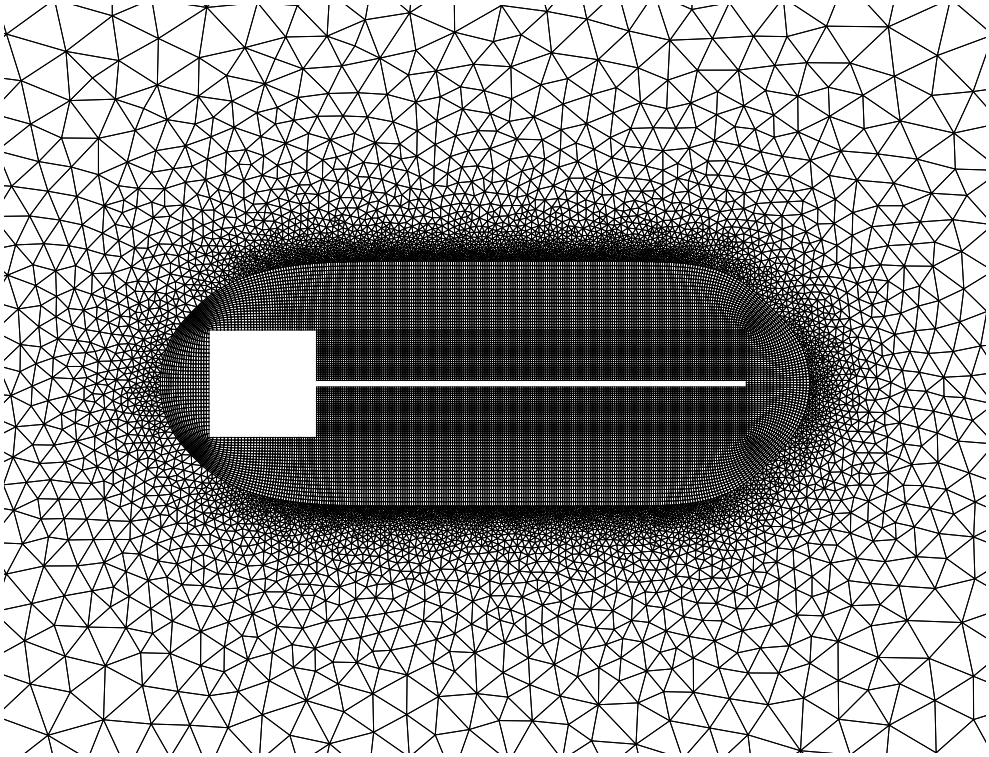
Les maillages du domaine fluide ont été générés par GRIDGEN ™. Un premier, plutôt grossier, compte 3654 cellules et 1918 nœuds (cf. fig. 6.9(a)). Il est entièrement non-structuré. Le second est beaucoup plus fin. Il est composé de 32944 cellules et de 27108 nœuds (cf. fig. 6.9(b)). Près du corps le maillage est structuré. Puis le reste du domaine est rempli de triangles.

En premier lieu, intéressons-nous à l'écoulement autour de la structure carré+filament, lorsque ce dernier est *indéformable*. À $t = 0\,s$, l'écoulement débute à l'entrée avec une vitesse imposée $U_\infty = 0.315\,m.s^{-1}$. Il n'y a pas d'initialisation progressive. Le cas-test est réalisé en premier sur le maillage grossier. Étudions l'évolution temporelle des efforts suivant $\overrightarrow{y}$ (cf. fig. 6.10(a)) : de $t = 0\,s$ à $t = 8\,s$ la courbe n'est pas régulière. L'écoulement n'est donc pas encore établi. Pour $t > 8\,s$, la courbe est quasi-sinusoïdale. En traçant une transformée de Fourier de cette partie de la courbe (cf. fig. 6.10(b)), nous obtenons la première fréquence caractéristique de l'écoulement : $f_1^f = 3.74\,Hz$. C'est en accord avec la valeur trouvée par Hübner dans [50].

Étudions maintenant le comportement du fluide autour de la structure : pour $t > 8\,s$, un lâcher périodique de tourbillons est établi. Ces vortex sont lâchés depuis les coins du carré à la fréquence de $3.74\,Hz$, ce qui correspond à la première fréquence obtenue via les efforts F_y. Durant une période de lâcher, trois tourbillons se déplacent le long du filament indéformable : un qui vient d'être lâché, un second au milieu de la poutre mais du côté



(a) Maillage grossier



(b) Maillage fin

FIG. 6.9: Maillages fluides autour du filament

opposé au premier et enfin un troisième qui se détache (cf. fig. 6.11). Ce lâcher de tourbillons entraîne une oscillation des efforts de part et d'autre du filament. Si celui-ci est déformable, il se met à vibrer.

Considérons à présent le filament comme une *poutre déformable*. Puis, résolvons le problème couplé fluide-structure à l'aide du code IFS décrit dans cette thèse.

Dans un premier temps, les calculs sont effectués sur le maillage grossier 6.9(a) avec un pas de temps $\Delta t = 5 \times 10^{-4}\, s$. Le maillage structure se compose de 100 éléments finis. Initialement, le fluide et la structure sont au repos. À $t = 0$, la vitesse du fluide en entrée est instantanément imposée à $U_\infty = 0.315\, m.s^{-1}$. Les efforts fluides entraînent une déformation de la poutre. Le filament se met à osciller dans son second mode (cf. fig. 6.12). L'évolution de l'extrémité libre de la poutre (le point B) est décrite par la courbe 6.13(a). On peut alors comparer ces résultats avec ceux présentés dans [100] : J.G. Valdés Vásquez confronte ses données avec celles d'Hübner et montre qu'elles concordent (cf. fig. 6.13(b)). La première fréquence des oscillations qu'il obtient (cf. fig. 6.14(b)), $f_1^c = 3.22\, Hz$ est proche de celle d'Hübner ($f_1^c = 3.10\, Hz$). Les résultats du code IFS développé concordent eux aussi de manière générale :

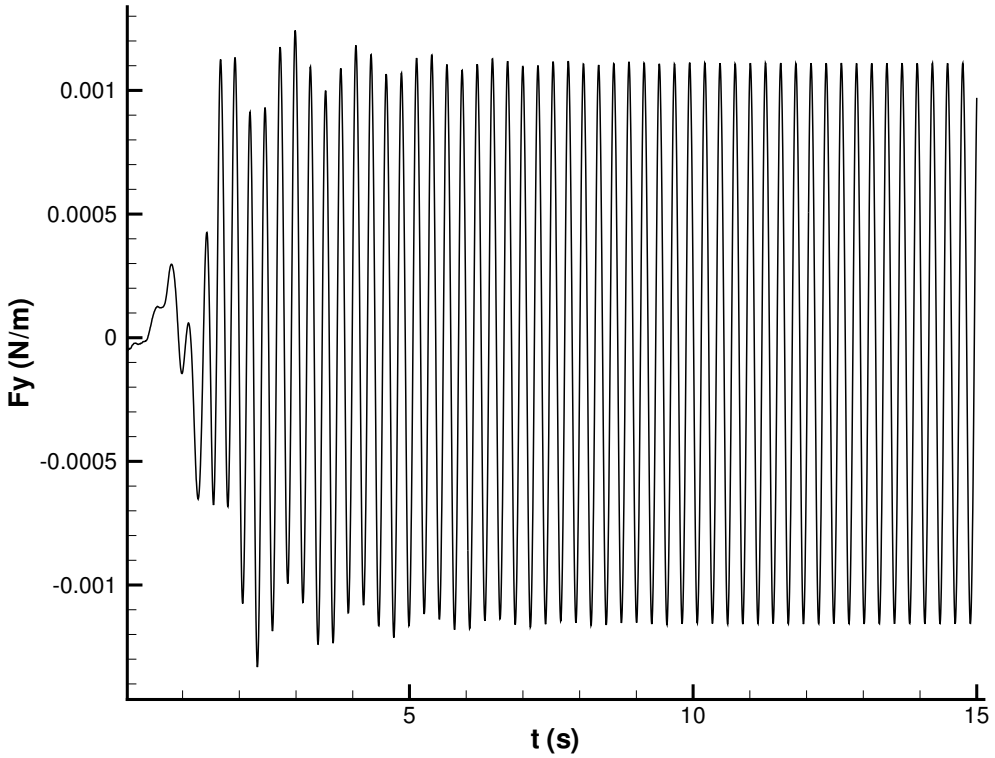
- ➢ L'évolution du point B de la figure 6.13(a) est semblable à celles d'Hübner et de Valdéz Vásquez. Les oscillations du point B ont un maximum à $y = 0.008\, m$.
- ➢ Les 3 premières fréquences d'oscillations (cf. fig. 6.14(a)) sont en très bon accord avec celles de Valdéz Vásquez (cf. fig. 6.14(b)).

Il est intéressant de s'attarder sur la valeur de la première fréquence du problème couplé f_1^c. On trouve le même ordre de grandeur que pour la première fréquence de l'écoulement f_1^f et que pour la seconde fréquence de la structure $f_2^s = 4.09\, Hz$. Pourtant, elle est plus basse. On peut penser que c'est une *influence de la première fréquence de la structure* ($f_1^s = 0.61\, Hz$) sur le problème couplé IFS.

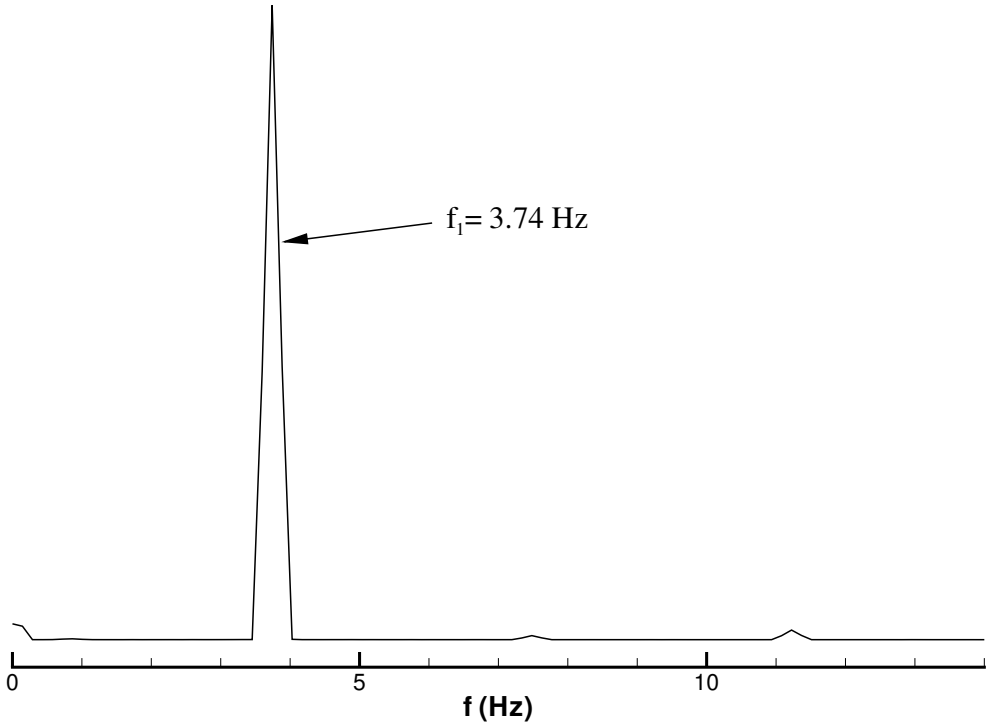
Cependant, en observant l'évolution du point B au cours du temps, on note que le mouvement couplé s'établit moins vite dans notre cas. De plus, pour $t > 16\, s$ des oscillations de plus en plus fortes apparaissent. Le maillage ayant servi pour les premiers calculs étant très grossier, de trop grandes erreurs numériques peuvent avoir altéré les résultats et engendré de tels problèmes. Par conséquent le calcul a été effectué sur le maillage fin 6.9(b).

Cette fois-ci le régime s'établit un peu plus vite (cf. fig. 6.15(a)). La courbe des oscillations du point B est plus régulière et offre une meilleure concordance avec les autres résultats. La première fréquence d'oscillations calculée à partir des efforts F_y diminue un peu ($f_1^c = 3.07\, Hz$) (cf. fig. 6.15(b)), mais reste toujours en très bon accord avec le résultat de Hübner ($f_1^c = 3.10\, Hz$).

L'assemblage de figures 6.16 présente l'écoulement et la déformée de la poutre lors d'une demi-période caractéristique du problème IFS. Sur ces images, le champ de pression est représenté. Chaque vortex est visible, car la pression est alors minimale (en bleu). On retrouve les trois gros tourbillons lâchés derrière le carré. Le deuxième mode de vibration de la poutre est bien visible.



(a) Évolution temporelle des efforts F_y fluides sur le filament



(b) Transformée de Fourier des efforts F_y pour $t > 8\,s$

FIG. 6.10: Étude des efforts F_y dans le cas du filament indéformable

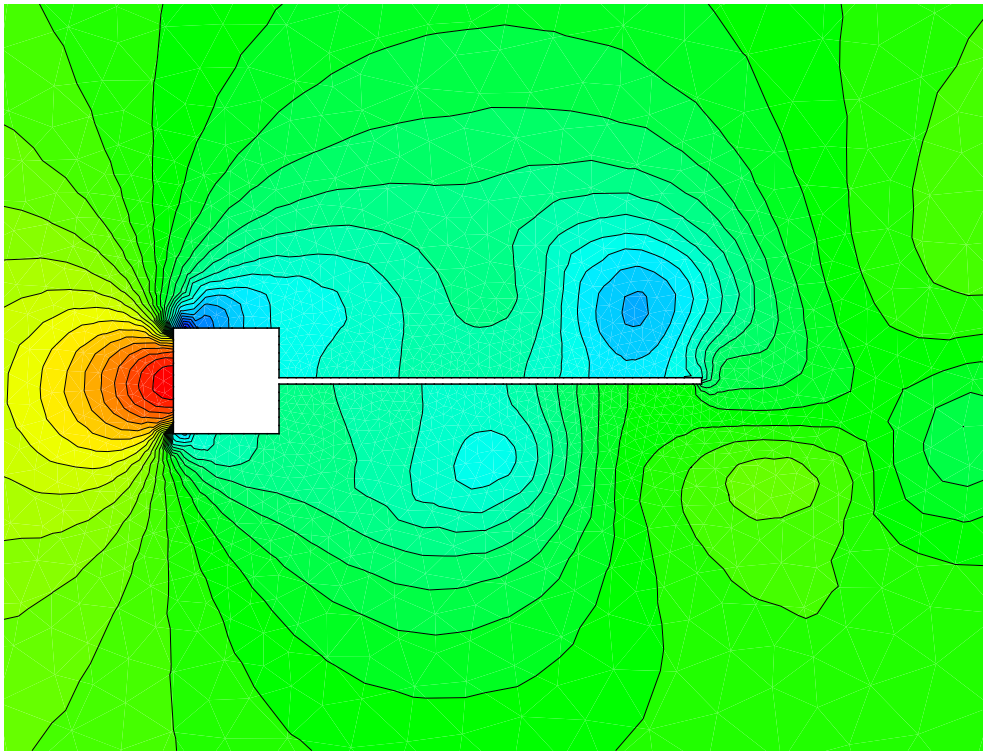


FIG. 6.11: Lâcher de tourbillons autour du filament indéformable

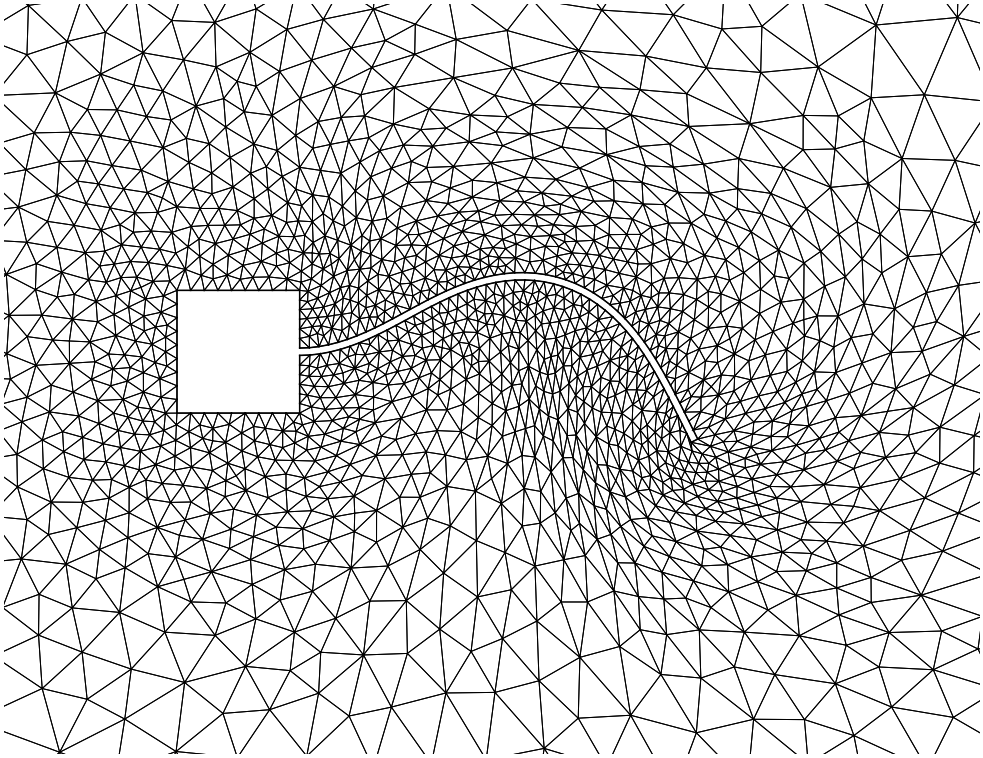
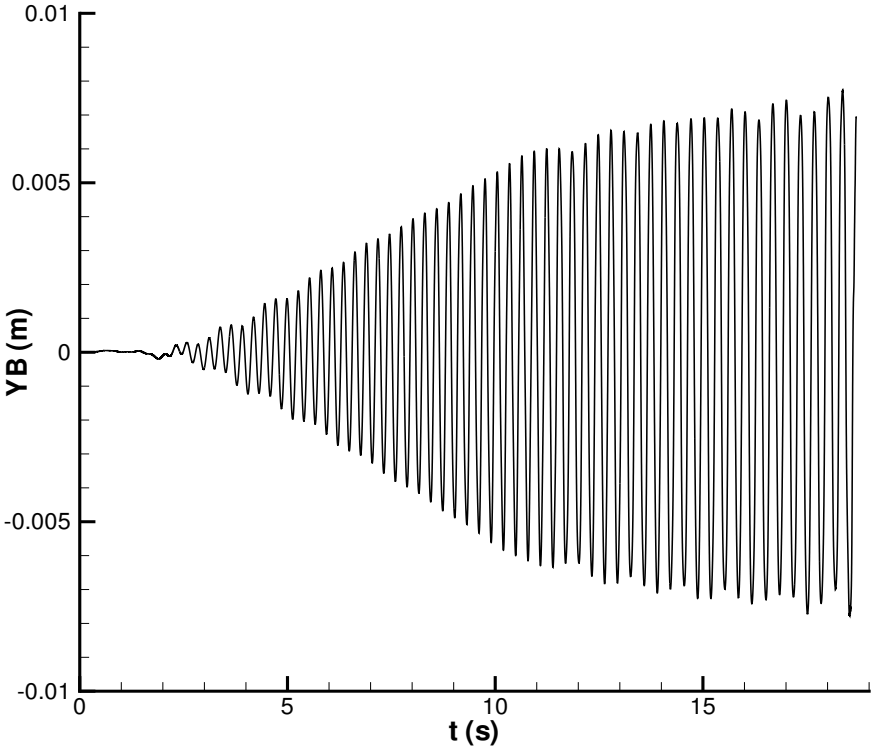
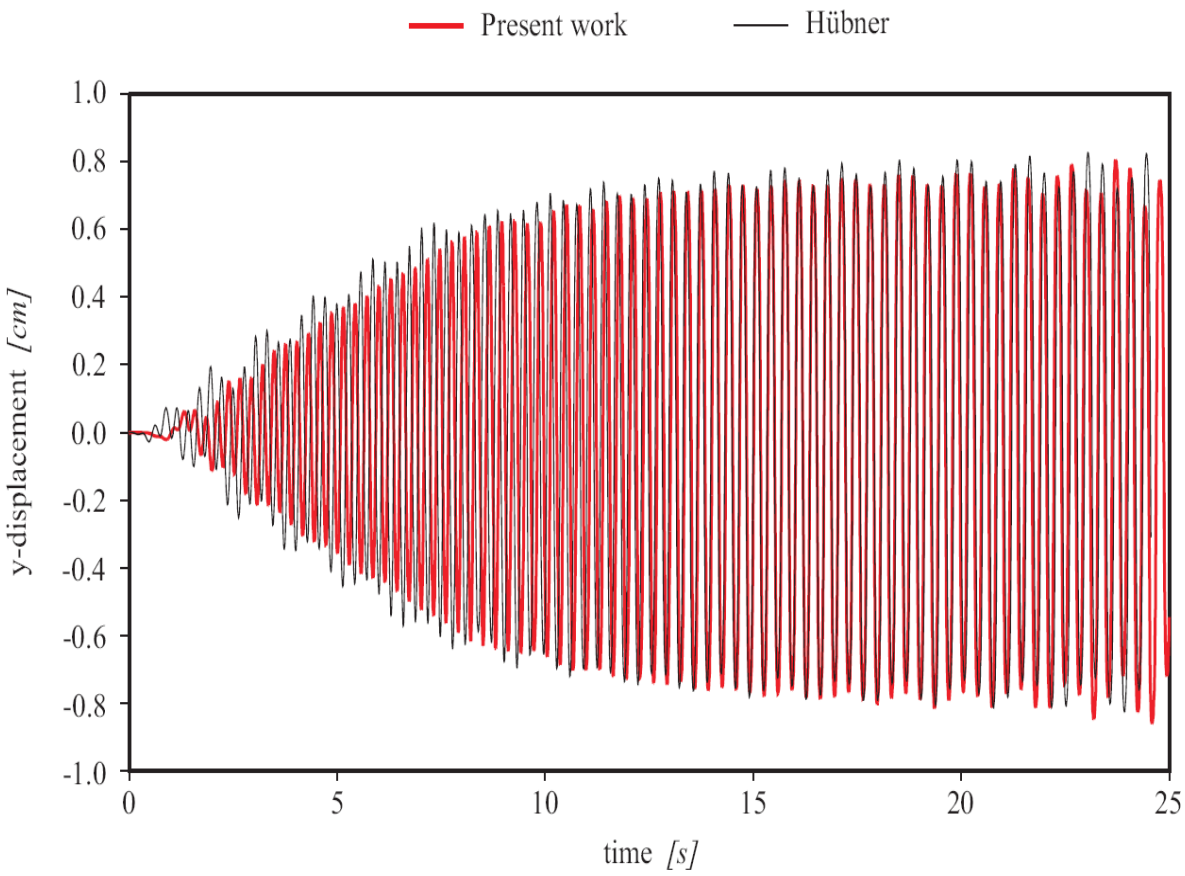


FIG. 6.12: Déformation du maillage grossier

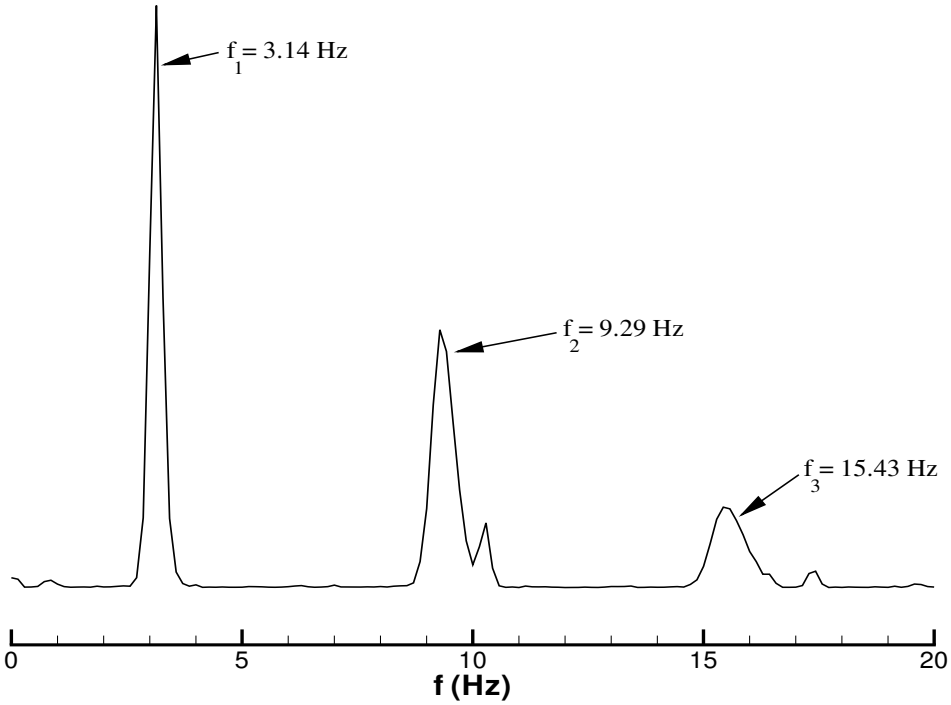


(a) Résultats obtenus avec le code IFS sur le maillage grossier

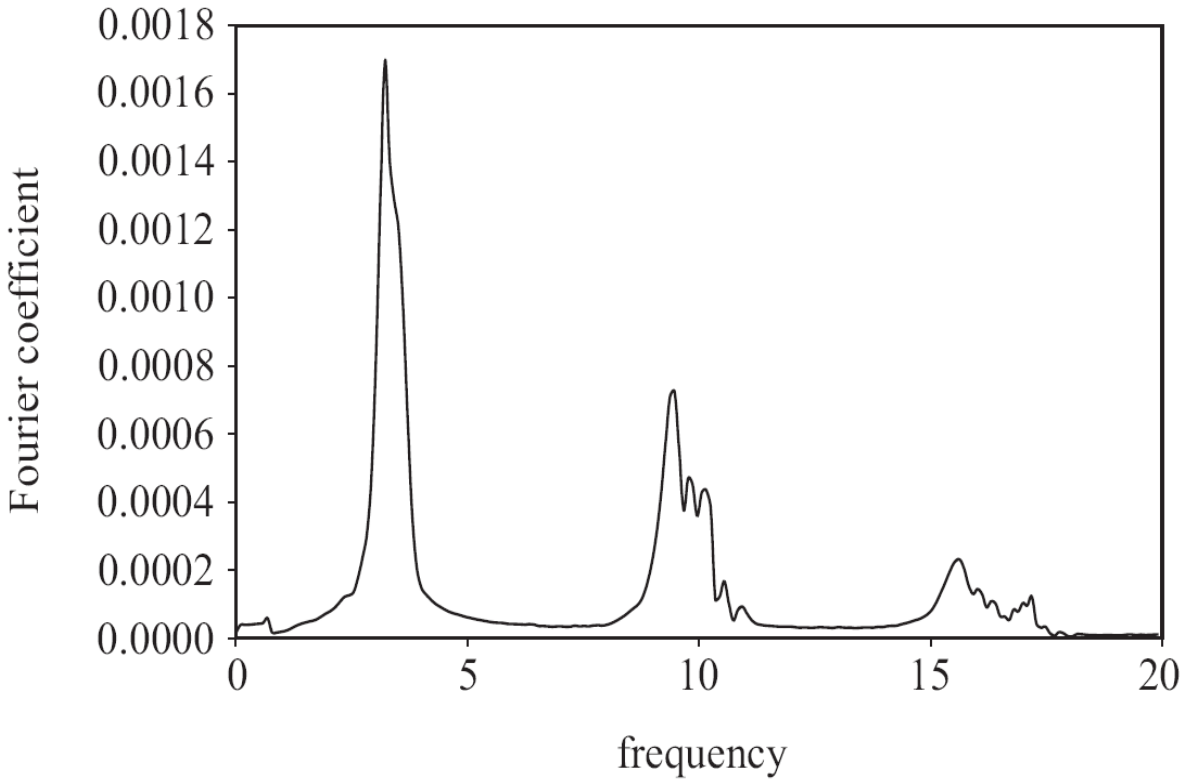


(b) Résultats extraits de la thèse [100]

FIG. 6.13: Évolution temporelle de l'extrémité libre du filament

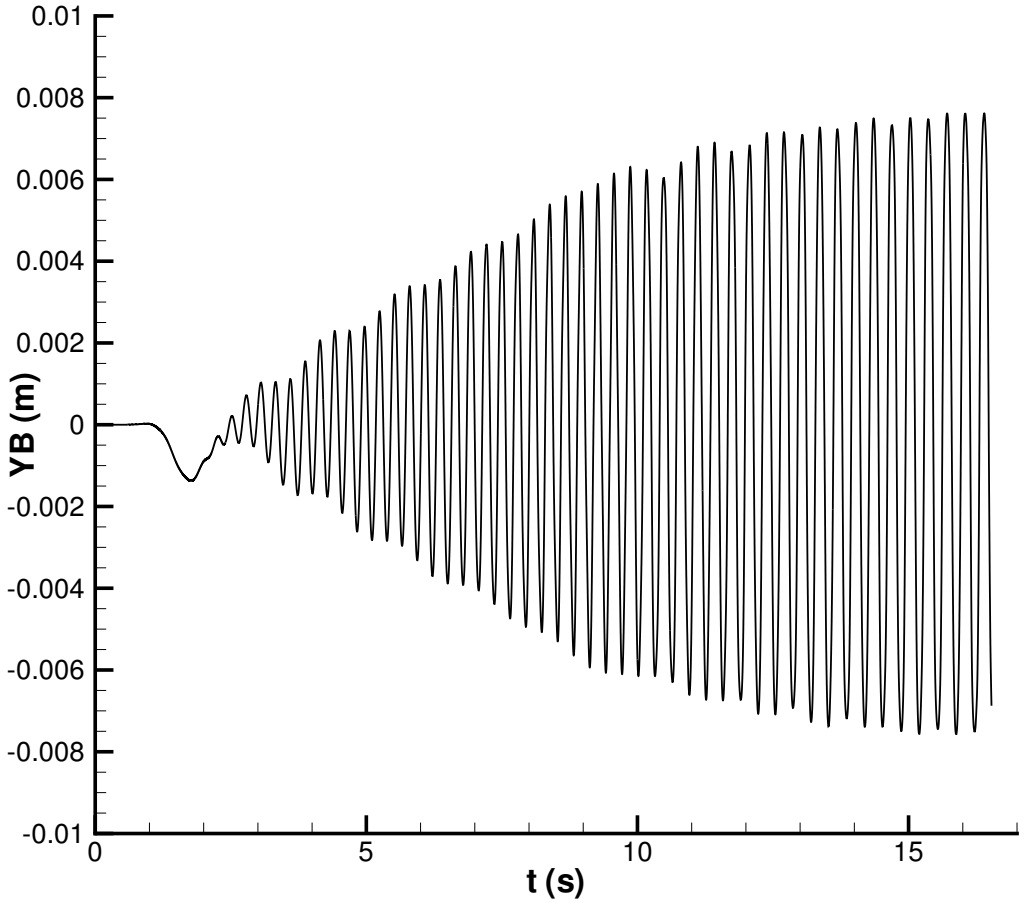


(a) Résultats obtenus avec le code IFS sur le maillage grossier (pour $t > 9.99\,s$)

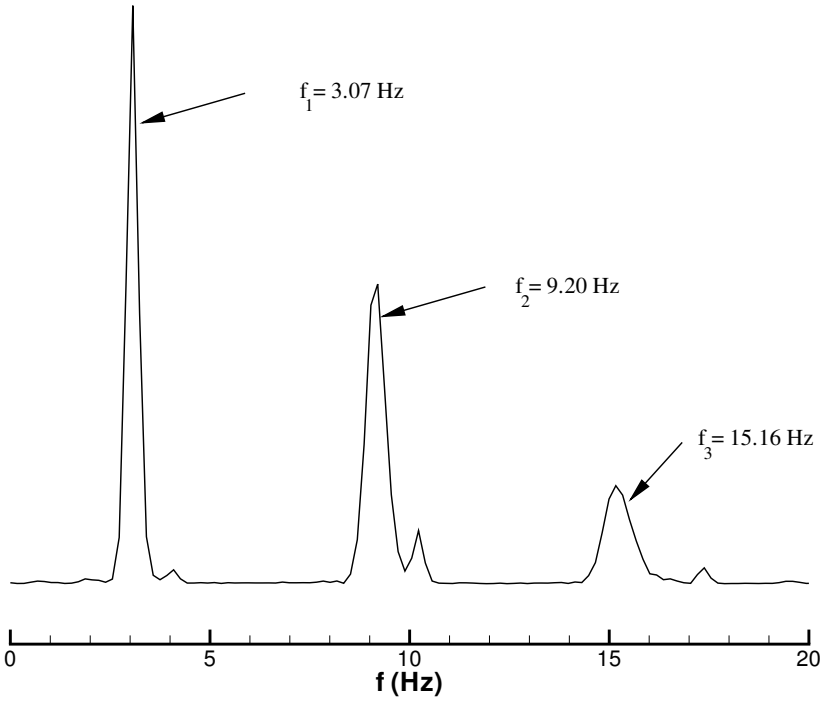


(b) Résultats extraits de la thèse [100]

FIG. 6.14: Transformée de Fourier des efforts F_y

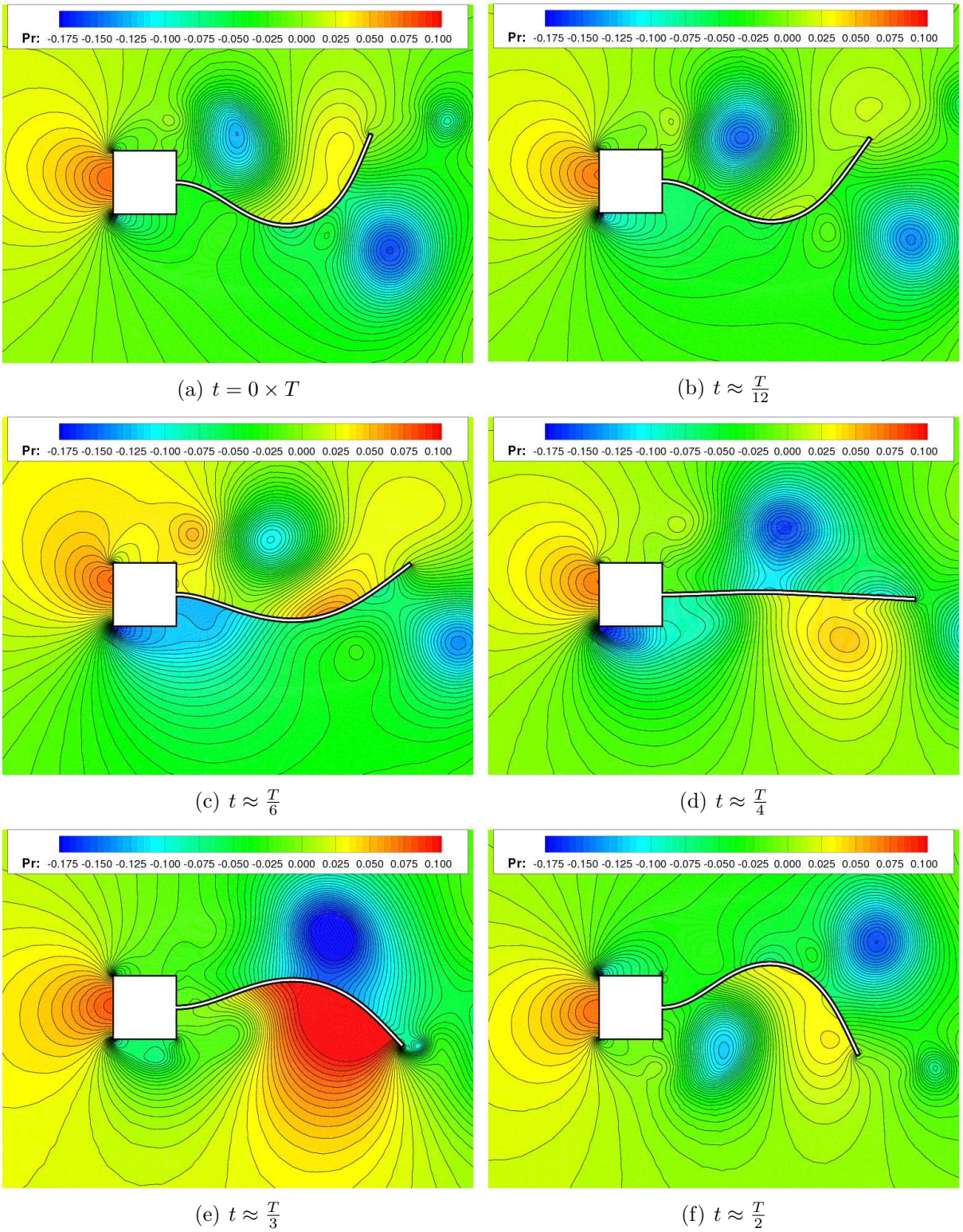


(a) Évolution temporelle de l'extrémité libre du filament



(b) Transformée de Fourier des efforts F_y pour $t > 9.99\,s$

FIG. 6.15: Résultats obtenus pour le filament déformable avec le maillage fin



(a) $t = 0 \times T$ (b) $t \approx \frac{T}{12}$

(c) $t \approx \frac{T}{6}$ (d) $t \approx \frac{T}{4}$

(e) $t \approx \frac{T}{3}$ (f) $t \approx \frac{T}{2}$

FIG. 6.16: Champ de pression autour du filament qui se déforme pendant une demi-période $\frac{T}{2} \approx 0.16\,s$ (calcul réalisé sur le maillage fin)

Les résultats obtenus concordent avec ceux obtenus par Hübner et Valdés Vásquez, tant au niveau de l'amplitude maximale que sur les fréquences des oscillations. Cependant, sur le maillage fluide grossier comme sur le maillage fluide fin, on constate que le couplage IFS explose à $t = 18.5\,s$ pour le maillage grossier et à $t = 16.5\,s$ pour le maillage fin.

Le problème ne vient pas du maillage, puisqu'il apparaît aussi lors de l'utilisation du maillage fin. Notons à ce propos que le maillage grossier est suffisant pour estimer la fréquence des oscillations f_1^c. Le maillage fin est nécessaire si l'on veut visualiser l'écoulement dans le détail. Remarquons aussi que le nombre de noeuds du maillage structure peut être réduit de 100 à 20 sans dégrader les courbes de déplacement du point extrémité. Il en résulte un petit gain en temps. Par contre, pour une vue de la déformation globale de la poutre (cf. fig. 6.16) et du maillage fluide déformé (cf. fig. 6.12), il est préférable d'avoir un nombre d'éléments élevé.

Le problème de couplage n'est pas dû à la taille du pas de temps. Il a été choisi très petit ($\Delta t = 5 \times 10^{-4}\,s$), par rapport à ce que l'on trouve dans la littérature ($\Delta t = 5 \times 10^{-3}\,s$ pour Valdés Vásquez). En effet, avec $\Delta t = 5 \times 10^{-3}\,s$ notre code IFS ne dépasse pas les $12\,s$. Cela entraîne une grande durée de calcul pour ce cas-test 2D allant de 6 jours pour le maillage grossier à 10 pour le maillage fin (machine utilisée : Intel Pentium 4 cadencé à 3.2GHz avec 2Go de RAM). Les limites du code IFS sont donc mises en évidence avec cette application.

Pour repousser ces frontières, on peut envisager d'utiliser la méthode de la masse ajoutée qui amène plus de robustesse. Mais, en premier lieu, intéressons-nous au couplage en temps. Le schéma standard de Newmark présente certaines failles. Il génère des perturbations dans les hautes fréquences qui peuvent être déstabilisantes pour le couplage.

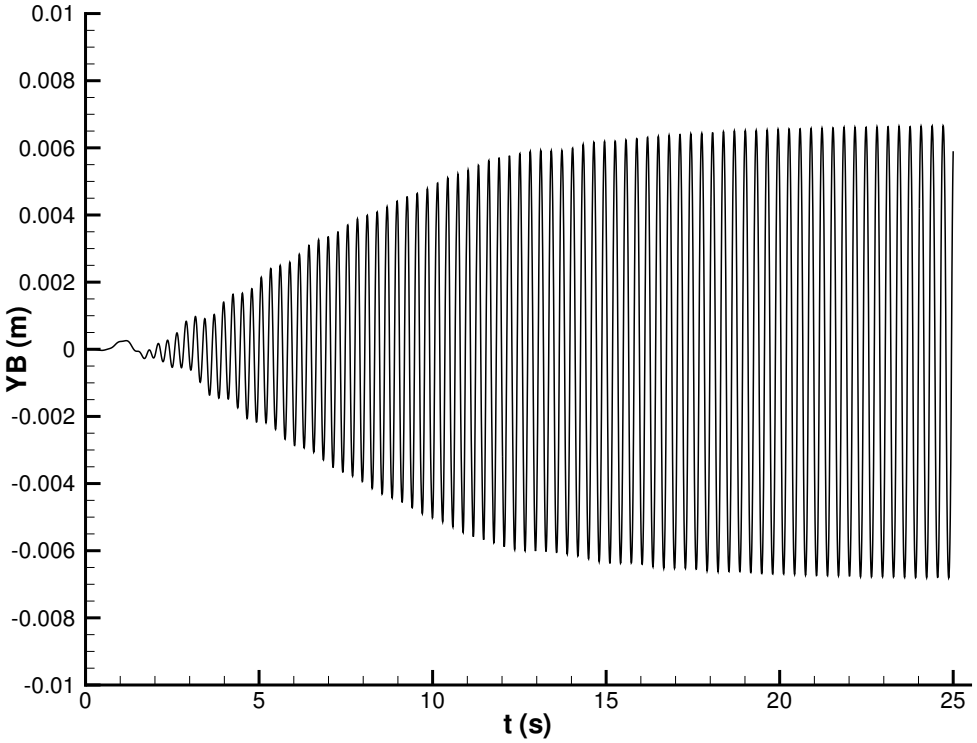
Dans un premier temps, une simple redéfinition des paramètres γ et β du schéma standard de Newmark permet d'obtenir un **schéma de Newmark amorti**. On introduit le coefficient d'amortissement α_{amort} de la manière suivante :

$$\alpha_{amort} \in \left[0, \frac{1}{3}\right], \qquad \gamma = \frac{1 + 2\alpha_{amort}}{2}, \qquad \beta = \frac{(1 + \alpha_{amort})^2}{4} \tag{6.1}$$

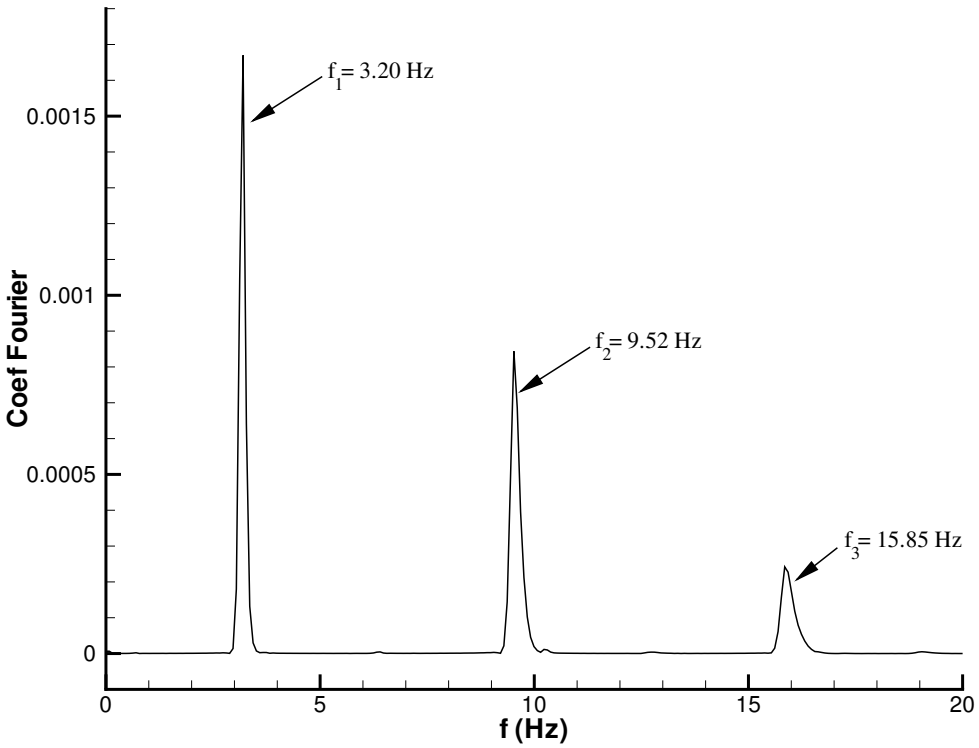
On retrouve le schéma standard de Newmark pour $\alpha_{amort} = 0$.

En prenant $\alpha_{amort} = 0.035$, le calcul IFS se termine sans problème avec un pas de temps plus grand ($\Delta t = 5 \times 10^{-3}\,s$). Le gain en temps est très important : le calcul prend à peine 5 heures sur la même machine que précédemment. En traçant les déplacements de l'extrémité libre de la poutre par rapport au temps (cf. fig. 6.17(a)), on remarque tout de suite que la courbe est plus lisse que les précédentes (cf. fig. 6.10(a) et 6.15(a)). La deuxième remarque importante est que l'amplitude maximale n'est plus la même ($0.0065\,m$, au lieu de $0.008\,m$). Au niveau des fréquences obtenues par transformée de Fourier sur les efforts F_y (cf. fig. 6.17(b)), on retrouve une valeur de $f_1^c = 3.20\,Hz$ très proche de celle de Valdés Vásquez ($f_1^c = 3.22\,Hz$ pour un même pas de temps $\Delta t = 5 \times 10^{-3}\,s$). La différence entre cette valeur de f_1^c et celle obtenue précédemment s'explique par le pas de temps plus grand. Le calcul est moins précis.

Pour traiter ce cas-test, Valdés Vásquez, dans sa thèse [100], utilise un schéma temporel de Newmark modifié : **"the generalized-α method"** présentée dans [24]. C'est une bonne



(a) Évolution temporelle de l'extrémité libre du filament



(b) Transformée de Fourier des efforts F_y pour $t > 9.99\,s$

FIG. 6.17: Résultats obtenus pour le filament déformable avec le schéma de Newmark amorti ($\alpha_{amort} = 0.035$)

amélioration du schéma standard de Newmark. Elle permet de dissiper les perturbations hautes fréquences générées par ce dernier. D'autres schémas de Newmark modifiés existent et sont plus simples à intégrer. Dans la littérature, on trouve par exemple la méthode **HHT** (Hilbert-Hughes-Taylor) (cf. [48]) ou encore celle de **Bossak** (cf. [108]). Cette dernière a été intégrée dans le code IFS. Cette méthode amène une *dissipation numérique des hautes fréquences sans dégrader l'ordre de précision*. Le calcul a donc été relancé avec un coefficient d'amortissement de 0.1. Le pas de temps a été fixé à $\Delta t = 10^{-3}\,s$ pour que la technique de résolution rapide explicite puisse fonctionner. En effet, avec la méthode de Bossak, l'amplitude des oscillations n'est plus amortie. Le mouvement est donc plus violent et le remaillage plus difficile. Avec ces paramètres, la simulation se termine sans problème en 16 heures (sur la même machine que précédemment) pour un temps physique de $50\,s$. La courbe des oscillations du point extrémité libre (cf. fig. 6.18) montre que l'amplitude maximale est de $0.008\,m$. La première fréquence des oscillations descend vers $f_1^c = 3\,Hz$. De fortes oscillations apparaissent pour $t > 20\,s$, mais le couplage tient bon grâce à l'amortissement numérique des hautes fréquences fourni par la méthode de Bossak (cf. fig.6.19). On voit donc avec cet exemple les possibilités qu'elle offre.

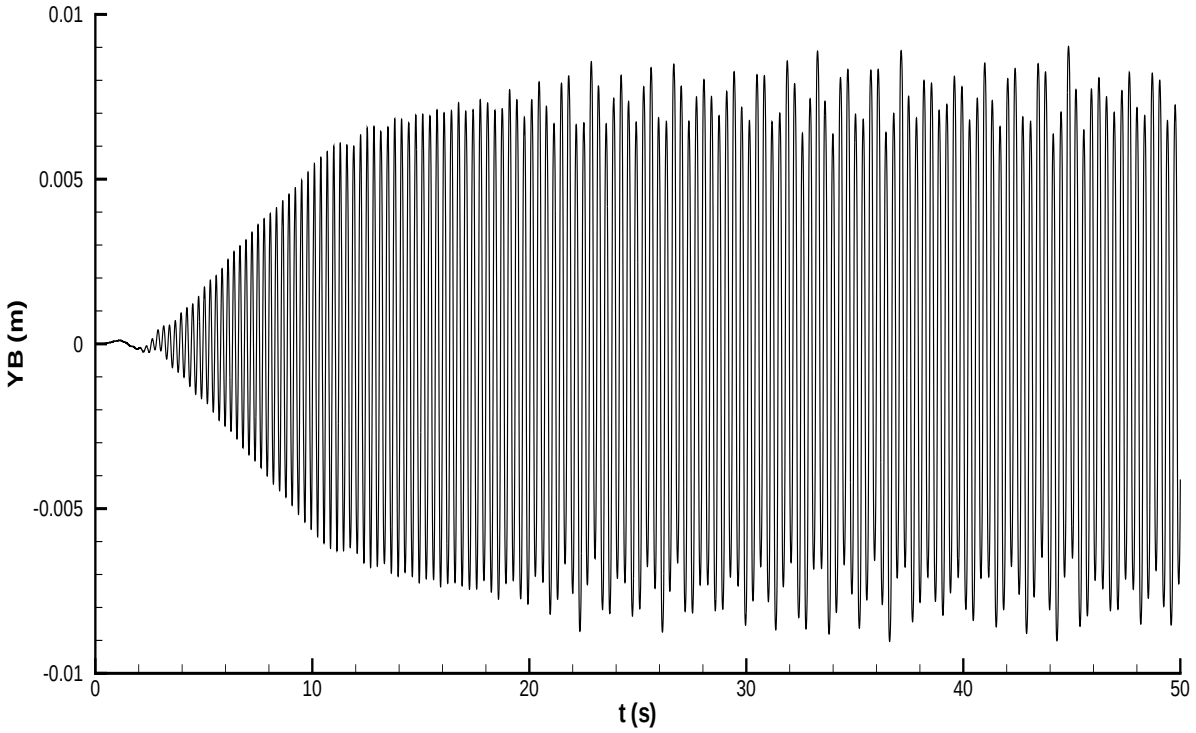


FIG. 6.18: Évolution temporelle de l'extrémité libre du filament en utilisant la méthode de Bossak

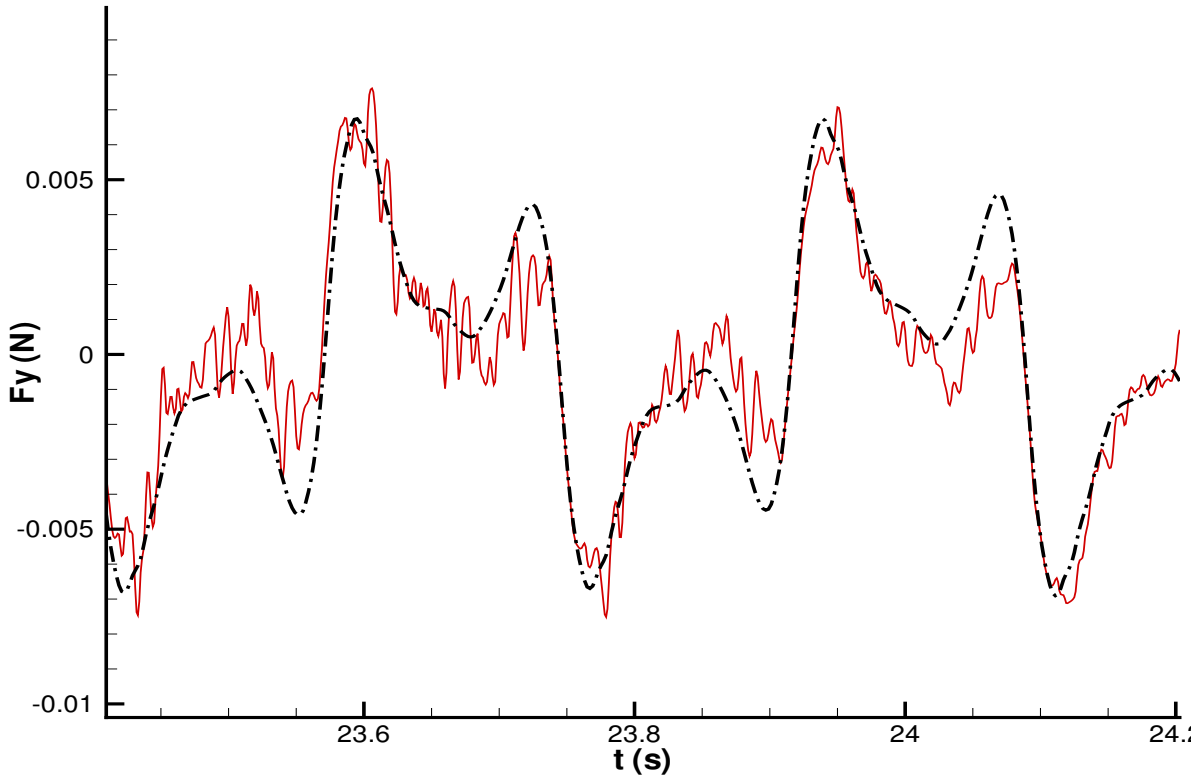


FIG. 6.19: Comparaison des efforts F_y obtenus avec le schéma standard de Newmark (en rouge et en trait continu) et avec celui de Bossak (en noir et en trait-pointillé)

6.2 Cas-tests 3D

Les deux premières applications tridimensionnelles consistent en un câble fixé à une extrémité et plongé dans un milieu multifluide. Elles ont pour but de montrer les possibilités qu'offre le code IFS en 3D. Elles ne sont pas vérifiées par l'expérience et sont réalisées sur un maillage léger pour diminuer le temps de calcul.

6.2.1 Câble fixé à une extrémité et soumis à un courant

Un câble déformable est plongé en partie dans l'air et en partie dans l'eau (cf. fig. 6.20). L'extrémité émergeant dans l'air est fixe, alors que celle dans l'eau est libre. Les fluides sont animés d'une vitesse constante $\overrightarrow{V}$, dirigée suivant $\overrightarrow{x}$.

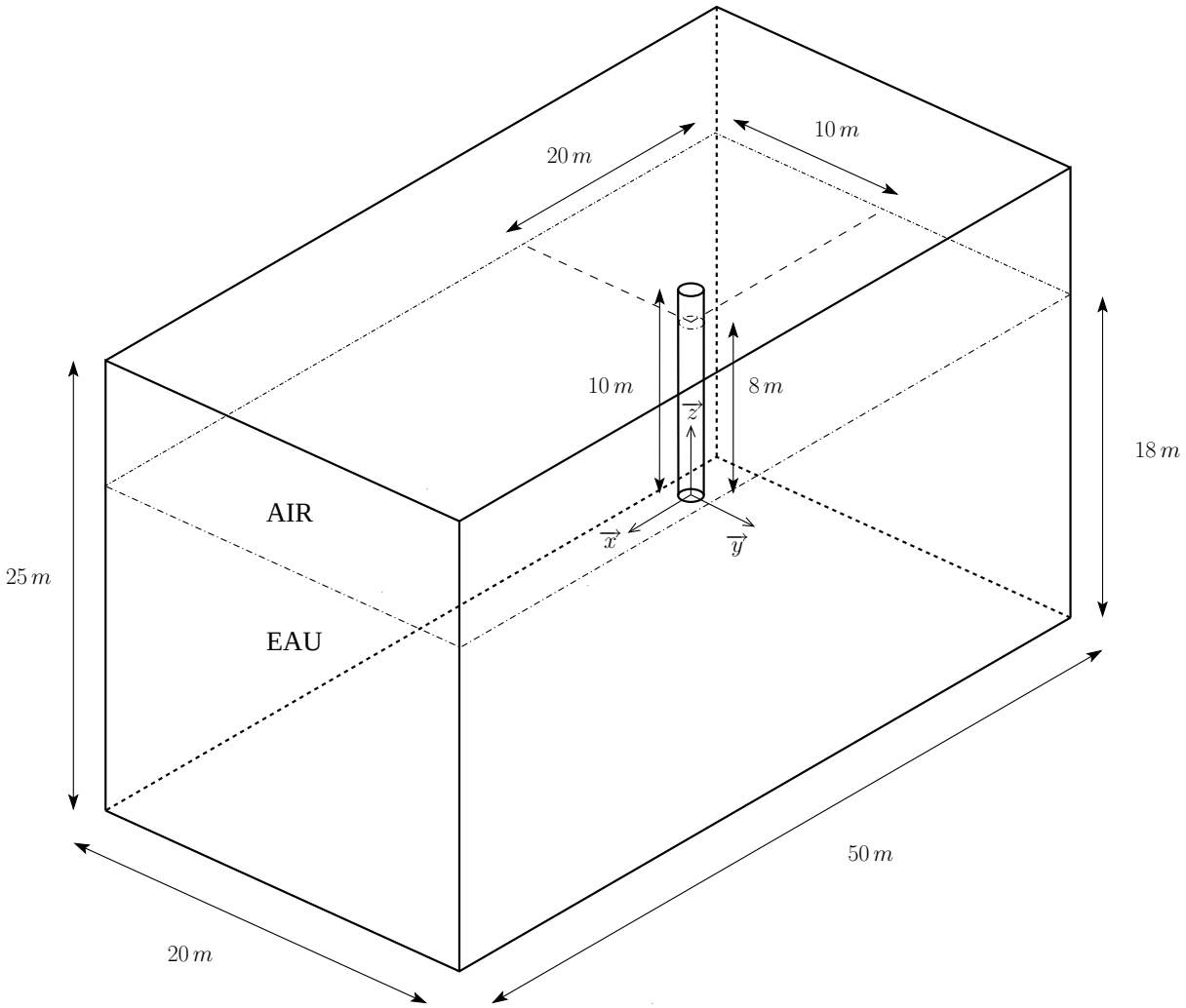


FIG. 6.20: Câble fixé à une extrémité et plongé en partie dans l'air et en partie dans l'eau

On teste les limites du code IFS. Un matériau qui n'existe pas est sélectionné pour la poutre. Il se rapproche d'un polymère par la valeur de son module d'Young, mais sa masse volumique est plus faible. Il permet une déformation importante de la structure et

entraîne aussi a priori des problèmes de stabilité plus importants. Les caractéristiques de ce premier cas-test sont présentées dans le tableau 6.4.

Longueur du câble	$L = 10\,m$
Diamètre du câble	$D = 0.2\,m$
Module d'Young du câble	$E = 3500\,\text{MPa}$
Masse volumique du câble	$\rho_s = 500\,kg.m^{-3}$
Masse volumique du fluide 1 (Eau)	$\rho_{eau} = 998.1\,kg.m^{-3}$
Viscosité dynamique (Eau)	$\mu_{eau} = 0.001\,Pa.s$
Masse volumique du fluide 2 (Air)	$\rho_{air} = 1.2\,kg.m^{-3}$
Viscosité dynamique (Air)	$\mu_{air} = 1.82 \times 10^{-5}\,Pa.s$
Vitesse des fluides	$V_x = 2.5\,m.s^{-1}$
Vitesse réduite des fluides	$V_r = \dfrac{V_x}{f_s\,D} = 16.89$

TAB. 6.4: Conditions du cas-test du câble soumis à un courant (unités SI)

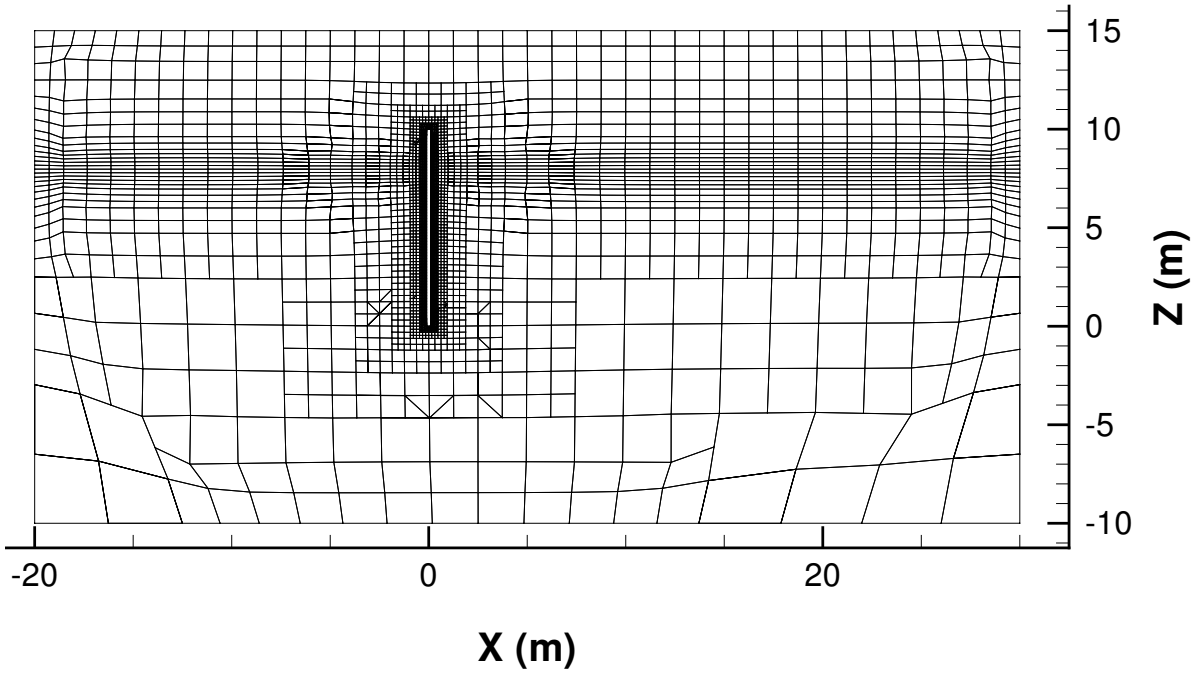
Le calcul est exécuté sur un maillage HEXPRESS ™ léger de 65000 cellules (cf. fig. 6.21). Il permet un calcul rapide sur des machines monoprocesseur. Ainsi, les limites du code IFS sont vite trouvées. Les conditions aux limites sont les suivantes :

- ➢ Vitesse imposée en entrée, en sortie et sur les frontières gauche, droite et inférieure
- ➢ Pression imposée sur la frontière supérieure
- ➢ Loi de paroi sur le câble

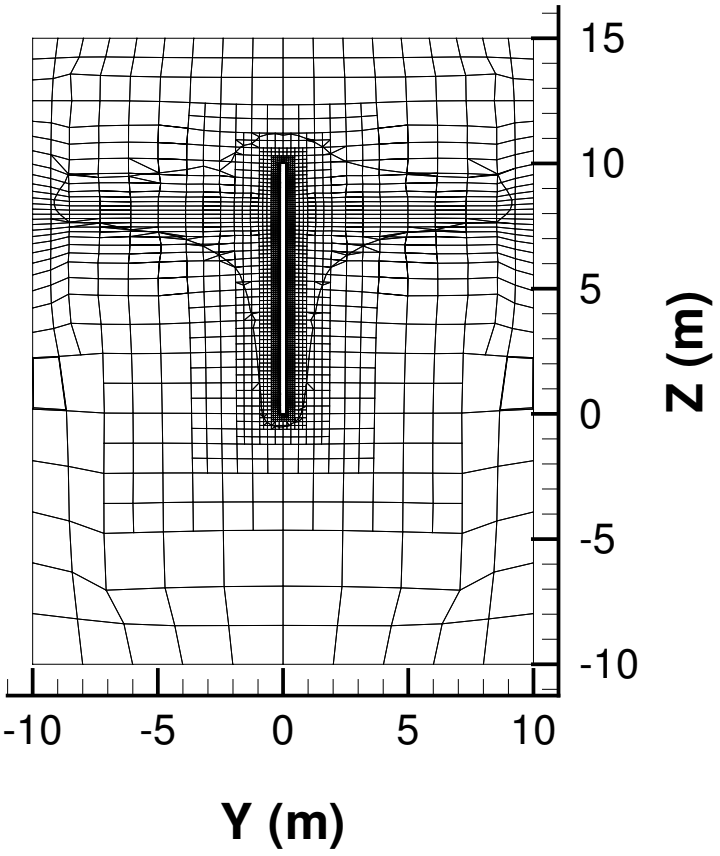
Le temps physique est de $20\,s$ avec un pas de temps de $10^{-2}\,s$. Le temps de calcul s'élève à 14 heures (sur un PC AMD Athlon 64 X2 Dual-core 4600+ cadencé à 2.4GHz avec 8GO de RAM). Le modèle de turbulence utilisé est celui de Menter ($k-\omega\,SST$).

À $t = 0\,s$, l'eau et l'air sont instantanément en mouvement et exercent une force sur le câble. Ce dernier se déforme donc : tout d'abord la poutre se courbe de manière à offrir le moins de résistance possible aux fluides (cf. fig. 6.22). Une fois qu'il a atteint cette position, le riser se met à osciller de manière périodique (cf. fig. 6.23(a)). Ce sont les **VIV** : les vortex créés par l'eau font vibrer le cylindre. La fréquence dominante des oscillations est calculée par transformée de Fourier (cf. fig. 6.23(b)) et vaut $f = 2.4\,Hz$. Cette valeur est donnée à titre indicatif. Le maillage est extrêmement léger pour un calcul 3D aussi complexe que celui-ci.

Cet exemple montre que le code IFS permet de traiter les interactions entre un câble fixé à une extrémité et son environnement multifluide, et ceci en grand déplacement. On peut à ce sujet examiner la déformation du maillage autour du câble (cf. fig. 6.24) et voir que ce dernier reste orthogonal à la paroi.

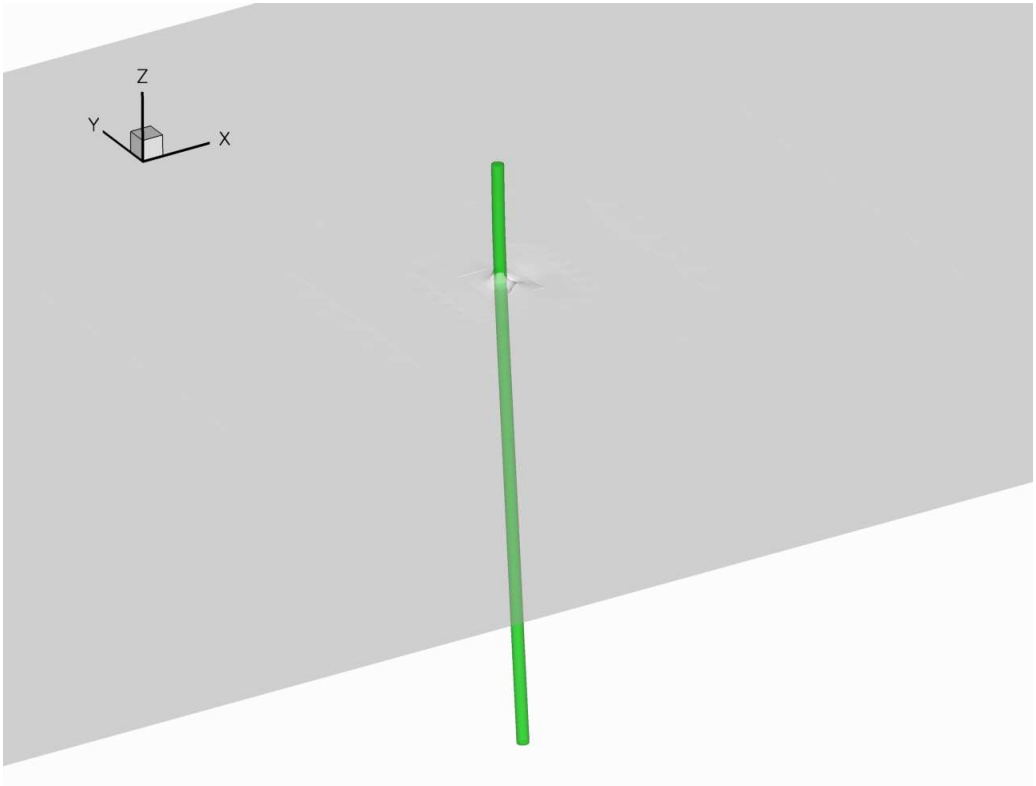


(a) Coupe suivant $(\overrightarrow{z}, \overrightarrow{x})$

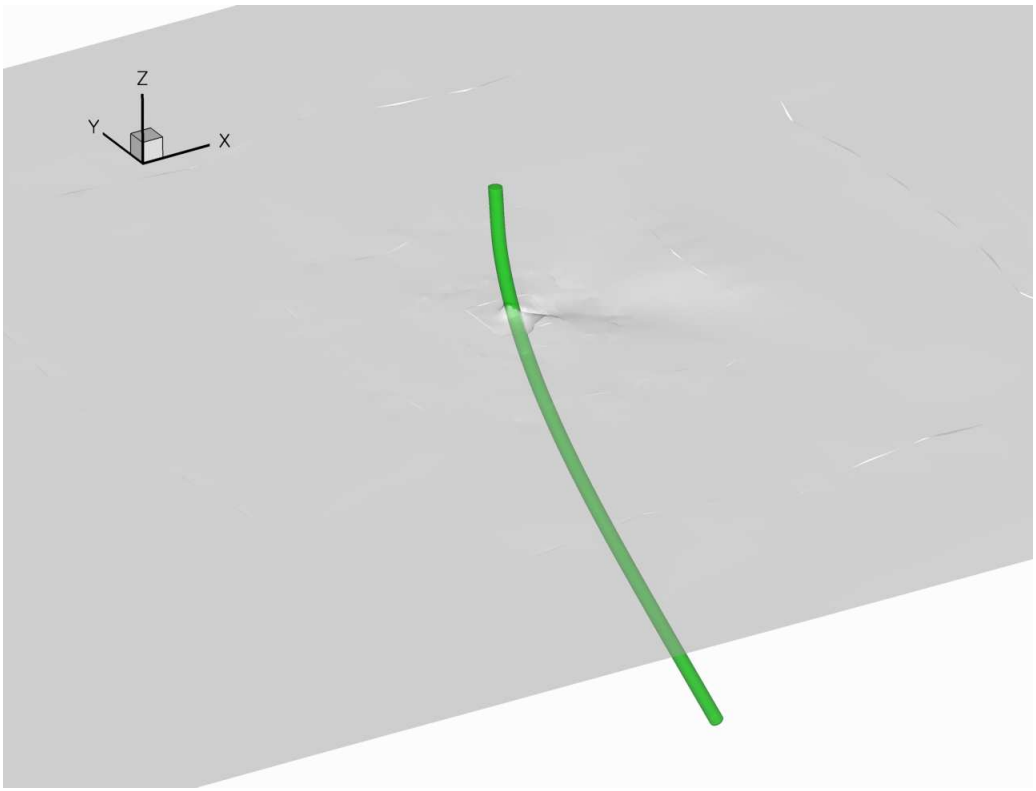


(b) Coupe suivant $(\overrightarrow{y}, \overrightarrow{z})$

FIG. 6.21: Maillage HEXPRESS™ léger

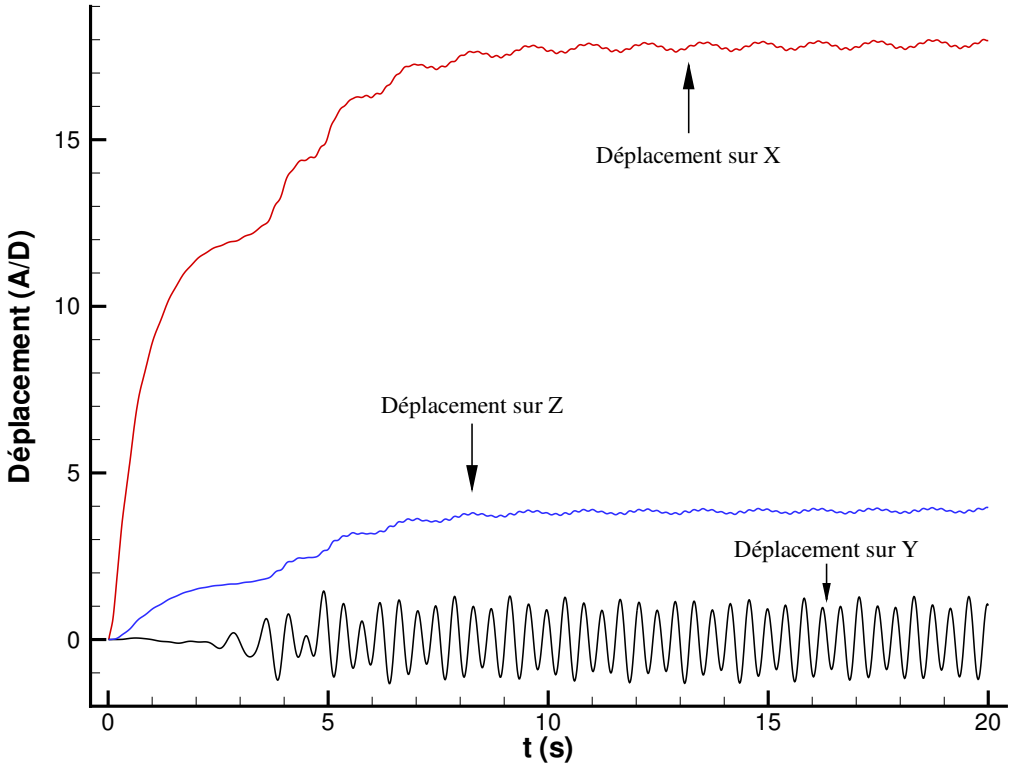


(a) Configuration initiale

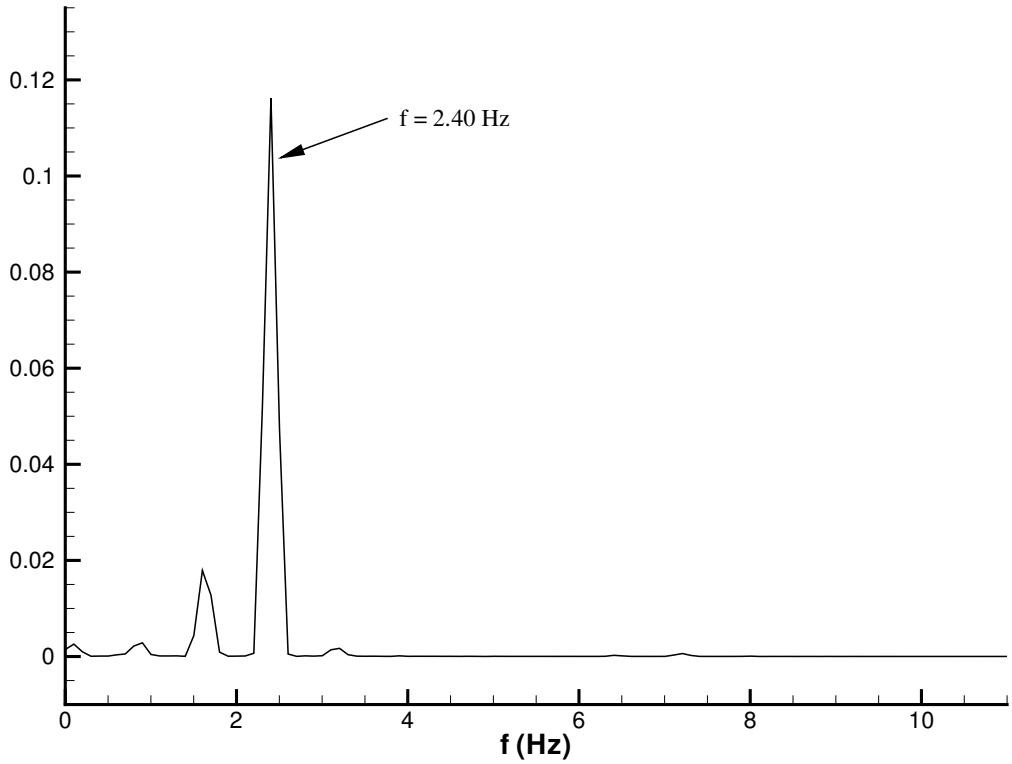


(b) Configuration déformée à $t = 15\,s$

FIG. 6.22: Câble soumis à un courant (avec en bleu-vert la surface libre)

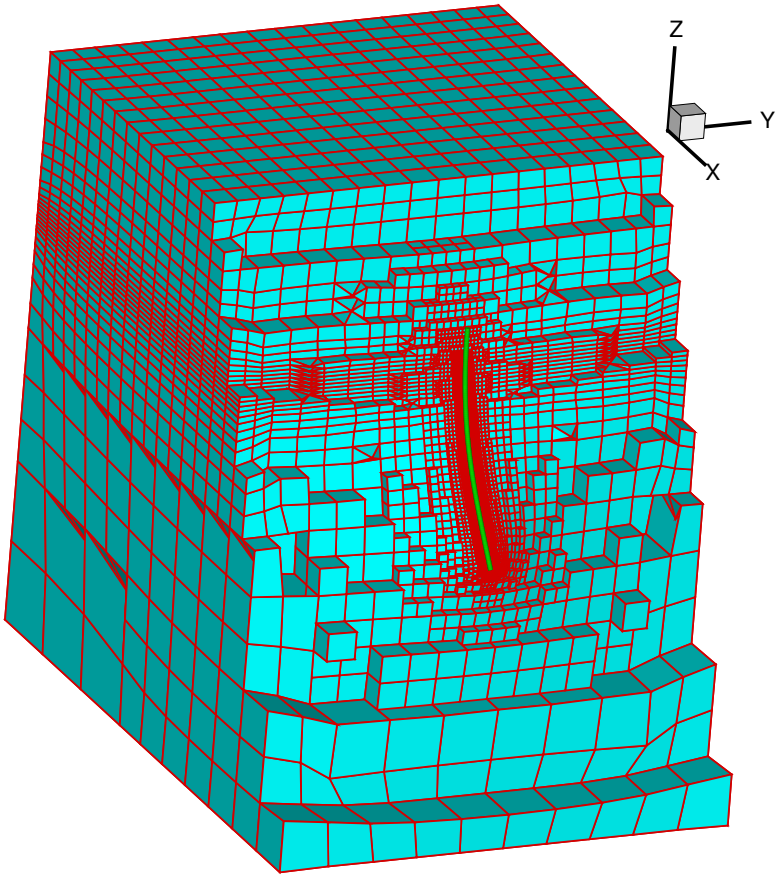


(a) Évolution des déplacements adimensionnels au cours du temps

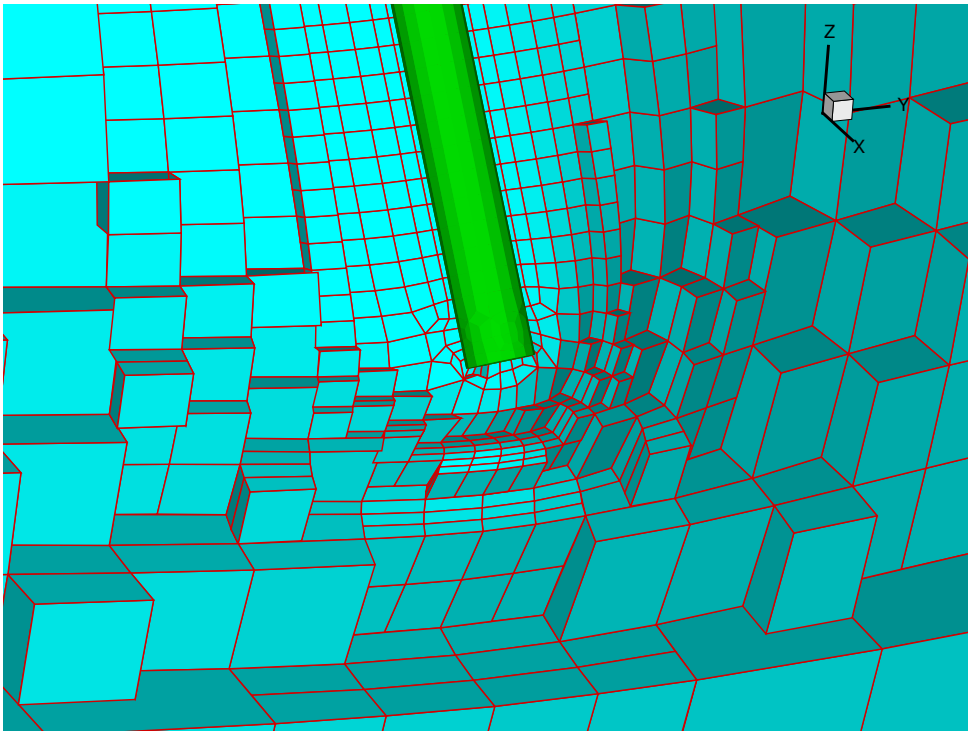


(b) Transformée de Fourier du déplacement suivant $\overrightarrow{y}$

FIG. 6.23: Étude des déplacements réduits ($\frac{A}{D}$) de l'extrémité libre du câble soumis à un courant



(a) Vue globale



(b) Zoom sur l'extrémité libre du câble

FIG. 6.24: Maillage déformé autour du câble à l'instant $t = 15\,s$

Le maillage utilisé permet un calcul monoprocesseur. On profite de cette particularité pour comparer le résultat avec celui obtenu via un calcul en parallèle. Le maillage initial est divisé à l'aide de $M\epsilon TIS$ (cf. [67]) en deux blocs de 30000 cellules environ. Les déplacements de la poutre sont très proches (cf. fig. 6.25), ce qui valide l'implémentation du calcul parallèle dans le code IFS. Les résultats ne sont pas identiques, car la résolution du problème n'est pas faite au zéro machine.

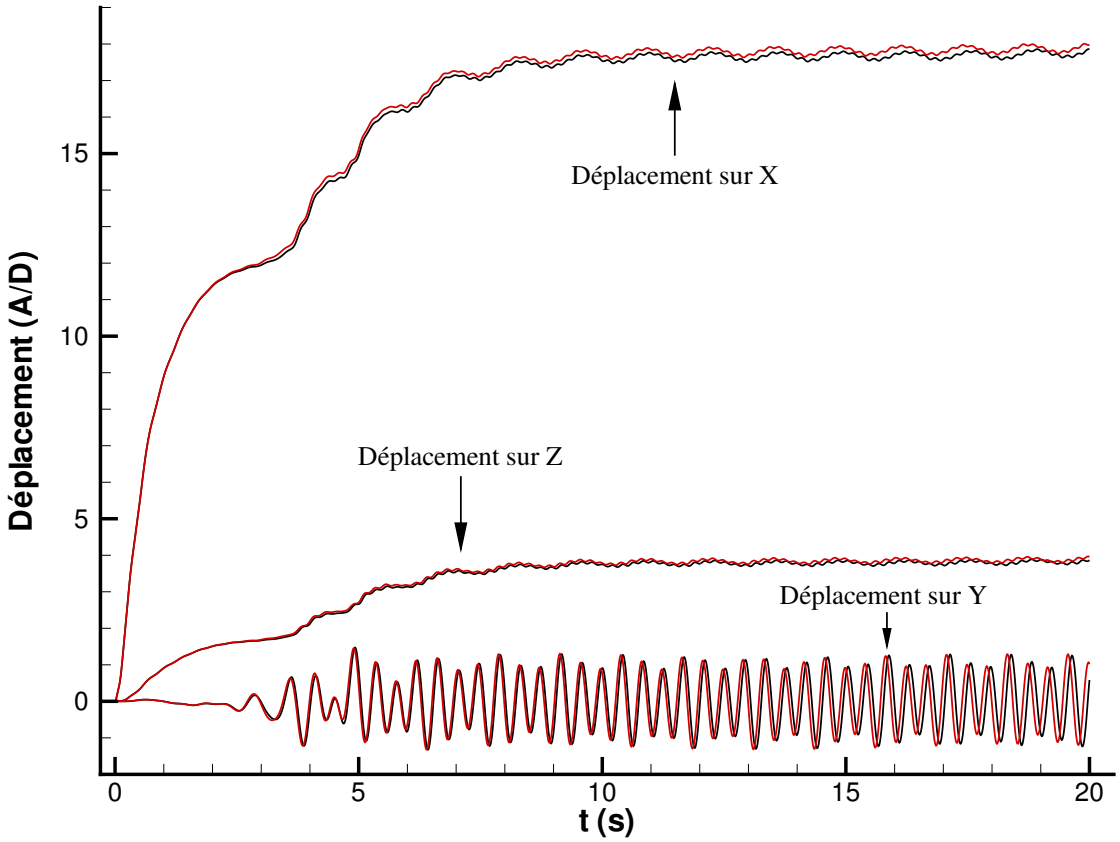


FIG. 6.25: Comparaison des déplacements adimensionnels entre le calcul monoprocesseur (en rouge) et le calcul parallèle (en noir)

6.2.2 Câble tracté dans un environnement multifluide

Comme dans le cas précédent, un câble déformable est plongé en partie dans l'air et en partie dans l'eau (cf. fig. 6.20). Mais cette fois-ci, l'extrémité supérieure du câble n'est plus fixe dans un environnement multifluide en mouvement, mais tractée dans un environnement au repos.

Le maillage utilisé est le même que précédemment (cf. fig. 6.21), car il permet des calculs 3D rapides. Les conditions du cas-test sont identiques aux précédentes (cf. tab. 6.5), sauf pour la vitesse des fluides qui est initialement nulle.

Longueur du câble	$L = 10\,m$
Diamètre du câble	$D = 0.2\,m$
Module d'Young du câble	$E = 3500\,\text{MPa}$
Masse volumique du câble	$\rho_s = 500\,kg.m^{-3}$
Masse volumique du fluide 1 (Eau)	$\rho_{eau} = 998.1\,kg.m^{-3}$
Viscosité dynamique (Eau)	$\mu_{eau} = 0.001\,Pa.s$
Masse volumique du fluide 2 (Air)	$\rho_{air} = 1.2\,kg.m^{-3}$
Viscosité dynamique (Air)	$\mu_{air} = 1.82 \times 10^{-5}\,Pa.s$
Vitesse des fluides à l'infini	$V_\infty^f = 0\,m.s^{-1}$

Tab. 6.5: Conditions du cas-test du câble tracté (unités SI)

Le temps physique est de $20\,s$ avec un pas de temps de $10^{-2}\,s$. Le temps de calcul s'élève à environ 16 heures (sur la même machine que pour le cas-test 6.2.1). Le modèle de turbulence utilisé est le $k-\omega\,SST$ de Menter.

Les extrémités du câble peuvent être contrôlées grâce à la méthode des multiplicateurs de Lagrange présentée à la section § 3.1.9. Dans notre cas, seule l'extrémité supérieure est imposée, l'autre étant libre. Plusieurs lois de mouvement ont été codées pour forcer le déplacement des extrémités. Celle utilisée ici se compose de trois phases :

- ➢ $t \in [0, t_1[$: vitesse imposée à V_1^s
- ➢ $t \in [t_1, t_2[$: phase d'accélération pour passer de V_1^s à V_2^s
- ➢ $t \in [t_2, +\infty[$: vitesse imposée à V_2^s

Pour l'exemple présenté, V_1^s et t_1 sont nuls. V_2^s et t_2 sont respectivement fixés à $2.5\,m.s^{-1}$ et $1\,s$. Le câble accélère donc pendant $1\,s$ avant d'atteindre sa vitesse nominale de $2.5\,m.s^{-1}$. La figure 6.26 montre sa position tous les $0.5\,s$. Le câble parcourt plus de $40\,m$, i.e. plus de 200 fois son diamètre.

Pour que la procédure de remaillage se passe correctement avec de tels déplacements, *le domaine de calcul subit globalement un mouvement en bloc* (cf. fig. 6.27). Pour ce faire un point de référence est nécessaire (en rouge sur la figure 6.28(a)). L'assemblage

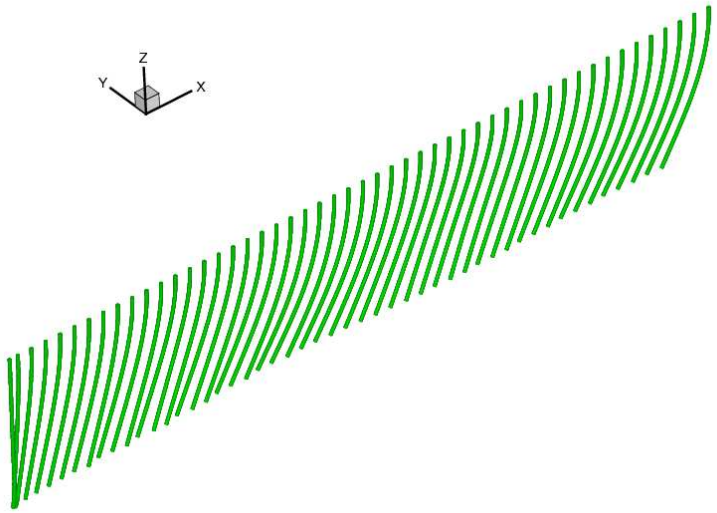


FIG. 6.26: Câble tracté dans un milieu multifluide (sondages tous les $0.5\,s$)

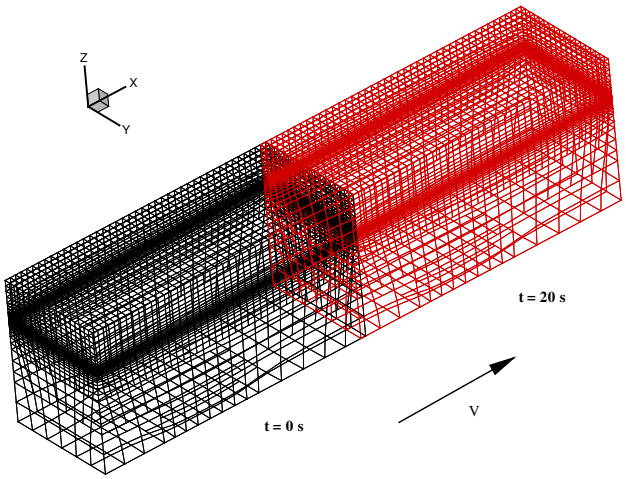
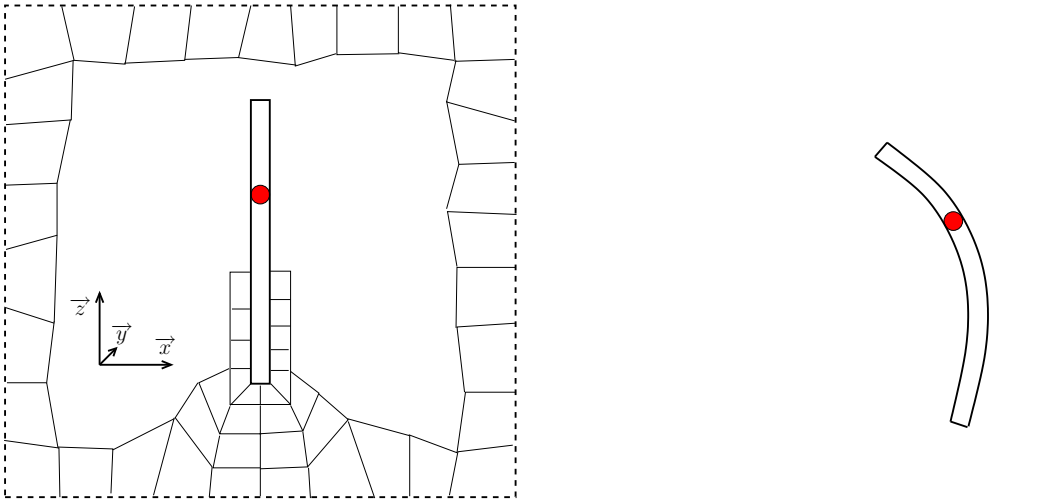


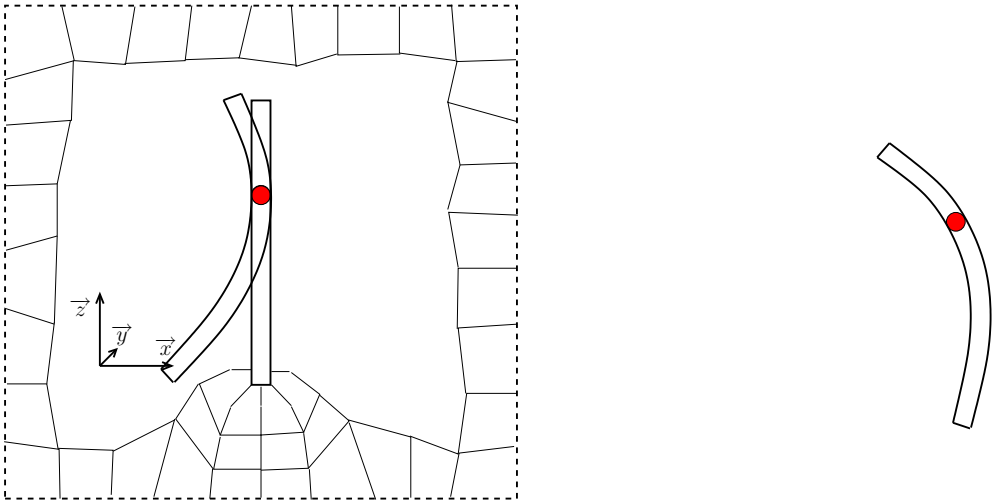
FIG. 6.27: Mouvement en bloc du domaine de calcul (domaine initial en noir ; domaine final en rouge)

des figures 6.28 explique comment coexistent les déplacements rigides et le processus de remaillage. L'algorithme général est le suivant :

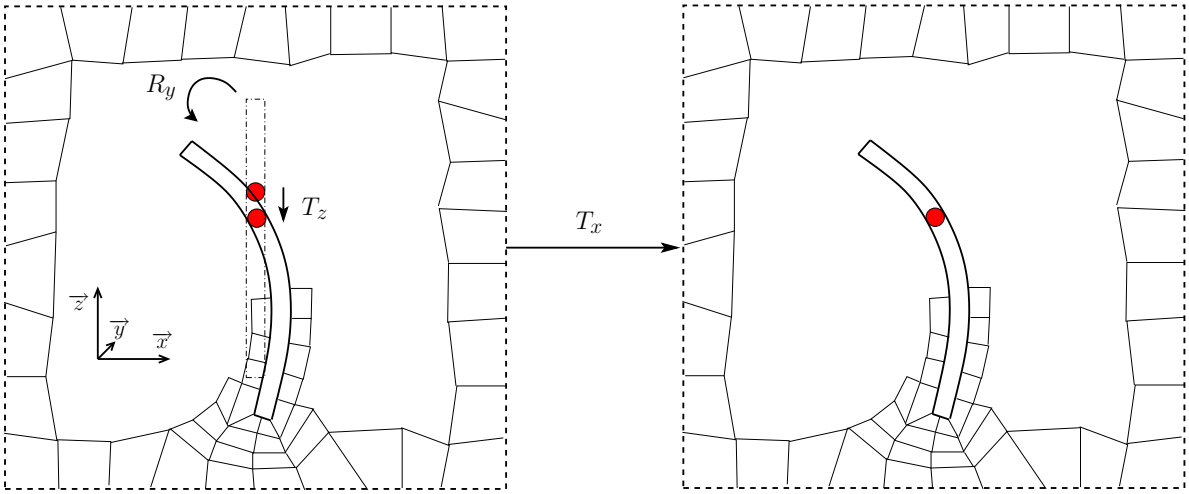
- ➢ Les données de départ sont la configuration initiale du maillage avec la poutre non-déformée et la position de la poutre à t_c (cf. fig. 6.28(a)).
- ➢ On déplace le point de référence (en rouge) de la poutre à t_c sur celui de la poutre non-déformée en prenant soin que les tangentes en ces deux points soient confondues (cf. fig. 6.28(b)).
- ➢ Après avoir choisi les degrés de liberté pris en compte par le remaillage (dans notre exemple T_z et R_y), le processus de remaillage opère. On termine la procédure en effectuant les mouvements rigides sur les degrés de liberté restants (dans notre exemple T_x) (cf. fig. 6.28(c)).



(a) Données fournies par le code poutre à t_c



(b) Transfert de la poutre courante dans le repère de référence



(c) Gestion des mouvements rigides, remaillage et transfert du maillage obtenu vers le repère courant

FIG. 6.28: Exemple de gestion du remaillage et des mouvements rigides

Pour accélérer le calcul IFS, on peut choisir de n'utiliser cette procédure qu'à la première itération IFS (dans le cas fluide seul, à la première itération non-linéaire). Pour les itérations IFS suivantes du même pas de temps, les positions des noeuds du maillage sont réajustées via la méthode de remaillage par pondération.

Notons que dans le cas-test du câble tracté, le point de référence choisi est l'extrémité supérieure, dont le mouvement est imposé. Les degrés de liberté qui sont gérés par mouvement en bloc sont les translations suivant $\overrightarrow{x}$ (T_x) et suivant $\overrightarrow{y}$ (T_y). Les autres sont traitées par le remaillage.

Ce calcul doit théoriquement donner les mêmes résultats que ceux présentés à la section §6.2.1. En effet, il n'y a aucune différence entre tracter un câble à une vitesse uniforme $\overrightarrow{V}$ dans un fluide au repos et soumettre un même câble à un fluide se déplaçant à cette même vitesse $\overrightarrow{V}$. On compare donc les déplacements suivant $\overrightarrow{y}$ et $\overrightarrow{z}$ (cf. fig. 6.29). Les courbes ne sont pas comparables pendant la phase d'accélération. On attend que le régime soit devenu périodique ($t \geqslant 8\,s$). Le déplacement suivant $\overrightarrow{y}$ est quasi-identique à celui trouvé précédemment. La fréquence des oscillations du câble est la même. Le déplacement suivant $\overrightarrow{z}$ est en phase avec le précédent et du même ordre de grandeur. Les faibles différences entre les courbes peuvent s'expliquer de manière numérique : bien que le problème physique soit identique (mise à part la phase transitoire où dans un cas le câble subit une accélération et voit un fluide au repos), le fait d'avoir un maillage mobile dans un écoulement au repos modifie le traitement numérique par rapport au cas précédent, notamment dans la formation de l'équation de pression. Ajoutons à ceci que le maillage étant très grossier, les erreurs commises par le calcul sont donc plus importantes.

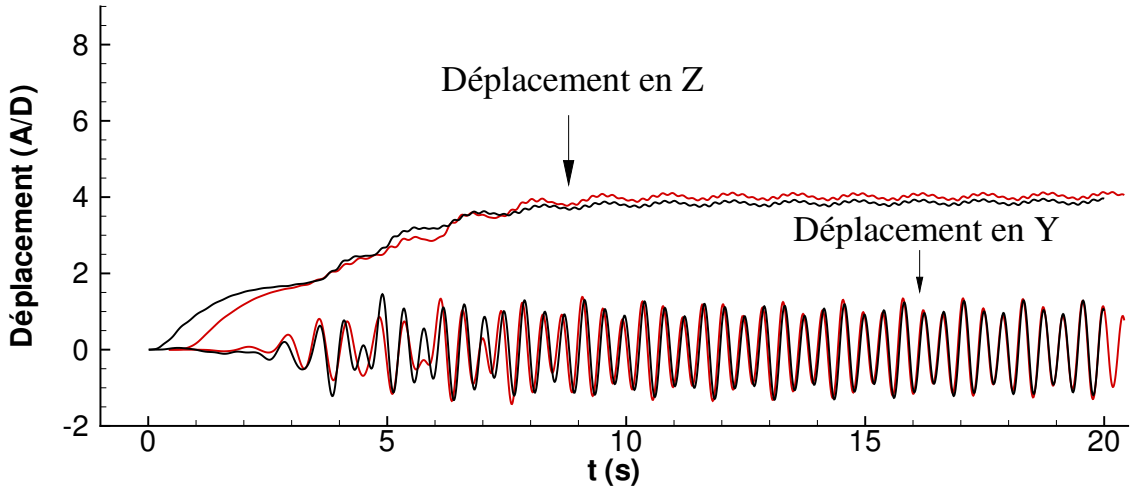


FIG. 6.29: Comparaison des déplacements adimensionnels suivant $\overrightarrow{y}$ et $\overrightarrow{z}$ (en rouge) avec ceux obtenus dans la section §6.2.1 (en noir)

Cette application met en évidence de nouvelles possibilités offertes par le code : dans les expériences réalisées en bassin, les structures élancées sont souvent déplacées dans un environnement fluide au repos. On peut donc réaliser exactement la même cinématique. Toutes sortes de mouvements peuvent être imposées (comme une loi sinusoïdale), on peut alors étudier la réponse du câble plongé dans l'eau et ainsi excité.

6.2.3 Riser : VIV

Pour valider le code IFS, une comparaison avec des données expérimentales est nécessaire. Après recherches, le cas-test mené par J.R. Chaplin en 2004 (cf. [22], [21]) a été choisi. Il s'agit d'un riser soumis à une tension verticale et subissant un courant sur 45% de sa longueur. Le but de cette expérience est de permettre de mieux comprendre le phénomène VIV pour les risers et de fournir une base de données[1] solide pour le développement de modèles de prédiction.

L'expérience s'est déroulée dans le bassin "Delta Flume"[2] du "Delft Hydraulics". Ce dernier est long de $230\,m$, large de $5\,m$ et a une profondeur d'eau de $6.5\,m$. Le riser fait $13.12\,m$ de hauteur pour un diamètre de $0.028\,m$, ce qui fait un rapport hauteur sur diamètre d'environ 470. Divers instruments de mesures (accéléromètres, jauges de contraintes...) y sont intégrés (détails dans [22]). Les propriétés du tube sont présentées dans le tableau 6.6.

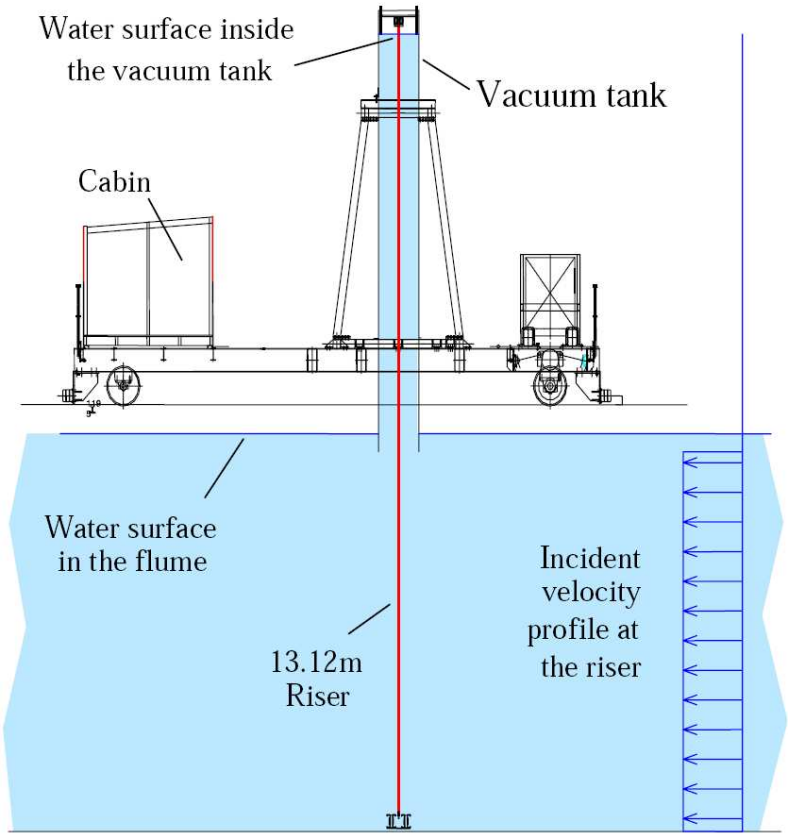
Diamètre extérieur	$D = 0.028\,m$
Longueur entre ses extrémités	$L = 13.12\,m$
Masse linéique	$\lambda_s = 1.47\,kg.m^{-1}$
Masse linéique (incluant l'eau interne une fois submergé)	$\lambda'_s = 1.85\,kg.m^{-1}$
Poids submergé linéique	$12.1\,N.m^{-1}$
Raideur de flexion	$EI = 29.9\,N.m^2$
Raideur de traction-compression	$EA = 5880000\,N$
Amortissement structurel	0.33% de l'amortissement critique
État de la surface	fluoroplastique très lisse (FEP)
Raideur du ressort du système de précontrainte	$T = 11.4\,kN.m^{-1}$ ou $T = 38.1\,kN.m^{-1}$

Tab. 6.6: Propriétés physiques du riser utilisé par J.R. Chaplin et son équipe (unités SI)

Les extrémités du riser sont reliées à un dispositif permettant de le déplacer le long du bassin (vitesse maximum : $1\,m.s^{-1}$). Celui-ci est décrit sur les figures 6.30. Le "vacuum tank" mesure $0.8\,m$ par $0.3\,m$ pour $7.54\,m$ de haut. Il accueille en son sein le riser. Durant l'expérience, il est généralement entièrement rempli d'eau. L'extrémité supérieure du riser y est fixée par l'intermédiaire d'un système de précontrainte. L'extrémité inférieure est reliée à un cadre bougeant en bloc avec le "vacuum tank".

Cette campagne de mesures fournit de nombreux cas-tests (cf. tab. 6.7) couvrant une plage de nombres de Reynolds allant d'environ 2500 à 30000, associés à différentes tensions. Sur le site internet[1], on trouve les mesures obtenues pour les différentes configurations. Onze codes de calcul développés par différents laboratoires ou entreprises ont été testés :

[1] http://www.civil.soton.ac.uk/hydraulics/riser
[2] http://www.wldelft.nl/facil/delta



(a) Schéma (la structure supportant le bas du riser n'est pas dessinée)

(b) Photo du "vacuum tank"

FIG. 6.30: Présentation de l'expérience (images extraites de [22])

- ➢ 4 codes CFD utilisant la méthode du découpage de l'écoulement en plans horizontaux (Norsk Hydro, USP, DeepFlow et VIVIC).
- ➢ 2 codes basés aussi sur l'hypothèse d'un écoulement fluide 2D par tranches, mais calculant la force exercée sur le riser avec des méthodes empiriques (Orcina Vortex Tracking et Orcina Wake Oscillator).
- ➢ 5 codes (VIVA, VIVANA, VICoMo, SHEAR7 et ABAVIV) d'un genre différent : ils sont fondés sur des mesures faites sur cylindres rigides excités et permettent d'identifier l'amplitude du mode dans lequel le riser va vibrer.

Les résultats sont en ligne[1] et ont été discutés dans [21].

Numéro du cas-test	Vitesse du "vacuum tank" ($m.s^{-1}$)	Tension au sommet (N)
1	0.16	405
2	0.21	407
3	0.31	457
4	0.40	506
5	0.54	598
6	0.60	670
7	0.70	743
8	0.85	923
9	0.95	1002
10	0.11	803
11	0.41	840
12	0.51	872
13	0.21	1916
14	0.40	1926
15	0.71	2018

TAB. 6.7: Liste des cas-tests réalisés par J.R. Chaplin et son équipe [21]

Pour débuter la validation tridimensionnelle du code IFS, le cas-test numéro 1 a été sélectionné. La faible vitesse ($V_x = -0.16\,m.s^{-1}$) assure des déformations raisonnables du riser, ce qui est un bon point de départ. De plus, il a été traité par tous les autres codes numériques. Une première modélisation de l'expérience a été réalisée (cf. fig. 6.31). Le domaine de calcul fluide se compose de deux "boites" : l'une représentant le "vacuum tank", l'autre une partie du bassin "Delta Flume". En première approximation, on a décidé de ne pas mettre d'écoulement avec surface libre. Les hauteurs des boites ont alors été calculées par rapport à la hauteur du riser qui subit l'écoulement en mouvement. La boite représentant le "Delta Flume" fait donc 45% de la longueur du tube en hauteur. La modélisation du "vacuum tank" s'élève par conséquent à 55% de la longueur du riser.

Dans l'expérience, le tube est tracté dans de l'eau au repos, afin d'obtenir un profil uniforme de vitesse sur sa partie inférieure. Dans le cadre d'une simulation numérique, le choix d'un domaine de calcul fixe avec un fluide en mouvement est aussi possible, les deux

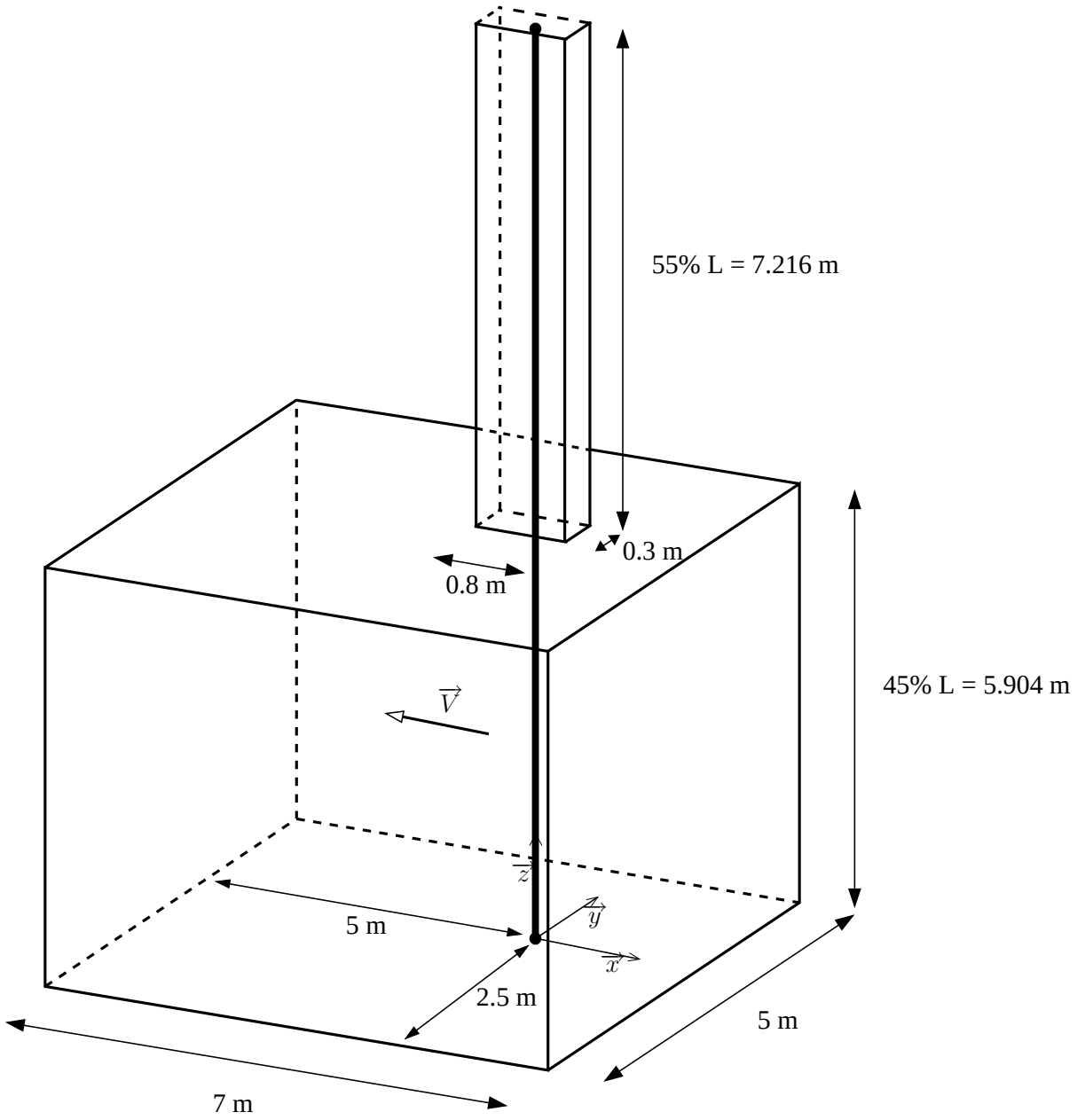


FIG. 6.31: Schéma du domaine de calcul

problèmes étant équivalents physiquement. Dans les simulations qui suivent, ce dernier choix est retenu.

Le système de précontrainte situé à l'extrémité supérieure du tube *n'a pas été modélisé*. Dans notre modélisation le riser est donc directement fixé au "vacuum tank" et la position de son extrémité ne varie pas dans le temps. Ne disposant que de liaisons encastrement dans le code IFS développé, ces dernières sont utilisées, dans un premier temps, aux deux extrémités.

Un maillage a alors été généré avec HEXPRESS™ (cf. fig. 6.32). Il contient environ 650000 cellules pour 490000 noeuds. Il a été raffiné près du riser pour tenir compte de la distance à la paroi y^+, car un modèle de loi de paroi va être utilisé pour la turbulence autour du câble. Sur les autres frontières solides (côtés du "Delta Flume" et du "vacuum tank") une condition de glissement est définie. À l'entrée, la vitesse est imposée. Enfin, pour les deux dernières frontières (en sortie et sur le dessus de la boite modélisant le "Delta Flume"), une condition de pression est employée.

Les liaisons encastrement évitent que la poutre soit libre de tourner sur elle-même, le modèle d'Euler-Bernoulli est donc utilisable. Un calcul est lancé avec ce modèle et un autre avec celui de Rayleigh à titre de comparaison. Les calculs sont réalisés en parallèle sur 4 processeurs. Le temps de simulation est de $60\,s$ avec un pas de temps de $\Delta t = 10^{-2}\,s$ et le coût CPU de 7 jours.

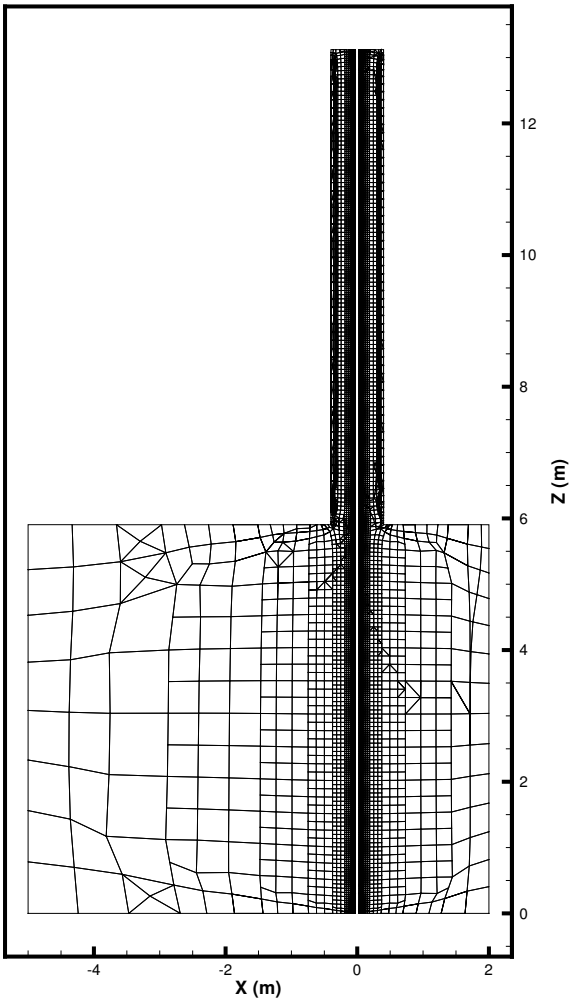
La figure 6.33 montre une comparaison de la déformée moyenne du riser (moyenne sur $60\,s$) dans la direction de l'écoulement entre les données expérimentales et les prédictions numériques. La courbe est présentée sous forme *adimensionnelle* : l'amplitude des déplacements suivant $\overrightarrow{x}$ est divisée par le diamètre du câble, tandis que l'on ramène la longueur du riser à 1. On observe tout d'abord que *le choix du modèle de poutre ne modifie pas de manière substantielle les résultats*. On note ensuite que *l'utilisation de liaisons encastrement* est un *gros problème de modélisation* : la tangente aux extrémités étant imposée verticale, la déformée prédite ne correspond pas aux mesures expérimentales.

Le développement de la prise en compte de nouvelles liaisons a donc été nécessaire. Les tangentes aux extrémités du câble doivent être libres dans ce cas-test. Deux liaisons ont alors été étudiées :

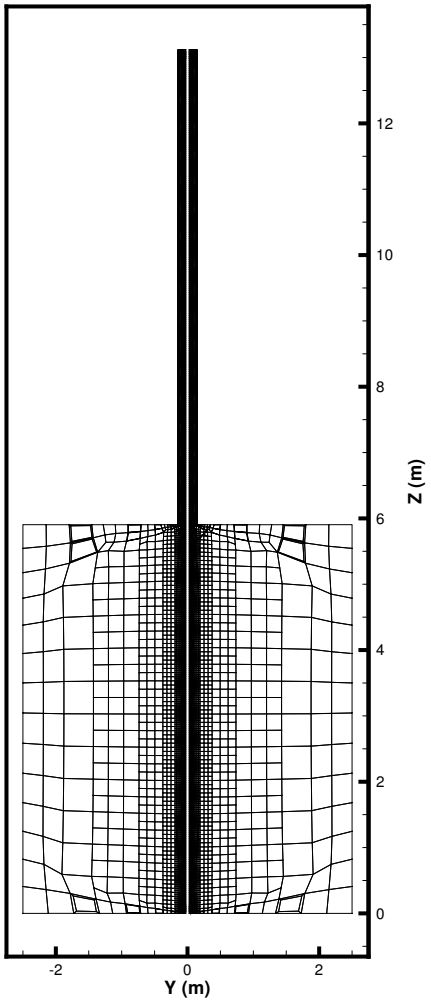
- La liaison **rotule** qui impose la position de l'extrémité, mais qui la laisse libre en rotation.
- La liaison **double pivot** (les deux pivots ont leurs axes perpendiculaires) qui permet d'imposer la position de l'extrémité, en empêchant la poutre de tourner sur elle-même, mais en laissant les tangentes libres.

Le problème des mouvements imposés aux extrémités abordé à la section § 3.1.9 est repris en prenant un vecteur "contraintes géométriques" 3.66 dont la dimension est diminuée : pour la liaison rotule, seule la position est considérée ; pour la liaison double pivot, la position et l'angle de torsion sont imposés.

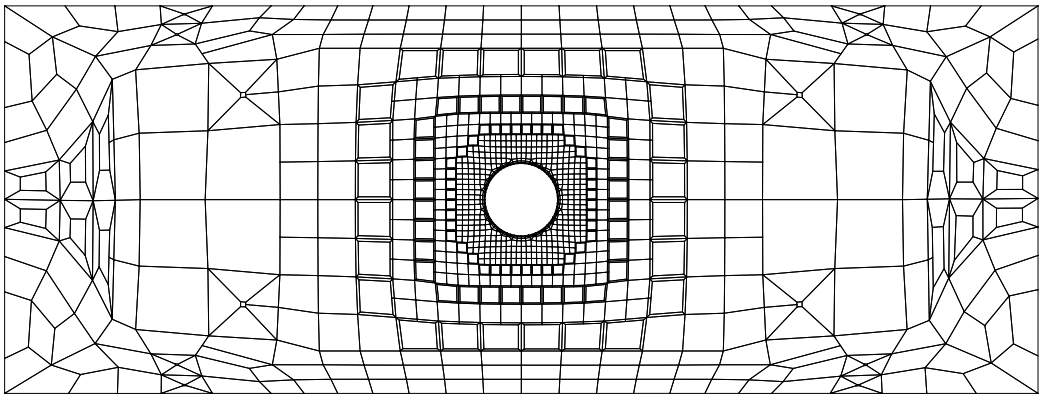
Un nouveau maillage est généré avec HEXPRESS™ (cf. fig. 6.34) pour que les liaisons rotule ou double pivot puissent être employées : *les extrémités du câble ne font plus partie*



(a) Coupe suivant le plan y=0



(b) Coupe suivant le plan x=0



(c) Coupe suivant le plan $z = 10\,m$

FIG. 6.32: Maillage du domaine de calcul pour le cas-test de J.R. Chaplin et de son équipe

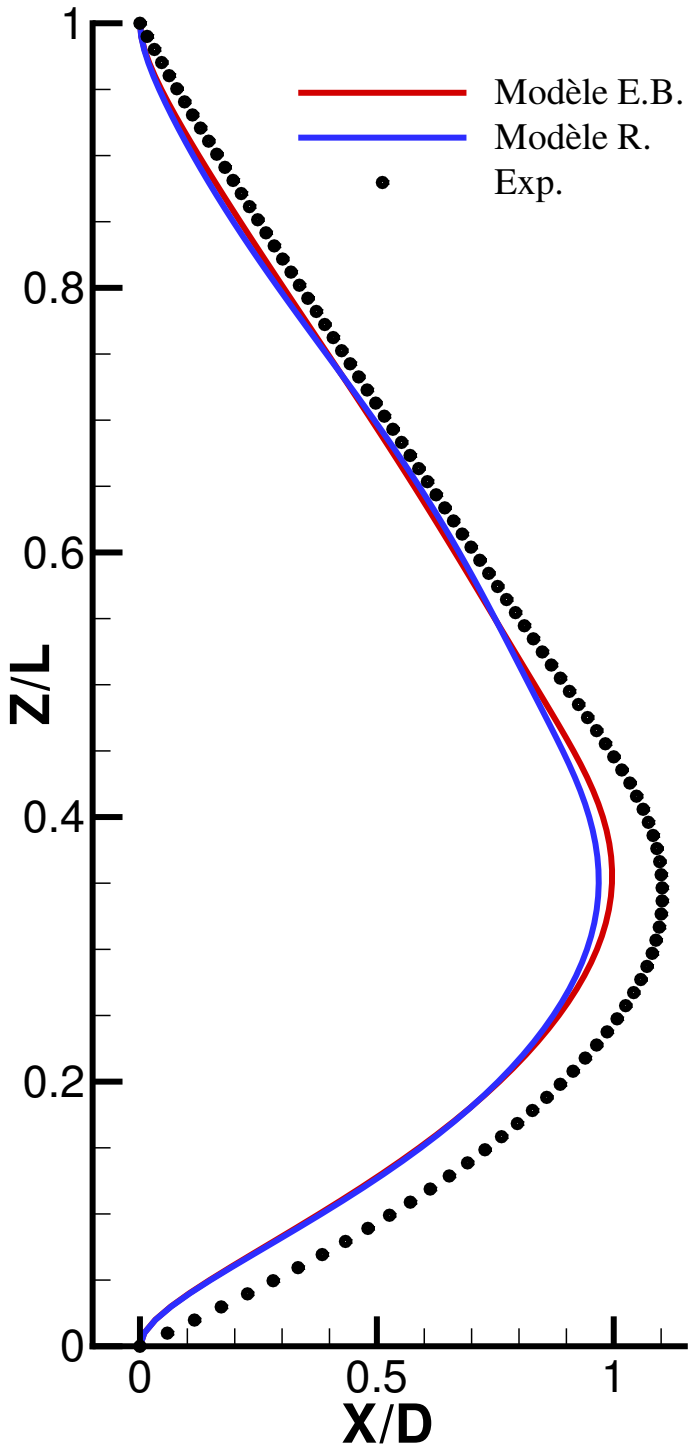


FIG. 6.33: Comparaison de la déformée moyenne du riser dans la direction de l'écoulement (extrémités encastrées) (modèle Euler-Bernoulli en rouge ; modèle de Rayleigh en bleu ; mesures expérimentales en noir)

des frontières extérieures (cf. fig. 6.34(c) et 6.34(d)). Il contient 670000 cellules et 520000 noeuds. Le même calcul que précédemment est alors lancé en utilisant ce nouveau maillage et le type de liaison double pivot. Ainsi, les deux modèles de poutre peuvent être testés, car la poutre n'est pas libre de tourner sur elle-même. Comme avant, la simulation est faite en parallèle sur 4 processeurs. Le temps de simulation est de $60\,s$ avec un pas de temps $\Delta t = 10^{-2}\,s$ et le coût CPU de 7 jours.

Si l'on trace comme précédemment la déformée moyenne du riser (moyenne sur $60\,s$) (cf. fig. 6.35) dans la direction de l'écoulement obtenue avec cette nouvelle modélisation, on voit une nette amélioration : dans la partie supérieure et inférieure du montage, la déformée prédite correspond *exactement* à l'expérience. Notons que les deux modèles de poutre donnent encore une fois des résultats très proches. Le maximum de l'erreur pour les deux modèles (Rayleigh et Euler-Bernoulli) est de 6 % et se situe au tiers de la poutre (en partant du bas).

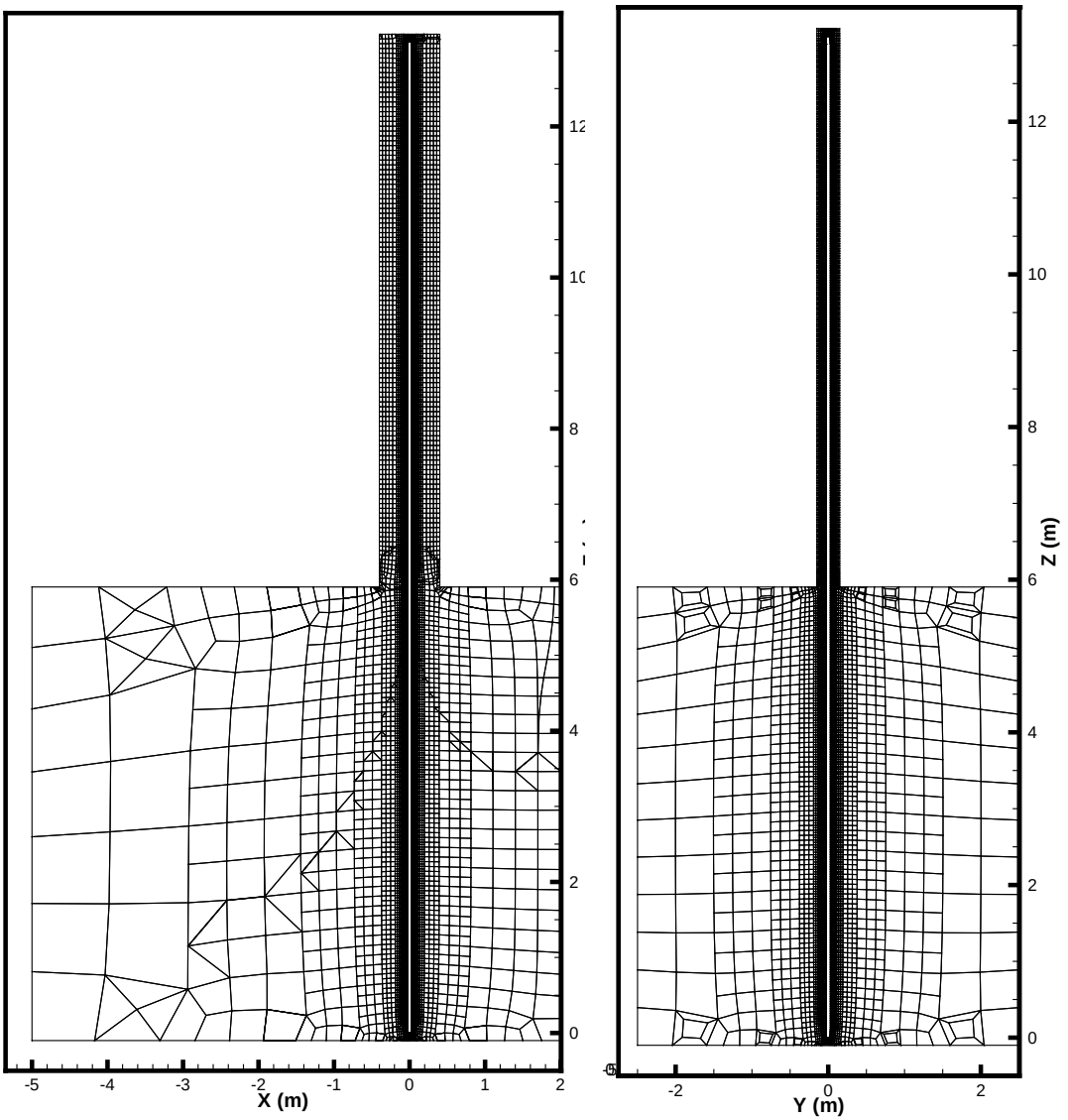
À la vue de ces bons résultats sur la déformée moyenne, cette modélisation peut être considérée comme suffisante dans un premier temps. On peut alors se pencher sur le comportement du riser lors de ce test. Nous avons à notre disposition l'enveloppe des déplacements expérimentaux du câble dans le sens de l'écoulement et perpendiculairement à ce dernier. La figure 6.36(a) compare l'enveloppe des déplacements du câble prédite par le calcul (pour $t \in [40\,s, 60\,s]$ et pour le modèle d'Euler-Bernoulli) dans le sens de l'écoulement avec celle observée expérimentalement. Le mode dominant d'oscillations du câble (le mode 4) est bien trouvé par le code IFS. Par contre l'amplitude des oscillations est environ deux fois plus importante. La figure 6.36(b) montre l'enveloppe des déplacements perpendiculaires à l'écoulement du câble, estimée par l'ordinateur (pour $t \in [40\,s, 60\,s]$ et pour le modèle d'Euler-Bernoulli), avec celle expérimentale. Le mode dominant de vibration du câble (le mode 2) est correctement reproduit par le programme IFS. Cependant, comme précédemment, l'amplitude du signal est environ deux fois plus grande.

L'étape suivante est de savoir comment se place le code IFS développé par rapport aux autres programmes de calcul qui ont été testés sur cette même configuration. Ainsi, les forces et les faiblesses de la méthode pourront être soulignées. On récupère les résultats des calculs numériques effectués par les autres codes sur le site de l'expérience[3] (pas de données pour le code Deepflow) et on effectue un comparatif des différentes prédictions.

Tout d'abord, intéressons-nous à la déformée moyenne du riser dans la direction de l'écoulement. La figure 6.37 compare six codes extérieurs (ceux dont les données contiennent la déformée moyenne) avec l'expérience et **ISIS**. Un simple coup d'oeil permet de se rendre compte que les résultats obtenus avec le code IFS basé sur **ISIS** sont de loin les plus proches de la réalité. Parmi les codes extérieurs, le programme ABAVIV donne la déformée moyenne la plus réaliste, mais l'erreur maximale est de 18 %, alors que dans notre cas elle n'est que de 6 %.

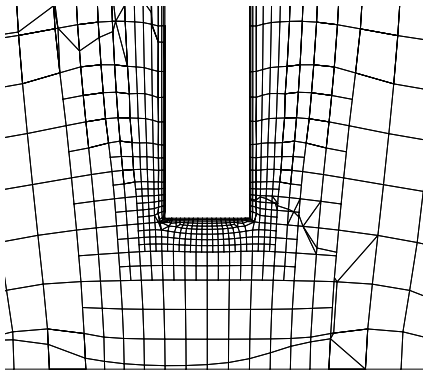
L'ensemble des figures 6.38 compare l'enveloppe des déplacements du riser dans le sens de l'écoulement prédite par **ISIS** avec quatre autres codes (ceux dont les données le permettent) et avec l'expérience. On voit que parmi les codes de la famille CFD (Norsk Hydro, USP et VIVIC), **ISIS** est le seul à prédire le mode dominant (mode 4). Le code

[3] http://www.civil.soton.ac.uk/hydraulics/riser

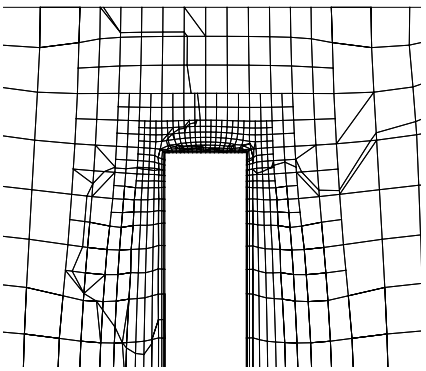


(a) Coupe suivant le plan y=0

(b) Coupe suivant le plan x=0



(c) Zoom sur l'extrémité inférieure du riser (coupe suivant le plan y=0



(d) Zoom sur l'extrémité supérieure du riser (coupe suivant le plan y=0

FIG. 6.34: Maillage du domaine de calcul pour les liaisons rotule et double pivot

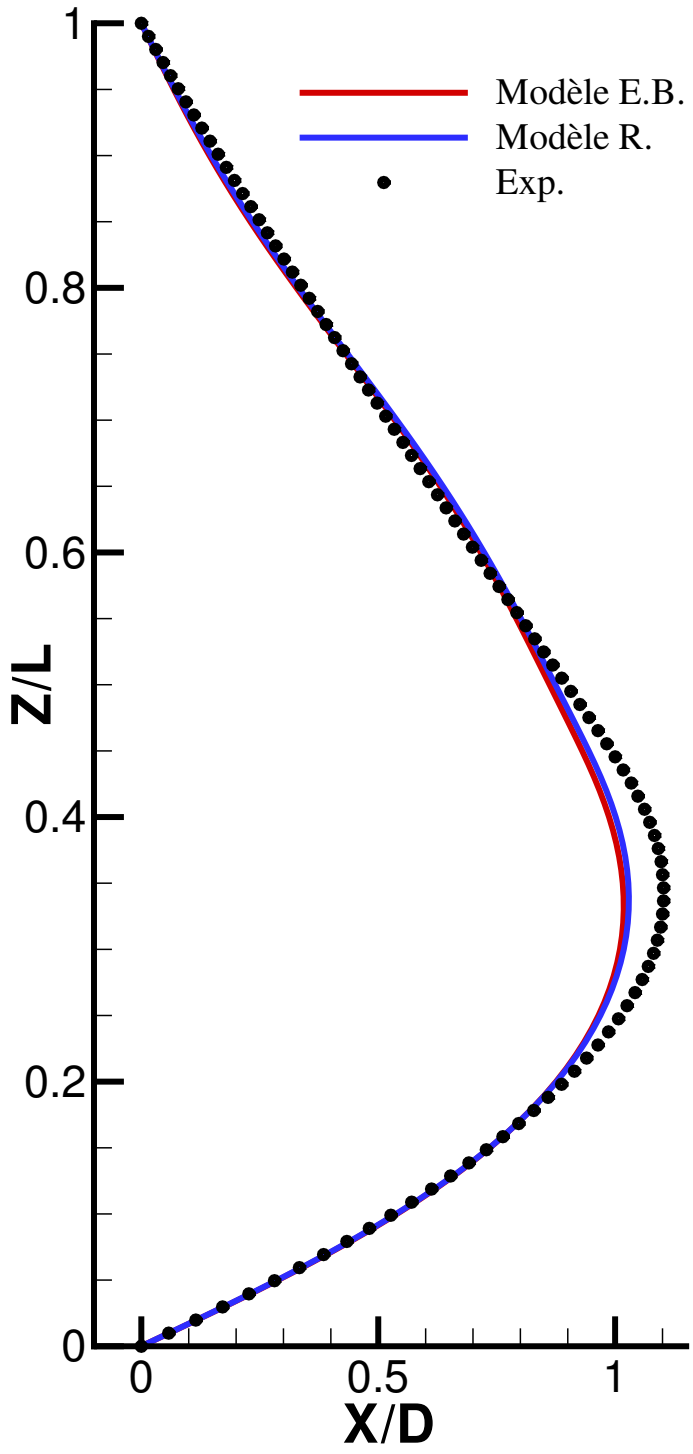
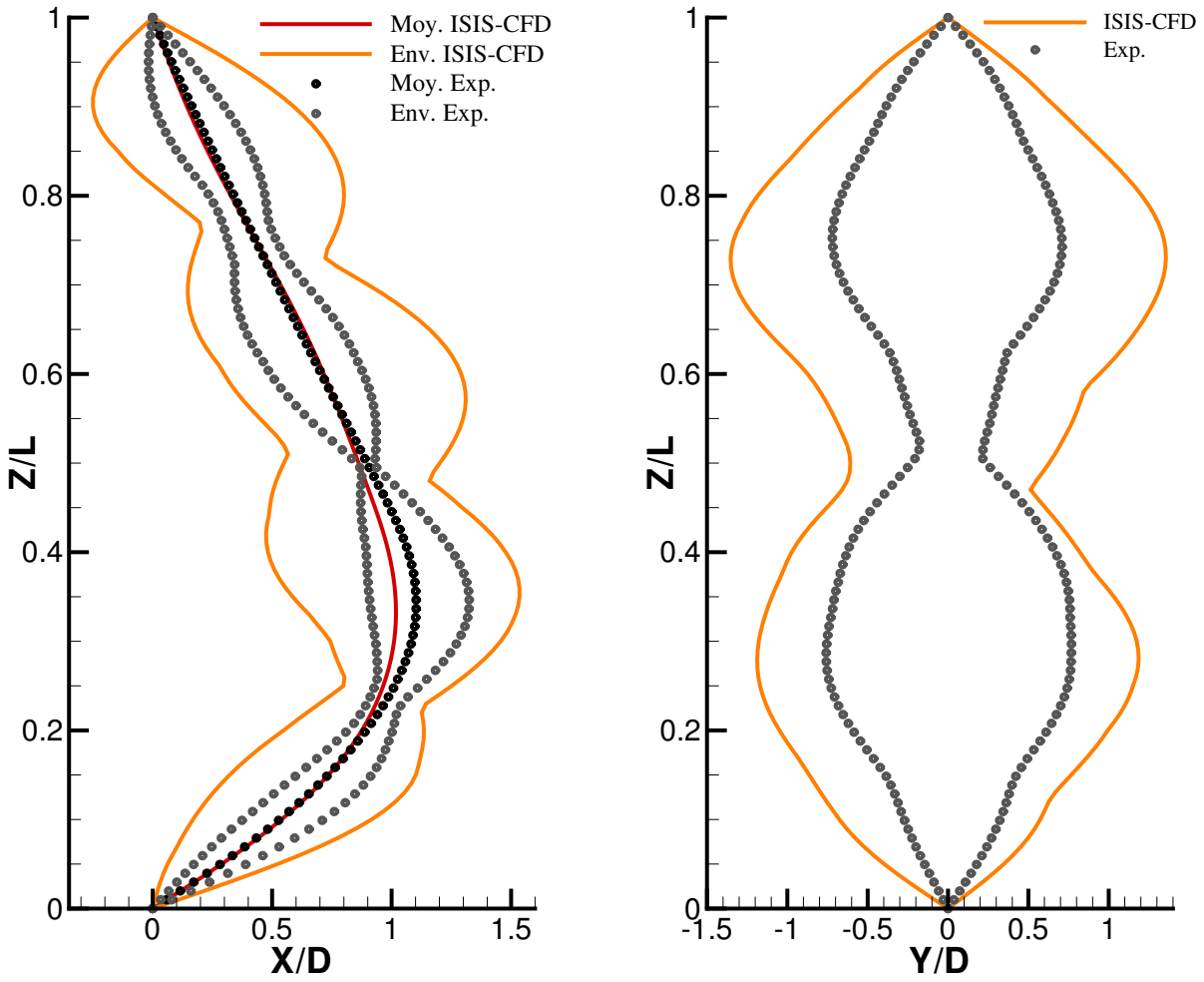


FIG. 6.35: Comparaison de la déformée moyenne du riser dans la direction de l'écoulement (liaisons double pivot) (modèle Euler-Bernoulli en rouge ; modèle de Rayleigh en bleu ; mesures expérimentales en noir)



(a) Comparaison entre l'enveloppe prédite des déplacements du riser dans le plan $(\overrightarrow{x}, \overrightarrow{z})$ (en orange) et celle observée expérimentalement (en gris)

(b) Comparaison entre l'enveloppe prédite des déplacements du riser dans le plan $(\overrightarrow{y}, \overrightarrow{z})$ (en orange) et celle observée expérimentalement (en gris)

FIG. 6.36: Comparaison de la prédiction numérique avec l'expérience dans le cas de la modélisation avec des liaisons double pivot

Orcina V.T. trouve un mode de vibration du câble supérieur (mode 5) avec des amplitudes plus importantes que dans la réalité.

Les graphiques 6.39 montrent côte à côte les prédictions de chaque famille de codes avec celles d'**ISIS** et l'expérience. Parmi les codes CFD, Norsk Hydro et VIVIC donnent de bons résultats : le mode dominant (mode 2) est bien trouvé et les amplitudes des oscillations aussi. **ISIS**, lui, surestime cette amplitude. Dans la famille Orcina, Orcina Milan donne les meilleures prédictions. Orcina V.T. détecte un mode de vibration supérieur (mode 3). Les codes "empiriques" (ABAVIV, VIVANA, VICoMo et SHEAR7) sont faits pour évaluer le comportement des câbles dans un plan perpendiculaire à l'écoulement : ils trouvent donc bien le mode dominant (mode 2) et estiment correctement les amplitudes.

Pour résumer, **ISIS** donne la meilleure déformée moyenne dans le sens de l'écoulement tous codes confondus et trouve les bons modes de vibration du riser. Par contre, il surestime les amplitudes de ces oscillations et par conséquent donne des résultats moins bons que les programmes prévus à cet effet (ABAVIV, VIVANA, VICoMo et SHEAR7).

Les résultats obtenus avec **ISIS** sont corrects, mais ne sont pas parfaits. Comment peuvent-ils être améliorés ?

La première idée est de diminuer le pas de temps et de relancer les calculs. Mais avec $\Delta t = 5 \times 10^{-3}\, s$, les résultats sont très semblables. On en conclut donc qu'*un pas de temps de* $10^{-2}\, s$ *est suffisant* pour les simulations.

Le maillage peut aussi être en cause, car celui utilisé est relativement grossier dans le sillage. Cela pourrait avoir un effet sur la déformée moyenne du câble. Par contre, ça ne diminuera sans doute pas l'amplitude des oscillations.

La modélisation faite souffre d'une faiblesse : *le système de précontrainte n'a pas été modélisé*. Il a, d'après J.R. Chaplin, un effet important sur l'étirement du riser. L'erreur de 6 % observée sur la déformée moyenne dans le sens de l'écoulement pourrait être réduite.

Un autre problème de modélisation est à envisager : *le solveur poutre n'a pas d'amortissement structurel*. Ce dernier est pourtant bien présent dans l'expérience et joue un rôle au niveau des oscillations du câble. La prise en compte de ce phénomène physique pourrait permettre de réduire l'amplitude du déplacement du riser tant dans le sens de l'écoulement que perpendiculairement à celui-ci.

Enfin, vu que le nombre de Reynolds est de 4471, la turbulence et les phénomènes transitionnels jouent un rôle primordial. Pour un tel nombre de Reynolds, l'écoulement autour d'un cylindre est qualifié de **sous-critique** : le décollement de l'écoulement est laminaire, alors que les vortex en aval sont turbulents (cf. [69]). Ce régime *transitionnel* est *très difficile à modéliser numériquement sur le plan de la turbulence*. Le modèle utilisé est, rappelons-le, celui de Menter ($k - \omega\, SST$) et la condition aux limites sur le riser est de type loi de paroi. À ce propos, un calcul bas Reynolds serait préférable. De manière générale, on peut dire que la modélisation de la turbulence faite pour ce problème peut poser problème et entraîner une mauvaise estimation des efforts fluides sur le riser.

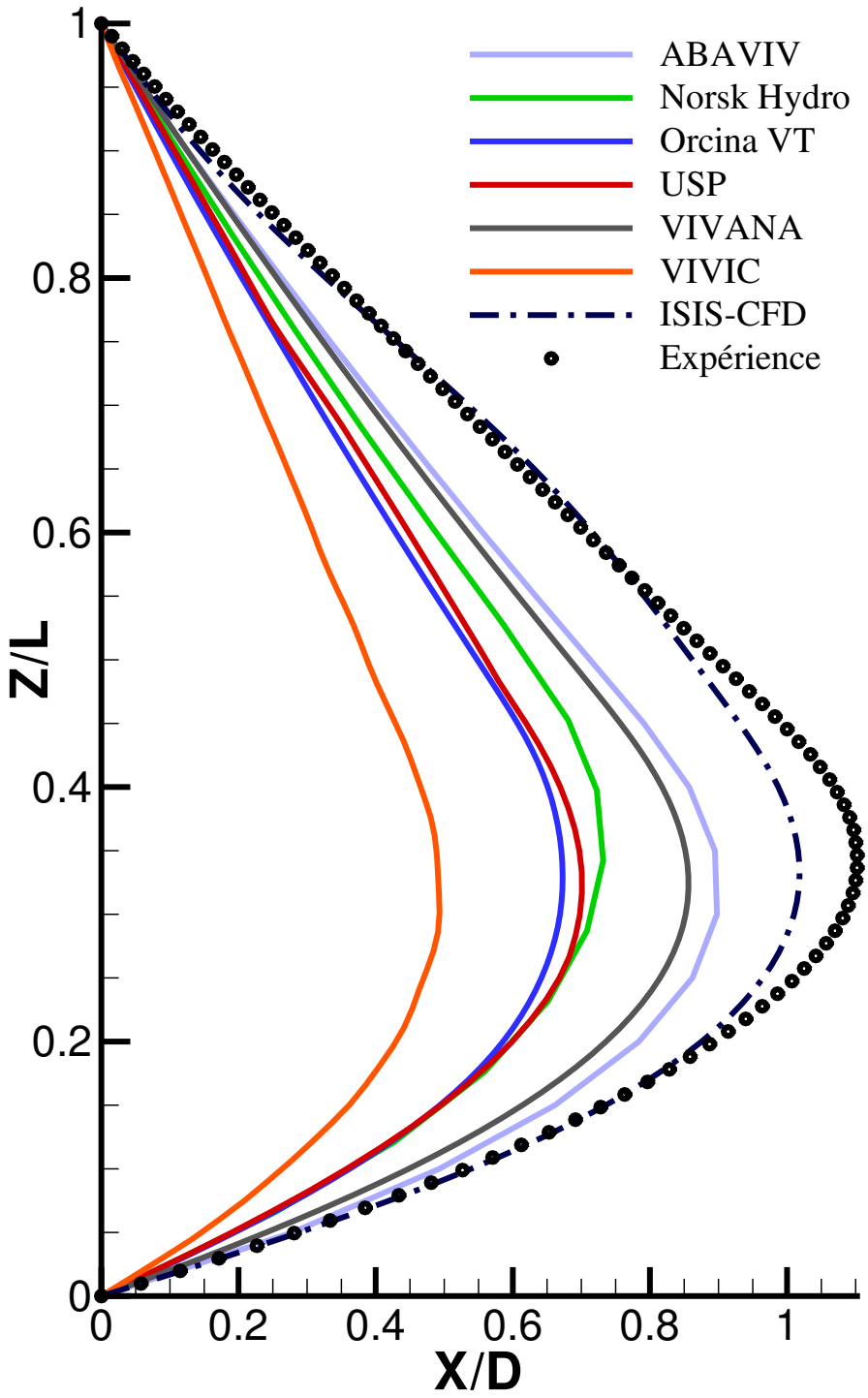
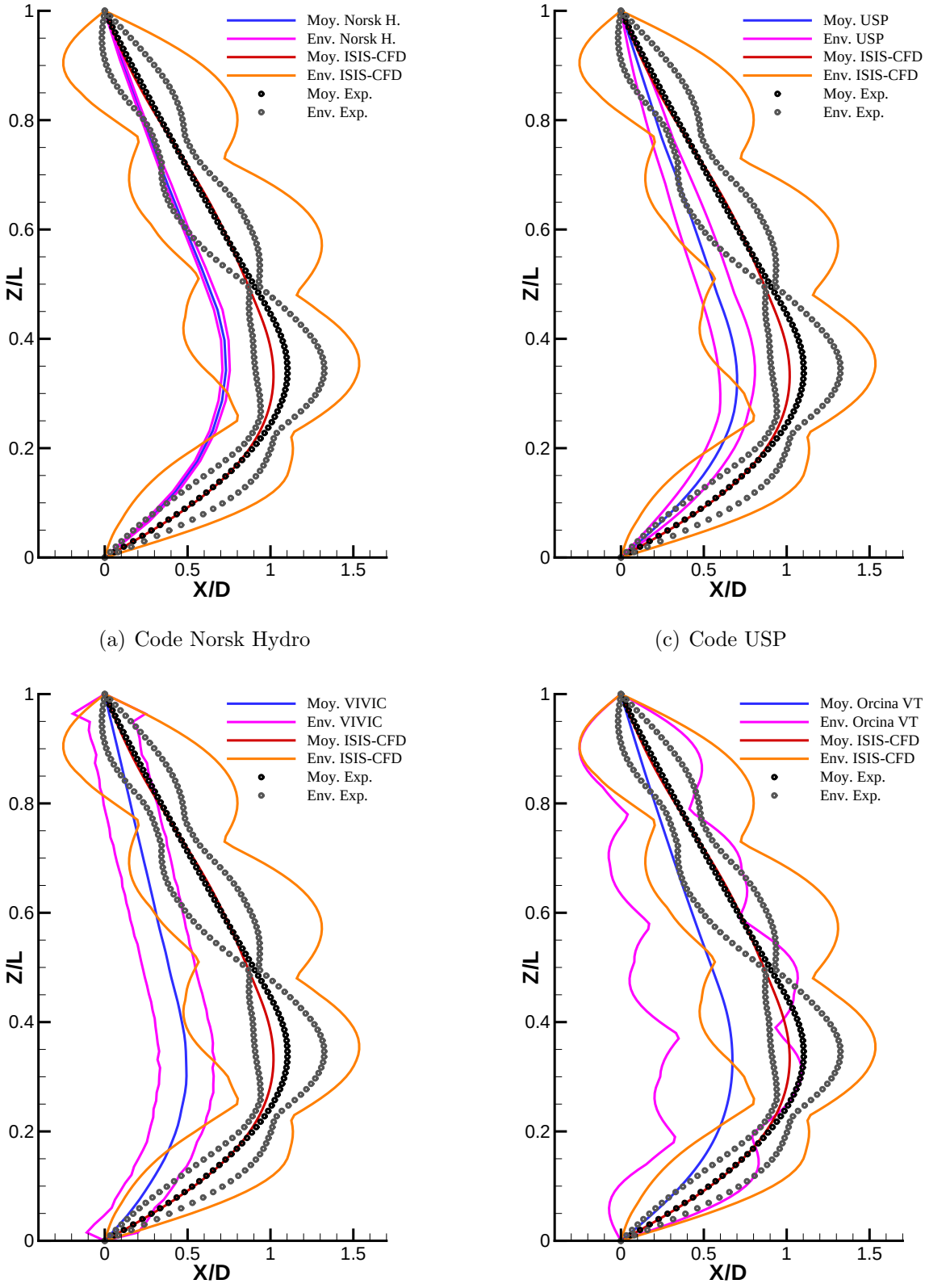


FIG. 6.37: Comparaison de la déformée moyenne du riser dans la direction de l'écoulement entre les différents codes numériques et les données expérimentales



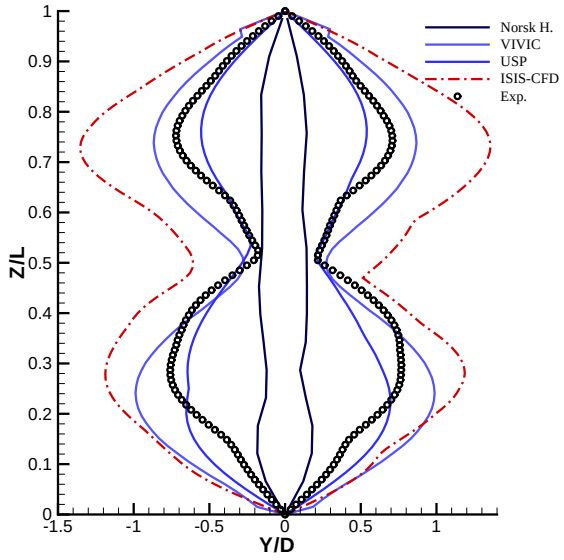
(a) Code Norsk Hydro

(c) Code USP

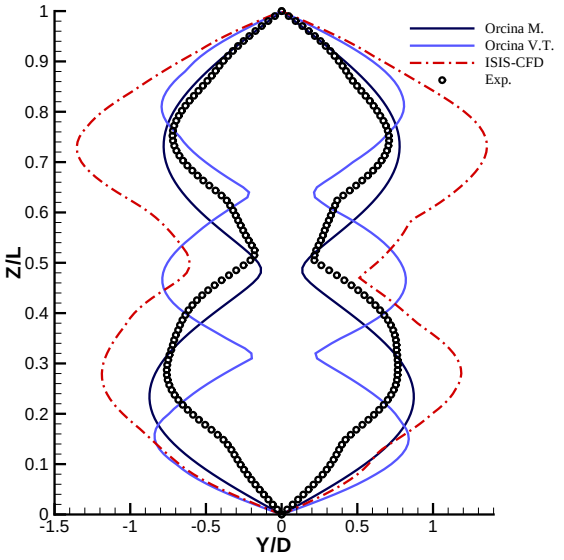
(b) Code VIVIC

(d) Code Orcina V.T.

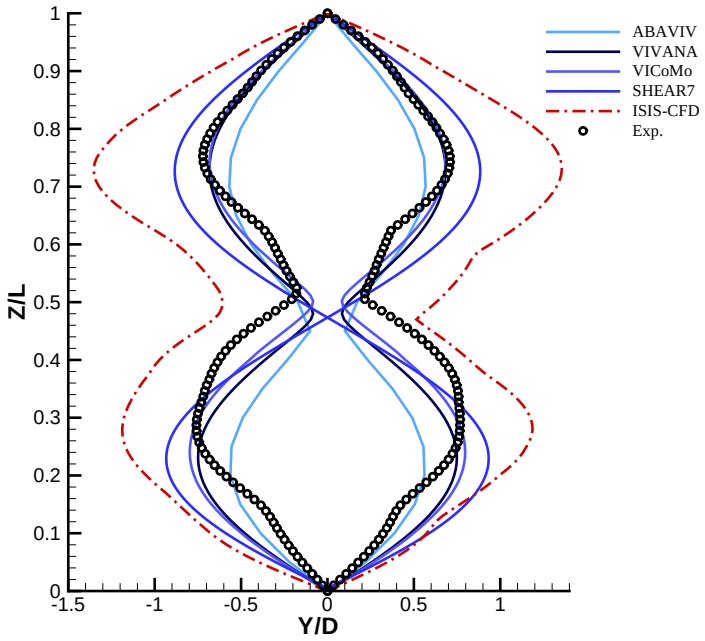
FIG. 6.38: Comparaison des différents codes numériques testés sur le cas-test 1 de Chaplin pour l'enveloppe des déplacements du riser dans le plan $(\overrightarrow{x}, \overrightarrow{z})$



(a) Comparaison entre la famille des codes CFD et ISIS-CFD



(b) Comparaison entre la famille des codes Orcina et ISIS-CFD



(c) Comparaison entre la famille des codes empiriques et ISIS-CFD

FIG. 6.39: Comparaison des différents codes numériques testés sur le cas-test 1 de Chaplin pour l'enveloppe des déplacements du riser dans le plan $(\overrightarrow{y}, \overrightarrow{z})$

6.3 Conclusion

Les quelques applications IFS réalisées ont montré les possibilités qu'offre le code IFS développé lors de cette thèse. En 2D, on a vérifié que le code en mode structure dynamique et calcul fluide instationnaire convergeait vers la solution obtenue de manière statique pour la structure et stationnaire pour le fluide. Puis, le cas-test très instationnaire de Hübner a permis de tester les limites du code tant au niveau remaillage, qu'au niveau du couplage fluide-structure. Durant cet exemple, le schéma standard de Newmark a dû être remplacé par la méthode de Bossak, qui apporte une dissipation numérique des hautes fréquences générées par le schéma standard, et ce, sans dégrader l'ordre de précision.

Deux cas-tests tridimensionnels ont mis à l'épreuve le code fluide-structure. On a vu que les grands déplacements d'un câble fixé à une extrémité et subissant un courant étaient bien supportés. La possibilité d'imposer des mouvements en bout de riser et de les prendre en compte par remaillage avec mouvement en bloc a aussi été intégrée.

Une application 3D se basant sur la campagne de mesures faites par J.R. Chaplin et son équipe en 2004 a été réalisée. Elle apporte une comparaison des résultats numériques avec l'expérience.

Conclusion générale et perspectives

À l'heure actuelle, les codes Navier-Stokes sont capables d'aborder des problèmes fluides à géométrie compliquée et à fort nombre de Reynolds. L'évolution des moyens de calcul permet d'envisager une modélisation fidèle de phénomènes physiques de plus en plus complexes. Cela conduit différentes disciplines à se rapprocher et à échanger leurs connaissances et techniques. Parmi ces domaines multidisciplinaires, la mécanique des interactions fluide-structure, communément appelée IFS, suscite un grand intérêt auprès de la communauté scientifique. Dès 2001, l'Équipe de Modélisation Numérique du laboratoire de Mécanique des Fluides de l'École Centrale Nantes a décidé d'en faire un de ses axes de recherche. Le but final étant de munir le solveur fluide, développé par le laboratoire et nommé **ISIS**, de modules permettant de résoudre toutes sortes de problèmes d'IFS.

La thématique IFS a débuté avec les travaux d'A. Leroyer sur la gestion des corps solides et déformables en mouvement imposé (cf. [57]). Cela a mené l'EMN à intégrer le projet *ROBEA/RAAMO*, qui a pour objectif de créer un robot anguille autonome et autopropulsé. Dans ce contexte, des simulations RANSE sont nécessaires pour valider la partie commande. Les échanges avec les mécaniciens des structures et des roboticiens, comme F. Boyer, ont attiré l'attention de l'équipe sur les possibilités des solveurs poutre en grand déplacement et sur celle de la théorie de Cosserat. En effet, cette dernière permet l'extension aux corps poutres de la technique de remaillage par pondération analytique développée initialement pour les corps solides. L'idée de coupler un tel programme avec **ISIS** s'est imposée d'elle-même.

Le développement d'un code structure complet 2D/3D a été écarté, laissant cela aux équipes spécialisées comme le GEM[1]. Le couplage de deux codes complets, via échange de fichiers de données ou encore via des bibliothèques, a été pour l'instant mis de côté. L'utilisation de modules plus réduits, comme un solveur monodimensionnel de type poutre, permet une gestion plus souple et une intégration complète dans **ISIS**. Ainsi, cette thèse a vu le jour sous le nom d'*Interaction Fluide-Structure pour les corps élancés*.

* * *

Une recherche non-exhaustive des travaux réalisés en IFS a permis d'avoir une vue générale des outils et méthodes utilisés. Les différents algorithmes de couplage ont été analysés et évalués. Ne pouvant nous tourner vers un code monolithique, trop rigide, l'algorithme explicite a été codé pour la validation du solveur structure. Mais ce dernier, trop peu robuste dans le cadre d'applications IFS, a rapidement été écarté. L'algorithme itératif/implicite a par conséquent été sélectionné. C'est alors qu'ayant une vue d'ensemble,

[1] http://www.ec-nantes.fr/version-francaise/recherche/laboratoires/gem

les quatre éléments constitutifs d'un code IFS ont pu être abordés.

Le premier d'entre eux est **ISIS**, un solveur fluide de type RANSE. Il traite des problèmes d'écoulements visqueux turbulents incompressibles à haut nombre de Reynolds. Il permet, grâce à sa formulation Volumes-Finis non-structurés, de prendre en compte des géométries complexes. Au début de ce travail de thèse, la gestion des corps solides et déformables en mouvement imposé était d'ores et déjà développée, mais pas le calcul structure en lui-même.

Pour prédire les déformations des corps élancés, il a fallu développer un solveur poutre fine. Il est indépendant d'**ISIS** et peut être utilisé séparément. Il est basé sur les travaux de F. Boyer et D. Primault. La théorie s'appuie sur une approche de Cosserat avec une formulation géométriquement exacte. Deux modèles ont été intégrés : celui d'Euler-Bernoulli et celui de Rayleigh. Il est à noter que seul ce dernier peut être employé en 3D, car le modèle d'Euler-Bernoulli ne peut prendre en compte une poutre libre de tourner sur elle-même.

La phase de compréhension et de programmation de la théorie a été suivie par celle de la validation. Elle a été faite en bidimensionnel, ainsi qu'en tridimensionnel, sur des poutres de différentes longueurs, en petit et grand déplacement.

Une fois vérifié, le solveur poutre a été couplé à **ISIS**. Et ce de deux manières : d'une part de façon temporelle et d'autre part de façon spatiale. Une attention particulière a été portée au couplage en espace sur les interpolations et au transfert des informations à l'interface fluide-structure, afin d'être le plus précis possible et d'assurer la conservation des efforts. Une grande partie de cette thèse a été consacrée au processus de mise à jour du maillage. En effet, le module de remaillage doit être capable de recalculer le maillage autour d'un corps de forme quelconque et en grand déplacement. Pour cela, une approche pseudo-solide a été utilisée : le domaine de calcul fluide est considéré comme une structure solide élastique. Une formulation Volumes-Finis a vu le jour, afin de profiter des outils déjà présents dans **ISIS**. Cette dernière est originale, car elle permet, via une loi de comportement locale du pseudo-solide, de contrôler la déformation du maillage. On peut alors garder un mouvement quasi-rigide proche du corps en déplacement, ce qui permet de conserver la qualité du maillage et ainsi la précision, même en grand déplacement.

Une fois codée et testée, la méthode pseudo-structure a donné de bons résultats. Le problème qui est alors resté a été le coût CPU. En effet, on résout plusieurs fois un système dont la taille croit avec celle du maillage. Deux techniques de résolution ont dû être introduites pour réduire ce temps de calcul : une très rapide où l'on explicite le terme source, ce qui découple le problème, mais qui par conséquent n'est utilisable que pour des pas de temps faibles ; l'autre, sans découplage du problème, qui permet d'utiliser le module comme un outil indépendant de déformation de maillages. Mais elle a l'inconvénient d'être plus lente. Dans le cadre des applications IFS, la première méthode est utilisée. Par contre pour l'optimisation de forme, la seconde est nécessaire. De plus, pour diminuer la durée des simulations IFS, le remaillage pseudo-structure n'est réalisé qu'une seule fois par pas de temps, à la première itération IFS. Pour les suivantes, l'algorithme de remaillage par coefficients de pondération permet un réajustement des noeuds. Cette procédure étant

analytique, le gain en temps est important.

Enfin, pour les maillages autour de géométries difficiles, une utilisation conjointe des différents algorithmes de remaillage est possible : proche du corps, un mouvement rigide des noeuds, puis un maillage se déformant par pondération, le reste du domaine étant géré de manière pseudo-structure.

Tous ces programmes ont été codés et validés pour fonctionner en 2D, comme en 3D. Le code IFS peut être lancé en parallèle : **ISIS** fait appel à la bibliothèque MPI, couplée à $M\epsilon TIS$; le module de remaillage aussi. Le solveur poutre n'a pas besoin d'être parallélisé, car le coût de calcul et de mémoire qu'il entraîne est faible.

Le module de remaillage pseudo-structure a été utilisé avec succès sur divers solides, pour lesquels les algorithmes moins généraux déjà présents dans **ISIS** donnaient de piètres résultats : des corps élancées déformables 2D et 3D, comme des poutres ou encore une modélisation d'anguille (projet *ROBEA/RAAMO*) ; des corps indéformables en déplacement libre (comme le cas du "batteur-caisse") ou imposé. Le programme a aussi été employé pour le projet *VIRTUE* dans le but de générer des maillages autour d'une coque déformée d'un navire en vue d'une optimisation de celle-ci. Enfin, il a été testé en calcul parallèle sur un cas très complexe : un requin, modélisé de façon réaliste, en train de nager.

Quelques applications IFS ont été testées : un cas-test 2D pour valider le calcul stationnaire et statique du code IFS ; puis un calcul 2D sur une configuration très instationnaire. Cette dernière a permis de mettre à l'épreuve le couplage IFS développé. À ce propos, on note que ce code IFS non-monolithique, mais fortement couplé, parvient à traiter des cas difficiles, qui étaient réservés par le passé aux codes monolithiques. L'algorithme de discrétisation temporelle de Newmark a ici montré ses limites. Plusieurs schémas modifiés existent. Celui de Bossak a été intégré. Il permet de dissiper numériquement uniquement les hautes fréquences générées par le schéma standard, qui sont déstabilisantes pour le couplage IFS, et ce, sans dégrader l'ordre de précision.

Les possibilités du code en 3D ont été montrées par l'intermédiaire de deux exemples mettant en scène un câble déformable dans un milieu multifluide. Le code IFS peut traiter la physique d'un corps élancé fixé à une extrémité et subissant un fort courant. On peut aussi lui imposer une loi de mouvement à l'une de ses extrémités. Le code est capable de gérer ces grands déplacements : l'utilisateur définit des degrés de liberté pour la procédure de remaillage, les degrés de liberté restants étant pris en compte par mouvement rigide.

Un cas-test IFS 3D a aussi été réalisé. Il s'agit d'une comparaison avec l'expérience réalisée en 2004 par J.R. Chaplin et son équipe. Cette dernière consiste à étudier les déformations d'un riser sous l'action d'un courant. Les bons résultats tridimensionnels obtenus montrent la validité du code. La comparaison de ceux-ci avec ceux d'autres codes IFS souligne le fait qu'il est nécessaire d'avoir recours à un solveur fluide 3D pour traiter un tel problème.

Lors de ce doctorat, deux communications ont été effectuées : une dans le cadre du GDR d'IFS[2] de 2007 à Paris sur le module de remaillage ; la seconde au congrès interna-

[2] http://www.cnam.fr/lmssc/seminaires/interactfluidstruc/index.html

tional FIV2008 à Prague sur l'ensemble des travaux (cf. annexe D).

* * *

Les programmes, écrits lors de cette thèse, sont pleinement fonctionnels. Cependant, des améliorations peuvent encore être faites :

Au niveau du solveur structure, l'équipe prévoit d'avoir des poutres à géométrie variable et avec des propriétés physiques dépendantes de l'abscisse curviligne. Avec la possibilité supplémentaire d'imposer une loi de mouvement en un point différent des extrémités, des applications d'hydrodynamique navale comme des barges flexibles[3] ou encore des modèles de navires déformables (cf. [68]) seraient envisageables.

ISIS permet la gestion de plusieurs corps dans un même calcul. Cependant, le couplage IFS avec le solveur poutre permet à ce jour de ne définir qu'une seule poutre. La prise en charge des déformations de plusieurs poutres est possible, mais n'a pas été abordée.

Du côté du remaillage, le développement d'une formulation couplée sur les déplacements des noeuds de maillage apporterait un gain en temps lors de la résolution.

Le programme IFS développé est relativement nouveau et nécessite d'être testé sur davantage d'applications. À ce propos, le cas-test IFS 3D de J.R. Chaplin et de son équipe fournit de nombreuses données expérimentales pour différentes configurations de son expérience. Il serait donc intéressant de faire des simulations numériques pour des vitesses du fluide plus importantes et de voir si le code IFS donne encore de bons résultats par rapport à l'expérience.

De plus, un nouveau doctorat poursuivra l'avancée du laboratoire dans le domaine de la mécanique des interactions fluide-structure : le travail des trois prochaines années de M. Durand sera consacré à la réalisation d'un couplage entre **ISIS** et un code de déformation des membranes et coques. Ainsi, la prise en charge des mouvements des voiles des bateaux sera possible.

[3] https://www.irphe.univ-mrs.fr/-Interaction-Ocean-Atmosphere-

Annexe A

Calcul des coefficients de la matrice orthotrope en 3D pour le remailleur (MAPLE ™)

```
> restart;
> with(linalg):with(codegen,fortran):with(LinearAlgebra):
```

```
Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the assigned name GramSchmidt now has a global binding
```

P : matrice de passage B_0 (base xyz) —> B_1 (base locale 123)

[P11,P21,P31]^T : coordonnées du premier vecteur de la base locale exprimé dans X,Y,Z (vecteur normal à iso-CoefPond)

```
> P:=Matrix([[P11,P12,P13],[P21,P22,P23],[P31,P32,P33]]);
```

$$P := \begin{bmatrix} P11 & P12 & P13 \\ P21 & P22 & P23 \\ P31 & P32 & P33 \end{bmatrix}$$

```
> transpose(P);
```

$$\begin{bmatrix} P11 & P21 & P31 \\ P12 & P22 & P32 \\ P13 & P23 & P33 \end{bmatrix}$$

```
> Ptrans:=Matrix([[P11, P21, P31], [P12, P22, P32], [P13, P23,
> P33]]);
```

$$Ptrans := \begin{bmatrix} P11 & P21 & P31 \\ P12 & P22 & P32 \\ P13 & P23 & P33 \end{bmatrix}$$

```
> sigma:=Matrix([[sigma11,sigma12,sigma13],[sigma12,sigma22,sigma23],[s
> igma13,sigma23,sigma33]]);
```

$$\sigma := \begin{bmatrix} \sigma 11 & \sigma 12 & \sigma 13 \\ \sigma 12 & \sigma 22 & \sigma 23 \\ \sigma 13 & \sigma 23 & \sigma 33 \end{bmatrix}$$

```
> epsilon_123:=Matrix([[epsilon11,epsilon12,epsilon13],[epsilon12,epsil
> on22,epsilon23],[epsilon13,epsilon23,epsilon33]]);
```

$$epsilon_123 := \begin{bmatrix} \varepsilon 11 & \varepsilon 12 & \varepsilon 13 \\ \varepsilon 12 & \varepsilon 22 & \varepsilon 23 \\ \varepsilon 13 & \varepsilon 23 & \varepsilon 33 \end{bmatrix}$$

```
> epsilon_xyz:=Matrix([[eps_xx,eps_xy,eps_xz],[eps_xy,eps_yy,eps_yz],[e
> ps_xz,eps_yz,eps_zz]]);
```

$$epsilon_xyz := \begin{bmatrix} eps_xx & eps_xy & eps_xz \\ eps_xy & eps_yy & eps_yz \\ eps_xz & eps_yz & eps_zz \end{bmatrix}$$

```
> epsilon:=map(expand,Ptrans.epsilon_xyz.P);
```

$$\begin{aligned}
&\varepsilon := \\
&[P11^2\, eps_xx + 2\, P11\, P21\, eps_xy + 2\, P11\, P31\, eps_xz + P21^2\, eps_yy \\
&+ 2\, P21\, P31\, eps_yz + P31^2\, eps_zz, P12\, P11\, eps_xx + P12\, P21\, eps_xy \\
&+ P12\, P31\, eps_xz + P22\, P11\, eps_xy + P22\, P21\, eps_yy + P22\, P31\, eps_yz \\
&+ P32\, P11\, eps_xz + P32\, P21\, eps_yz + P32\, P31\, eps_zz, P13\, P11\, eps_xx \\
&+ P13\, P21\, eps_xy + P13\, P31\, eps_xz + P23\, P11\, eps_xy + P23\, P21\, eps_yy \\
&+ P23\, P31\, eps_yz + P33\, P11\, eps_xz + P33\, P21\, eps_yz + P33\, P31\, eps_zz] \\
&[P12\, P11\, eps_xx + P12\, P21\, eps_xy + P12\, P31\, eps_xz + P22\, P11\, eps_xy \\
&+ P22\, P21\, eps_yy + P22\, P31\, eps_yz + P32\, P11\, eps_xz + P32\, P21\, eps_yz \\
&+ P32\, P31\, eps_zz, P12^2\, eps_xx + 2\, P12\, P22\, eps_xy + 2\, P12\, P32\, eps_xz \\
&+ P22^2\, eps_yy + 2\, P22\, P32\, eps_yz + P32^2\, eps_zz, P13\, P12\, eps_xx \\
&+ P13\, P22\, eps_xy + P13\, P32\, eps_xz + P23\, P12\, eps_xy + P23\, P22\, eps_yy \\
&+ P23\, P32\, eps_yz + P33\, P12\, eps_xz + P33\, P22\, eps_yz + P33\, P32\, eps_zz] \\
&[P13\, P11\, eps_xx + P13\, P21\, eps_xy + P13\, P31\, eps_xz + P23\, P11\, eps_xy \\
&+ P23\, P21\, eps_yy + P23\, P31\, eps_yz + P33\, P11\, eps_xz + P33\, P21\, eps_yz \\
&+ P33\, P31\, eps_zz, P13\, P12\, eps_xx + P13\, P22\, eps_xy + P13\, P32\, eps_xz \\
&+ P23\, P12\, eps_xy + P23\, P22\, eps_yy + P23\, P32\, eps_yz + P33\, P12\, eps_xz \\
&+ P33\, P22\, eps_yz + P33\, P32\, eps_zz, P13^2\, eps_xx + 2\, P13\, P23\, eps_xy \\
&+ 2\, P13\, P33\, eps_xz + P23^2\, eps_yy + 2\, P23\, P33\, eps_yz + P33^2\, eps_zz]
\end{aligned}$$

```
> epsilon11:=collect(epsilon[1,1],[eps_xy,eps_yz,eps_xz]);
```

$$\begin{aligned}
&\varepsilon 11 := P11^2\, eps_xx + 2\, P11\, P21\, eps_xy + 2\, P11\, P31\, eps_xz + P21^2\, eps_yy \\
&+ 2\, P21\, P31\, eps_yz + P31^2\, eps_zz
\end{aligned}$$

```
> epsilon22:=collect(epsilon[2,2],[eps_xy,eps_yz,eps_xz]);
```

$$\begin{aligned}
&\varepsilon 22 := P12^2\, eps_xx + 2\, P12\, P22\, eps_xy + 2\, P12\, P32\, eps_xz + P22^2\, eps_yy \\
&+ 2\, P22\, P32\, eps_yz + P32^2\, eps_zz
\end{aligned}$$

```
> epsilon33:=collect(epsilon[3,3],[eps_xy,eps_yz,eps_xz]);
```

$$\varepsilon 33 := P13^2\ eps_xx + 2\ P13\ P23\ eps_xy + 2\ P13\ P33\ eps_xz + P23^2\ eps_yy + 2\ P23\ P33\ eps_yz + P33^2\ eps_zz$$

```
> epsilon12:=collect(epsilon[1,2],[eps_xy,eps_yz,eps_xz]);
```

$$\varepsilon 12 := (P12\ P21 + P22\ P11)\ eps_xy + (P32\ P21 + P22\ P31)\ eps_yz + (P32\ P11 + P12\ P31)\ eps_xz + P12\ P11\ eps_xx + P22\ P21\ eps_yy + P32\ P31\ eps_zz$$

```
> epsilon23:=collect(epsilon[2,3],[eps_xy,eps_yz,eps_xz]);
```

$$\varepsilon 23 := (P13\ P22 + P23\ P12)\ eps_xy + (P33\ P22 + P23\ P32)\ eps_yz + (P33\ P12 + P13\ P32)\ eps_xz + P13\ P12\ eps_xx + P23\ P22\ eps_yy + P33\ P32\ eps_zz$$

```
> epsilon13:=collect(epsilon[1,3],[eps_xy,eps_yz,eps_xz]);
```

$$\varepsilon 13 := (P13\ P21 + P23\ P11)\ eps_xy + (P33\ P21 + P23\ P31)\ eps_yz + (P33\ P11 + P13\ P31)\ eps_xz + P13\ P11\ eps_xx + P23\ P21\ eps_yy + P33\ P31\ eps_zz$$

Expression de Sigma_xyz en fonction des Sigma dans la base locale

```
> Sigma_xyz:=map(expand,P.sigma.Ptrans);
```

$$Sigma_xyz :=$$
$$[P11^2\,\sigma 11 + 2\ P11\ P12\,\sigma 12 + 2\ P11\ P13\,\sigma 13 + P12^2\,\sigma 22 + 2\ P12\ P13\,\sigma 23 + P13^2\,\sigma 33,\ P21\ P11\,\sigma 11 + P21\ P12\,\sigma 12 + P21\ P13\,\sigma 13 + P22\ P11\,\sigma 12 + P22\ P12\,\sigma 22 + P22\ P13\,\sigma 23 + P23\ P11\,\sigma 13 + P23\ P12\,\sigma 23 + P23\ P13\,\sigma 33,$$
$$P31\ P11\,\sigma 11 + P31\ P12\,\sigma 12 + P31\ P13\,\sigma 13 + P32\ P11\,\sigma 12 + P32\ P12\,\sigma 22 + P32\ P13\,\sigma 23 + P33\ P11\,\sigma 13 + P33\ P12\,\sigma 23 + P33\ P13\,\sigma 33]$$
$$[P21\ P11\,\sigma 11 + P21\ P12\,\sigma 12 + P21\ P13\,\sigma 13 + P22\ P11\,\sigma 12 + P22\ P12\,\sigma 22 + P22\ P13\,\sigma 23 + P23\ P11\,\sigma 13 + P23\ P12\,\sigma 23 + P23\ P13\,\sigma 33,\ P21^2\,\sigma 11 + 2\ P21\ P22\,\sigma 12 + 2\ P21\ P23\,\sigma 13 + P22^2\,\sigma 22 + 2\ P22\ P23\,\sigma 23 + P23^2\,\sigma 33,$$
$$P31\ P21\,\sigma 11 + P31\ P22\,\sigma 12 + P31\ P23\,\sigma 13 + P32\ P21\,\sigma 12 + P32\ P22\,\sigma 22 + P32\ P23\,\sigma 23 + P33\ P21\,\sigma 13 + P33\ P22\,\sigma 23 + P33\ P23\,\sigma 33]$$
$$[P31\ P11\,\sigma 11 + P31\ P12\,\sigma 12 + P31\ P13\,\sigma 13 + P32\ P11\,\sigma 12 + P32\ P12\,\sigma 22 + P32\ P13\,\sigma 23 + P33\ P11\,\sigma 13 + P33\ P12\,\sigma 23 + P33\ P13\,\sigma 33,\ P31\ P21\,\sigma 11 + P31\ P22\,\sigma 12 + P31\ P23\,\sigma 13 + P32\ P21\,\sigma 12 + P32\ P22\,\sigma 22 + P32\ P23\,\sigma 23 + P33\ P21\,\sigma 13 + P33\ P22\,\sigma 23 + P33\ P23\,\sigma 33,\ P31^2\,\sigma 11 + 2\ P31\ P32\,\sigma 12 + 2\ P31\ P33\,\sigma 13 + P32^2\,\sigma 22 + 2\ P32\ P33\,\sigma 23 + P33^2\,\sigma 33]$$

```
> sigma11:=epsilon11+fnu*epsilon22+fnu*epsilon33;
```

$$\sigma 11 := P11^2\ eps_xx + 2\ P11\ P21\ eps_xy + 2\ P11\ P31\ eps_xz + P21^2\ eps_yy + 2\ P21\ P31\ eps_yz + P31^2\ eps_zz + fnu(P12^2\ eps_xx + 2\ P12\ P22\ eps_xy + 2\ P12\ P32\ eps_xz + P22^2\ eps_yy + 2\ P22\ P32\ eps_yz + P32^2\ eps_zz) + fnu(P13^2\ eps_xx + 2\ P13\ P23\ eps_xy + 2\ P13\ P33\ eps_xz + P23^2\ eps_yy + 2\ P23\ P33\ eps_yz + P33^2\ eps_zz)$$

```
> sigma22:=fnu*epsilon11+epsilon22+fnu*epsilon33;
```

$$\sigma 22 := fnu(P11^2\ eps_xx + 2\ P11\ P21\ eps_xy + 2\ P11\ P31\ eps_xz + P21^2\ eps_yy + 2\ P21\ P31\ eps_yz + P31^2\ eps_zz) + P12^2\ eps_xx + 2\ P12\ P22\ eps_xy + 2\ P12\ P32\ eps_xz + P22^2\ eps_yy + 2\ P22\ P32\ eps_yz + P32^2\ eps_zz + fnu(P13^2\ eps_xx + 2\ P13\ P23\ eps_xy + 2\ P13\ P33\ eps_xz + P23^2\ eps_yy + 2\ P23\ P33\ eps_yz + P33^2\ eps_zz)$$

```
> sigma33:=fnu*epsilon11+fnu*epsilon22+epsilon33;
```

$$\sigma 33 := fnu(P11^2\ eps_xx + 2\ P11\ P21\ eps_xy + 2\ P11\ P31\ eps_xz + P21^2\ eps_yy + 2\ P21\ P31\ eps_yz + P31^2\ eps_zz) + fnu(P12^2\ eps_xx + 2\ P12\ P22\ eps_xy + 2\ P12\ P32\ eps_xz + P22^2\ eps_yy + 2\ P22\ P32\ eps_yz + P32^2\ eps_zz) + P13^2\ eps_xx + 2\ P13\ P23\ eps_xy + 2\ P13\ P33\ eps_xz + P23^2\ eps_yy + 2\ P23\ P33\ eps_yz + P33^2\ eps_zz$$

```
> sigma12:=G12surA*epsilon12;
```

$$\sigma 12 := G12surA((P12\ P21 + P22\ P11)\ eps_xy + (P32\ P21 + P22\ P31)\ eps_yz + (P32\ P11 + P12\ P31)\ eps_xz + P12\ P11\ eps_xx + P22\ P21\ eps_yy + P32\ P31\ eps_zz)$$

```
> sigma23:=G23surA*epsilon23;
```

$$\sigma 23 := G23surA((P13\ P22 + P23\ P12)\ eps_xy + (P33\ P22 + P23\ P32)\ eps_yz + (P33\ P12 + P13\ P32)\ eps_xz + P13\ P12\ eps_xx + P23\ P22\ eps_yy + P33\ P32\ eps_zz)$$

```
> sigma13:=G13surA*epsilon13;
```

$$\sigma 13 := G13surA((P13\ P21 + P23\ P11)\ eps_xy + (P33\ P21 + P23\ P31)\ eps_yz + (P33\ P11 + P13\ P31)\ eps_xz + P13\ P11\ eps_xx + P23\ P21\ eps_yy + P33\ P31\ eps_zz)$$

```
> sigma_xx:=collect(Sigma_xyz[1,1],[eps_xx,eps_yy,eps_zz,eps_xy,eps_yz,
> eps_xz]);
```

$$
\begin{aligned}
sigma_xx := {} & (P11^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11^2\,P12^2\,G12surA \\
& + 2\,P11^2\,P13^2\,G13surA + P12^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2) \\
& + 2\,P12^2\,P13^2\,G23surA + P13^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2))\,eps_xx + (\\
& P11^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P11\,P12\,G12surA\,P22\,P21 \\
& + 2\,P11\,P13\,G13surA\,P23\,P21 + P12^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2) \\
& + 2\,P12\,P13\,G23surA\,P23\,P22 + P13^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2))\,eps_yy + (\\
& P11^2\,(fnu\,P32^2 + fnu\,P33^2 + P31^2) + 2\,P11\,P12\,G12surA\,P32\,P31 \\
& + 2\,P11\,P13\,G13surA\,P33\,P31 + P12^2\,(P32^2 + fnu\,P31^2 + fnu\,P33^2) \\
& + 2\,P12\,P13\,G23surA\,P33\,P32 + P13^2\,(fnu\,P32^2 + fnu\,P31^2 + P33^2))\,eps_zz + (\\
& P11^2\,(2\,fnu\,P12\,P22 + 2\,P11\,P21 + 2\,fnu\,P13\,P23) \\
& + 2\,P11\,P12\,G12surA\,(P12\,P21 + P22\,P11) \\
& + 2\,P11\,P13\,G13surA\,(P13\,P21 + P23\,P11) \\
& + P12^2\,(2\,P12\,P22 + 2\,fnu\,P11\,P21 + 2\,fnu\,P13\,P23) \\
& + 2\,P12\,P13\,G23surA\,(P13\,P22 + P23\,P12) \\
& + P13^2\,(2\,fnu\,P11\,P21 + 2\,P13\,P23 + 2\,fnu\,P12\,P22))\,eps_xy + (\\
& P11^2\,(2\,fnu\,P23\,P33 + 2\,P21\,P31 + 2\,fnu\,P22\,P32) \\
& + 2\,P11\,P12\,G12surA\,(P32\,P21 + P22\,P31) \\
& + 2\,P11\,P13\,G13surA\,(P33\,P21 + P23\,P31) \\
& + P12^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P23\,P33 + 2\,P22\,P32) \\
& + 2\,P12\,P13\,G23surA\,(P33\,P22 + P23\,P32) \\
& + P13^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P22\,P32 + 2\,P23\,P33))\,eps_yz + (\\
& P11^2\,(2\,fnu\,P12\,P32 + 2\,fnu\,P13\,P33 + 2\,P11\,P31) \\
& + 2\,P11\,P12\,G12surA\,(P32\,P11 + P12\,P31) \\
& + 2\,P11\,P13\,G13surA\,(P33\,P11 + P13\,P31) \\
& + P12^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P13\,P33 + 2\,P12\,P32) \\
& + 2\,P12\,P13\,G23surA\,(P33\,P12 + P13\,P32) \\
& + P13^2\,(2\,fnu\,P11\,P31 + 2\,P13\,P33 + 2\,fnu\,P12\,P32))\,eps_xz
\end{aligned}
$$

```
> C11:=coeff(sigma_xx,eps_xx);
```

$$
\begin{aligned}
C11 := {} & P11^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11^2\,P12^2\,G12surA \\
& + 2\,P11^2\,P13^2\,G13surA + P12^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2) \\
& + 2\,P12^2\,P13^2\,G23surA + P13^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2)
\end{aligned}
$$

```
> fortran(C11);

      t0 =
P11**2*(fnu*P12**2+P11**2+fnu*P13**2)+2*P11**2*P12**2*G12surA


#+2*P11**2*P13**2*G13surA+P12**2*(fnu*P11**2+P12**2+fnu*P13**2)+2*P

     #12**2*P13**2*G23surA+P13**2*(fnu*P12**2+P13**2+fnu*P11**2)

> C12:=coeff(sigma_xx,eps_yy);
```

$$C12 := P11^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P11\,P12\,G12surA\,P22\,P21 + 2\,P11\,P13\,G13surA\,P23\,P21 + P12^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2) + 2\,P12\,P13\,G23surA\,P23\,P22 + P13^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2)$$

```
> fortran(C12);
```

```
      t0 =
P11**2*(fnu*P23**2+P21**2+fnu*P22**2)+2*P11*P12*G12surA*P22*P

#21+2*P11*P13*G13surA*P23*P21+P12**2*(fnu*P21**2+P22**2+fnu*P23**2)

     #+2*P12*P13*G23surA*P23*P22+P13**2*(fnu*P21**2+P23**2+fnu*P22**2)
```

```
> C13:=coeff(sigma_xx,eps_zz);
```

$$C13 := P11^2\,(fnu\,P32^2 + fnu\,P33^2 + P31^2) + 2\,P11\,P12\,G12surA\,P32\,P31 + 2\,P11\,P13\,G13surA\,P33\,P31 + P12^2\,(P32^2 + fnu\,P31^2 + fnu\,P33^2) + 2\,P12\,P13\,G23surA\,P33\,P32 + P13^2\,(fnu\,P32^2 + fnu\,P31^2 + P33^2)$$

```
> fortran(C13);
```

```
      t0 =
P11**2*(fnu*P32**2+fnu*P33**2+P31**2)+2*P11*P12*G12surA*P32*P

#31+2*P11*P13*G13surA*P33*P31+P12**2*(P32**2+fnu*P31**2+fnu*P33**2)

     #+2*P12*P13*G23surA*P33*P32+P13**2*(fnu*P32**2+fnu*P31**2+P33**2)
```

```
> C14:=coeff(sigma_xx,eps_xy);
```

$$C14 := P11^2\,(2\,fnu\,P12\,P22 + 2\,P11\,P21 + 2\,fnu\,P13\,P23) + 2\,P11\,P12\,G12surA\,(P12\,P21 + P22\,P11) + 2\,P11\,P13\,G13surA\,(P13\,P21 + P23\,P11) + P12^2\,(2\,P12\,P22 + 2\,fnu\,P11\,P21 + 2\,fnu\,P13\,P23) + 2\,P12\,P13\,G23surA\,(P13\,P22 + P23\,P12) + P13^2\,(2\,fnu\,P11\,P21 + 2\,P13\,P23 + 2\,fnu\,P12\,P22)$$

```
> fortran(C14);
```

```
      t0 =
P11**2*(2*fnu*P12*P22+2*P11*P21+2*fnu*P13*P23)+2*P11*P12*G12s

#urA*(P12*P21+P22*P11)+2*P11*P13*G13surA*(P13*P21+P23*P11)+P12**2*(

#2*P12*P22+2*fnu*P11*P21+2*fnu*P13*P23)+2*P12*P13*G23surA*(P13*P22+

     #P23*P12)+P13**2*(2*fnu*P11*P21+2*P13*P23+2*fnu*P12*P22)
```

```
> C15:=coeff(sigma_xx,eps_yz);
```

$$
\begin{aligned}
C15 := {} & P11^2\,(2\,fnu\,P23\,P33 + 2\,P21\,P31 + 2\,fnu\,P22\,P32) \\
& + 2\,P11\,P12\,G12surA\,(P32\,P21 + P22\,P31) \\
& + 2\,P11\,P13\,G13surA\,(P33\,P21 + P23\,P31) \\
& + P12^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P23\,P33 + 2\,P22\,P32) \\
& + 2\,P12\,P13\,G23surA\,(P33\,P22 + P23\,P32) \\
& + P13^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P22\,P32 + 2\,P23\,P33)
\end{aligned}
$$

```
> fortran(C15);

      t0 =
P11**2*(2*fnu*P23*P33+2*P21*P31+2*fnu*P22*P32)+2*P11*P12*G12s

#urA*(P32*P21+P22*P31)+2*P11*P13*G13surA*(P33*P21+P23*P31)+P12**2*(

#2*fnu*P21*P31+2*fnu*P23*P33+2*P22*P32)+2*P12*P13*G23surA*(P33*P22+

     #P23*P32)+P13**2*(2*fnu*P21*P31+2*fnu*P22*P32+2*P23*P33)

> C16:=coeff(sigma_xx,eps_xz);
```

$$
\begin{aligned}
C16 := {} & P11^2\,(2\,fnu\,P12\,P32 + 2\,fnu\,P13\,P33 + 2\,P11\,P31) \\
& + 2\,P11\,P12\,G12surA\,(P32\,P11 + P12\,P31) \\
& + 2\,P11\,P13\,G13surA\,(P33\,P11 + P13\,P31) \\
& + P12^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P13\,P33 + 2\,P12\,P32) \\
& + 2\,P12\,P13\,G23surA\,(P33\,P12 + P13\,P32) \\
& + P13^2\,(2\,fnu\,P11\,P31 + 2\,P13\,P33 + 2\,fnu\,P12\,P32)
\end{aligned}
$$

```
> fortran(C16);

      t0 =
P11**2*(2*fnu*P12*P32+2*fnu*P13*P33+2*P11*P31)+2*P11*P12*G12s

#urA*(P32*P11+P12*P31)+2*P11*P13*G13surA*(P33*P11+P13*P31)+P12**2*(

#2*fnu*P11*P31+2*fnu*P13*P33+2*P12*P32)+2*P12*P13*G23surA*(P33*P12+

     #P13*P32)+P13**2*(2*fnu*P11*P31+2*P13*P33+2*fnu*P12*P32)

> sigma_yy:=collect(Sigma_xyz[2,2],[eps_xx,eps_yy,eps_zz,eps_xy,eps_yz,
> eps_xz]);
```

$$
\begin{aligned}
sigma_yy := {} & (P21^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11\,P12\,G12surA\,P22\,P21 \\
& + 2\,P11\,P13\,G13surA\,P23\,P21 + P22^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2) \\
& + 2\,P12\,P13\,G23surA\,P23\,P22 + P23^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2))\,eps_xx + (\\
& P21^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P21^2\,P22^2\,G12surA \\
& + 2\,P21^2\,P23^2\,G13surA + P22^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2) \\
& + 2\,P22^2\,P23^2\,G23surA + P23^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2))\,eps_yy + (\\
& P21^2\,(fnu\,P32^2 + fnu\,P33^2 + P31^2) + 2\,P21\,P22\,G12surA\,P32\,P31 \\
& + 2\,P21\,P23\,G13surA\,P33\,P31 + P22^2\,(P32^2 + fnu\,P31^2 + fnu\,P33^2) \\
& + 2\,P22\,P23\,G23surA\,P33\,P32 + P23^2\,(fnu\,P32^2 + fnu\,P31^2 + P33^2))\,eps_zz + (\\
& P21^2\,(2\,fnu\,P12\,P22 + 2\,P11\,P21 + 2\,fnu\,P13\,P23) \\
& + 2\,P21\,P22\,G12surA\,(P12\,P21 + P22\,P11) \\
& + 2\,P21\,P23\,G13surA\,(P13\,P21 + P23\,P11) \\
& + P22^2\,(2\,P12\,P22 + 2\,fnu\,P11\,P21 + 2\,fnu\,P13\,P23) \\
& + 2\,P22\,P23\,G23surA\,(P13\,P22 + P23\,P12) \\
& + P23^2\,(2\,fnu\,P11\,P21 + 2\,P13\,P23 + 2\,fnu\,P12\,P22))\,eps_xy + (\\
& P21^2\,(2\,fnu\,P23\,P33 + 2\,P21\,P31 + 2\,fnu\,P22\,P32) \\
& + 2\,P21\,P22\,G12surA\,(P32\,P21 + P22\,P31) \\
& + 2\,P21\,P23\,G13surA\,(P33\,P21 + P23\,P31) \\
& + P22^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P23\,P33 + 2\,P22\,P32) \\
& + 2\,P22\,P23\,G23surA\,(P33\,P22 + P23\,P32) \\
& + P23^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P22\,P32 + 2\,P23\,P33))\,eps_yz + (\\
& P21^2\,(2\,fnu\,P12\,P32 + 2\,fnu\,P13\,P33 + 2\,P11\,P31) \\
& + 2\,P21\,P22\,G12surA\,(P32\,P11 + P12\,P31) \\
& + 2\,P21\,P23\,G13surA\,(P33\,P11 + P13\,P31) \\
& + P22^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P13\,P33 + 2\,P12\,P32) \\
& + 2\,P22\,P23\,G23surA\,(P33\,P12 + P13\,P32) \\
& + P23^2\,(2\,fnu\,P11\,P31 + 2\,P13\,P33 + 2\,fnu\,P12\,P32))\,eps_xz
\end{aligned}
$$

```
> C21:=coeff(sigma_yy,eps_xx);
```

$$
\begin{aligned}
C21 := {} & P21^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11\,P12\,G12surA\,P22\,P21 \\
& + 2\,P11\,P13\,G13surA\,P23\,P21 + P22^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2) \\
& + 2\,P12\,P13\,G23surA\,P23\,P22 + P23^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2)
\end{aligned}
$$

```
> fortran(C21);

      t0 =
P21**2*(fnu*P12**2+P11**2+fnu*P13**2)+2*P11*P12*G12surA*P22*P

#21+2*P11*P13*G13surA*P23*P21+P22**2*(fnu*P11**2+P12**2+fnu*P13**2)

     #+2*P12*P13*G23surA*P23*P22+P23**2*(fnu*P12**2+P13**2+fnu*P11**2)
> C22:=coeff(sigma_yy,eps_yy);
```

$$C22 := P21^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P21^2\,P22^2\,G12surA + 2\,P21^2\,P23^2\,G13surA + P22^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2) + 2\,P22^2\,P23^2\,G23surA + P23^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2)$$

```
> fortran(C22);
```

```
      t0 =
P21**2*(fnu*P23**2+P21**2+fnu*P22**2)+2*P21**2*P22**2*G12surA

#+2*P21**2*P23**2*G13surA+P22**2*(fnu*P21**2+P22**2+fnu*P23**2)+2*P

     #22**2*P23**2*G23surA+P23**2*(fnu*P21**2+P23**2+fnu*P22**2)
```

```
> C23:=coeff(sigma_yy,eps_zz);
```

$$C23 := P21^2\,(fnu\,P32^2 + fnu\,P33^2 + P31^2) + 2\,P21\,P22\,G12surA\,P32\,P31 + 2\,P21\,P23\,G13surA\,P33\,P31 + P22^2\,(P32^2 + fnu\,P31^2 + fnu\,P33^2) + 2\,P22\,P23\,G23surA\,P33\,P32 + P23^2\,(fnu\,P32^2 + fnu\,P31^2 + P33^2)$$

```
> fortran(C23);
```

```
      t0 =
P21**2*(fnu*P32**2+fnu*P33**2+P31**2)+2*P21*P22*G12surA*P32*P

#31+2*P21*P23*G13surA*P33*P31+P22**2*(P32**2+fnu*P31**2+fnu*P33**2)

     #+2*P22*P23*G23surA*P33*P32+P23**2*(fnu*P32**2+fnu*P31**2+P33**2)
```

```
> C24:=coeff(sigma_yy,eps_xy);
```

$$C24 := P21^2\,(2\,fnu\,P12\,P22 + 2\,P11\,P21 + 2\,fnu\,P13\,P23) + 2\,P21\,P22\,G12surA\,(P12\,P21 + P22\,P11) + 2\,P21\,P23\,G13surA\,(P13\,P21 + P23\,P11) + P22^2\,(2\,P12\,P22 + 2\,fnu\,P11\,P21 + 2\,fnu\,P13\,P23) + 2\,P22\,P23\,G23surA\,(P13\,P22 + P23\,P12) + P23^2\,(2\,fnu\,P11\,P21 + 2\,P13\,P23 + 2\,fnu\,P12\,P22)$$

```
> fortran(C24);
```

```
      t0 =
P21**2*(2*fnu*P12*P22+2*P11*P21+2*fnu*P13*P23)+2*P21*P22*G12s

#urA*(P12*P21+P22*P11)+2*P21*P23*G13surA*(P13*P21+P23*P11)+P22**2*(

#2*P12*P22+2*fnu*P11*P21+2*fnu*P13*P23)+2*P22*P23*G23surA*(P13*P22+

     #P23*P12)+P23**2*(2*fnu*P11*P21+2*P13*P23+2*fnu*P12*P22)
```

```
> C25:=coeff(sigma_yy,eps_yz);
```

$$
\begin{aligned}
C25 := {} & P21^2\,(2\,fnu\,P23\,P33 + 2\,P21\,P31 + 2\,fnu\,P22\,P32) \\
& + 2\,P21\,P22\,G12surA\,(P32\,P21 + P22\,P31) \\
& + 2\,P21\,P23\,G13surA\,(P33\,P21 + P23\,P31) \\
& + P22^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P23\,P33 + 2\,P22\,P32) \\
& + 2\,P22\,P23\,G23surA\,(P33\,P22 + P23\,P32) \\
& + P23^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P22\,P32 + 2\,P23\,P33)
\end{aligned}
$$

```
> fortran(C25);
```

```
      t0 =
P21**2*(2*fnu*P23*P33+2*P21*P31+2*fnu*P22*P32)+2*P21*P22*G12s

#urA*(P32*P21+P22*P31)+2*P21*P23*G13surA*(P33*P21+P23*P31)+P22**2*(

#2*fnu*P21*P31+2*fnu*P23*P33+2*P22*P32)+2*P22*P23*G23surA*(P33*P22+

     #P23*P32)+P23**2*(2*fnu*P21*P31+2*fnu*P22*P32+2*P23*P33)
```

```
> C26:=coeff(sigma_yy,eps_xz);
```

$$
\begin{aligned}
C26 := {} & P21^2\,(2\,fnu\,P12\,P32 + 2\,fnu\,P13\,P33 + 2\,P11\,P31) \\
& + 2\,P21\,P22\,G12surA\,(P32\,P11 + P12\,P31) \\
& + 2\,P21\,P23\,G13surA\,(P33\,P11 + P13\,P31) \\
& + P22^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P13\,P33 + 2\,P12\,P32) \\
& + 2\,P22\,P23\,G23surA\,(P33\,P12 + P13\,P32) \\
& + P23^2\,(2\,fnu\,P11\,P31 + 2\,P13\,P33 + 2\,fnu\,P12\,P32)
\end{aligned}
$$

```
> fortran(C26);
```

```
      t0 =
P21**2*(2*fnu*P12*P32+2*fnu*P13*P33+2*P11*P31)+2*P21*P22*G12s

#urA*(P32*P11+P12*P31)+2*P21*P23*G13surA*(P33*P11+P13*P31)+P22**2*(

#2*fnu*P11*P31+2*fnu*P13*P33+2*P12*P32)+2*P22*P23*G23surA*(P33*P12+

     #P13*P32)+P23**2*(2*fnu*P11*P31+2*P13*P33+2*fnu*P12*P32)
```

```
> sigma_zz:=collect(Sigma_xyz[3,3],[eps_xx,eps_yy,eps_zz,eps_xy,eps_yz,
> eps_xz]);
```

$$sigma_zz := (P31^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11\,P12\,G12surA\,P32\,P31$$
$$+ 2\,P11\,P13\,G13surA\,P33\,P31 + P32^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2)$$
$$+ 2\,P12\,P13\,G23surA\,P33\,P32 + P33^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2))\,eps_xx + ($$
$$P31^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P21\,P22\,G12surA\,P32\,P31$$
$$+ 2\,P21\,P23\,G13surA\,P33\,P31 + P32^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2)$$
$$+ 2\,P22\,P23\,G23surA\,P33\,P32 + P33^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2))\,eps_yy + ($$
$$P31^2\,(fnu\,P32^2 + fnu\,P33^2 + P31^2) + 2\,P32^2\,P31^2\,G12surA$$
$$+ 2\,P33^2\,P31^2\,G13surA + P32^2\,(P32^2 + fnu\,P31^2 + fnu\,P33^2)$$
$$+ 2\,P33^2\,P32^2\,G23surA + P33^2\,(fnu\,P32^2 + fnu\,P31^2 + P33^2))\,eps_zz + ($$
$$P31^2\,(2\,fnu\,P12\,P22 + 2\,P11\,P21 + 2\,fnu\,P13\,P23)$$
$$+ 2\,P32\,P31\,G12surA\,(P12\,P21 + P22\,P11)$$
$$+ 2\,P33\,P31\,G13surA\,(P13\,P21 + P23\,P11)$$
$$+ P32^2\,(2\,P12\,P22 + 2\,fnu\,P11\,P21 + 2\,fnu\,P13\,P23)$$
$$+ 2\,P33\,P32\,G23surA\,(P13\,P22 + P23\,P12)$$
$$+ P33^2\,(2\,fnu\,P11\,P21 + 2\,P13\,P23 + 2\,fnu\,P12\,P22))\,eps_xy + ($$
$$P31^2\,(2\,fnu\,P23\,P33 + 2\,P21\,P31 + 2\,fnu\,P22\,P32)$$
$$+ 2\,P32\,P31\,G12surA\,(P32\,P21 + P22\,P31)$$
$$+ 2\,P33\,P31\,G13surA\,(P33\,P21 + P23\,P31)$$
$$+ P32^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P23\,P33 + 2\,P22\,P32)$$
$$+ 2\,P33\,P32\,G23surA\,(P33\,P22 + P23\,P32)$$
$$+ P33^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P22\,P32 + 2\,P23\,P33))\,eps_yz + ($$
$$P31^2\,(2\,fnu\,P12\,P32 + 2\,fnu\,P13\,P33 + 2\,P11\,P31)$$
$$+ 2\,P32\,P31\,G12surA\,(P32\,P11 + P12\,P31)$$
$$+ 2\,P33\,P31\,G13surA\,(P33\,P11 + P13\,P31)$$
$$+ P32^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P13\,P33 + 2\,P12\,P32)$$
$$+ 2\,P33\,P32\,G23surA\,(P33\,P12 + P13\,P32)$$
$$+ P33^2\,(2\,fnu\,P11\,P31 + 2\,P13\,P33 + 2\,fnu\,P12\,P32))\,eps_xz$$

```
> C31:=coeff(sigma_zz,eps_xx);
```

$$C31 := P31^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11\,P12\,G12surA\,P32\,P31$$
$$+ 2\,P11\,P13\,G13surA\,P33\,P31 + P32^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2)$$
$$+ 2\,P12\,P13\,G23surA\,P33\,P32 + P33^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2)$$

```
> fortran(C31);

      t0 =
P31**2*(fnu*P12**2+P11**2+fnu*P13**2)+2*P11*P12*G12surA*P32*P


#31+2*P11*P13*G13surA*P33*P31+P32**2*(fnu*P11**2+P12**2+fnu*P13**2)

     #+2*P12*P13*G23surA*P33*P32+P33**2*(fnu*P12**2+P13**2+fnu*P11**2)
> C32:=coeff(sigma_zz,eps_yy);
```

$$C32 := P31^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P21\,P22\,G12surA\,P32\,P31 + 2\,P21\,P23\,G13surA\,P33\,P31 + P32^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2) + 2\,P22\,P23\,G23surA\,P33\,P32 + P33^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2)$$

```
> fortran(C32);
```

```
      t0 =
P31**2*(fnu*P23**2+P21**2+fnu*P22**2)+2*P21*P22*G12surA*P32*P

#31+2*P21*P23*G13surA*P33*P31+P32**2*(fnu*P21**2+P22**2+fnu*P23**2)

     #+2*P22*P23*G23surA*P33*P32+P33**2*(fnu*P21**2+P23**2+fnu*P22**2)
```

```
> C33:=coeff(sigma_zz,eps_zz);
```

$$C33 := P31^2\,(fnu\,P32^2 + fnu\,P33^2 + P31^2) + 2\,P32^2\,P31^2\,G12surA + 2\,P33^2\,P31^2\,G13surA + P32^2\,(P32^2 + fnu\,P31^2 + fnu\,P33^2) + 2\,P33^2\,P32^2\,G23surA + P33^2\,(fnu\,P32^2 + fnu\,P31^2 + P33^2)$$

```
> fortran(C33);
```

```
      t0 =
P31**2*(fnu*P32**2+fnu*P33**2+P31**2)+2*P32**2*P31**2*G12surA

#+2*P33**2*P31**2*G13surA+P32**2*(P32**2+fnu*P31**2+fnu*P33**2)+2*P

     #33**2*P32**2*G23surA+P33**2*(fnu*P32**2+fnu*P31**2+P33**2)
```

```
> C34:=coeff(sigma_zz,eps_xy);
```

$$C34 := P31^2\,(2\,fnu\,P12\,P22 + 2\,P11\,P21 + 2\,fnu\,P13\,P23) + 2\,P32\,P31\,G12surA\,(P12\,P21 + P22\,P11) + 2\,P33\,P31\,G13surA\,(P13\,P21 + P23\,P11) + P32^2\,(2\,P12\,P22 + 2\,fnu\,P11\,P21 + 2\,fnu\,P13\,P23) + 2\,P33\,P32\,G23surA\,(P13\,P22 + P23\,P12) + P33^2\,(2\,fnu\,P11\,P21 + 2\,P13\,P23 + 2\,fnu\,P12\,P22)$$

```
> fortran(C34);
```

```
      t0 =
P31**2*(2*fnu*P12*P22+2*P11*P21+2*fnu*P13*P23)+2*P32*P31*G12s

#urA*(P12*P21+P22*P11)+2*P33*P31*G13surA*(P13*P21+P23*P11)+P32**2*(

#2*P12*P22+2*fnu*P11*P21+2*fnu*P13*P23)+2*P33*P32*G23surA*(P13*P22+

     #P23*P12)+P33**2*(2*fnu*P11*P21+2*P13*P23+2*fnu*P12*P22)
```

```
> C35:=coeff(sigma_zz,eps_yz);
```

$$
\begin{aligned}
C35 := {} & P31^2\,(2\,fnu\,P23\,P33 + 2\,P21\,P31 + 2\,fnu\,P22\,P32) \\
& + 2\,P32\,P31\,G12surA\,(P32\,P21 + P22\,P31) \\
& + 2\,P33\,P31\,G13surA\,(P33\,P21 + P23\,P31) \\
& + P32^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P23\,P33 + 2\,P22\,P32) \\
& + 2\,P33\,P32\,G23surA\,(P33\,P22 + P23\,P32) \\
& + P33^2\,(2\,fnu\,P21\,P31 + 2\,fnu\,P22\,P32 + 2\,P23\,P33)
\end{aligned}
$$

```
> fortran(C35);
```

```
      t0 =
P31**2*(2*fnu*P23*P33+2*P21*P31+2*fnu*P22*P32)+2*P32*P31*G12s

#urA*(P32*P21+P22*P31)+2*P33*P31*G13surA*(P33*P21+P23*P31)+P32**2*(

#2*fnu*P21*P31+2*fnu*P23*P33+2*P22*P32)+2*P33*P32*G23surA*(P33*P22+

     #P23*P32)+P33**2*(2*fnu*P21*P31+2*fnu*P22*P32+2*P23*P33)
```

```
> C36:=coeff(sigma_zz,eps_xz);
```

$$
\begin{aligned}
C36 := {} & P31^2\,(2\,fnu\,P12\,P32 + 2\,fnu\,P13\,P33 + 2\,P11\,P31) \\
& + 2\,P32\,P31\,G12surA\,(P32\,P11 + P12\,P31) \\
& + 2\,P33\,P31\,G13surA\,(P33\,P11 + P13\,P31) \\
& + P32^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P13\,P33 + 2\,P12\,P32) \\
& + 2\,P33\,P32\,G23surA\,(P33\,P12 + P13\,P32) \\
& + P33^2\,(2\,fnu\,P11\,P31 + 2\,fnu\,P12\,P32)
\end{aligned}
$$

```
> fortran(C36);
```

```
      t0 =
P31**2*(2*fnu*P12*P32+2*fnu*P13*P33+2*P11*P31)+2*P32*P31*G12s

#urA*(P32*P11+P12*P31)+2*P33*P31*G13surA*(P33*P11+P13*P31)+P32**2*(

#2*fnu*P11*P31+2*fnu*P13*P33+2*P12*P32)+2*P33*P32*G23surA*(P33*P12+

     #P13*P32)+P33**2*(2*fnu*P11*P31+2*P13*P33+2*fnu*P12*P32)
```

```
> sigma_xy:=collect(Sigma_xyz[1,2],[eps_xx,eps_yy,eps_zz,eps_xy,eps_yz,
> eps_xz]);
```

$$
\begin{aligned}
&sigma_xy := (P13^2\ P21\ G13surA\ P11 + P22\ P11^2\ G12surA\ P12 \\
&+ P12\ P22\ (fnu\ P11^2 + P12^2 + fnu\ P13^2) + P13^2\ P22\ G23surA\ P12 \\
&+ P23\ P11^2\ G13surA\ P13 + P11\ P21\ (fnu\ P12^2 + P11^2 + fnu\ P13^2) \\
&+ P23\ P12^2\ G23surA\ P13 + P12^2\ P21\ G12surA\ P11 \\
&+ P13\ P23\ (fnu\ P12^2 + P13^2 + fnu\ P11^2))eps_xx + (P12\ P21^2\ G12surA\ P22 \\
&+ P13\ P23\ (fnu\ P21^2 + P23^2 + fnu\ P22^2) + P13\ P21^2\ G13surA\ P23 \\
&+ P22^2\ P11\ G12surA\ P21 + P12\ P22\ (fnu\ P21^2 + P22^2 + fnu\ P23^2) \\
&+ P11\ P21\ (fnu\ P23^2 + P21^2 + fnu\ P22^2) + P13\ P22^2\ G23surA\ P23 \\
&+ P23^2\ P12\ G23surA\ P22 + P23^2\ P11\ G13surA\ P21)eps_yy + (\\
&P23\ P12\ G23surA\ P33\ P32 + P23\ P11\ G13surA\ P33\ P31 \\
&+ P12\ P21\ G12surA\ P32\ P31 + P13\ P23\ (fnu\ P32^2 + fnu\ P31^2 + P33^2) \\
&+ P13\ P21\ G13surA\ P33\ P31 + P11\ P21\ (fnu\ P32^2 + fnu\ P33^2 + P31^2) \\
&+ P22\ P11\ G12surA\ P32\ P31 + P13\ P22\ G23surA\ P33\ P32 \\
&+ P12\ P22\ (P32^2 + fnu\ P31^2 + fnu\ P33^2))eps_zz + (\\
&P13\ P22\ G23surA\ (P13\ P22 + P23\ P12) \\
&+ P12\ P22\ (2\ P12\ P22 + 2\ fnu\ P11\ P21 + 2\ fnu\ P13\ P23) \\
&+ P23\ P12\ G23surA\ (P13\ P22 + P23\ P12) \\
&+ P23\ P11\ G13surA\ (P13\ P21 + P23\ P11) \\
&+ P12\ P21\ G12surA\ (P12\ P21 + P22\ P11) \\
&+ P11\ P21\ (2\ fnu\ P12\ P22 + 2\ P11\ P21 + 2\ fnu\ P13\ P23) \\
&+ P13\ P23\ (2\ fnu\ P11\ P21 + 2\ P13\ P23 + 2\ fnu\ P12\ P22) \\
&+ P22\ P11\ G12surA\ (P12\ P21 + P22\ P11) \\
&+ P13\ P21\ G13surA\ (P13\ P21 + P23\ P11))eps_xy + (\\
&P22\ P11\ G12surA\ (P32\ P21 + P22\ P31) \\
&+ P13\ P21\ G13surA\ (P33\ P21 + P23\ P31) \\
&+ P13\ P22\ G23surA\ (P33\ P22 + P23\ P32) \\
&+ P12\ P22\ (2\ fnu\ P21\ P31 + 2\ fnu\ P23\ P33 + 2\ P22\ P32) \\
&+ P23\ P12\ G23surA\ (P33\ P22 + P23\ P32) \\
&+ P11\ P21\ (2\ fnu\ P23\ P33 + 2\ P21\ P31 + 2\ fnu\ P22\ P32) \\
&+ P23\ P11\ G13surA\ (P33\ P21 + P23\ P31) \\
&+ P13\ P23\ (2\ fnu\ P21\ P31 + 2\ fnu\ P22\ P32 + 2\ P23\ P33) \\
&+ P12\ P21\ G12surA\ (P32\ P21 + P22\ P31))eps_yz + (\\
&P11\ P21\ (2\ fnu\ P12\ P32 + 2\ fnu\ P13\ P33 + 2\ P11\ P31) \\
&+ P23\ P11\ G13surA\ (P33\ P11 + P13\ P31) \\
&+ P13\ P23\ (2\ fnu\ P11\ P31 + 2\ P13\ P33 + 2\ fnu\ P12\ P32) \\
&+ P12\ P21\ G12surA\ (P32\ P11 + P12\ P31) \\
&+ P22\ P11\ G12surA\ (P32\ P11 + P12\ P31) \\
&+ P13\ P21\ G13surA\ (P33\ P11 + P13\ P31) \\
&+ P13\ P22\ G23surA\ (P33\ P12 + P13\ P32) \\
&+ P12\ P22\ (2\ fnu\ P11\ P31 + 2\ fnu\ P13\ P33 + 2\ P12\ P32) \\
&+ P23\ P12\ G23surA\ (P33\ P12 + P13\ P32))eps_xz
\end{aligned}
$$

```
> C41:=coeff(sigma_xy,eps_xx);
```

$$C41 := P13^2\ P21\ G13surA\ P11 + P22\ P11^2\ G12surA\ P12 + P12\ P22\ (fnu\ P11^2 + P12^2 + fnu\ P13^2) + P13^2\ P22\ G23surA\ P12 + P23\ P11^2\ G13surA\ P13 + P11\ P21\ (fnu\ P12^2 + P11^2 + fnu\ P13^2) + P23\ P12^2\ G23surA\ P13 + P12^2\ P21\ G12surA\ P11 + P13\ P23\ (fnu\ P12^2 + P13^2 + fnu\ P11^2)$$

```
> fortran(C41);

      t0 =
P13**2*P21*G13surA*P11+P22*P11**2*G12surA*P12+P12*P22*(fnu*P1

#1**2+P12**2+fnu*P13**2)+P13**2*P22*G23surA*P12+P23*P11**2*G13surA*

#P13+P11*P21*(fnu*P12**2+P11**2+fnu*P13**2)+P23*P12**2*G23surA*P13+

     #P12**2*P21*G12surA*P11+P13*P23*(fnu*P12**2+P13**2+fnu*P11**2)
> C42:=coeff(sigma_xy,eps_yy);
```

$$C42 := P12\ P21^2\ G12surA\ P22 + P13\ P23\ (fnu\ P21^2 + P23^2 + fnu\ P22^2) + P13\ P21^2\ G13surA\ P23 + P22^2\ P11\ G12surA\ P21 + P12\ P22\ (fnu\ P21^2 + P22^2 + fnu\ P23^2) + P11\ P21\ (fnu\ P23^2 + P21^2 + fnu\ P22^2) + P13\ P22^2\ G23surA\ P23 + P23^2\ P12\ G23surA\ P22 + P23^2\ P11\ G13surA\ P21$$

```
> fortran(C42);

      t0 =
P12*P21**2*G12surA*P22+P13*P23*(fnu*P21**2+P23**2+fnu*P22**2)

#+P13*P21**2*G13surA*P23+P22**2*P11*G12surA*P21+P12*P22*(fnu*P21**2

#+P22**2+fnu*P23**2)+P11*P21*(fnu*P23**2+P21**2+fnu*P22**2)+P13*P22

     #**2*G23surA*P23+P23**2*P12*G23surA*P22+P23**2*P11*G13surA*P21
> C43:=coeff(sigma_xy,eps_zz);
```

$$C43 := P23\ P12\ G23surA\ P33\ P32 + P23\ P11\ G13surA\ P33\ P31 + P12\ P21\ G12surA\ P32\ P31 + P13\ P23\ (fnu\ P32^2 + fnu\ P31^2 + P33^2) + P13\ P21\ G13surA\ P33\ P31 + P11\ P21\ (fnu\ P32^2 + fnu\ P33^2 + P31^2) + P22\ P11\ G12surA\ P32\ P31 + P13\ P22\ G23surA\ P33\ P32 + P12\ P22\ (P32^2 + fnu\ P31^2 + fnu\ P33^2)$$

```
> fortran(C43);

      t0 =
P23*P12*G23surA*P33*P32+P23*P11*G13surA*P33*P31+P12*P21*G12su

#rA*P32*P31+P13*P23*(fnu*P32**2+fnu*P31**2+P33**2)+P13*P21*G13surA*
```

```
#P33*P31+P11*P21*(fnu*P32**2+fnu*P33**2+P31**2)+P22*P11*G12surA*P32

#*P31+P13*P22*G23surA*P33*P32+P12*P22*(P32**2+fnu*P31**2+fnu*P33**2

     #)
>  C44:=coeff(sigma_xy,eps_xy);
```

$$
\begin{aligned}
C44 := {} & P13\ P22\ G23surA\,(P13\ P22 + P23\ P12) \\
& + P12\ P22\,(2\,P12\ P22 + 2\,fnu\ P11\ P21 + 2\,fnu\ P13\ P23) \\
& + P23\ P12\ G23surA\,(P13\ P22 + P23\ P12) \\
& + P23\ P11\ G13surA\,(P13\ P21 + P23\ P11) \\
& + P12\ P21\ G12surA\,(P12\ P21 + P22\ P11) \\
& + P11\ P21\,(2\,fnu\ P12\ P22 + 2\,P11\ P21 + 2\,fnu\ P13\ P23) \\
& + P13\ P23\,(2\,fnu\ P11\ P21 + 2\,P13\ P23 + 2\,fnu\ P12\ P22) \\
& + P22\ P11\ G12surA\,(P12\ P21 + P22\ P11) \\
& + P13\ P21\ G13surA\,(P13\ P21 + P23\ P11)
\end{aligned}
$$

```
>  fortran(C44);

      t0 =
P13*P22*G23surA*(P13*P22+P23*P12)+P12*P22*(2*P12*P22+2*fnu*P1

#1*P21+2*fnu*P13*P23)+P23*P12*G23surA*(P13*P22+P23*P12)+P23*P11*G13

#surA*(P13*P21+P23*P11)+P12*P21*G12surA*(P12*P21+P22*P11)+P11*P21*(

#2*fnu*P12*P22+2*P11*P21+2*fnu*P13*P23)+P13*P23*(2*fnu*P11*P21+2*P1

#3*P23+2*fnu*P12*P22)+P22*P11*G12surA*(P12*P21+P22*P11)+P13*P21*G13

     #surA*(P13*P21+P23*P11)
>  C45:=coeff(sigma_xy,eps_yz);
```

$$
\begin{aligned}
C45 := {} & P22\ P11\ G12surA\,(P32\ P21 + P22\ P31) \\
& + P13\ P21\ G13surA\,(P33\ P21 + P23\ P31) \\
& + P13\ P22\ G23surA\,(P33\ P22 + P23\ P32) \\
& + P12\ P22\,(2\,fnu\ P21\ P31 + 2\,fnu\ P23\ P33 + 2\,P22\ P32) \\
& + P23\ P12\ G23surA\,(P33\ P22 + P23\ P32) \\
& + P11\ P21\,(2\,fnu\ P23\ P33 + 2\,P21\ P31 + 2\,fnu\ P22\ P32) \\
& + P23\ P11\ G13surA\,(P33\ P21 + P23\ P31) \\
& + P13\ P23\,(2\,fnu\ P21\ P31 + 2\,fnu\ P22\ P32 + 2\,P23\ P33) \\
& + P12\ P21\ G12surA\,(P32\ P21 + P22\ P31)
\end{aligned}
$$

```
>  fortran(C45);
```

```
      t0 =
P22*P11*G12surA*(P32*P21+P22*P31)+P13*P21*G13surA*(P33*P21+P2

#3*P31)+P13*P22*G23surA*(P33*P22+P23*P32)+P12*P22*(2*fnu*P21*P31+2*

#fnu*P23*P33+2*P22*P32)+P23*P12*G23surA*(P33*P22+P23*P32)+P11*P21*(

#2*fnu*P23*P33+2*P21*P31+2*fnu*P22*P32)+P23*P11*G13surA*(P33*P21+P2

#3*P31)+P13*P23*(2*fnu*P21*P31+2*fnu*P22*P32+2*P23*P33)+P12*P21*G12

     #surA*(P32*P21+P22*P31)
```

```
> C46:=coeff(sigma_xy,eps_xz);
```

$$
\begin{aligned}
C46 := {} & P11\ P21\ (2\,fnu\ P12\ P32 + 2\,fnu\ P13\ P33 + 2\,P11\ P31) \\
& + P23\ P11\ G13surA\,(P33\ P11 + P13\ P31) \\
& + P13\ P23\ (2\,fnu\ P11\ P31 + 2\,P13\ P33 + 2\,fnu\ P12\ P32) \\
& + P12\ P21\ G12surA\,(P32\ P11 + P12\ P31) \\
& + P22\ P11\ G12surA\,(P32\ P11 + P12\ P31) \\
& + P13\ P21\ G13surA\,(P33\ P11 + P13\ P31) \\
& + P13\ P22\ G23surA\,(P33\ P12 + P13\ P32) \\
& + P12\ P22\ (2\,fnu\ P11\ P31 + 2\,fnu\ P13\ P33 + 2\,P12\ P32) \\
& + P23\ P12\ G23surA\,(P33\ P12 + P13\ P32)
\end{aligned}
$$

```
> fortran(C46);

      t0 =
P11*P21*(2*fnu*P12*P32+2*fnu*P13*P33+2*P11*P31)+P23*P11*G13su

#rA*(P33*P11+P13*P31)+P13*P23*(2*fnu*P11*P31+2*P13*P33+2*fnu*P12*P3

#2)+P12*P21*G12surA*(P32*P11+P12*P31)+P22*P11*G12surA*(P32*P11+P12*

#P31)+P13*P21*G13surA*(P33*P11+P13*P31)+P13*P22*G23surA*(P33*P12+P1

#3*P32)+P12*P22*(2*fnu*P11*P31+2*fnu*P13*P33+2*P12*P32)+P23*P12*G23

     #surA*(P33*P12+P13*P32)
```

```
> sigma_yz:=collect(Sigma_xyz[2,3],[eps_xx,eps_yy,eps_zz,eps_xy,eps_yz,
> eps_xz]);
```

$$
\begin{aligned}
&sigma_yz := (P23\ P31\ G13surA\ P13\ P11 + P32\ P21\ G12surA\ P12\ P11 \\
&+ P22\ P32\ (fnu\ P11^2 + P12^2 + fnu\ P13^2) + P23\ P32\ G23surA\ P13\ P12 \\
&+ P33\ P21\ G13surA\ P13\ P11 + P21\ P31\ (fnu\ P12^2 + P11^2 + fnu\ P13^2) \\
&+ P33\ P22\ G23surA\ P13\ P12 + P22\ P31\ G12surA\ P12\ P11 \\
&+ P23\ P33\ (fnu\ P12^2 + P13^2 + fnu\ P11^2))eps_xx + (P22^2\ P31\ G12surA\ P21 \\
&+ P23\ P33\ (fnu\ P21^2 + P23^2 + fnu\ P22^2) + P23^2\ P31\ G13surA\ P21 \\
&+ P32\ P21^2\ G12surA\ P22 + P22\ P32\ (fnu\ P21^2 + P22^2 + fnu\ P23^2) \\
&+ P21\ P31\ (fnu\ P23^2 + P21^2 + fnu\ P22^2) + P23^2\ P32\ G23surA\ P22 \\
&+ P33\ P22^2\ G23surA\ P23 + P33\ P21^2\ G13surA\ P23)eps_yy + (\\
&P33^2\ P22\ G23surA\ P32 + P33^2\ P21\ G13surA\ P31 + P22\ P31^2\ G12surA\ P32 \\
&+ P23\ P33\ (fnu\ P32^2 + fnu\ P31^2 + P33^2) + P23\ P31^2\ G13surA\ P33 \\
&+ P21\ P31\ (fnu\ P32^2 + fnu\ P33^2 + P31^2) + P32^2\ P21\ G12surA\ P31 \\
&+ P23\ P32^2\ G23surA\ P33 + P22\ P32\ (P32^2 + fnu\ P31^2 + fnu\ P33^2))eps_zz + (\\
&P23\ P32\ G23surA\ (P13\ P22 + P23\ P12) \\
&+ P22\ P32\ (2\ P12\ P22 + 2\ fnu\ P11\ P21 + 2\ fnu\ P13\ P23) \\
&+ P33\ P22\ G23surA\ (P13\ P22 + P23\ P12) \\
&+ P33\ P21\ G13surA\ (P13\ P21 + P23\ P11) \\
&+ P22\ P31\ G12surA\ (P12\ P21 + P22\ P11) \\
&+ P21\ P31\ (2\ fnu\ P12\ P22 + 2\ P11\ P21 + 2\ fnu\ P13\ P23) \\
&+ P23\ P33\ (2\ fnu\ P11\ P21 + 2\ P13\ P23 + 2\ fnu\ P12\ P22) \\
&+ P32\ P21\ G12surA\ (P12\ P21 + P22\ P11) \\
&+ P23\ P31\ G13surA\ (P13\ P21 + P23\ P11))eps_xy + (\\
&P32\ P21\ G12surA\ (P32\ P21 + P22\ P31) \\
&+ P23\ P31\ G13surA\ (P33\ P21 + P23\ P31) \\
&+ P23\ P32\ G23surA\ (P33\ P22 + P23\ P32) \\
&+ P22\ P32\ (2\ fnu\ P21\ P31 + 2\ fnu\ P23\ P33 + 2\ P22\ P32) \\
&+ P33\ P22\ G23surA\ (P33\ P22 + P23\ P32) \\
&+ P21\ P31\ (2\ fnu\ P23\ P33 + 2\ P21\ P31 + 2\ fnu\ P22\ P32) \\
&+ P33\ P21\ G13surA\ (P33\ P21 + P23\ P31) \\
&+ P23\ P33\ (2\ fnu\ P21\ P31 + 2\ fnu\ P22\ P32 + 2\ P23\ P33) \\
&+ P22\ P31\ G12surA\ (P32\ P21 + P22\ P31))eps_yz + (\\
&P21\ P31\ (2\ fnu\ P12\ P32 + 2\ fnu\ P13\ P33 + 2\ P11\ P31) \\
&+ P33\ P21\ G13surA\ (P33\ P11 + P13\ P31) \\
&+ P23\ P33\ (2\ fnu\ P11\ P31 + 2\ P13\ P33 + 2\ fnu\ P12\ P32) \\
&+ P22\ P31\ G12surA\ (P32\ P11 + P12\ P31) \\
&+ P32\ P21\ G12surA\ (P32\ P11 + P12\ P31) \\
&+ P23\ P31\ G13surA\ (P33\ P11 + P13\ P31) \\
&+ P23\ P32\ G23surA\ (P33\ P12 + P13\ P32) \\
&+ P22\ P32\ (2\ fnu\ P11\ P31 + 2\ fnu\ P13\ P33 + 2\ P12\ P32) \\
&+ P33\ P22\ G23surA\ (P33\ P12 + P13\ P32))eps_xz
\end{aligned}
$$

```
> C51:=coeff(sigma_yz,eps_xx);
```

$$C51 := P23\ P31\ G13surA\ P13\ P11 + P32\ P21\ G12surA\ P12\ P11 + P22\ P32\ (fnu\ P11^2 + P12^2 + fnu\ P13^2) + P23\ P32\ G23surA\ P13\ P12 + P33\ P21\ G13surA\ P13\ P11 + P21\ P31\ (fnu\ P12^2 + P11^2 + fnu\ P13^2) + P33\ P22\ G23surA\ P13\ P12 + P22\ P31\ G12surA\ P12\ P11 + P23\ P33\ (fnu\ P12^2 + P13^2 + fnu\ P11^2)$$

```
> fortran(C51);

      t0 =
P23*P31*G13surA*P13*P11+P32*P21*G12surA*P12*P11+P22*P32*(fnu*

#P11**2+P12**2+fnu*P13**2)+P23*P32*G23surA*P13*P12+P33*P21*G13surA*

#P13*P11+P21*P31*(fnu*P12**2+P11**2+fnu*P13**2)+P33*P22*G23surA*P13

#*P12+P22*P31*G12surA*P12*P11+P23*P33*(fnu*P12**2+P13**2+fnu*P11**2

     #)
> C52:=coeff(sigma_yz,eps_yy);
```

$$C52 := P22^2\ P31\ G12surA\ P21 + P23\ P33\ (fnu\ P21^2 + P23^2 + fnu\ P22^2) + P23^2\ P31\ G13surA\ P21 + P32\ P21^2\ G12surA\ P22 + P22\ P32\ (fnu\ P21^2 + P22^2 + fnu\ P23^2) + P21\ P31\ (fnu\ P23^2 + P21^2 + fnu\ P22^2) + P23^2\ P32\ G23surA\ P22 + P33\ P22^2\ G23surA\ P23 + P33\ P21^2\ G13surA\ P23$$

```
> fortran(C52);

      t0 =
P22**2*P31*G12surA*P21+P23*P33*(fnu*P21**2+P23**2+fnu*P22**2)

#+P23**2*P31*G13surA*P21+P32*P21**2*G12surA*P22+P22*P32*(fnu*P21**2

#+P22**2+fnu*P23**2)+P21*P31*(fnu*P23**2+P21**2+fnu*P22**2)+P23**2*

     #P32*G23surA*P22+P33*P22**2*G23surA*P23+P33*P21**2*G13surA*P23
> C53:=coeff(sigma_yz,eps_zz);
```

$$C53 := P33^2\ P22\ G23surA\ P32 + P33^2\ P21\ G13surA\ P31 + P22\ P31^2\ G12surA\ P32 + P23\ P33\ (fnu\ P32^2 + fnu\ P31^2 + P33^2) + P23\ P31^2\ G13surA\ P33 + P21\ P31\ (fnu\ P32^2 + fnu\ P33^2 + P31^2) + P32^2\ P21\ G12surA\ P31 + P23\ P32^2\ G23surA\ P33 + P22\ P32\ (P32^2 + fnu\ P31^2 + fnu\ P33^2)$$

```
> fortran(C53);

      t0 =
P33**2*P22*G23surA*P32+P33**2*P21*G13surA*P31+P22*P31**2*G12s
```

```
#urA*P32+P23*P33*(fnu*P32**2+fnu*P31**2+P33**2)+P23*P31**2*G13surA*

#P33+P21*P31*(fnu*P32**2+fnu*P33**2+P31**2)+P32**2*P21*G12surA*P31+

     #P23*P32**2*G23surA*P33+P22*P32*(P32**2+fnu*P31**2+fnu*P33**2)
```

```
> C54:=coeff(sigma_yz,eps_xy);
```

$$
\begin{aligned}
C54 := {} & P23\ P32\ G23surA\,(P13\ P22 + P23\ P12) \\
& + P22\ P32\,(2\,P12\ P22 + 2\,fnu\ P11\ P21 + 2\,fnu\ P13\ P23) \\
& + P33\ P22\ G23surA\,(P13\ P22 + P23\ P12) \\
& + P33\ P21\ G13surA\,(P13\ P21 + P23\ P11) \\
& + P22\ P31\ G12surA\,(P12\ P21 + P22\ P11) \\
& + P21\ P31\,(2\,fnu\ P12\ P22 + 2\,P11\ P21 + 2\,fnu\ P13\ P23) \\
& + P23\ P33\,(2\,fnu\ P11\ P21 + 2\,P13\ P23 + 2\,fnu\ P12\ P22) \\
& + P32\ P21\ G12surA\,(P12\ P21 + P22\ P11) \\
& + P23\ P31\ G13surA\,(P13\ P21 + P23\ P11)
\end{aligned}
$$

```
> fortran(C54);

      t0 =
P23*P32*G23surA*(P13*P22+P23*P12)+P22*P32*(2*P12*P22+2*fnu*P1

#1*P21+2*fnu*P13*P23)+P33*P22*G23surA*(P13*P22+P23*P12)+P33*P21*G13

#surA*(P13*P21+P23*P11)+P22*P31*G12surA*(P12*P21+P22*P11)+P21*P31*(

#2*fnu*P12*P22+2*P11*P21+2*fnu*P13*P23)+P23*P33*(2*fnu*P11*P21+2*P1

#3*P23+2*fnu*P12*P22)+P32*P21*G12surA*(P12*P21+P22*P11)+P23*P31*G13

     #surA*(P13*P21+P23*P11)
```

```
> C55:=coeff(sigma_yz,eps_yz);
```

$$
\begin{aligned}
C55 := {} & P32\ P21\ G12surA\,(P32\ P21 + P22\ P31) \\
& + P23\ P31\ G13surA\,(P33\ P21 + P23\ P31) \\
& + P23\ P32\ G23surA\,(P33\ P22 + P23\ P32) \\
& + P22\ P32\,(2\,fnu\ P21\ P31 + 2\,fnu\ P23\ P33 + 2\,P22\ P32) \\
& + P33\ P22\ G23surA\,(P33\ P22 + P23\ P32) \\
& + P21\ P31\,(2\,fnu\ P23\ P33 + 2\,P21\ P31 + 2\,fnu\ P22\ P32) \\
& + P33\ P21\ G13surA\,(P33\ P21 + P23\ P31) \\
& + P23\ P33\,(2\,fnu\ P21\ P31 + 2\,fnu\ P22\ P32 + 2\,P23\ P33) \\
& + P22\ P31\ G12surA\,(P32\ P21 + P22\ P31)
\end{aligned}
$$

```
> fortran(C55);
```

```
      t0 =
P32*P21*G12surA*(P32*P21+P22*P31)+P23*P31*G13surA*(P33*P21+P2

#3*P31)+P23*P32*G23surA*(P33*P22+P23*P32)+P22*P32*(2*fnu*P21*P31+2*

#fnu*P23*P33+2*P22*P32)+P33*P22*G23surA*(P33*P22+P23*P32)+P21*P31*(

#2*fnu*P23*P33+2*P21*P31+2*fnu*P22*P32)+P33*P21*G13surA*(P33*P21+P2

#3*P31)+P23*P33*(2*fnu*P21*P31+2*fnu*P22*P32+2*P23*P33)+P22*P31*G12

     #surA*(P32*P21+P22*P31)
```

```
> C56:=coeff(sigma_yz,eps_xz);
```

$$
\begin{aligned}
C56 := {} & P21\ P31\ (2\,fnu\ P12\ P32 + 2\,fnu\ P13\ P33 + 2\,P11\ P31) \\
& + P33\ P21\ G13surA\,(P33\ P11 + P13\ P31) \\
& + P23\ P33\ (2\,fnu\ P11\ P31 + 2\,P13\ P33 + 2\,fnu\ P12\ P32) \\
& + P22\ P31\ G12surA\,(P32\ P11 + P12\ P31) \\
& + P32\ P21\ G12surA\,(P32\ P11 + P12\ P31) \\
& + P23\ P31\ G13surA\,(P33\ P11 + P13\ P31) \\
& + P23\ P32\ G23surA\,(P33\ P12 + P13\ P32) \\
& + P22\ P32\ (2\,fnu\ P11\ P31 + 2\,fnu\ P13\ P33 + 2\,P12\ P32) \\
& + P33\ P22\ G23surA\,(P33\ P12 + P13\ P32)
\end{aligned}
$$

```
> fortran(C56);

      t0 =
P21*P31*(2*fnu*P12*P32+2*fnu*P13*P33+2*P11*P31)+P33*P21*G13su

#rA*(P33*P11+P13*P31)+P23*P33*(2*fnu*P11*P31+2*P13*P33+2*fnu*P12*P3

#2)+P22*P31*G12surA*(P32*P11+P12*P31)+P32*P21*G12surA*(P32*P11+P12*

#P31)+P23*P31*G13surA*(P33*P11+P13*P31)+P23*P32*G23surA*(P33*P12+P1

#3*P32)+P22*P32*(2*fnu*P11*P31+2*fnu*P13*P33+2*P12*P32)+P33*P22*G23

     #surA*(P33*P12+P13*P32)
```

```
> sigma_xz:=collect(Sigma_xyz[1,3],[eps_xx,eps_yy,eps_zz,eps_xy,eps_yz,
> eps_xz]);
```

$$
\begin{aligned}
&sigma_xz := (P13^2\ P31\ G13surA\ P11 + P32\ P11^2\ G12surA\ P12 \\
&+ P12\ P32\ (fnu\ P11^2 + P12^2 + fnu\ P13^2) + P13^2\ P32\ G23surA\ P12 \\
&+ P33\ P11^2\ G13surA\ P13 + P11\ P31\ (fnu\ P12^2 + P11^2 + fnu\ P13^2) \\
&+ P33\ P12^2\ G23surA\ P13 + P12^2\ P31\ G12surA\ P11 \\
&+ P13\ P33\ (fnu\ P12^2 + P13^2 + fnu\ P11^2))eps_xx + (P12\ P31\ G12surA\ P21\ P22 \\
&+ P13\ P33\ (fnu\ P21^2 + P23^2 + fnu\ P22^2) + P13\ P31\ G13surA\ P21\ P23 \\
&+ P32\ P11\ G12surA\ P21\ P22 + P12\ P32\ (fnu\ P21^2 + P22^2 + fnu\ P23^2) \\
&+ P11\ P31\ (fnu\ P23^2 + P21^2 + fnu\ P22^2) + P13\ P32\ G23surA\ P22\ P23 \\
&+ P33\ P12\ G23surA\ P22\ P23 + P33\ P11\ G13surA\ P21\ P23)eps_yy + (\\
&P33^2\ P12\ G23surA\ P32 + P33^2\ P11\ G13surA\ P31 + P12\ P31^2\ G12surA\ P32 \\
&+ P13\ P33\ (fnu\ P32^2 + fnu\ P31^2 + P33^2) + P13\ P31^2\ G13surA\ P33 \\
&+ P11\ P31\ (fnu\ P32^2 + fnu\ P33^2 + P31^2) + P32^2\ P11\ G12surA\ P31 \\
&+ P13\ P32^2\ G23surA\ P33 + P12\ P32\ (P32^2 + fnu\ P31^2 + fnu\ P33^2))eps_zz + (\\
&P13\ P32\ G23surA\ (P13\ P22 + P23\ P12) \\
&+ P12\ P32\ (2\ P12\ P22 + 2\ fnu\ P11\ P21 + 2\ fnu\ P13\ P23) \\
&+ P33\ P12\ G23surA\ (P13\ P22 + P23\ P12) \\
&+ P33\ P11\ G13surA\ (P13\ P21 + P23\ P11) \\
&+ P12\ P31\ G12surA\ (P12\ P21 + P22\ P11) \\
&+ P11\ P31\ (2\ fnu\ P12\ P22 + 2\ P11\ P21 + 2\ fnu\ P13\ P23) \\
&+ P13\ P33\ (2\ fnu\ P11\ P21 + 2\ P13\ P23 + 2\ fnu\ P12\ P22) \\
&+ P32\ P11\ G12surA\ (P12\ P21 + P22\ P11) \\
&+ P13\ P31\ G13surA\ (P13\ P21 + P23\ P11))eps_xy + (\\
&P32\ P11\ G12surA\ (P32\ P21 + P22\ P31) \\
&+ P13\ P31\ G13surA\ (P33\ P21 + P23\ P31) \\
&+ P13\ P32\ G23surA\ (P33\ P22 + P23\ P32) \\
&+ P12\ P32\ (2\ fnu\ P21\ P31 + 2\ fnu\ P23\ P33 + 2\ P22\ P32) \\
&+ P33\ P12\ G23surA\ (P33\ P22 + P23\ P32) \\
&+ P11\ P31\ (2\ fnu\ P23\ P33 + 2\ P21\ P31 + 2\ fnu\ P22\ P32) \\
&+ P33\ P11\ G13surA\ (P33\ P21 + P23\ P31) \\
&+ P13\ P33\ (2\ fnu\ P21\ P31 + 2\ fnu\ P22\ P32 + 2\ P23\ P33) \\
&+ P12\ P31\ G12surA\ (P32\ P21 + P22\ P31))eps_yz + (\\
&P11\ P31\ (2\ fnu\ P12\ P32 + 2\ fnu\ P13\ P33 + 2\ P11\ P31) \\
&+ P33\ P11\ G13surA\ (P33\ P11 + P13\ P31) \\
&+ P13\ P33\ (2\ fnu\ P11\ P31 + 2\ P13\ P33 + 2\ fnu\ P12\ P32) \\
&+ P12\ P31\ G12surA\ (P32\ P11 + P12\ P31) \\
&+ P32\ P11\ G12surA\ (P32\ P11 + P12\ P31) \\
&+ P13\ P31\ G13surA\ (P33\ P11 + P13\ P31) \\
&+ P13\ P32\ G23surA\ (P33\ P12 + P13\ P32) \\
&+ P12\ P32\ (2\ fnu\ P11\ P31 + 2\ fnu\ P13\ P33 + 2\ P12\ P32) \\
&+ P33\ P12\ G23surA\ (P33\ P12 + P13\ P32))eps_xz
\end{aligned}
$$

```
> C61:=coeff(sigma_xz,eps_xx);
```

$$C61 := P13^2\ P31\ G13surA\ P11 + P32\ P11^2\ G12surA\ P12 + P12\ P32\ (fnu\ P11^2 + P12^2 + fnu\ P13^2) + P13^2\ P32\ G23surA\ P12 + P33\ P11^2\ G13surA\ P13 + P11\ P31\ (fnu\ P12^2 + P11^2 + fnu\ P13^2) + P33\ P12^2\ G23surA\ P13 + P12^2\ P31\ G12surA\ P11 + P13\ P33\ (fnu\ P12^2 + P13^2 + fnu\ P11^2)$$

```
> fortran(C61);

      t0 =
P13**2*P31*G13surA*P11+P32*P11**2*G12surA*P12+P12*P32*(fnu*P1

#1**2+P12**2+fnu*P13**2)+P13**2*P32*G23surA*P12+P33*P11**2*G13surA*

#P13+P11*P31*(fnu*P12**2+P11**2+fnu*P13**2)+P33*P12**2*G23surA*P13+

     #P12**2*P31*G12surA*P11+P13*P33*(fnu*P12**2+P13**2+fnu*P11**2)
> C62:=coeff(sigma_xz,eps_yy);
```

$$C62 := P12\ P31\ G12surA\ P21\ P22 + P13\ P33\ (fnu\ P21^2 + P23^2 + fnu\ P22^2) + P13\ P31\ G13surA\ P21\ P23 + P32\ P11\ G12surA\ P21\ P22 + P12\ P32\ (fnu\ P21^2 + P22^2 + fnu\ P23^2) + P11\ P31\ (fnu\ P23^2 + P21^2 + fnu\ P22^2) + P13\ P32\ G23surA\ P22\ P23 + P33\ P12\ G23surA\ P22\ P23 + P33\ P11\ G13surA\ P21\ P23$$

```
> fortran(C62);

      t0 =
P12*P31*G12surA*P21*P22+P13*P33*(fnu*P21**2+P23**2+fnu*P22**2

#)+P13*P31*G13surA*P21*P23+P32*P11*G12surA*P21*P22+P12*P32*(fnu*P21

#**2+P22**2+fnu*P23**2)+P11*P31*(fnu*P23**2+P21**2+fnu*P22**2)+P13*

#P32*G23surA*P22*P23+P33*P12*G23surA*P22*P23+P33*P11*G13surA*P21*P2

     #3
> C63:=coeff(sigma_xz,eps_zz);
```

$$C63 := P33^2\ P12\ G23surA\ P32 + P33^2\ P11\ G13surA\ P31 + P12\ P31^2\ G12surA\ P32 + P13\ P33\ (fnu\ P32^2 + fnu\ P31^2 + P33^2) + P13\ P31^2\ G13surA\ P33 + P11\ P31\ (fnu\ P32^2 + fnu\ P33^2 + P31^2) + P32^2\ P11\ G12surA\ P31 + P13\ P32^2\ G23surA\ P33 + P12\ P32\ (P32^2 + fnu\ P31^2 + fnu\ P33^2)$$

```
> fortran(C63);

      t0 =
P33**2*P12*G23surA*P32+P33**2*P11*G13surA*P31+P12*P31**2*G12s
```

```
#urA*P32+P13*P33*(fnu*P32**2+fnu*P31**2+P33**2)+P13*P31**2*G13surA*

#P33+P11*P31*(fnu*P32**2+fnu*P33**2+P31**2)+P32**2*P11*G12surA*P31+

     #P13*P32**2*G23surA*P33+P12*P32*(P32**2+fnu*P31**2+fnu*P33**2)
```

```
> C64:=coeff(sigma_xz,eps_xy);
```

$$
\begin{aligned}
C64 := {} & P13\ P32\ G23surA\,(P13\ P22 + P23\ P12) \\
& + P12\ P32\,(2\,P12\ P22 + 2\,fnu\ P11\ P21 + 2\,fnu\ P13\ P23) \\
& + P33\ P12\ G23surA\,(P13\ P22 + P23\ P12) \\
& + P33\ P11\ G13surA\,(P13\ P21 + P23\ P11) \\
& + P12\ P31\ G12surA\,(P12\ P21 + P22\ P11) \\
& + P11\ P31\,(2\,fnu\ P12\ P22 + 2\,P11\ P21 + 2\,fnu\ P13\ P23) \\
& + P13\ P33\,(2\,fnu\ P11\ P21 + 2\,P13\ P23 + 2\,fnu\ P12\ P22) \\
& + P32\ P11\ G12surA\,(P12\ P21 + P22\ P11) \\
& + P13\ P31\ G13surA\,(P13\ P21 + P23\ P11)
\end{aligned}
$$

```
> fortran(C64);

      t0 =
P13*P32*G23surA*(P13*P22+P23*P12)+P12*P32*(2*P12*P22+2*fnu*P1

#1*P21+2*fnu*P13*P23)+P33*P12*G23surA*(P13*P22+P23*P12)+P33*P11*G13

#surA*(P13*P21+P23*P11)+P12*P31*G12surA*(P12*P21+P22*P11)+P11*P31*(

#2*fnu*P12*P22+2*P11*P21+2*fnu*P13*P23)+P13*P33*(2*fnu*P11*P21+2*P1

#3*P23+2*fnu*P12*P22)+P32*P11*G12surA*(P12*P21+P22*P11)+P13*P31*G13

     #surA*(P13*P21+P23*P11)
```

```
> C65:=coeff(sigma_xz,eps_yz);
```

$$
\begin{aligned}
C65 := {} & P32\ P11\ G12surA\,(P32\ P21 + P22\ P31) \\
& + P13\ P31\ G13surA\,(P33\ P21 + P23\ P31) \\
& + P13\ P32\ G23surA\,(P33\ P22 + P23\ P32) \\
& + P12\ P32\,(2\,fnu\ P21\ P31 + 2\,fnu\ P23\ P33 + 2\,P22\ P32) \\
& + P33\ P12\ G23surA\,(P33\ P22 + P23\ P32) \\
& + P11\ P31\,(2\,fnu\ P23\ P33 + 2\,P21\ P31 + 2\,fnu\ P22\ P32) \\
& + P33\ P11\ G13surA\,(P33\ P21 + P23\ P31) \\
& + P13\ P33\,(2\,fnu\ P21\ P31 + 2\,fnu\ P22\ P32 + 2\,P23\ P33) \\
& + P12\ P31\ G12surA\,(P32\ P21 + P22\ P31)
\end{aligned}
$$

```
> fortran(C65);
```

```
      t0 =
P32*P11*G12surA*(P32*P21+P22*P31)+P13*P31*G13surA*(P33*P21+P2

#3*P31)+P13*P32*G23surA*(P33*P22+P23*P32)+P12*P32*(2*fnu*P21*P31+2*

#fnu*P23*P33+2*P22*P32)+P33*P12*G23surA*(P33*P22+P23*P32)+P11*P31*(

#2*fnu*P23*P33+2*P21*P31+2*fnu*P22*P32)+P33*P11*G13surA*(P33*P21+P2

#3*P31)+P13*P33*(2*fnu*P21*P31+2*fnu*P22*P32+2*P23*P33)+P12*P31*G12

     #surA*(P32*P21+P22*P31)
> C66:=coeff(sigma_xz,eps_xz);
```

$$
\begin{aligned}
C66 := {} & P11\ P31\ (2\,fnu\ P12\ P32 + 2\,fnu\ P13\ P33 + 2\,P11\ P31) \\
& + P33\ P11\ G13surA\,(P33\ P11 + P13\ P31) \\
& + P13\ P33\ (2\,fnu\ P11\ P31 + 2\,P13\ P33 + 2\,fnu\ P12\ P32) \\
& + P12\ P31\ G12surA\,(P32\ P11 + P12\ P31) \\
& + P32\ P11\ G12surA\,(P32\ P11 + P12\ P31) \\
& + P13\ P31\ G13surA\,(P33\ P11 + P13\ P31) \\
& + P13\ P32\ G23surA\,(P33\ P12 + P13\ P32) \\
& + P12\ P32\ (2\,fnu\ P11\ P31 + 2\,fnu\ P13\ P33 + 2\,P12\ P32) \\
& + P33\ P12\ G23surA\,(P33\ P12 + P13\ P32)
\end{aligned}
$$

```
> fortran(C66);

      t0 =
P11*P31*(2*fnu*P12*P32+2*fnu*P13*P33+2*P11*P31)+P33*P11*G13su

#rA*(P33*P11+P13*P31)+P13*P33*(2*fnu*P11*P31+2*P13*P33+2*fnu*P12*P3

#2)+P12*P31*G12surA*(P32*P11+P12*P31)+P32*P11*G12surA*(P32*P11+P12*

#P31)+P13*P31*G13surA*(P33*P11+P13*P31)+P13*P32*G23surA*(P33*P12+P1

#3*P32)+P12*P32*(2*fnu*P11*P31+2*fnu*P13*P33+2*P12*P32)+P33*P12*G23

     #surA*(P33*P12+P13*P32)
> C12;
```

$$
\begin{aligned}
& P11^2\,(fnu\ P23^2 + P21^2 + fnu\ P22^2) + 2\,P11\ P12\ G12surA\ P22\ P21 \\
& + 2\,P11\ P13\ G13surA\ P23\ P21 + P12^2\,(fnu\ P21^2 + P22^2 + fnu\ P23^2) \\
& + 2\,P12\ P13\ G23surA\ P23\ P22 + P13^2\,(fnu\ P21^2 + P23^2 + fnu\ P22^2)
\end{aligned}
$$

```
> C21;
```

$$P21^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11\,P12\,G12surA\,P22\,P21 + 2\,P11\,P13\,G13surA\,P23\,P21 + P22^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2) + 2\,P12\,P13\,G23surA\,P23\,P22 + P23^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2)$$

la matrice C est symétrique : C12=C21 ...

```
> C11;
```

$$P11^2\,(fnu\,P12^2 + P11^2 + fnu\,P13^2) + 2\,P11^2\,P12^2\,G12surA + 2\,P11^2\,P13^2\,G13surA + P12^2\,(fnu\,P11^2 + P12^2 + fnu\,P13^2) + 2\,P12^2\,P13^2\,G23surA + P13^2\,(fnu\,P12^2 + P13^2 + fnu\,P11^2)$$

```
> C22;
```

$$P21^2\,(fnu\,P23^2 + P21^2 + fnu\,P22^2) + 2\,P21^2\,P22^2\,G12surA + 2\,P21^2\,P23^2\,G13surA + P22^2\,(fnu\,P21^2 + P22^2 + fnu\,P23^2) + 2\,P22^2\,P23^2\,G23surA + P23^2\,(fnu\,P21^2 + P23^2 + fnu\,P22^2)$$

Invariance par rotation theta suivant vecteur tangent au coe de pond dans la base locale :

```
> P_inv:=Matrix([[1,0,0],[0,cos(theta),-sin(theta)],[0,sin(theta),cos(t
> heta)]]);
```

$$P_inv := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

```
> P_inv_trans:=Matrix([[1, 0, 0], [0, cos(theta), sin(theta)], [0,
> -sin(theta), cos(theta)]]);
```

$$P_inv_trans := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

```
> sigma_loc:=Matrix([[sigma11_loc,sigma12_loc,sigma13_loc],[sigma12_loc
> ,sigma22_loc,sigma23_loc],[sigma13_loc,sigma23_loc,sigma33_loc]]);
```

$$sigma_loc := \begin{bmatrix} sigma11_loc & sigma12_loc & sigma13_loc \\ sigma12_loc & sigma22_loc & sigma23_loc \\ sigma13_loc & sigma23_loc & sigma33_loc \end{bmatrix}$$

```
> epsilon_loc:=Matrix([[epsilon11_loc,epsilon12_loc,epsilon13_loc],[eps
> ilon12_loc,epsilon22_loc,epsilon23_loc],[epsilon13_loc,epsilon23_loc,e
> psilon33_loc]]);
```

$$epsilon_loc := \begin{bmatrix} epsilon11_loc & epsilon12_loc & epsilon13_loc \\ epsilon12_loc & epsilon22_loc & epsilon23_loc \\ epsilon13_loc & epsilon23_loc & epsilon33_loc \end{bmatrix}$$

```
> epsilon_new:=Matrix([[epsilon11_new,epsilon12_new,epsilon13_new],[eps
> ilon12_new,epsilon22_new,epsilon23_new],[epsilon13_new,epsilon23_new,e
> psilon33_new]]);
```

$$epsilon_new := \begin{bmatrix} epsilon11_new & epsilon12_new & epsilon13_new \\ epsilon12_new & epsilon22_new & epsilon23_new \\ epsilon13_new & epsilon23_new & epsilon33_new \end{bmatrix}$$

```
> epsilon_inv:=map(expand,P_inv_trans.epsilon_new.P_inv);
```

$$epsilon_inv :=$$
$$[epsilon11_new\,,\ \cos(\theta)\ epsilon12_new + \sin(\theta)\ epsilon13_new\,,$$
$$-\sin(\theta)\ epsilon12_new + \cos(\theta)\ epsilon13_new]$$
$$\Big[\cos(\theta)\ epsilon12_new + \sin(\theta)\ epsilon13_new\,,$$
$$\cos(\theta)^2\ epsilon22_new + 2\sin(\theta)\cos(\theta)\ epsilon23_new + \sin(\theta)^2\ epsilon33_new\,,$$
$$-\sin(\theta)\cos(\theta)\ epsilon22_new - \sin(\theta)^2\ epsilon23_new + \cos(\theta)^2\ epsilon23_new$$
$$+ \cos(\theta)\sin(\theta)\ epsilon33_new\Big]$$
$$\Big[-\sin(\theta)\ epsilon12_new + \cos(\theta)\ epsilon13_new\,, -\sin(\theta)\cos(\theta)\ epsilon22_new$$
$$- \sin(\theta)^2\ epsilon23_new + \cos(\theta)^2\ epsilon23_new + \cos(\theta)\sin(\theta)\ epsilon33_new$$
$$,\ \sin(\theta)^2\ epsilon22_new - 2\sin(\theta)\cos(\theta)\ epsilon23_new + \cos(\theta)^2\ epsilon33_new\Big]$$

```
> epsilon11_loc:=collect(epsilon_inv[1,1],[epsilon12_new,epsilon23_new,
> epsilon13_new]);
```

$$epsilon11_loc := epsilon11_new$$

```
> epsilon22_loc:=collect(epsilon_inv[2,2],[epsilon12_new,epsilon23_new,
> epsilon13_new]);
```

$$epsilon22_loc :=$$
$$\cos(\theta)^2\ epsilon22_new + 2\sin(\theta)\cos(\theta)\ epsilon23_new + \sin(\theta)^2\ epsilon33_new$$

```
> epsilon33_loc:=collect(epsilon_inv[3,3],[epsilon12_new,epsilon23_new,
> epsilon13_new]);
```

$$epsilon33_loc :=$$
$$\sin(\theta)^2\ epsilon22_new - 2\sin(\theta)\cos(\theta)\ epsilon23_new + \cos(\theta)^2\ epsilon33_new$$

```
> epsilon12_loc:=collect(epsilon_inv[1,2],[epsilon12_new,epsilon23_new,
> epsilon13_new]);
```

$$epsilon12_loc := \cos(\theta)\ epsilon12_new + \sin(\theta)\ epsilon13_new$$

```
> epsilon23_loc:=collect(epsilon_inv[2,3],[epsilon12_new,epsilon23_new,
> epsilon13_new]);
```

$$epsilon23_loc := (-\sin(\theta)^2 + \cos(\theta)^2)\ epsilon23_new - \sin(\theta)\cos(\theta)\ epsilon22_new$$
$$+ \cos(\theta)\sin(\theta)\ epsilon33_new$$

```
> epsilon13_loc:=collect(epsilon_inv[1,3],[epsilon12_new,epsilon23_new,
> epsilon13_new]);
```

$$epsilon13_loc := -\sin(\theta)\ epsilon12_new + \cos(\theta)\ epsilon13_new$$

Expression de Sigma_new en fonction des Sigma_loc dans la base locale

```
> Sigma_new:=map(expand,P_inv.sigma_loc.P_inv_trans);
```

$$Sigma_new :=$$
$$[sigma11_loc\,,\ \cos(\theta)\ sigma12_loc - \sin(\theta)\ sigma13_loc\,,$$
$$\sin(\theta)\ sigma12_loc + \cos(\theta)\ sigma13_loc]$$
$$\Big[\cos(\theta)\ sigma12_loc - \sin(\theta)\ sigma13_loc\,,$$
$$\cos(\theta)^2\ sigma22_loc - 2\sin(\theta)\cos(\theta)\ sigma23_loc + \sin(\theta)^2\ sigma33_loc\,,$$
$$\sin(\theta)\cos(\theta)\ sigma22_loc - \sin(\theta)^2\ sigma23_loc + \cos(\theta)^2\ sigma23_loc$$
$$- \cos(\theta)\sin(\theta)\ sigma33_loc\Big]$$
$$\Big[\sin(\theta)\ sigma12_loc + \cos(\theta)\ sigma13_loc\,, \sin(\theta)\cos(\theta)\ sigma22_loc$$
$$- \sin(\theta)^2\ sigma23_loc + \cos(\theta)^2\ sigma23_loc - \cos(\theta)\sin(\theta)\ sigma33_loc,$$
$$\sin(\theta)^2\ sigma22_loc + 2\sin(\theta)\cos(\theta)\ sigma23_loc + \cos(\theta)^2\ sigma33_loc\Big]$$

```
> sigma11_loc:=epsilon11_loc+fnu*epsilon22_loc+fnu*epsilon33_loc;
```

$$sigma11_loc := epsilon11_new + fnu$$
$$(\cos(\theta)^2\ epsilon22_new + 2\sin(\theta)\cos(\theta)\ epsilon23_new + \sin(\theta)^2\ epsilon33_new)$$
$$+ fnu$$
$$(\sin(\theta)^2\ epsilon22_new - 2\sin(\theta)\cos(\theta)\ epsilon23_new + \cos(\theta)^2\ epsilon33_new)$$

```
> sigma22_loc:=fnu*epsilon11_loc+epsilon22_loc+fnu*epsilon33_loc;
```

$$sigma22_loc := fnu\ epsilon11_new + \cos(\theta)^2\ epsilon22_new$$
$$+ 2\sin(\theta)\cos(\theta)\ epsilon23_new + \sin(\theta)^2\ epsilon33_new + fnu$$
$$(\sin(\theta)^2\ epsilon22_new - 2\sin(\theta)\cos(\theta)\ epsilon23_new + \cos(\theta)^2\ epsilon33_new)$$

```
> sigma33_loc:=fnu*epsilon11_loc+fnu*epsilon22_loc+epsilon33_loc;
```

$$sigma33_loc := fnu\ epsilon11_new + fnu$$
$$(\cos(\theta)^2\ epsilon22_new + 2\sin(\theta)\cos(\theta)\ epsilon23_new + \sin(\theta)^2\ epsilon33_new)$$
$$+ \sin(\theta)^2\ epsilon22_new - 2\sin(\theta)\cos(\theta)\ epsilon23_new$$
$$+ \cos(\theta)^2\ epsilon33_new$$

```
> sigma12_loc:=GsurA12*epsilon12_loc;
```

$$sigma12_loc := GsurA12\ (\cos(\theta)\ epsilon12_new + \sin(\theta)\ epsilon13_new)$$

```
> sigma23_loc:=GsurA23*epsilon23_loc;
```

$$sigma23_loc := GsurA23((-\sin(\theta)^2 + \cos(\theta)^2)\ epsilon23_new$$
$$- \sin(\theta)\cos(\theta)\ epsilon22_new + \cos(\theta)\sin(\theta)\ epsilon33_new)$$

```
> sigma13_loc:=GsurA13*epsilon13_loc;
```

$$sigma13_loc := GsurA13\ (-\sin(\theta)\ epsilon12_new + \cos(\theta)\ epsilon13_new)$$

```
> sigma_new_11:=collect(Sigma_new[1,1],[epsilon11_new,epsilon22_new,eps
> ilon33_new,epsilon12_new,epsilon23_new,epsilon13_new]);
```

$$sigma_new_11 := epsilon11_new + (fnu\cos(\theta)^2 + fnu\sin(\theta)^2)\ epsilon22_new$$
$$+ (fnu\cos(\theta)^2 + fnu\sin(\theta)^2)\ epsilon33_new$$

```
> sigma_new_22:=collect(Sigma_new[2,2],[epsilon11_new,epsilon22_new,eps
> ilon33_new,epsilon12_new,epsilon23_new,epsilon13_new]);
```

$$sigma_new_22 := (fnu\cos(\theta)^2 + fnu\sin(\theta)^2)\ epsilon11_new + (\cos(\theta)^2\ (fnu\sin(\theta)^2 + \cos(\theta)^2) + 2\sin(\theta)^2\cos(\theta)^2\ GsurA23 + \sin(\theta)^2\ (fnu\cos(\theta)^2 + \sin(\theta)^2))epsilon22_new + (\cos(\theta)^2\ (fnu\cos(\theta)^2 + \sin(\theta)^2) - 2\sin(\theta)^2\cos(\theta)^2\ GsurA23 + \sin(\theta)^2\ (fnu\sin(\theta)^2 + \cos(\theta)^2))epsilon33_new + (\cos(\theta)^2\ (-2\,fnu\sin(\theta)\cos(\theta) + 2\sin(\theta)\cos(\theta)) - 2\sin(\theta)\cos(\theta)\ GsurA23\ (-\sin(\theta)^2 + \cos(\theta)^2) + \sin(\theta)^2\ (2\,fnu\sin(\theta)\cos(\theta) - 2\sin(\theta)\cos(\theta)))epsilon23_new$$

```
> sigma_new_33:=collect(Sigma_new[3,3],[epsilon11_new,epsilon22_new,eps
> ilon33_new,epsilon12_new,epsilon23_new,epsilon13_new]);
```

$$sigma_new_33 := (fnu\cos(\theta)^2 + fnu\sin(\theta)^2)\ epsilon11_new + (\cos(\theta)^2\ (fnu\cos(\theta)^2 + \sin(\theta)^2) - 2\sin(\theta)^2\cos(\theta)^2\ GsurA23 + \sin(\theta)^2\ (fnu\sin(\theta)^2 + \cos(\theta)^2))epsilon22_new + (\cos(\theta)^2\ (fnu\sin(\theta)^2 + \cos(\theta)^2) + 2\sin(\theta)^2\cos(\theta)^2\ GsurA23 + \sin(\theta)^2\ (fnu\cos(\theta)^2 + \sin(\theta)^2))epsilon33_new + (\sin(\theta)^2\ (-2\,fnu\sin(\theta)\cos(\theta) + 2\sin(\theta)\cos(\theta)) + 2\sin(\theta)\cos(\theta)\ GsurA23\ (-\sin(\theta)^2 + \cos(\theta)^2) + \cos(\theta)^2\ (2\,fnu\sin(\theta)\cos(\theta) - 2\sin(\theta)\cos(\theta)))epsilon23_new$$

```
> sigma_new_12:=collect(Sigma_new[1,2],[epsilon11_new,epsilon22_new,eps
> ilon33_new,epsilon12_new,epsilon23_new,epsilon13_new]);
```

$$sigma_new_12 := (\cos(\theta)^2\ GsurA12 + \sin(\theta)^2\ GsurA13)\ epsilon12_new + (\cos(\theta)\ GsurA12\sin(\theta) - \sin(\theta)\ GsurA13\cos(\theta))\ epsilon13_new$$

```
> sigma_new_23:=collect(Sigma_new[2,3],[epsilon11_new,epsilon22_new,eps
> ilon33_new,epsilon12_new,epsilon23_new,epsilon13_new]);
```

$$sigma_new_23 := (\sin(\theta)\cos(\theta)\ (fnu\sin(\theta)^2 + \cos(\theta)^2) + \sin(\theta)^3\ GsurA23\cos(\theta) - \cos(\theta)^3\ GsurA23\sin(\theta) - \sin(\theta)\cos(\theta)\ (fnu\cos(\theta)^2 + \sin(\theta)^2))epsilon22_new + (\sin(\theta)\cos(\theta)\ (fnu\cos(\theta)^2 + \sin(\theta)^2) - \sin(\theta)^3\ GsurA23\cos(\theta) + \cos(\theta)^3\ GsurA23\sin(\theta) - \sin(\theta)\cos(\theta)\ (fnu\sin(\theta)^2 + \cos(\theta)^2))epsilon33_new + (\sin(\theta)\cos(\theta)\ (-2\,fnu\sin(\theta)\cos(\theta) + 2\sin(\theta)\cos(\theta)) - \sin(\theta)^2\ GsurA23\ (-\sin(\theta)^2 + \cos(\theta)^2) + \cos(\theta)^2\ GsurA23\ (-\sin(\theta)^2 + \cos(\theta)^2) - \sin(\theta)\cos(\theta)\ (2\,fnu\sin(\theta)\cos(\theta) - 2\sin(\theta)\cos(\theta)))epsilon23_new$$

```
> sigma_new_13:=collect(Sigma_new[1,3],[epsilon11_new,epsilon22_new,eps
> ilon33_new,epsilon12_new,epsilon23_new,epsilon13_new]);
```

$$sigma_new_13 := (\cos(\theta)\ GsurA12\sin(\theta) - \sin(\theta)\ GsurA13\cos(\theta))\ epsilon12_new + (\sin(\theta)^2\ GsurA12 + \cos(\theta)^2\ GsurA13)\ epsilon13_new$$

```
> DET:=Determinant(P);
```

$$DET := P11\ P33\ P22 - P11\ P23\ P32 + P21\ P13\ P32 - P21\ P33\ P12 + P31\ P23\ P12 - P31\ P13\ P22$$

```
> fortran(DET);
```

```
      t0 =
P11*P33*P22-P11*P23*P32+P21*P13*P32-P21*P33*P12+P31*P23*P12-P
```

```
#31*P13*P22
```

Calcul des coefficients pour la discrétisation 2D en contraintes planes

```
> Q:=Matrix([[P11,P12],[P21,P22]]);
```

$$Q := \begin{bmatrix} P11 & P12 \\ P21 & P22 \end{bmatrix}$$

```
> transpose(Q);
```

$$\begin{bmatrix} P11 & P21 \\ P12 & P22 \end{bmatrix}$$

```
> Qtrans:=Matrix([[P11, P21], [P12, P22]]);
```

$$Qtrans := \begin{bmatrix} P11 & P21 \\ P12 & P22 \end{bmatrix}$$

```
> Sigma:=Matrix([[Sigma11,Sigma12],[Sigma12,Sigma22]]);
```

$$\Sigma := \begin{bmatrix} \Sigma 11 & \Sigma 12 \\ \Sigma 12 & \Sigma 22 \end{bmatrix}$$

```
> Epsilon_123:=Matrix([[Epsilon11,Epsilon12],[Epsilon12,Epsilon22]]);
```

$$Epsilon_123 := \begin{bmatrix} E11 & E12 \\ E12 & E22 \end{bmatrix}$$

```
> Epsilon_xyz:=Matrix([[Eps_xx,Eps_xy],[Eps_xy,Eps_yy]]);
```

$$Epsilon_xyz := \begin{bmatrix} Eps_xx & Eps_xy \\ Eps_xy & Eps_yy \end{bmatrix}$$

```
> Epsilon:=map(expand,Qtrans.Epsilon_xyz.Q);
```

$$E := [P11^2\ Eps_xx + 2\ P11\ P21\ Eps_xy + P21^2\ Eps_yy\ ,\ P12\ P11\ Eps_xx + P12\ P21\ Eps_xy + P22\ P11\ Eps_xy + P22\ P21\ Eps_yy]$$
$$[P12\ P11\ Eps_xx + P12\ P21\ Eps_xy + P22\ P11\ Eps_xy + P22\ P21\ Eps_yy\ ,\ P12^2\ Eps_xx + 2\ P12\ P22\ Eps_xy + P22^2\ Eps_yy]$$

```
> Epsilon11:=collect(Epsilon[1,1],[Eps_xy,Eps_yz]);
```

$$E11 := P11^2\ Eps_xx + 2\ P11\ P21\ Eps_xy + P21^2\ Eps_yy$$

```
> Epsilon22:=collect(Epsilon[2,2],[Eps_xy,Eps_yz]);
```

$$E22 := P12^2\ Eps_xx + 2\ P12\ P22\ Eps_xy + P22^2\ Eps_yy$$

```
> Epsilon12:=collect(Epsilon[1,2],[Eps_xy,Eps_yz]);
```

$$E12 := (P12\ P21 + P22\ P11)\ Eps_xy + P12\ P11\ Eps_xx + P22\ P21\ Eps_yy$$

Expression de Sigma2D_xyz en fonction des Sigma dans la base locale

```
> Sigma2D_xyz:=map(expand,Q.Sigma.Qtrans);
```

$$
\begin{aligned}
&\mathit{Sigma2D_xyz} := \\
&[\mathit{P11}^2\,\Sigma 11 + 2\,\mathit{P11}\;\mathit{P12}\,\Sigma 12 + \mathit{P12}^2\,\Sigma 22\,, \\
&\mathit{P21}\;\mathit{P11}\,\Sigma 11 + \mathit{P21}\;\mathit{P12}\,\Sigma 12 + \mathit{P22}\;\mathit{P11}\,\Sigma 12 + \mathit{P22}\;\mathit{P12}\,\Sigma 22] \\
&[\mathit{P21}\;\mathit{P11}\,\Sigma 11 + \mathit{P21}\;\mathit{P12}\,\Sigma 12 + \mathit{P22}\;\mathit{P11}\,\Sigma 12 + \mathit{P22}\;\mathit{P12}\,\Sigma 22\,, \\
&\mathit{P21}^2\,\Sigma 11 + 2\,\mathit{P21}\;\mathit{P22}\,\Sigma 12 + \mathit{P22}^2\,\Sigma 22]
\end{aligned}
$$

```
> Sigma11:=Epsilon11+gnu*Epsilon22;
```

$$
\begin{aligned}
\Sigma 11 := {} & \mathit{P11}^2\;\mathit{Eps_xx} + 2\,\mathit{P11}\;\mathit{P21}\;\mathit{Eps_xy} + \mathit{P21}^2\;\mathit{Eps_yy} \\
& + \mathit{gnu}\,(\mathit{P12}^2\;\mathit{Eps_xx} + 2\,\mathit{P12}\;\mathit{P22}\;\mathit{Eps_xy} + \mathit{P22}^2\;\mathit{Eps_yy})
\end{aligned}
$$

```
> Sigma22:=gnu*Epsilon11+Epsilon22;
```

$$
\begin{aligned}
\Sigma 22 := {} & \mathit{gnu}\,(\mathit{P11}^2\;\mathit{Eps_xx} + 2\,\mathit{P11}\;\mathit{P21}\;\mathit{Eps_xy} + \mathit{P21}^2\;\mathit{Eps_yy}) + \mathit{P12}^2\;\mathit{Eps_xx} \\
& + 2\,\mathit{P12}\;\mathit{P22}\;\mathit{Eps_xy} + \mathit{P22}^2\;\mathit{Eps_yy}
\end{aligned}
$$

```
> Sigma12:=G12surA*Epsilon12;
```

$$\Sigma 12 := \mathit{G12surA}\,((\mathit{P12}\;\mathit{P21} + \mathit{P22}\;\mathit{P11})\;\mathit{Eps_xy} + \mathit{P12}\;\mathit{P11}\;\mathit{Eps_xx} + \mathit{P22}\;\mathit{P21}\;\mathit{Eps_yy})$$

```
> Sigma_xx:=collect(Sigma2D_xyz[1,1],[Eps_xx,Eps_yy,Eps_xy]);
```

$$
\begin{aligned}
&\mathit{Sigma_xx} := (\mathit{P11}^2\,(\mathit{P11}^2 + \mathit{gnu}\;\mathit{P12}^2) + 2\,\mathit{P11}^2\;\mathit{P12}^2\;\mathit{G12surA} + \mathit{P12}^2\,(\mathit{gnu}\;\mathit{P11}^2 + \mathit{P12}^2))\;\mathit{Eps_xx} + (\\
&\mathit{P11}^2\,(\mathit{P21}^2 + \mathit{gnu}\;\mathit{P22}^2) + 2\,\mathit{P11}\;\mathit{P12}\;\mathit{G12surA}\;\mathit{P22}\;\mathit{P21} + \mathit{P12}^2\,(\mathit{gnu}\;\mathit{P21}^2 + \mathit{P22}^2) \\
&)\mathit{Eps_yy} + (\mathit{P11}^2\,(2\,\mathit{P11}\;\mathit{P21} + 2\,\mathit{gnu}\;\mathit{P12}\;\mathit{P22}) \\
&+ 2\,\mathit{P11}\;\mathit{P12}\;\mathit{G12surA}\,(\mathit{P12}\;\mathit{P21} + \mathit{P22}\;\mathit{P11}) \\
&+ \mathit{P12}^2\,(2\,\mathit{gnu}\;\mathit{P11}\;\mathit{P21} + 2\,\mathit{P12}\;\mathit{P22}))\mathit{Eps_xy}
\end{aligned}
$$

```
> C11_2D:=coeff(Sigma_xx,Eps_xx);
```

$$\mathit{C11_2D} := \mathit{P11}^2\,(\mathit{P11}^2 + \mathit{gnu}\;\mathit{P12}^2) + 2\,\mathit{P11}^2\;\mathit{P12}^2\;\mathit{G12surA} + \mathit{P12}^2\,(\mathit{gnu}\;\mathit{P11}^2 + \mathit{P12}^2)$$

```
> fortran(C11_2D);

      t0 =
P11**2*(P11**2+gnu*P12**2)+2*P11**2*P12**2*G12surA+P12**2*(gn

     #u*P11**2+P12**2)
```

```
> C12_2D:=coeff(Sigma_xx,Eps_yy);
```

$$
\begin{aligned}
&\mathit{C12_2D} := \\
&\mathit{P11}^2\,(\mathit{P21}^2 + \mathit{gnu}\;\mathit{P22}^2) + 2\,\mathit{P11}\;\mathit{P12}\;\mathit{G12surA}\;\mathit{P22}\;\mathit{P21} + \mathit{P12}^2\,(\mathit{gnu}\;\mathit{P21}^2 + \mathit{P22}^2)
\end{aligned}
$$

```
> fortran(C12_2D);

      t0 =
P11**2*(P21**2+gnu*P22**2)+2*P11*P12*G12surA*P22*P21+P12**2*(

     #gnu*P21**2+P22**2)
```

```
> C14_2D:=coeff(Sigma_xx,Eps_xy);
```

$$C14_2D := P11^2\,(2\,P11\,P21 + 2\,gnu\,P12\,P22) + 2\,P11\,P12\,G12surA\,(P12\,P21 + P22\,P11) + P12^2\,(2\,gnu\,P11\,P21 + 2\,P12\,P22)$$

```
> fortran(C14_2D);
```

```
      t0 =
P11**2*(2*P11*P21+2*gnu*P12*P22)+2*P11*P12*G12surA*(P12*P21+P
     #22*P11)+P12**2*(2*gnu*P11*P21+2*P12*P22)
```

```
> Sigma_yy:=collect(Sigma2D_xyz[2,2],[Eps_xx,Eps_yy,Eps_xy]);
```

$$Sigma_yy := (P21^2\,(P11^2 + gnu\,P12^2) + 2\,P11\,P12\,G12surA\,P22\,P21 + P22^2\,(gnu\,P11^2 + P12^2))Eps_xx + (P21^2\,(P21^2 + gnu\,P22^2) + 2\,P21^2\,P22^2\,G12surA + P22^2\,(gnu\,P21^2 + P22^2))Eps_yy + (P21^2\,(2\,P11\,P21 + 2\,gnu\,P12\,P22) + 2\,P21\,P22\,G12surA\,(P12\,P21 + P22\,P11) + P22^2\,(2\,gnu\,P11\,P21 + 2\,P12\,P22))Eps_xy$$

```
> C22_2D:=coeff(Sigma_yy,Eps_yy);
```

$$C22_2D := P21^2\,(P21^2 + gnu\,P22^2) + 2\,P21^2\,P22^2\,G12surA + P22^2\,(gnu\,P21^2 + P22^2)$$

```
> fortran(C22_2D);
```

```
      t0 =
P21**2*(P21**2+gnu*P22**2)+2*P21**2*P22**2*G12surA+P22**2*(gn
     #u*P21**2+P22**2)
```

```
> C24_2D:=coeff(Sigma_yy,Eps_xy);
```

$$C24_2D := P21^2\,(2\,P11\,P21 + 2\,gnu\,P12\,P22) + 2\,P21\,P22\,G12surA\,(P12\,P21 + P22\,P11) + P22^2\,(2\,gnu\,P11\,P21 + 2\,P12\,P22)$$

```
> fortran(C24_2D);
```

```
      t0 =
P21**2*(2*P11*P21+2*gnu*P12*P22)+2*P21*P22*G12surA*(P12*P21+P
     #22*P11)+P22**2*(2*gnu*P11*P21+2*P12*P22)
```

```
> Sigma_xy:=collect(Sigma2D_xyz[1,2],[Eps_xx,Eps_yy,Eps_xy]);
```

$$Sigma_xy := (P11\,P21\,(P11^2 + gnu\,P12^2) + P12^2\,P21\,G12surA\,P11 + P22\,P11^2\,G12surA\,P12 + P12\,P22\,(gnu\,P11^2 + P12^2))Eps_xx + (P11\,P21\,(P21^2 + gnu\,P22^2) + P12\,P21^2\,G12surA\,P22 + P22^2\,P11\,G12surA\,P21 + P12\,P22\,(gnu\,P21^2 + P22^2))Eps_yy + (P11\,P21\,(2\,P11\,P21 + 2\,gnu\,P12\,P22) + P12\,P21\,G12surA\,(P12\,P21 + P22\,P11) + P22\,P11\,G12surA\,(P12\,P21 + P22\,P11) + P12\,P22\,(2\,gnu\,P11\,P21 + 2\,P12\,P22))Eps_xy$$

```
> C44_2D:=coeff(Sigma_xy,Eps_xy);
```

$$C44_2D := P11\ P21\ (2\ P11\ P21 + 2\ gnu\ P12\ P22) + P12\ P21\ G12surA\ (P12\ P21 + P22\ P11) + P22\ P11\ G12surA\ (P12\ P21 + P22\ P11) + P12\ P22\ (2\ gnu\ P11\ P21 + 2\ P12\ P22)$$

```
> fortran(C44_2D);

      t0 =
P11*P21*(2*P11*P21+2*gnu*P12*P22)+P12*P21*G12surA*(P12*P21+P2


#2*P11)+P22*P11*G12surA*(P12*P21+P22*P11)+P12*P22*(2*gnu*P11*P21+2*

     #P12*P22)
```

Annexe B

Quaternions et composition de rotations

Cette annexe est tirée de la thèse de A. Leroyer [57].

B.1 Un peu d'histoire

> Son : *Well, Papa, can you multiply triplets ?*
> Father : *No [sadly shaking his head], I can only add and substract them*
> (William Rowan Hamilton, Conversation with his son (1843))

Sir William Rowan Hamilton, mathématicien et astronome irlandais (1805-1865) publia plusieurs travaux fondamentaux sur les principes de l'optique géométrique et sur la dynamique. Nommé professeur d'astronomie à vingt-deux ans, il s'orienta ensuite vers l'étude de l'algèbre. Après la publication en 1835 d'une théorie rigoureuse sur les nombres complexes comme couple de réels, il s'efforça d'étendre cette conception pour des espaces de dimension supérieure. Son travail aboutit en 1843 à la théorie des quaternions qu'il développa dans *Lectures on Quaternions* (1853) et *Elements of Quaternions* (1866), fondés sur une extension à l'espace de la représentation plane des nombres complexes. Hamilton développa alors une théorie différentielle des quaternions qu'il appliqua à la cinématique et à la dynamique. L'enthousiasme et l'espoir trop ambitieux suscités par cette nouvelle théorie ont poussé certains partisans d'Hamilton à créer une *Société pour la diffusion des quaternions* et à mener une lutte aveugle et stérile contre les autres méthodes d'analyse vectorielle qui s'élaboraient dans le même temps.

Un siècle et demi plus tard, on se rend compte que les quaternions n'ont pas l'importance que certains leur accordaient. Pourtant comme nous allons le voir, ils sont un outil mathématique très performant, notamment pour l'étude des rotations dans l'espace.

B.2 L'algèbre des quaternions $\mathbb{H}$

B.2.1 Définition

L'ensemble $\mathbb{H}$ des quaternions est défini par :

Vect(**e**,**i**,**j**,**k**)= { $\mathbf{x}$ / $\mathbf{x} = x_0\mathbf{e} + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k}$, $(x_0, x_1, x_2, x_3) \in \mathbb{R}^4$ }
où (**e**,**i**,**j**,**k**) sont les vecteurs de base.

La table suivante (cf. tab. B.1) lui donne une structure d'algèbre (multiplication associative, distributive par rapport à l'addition et d'élément neutre **e**) non-commutative en général :

x	**e**	**i**	**j**	**k**
e	**e**	**i**	**j**	**k**
i	**i**	-**e**	**k**	-**j**
j	**j**	-**k**	-**e**	**i**
k	**k**	**j**	-**i**	-**e**

TAB. B.1: Table de multiplication "colonne × ligne"

Ainsi, $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -\mathbf{e}$ et $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$

Remarque :
$x_0\mathbf{e}$ est appelé **partie réelle**
$x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k}$ est appelé **partie pure** ; elle est assimilable à un vecteur.
La décomposition partie réelle – partie pure est unique.

Les quaternions peuvent être perçus comme une extension des complexes à l'espace $\mathbb{R}^3$. On peut donc être surpris d'être en présence d'un espace de dimension 4, alors que les complexes — espace de dimension 2 — ont leur représentation dans le plan lui aussi de dimension 2. Remarquons aussi que les produits entre **i**,**j**,**k** sont identiques à ceux obtenus avec le produit vectoriel d'une base orthonormée directe de $\mathbb{R}^3$.

B.2.2 Opérations dans l'algèbre des quaternions $\mathbb{H}$

B.2.2.1 Conjugaison

Le conjugué de $\mathbf{x} \in \mathbb{H}$ est noté $\overline{\mathbf{x}}$. On a la relation suivante :

$$\mathbf{x} = x_0\mathbf{e} + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \qquad \overline{\mathbf{x}} = x_0\mathbf{e} - x_1\mathbf{i} - x_2\mathbf{j} - x_3\mathbf{k}$$

Relation remarquable : $\overline{\mathbf{xy}} = \overline{\mathbf{y}}\,\overline{\mathbf{x}}$

B.2.2.2 Norme

L'algèbre peut être normée "naturellement" par la norme euclidienne en posant

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}\overline{\mathbf{x}}}$$

On vérifie que $\|\mathbf{xy}\| = \|\mathbf{x}\|.\|\mathbf{y}\|$

Tout quaternion $\mathbf{x}$ non-nul a un inverse égal à $\mathbf{x}^{-1} = \dfrac{\overline{\mathbf{x}}}{\|\mathbf{x}\|^2}$

L'algèbre des quaternions a donc une *structure de corps.*

B.2.3 L'ensemble $\mathbb{P}$ des quaternions purs

B.2.3.1 Définition

Un quaternion $\mathbf{x}$ est pur si et seulement si sa partie réelle est nulle.

B.2.3.2 Propriétés

$$\overline{\mathbf{x}} \in \mathbb{P} \Leftrightarrow \overline{\mathbf{x}} = -\mathbf{x}$$

L'ensemble $\mathbb{P}$ des quaternions purs est un sous espace vectoriel de dimension 3. Il est isomorphe à $\mathbb{R}^3$. Associons une base $(\overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$ orthonormée directe "image" de la base des quaternions purs $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. À tout quaternion $\mathbf{q} \in \mathbb{P}$ ($\mathbf{q} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$), on peut associer un vecteur $\overrightarrow{q}$ possédant les mêmes coordonnées dans une base $(\overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$ orthonormée directe de $\mathbb{R}^3$.

B.2.3.3 Notations

Les vecteurs en gras représentent les éléments de $\mathbb{H}$.
Les vecteurs fléchés représentent les éléments de $\mathbb{R}^3$.

Les règles de calculs concernant le produit des quaternions $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ étant identiques au produit vectoriel, on peut facilement démontrer la relation suivante :

$$\forall \mathbf{p}, \mathbf{q} \in \mathbb{P}, \quad \mathbf{p}\,\mathbf{q} = -(\overrightarrow{p}.\overrightarrow{q})\mathbf{e} + \underbrace{\mathbf{p}_{\wedge}\mathbf{q}}$$

quaternion ayant les mêmes coordonnées que le vecteur $\overrightarrow{p} \wedge \overrightarrow{q}$

Et donc, lorsque $\mathbf{p}$ et $\mathbf{q}$ sont purs, on a : Pur $(\mathbf{pq}) = \mathbf{p}_{\wedge}\mathbf{q}$.

B.2.4 L'ensemble $\mathbb{S}$ des quaternions de norme 1

B.2.4.1 Isomorphisme avec GL($\mathbb{R}^3$)

Cet ensemble muni de la loi $\times$ est un sous-groupe des quaternions $\mathbb{H}$. On peut montrer facilement que $\mathbb{S}$ est isomorphe au groupe des rotations de l'espace vectoriel $\mathbb{R}^3$ muni de la loi $\circ$ (composition).

Chaque élément r de $\mathbb{S}$ peut être mis sous la forme : $\mathbf{r} = \cos(\frac{\theta}{2}) + \mathbf{u}\sin(\frac{\theta}{2})$ où $\mathbf{u}$ est un quaternion pur de norme 1.

On peut montrer qu'à toute rotation d'angle θ et d'axe dirigé par le vecteur unitaire $\overrightarrow{u}$ correspond le quaternion $\mathbf{r} = \cos(\frac{\theta}{2}) + \mathbf{u}\sin(\frac{\theta}{2})$:

Soit $(\overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$ une base orthonormée directe de $\mathbb{R}^3$ associée à la base des quaternions purs $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. Pour tout vecteur $\overrightarrow{p} = p_1\overrightarrow{i} + p_2\overrightarrow{j} + p_3\overrightarrow{k}$, le vecteur $\overrightarrow{q}$ image de $\overrightarrow{p}$ par la rotation $(\theta, \overrightarrow{u})$ aura les mêmes coordonnées dans la base $(\overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$ que le quaternion $\mathbf{rpr\bar{}}$ où $\mathbf{p}$ est le quaternion pur associé au vecteur $\overrightarrow{p}$.

On a donc le schéma récapitulatif suivant (Fig. B.1) :

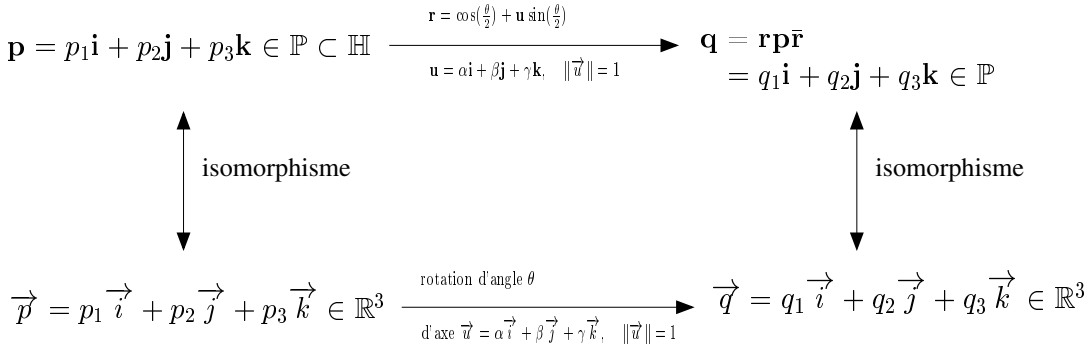


Fig. B.1: Tableau récapitulatif

B.2.4.2 Lien entre quaternion et matrice de passage

Soient $\beta_0 = (\overrightarrow{i_0}, \overrightarrow{j_0}, \overrightarrow{k_0})$ et $\beta_1 = (\overrightarrow{i_1}, \overrightarrow{j_1}, \overrightarrow{k_1})$, deux bases orthonormales directes et $\mathcal{R}_{0\to1}$, la rotation qui transforme la base β_0 en β_1. On note $\mathbf{Q}_{\beta_0\to\beta_1} = (q_0, q_1, q_2, q_3)$ le quaternion représentant la rotation $\mathcal{R}_{0\to1}$ et $P_{\beta_0\to\beta_1}$ la matrice de passage associée à $\mathcal{R}_{0\to1}$ de coefficient générique $P_{i,j}$. Par définition de $P_{\beta_0\to\beta_S}$ (cf. annexe B.4), on a :

$$\overrightarrow{i_1}\Big|_{\beta_0} = P_{\beta_0\to\beta_1}\,\overrightarrow{i_0}\Big|_{\beta_0} \Longrightarrow \overrightarrow{i_1}\Big|_{\beta_0} = P_{1,1}\,\overrightarrow{i_0} + P_{2,1}\,\overrightarrow{j_0} + P_{3,1}\,\overrightarrow{k_0} \tag{B.1}$$

En terme de quaternions, on peut écrire :

$$\begin{aligned}\mathbf{i_1} =& \mathcal{R}_{0\to1}(\mathbf{i_0}) = \mathbf{Q}_{\beta_0\to\beta_1}\mathbf{i_0}\overline{\mathbf{Q}_{\beta_0\to\beta_1}}\\ =& (q_0\mathbf{e} + q_1\mathbf{i_0} + q_2\mathbf{j_0} + q_3\mathbf{k_0})(\mathbf{i_0})(q_0\mathbf{e} - q_1\mathbf{i_0} - q_2\mathbf{j_0} - q_3\mathbf{k_0})\\ =& \left(2\left(q_0^2 + q_1^2\right) - 1\right)\mathbf{i_0} + 2(q_1q_2 + q_0q_3)\mathbf{j_0} + 2(q_1q_3 - q_0q_2)\mathbf{k_0}\end{aligned}$$

Par identification, on obtient :

$$P_{1,1} = 2\left(q_0^2 + q_1^2\right) - 1 \qquad P_{2,1} = 2\left(q_1 q_2 + q_0 q_3\right) \qquad P_{3,1} = 2\left(q_1 q_3 - q_0 q_2\right)$$

Un calcul similaire avec $\mathbf{j_1}$ et $\mathbf{k_1}$ permet d'en déduire l'expression de la matrice de passage :

$$P_{\beta_0 \to \beta_1} = \begin{bmatrix} 2\left(q_0^2 + q_1^2\right) - 1 & 2\left(q_1 q_2 - q_0 q_3\right) & 2\left(q_1 q_3 + q_0 q_2\right) \\ 2\left(q_1 q_2 + q_0 q_3\right) & 2\left(q_0^2 + q_2^2\right) - 1 & 2\left(q_2 q_3 - q_0 q_1\right) \\ 2\left(q_1 q_3 - q_0 q_2\right) & 2\left(q_2 q_3 + q_0 q_1\right) & 2\left(q_0^2 + q_3^2\right) - 1 \end{bmatrix} \tag{B.2}$$

B.2.5 La dérivation temporelle d'un quaternion

Elle est définie comme pour les vecteurs par la dérivée de ses composantes.

$$\frac{d\mathbf{q}}{dt} = \dot{\mathbf{q}} = \dot{q_0}\mathbf{e} + \dot{q_1}\mathbf{i} + \dot{q_2}\mathbf{j} + \dot{q_3}\mathbf{k} \tag{B.3}$$

La notion de dérivation des quaternions est une *notion intrinsèque*. Elle caractérise l'évolution d'une transformation géométrique par rapport à une base donnée. Pour le cas d'un quaternion de norme 1, qui caractérise une rotation, sa dérivée représente la variation de cette rotation dans le temps.

B.2.5.1 Propriété (utile pour la suite)

$$\text{Pour tout quaternion } \mathbf{q} \in \mathbb{S}, \quad \dot{\mathbf{q}}\,\overline{\mathbf{q}} \in \mathbb{P}, \quad \overline{\mathbf{q}}\,\dot{\mathbf{q}} \in \mathbb{P} \tag{B.4}$$

Démonstration :
Soit $\mathbf{q} = q_0\mathbf{e} + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \in \mathbb{S}$
On a alors $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ d'où
$\mathcal{R}e(\dot{\mathbf{q}}\overline{\mathbf{q}}) = \mathcal{R}e(\overline{\mathbf{q}}\dot{\mathbf{q}}) = q_0\dot{q_0} + q_1\dot{q_1} + q_2\dot{q_2} + q_3\dot{q_3} = \dfrac{d(q_0^2 + q_1^2 + q_2^2 + q_3^2)}{dt} = \dfrac{d(1)}{dt} = 0$

B.2.6 Représentation matricielle de $(\mathbf{e}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Il existe des représentations des quaternions par des matrices 4x4 réelles possédant les mêmes propriétés que les quaternions en utilisant le produit matriciel classique. En voici l'une d'entre d'elles :

$$\mathbf{e} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{i} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{j} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad \mathbf{k} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

B.3 Relation entre rotation instantanée et quaternions

Dans l'annexe B de la thèse d'A. Leroyer [57], il est montré que la résolution des équations issues du théorème du moment dynamique posait problème pour un choix de variables classiques (ψ,θ,ϕ). L'objectif est donc de substituer aux 3 angles (ψ,θ,ϕ) décrivant le passage entre le référentiel Galiléen $\mathcal{R}_0$ et le repère principal lié au solide $\mathcal{R}_S$ un quaternion caractérisant cette transformation en vue d'éviter les configurations singulières. Le choix des inconnues du système va donc englober le vecteur instantané de rotation $\overrightarrow{\Omega_0^S}$ et le quaternion caractérisant à un instant donné le passage entre β_0 et β_S.

Pour obtenir un système du premier ordre de la forme $\dfrac{dy_i}{dt} = \mathcal{F}_i(t, y_1(t), ..., y_n(t))$, il faut avoir une relation entre la dérivée du quaternion qui caractérise l'orientation du solide à un instant donné et $\overrightarrow{\Omega_0^S}$ qui intervient par ses dérivées dans le Principe Fondamental de la Dynamique (PFD).

Soit $\beta_0 = (\overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K})$ la base canonique de $\mathbb{R}^3$ associée au référentiel galiléen.
Soit $\beta_S = (\overrightarrow{i_S}, \overrightarrow{j_S}, \overrightarrow{k_S})$ une base orthonormale directe associée au référentiel $\mathcal{R}_S$ lié au solide S.
Soit $\mathbf{Q}_{\beta_\mathbf{0} \to \beta_\mathbf{S}}$ le quaternion représentant la rotation entre β_0 et β_S dans la base $(\mathbf{e}, \mathbf{I}, \mathbf{J}, \mathbf{K})$. Il représente une transformation unique. $(\overrightarrow{i_S}, \overrightarrow{j_S}, \overrightarrow{k_S})$ étant les images respectives de $(\overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K})$, on peut donc écrire ($\mathbf{Q}_{\beta_\mathbf{0} \to \beta_\mathbf{S}}$ sera noté par la suite $\mathbf{Q}$ pour alléger les notations) :

$$\mathbf{i_S} = \mathbf{Q}\mathbf{I}\overline{\mathbf{Q}} \qquad \mathbf{j_S} = \mathbf{Q}\mathbf{J}\overline{\mathbf{Q}} \qquad \mathbf{k_S} = \mathbf{Q}\mathbf{K}\overline{\mathbf{Q}} \tag{B.5}$$

Les coordonnées des quaternions purs $(\mathbf{i_S}, \mathbf{j_S}, \mathbf{k_S})$ dans la base de référence $(\mathbf{I}, \mathbf{J}, \mathbf{K})$ peuvent être interprétées comme les coordonnées dans la base $(\overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K})$ des vecteurs

$(\overrightarrow{i_S}, \overrightarrow{j_S}, \overrightarrow{k_S})$.

Remarque importante : $\mathbf{Q}$ a la même expression dans les deux bases $(\mathbf{e}, \mathbf{I}, \mathbf{J}, \mathbf{K})$ et $(\mathbf{e}, \mathbf{i_S}, \mathbf{j_S}, \mathbf{k_S})$. Cela se comprend aisément lorsqu'on revient à la définition de $\mathbf{Q}$. Il représente la transformation de β_0 à β_S définie par son axe orienté dirigé par le vecteur unitaire $\overrightarrow{u}$ et son angle θ. Dans une base quelconque de quaternions $(\mathbf{e}, \mathbf{i_q}, \mathbf{j_q}, \mathbf{k_q})$, il s'exprime par (voir B.2.4) :

$$\mathbf{Q} = \cos(\frac{\theta}{2})\mathbf{e} + \sin(\frac{\theta}{2})(\alpha_q\,\mathbf{i_q} + \beta_q\,\mathbf{j_q} + \gamma_q\,\mathbf{k_q}) \qquad \text{avec } \mathbf{u} = \alpha_q\,\mathbf{i_q} + \beta_q\,\mathbf{j_q} + \gamma_q\,\mathbf{k_q}$$

Or, la rotation en question transforme $(\overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K})$ en $(\overrightarrow{i_S}, \overrightarrow{j_S}, \overrightarrow{k_S})$ et conserve le vecteur $\overrightarrow{u}$. Il vient immédiatement :

$$\overrightarrow{u} = \alpha\,\overrightarrow{I} + \beta\,\overrightarrow{J} + \gamma\,\overrightarrow{K} \quad \Longrightarrow \quad \mathcal{R}(\overrightarrow{u}) = \overrightarrow{u} = \alpha\,\overrightarrow{i_S} + \beta\,\overrightarrow{j_S} + \gamma\,\overrightarrow{k_S}$$

Le quaternion $\mathbf{u}$ associé au vecteur $\overrightarrow{u}$ a donc les mêmes coordonnées dans les deux bases $(\mathbf{I}, \mathbf{J}, \mathbf{K})$ et $(\mathbf{i_S}, \mathbf{j_S}, \mathbf{k_S})$. Il en est donc de même pour le quaternion $\mathbf{Q}$.

Le vecteur instantané de rotation $\overrightarrow{\Omega_0^S}$ est une donnée intrinsèque qui ne dépend que du mouvement. Ce sont les expressions dans les différentes bases qui diffèrent.

$$\textit{Notation :} \qquad \overrightarrow{\Omega_0^S} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}_{\beta_0} \qquad \overrightarrow{\Omega_0^S} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}_{\beta_S}$$

On note $\mathbf{\Omega_0}$ le quaternion pur associé au vecteur $\overrightarrow{\Omega_0^S}$ exprimé dans la base $(\overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K})$ et $\mathbf{\Omega_S}$ le quaternion pur associé au vecteur $\overrightarrow{\Omega_0^S}$ exprimé dans la base $(\overrightarrow{i_S}, \overrightarrow{j_S}, \overrightarrow{k_S})$. On a alors :

$$\mathbf{\Omega_0} = \alpha\mathbf{I} + \beta\mathbf{J} + \gamma\mathbf{K} \qquad \mathbf{\Omega_S} = p\mathbf{i_S} + q\mathbf{j_S} + r\mathbf{k_S}$$

B.3.1 Relation entre Q et Ω_0

Dérivons la relation $\mathbf{i_S} = \mathbf{Q}\mathbf{I}\overline{\mathbf{Q}}$ entre les quaternions $\mathbf{i_S}$ et $\mathbf{I}$, la base canonique d'écriture des quaternions étant $(\mathbf{e}, \mathbf{I}, \mathbf{J}, \mathbf{K})$.

Remarque : Choisir une base de référence revient à se placer dans le référentiel de $\mathbb{R}^3$ associé à cette base.

$$\text{On obtient alors : } \frac{d\mathbf{i_S}}{dt} = \dot{\mathbf{Q}}\mathbf{I}\overline{\mathbf{Q}} + \mathbf{Q}\mathbf{I}\overline{\dot{\mathbf{Q}}}$$

$$\text{Comme } \mathbf{Q} \in \mathbb{S}, \mathbf{Q}^{-1} = \overline{\mathbf{Q}}, \text{on a alors avec (B.5)} \quad \overline{\mathbf{Q}}\mathbf{i_S} = \mathbf{I}\overline{\mathbf{Q}} \text{ et } \mathbf{i_S}\mathbf{Q} = \mathbf{Q}\mathbf{I}$$

$$\text{d'où} \quad \frac{d\mathbf{i_S}}{dt} = \dot{\mathbf{Q}}\overline{\mathbf{Q}}\mathbf{i_S} + \mathbf{i_S}\mathbf{Q}\overline{\dot{\mathbf{Q}}}$$

$$\mathbf{i_S} \in \mathbb{P} \text{ donc } \overline{\mathbf{i_S}} = -\mathbf{i_S} \qquad \Longrightarrow \qquad \mathbf{i_S}\mathbf{Q}\dot{\overline{\mathbf{Q}}} = -\overline{\dot{\mathbf{Q}}\overline{\mathbf{Q}}\mathbf{i_S}}$$

$$\text{En notant } \mathbf{m} = \dot{\mathbf{Q}}\overline{\mathbf{Q}}\mathbf{i_S}, \text{on peut écrire : } \frac{d\mathbf{i_S}}{dt} = \mathbf{m} - \overline{\mathbf{m}} = Pur(\mathbf{2m})$$

Or, la propriété vue en B.2.3 permet d'affirmer que $2\dot{\mathbf{Q}}\overline{\mathbf{Q}} \in \mathbb{P}$

$$\text{Finalement, en utilisant la notation de B.2.3 , } \quad \frac{d\mathbf{i_S}}{dt} = 2\dot{\mathbf{Q}}\overline{\mathbf{Q}} \wedge \mathbf{i_S}$$

Remarque : un calcul similaire permet d'obtenir la même expression avec $\mathbf{j_S}$ et $\mathbf{k_S}$.

B.3.1.1 Conclusion

Par analogie avec la définition du vecteur rotation instantanée dans l'espace, on en déduit que :

$$\boldsymbol{\Omega_0} = \underbrace{2\dot{\mathbf{Q}}\overline{\mathbf{Q}}}_{= \alpha\mathbf{I} + \beta\mathbf{J} + \gamma\mathbf{K}} \qquad \Longrightarrow \qquad \overrightarrow{\Omega_{R_0}^{R_S}} = \alpha\overrightarrow{I} + \beta\overrightarrow{J} + \gamma\overrightarrow{K}$$

$$\text{L'équation d'évolution s'écrit : } \quad \boxed{\dot{\mathbf{Q}} = \frac{1}{2}\boldsymbol{\Omega_0}\mathbf{Q}}$$

Rappel : la base de référence choisie étant $(\mathbf{e}, \mathbf{I}, \mathbf{J}, \mathbf{K})$, c'est bien le quaternion $\boldsymbol{\Omega_0}$, image du vecteur $\overrightarrow{\Omega_0^S}$ exprimé dans la base $(\overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K})$, qui est égal à l'expression trouvée.

B.3.2 Relation entre Q et Ω_S

En prenant comme base de référence des quaternions $\mathcal{H}_S = (\mathbf{e}, \mathbf{i_S}, \mathbf{j_S}, \mathbf{k_S})$, un calcul similaire permet d'exprimer une relation entre $\mathbf{I}$ et $\mathbf{i_S}$:

$$\text{On obtient } \frac{d\mathbf{I}}{dt} = 2\dot{\overline{\mathbf{Q}}}\mathbf{Q}\mathbf{I}$$

La base de référence étant $\mathcal{H}_S$, le quaternion $2\dot{\overline{\mathbf{Q}}}\mathbf{Q}$ est l'image du vecteur $\overrightarrow{\Omega_{R_S}^{R_0}}$ exprimé dans $\mathcal{R}_S$.

Il vient alors $-\boldsymbol{\Omega_S} = 2\dot{\overline{\mathbf{Q}}}\mathbf{Q}$

$$= -2\overline{\mathbf{Q}}\dot{\mathbf{Q}} \qquad \text{car } 2\overline{\mathbf{Q}}\dot{\mathbf{Q}} \in \mathbb{P} \Rightarrow \overline{2\overline{\mathbf{Q}}\dot{\mathbf{Q}}} = -2\dot{\overline{\mathbf{Q}}}\mathbf{Q}$$

$$\text{Conséquence : } \qquad \boldsymbol{\Omega_S} = 2\overline{\mathbf{Q}}\dot{\mathbf{Q}}$$

La multiplication par $\mathbf{Q}$ permet d'obtenir la relation d'évolution du quaternion $\mathbf{Q}$:

$$\boxed{\dot{\mathbf{Q}} = \frac{1}{2}\mathbf{Q}\boldsymbol{\Omega_S}}$$

Remarque : pour parvenir à ce résultat, on peut aussi raisonner de la façon suivante.

Considérons le quaternion $\mathbf{\Omega'_0}$ écrit dans la base mobile ayant les mêmes coordonnées que $\mathbf{\Omega_0}$. On a alors $\mathbf{\Omega'_0} = \alpha\mathbf{i_S} + \beta\mathbf{j_S} + \gamma\mathbf{k_S}$.

$\mathbf{\Omega'_0}$ n'est alors plus associé au vecteur intrinsèque $\overrightarrow{\Omega_0^S}$. Pour obtenir de nouveau le vecteur associé à $\overrightarrow{\Omega_0^S}$, il faut effectuer le changement de base qui transforme $(\mathbf{i_S}, \mathbf{j_S}, \mathbf{k_S})$ en $(\mathbf{I}, \mathbf{J}, \mathbf{K})$, qui n'est autre que la rotation inverse effectuée par $\mathbf{Q}$, i.e. la rotation de quaternion $\overline{\mathbf{Q}}$. Les expressions de $\mathbf{Q}$ étant identiques sur les bases $(\mathbf{e}, \mathbf{I}, \mathbf{J}, \mathbf{K})$ et $(\mathbf{e}, \mathbf{i_S}, \mathbf{j_S}, \mathbf{k_S})$, on en déduit alors le même résultat :

$$\mathbf{\Omega_S} = \mathbf{R}_{\overline{\mathbf{Q}}}(\mathbf{\Omega'_0}) = \overline{\mathbf{Q}}\left[\dot{\mathbf{Q}}\overline{\mathbf{Q}}\right]\mathbf{Q} = 2\overline{\mathbf{Q}}\dot{\mathbf{Q}}$$

B.4 Rotation, matrices de passage et changement de base

B.4.1 Matrices de passage

B.4.1.1 Définition

Soient $\mathcal{B}_0$ et $\mathcal{B}_1$, deux bases orthonormales directes. On note $R_{0\to1}$, la rotation qui transforme la base $\mathcal{B}_0$ en $\mathcal{B}_1$. Par définition, on appelle matrice de passage de $\mathcal{B}_0$ à $\mathcal{B}_1$, et on note $P_{\mathcal{B}_0\to\mathcal{B}_1}$ la matrice dont les colonnes représentent les coordonnées dans la base $\mathcal{B}_0$ des vecteurs de base de $\mathcal{B}_1$.

$$P_{\mathcal{B}_0\to\mathcal{B}_1} = \underset{\mathcal{B}_0}{\text{Mat}}\ \mathcal{B}_1 = \underset{\mathcal{B}_1,\mathcal{B}_0}{\text{Mat}}\ Id$$

B.4.1.2 Propriétés

Les matrices de passage appartiennent au groupe orthogonal $\mathrm{GL}(\mathbb{R}^3)$.

$$P_{\mathcal{B}_1\to\mathcal{B}_0} = P^t_{\mathcal{B}_0\to\mathcal{B}_1} = P^{-1}_{\mathcal{B}_0\to\mathcal{B}_1}$$

B.4.2 Changement de base

Soit $\overrightarrow{X}$ un vecteur de $\mathbb{R}^3$. Notons $\overrightarrow{X}|_{\mathcal{B}_0}$ et $\overrightarrow{X}|_{\mathcal{B}_1}$ les coordonnées de ce vecteur respectivement dans $\mathcal{B}_0$ et $\mathcal{B}_1$. $\overrightarrow{X}|_{\mathcal{B}_0}$ et $\overrightarrow{X}|_{\mathcal{B}_1}$ représentent le même vecteur, mais exprimé dans des bases différentes. On a alors :

$$\overrightarrow{X}|_{\mathcal{B}_1} = \underset{\mathcal{B}_0,\mathcal{B}_1}{\text{Mat}}\ Id\ \overrightarrow{X}|_{\mathcal{B}_0} \qquad \overrightarrow{X}|_{\mathcal{B}_0} = \underset{\mathcal{B}_1,\mathcal{B}_0}{\text{Mat}}\ Id\ \overrightarrow{X}|_{\mathcal{B}_1}$$

On en déduit alors les relations avec les matrices de passage :

$$\overrightarrow{X}|_{\mathcal{B}_1} = P_{\mathcal{B}_1\to\mathcal{B}_0}\overrightarrow{X}|_{\mathcal{B}_0} \qquad \overrightarrow{X}|_{\mathcal{B}_0} = P_{\mathcal{B}_0\to\mathcal{B}_1}\overrightarrow{X}|_{\mathcal{B}_1}$$

B.4.3 Expression des vecteurs images de $R_{0\to1}$

Soit $\overrightarrow{X}$ un vecteur de $\mathbb{R}^3$ et $\overrightarrow{Y}$ l'image de $\overrightarrow{X}$ par la rotation $R_{0\to1}$. On note $\overrightarrow{X}|_{\mathcal{B}_i}$ et $\overrightarrow{Y}|_{\mathcal{B}_i}$ les coordonnées de $\overrightarrow{X}$ et $\overrightarrow{Y}$ dans la base $\mathcal{B}_i$. Par définition, on a :

$$\overrightarrow{Y}|_{\mathcal{B}_1} = \overrightarrow{X}|_{\mathcal{B}_0}$$

On en déduit alors immédiatement :

$$\overrightarrow{Y}|_{\mathcal{B}_0} = P_{\mathcal{B}_0\to\mathcal{B}_1}\overrightarrow{X}|_{\mathcal{B}_0} \qquad \overrightarrow{Y}|_{\mathcal{B}_1} = P_{\mathcal{B}_0\to\mathcal{B}_1}\overrightarrow{X}|_{\mathcal{B}_1}$$

Remarque : à la différence de B.4.2, ces relations ne traduisent pas les expressions d'un même vecteur dans des bases différentes mais bien l'image $\overrightarrow{Y}$ d'un vecteur $\overrightarrow{X}$ par une rotation, $\overrightarrow{X}$ et $\overrightarrow{Y}$ étant exprimés dans la même base. La matrice de passage est vue ici comme la représentante de l'application linéaire $R_{0\to1}$:

$$P_{\mathcal{B}_0\to\mathcal{B}_1} = \underset{\mathcal{B}_0,\,\mathcal{B}_0}{\mathrm{Mat}}\, R_{0\to1} = \underset{\mathcal{B}_1,\,\mathcal{B}_1}{\mathrm{Mat}}\, R_{0\to1}$$

B.5 Composition de rotations successives

B.5.1 Position du problème

Soient R_1 et R_2 deux rotations de $\mathbb{R}^3$ respectivement d'axe orienté par $\overrightarrow{u_1} = (\alpha_1, \beta_1, \gamma_1)$ et $\overrightarrow{u_2} = (\alpha_2, \beta_2, \gamma_2)$ et d'angle θ_1 et θ_2. Soit $\mathcal{B}_0$ la base canonique de $\mathbb{R}^3$. On définit alors par $\mathcal{B}_1$ et $\mathcal{B}_2$ les bases successives obtenues par application de R_1 puis R_2. On note aussi $\overrightarrow{Y}$ l'image d'un vecteur $\overrightarrow{X}$ par R_1 et $\overrightarrow{Z}$ l'image de $\overrightarrow{Y}$ par R_2.

$$\mathcal{B}_0 \overset{R_1}{\longmapsto} \mathcal{B}_1 \overset{R_2}{\longmapsto} \mathcal{B}_2 \qquad \overrightarrow{X} \overset{R_1}{\longmapsto} \overrightarrow{Y} \overset{R_2}{\longmapsto} \overrightarrow{Z}$$

La composition des rotations $R_2 \circ R_1$ s'exprime alors par la relation suivante :

$$\overrightarrow{Z} = R_2 \circ R_1(\overrightarrow{X}) = R_2(\underbrace{R_1(\overrightarrow{X})}_{\overrightarrow{Y}})$$

Par définition des rotations, on a :

$$\overrightarrow{X}|_{\mathcal{B}_0} \overset{R_1}{\longmapsto} \overrightarrow{Y}|_{\mathcal{B}_1} = \overrightarrow{X}|_{\mathcal{B}_0} \overset{R_2}{\longmapsto} \overrightarrow{Z}|_{\mathcal{B}_2} = \overrightarrow{Y}|_{\mathcal{B}_1} \tag{B.6}$$

Examinons alors les caractéristiques de la transformation composée $R_2 \circ R_1$ en terme de matrices de passage et de quaternions.

B.5.2 Expression en terme de matrices de passage

En se basant sur (B.6) et en utilisant les formules de changement de bases (voir B.4.2), on en déduit facilement la relation suivante :

$$\overrightarrow{Z}|_{\mathcal{B}_0} = P_{\mathcal{B}_0\to\mathcal{B}_1} \underbrace{P_{\mathcal{B}_1\to\mathcal{B}_2}\overrightarrow{Z}|_{\mathcal{B}_2}}_{\overrightarrow{Z}|_{\mathcal{B}_1}} = P_{\mathcal{B}_0\to\mathcal{B}_1}P_{\mathcal{B}_1\to\mathcal{B}_2}\overrightarrow{X}|_{\mathcal{B}_0}$$

Il vient alors :

$$\underset{\mathcal{B}_0,\mathcal{B}_0}{\text{Mat}}\, R_2\circ R_1 \;=\; \underset{\mathcal{B}_2,\mathcal{B}_2}{\text{Mat}}\, R_2\circ R_1 \;= P_{\mathcal{B}_0\to\mathcal{B}_2} = P_{\mathcal{B}_0\to\mathcal{B}_1}P_{\mathcal{B}_1\to\mathcal{B}_2} \tag{B.7}$$

B.5.2.1 Rotations successives s'appuyant sur la base courante

Les décompositions en rotations successives de type angles Euler (ou équivalent) s'appuient en général sur des rotations définies à partir de la base courante. C'est le cas où les composantes des vecteurs $\overrightarrow{u_1}$ et $\overrightarrow{u_2}$ sont respectivement données dans $\mathcal{B}_0$ et $\mathcal{B}_1$. Les expressions des deux matrices de passage $P_{\mathcal{B}_0\to\mathcal{B}_1}$ et $P_{\mathcal{B}_1\to\mathcal{B}_2}$ sont alors indépendantes et sont facilement calculables.

B.5.2.2 Rotations successives s'appuyant sur la base de référence

Dans le cas contraire où le vecteur $\overrightarrow{u_2}$ est lié à la base primaire $\mathcal{B}_0$, la rotation peut toujours s'exprimer par la matrice de passage $P_{\mathcal{B}_1\to\mathcal{B}_2}$. Cependant l'expression de celle-ci dépend des caractéristiques de la première rotation (puisque l'expression du vecteur $\overrightarrow{u_2}$ dans la base $\mathcal{B}_1$ est fonction de R_1). Il peut donc être plus judicieux de caractériser la rotation R_2 par une matrice de passage $P_{\mathcal{B}_0\to\mathcal{B}_3}$ indépendante de R_1 définie par :

$$P_{\mathcal{B}_0\to\mathcal{B}_3} = \underset{\mathcal{B}_0,\mathcal{B}_0}{\text{Mat}}\, R_2$$

On peut alors écrire :

$$\overrightarrow{X}|_{\mathcal{B}_0} \overset{R_1}{\longmapsto} \overrightarrow{Y}|_{\mathcal{B}_0} = \underbrace{\underset{\mathcal{B}_0,\mathcal{B}_0}{\text{Mat}}\, R_1}_{P_{\mathcal{B}_0\to\mathcal{B}_1}} \overrightarrow{X}|_{\mathcal{B}_0} \overset{R_2}{\longmapsto} \overrightarrow{Z}|_{\mathcal{B}_0} = \underbrace{\underset{\mathcal{B}_0,\mathcal{B}_0}{\text{Mat}}\, R_2}_{P_{\mathcal{B}_0\to\mathcal{B}_3}} \overrightarrow{Y}|_{\mathcal{B}_0} \tag{B.8}$$

$$\text{d'où } \overrightarrow{Z}|_{\mathcal{B}_0} = P_{\mathcal{B}_0\to\mathcal{B}_3}P_{\mathcal{B}_0\to\mathcal{B}_1}\overrightarrow{X}|_{\mathcal{B}_0}$$

Dans ce cas, nous avons (on remarquera ici l'ordre inversé par rapport à (B.7)) :

$$\underset{\mathcal{B}_0,\mathcal{B}_0}{\text{Mat}}\, R_2\circ R_1 \;= P_{\mathcal{B}_0\to\mathcal{B}_3}P_{\mathcal{B}_0\to\mathcal{B}_1}$$

B.5.2.3 Relation entre $P_{\mathcal{B}_0\to\mathcal{B}_3}$ et $P_{\mathcal{B}_1\to\mathcal{B}_2}$

$$\begin{aligned} P_{\mathcal{B}_1\to\mathcal{B}_2} = \underset{\mathcal{B}_1,\mathcal{B}_1}{\text{Mat}}\, R_2 &= \underset{\mathcal{B}_0,\mathcal{B}_1}{\text{Mat}}\, Id \quad \underset{\mathcal{B}_0,\mathcal{B}_0}{\text{Mat}}\, R_2 \quad \underset{\mathcal{B}_1,\mathcal{B}_0}{\text{Mat}}\, Id \\ &= P_{\mathcal{B}_1\to\mathcal{B}_0} \quad P_{\mathcal{B}_0\to\mathcal{B}_3} \quad P_{\mathcal{B}_0\to\mathcal{B}_1} \end{aligned}$$

On retrouve ici la dépendance de $P_{\mathcal{B}_1\to\mathcal{B}_2}$ à la rotation R_1.

B.5.3 Expression en terme de quaternions

Le problème est complètement similaire pour les quaternions. On note $\mathcal{H}_0$ la base de quaternions associée à la base canonique de $\mathbb{R}^3$.

Remarque : tous les quaternions associés aux expressions des vecteurs dans une base donnée seront ainsi exprimés dans $\mathcal{H}_0$.

Soient $\mathbf{R_1} = (q_1^0, r_1^0, s_1^0, t_1^0)$ et $\mathbf{R_2} = (q_2^0, r_2^0, s_2^0, t_2^0)$ les quaternions exprimés dans la base $\mathcal{H}_0$ représentant les rotations R_1 et R_2 définies précédemment.
Si les deux rotations sont effectuées relativement à la base de référence $\mathcal{B}_0$, c'est-à-dire si les composantes de $\overrightarrow{u_1}$ et de $\overrightarrow{u_2}$ sont exprimées dans $\mathcal{B}_0$, alors on a :

$$(q_1^0, r_1^0, s_1^0, t_1^0) = \left(cos(\frac{\theta_1}{2}), \alpha_1 sin(\frac{\theta_1}{2}), \beta_1 sin(\frac{\theta_1}{2}), \gamma_1 sin(\frac{\theta_1}{2})\right)$$

$$(q_2^0, r_2^0, s_2^0, t_2^0) = \left(cos(\frac{\theta_2}{2}), \alpha_2 sin(\frac{\theta_2}{2}), \beta_2 sin(\frac{\theta_2}{2}), \gamma_2 sin(\frac{\theta_2}{2})\right)$$

Remarque : Dans ce cas, le quaternion $\mathbf{R_2}$ est l'équivalent de la matrice de passage $P_{\mathcal{B}_0\to\mathcal{B}_3}$ définie au paragraphe précédent et non à la matrice de passage $P_{\mathcal{B}_1\to\mathcal{B}_2}$.

La composition $R_2 \circ R_1$ est alors donnée par le produit de quaternion $\mathbf{R_2 R_1}$. En effet, soit $\overrightarrow{S}$ un vecteur , et $\overrightarrow{T}$ son image par la rotation $R_2 \circ R_1$. Si l'on note $\mathbf{S_0}$ et $\mathbf{T_0}$, les quaternions purs de $\mathcal{H}_0$ associés à $\overrightarrow{S}$ et $\overrightarrow{T}$ dans la base $\mathcal{B}_0$, on a, par définition, les relations suivantes :

$$\overrightarrow{T} = R_2 \circ R_1(\overrightarrow{S}) = R_2\left(R_1(\overrightarrow{S})\right)$$

$$\mathbf{T_0} = \mathbf{R_2}\left(\mathbf{R_1}\,\mathbf{S_0}\,\overline{\mathbf{R_1}}\right)\overline{\mathbf{R_2}} = \left(\mathbf{R_2 R_1}\right)\,\mathbf{S_0}\,\overline{\left(\mathbf{R_2 R_1}\right)}$$

Dans le cas de rotations successives où chacune d'entre elles s'appuie sur la base courante, les composantes des vecteurs directeurs $\overrightarrow{u_i}$ sont exprimées (généralement) dans la base courante. Leur expression est ainsi indépendante des autres rotations et souvent très simple (la plupart du temps, c'est une rotation autour d'un axe principal qui est effectuée). Ainsi, si l'on veut appliquer la relation précédente, il faut exprimer les différents vecteurs $\overrightarrow{u_i}$ dans la base de référence $\mathcal{B}_0$ sachant que leurs composantes dépendent des rotations précédentes.

En effet, si les composantes de $\overrightarrow{u_2}$ sont définies par rapport à la base $\mathcal{B}_1$ (c'est-à-dire si $\overrightarrow{u_2}|_{\mathcal{B}_1} = (\alpha_2, \beta_2, \gamma_2)$, le quaternion $\mathbf{R_2}$ représentant la rotation $\mathcal{B}_1 \longrightarrow \mathcal{B}_2$ n'a pas comme expression $\left(cos(\frac{\theta_2}{2}), \alpha_2 sin(\frac{\theta_2}{2}), \beta_2 sin(\frac{\theta_2}{2}), \gamma_2 sin(\frac{\theta_2}{2})\right)$ dans la base de quaternion de référence $\mathcal{H}_0$. Cette expression est valable dans les bases de quaternions associées à $\mathcal{B}_1$ et $\mathcal{B}_2$. On note alors $\mathbf{Q_{1\to 2}}$ ce quaternion.

$$\mathbf{Q_{1\to 2}}|_{\mathcal{H}_0} = \left(cos(\frac{\theta_2}{2}), \alpha_2 sin(\frac{\theta_2}{2}), \beta_2 sin(\frac{\theta_2}{2}), \gamma_2 sin(\frac{\theta_2}{2})\right)$$

De même, on note :

$$\mathbf{Q_{0\to 1}}|_{\mathcal{H}_0} = \left(cos(\frac{\theta_1}{2}), \alpha_1 sin(\frac{\theta_1}{2}), \beta_1 sin(\frac{\theta_1}{2}), \gamma_1 sin(\frac{\theta_1}{2})\right) = \mathbf{R_1}$$

Exprimons alors la rotation $R_2 \circ R_1$ en fonction de $\mathbf{Q_{0\to 1}}$ et $\mathbf{Q_{1\longrightarrow 2}}$. Par définition :

$$\overrightarrow{T}|_{\mathcal{B}_2} = \overrightarrow{S}|_{\mathcal{B}_0}$$

L'objectif est de trouver le quaternion $\mathbf{Q_c}$ tel que :

$$\mathbf{T_0} = \mathbf{Q_c}\,\mathbf{S_0}\,\overline{\mathbf{Q_c}}$$

Pour cela, exprimons en terme de quaternions le vecteur $\overrightarrow{T}|_{\mathcal{B}_0}$ en fonction de $\overrightarrow{T}|_{\mathcal{B}_2}$.

On note $\mathbf{T_1}$ et $\mathbf{T_2}$ les quaternions de $\mathcal{H}_0$ associés aux vecteurs $\overrightarrow{T}|_{\mathcal{B}_1}$ et $\overrightarrow{T}|_{\mathcal{B}_2}$. L'analogie avec les matrices de passage permet d'écrire :

$$\overrightarrow{T}|_{\mathcal{B}_1} = P_{\mathcal{B}_1\to\mathcal{B}_2}\overrightarrow{T}|_{\mathcal{B}_2} \qquad \mathbf{T_1} = \mathbf{Q_{1\to 2}}\mathbf{T_2}\overline{\mathbf{Q_{1\to 2}}}$$

$$\overrightarrow{T}|_{\mathcal{B}_0} = P_{\mathcal{B}_0\to\mathcal{B}_1}\overrightarrow{T}|_{\mathcal{B}_1} \qquad \mathbf{T_0} = \mathbf{Q_{0\to 1}}\mathbf{T_1}\overline{\mathbf{Q_{0\to 1}}} = \mathbf{R_1}\mathbf{T_1}\overline{\mathbf{R_1}}$$

$$\text{d'où } \mathbf{T_0} = \mathbf{Q_{0\to 1}}\mathbf{Q_{1\to 2}}\mathbf{T_2}\overline{\mathbf{Q_{0\to 1}}\mathbf{Q_{1\to 2}}}$$

Étant donné que $\mathbf{T_2} = \mathbf{S_0}$, on en déduit :

$$\mathbf{Q_c} = \mathbf{Q_{0\to 1}}\mathbf{Q_{1\to 2}} = \mathbf{R_1}\mathbf{Q_{1\to 2}}$$

B.5.3.1 Généralisation

On considère les n rotations R_i de vecteurs $\overrightarrow{u_i} = (\alpha_i, \beta_i, \gamma_i)$ et d'angle θ_i. On associe à chaque rotation R_i le quaternion $\mathbf{Q_{i-1\to i}}$ défini dans la base canonique $\mathcal{H}_0$ par

$$\mathbf{Q_{i-1\to i}} = \left(cos(\frac{\theta_i}{2}), \alpha_i sin(\frac{\theta_i}{2}), \beta_i sin(\frac{\theta_i}{2}), \gamma_i sin(\frac{\theta_i}{2})\right)$$

Le produit $\mathbf{Q_i} = \mathbf{Q_{n-1\to n}}...\mathbf{Q_{1\to 2}}\mathbf{Q_{0\to 1}}$ représente la composition des n rotations $R_n \circ ...R_2 \circ R_1$ toutes effectuées à partir de la base initiale, c'est-à-dire avec tous les $(\alpha_i, \beta_i, \gamma_i)$ exprimés dans $\mathcal{B}_0$.

Le produit $\mathbf{Q_c} = \mathbf{Q_{0\to 1}}\mathbf{Q_{1\to 2}}...\mathbf{Q_{n-1\to n}}$ représente quant à lui la composition des n rotations $R_n \circ ...R_2 \circ R_1$ où chaque rotation successive est définie dans la base courante $\mathcal{B}_{i-1}$. Les composantes $(\alpha_i, \beta_i, \gamma_i)$ sont alors données dans la base $\mathcal{B}_{i-1}$.

B.5.3.2 Exemple

Soient $\mathbf{Q_1} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0)$, $\mathbf{Q_2} = (\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0)$ et $\mathbf{Q_3} = (\frac{\sqrt{2}}{2}, 0, 0, \frac{\sqrt{2}}{2})$.

Ces 3 quaternions expriment dans leur base d'écriture les 3 rotations d'angles +90° et d'axe respectif $\overrightarrow{x}$, $\overrightarrow{y}$ et $\overrightarrow{z}$. Exprimés dans la base $\mathcal{B}_0$, ils expriment les 1/4 de tour d'axe respectif $\overrightarrow{x_0}$, $\overrightarrow{y_0}$ et $\overrightarrow{z_0}$.

Le quaternion $\mathbf{Q_i} = \mathbf{Q_3Q_2Q_1}$ exprime alors la rotation issue de la composition des 1/4 de tour d'axe successif $\overrightarrow{x_0}$, $\overrightarrow{y_0}$ et $\overrightarrow{z_0}$ (voir FIG. B.2).

Le quaternion $\mathbf{Q_c} = \mathbf{Q_1Q_2Q_3}$ a pour effet d'effectuer les rotations non plus par rapport à la base initiale mais par rapport à la base courante. On effectue alors les 1/4 de tour successivement par rapport à $\overrightarrow{x_0}$, $\overrightarrow{y_1}$ et $\overrightarrow{z_2}$, comme représenté FIG. B.3.

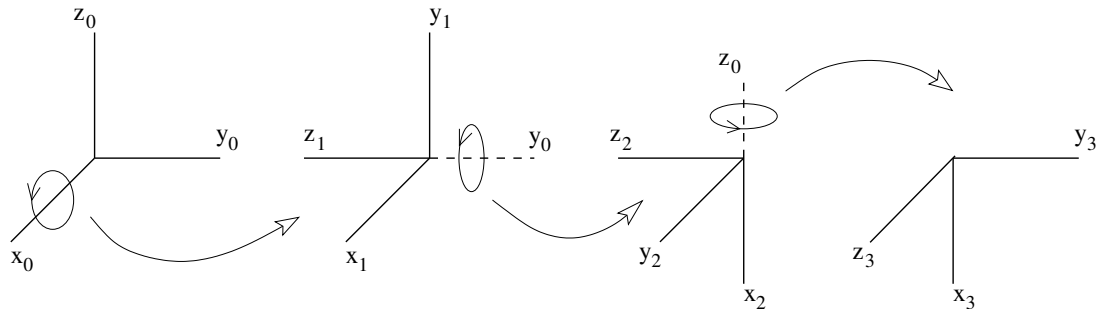


FIG. B.2: 1/4 de tour d'axe successif x, y et z par rapport à la base $\mathcal{B}_0$

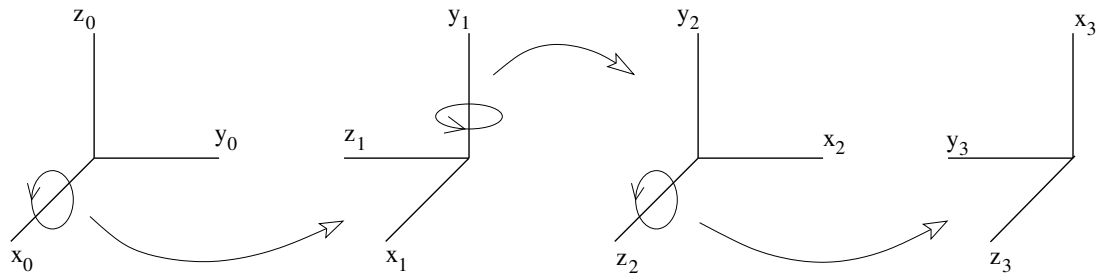


FIG. B.3: 1/4 de tour d'axe successif x, y et z par rapport aux bases courantes

Annexe C

Résolution de l'équation d'orientation par les quaternions

C.1 Avant-propos

Ce document a été initialement écrit par A. Leroyer pour résoudre le problème d'interpolation de quaternions rencontré dans le calcul de la courbure de l'anguille *ROBEA/RAAMO*. Il traitait aussi des angles de Cardan. J'ai simplifié ce document, afin qu'il ne touche que l'interpolation de quaternions. Les notations s'appuient sur celles de la thèse [57].

C.2 Problème

Lorsque l'on veut résoudre l'équation d'orientation en utilisant un quaternion, (ou lorsqu'on veut avoir accès à l'orientation d'une poutre en fonction de son abscisse curviligne), on est amené à résoudre une équation différentielle du type :

$$\frac{dq}{dx} = \mathbb{A}\, q$$

où q est un quaternion d'orientation, donc de norme unitaire.

Dans le premier cas, $x \equiv t$ et $x \equiv s$ dans le cas de l'orientation de la ligne neutre d'une poutre.

Si on utilise une discrétisation classique du terme $\frac{dq}{dx}$, il est nécessaire d'avoir recours à une renormalisation, car la résolution numérique ne permet alors pas de maintenir la norme du quaternion strictement égale à 1. Pour éviter cette phase de renormalisation, il est préférable de se baser directement sur la résolution analytique de l'équation différentielle. Cela permet alors d'avoir accès à une expression de quaternion directement normée.

C.3 Cas de l'évolution temporelle du quaternion d'orientation

La résolution se fait dans la base fixe $\mathcal{B}_0$. On note (q_0, q_1, q_2, q_3) les coordonnées du quaternion d'orientation $\mathbf{Q}$ dans la base $\mathcal{H}_0$ associée à $\mathcal{B}_0$ et (α, β, γ) les coordonnées du vecteur instantané de rotation $\overrightarrow{\Omega_0^1}$ dans la base $\mathcal{B}_0$.

L'équation d'évolution du quaternion est alors (cf. [57]) :

$$\frac{d}{dt}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\alpha & -\beta & -\gamma \\ \alpha & 0 & -\gamma & \beta \\ \beta & \gamma & 0 & -\alpha \\ \gamma & -\beta & \alpha & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{C.1}$$

On pose alors :

$$\mathbb{A} = \frac{1}{2}\begin{bmatrix} 0 & -\alpha & -\beta & -\gamma \\ \alpha & 0 & -\gamma & \beta \\ \beta & \gamma & 0 & -\alpha \\ \gamma & -\beta & \alpha & 0 \end{bmatrix} \tag{C.2}$$

En effectuant un développement limité de $q(t+dt) - q(t)$, et en utilisant (C.1), on peut montrer que pour $\mathbb{A}$ à coefficients constants entre t et $t+dt$, on a :

$$q(t+dt) = \exp\left(\mathbb{A}dt\right) q(t) \tag{C.3}$$

C.3.1 Calcul de l'exponentielle de $\mathbb{A}$

La matrice $\mathbb{A}$ est antisymétrique et possède une propriété particulière. En effet, on a :

$$\begin{aligned} \mathbb{A}^2 &= -\tfrac{1}{4}\left(\alpha^2+\beta^2+\gamma^2\right)\mathbb{I} \\ \mathbb{A}^3 &= -\tfrac{1}{4}\left(\alpha^2+\beta^2+\gamma^2\right)\mathbb{A} \end{aligned}$$

Plus généralement,

$$\begin{aligned} \mathbb{A}^{2n} &= \left(-\tfrac{1}{4}\right)^n\left(\alpha^2+\beta^2+\gamma^2\right)^n\mathbb{I} \\ \mathbb{A}^{2n+1} &= \left(-\tfrac{1}{4}\right)^n\left(\alpha^2+\beta^2+\gamma^2\right)^n\mathbb{A} \end{aligned}$$

Si on pose $\theta = \dfrac{dt}{2}\sqrt{\alpha^2+\beta^2+\gamma^2}$ et $\omega = \sqrt{\alpha^2+\beta^2+\gamma^2}$, on en déduit alors par définition de l'exponentielle $\left(\exp(\Delta) = \displaystyle\sum_0^{+\infty}\frac{\Delta^n}{n!}\right)$:

$$\exp(\mathbb{A}dt) = \sum_{0}^{+\infty}(-1)^n \frac{\theta^{2n}}{(2n)!}\mathbb{I} + \sum_{0}^{+\infty}(-1)^n \frac{\theta^{2n+1}}{(2n+1)!}\frac{\mathbb{A}dt}{\theta}$$

$$\text{i.e. } \exp(\mathbb{A}dt) = \cos(\theta)\,\mathbb{I} + \sin(\theta)\,\frac{2\,\mathbb{A}}{\omega}$$

On a alors :

$$\begin{aligned}
q_0(t+dt) &= \cos(\theta)q_0(t) + \sin(\theta)\left[-\tfrac{\alpha}{\omega}q_1(t) - \tfrac{\beta}{\omega}q_2(t) - \tfrac{\gamma}{\omega}q_3(t)\right]\\
q_1(t+dt) &= \cos(\theta)q_1(t) + \sin(\theta)\left[+\tfrac{\alpha}{\omega}q_0(t) - \tfrac{\gamma}{\omega}q_2(t) + \tfrac{\beta}{\omega}q_3(t)\right]\\
q_2(t+dt) &= \cos(\theta)q_2(t) + \sin(\theta)\left[+\tfrac{\beta}{\omega}q_0(t) + \tfrac{\gamma}{\omega}q_1(t) - \tfrac{\alpha}{\omega}q_3(t)\right]\\
q_3(t+dt) &= \cos(\theta)q_3(t) + \sin(\theta)\left[+\tfrac{\gamma}{\omega}q_0(t) - \tfrac{\beta}{\omega}q_1(t) + \tfrac{\alpha}{\omega}q_2(t)\right]
\end{aligned}$$

Comme $q(t)$ est supposé de norme unitaire et que $\frac{\alpha}{\omega}^2 + \frac{\beta}{\omega}^2 + \frac{\gamma}{\omega}^2 = 1$, on peut montrer que $q(t+dt)$ est aussi de norme unitaire !

Ainsi, aucune normalisation n'est nécessaire au cours des itérations temporelles. Concernant la résolution numérique, on prendra les coefficients de la matrice $\mathbb{A}$ en $t + dt/2$, ce qui permet d'avoir un schéma du deuxième ordre.

C.4 Cas de l'évolution spatiale du quaternion d'orientation

Dans ce cas, le vecteur courbure (équivalent du vecteur instantané de rotation $\overrightarrow{\Omega_0^1}$) est spécifié non plus dans la base fixe $\mathcal{B}_0$ mais dans la base locale d'abscisse curviligne courante. On note encore (k_1, k_2, k_3) les composantes du vecteur courbure dans cette base.

L'équation d'évolution du quaternion est alors (cf. [57]) :

$$\frac{d}{ds}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -k_1 & -k_2 & -k_3 \\ k_1 & 0 & k_3 & -k_2 \\ k_2 & -k_3 & 0 & k_1 \\ k_3 & k_2 & -k_1 & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{C.4}$$

De même que précédemment, on a :

$$q(s+ds) = \exp\left(\mathbb{K}ds\right) q(s)$$

En posant :

$$\mathbb{K} = \frac{1}{2}\begin{bmatrix} 0 & -k_1 & -k_2 & -k_3 \\ k_1 & 0 & k_3 & -k_2 \\ k_2 & -k_3 & 0 & k_1 \\ k_3 & k_2 & -k_1 & 0 \end{bmatrix} \tag{C.5}$$

C.4.1 Calcul de l'exponentielle de $\mathbb{K}$

On a les mêmes propriétés que pour la matrice $\mathbb{A}$:

$$\begin{aligned} \mathbb{K}^{2n} &= \left(-\tfrac{1}{4}\right)^n \left(k_1^2 + k_2^2 + k_3^2\right)^n \mathbb{I} \\ \mathbb{K}^{2n+1} &= \left(-\tfrac{1}{4}\right)^n \left(k_1^2 + k_2^2 + k_3^2\right)^n \mathbb{K} \end{aligned}$$

Ainsi, en posant $\theta_k = \dfrac{ds}{2}\sqrt{k_1^2 + k_2^2 + k_3^2}$ et $\kappa = \sqrt{k_1^2 + k_2^2 + k_3^2}$, on a :

$$\exp(\mathbb{K}ds) = \cos(\theta_k)\,\mathbb{I} + \sin(\theta_k)\,\frac{2\,\mathbb{K}}{\kappa}$$

Ce qui donne :

$$\begin{aligned} q_0(s+ds) &= \cos(\theta_k)q_0(s) + \sin(\theta_k)\left[-\tfrac{k_1}{\kappa}q_1(s) - \tfrac{k_2}{\kappa}q_2(s) - \tfrac{k_3}{\kappa}q_3(s)\right] \\ q_1(s+ds) &= \cos(\theta_k)q_1(s) + \sin(\theta_k)\left[+\tfrac{k_1}{\kappa}q_0(s) + \tfrac{k_3}{\kappa}q_2(s) - \tfrac{k_2}{\kappa}q_3(s)\right] \\ q_2(s+ds) &= \cos(\theta_k)q_2(s) + \sin(\theta_k)\left[+\tfrac{k_2}{\kappa}q_0(s) - \tfrac{k_3}{\kappa}q_1(s) + \tfrac{k_1}{\kappa}q_3(s)\right] \\ q_3(s+ds) &= \cos(\theta_k)q_3(s) + \sin(\theta_k)\left[+\tfrac{k_3}{\kappa}q_0(s) + \tfrac{k_2}{\kappa}q_1(s) - \tfrac{k_1}{\kappa}q_2(s)\right] \end{aligned}$$

Là encore, on peut montrer que $q(s+ds)$ est de norme unitaire.

Annexe D

Communication au congrès FIV2008

D.1 Description

Le congrès Flow-Induced Vibrations (FIV) de l'année 2008 s'est déroulé à Prague en République Tchèque du 30 Juin au 3 Juillet. Le FIV2008[1] est un des principaux rendez-vous de l'IFS. Il réunit de nombreux corps de métier (hydrodynamique, nucléaire...). Cette année, plus de 200 chercheurs ont participé à l'événement.

D.2 Long abstract

[1] http://www.it.cas.cz/fiv2008

Flow Induced Vibration, Zolotarev & Horacek eds. Institute of Thermomechanics, Prague, 2008

LARGE DEFORMATIONS ON ELONGATED BODIES

De Nayer Guillaume, Visonneau Michel, Leroyer Alban
Ecole Centrale Nantes, Nantes, France

Boyer Frederic
Ecole des Mines de Nantes, Nantes, France

ABSTRACT

In this paper the coupling of a structural solver for elongated structures in large deformations and in large displacements with the flow solver ISIS-CFD is presented. ISIS-CFD is a 3D finite volume solver based on the incompressible unsteady Reynolds-averaged Navier-Stokes equations. The finite element structural solver is also 3D and uses Euler-Bernoulli or Rayleigh kinematics with the Cosserat hypothesis. A remeshing procedure based on the pseudo-solid approximation is detailed. This coupled algothim is first validated and then applied to a geometrically simple 2D test case.

1. INTRODUCTION

Elongated structures, like pillars supporting oil platforms or cables and risers, are frequently met in the industrial domain. The stakes of the fluid/structure interaction (FSI) around these bodies are therefore important. This is the reason why the CFD team of the Fluid Mechanics Laboratory from Centrale Nantes has started the development of FSI for elongated structures with the help of its in-house RANSE solver ISIS-CFD.

To carry out FSI modelling, four points are essential to address :

- The transfer of the efforts exerted by the fluid on the structure,
- The transfer of structure displacements to the fluid field,
- The fluid domain remeshing,
- The resolution of the dynamic structure problem.

2. ISIS-CFD, THE FLOW SOLVER

The ISIS-CFD flow solver, developed by the EMN (Equipe Modélisation Numérique) of the Fluid Mechanics Laboratory of the Ecole Centrale of Nantes, uses the incompressible unsteady Reynolds-averaged Navier-Stokes equations (RANSE). The solver is based on the finite volume method to build a spatial discretization of the transport equations. The face-based method is generalized to two-dimensional or three dimensional unstructured meshes for which non-overlapping control volumes are bounded by an arbitrary number of constitutive faces. The velocity field is obtained from the momentum conservation equations and the pressure field is extracted from the mass conservation constraint, or continuity equation, transformed into a pressure-equation. A second-order accurate three-level fully implicit time discretization is used. Surface and volume integrals are evaluated using second-order accurate approximation. In the case of turbulent flows, additional transport equations for modelled variables are solved in a form similar to the momentum equations and they can be discretized and solved using the same principles. Several turbulence models, ranging from the one-equation Spalart-Allmaras model (cf P. Spalart and S. Allmaras (1992)), two-equation $k-\omega$ closures (cf F.R. Menter (1993)), to a full stress transport $R_{ij}-\omega$ model (cf G.B. Deng and al (2005)), are implemented in the flow solver to take into account the turbulence phenomena.

3. FLOW FIELDS TRANFER BETWEEN SOLID AND FLUID MESHES

One of the difficulties in fluid/structure interaction is that the fluid and solid meshes do not match at the body boundary (cf A. de Boer et al. (2007)). In the special case of elongated structures, these ones are modelled by the beam theory. The solid grid is consequently the beam neutral line, whereas the structure is viewed from the fluid domain by a set of faces around this neutral line. In order to calculate the fluid efforts on

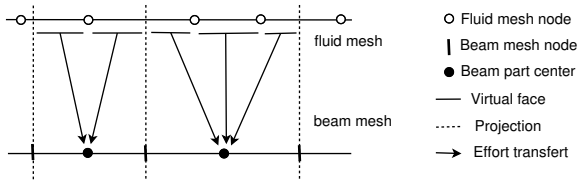


Figure 1: Fluid efforts interpolation on the beam

the beam, an interpolation is needed. The global conservation of the efforts being the most important property to fulfill, an ad-hoc interpolation has been developed.

The connectivities between fluid boundary-nodes and the beam sections are generated once at the launch of ISIS-CFD. They allow to determine to which part of the beam the orthogonal projection of the fluid boundary-nodes belongs. One has also the possibility to know the orthogonal projection of the beam nodes on the fluid boundary faces. With these data, the efforts are linearly interpolated. A fluid face, which entirely belongs to a beam section, provides the total fluid effort. In case of an orthogonal plane to a beam node cutting a fluid face, this one is virtually divided and the efforts are linearly distributed between the corresponding beam parts (cf fig. 1).

After the calculation of the beam deformation, this one is transmitted to the fluid grid with the same connectivities which were used before. The fluid mesh nodes which belong to the structure are displaced according to the beam kinematics.

4. FLUID DOMAIN REMESHING

At the end, the fluid mesh must be deformed in order to take into account the new beam position. ISIS-CFD integrates several remeshing tools. The first is an analytic regridding based on a weighting coefficient (cf fig. 2) which can be used with confidence for small and moderate deformations (cf A. Leroyer and M. Visonneau (2005)). Another regridding module has been developed in order to account for large deformations. This one is based on a pseudo-solid approach to build a consistent and robust unstructured grid deformation strategy. The fluid domain is considered as a linear elastic solid structure obeying structural equations which are linearised and used even in the case of large deformations since one does not need to follow the physics for this virtual elastic grid. The control parameters are the non-uniform Young modulus E, Poisson Coefficient μ and shearing coefficient G.

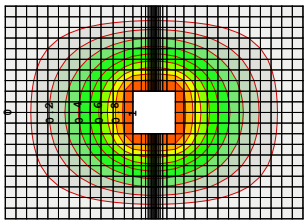


Figure 2: weighting coefficient calculated with a resolution of a lagrangian

4.1. Discretisations

In order to solve the structural problem, several behaviour laws have been studied. At first, a simple one, based on the isotropic case, has been implemented: $\vec{\vec{\sigma}} = 2\mu\ \vec{\vec{\epsilon}} + \lambda tr(\vec{\vec{\epsilon}})\ \vec{\vec{I}}$
So we have :

$$\int\int\int_V div(\vec{\vec{\sigma}})dV = 0 \Longleftrightarrow$$
$$\int\int_S \mu\ grad\vec{\vec{U}}\ .\vec{n}\,dS + (\mu+\lambda)\ div(\vec{\vec{U}})\ .\vec{n}\,dS = 0$$

where $\vec{U}$ is the field of cell centers displacements, with

$$\begin{cases} \mu = \dfrac{E}{2(1+\nu)} \\ \lambda = \dfrac{\nu E}{(1+\nu)(1-2\nu)} \end{cases}$$

This first deformation approach is easy to implement, but does not allow to control the shear stress coefficient G (in this isotropic case $G = G_{iso}$). With this coefficient and the introduced structural anisotropy, one can theoretically maintain a rigid movement near the body and keep the orthogonality of cells near the body wall, a highly desirable property for finite-volume discretizations. Therefore, a 2D/3D volume finite discretisation using G has been developed :

For an orthotropical (orthogonal + anisotropic) material, the following behaviour law is given (hat notation) :

$$\hat{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = K \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{G_{xy}}{K} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{yz}}{K} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{xz}}{K} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix}$$

with $K = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$ and $\epsilon_{xy} = \frac{1}{2}(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x})$ where G_{ij} is the shear stress coefficient in the $\vec{i}$ and $\vec{j}$ directions.

This symmetrical matrix gives an anisotropic discretisation with main directions. But the goal is to maintain the orthogonality of the mesh

around the structure during the deformation. So, one defines this matrix in a local basis $(\overrightarrow{n},\overrightarrow{t_2},\overrightarrow{t_3})$, with the vector $\overrightarrow{n}$ directed along the gradient vector of the weighting coefficient (their isovalues follow the body surfaces curves as shows in fig. 2). So the main deformation directions are local. And consequently one has a local behaviour depending to the local orientation of body surface. The general problem is expressed with the cartesian basis, so the behaviour law matrix must be rewritten in the cartesian basis.

Let P be the basis transformation matrix $(\overrightarrow{x},\overrightarrow{y},\overrightarrow{z})$ to $(\overrightarrow{n},\overrightarrow{t_2},\overrightarrow{t_3})$, one has :

$$\sigma_{(\overrightarrow{x},\overrightarrow{y},\overrightarrow{z})} = P\sigma_{(\overrightarrow{n},\overrightarrow{t_2},\overrightarrow{t_3})}P^{-1}$$

with

$$\overrightarrow{n} = n_x\overrightarrow{x} + n_y\overrightarrow{y} + n_z\overrightarrow{z}$$

$$P = \begin{bmatrix} n_x & P_{12} & P_{13} \\ n_y & P_{22} & P_{23} \\ n_z & P_{32} & P_{33} \end{bmatrix}$$

and consequently :

$$\widehat{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = K \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix}$$

The C_{ij} coefficients have been calculated with MAPLE.

One notices that the G_{ij} coefficients are independent. G_{nt_2} and G_{nt_3} are chosen equal and stronger than the isotropic value, instead of $G_{t_2t_3}$ which is taken equal to $G_{iso} = \frac{E}{(1+\nu)}$. Therefore, the behaviour of a transverse isotropic material is obtained.

In order to have the mode implicit resolution of the problem, one rewrites the previous matrix so that the isotropic term appears :

$$\widehat{\sigma} = K \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{G_{iso}}{K} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{iso}}{K} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{iso}}{K} \end{bmatrix} \widehat{\epsilon} + K\left[C - C_{iso}\right] \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix}$$

The first matrix is calculated as the isotropic case (implicit calculation). The second matrix is explicitly taken into account.

This remeshing technique allows to update a grid around deformable (cf fig. 3) or undeformable bodies. With the use of the shear stress coefficient G one can maintain a rigid movement near the body and so keep the orthogonality of cells near the body wall. One has also the possibility to couple this regridding process with one of the others in order to obtain a better control of the deformation. This module uses the same discretisation tools as the fluid solver and the connectivities are identical. Moreover, the mesh deformation is entirely parallelized using MPI communication tools.

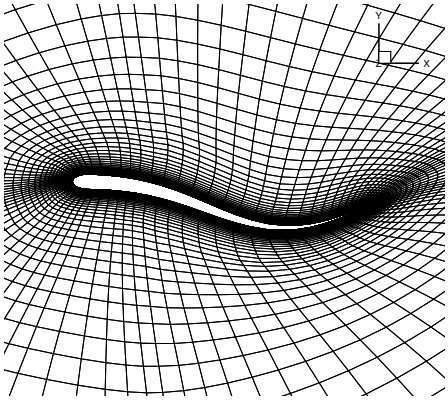


Figure 3: 3D meshes deformations around an eel-like body (ROBEA project) (cf A. Leroyer (2004))

5. THE STRUCTURAL SOLVER

5.1. Modelisation and hypothesis

With the distribution on the beam of the efforts exerted by the fluid, one can calculate its deformation, i.e. the displacements of its nodes. The structural solver is a "black box" based on the work of F. Boyer and D. Primault (cf F. Boyer (2004) and F. Boyer (2005)). The structural solver implemented into ISIS-CFD is based on the Cosserat approximation. The beam is considered as a monodimensional medium, built with a continuous stack of rigid micro-solid structures (cf E. and F. Cosserat (1909)). In this approximation, the beam sections must be rigid and plane. Without any other hypothesis, this kinematic is called "Timoshenko Reissner" (cf W. Weaver, Timoshenko and al (1990) and E. Reissner (1973)).

In our case, the beams are fine and elongated (cables, risers...). Then the Kirchoff hypothesis can be used with confidence : the beam sections are orthogonal to the neutral line of the beam. With this kinematic called "Kirchoff", two beam models are built : the "Rayleigh" model (cf L. Meirovitch (1967)) and the "Euler-

Bernoulli" model (cf L. Meirovitch (1967)). In this last model, the angular kinetic energy of the beam sections is neglected. With this simplification, the analytic equations are lighter. But a problem arises when the beam is free to rotate on itself.

The structural solver integrates both beam models resulting from the Kirchoff hypothesis : the "Rayleigh" and "Euler-Bernoulli" models.

5.2. The Euler-Bernoulli & Rayleigh kinematics

The Rayleigh kinematics (cf F. Boyer (2004) and F. Boyer (2005)) allow to solve correctly the cables problems thanks to the flexion-torsion coupling. Indeed the numerical solution is based on a variational formulation of the Kirchoff kinematics and on an exact modelisation of the most complex geometrical non-linearity : the flexion-torsion coupling (cf J.C. Alexander and Antman (1982)). An important notion appears here : the geometrically exact approach. This expression comes from Simo (cf J.C. Simo (1985)) and means that the approximations are done at the end of all the developments.

This kinematic needs a parametrisation. The rotation parametrisation is the most complex. In the Reissner kinematic case, the SO_3 Lie group is used. But the Kirchoff constraint reduces this space to a bidimensional SO_3 subspace. This one is parametrized by the position field of the beam neutral line and becomes the SO_2 plane rotations group. At first the rotation field is changed to the composition of two rotations : the movement of the beam neutral line only and the movement around this neutral line. Several parametrisations can be done for the rotation angles. In the implemented structural solver, the Eulerian parametrisation is used, due to its relative simplicity.

5.3. Validation test cases

5.3.1. Static validation

At first, in order to validate the solver implementation statically, a very simple 2D test case has been chosen: a beam with a uniform load. The implemented structural solver can treat the Euler-Bernoulli and Rayleigh kinetics which are compared with the help of this test case.

The comparison between the numerical and theoretical results shows that the solver consistently gives a numerical result which tends towards the theoretical value when the structural mesh is refined (cf tab. 1 and 2). With the Euler-Bernoulli kinetics, the convergence is 1.45 order for a load of 10 and 1.18 order for a load of 100. With the Rayleigh kinetics, the convergence is 1.2 order for a load of 10 and 1.6 order for a load of 100.

Load (q) Time step (h)	q=10	q=100
h=0.1	0.004907	0.048334
h=0.02	0.004252	0.043551
h=0.01	0.004312	0.042896
Theoretical value	0.0042857	0.042857

Table 1: *Euler-Bernoulli static validation (SI units).*

Load (q) Time step (h)	q=10	q=100
h=0.1	0.004837	0.048335
h=0.02	0.004369	0.043553
h=0.01	0.004320	0.042970
Theoretical value	0.0042857	0.042857

Table 2: *Rayleigh static validation (SI units).*

5.3.2. Dynamic validation

Then a dynamic validation has been performed : an embedded beam is loaded during a short period at the beginning, then is released in a fluid at rest (cf fig. 4). In the vacuum the beam will oscillate without damping at a frequency equal to the first theoretical eigenfrequency (cf C. Pertersen (1996)) :

$$f_1 = 0.5595 \frac{1}{L^2} \sqrt{\frac{EI}{m/L}}$$

In the following test-case the beam length L is set to $1m$, its thickness d to $0.01m$, its volumic

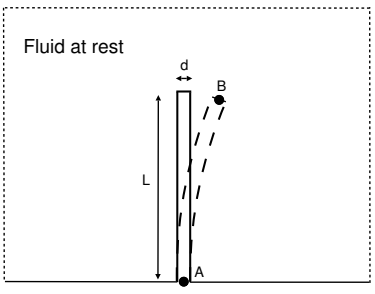


Figure 4: Free embedded (point A) beam in a fluid at rest

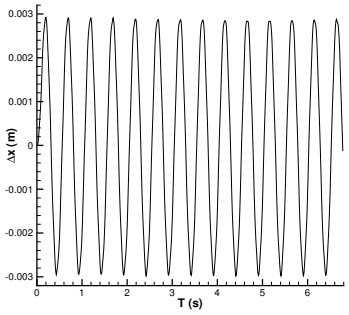


Figure 5: Oscillations of the point B in the vacuum

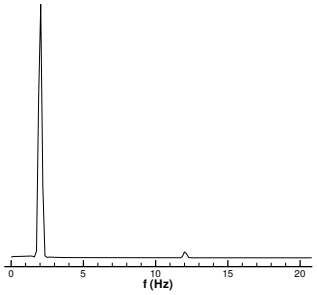


Figure 6: Eigenfrequencies of the beam oscillations

mass $\rho_{beam} = 2250\ SI$ and its Young modulus $E = 3500MPa$. Consequently, the theoretical first eigenfrequency f_1 is : $f_1 = 2.01Hz$. In the figure 5 we can see the evolution of the point B. The oscillations frequency is around $2.04Hz$ (cf fig. 6) which gives an relative error inferior to 2%.

When the beam oscillates in a fluid, like air or water, the oscillations are damped and this first eigenfrequency is modified. In air (with a viscosity set to 0.2 SI) this one is around $1.8Hz$. The higher is the fluid viscosity, the higher is the damping of the oscillations important.

5.4. First test-case and results

In this test-case the same embedded beam is used. But it is suddenly loaded by a constant incoming flow and oscillates as long as it reaches a steady deformation state (cf fig. 7). The test-case conditions are presented in the table 3. The resulting Reynold number is $Re = 50$.

The used fluid mesh is a totally unstructured mesh which contains around 30000 cells (cf fig. 8). The line which is the solid mesh is composed of 100 segments.

The time increment is $\delta t = 0.001s$ in order to

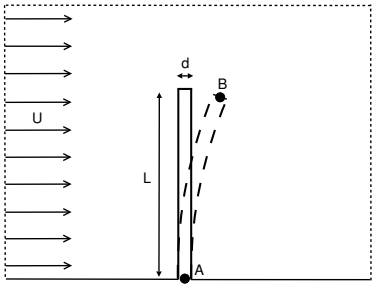


Figure 7: Embedded beam in a fluid in movement

Length	$L = 1\ m$
Thickness	$d = 0.01\ m$
Young modulus	$E = 3500\ MPa$
Solid volumic mass	$\rho_s = 1200\ kg\ m^{-3}$
Fluid volumic mass	$\rho_f = 1\ kg\ m^{-3}$
Dynamic viscosity	$\mu_f = 0.2\ Pa\ s$
Constant inflow velocity	$U = 10\ m\ s^{-1}$

Table 3: *Test-case conditions (SI units).*

reach approximatively more than 300 time steps per oscillation period, which is a good value to have a good accuracy.

The results show that the beam oscillates during several seconds and then finds a steady state. The free extremity (point B) moves aproximatively $1.5cm$ from the free steady state (cf fig. 9). The flow at the steady state has a big recirculation to the rear of the beam and two little on each sides (cf fig. 10).

6. CONCLUSION

In this paper the methods to perform simulations coupling a RANSE solver with a structural solver for elongated bodies in large deformations and

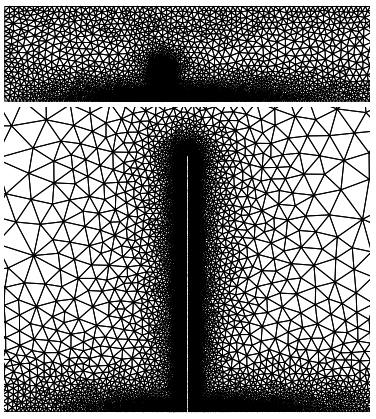


Figure 8: Mesh around the beam and far away

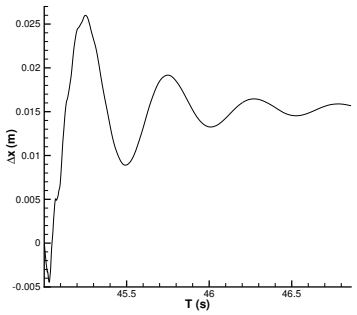


Figure 9: Oscillations of the point B in the fluid

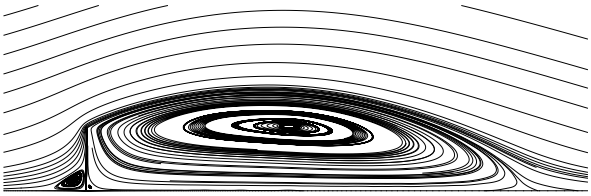


Figure 10: Flow around the deformed beam at the steady state

large displacements have been presented. This one is based on the Cosserat hypothesis with the Rayleigh or Euler-Bernoulli kinematics.

A general remeshing procedure is also detailed. This one is based on the pseudo-solid approximation and allows to control the mesh deformation through a particular local behaviour law.

First the 3D structural solver was verified with 2D standard test-cases. Then the coupled code was used to an example with 2D simple geometry.

In the future in order to improve the FSI into ISIS-CFD, the replacement of the Newmark algorithm for the time integration is planned, because it gives some high frequencies perturbations. More validation 2D & 3D test-cases will be performed with comparisons to experimental data.

7. REFERENCES

P. Spalart and S. Allmaras, 1992, A one-equation turbulence model for aerodynamic flows. In *AIAA 30th Aerospace Sciences Meeting* **92**: 0439.

F.R. Menter, 1993, Zonal two-equation $k-\omega$ turbulence models for aerodynamic flows. In *AIAA 24th Fluid Dynamics Conference*: 93-2906.

G.B. Deng and al, 2005, Three-dimensional flow computation with Reynolds stress and algebraic stress models. In *Proceedings of the ERCOFTAC International Symposium on Engineering Turbulence Modelling and Measurements* : 389-398.

A. de Boer et al., 2007, Review of coupling methods for non-matching meshes. In *Comput. Methods Appl. Mech. Engrg.* **196**: 515-1525.

A. Leroyer and M. Visonneau, 2005, Numerical methods for RANSE simulations of a self-propelled fish-like body. In *Journal of Fluids and Structures* **20**: 975-991.

A. Leroyer, 2004, Etude du couplage écoulement/mouvement pour des corps solides ou à déformation imposée par résolution des équations de Navier-Stokes. Contribution à la modélisation numérique de la cavitation. $ftp://ftpa.ec-nantes.fr/pub/DMN/Thesis/these_leroyer.pdf$

F. Boyer and D. Primault, 2004, Finite element of slender beams in finite transformations : a geometrically exact approach. In *International Journal for numerical methods in engineering* **59**: 669-702.

F. Boyer and D. Primault, 2005, Finite element of non-linear cables : applications to robotics. In *Far East Journal of Applied Mathematics* **19**: 1-34.

E. and F. Cosserat, 1909, Théorie des corps déformables. Hermann, Paris.

W. Weaver, Timoshenko and al, 1990, Vibrations problems in engineering. John Wiley & Sons, New York.

E. Reissner, 1973, On a one-dimensional large displacement finite-strain theory. Stud. Appl. Math. **52**: 87-95.

L. Meirovitch, 1967, Analytical methods in vibrations. Mc Millan Publish., New York.

J.C. Alexander and Antman, 1982, The anbiguous twist of Love. In *Quaterly of Applied mathematics* **2**: 83-92.

J.C. Simo, 1985, A finite strain beam formulation. The three-dimensional dynamic problem. Part I. In *Comp. Meth. Appl. Mech. Eng.* **49**: 55-70.

C. Pertersen, 1996, Dynamik der Baukonstruktionen. Vieweg, Braunschweig/Wiesbaden.

D.3 Errata

Un bogue sur la gestion des efforts fluides au sein du code IFS a entraîné plusieurs erreurs dans les résultats présentés au FIV2008.

1. Les ordres de convergence des modèles d'Euler-Bernoulli et de Rayleigh présentés à la section 5.3.1 ne sont pas égaux respectivement à 1.45 et 1.2. Les valeurs correctes sont données à la section §3.2.1.1.
2. Le cas-test 5.4 est repris dans le chapitre 6 à la section §6.1.1. Les figures 9 et 10 de la publication sont fausses. Elles ont été refaites dans ce manuscrit une fois le bogue corrigé.

Bibliographie

[1] *ROBCAD*. Software Hand Book, 2000. 50

[2] *DENNEB*. Software Hand Book, 2001. 50

[3] Al Mikdad, M., and Ibrahimbegovic, A. Dynamique et schémas d'intégration pour modèles de poutres géométriquement exacts. *Revue européenne des éléments finis 6* (1997), 471–502. 55, 57

[4] Alexander, J. C., and Antman, S. S. The ambiguous twist of love. *Quaterly of Applied Mathematics 2* (1982), 83–92. 50

[5] Aliabadi, S., Johnson, A., and Abedi, J. Comparison of finite element and pendulum models for simulation of sloshing. *Computers & Fluids 32* (2003), 535–545. 11

[6] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999. 63, 156

[7] Antman, S. S. *The theory of rods*. Springer, Berlin, 1972. 50

[8] Bathe, K. J., and Boulourchi, S. Large displacement analysis of three-dimensional beam structures. *Int. J. Numer. Meth. Engng. 14* (1979), 961–986. 50

[9] Batina, J. T. Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes. *AIAA Journal 28* (1990), 1381–1388. 102

[10] Becache, E., Chaigne, A., Derveaux, G., and Joly, P. Numerical simulation of a guitar. *Computers and Structures 83* (2005), 107–126. 14

[11] Beckert, A., and Wendland, H. Multivariate interpolation for fluid-structure interaction problems using radial basis functions. *Aerospace Sci. Technol. 5* (2001), 125–134. 153

[12] Behdinan, K., Stylianou, M. C., and Tabarrok, B. Static and dynamic analysis of flexible beams : a consistent updated lagrangian formulation. *Transactions of the CSME 21* (1997), 141–177. 50, 80

[13] Berglund, M. D. The Boeing Delta IV launch vehicle-pulse-settling approach for second-stage hydrogen propellant management. *Acta Astronautica 61* (2007), 416–424. 11

[14] Bisshop, K. E., and Drucker, D. E. Large deflections of cantilever beams. *Quaterly of Applied Mathematics 3* (1945), 272–275. 79

[15] Boyer, F., and Alamir, M. Robot anguille sous-marin en 3D. *Techniques de l'Ingénieur 7* (2006), 1–32. 11, 137

[16] Boyer, F., and Primault, D. Finite element of slender beams in finite transformations : a geometrically exact approach. *Int. J. Numer. Meth. Engng 59* (2004), 669–702. 49, 68, 78

[17] Boyer, F., and Primault, D. Finite element of nonlinear cables : applications to robotics. *Far East J. Appl. Math. 19* (2005), 1–34. 49, 78

[18] Braun, A. L., and Awruch, A. M. Finite element simulation of the wind action over bridge sectional models : Application to the Guama River Bridge (Para State, Brazil). *Finite Elements in Analysis and Design 44* (2008), 105–122. 7

[19] Burgess, J. J. Bending stiffness in a simulation of undersea cable deployment. *Int. J. Offshore and Polar Engineering 3* (1993), 197–204. 50

[20] Cardona, A., Gérardin, M., and Doan, D. B. Rigid and flexible joint modelling in multibody dynamics using finite element. *Comp. Meth. Appl. Mech. Eng. 89* (1991), 395–418. 69

[21] Chaplin, J. R., Bearman, P. W., Cheng, Y., Fontaine, E., Graham, J. M. R., Herfjord, K., Huera Huarte, F. J., Isherwood, M., Lambrakos, K., Larsen, C. M., Menegheni, J. R., Moe, G., Pattenden, R. J., Triantafyllou, M. S., and Willden, R. H. J. Blind predictions of laboratory measurements of vortex-induced vibrations of a tension riser. *Journal of Fluids and Structures 21* (2005), 25–40. xiii, 195, 197

[22] Chaplin, J. R., Bearman, P. W., Huera Huarte, F. J., and Pattenden, R. J. Laboratory measurements of vortex-induced vibrations of a vertical tension riser in a stepped current. *Journal of Fluids and Structures 21* (2005), 3–24. xi, 163, 195, 196

[23] Chikhaoui, O. *Simulations des grandes échelles et modélisations hybrides RANSE/LES pour le calcul d'écoulements turbulents de grande complexité.* PhD thesis, Ecole Centrale Nantes, France, 2007. 48

[24] Chung, J., and Hulbert, G. M. A time integration algorithm for structural dynamics with improved numerical dissipation : the generalized-α method. *Journal of Applied Mechanics 60* (1993), 371–375. 180

[25] Cirak, F., and Radovitzky, R. A Lagrangian-Eulerian shell-fluid coupling algorithm based on level sets. *Computers and Structures 83* (2005), 491–498. 14

[26] Cosserat, E., and Cosserat, F. *Théorie des corps déformables.* Hermann, 1909. 50

[27] Dauby, D. *Simulation numérique d'écoulements cavitants par résolution des équations de Navier-Stokes en moyenne de Reynolds. Application à la cavitation de tourbillon d'extrémité.* PhD thesis, Ecole Centrale Nantes, France, 2007. 48, 137

[28] de Boer, A., van Zuijlen, A. H., and Bijl, H. Review of coupling methods for non-matching meshes. *Comput. Methods Appl. Mech. Engrg. 196* (2007), 515–1525. 16, 153

[29] Degand, C. A three-dimensional torsional springs analogy method for unstructured dynamic meshes. *Computers & Structures 80* (2002), 305–316. 102

[30] Demirdžić, I., and Muzaferija, S. Finite volume method for stress analysis in complex domains. *Int. J. Num. Meth. Engrg. 37* (1994), 3751–3766. 21

[31] Demirdžić, I., and Muzaferija, S. Numerical method for coupled fluid flow, heat transfert and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Computer Methods in Applied Mechanics and Engineering 125* (1995), 235–255. 34

[32] Demirdžić, I., Muzaferija, S., and Perić, M. Benchmark solutions of some structural analysis problems using finite-volume method and multigrid acceleration. *Int. J. Num. Meth. Engrg. 40* (1997), 1893–1908. 21

[33] Dettmer, W. G. *Finite Element Modelling of Fluid Flow with Moving Free Surfaces and Interfaces Including Fluid-Solid Interaction.* PhD thesis, School of Engineering, University of Wales, Swansea, 2004. 170

[34] Eberly, D. Geometric Tools, LLC. http://www.geometrictools.com. 157

[35] Ezcurra, A., Elejabarrieta, M. J., and Santamaria, C. Fluid-structure coupling in the guitar box : numerical and experimental comparative study. *Applied Acoustics 66* (2005), 411–425. 14

[36] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer methods in applied mechanics and engineering 163* (1998), 231–245. 102

[37] Farhat, C., Geuzaine, P., and Brown, G. Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers & Fluids 32* (2003), 3–29. 13

[38] Farhat, C., Lesoinne, M., and LeTallec, P. Load and motion transfer algorithms for fluid.structure interaction problems with non-matching discrete interfaces : momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput. Methods Appl. Mech. Engrg. 157* (1998), 95–114. 16, 153

[39] Farnell, D. J. J., David, T., and Barton, D. C. Numerical model of self-propulsion in a fluid. *Journal of the royal society Interface 2* (2005), 79–88. 10

[40] Ferziger, J. H., and Perić, M. *Computational methods for fluid dynamics.* Springer Verlag, Berlin - New York, 2002. 3rd Edition. 34

[41] Frandsen, J. B. Numerical bridge deck studies using finite elements. Part I : flutter. *Journal of Fluids and Structures 19* (2004), 171–191. 7

[42] Girodroux-Lavigne, P., Grisval, J. P., Guillemot, S., Henshaw, M., Karlsson, A., Selmin, V., Smith, J., Teupootahiti, E., and Winzell, B. Comparison of static and dynamic fluid-structure interaction solutions in the case of a highly flexible modern transport aircraft wing. *Aerospace Science and Technology 7* (2003), 121–133. 13

[43] Glueck, M., Breuer, M., Durst, F., Halfmann, A., and Rank, E. Computation of fluid-structure interaction on lightweight structures. *Wind engrg. and industrial aerodynamics 89* (2001), 1351–1368. 21, 166, 168

[44] Greco, M., Landrini, M., and Faltinsen, O. M. Impact flows and loads on ship-deck structures. *Journal of Fluids and Structures 19* (2004), 251–275. 10

[45] Green, A. E., and Naghdi, P. M. Non-isothermal theory of rods, plates and shells. *Int. J. of Solids and Structures 6* (1970), 209–244. 50

[46] Guilmineau, E. *Contribution à la prédiction du décrochage sur des ailes en incidence au moyen des équations de Navier-Stokes-Reynolds.* PhD thesis, Ecole Centrale de Nantes, 1995. 46

[47] Hay, A. *Etude des stratégies d'estimation d'erreur numérique et d'adaptation locale de maillages non-structurés pour les équations de Navier-Stokes en moyenne de Reynolds.* PhD thesis, Ecole Centrale de Nantes, France, 2004. 30, 48

[48] Hilbert, H. M., Hughes, T. J. R., and Taylor, R. L. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics 5* (1977), 283–292. 182

[49] Hrona, J., and Madlikb, M. Fluid-structure interaction with applications in biomechanics. *Nonlinear Analysis : Real World Applications 8* (2007), 1431–1458. 11

[50] Huebner, B., Walhorn, E., and Dinkier, D. A monolithic approach to fluid-structure interaction using space-time finite element. *Comput. Methods Appl. Mech. Engrg. 193* (2004), 2087–2104. 15, 21, 171

[51] Ibrahimbegovic, A., and Mamouri, S. On rigid components and joint constraints in nonlinear dynamics of flexible multibody systems employing 3D geometrically exact beam model. *Comp. Meth. Appl. Mech. Eng. 188* (2000), 805–831. 69

[52] Irvine, H. M. *Cable Structures.* Massachussetts Inst. Tech. Press, 1981. 50

[53] Issa, R. I. Solution of the implicitly discretized fluid flow equations by operator-splitting. *Journal of Computational Physics 62* (1985), 40–65. 42

[54] Jasak, H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows.* PhD thesis, University of London, 1996. 35

[55] Le Tallec, P. *Introduction à la dynamique des structures.* Ellipse, 2000. 83, 88

[56] Leonard, B. P. Simple high-accuracy resolution program for convective modelling of discontinuities. *International Journal for Numerical Methods in Fluids 8* (1988), 1291–1318. 35

[57] Leroyer, A. *Etude du couplage écoulement/mouvement pour des corps solides ou à déformation imposée par résolution des équations de Navier-Stokes. Contribution à la modélisation numérique de la cavitation.* PhD thesis, Ecole Centrale de Nantes, France, 2004. disponible en ligne : ftp://ftpa.ec-nantes.fr/user/ftp-emn/pub/Thesis/these_leroyer.pdf. 49, 101, 128, 129, 137, 160, 211, 249, 254, 263, 264, 265

[58] Leroyer, A., and Visonneau, M. Numerical methods for RANSE simulations of a self-propelled fish-like body. *Journal of Fluids and Structures 20* (2005), 975–991. 11, 137

[59] Leroyer, A., and Visonneau, M. Simulation d'un robot-anguille auto-propulsé par résolution des équations de Navier-Stokes. *Mécanique et Industries 7* (2006), 189–199. 11, 137

[60] Liu, D., and Lin, P. A numerical study of three-dimensional liquid sloshing in tanks. *Journal of Computational Physics 227* (2008), 3921–3939. 11

[61] Liu, W. K., Liu, Y., Farrell, D., Zhang, L., Wang, X. S., Fukui, Y., Patankar, N., Zhang, Y., Bajaj, C., Lee, J., Hong, J., Chen, X., and Hsu, H. Immersed finite element method and its applications to biological systems. *Comput. Methods Appl. Mech. Engrg. 195* (2006), 1722–1749. 11

[62] Longatte, E., Bendjeddou, Z., and Souli, M. Methods for numerical study of tube bundle vibrations in cross-flows. *Journal of Fluids and Structures 18* (2003), 513–528. 7

[63] Mavriplis, D. J. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *16th AIAA Computational Fluid Dynamics Conference* (Orlando, FL, 2003), AIAA Paper 2003-3986. 39

[64] Meirovitch, L. *Analytical methods in vibrations*. Mc Millan Publish., 1967. 50

[65] Menter, F. R. Zonal two-equation $k-\omega$ turbulence models for aerodynamic flows. In *AIAA 24th Fluid Dynamics Conference* (Orlando, FL, 1993), AIAA Paper 93-2906. 46

[66] Menter, F. R. Two-equation eddy viscosity turbulence models for engineering applications. *AIAA Journal 32* (1994), 1299–1310. 47

[67] MϵTiS. Family of Multilevel Partitionning Algorithms. http://www-users.cs.umn.edu/~karypis/metis/. 44, 190

[68] Miyake, R., Matsumoto, T., Zhu, T., and Abe, N. Experimental Studies on the Hydroelastic Response due to Springing Using a Flexible Mega-Container Ship Model. 8^{th} International Conference on HydroDynamics. 214

[69] Molin, B. *Hydrodynamique des structures offshore*. Editions TECHNIP, 2002. 10, 206

[70] MPI. The Message Passing Interface (MPI) standard. http://www-unix.mcs.anl.gov/mpi. 44

[71] Pahlke, K., and van der Wall, B. G. Chimera simulations of multibladed rotors in high-speed forward flight with weak fluid-structure-coupling. *Aerospace Science and Technology 9* (2005), 379–389. 13

[72] Patankar, S. V. *Numerical heat tranfert and fluid flow*. McGraw-Hill Book Company Inc., New York, 1980. 34, 42

[73] Païdoussis, M. P. *Fluid-Structure Interactions. Volume 1 : Slender Structure & Axial Flow*. Academic Press, 1998. 10

[74] Païdoussis, M. P. *Fluid-Structure Interactions. Volume 2 : Slender Structure & Axial Flow*. Academic Press, 2004. 10

[75] Pedley, T. J., and Hill, S. J. Large-amplitude undulatory fish swimming : fluid mechanics coupled to internal mechanics. *The Journal of Expirement Biology 202* (1999), 3431–3438. 10

[76] Petersen, C. *Dynamik der Baukonstruktionen*. Vieweg, 1996. 72

[77] Piperno, S., and Farhat, C. Partitioned procedures for the transient solution of coupled aeroelastic problems Part II : energy transfer analysis and three-dimensional applications. *Comput. Methods Appl. Mech. Engrg. 190* (2001), 3147–3170. 17

[78] Piperno, S., Farhat, C., and Larrouturou, B. Partitioned procedures for the transient solution of coupled aroelastic problems Part I : Model problem, theory and two-dimensional application. *Coraput. Methods Appl. Mech. Engrg. 124* (1995), 79–112. 17

[79] Primault, D. *Modélisation géometriquement exacte de poutres fines - application a la robotique.* PhD thesis, Ecole Centrale Nantes, Université de Nantes, 2003. 49, 57, 59, 62, 71, 78

[80] Reissner, E. On a one-dimensional large displacement finite-strain theory. *Stud. Appl. Math. 52* (1973), 87–95. 50

[81] Rhie, C. M., and Chow, W. L. A numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation. *AIAA Journal 17* (1983), 1525–1532. 41

[82] Robbe, M. F., Lepareux, M., Treille, E., and Cariou, Y. Numerical simulation of a Hypothetical Core Disruptive Accident in a small-scale model of a nuclear reactor. *Nuclear Engineering and Design 223* (2003), 159–196. 7

[83] Royis, P. *Mécanique des milieux continus, cours, exercices et problèmes.* ENTPE,Presse universitaire de Lyon, 2005. 104, 105

[84] Sathe, S., Benney, R., Charles, R., Doucette, E., Miletti, J., Senga, M., Stein, K., and Tezduyar, T. E. Fluid-structure interaction modeling of complex parachute designs with the space-time finite element techniques. *Computers & Fluids 36* (2007), 127–135. 5

[85] Servera, G. A weak coupling method between the dynamics code HOST and the 3D unsteady euler code WAVES. *Aerosp. Sci. Technol. 5* (2001), 397–408. 13

[86] Simo, J. C. A finite strain beam formulation. the three-dimensional dynamic problem. Part I. *Comp. Meth. Appl. Mech. Eng. 49* (1985), 55–70. 51

[87] Simo, J. C., and Fox, D. D. On a stress resultant geometrically exact shell model. Part I : formulation and optimal parametrization. *Comp. Meth. Appl. Mech. Eng. 72* (1989), 267–304. 51

[88] Simo, J. C., Fox, D. D., and Rifai, M. S. On a stress resultant geometrically exact shell model. Part II : The linear theory, computational aspects. *Journal of Computational Methods in Applied Engineering 73* (1989), 53–92. 51

[89] Simo, J. C., Fox, D. D., and Rifai, M. S. On a stress resultant geometrically exact shell model. Part III : computational aspects of the non linear theory. *Comp. Meth. Appl. Mech. Eng. 79* (1990), 21–70. 51

[90] Simo, J. C., and Wong, K. K. Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum. *Int. J. Numer. Meth. Engng. 31* (1991), 19–52. 64

[91] Smith, M. J., Cesnik, C. E. S., and Hodges, D. H. Evaluation of some data transfer algorithms for noncontiguous meshes. *J. Aerospace Engrg. 13* (2000), 52–58. 153

[92] Smith, M. J., Hodges, D. H., and Cesnik, C. E. S. Evaluation of computational algorithms suitable for fluid-structure interactions. *J. Aircraft 37* (2000), 282–294. 153

[93] Spalart, P. R., and Allmaras, S. R. A One-Equation Turbulence Model for Aerodynamic Flows. In *AIAA 30th Aerospace Sciences Meeting* (Reno, NV, 1992), AIAA Paper 92-0439. 44

[94] Sreejith, B., Jayaraj, K., Ganesan, N., Padmanabhan, C., Chellapandi, P., and Selvaraj, P. Finite element analysis of fluid-structure interaction in pipeline systems. *Nuclear Engineering and Design 227* (2004), 313–322. 11

[95] Stein, K., Benney, R., Tezduyar, T., and Potvin, J. Fluid-structure interactions of a cross parachute : numerical simulation. *Comput. Methods Appl. Mech. Engrg. 191* (2001), 673–687. 5, 21

[96] Steindorf, J. *Partitionerte Verfahren fuer Probleme der Fluid-Struktur Wechselwirkung*. PhD thesis, Technischen Universitaet Braunschweig, 2002. 170

[97] Tadjbakhsh, I., and Lagoudas, D. C. Variational theory of motion of curved twisted and extensible elastic rods. *Int. J. Engrg. Sci. 32* (1994), 569–577. 50

[98] Thévenaz, P., Blu, T., and Unser, M. Interpolation revisited. *IEEE Trans. Med. Imaging 19* (2000), 739–758. 152

[99] Turnock, S. R., and Wright, A. M. Directly coupled fluid structural model of a ship rudder behind a propeller. *Marine Structures 13* (2000), 53–72. 10

[100] Valdés Vázquez, J. G. *Nonlinear Analysis of Orthotropic Membrane and Shell Structures Including Fluid-Structure Interaction*. PhD thesis, Université polytechnique de Catalogne, 2007. 173, 176, 177, 180

[101] van der Ploeg, A., Raven, H., Windt, J., Leroyer, A., Queutey, P., Deng, G., and Visonneau, M. Computation of Free-surface Viscous Flows at Model and Full Scale – a Comparison of Two Different Approaches. 27 Symposium on Naval Hydrodynamics. 137

[102] Vasseur, X. *Etude numérique de techniques d'accélération de convergence lors de la résolution des équations de Navier-Stokes en formulation découplée ou fortement couplée*. PhD thesis, Ecole Centrale de Nantes, Nantes, 1998. 43

[103] Walhorn, E., Koelke, A., Huebner, B., and Dinkler, D. Fluid-structure coupling within a monolithic model involving free surface flow. *Computers & structures 83* (2005), 2100–2111. 15

[104] Wall, W. A., and Ramm, E. Fluid-structure interaction based upon a stabilized (ALE) finite element method. E. and S. Idelsohn, Eds., IV World Congress on Computational Mechanics. 170

[105] Weaver, W., Timoshenko, S. P., and Young, D. H. *Vibration problems in engineering*. John Wiley & Sons, 1990. 50

[106] Wilcox, D. C. Reassessment of the Scale-Determining Equation for Advanced Turbulence Models. *AIAA Journal 26*, 11 (1988), 1299–1310. 46

[107] Wilcox, D. C. *Turbulence Modeling for CFD, Second Edition*. DCW Industries Inc., 5354 Palm Drive, La Cañada, California, 2002. 46

[108] Wood, W. L., Bossak, M., and Zienkiewics, O. C. An alpha modification of Newmark's method. *Inernational Journal for Numerical Methods in Engineering 15* (1980), 1562–1566. 182

[109] Xia, G., and Lin, C. L. An unstructured finite volume approach for structural dynamics in response to fluid motions. *Computers & structures 86* (2008), 684–701. 21

[110] Xing, Y., Hadzic, I., and Perić, M. Predictions of Floating-Body Motion in Viscous Flow. In 4^{th} *Numerical Towing Tank Symposium* (Hamburg, 2001). 129

[111] Zhang, A., and Suzuki, K. A comparative study of numerical simulations for fluid-structure interaction of liquid-filled tank during ship collision. *Ocean Engineering 34* (2007), 645–652. 11

[112] Zhang, R. J., Wang, W. Q., Hou, S. H., and Chan, C. K. Seismic analysis of nuclear reactor core. *Computers and Structures 79* (2001), 1395–1404. 7

Résumé :

Les moyens de calcul actuels conduisent les différentes disciplines scientifiques à se rapprocher, afin de prendre en compte des phénomènes physiques de plus en plus complexes. Ainsi, un axe de recherche de l'Équipe de Modélisation Numérique du Laboratoire de Mécanique des fluides UMR6598 de l'École Centrale Nantes est l'Interaction Fluide-Structure (IFS). Dans ce contexte, le développement d'un solveur structure grand déplacement, limité aux corps élancés, puis son couplage avec le code RANSE volumes-finis non-structuré ISIS ont été réalisés. Le solveur poutre s'appuie sur la théorie de Cosserat et sur la méthode dite "géométriquement exacte". Une attention particulière a été portée au couplage en espace sur les interpolations et au transfert des informations à l'interface fluide-structure, afin d'être le plus précis possible et d'assurer la conservation des efforts. Le code poutre pouvant prendre en compte de grands mouvements, une technique originale a dû être écrite pour mettre à jour le domaine de calcul fluide. Elle se base sur l'approche pseudo-solide, et permet via une loi de comportement locale du pseudo-solide, de contrôler finement la déformation du maillage. Chaque partie du code IFS a été validée : le solveur poutre sur des cas-tests 2D/3D, en statique et en dynamique, en petit et grand déplacement ; le module de remaillage sur des géométries variées et en calcul parallèle. Enfin, quelques applications IFS ont été traitées : deux exemples bidimensionnels, l'un stationnaire, l'autre très instationnaire ; puis, le programme a montré ses possibilités en tridimensionnel sur un câble déformable fixé à une extrémité et subissant un courant, ou encore un riser tracté dans un environnement multifluide au repos.

Abstract :

The current computational resources lead the different scientific disciplines to get closer to each other, in order to consider more and more complex phenomena. Thus, one of the axis of research of the CFD Team from École Centrale Nantes is the Fluid-Structure Interaction (FSI). In this context, the development of a large displacement structural solver for elongated bodies and its coupling with the non-structured finite volume RANSE solver, ISIS, has been done. This thin beam solver relies on the Cosserat theory and on the "geometrically exact" approach. The space coupling on the interpolations and the information transfer at the fluid-structure interface were realised with caution, in order to be as accurate as possible and to fulfill the load conservation. Since the beam solver can be used for great displacements, an original technique had to be built to update the fluid computational domain. It is based on the pseudo solid approach, which allows a precise control over the mesh deformation through a local behaviour law of the pseudo-solid. Each part of the FSI code has been checked : the structural solver on 2D/3D test-cases, in statics and in dynamics, in small and large displacements ; the remeshing module has been tested on various geometries and with MPI. Finally, some FSI applications have been performed : two bidimensional examples, a steady case and a fully unsteady one ; then, the program has shown its capabilities in 3D on a deformable cable in a current with a fixed end. The computation of a riser towed in a multifluid environment at rest has also been done and studied.