



HAL
open science

Développement de méthodes de couplage aéro-thermo-mécanique pour la prédiction d'instabilités dans les structures aérospatiales chaudes

Jean-Didier Garaud

► **To cite this version:**

Jean-Didier Garaud. Développement de méthodes de couplage aéro-thermo-mécanique pour la prédiction d'instabilités dans les structures aérospatiales chaudes. Mathématiques [math]. Université Pierre et Marie Curie - Paris VI, 2008. Français. NNT: . tel-00359175

HAL Id: tel-00359175

<https://theses.hal.science/tel-00359175>

Submitted on 6 Feb 2009

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.



**THÈSE DE DOCTORAT DE
L'UNIVERSITÉ PIERRE ET MARIE CURIE**

Spécialité : Mathématiques

Présentée par
M. Jean-Didier GARAUD

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Développement de méthodes de couplage
aéro-thermo-mécanique pour la prédiction d'instabilités
dans les structures aérospatiales chaudes

Soutenue le 25 novembre 2008, devant le jury composé de :

M. Yvon MADAY	Professeur à l'UPMC, Paris 6	<i>Directeur de thèse</i>
M. Arjen ROOS	Ingénieur de recherches à l'ONERA	<i>Encadrant</i>
M. Thierry COUPEZ	Directeur de recherches, École des Mines ParisTech	<i>Rapporteur</i>
M. Christian REY	Professeur au LMT Cachan, ENS	<i>Rapporteur</i>
M. David DUREISSEIX	Professeur au LMGC, Montpellier 2	<i>Président</i>
Mme Laure BALLESTER	Ingénieur de recherches à SNECMA	<i>Examineur</i>
M. Serge PIPERNO	Directeur du Cermics, École des Ponts ParisTech	<i>Examineur</i>
M. François-Xavier ROUX	Professeur à l'UPMC & ONERA	<i>Examineur</i>

Table des matières

Introduction	3
I État de l’art	9
1 Mécanique des fluides et des structures	11
1.1 Mécanique des fluides	11
1.1.1 Équations de Navier-Stokes	11
1.1.2 Discrétisation spatiale par volumes finis	12
1.1.3 Discrétisation temporelle et états stationnaires	12
1.2 Mécanique des solides déformables	13
1.2.1 Lois de comportement	14
1.2.2 Conditions aux limites	15
1.2.3 Discrétisation et résolution	15
1.2.4 ZéBuLoN, code <i>objet</i>	15
1.3 Thermique du solide	16
1.4 Problèmes couplés	17
1.4.1 Aéro-mécanique	17
1.4.2 Aéro-thermique	18
1.4.3 Thermo-mécanique	19
2 Méthodes de couplage partitionnées	21
2.1 Introduction	21
2.2 Terminologie couplage faible / couplage fort	23
2.3 Couplage faible	23
2.3.1 Algorithme sous-cyclé	24
2.3.2 Variante parallèle	25
2.3.3 Algorithme à échanges désynchronisés	26
2.3.4 Prédiction	27
2.3.5 Fluide-structure ou structure-fluide ?	28
2.4 Couplage partitionné fort	28
2.4.1 Corrections par itérations, méthode de Jacobi	28
2.4.2 Autres méthodes de couplage fort	29
2.5 Critères de qualité	30
2.5.1 Création d’énergie à l’interface	30
2.5.2 Stabilité	30
2.6 Découplage en espace	31

2.7	Méthodes multi-domaines et conditions de raccord	32
II	Couplage de codes de calcul : étude théorique	33
3	Paradigme du couplage de code	35
3.1	Le problème couplé	35
3.2	Coupleur pilote ou codes indépendants	36
3.3	Quantités échangées, champs de vecteur	38
3.4	Assemblage d'un algorithme de couplage	39
3.5	Découplage spatial	40
3.6	Découplage temporel	42
3.6.1	Pile de réception et système du <i>serveur</i>	43
3.7	Moteur de couplage	44
3.7.1	Le moteur de couplage dans ZéBuLoN	44
3.7.2	Le moteur MpCCI	45
3.7.3	Mise en place d'algorithmes de couplage	46
3.7.4	Vérification globale de la convergence du problème	46
4	Développements préliminaires	49
4.1	Maillage mobile	49
4.1.1	Introduction	49
4.1.2	Comparaison de trois méthodes	51
4.1.3	Quelques cas tests	54
4.1.4	Intégration logicielle	57
4.1.5	Conclusion	57
4.2	Développement et validation d'une condition au bord couplée en pression	58
4.2.1	Solution analytique	58
4.2.2	Comparaison avec un calcul découplé	59
4.2.3	Confrontation entre le calcul couplé et le calcul non couplé	60
4.2.4	Conclusion	60
5	Le paradigme par l'exemple	63
5.1	Dirichlet→Neumann synchrone : problème de la barre bi-matériau	63
5.1.1	Présentation du problème	64
5.1.2	Algorithme de couplage partitionné	64
5.1.3	Illustrations numériques	66
5.1.4	Raccord par conditions mixtes	70
5.2	Couplage asynchrone : problème de thermo-mécanique	71
5.2.1	Présentation d'un problème modèle	71
5.2.2	Approche partitionnée de ce problème	71
5.2.3	Résultats sur maillages coïncidents	73
5.2.4	Résultats sur maillages non-coïncidents	73
5.2.5	Maillage quadratique	76
5.2.6	Asynchronisme et corrections	76
5.3	Couplage à trois codes	78
5.3.1	Algorithme synchrone	78

5.3.2	Algorithme asynchrone	79
III	Étude d'un tube de tuyère	81
6	Premières approches	83
6.1	Présentation générale de la tuyère	83
6.2	Schématisation d'un vol	86
6.2.1	Éprouvette et discrétisation	86
6.2.2	Chargements appliqués	87
6.3	Calculs fluides découplés	89
6.3.1	Conditions de fonctionnement	89
6.3.2	Modèles fluides 1D & 2D	89
6.3.3	Tube fluide 3D	91
6.4	Calculs mécaniques découplés du solide	95
6.4.1	Loi de comportement de l'Inconel 600	95
6.4.2	Calculs plans	96
6.4.3	Calculs 3D	98
6.4.4	Calculs 3D sur un tube raccourci	98
6.5	Calculs thermiques découplés du solide	99
6.5.1	Loi de comportement thermique	99
6.5.2	Calculs 2D	99
6.5.3	Calculs 3D	102
6.6	Bilan	102
7	Calculs couplés mécaniques	105
7.1	Calcul couplé mécanique des sollicitations d'un tube de tuyère	105
7.2	Modèles retenus	106
7.3	Algorithme de couplage	107
7.4	Mise en place du calcul	107
7.5	Résultats	111
8	Calculs couplés thermiques	115
8.1	Couplage par la méthode Dirichlet→Neumann	115
8.1.1	Première approche	115
8.1.2	Calcul de l'erreur	119
8.2	Prédicteurs	124
8.3	Critères de convergence et phase de correction	125
8.4	Pas de temps adaptatif	131
8.4.1	Description de la méthode	131
8.4.2	Résultats obtenus	133
8.5	Couplage par la méthode Dirichlet→Fourier	135
8.5.1	Loi de comportement flux-température	135
8.5.2	Résultats	136
8.6	Conclusion	139

9	Calculs couplés aéro-thermo-mécaniques	141
9.1	Calcul aéro-thermo-mécanique simplifié	142
9.1.1	Présentation des simplifications	142
9.1.2	Algorithme de couplage retenu	143
9.1.3	Résultats	146
9.2	Calcul aéro-thermo-mécanique complet	147
9.2.1	Algorithmes envisagés	149
9.2.2	Solution utilisable dans l'état actuel	151
	Conclusions et perspectives	153
	Annexes	161
A	Notations	163
B	Inconel 600	165
B.1	Comportement thermique	165
B.2	Comportement mécanique	166
C	Précisions logicielles	169
C.1	Brève description des diagrammes UML	169
C.2	Architecture du <i>plugin</i> de couplage ZéBuLoN	170
C.2.1	Classe "composant de problème couplé"	170
C.2.2	Classes "conditions au bord et paramètres couplés"	171
C.2.3	Factorisation des classes imposant la valeur de degrés de libertés	172
C.2.4	Algorithmes de couplage développés	173
D	Mises en données	175
D.1	Mises en données mécanique découplée pour ZéBuLoN	175
D.2	Mises en données mécanique couplée pour ZéBuLoN	176
D.3	Mises en données thermique couplée ZéBuLoN	177
	Bibliographie	179

Liste des tableaux

5.1	Influence des modules d'Young sur la convergence	69
5.2	Influence de l'élanement sur la convergence	70
B.1	Conductivité thermique et chaleur spécifique de l'Inconel 600	166

Liste des algorithmes

1	Algorithme partitionné séquentiel	24
2	Algorithme itératif corrigé	29
3	Algorithme partitionné avec corrections	65
4	Algorithme asynchrone de couplage en série	72
5	Algorithme synchrone de couplage à 3 codes	79
6	Algorithme itératif de résolution du problème thermique	126

Liste des figures

1	Simulation d'une configuration voilure–fuselage–nacelle	5
2	Exemple de couplage mécanique	5
3	Exemples de couplages thermiques	6
4	Moteur cryotechnique Vulcain 2 au banc d'essai	6
1.1	Problèmes multi-physiques : notations	18
2.1	Algorithme partitionné séquentiel	24
2.2	Algorithme partitionné séquentiel sous-cyclé	25
2.3	Algorithme partitionné parallèle	26
2.4	Algorithme partitionné à échanges désynchronisés	27
3.1	Couplage par code pilote	37
3.2	Couplage par bibliothèque externe	38
3.3	Quelques stratégies à assembler pour réaliser un algorithme couplé	40
3.4	Diagramme simplifié de la classe <i>moteur de couplage</i>	45
4.1	Gonflement d'un tube sous l'effet de la pression exercée par le fluide, et maillage fluide déformé en conséquence	50
4.2	Conservation des propriétés du maillage	51
4.3	Modèle du triplet de ressorts 1-D	53
4.4	Ressorts 1-D : préservation des couches limites	53
4.5	Ressorts 1-D : centrifugation des nœuds vers le bord	54
4.6	Cas test de la languette	55
4.7	Cas test de la languette	55
4.8	Cas test du propulseur	56
4.9	Cas de validation : poutre en flexion soumise à une pression non uniforme	58
4.10	Maillage de la poutre utilisé pour le calcul de validation	59
4.11	Résultat du calcul découplé de la poutre en flexion	59
4.12	Comparaison entre la solution analytique et le calcul découplé	60
4.13	Représentation schématique du domaine fluide et de la barre	60
4.14	Profil de pression obtenu le long de la colonne	61
4.15	Visualisation du résultat du calcul semi-couplé de validation	61
4.16	Résultat du calcul semi-couplé de validation	61
5.1	Barre bi-matériau décomposée en deux sous-domaines	64
5.2	Algorithme de Dirichlet→Neumann	65
5.3	Modèle discret et conditions aux limites pour la barre bi-matériau	66

5.4	Solution de référence du problème de la barre bi-matériau	67
5.5	Évolution de la solution au cours des itérations de couplage	68
5.6	Évolution de l'erreur à l'interface au cours des itérations de couplage	68
5.7	Calcul couplé thermo-mécanique à maillages coïncidents	73
5.8	Comparaison des résultats couplés par méthodes interne et externe	74
5.9	Maillages non-coïncidents du cube	74
5.10	Calcul couplé thermo-mécanique à maillages non-coïncidents	75
5.11	Comparaison des déplacements pour un calcul couplé sur maillages non-coïncidents	75
5.12	Comparaison du calcul thermo-mécanique pour des maillages linéaire et quadratique	76
5.13	Comparaison des déplacements le long des arêtes supérieures et droite pour un maillage linéaire et quadratique	77
5.14	Communications requises pour un couplage à trois codes	78
6.1	Vulcain II	83
6.2	Schéma simplifié de la tuyère Vulcain II	84
6.3	Gonflement (bulging) d'un tube	85
6.4	Géométrie simplifiée retenue pour l'étude couplée	87
6.5	Schématisation de l'évolution du chargement thermique	88
6.6	Température, pression et vitesse calculées sur le modèle 2D	90
6.7	Écoulement fluide sur une configuration 3D élargie	92
6.8	Écoulement dans un tube complet	93
6.9	Écoulement dans les 20 premiers centimètres du tube : champ de température	94
6.10	Écoulement dans les 20 premiers centimètres du tube : lignes de courant et vitesse du fluide.	94
6.11	Profil de pression le long du tube	95
6.12	Maillages et géométries éliminées	96
6.13	Déplacement (flèche) au milieu du tube, pour différentes pressions uniformes.	97
6.14	Calcul 3D sur un tube de 20 cm de long	98
6.15	Tube déformé sous une pression de 30 bars, à la 600 ^e seconde.	99
6.16	Déplacement vertical le long de l'axe principal du tube	100
6.17	Maillages 2D de référence et retenu d'une section de tube	101
6.18	Comparaison des champs thermiques	101
6.19	Champ thermique 3D obtenu par un calcul découplé	102
7.1	Algorithme de couplage mécanique	107
7.2	Implémentation de l'algorithme de couplage dans le code fluide	108
7.3	Implémentation de l'algorithme de couplage dans le code solide	109
7.4	Exemple de mise en données de l'algorithme de couplage	110
7.5	Résultat du calcul mécanique couplé	111
7.6	Pression le long de l'axe principal en début et fin de calcul	112
7.7	Déplacement vertical le long du tube, pour différents instants	112
7.8	Évolution du déplacement vertical en trois points du tube	113
8.1	Solution de référence (pour $\Delta_t = 1/1000^e s$)	116
8.2	Solution de référence (pour $\Delta_t = 1/1000^e s$)	117
8.3	Emplacements des nœuds suivis dans les calculs de ce chapitre	118

8.4	Flux reçus par la méthode Dirichlet→Neumann	118
8.5	Flux reçus par la méthode Dirichlet→Neumann	119
8.6	Comparaison des flux prédits et reçus dans la méthode Dirichlet→Neumann.	120
8.7	Comparaison des flux prédits et reçus dans la méthode Dirichlet→Neumann.	121
8.8	Ajustement de la séquence de couplage.	123
8.9	Flux de chaleur pariétal à $t = 6s$	123
8.10	Comparaison des prédicteurs constant, linéaire et quadratique	124
8.11	Comparaison des erreurs absolues pour les trois prédicteurs	125
8.12	Résultats obtenus par la méthode Dirichlet→Neumann avec corrections.	127
8.13	Évolution de l'erreur au cours des itérations	128
8.14	Localisation de l'erreur	128
8.15	Nombre cumulé d'itérations, pour les méthodes Dirichlet→Neumann avec corrections	129
8.16	Comparaison des résultats obtenus pour des méthodes avec ou sans correction	130
8.17	Résultats obtenus par la méthode corrigée, avec un critère de convergence relatif local	131
8.18	Résultats de la méthode du pas de temps adaptatif	133
8.19	Comparaison des coûts des méthodes à pas de temps fixe et adaptatif	134
8.20	Nombre d'itérations par incrément pour chacun des deux jeux de paramètres de la méthode de pas de temps adaptatif	135
8.21	Relation flux-température en différents nœuds de la paroi couplée	136
8.22	Résultats obtenus par la méthode Dirichlet→Fourier pour différentes fréquences de couplage.	137
8.23	Coût de la méthode Dirichlet→Fourier pour différentes valeurs de α	138
8.24	Coût de la méthode Dirichlet→Fourier pour un coefficient α local.	139
9.1	Différentes interactions, dans le cas du couplage aéro-thermo-mécanique	141
9.2	Couplage simplifié à trois codes	142
9.3	Algorithme de couplage simplifié à 3 codes	144
9.4	Mise en place de l'algorithme de couplage dans le code fluide	145
9.5	Mise en place de l'algorithme de couplage dans le code thermique	145
9.6	Mise en place de l'algorithme de couplage dans le code mécanique	145
9.7	Solution du calcul couplé à trois codes	146
9.8	Déplacements le long de l'axe principal du tube	147
9.9	Comparaison des champs de température appliqués dans le calcul couplé mécanique et à trois codes	148
9.10	Température le long de l'axe principal du tube à $t = 20s$	148
9.11	Algorithme de couplage à 3 codes	149
9.12	Algorithme de couplage à 3 codes : première solution	150
9.13	Algorithme de couplage à 3 codes : deuxième solution	150
B.1	Déformations thermiques en fonction de la température	165
B.2	Module d'Young en fonction de la température	166
C.1	Exemple de diagramme UML	169
C.2	Structure du plugin de couplage	171
C.3	Objets <i>conditions au bord</i> et <i>paramètres</i> couplés	172

C.4 Familles d'objets imposant les DDL	173
C.5 Présentation de quelques algorithmes de couplage développés	174

Remerciements

Après trois ans et quelques broutilles, voici venu le temps de m'atteler aux remerciements. Ils vont bien entendu en premier lieu à Yvon Maday, qui a eu la gentillesse d'accepter la direction de ma thèse. Nos rencontres ont à chaque fois été riches en idées pour m'occuper quelques mois !

Ils vont ensuite à Thierry Coupez et Christian Rey, qui ont eu la gentillesse et la patience de relire le manuscrit. Cela a été un plaisir de répondre à leurs remarques et critiques lors de la soutenance.

Laure-Sophie Ballester a apporté son point de vue, d'abord par sa relecture du manuscrit, puis à la soutenance. Je la remercie vivement pour les précisions qu'elle m'a apportées concernant Ariane, et pour ses remarques pertinentes sur la physique du couplage.

David Dureisseix et Serge Piperno ont bien voulu compléter le jury ; qu'ils en soient remerciés. Leur œil de numéricien, conscients des difficultés que l'on peut rencontrer lors du développement logiciel, a été apprécié.

François-Xavier Roux, chez qui j'ai fait mes premières armes en stage, a un bureau qui est resté grand ouvert tout au long de ma thèse. Tu n'as jamais été avare ni de temps, ni d'idées, ni de conseils ; merci pour tout cela !

Arjen Roos, encadrant officiel, et Frédéric Feyel, encadrant officieux, m'ont formé au jour le jour ; leur recette : prendre un stagiaire qui traîne au frais pas loin d'un gros cluster, ajouter une bonne dose de mécanicien, une lampée de programmeur, une livre de débogueur, assaisonner d'un soupçon de calculateur et d'une pincée de mailleur. Compléter à ras bord de café. Laisser mûrir le tout dans un grand bureau bien chauffé. Servir accompagné d'une présentation et d'un petit vin blanc doux.

Les deux comparses fluidistes, Marc Errera et Gilles Chaineray, sont venus apporter leur sauce à tout cela. Ils m'ont plus d'une fois dépanné avec mes calculs et développements fluides.

J'ai passé mes trois années à l'Onera dans le bâtiment E4. Que tous ses membres en soient remerciés : les chercheurs, les administratifs, le labo, les stagiaires, thésards, postdoc et autres intermittents. Pour être certain de ne pas en omettre, je vais éviter de les citer nommément¹. Certains méritent quand même une attention particulière : les caféinomanes Yann, Myriam et Eva, ainsi que les joueurs de belote JeF, Yannis, Norbert, FX, Vincent, Anaïs, Lionel.

Juliette, François-Xavier, Xavier, Ibrahima, Elsie, Pierre, mes compatriotes numériciens du CHP sont certainement conscients que j'ai beaucoup apprécié repasser du temps avec eux, à l'occasion d'une petite pause. Mais au cas où, je vous le rappelle ici !

¹Mais, oui, toi là, qui lit ce petit mot maintenant, rassure-toi, je ne t'avais pas oublié !

Je n'oublierai pas non plus les gens du Centre des Matériaux, qui ont eu la gentillesse de m'accueillir en sortie de thèse. Votre bonne humeur et votre soutien sur la fin ont été des plus appréciés.

Depuis mon arrivée à Châtillon, j'ai aussi eu le loisir de fréquenter pas mal d'équipes de volley-ball : celle de l'Onera, les loisirs, loisirs-compet, herb-volley et départementaux de Châtillon. Les connaissant, ces joueuses et joueurs apprécieront plus de bonnes passes que des remerciements. C'est donc tout ce que je vous souhaite, jusqu'à la cinquième génération.

Et bien entendu, mes remerciements les plus affectueux vont à papa, maman et sœurette qui ne sont probablement pas tout à fait étrangers à cette affaire . . .

Introduction

Introduction

Dans le cadre de campagnes de simulations numériques, il est devenu de plus en plus fréquent de chercher à représenter et modéliser des phénomènes où fluide et structure interagissent. C'est le cas dans des domaines aussi variés que la simulation du flottement d'ailes d'avions (Fig. 1), de résistance de ponts au vent, de suivi de sillages de navires, du comportement d'un véhicule sur surfaces mouillées, ou dans des cas d'écoulements internes, tels que dans les bioconduits (écoulement du sang dans les vaisseaux sanguins ou d'air dans les voies respiratoires, cf. Fig. 2).

Dans le domaine aéronautique et aérospatial, domaine principal de recherche à l'ONERA, les applications sont toutes aussi diverses : elles incluent les simulations aéro-mécaniques, interactions entre un gaz (le plus souvent l'air) et une structure (aile d'avion, pale d'hélicoptère, etc...), mais aussi les simulations aéro-thermiques, où l'interaction étudiée implique l'évolution thermique d'un fluide et d'une structure. Il s'agira par exemple de problèmes de refroidissement de chambres de combustion par effusion (Fig. 3a), c'est-à-dire par injection de gaz au travers d'une paroi perforée, ou de refroidissement d'aubes de turbines multiperforées par injection de gaz froids (Fig. 3b).

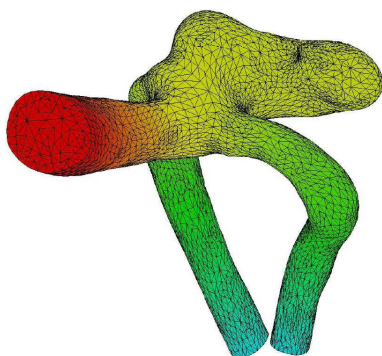


FIG. 2 – Exemple de couplage mécanique : champ de pression artérielle autour d'un anévrisme [26].

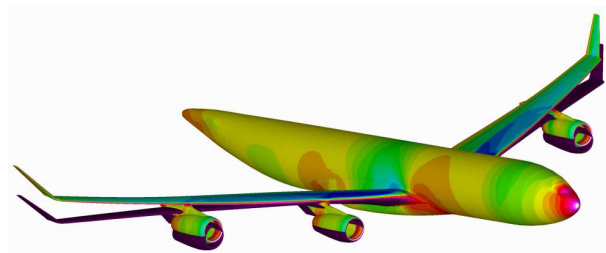


FIG. 1 – Simulation d'une configuration voilure-fuselage-nacelle. L'image montre le résultat d'un calcul couplé statique : champ de pression à la paroi sur la forme de vol prédite par le calcul ; la configuration non-déformée apparaît en noir. Image extraite de [36].

Ces nombreux exemples ont en commun qu'une simulation découplée de chacune des physiques est inadaptée à la situation. Les interactions entre fluide et structure sont au cœur du phénomène étudié, et il est donc impensable de ne pas les mettre au cœur de la simulation. Industriellement, la méthode la plus souvent employée consiste à coupler manuellement les domaines : un premier calcul, du fluide par exemple, est réalisé sur la configuration initiale de la structure, et les résultats sont utilisés comme conditions aux bords pour le solide. Ceci permet de réaliser un calcul solide avec des conditions relativement réalistes. Cependant, le calcul initial du fluide n'est réalisé que sur configuration initiale, et peut ne pas être représentatif sur la configuration déformée.

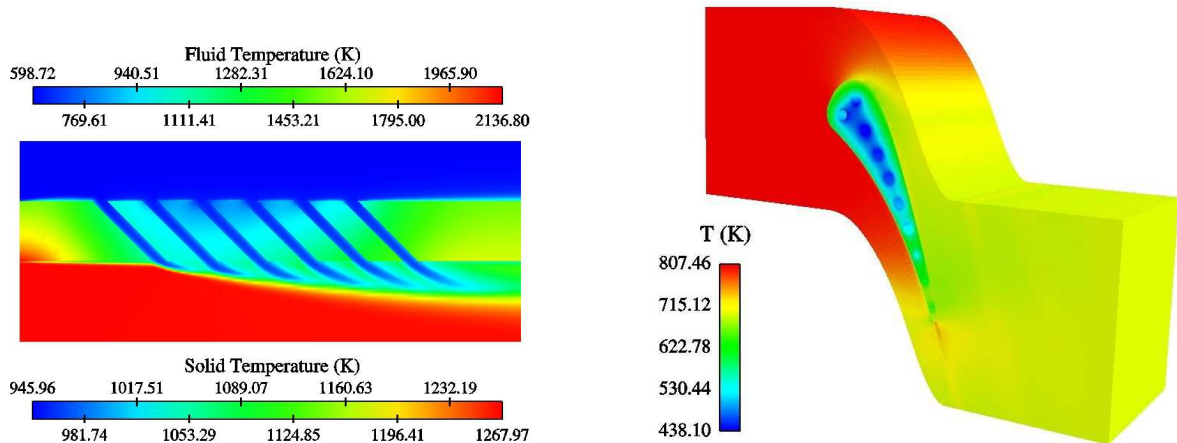


FIG. 3 – Exemples de couplages thermiques : refroidissement d’une paroi de chambre de combustion par effusion de gaz et champ température d’une aube de turbine refroidie par injection de gaz [13].



FIG. 4 – Moteur cryotechnique Vulcain 2 au banc d’essai ; image tirée de [72].

Nous disposons aussi d’ordres de grandeurs des déformations subies, permettant de rester dans un domaine réaliste.

L’accident survenu lors du vol 157 de la fusée Ariane 5 le 11 décembre 2002 nous a fourni un autre exemple de situation nécessitant une approche couplée. La commission d’enquête a mis en évidence la défaillance du système de refroidissement des tubes constituant le divergent du moteur Vulcain 2 (Fig. 4) : les déformations subies par les tubes modifient à la fois l’écoulement des gaz de refroidissement et des gaz d’échappements chauds, conduisant à leur surchauffe puis à leur rupture. Les études sur le matériau constituant les tubes, réalisées à la suite de l’accident [63], ont émis l’hypothèse que des simulations découplées des phénomènes de déformations n’arrivent pas à représenter complètement les conditions vues par les tubes. Une solution envisagée a été d’ajouter une composante couplée aux fluides à ces simulations.

C’est dans ce contexte qu’a été démarrée cette étude, afin de pouvoir proposer à l’avenir des possibilités de résolution couplée de ce genre de problèmes. Son objectif principal est à ce titre de démontrer la faisabilité de telles simulations couplées, à la fois thermiquement et mécaniquement. L’expérience acquise sur l’étude des déformations de la tuyère nous a permis de partir d’une base concrète et surtout réaliste concernant le matériau utilisé.

De plus, ce problème est représentatif de ceux rencontrés dans le domaine aérospatial. Il inclut à la fois un couplage thermique et un couplage mécanique entre fluide et structure. De plus, les (grandes) déformations subies par le matériau sont générées par plusieurs phénomènes : déformations élasto-visco-plastiques et dilatation thermique.

Deux écoles existent principalement pour la résolution de tels problèmes multi-physiques : celle du couplage monolithique, consistant à développer un code unique de résolution du problème couplé, et celle du couplage partitionné, consistant au contraire à utiliser deux codes préexistants pour la résolution séparée de chacune des physiques et incorporant un algorithme de couplage permettant d'assurer l'équilibre à l'interface.

Ces méthodes de couplage partitionné remontent au début des années 1980, et ont montré depuis leur faisabilité dans de nombreuses situations mécaniques : pour des applications aéro-élastiques [55, 61, 59, 60], navales [25], automobiles [43], en ingénierie civile [37] ; mais aussi pour des situations thermiques [13]. Sous leur forme originale de couplage dit *faible*, ces méthodes ont aussi montré leurs limites ; de nombreuses variantes ont donc vu le jour, afin d'appréhender les problèmes d'instabilités rencontrés [46, 12, 24]. Dans toutes ces applications, l'algorithme retenu est un compromis entre rapidité, efficacité, précision et faisabilité.

Par ailleurs, ces questions multi-physiques sont très semblables aux questions de décomposition de domaines : le calcul par décomposition de domaines peut se voir comme le couplage de deux physiques identiques. Cette analogie apporte donc au calcul multiphysique toute la richesse des méthodes de décomposition de domaines, actuellement très développées grâce à l'engouement pour le calcul parallèle.

C'est la solution partitionnée qui a été retenue. Le problème de couplage fluide-structure se ramène ainsi à une question de couplage entre le code de mécanique des fluides MSD et le code de mécanique du solide ZéBuLoN. Plusieurs départements de l'ONERA sont donc impliqués : le département DSNA (Simulations Numériques des écoulements et Aéroacoustique) qui développe MSD, le département DMSM (Matériaux et Structures Métalliques), qui développe le code ZéBuLoN conjointement avec l'École des Mines de Paris et Northwest Numerics, et le département DTIM (Traitement de l'Information et Modélisation), spécialiste du développement d'algorithmes parallèles et de résolution par sous-domaines. La collaboration entre ces départements est concrétisée au travers d'un plan de recherche fédérateur, dénommé CoCoMAT (couplage de codes en mécanique et aérothermique).

Nous commencerons cette présentation par un rappel des différents modèles, équations et méthodes de résolution en mécanique des fluides (section 1.1) et en mécanique des structures (section 1.2). Nous en profiterons pour présenter les deux logiciels qui ont été utilisés : MSD pour le fluide et ZéBuLoN pour le solide. Nous terminerons ces rappels par une présentation des différents couplages qui seront pris en compte : le couplage aéro-mécanique (couplant le fluide à la mécanique du solide), le couplage aéro-thermique (couplant le fluide à la thermique du solide) et le couplage thermo-mécanique (couplant la mécanique et la thermique du solide).

Au deuxième chapitre, nous présenterons l'ensemble des méthodes de couplage existantes, en nous concentrant principalement sur le couplage partitionné et ses nombreux algorithmes de résolution. Les concepts de couplage *faible* et *fort* seront précisés, montrant leurs avantages et inconvénients respectifs. Nous terminerons cette revue bibliographique par une incursion via les méthodes algébriques d'inversion de matrices par blocs ainsi que les méthodes de décomposition de domaines. Ces deux domaines sont en effet très étroitement liés au calculs couplés, et offrent un nouveau regard sur les méthodes partitionnées, y apportant ainsi des

méthodes innovantes.

Les questions qu'il faut traiter pour réaliser un couplage de codes sont nombreuses. Elles englobent entre autres les problèmes de discrétisation temporelles et spatiales de chaque domaine, et donc de raccord et réinterpolation entre ces discrétisations ; d'algorithmie permettant le couplage ; de déformations de maillage pour le fluide ; de conditions de raccord à l'interface entre les domaines.

Le chapitre 3 nous permettra de réaliser un bilan des besoins nécessaires à la réalisation de calculs couplés. La présentation de ces stratégies de couplage, dans le cadre d'un développement *objet* nous conduira au concept d'assemblage d'algorithme de couplage, taillé sur mesure pour le problème traité. Les développements logiciels nécessaires au couplage mécanique seront ensuite présentés : il s'agit de l'ajout d'un module de déformation de maillage et d'une condition aux bords couplée en pression. Nous mettrons finalement en application ces concepts de couplage sur trois cas. Le premier, un cas simple de couplage structure-structure, permettra d'illustrer justement les liens entre méthodes de couplage et méthodes multi-domaines. Les deux autres, un cas de couplage thermo-mécanique et un cas de couplage à trois codes, montreront la souplesse possible par l'assemblage de *briques* de couplage, et permettront de valider une partie des développements réalisés.

La troisième et dernière partie de cette thèse sera consacrée à l'étude des tubes de refroidissement du divergent de Vulcain 2. Le problème complet est très fortement couplé, tant au niveau mécanique que thermique. Nous l'aborderons donc par étapes successives : tout d'abord par un couplage uniquement mécanique. Le comportement du matériau est fortement non-linéaire, surtout aux très hautes températures auxquelles cette tuyère est soumise ; la prise en compte de ces non-linéarités nécessite donc un soin particulier qui sera détaillé dans cette partie.

Puis nous nous intéresserons à la simulation thermiquement couplée de l'écoulement dans un tube. Ce problème thermique se révèle très fortement couplé, et l'utilisation des méthodes simples de couplage y montre ses limites. Il nécessitera donc la mise en place de toute une panoplie de stratégies, qui nous permettront au final de réaliser cette simulation en un temps raisonnable.

Nous nous attacherons enfin au couplage aéro-thermo-mécanique, afin de réaliser la simulation du couplage complet du tube. Une première solution, bien adaptée à la situation du tube sera présentée, permettant de réaliser la simulation complète d'un vol. Nous terminerons en présentant quelques solutions théoriques et plus générales, qui pourront être envisagées par la suite.

Première partie

État de l'art

Chapitre 1

Mécanique des fluides et des structures

Nous faisons ici une présentation des modèles, équations et méthodes utilisés en mécanique des fluides et des solides, présentation qui est volontairement brève. Cela reflète une des caractéristiques du couplage partitionné, qui est de permettre un couplage entre différentes physiques sans pour autant être fortement impliqué au cœur des équations de chacune d'entre elles.

1.1 Mécanique des fluides

La simulation numérique des écoulements fluides est basée sur les équations de Navier-Stokes. Bien que celles-ci remontent à de nombreuses années, elles posent encore de nombreux problèmes, tant théoriques que numériques. Nous en présentons brièvement quelques concepts dans les pages qui suivent. Les livres de Candel [11] ou de Huerre [42] peuvent être consultés pour approfondir ce sujet.

Le logiciel MSD, développé à l'Onera, permet la simulation numérique de ces écoulements. C'est un logiciel très complet, et nous n'en utiliserons dans cette thèse qu'une petite partie. Ses capacités de calcul d'écoulements multi-espèces ou réactifs ne seront par exemple pas du tout abordées. Nous renvoyons à [50] ainsi qu'aux références qui y sont indiquées pour une description approfondie de ce logiciel.

1.1.1 Équations de Navier-Stokes

L'évolution d'un fluide est régie par les équations de Navier-Stokes. Celles-ci traduisent la conservation de trois quantités : masse, quantité de mouvement et énergie :

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{v}) = 0 \quad (1.1a)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \operatorname{div}(\rho \mathbf{v} \otimes \mathbf{v}) = \rho \mathbf{g} + \operatorname{div} \boldsymbol{\sigma}_F \quad (1.1b)$$

$$\frac{\partial (\rho E)}{\partial t} + \operatorname{div}(\rho \mathbf{v} E) = -\operatorname{div}(p \mathbf{v}) + \operatorname{div}(\boldsymbol{\sigma}_F \cdot \mathbf{v}) + \rho \mathbf{g} \cdot \mathbf{v} - \operatorname{div}(\mathbf{q}) \quad (1.1c)$$

où¹ $\rho(t, \mathbf{x})$ est la densité du gaz, $p(t, \mathbf{x})$ sa pression, $\mathbf{v}(t, \mathbf{x})$ sa vitesse, $\mathbf{g}(t, \mathbf{x})$ une force volumique extérieure (par exemple la gravité), E est l'énergie totale par unité de masse, et $\boldsymbol{\sigma}_F(t, \mathbf{x})$

¹L'ensemble des notations est rassemblé dans l'annexe A.

le tenseur de contraintes du fluide donné par $\underline{\sigma}_F = -p\tilde{\text{Id}} + \underline{\tau}$. Dans cette dernière relation, $\underline{\tau}(t, \mathbf{x}) = \mu(\text{grad } \mathbf{v} + \text{grad } \mathbf{v}^T)$ est le tenseur de contraintes visqueuses, et μ la viscosité du fluide, dépendante de la température et de la densité. Par ailleurs $\mathbf{v} \otimes \mathbf{v}$ désigne le tenseur dont la composante (i, j) est $v_i v_j$.

Suivant les caractéristiques du fluide, une loi d'état sera choisie, précisant la relation entre la pression, la densité et la température. Dans le cas d'un gaz parfait, ce sera par exemple la relation

$$p = r\rho T$$

r étant la constante de Mariotte du gaz.

Ces équations sont complétées de conditions initiales et de conditions aux bords : conditions d'entrée, de sortie ou de paroi suivant la situation.

1.1.2 Discrétisation spatiale par volumes finis

Les équations de Navier-Stokes peuvent être numériquement abordées de diverses manières : différences finies, volumes finis ou éléments finis par exemple.

Le logiciel MSD utilise pour la discrétisation spatiale une méthode de volumes finis, sur des maillages réglés. Son principe est de découper le domaine de calcul en éléments de volume. Les quantités représentatives de l'écoulement sont ensuite discrétisées sur ce maillage de manière "cell-centered", c'est-à-dire que chaque élément de volume se voit attribuer une valeur constante moyenne des quantités. Les équations de conservation (1.1) sont ensuite exprimées sur chacun de ces volumes élémentaires.

Dans les situations réalistes, ces équations sont beaucoup trop coûteuses à résoudre sous cette forme : les discrétisations nécessaires pour représenter l'ensemble de l'écoulement demandent une extrême finesse. Ainsi, de nombreux modèles simplifiés existent. Citons par exemple des modèles de turbulence, décomposant les inconnues (ρ, \mathbf{v} , etc. . .) en la somme d'une variable moyenne et de sa turbulence, ou des modèles de paroi permettant d'éviter une discrétisation trop fine à proximité de la paroi tout en captant d'éventuels décollements de l'écoulement.

Le calcul des flux en voisinage de paroi est aussi beaucoup plus complexe qu'une simple différence finie entre les températures de première maille et de maille fictive. En raison des modèles de turbulence, de viscosité, de couches limites, etc. . ., des modèles de calcul de flux de paroi sont utilisés. Outre les grandeurs mentionnées ci-dessus, ces modèles font aussi intervenir les nombres de Reynolds et Prandtl, la taille des mailles, l'épaisseur des couches limites. Selon la turbulence et les décollements attendus, différents modèles sont disponibles. Plus de détails concernant les lois de paroi peuvent être trouvés dans [17, 13].

Rappelons un point important concernant les discrétisations par volumes finis : les quantités pariétales sont décrites par éléments : les champs discrets sont ainsi constants sur chacune des faces des cellules du maillage.

Pour finir, mentionnons que le logiciel MSD tend à être remplacé par son successeur CÈDRE. Il s'agit toujours d'un solveur volumes finis, mais capable de gérer des maillages non structurés.

1.1.3 Discrétisation temporelle et états stationnaires

La discrétisation temporelle dans MSD est totalement indépendante de la discrétisation spatiale. Afin d'obtenir une méthode discrète numériquement stable, elle doit néanmoins respecter la condition de Courant-Friedrichs-Lewy (CFL). Cette condition impose une borne

maximale sur le pas de temps en fonction de la taille des mailles :

$$\Delta t < c\Delta x \quad (1.2)$$

c étant une constante dépendant du problème traité. Dans le cas de la mécanique des fluides, cette condition est généralement interprétée de la manière suivante : une particule du fluide ne doit pas parcourir plus d'une cellule pendant un pas de temps, la constante c étant alors égale à la vitesse maximale du fluide.

Nous nous intéresserons par la suite à des écoulements *stationnaires*, c'est-à-dire n'évoluant pas au cours du temps. Le logiciel MSD traite ces problèmes de la même manière que pour un problème instationnaire : partant de la condition initiale, l'état est avancé en temps jusqu'à sa stabilisation. Cependant, afin d'accélérer la recherche de cet état stationnaire, le logiciel utilise des pas de temps locaux, calculés en fonction des tailles de mailles et des vitesses locales du fluide. Deux mailles adjacentes peuvent donc évoluer avec des pas de temps différents. Les états intermédiaires n'auront pas de sens physique, mais l'état stationnaire obtenu *in fine* sera bien une solution physique.

Suivant les situations, cette méthode permet de gagner un facteur 10, voire plus, sur le temps de simulation, par rapport à des simulations à pas de temps fixé. C'est donc une méthode extrêmement utile si l'on ne cherche à obtenir que des états stationnaires.

1.2 Mécanique des solides déformables

Les équations de la mécanique du solide et des structures reposent sur la mécanique des milieux continus [27, 64]. Elles s'écrivent en formulation Lagrangienne, c'est-à-dire dans un repère attaché au solide, et suivant le déplacements de ses particules. L'équation fondamentale est déduite du principe de conservation de la quantité de mouvement et se traduit par l'équation

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} - \operatorname{div} \boldsymbol{\sigma} = \rho \mathbf{f} \quad (1.3)$$

où ρ est la densité du matériau², \mathbf{u} le champ de déplacements, $\boldsymbol{\sigma}$ le tenseur des contraintes d'ordre 2, et \mathbf{f} une force volumique externe, par exemple la gravité.

Nous nous intéresserons par la suite à des problèmes *quasi-statiques*, c'est-à-dire des problèmes mécaniques pour lesquels les effets d'inertie ou d'accélération du solide sont négligeables. L'équation précédente se réduit donc à

$$-\operatorname{div} \boldsymbol{\sigma} = \rho \mathbf{f} \quad (1.4)$$

L'expression de $\boldsymbol{\sigma}$ en terme des autres paramètres et variables du calcul se fait au travers des *lois de comportement* [7, 44]. Celles-ci feront en particulier intervenir le tenseur des déformations

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} (\operatorname{grad} \mathbf{u} + (\operatorname{grad} \mathbf{u})^T + (\operatorname{grad} \mathbf{u})^T (\operatorname{grad} \mathbf{u})) \quad (1.5)$$

Dans l'hypothèse des petites déformations, le terme quadratique est négligeable et le tenseur des déformations se réduit à

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} (\operatorname{grad} \mathbf{u} + (\operatorname{grad} \mathbf{u})^T) \quad (1.6)$$

²Notons que contrairement au cas de la mécanique des fluides, la densité est ici un paramètre, et non une inconnue du problème.

Cette relation contraintes–déformations, aussi appelée loi de comportement, est dépendante de nombreux facteurs : nature du matériau, nature et amplitude des déformations, paramètres extérieurs, etc. . . Elle pourra être linéaire, par exemple dans le cas de petites déformations élastiques, ou non-linéaire. Dans ce dernier cas, on distinguera principalement deux sources de non-linéarités :

- géométriques, dans le cas de grandes déformations ;
- matériaux, pour des déformations irréversibles plastiques par exemple.

1.2.1 Lois de comportement

Élasticité linéaire isotrope

Dans le cas de l'élasticité linéaire isotrope, la relation contraintes–déformations dépend simplement de deux paramètres, le module d'Young E et le coefficient de Poisson ν :

$$\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{12} \end{pmatrix} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1+\nu}{E} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1+\nu}{E} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1+\nu}{E} \end{bmatrix} \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{pmatrix} \quad (1.7)$$

Cette relation, nommée loi de Hooke, peut aussi se résumer sous la forme :

$$\underline{\underline{\varepsilon}} = \underline{\underline{\mathcal{S}}} \underline{\underline{\sigma}} = \frac{1+\nu}{E} \underline{\underline{\sigma}} - \frac{\nu}{E} \text{Tr}(\underline{\underline{\sigma}}) \underline{\underline{\text{Id}}} \quad (1.8)$$

Le tenseur $\underline{\underline{\mathcal{S}}}$ d'ordre 4 apparaissant dans cette relation est le tenseur de souplesse ; son inverse, noté $\underline{\underline{\mathcal{K}}}$, est appelé tenseur de rigidité.

Élasto-visco-plasticité

Afin de rendre compte des nombreux effets mécaniques (dilatation, viscosité, déformations plastiques, dislocations, endommagement, etc. . .), la loi de comportement présentée ci-dessus peut être progressivement enrichie. Les modèles de thermo-élasto-visco-plasticité [44, chap. 6] tiennent compte de l'ensemble de ces phénomènes. Dans cette situation, les déformations ont diverses origines : élastiques, thermiques et plastiques, et l'on décompose le tenseur total des déformations $\underline{\underline{\varepsilon}}^{tot}$ suivant ces origines :

$$\underline{\underline{\varepsilon}}^{tot} = \underline{\underline{\varepsilon}}^e + \underline{\underline{\varepsilon}}^{th} + \underline{\underline{\varepsilon}}^p$$

Le tenseur des déformations élastiques $\underline{\underline{\varepsilon}}^e$ vérifie l'équation (1.8).

Dans le cas de déformations thermique isotropes, le tenseur des déformations thermiques $\underline{\underline{\varepsilon}}^{th}$ est donné par :

$$\underline{\underline{\varepsilon}}^{th} = \alpha(T - T_0) \underline{\underline{\text{Id}}} \quad (1.9)$$

où α est le coefficient de dilatation thermique, T_0 une température de référence, et $\underline{\underline{\text{Id}}}$ désigne le tenseur identité.

Enfin, le tenseur des déformations visco-plastiques $\underline{\underline{\varepsilon}}^{vp}$ est donné par exemple par l'intégration d'une loi de type Chaboche :

$$\|\underline{\underline{\varepsilon}}^{vp}\| = \left\langle \frac{f}{K} \right\rangle^n = \left\langle \frac{\|\underline{\underline{\sigma}} - \underline{\underline{\mathbf{X}}}\| - R - k}{K} \right\rangle^n \quad (1.10)$$

dans laquelle K et n sont des coefficients de viscosité de Norton, \underline{X} le tenseur d'écouissage cinématique, R l'écouissage isotrope, et k un seuil initial d'écouissage. Les crochets $\langle x \rangle$ désignent la partie positive de x .

1.2.2 Conditions aux limites

Ces équations sont complétées de conditions initiales et de conditions au bord. Ces dernières sont de types extrêmement variés : déplacements imposés, pression imposée, cisaillement, torsion, etc . . . Nous renvoyons au manuel de l'utilisateur de ZéBuLoN [70] pour une description exhaustive des types de conditions au bord disponibles dans ce logiciel.

1.2.3 Discrétisation et résolution

La méthode de discrétisation par éléments finis est celle la plus couramment utilisée pour la résolution des problèmes de mécanique du solide. Nous renvoyons aux livres de Ern [18] pour une présentation générale de la méthode des éléments finis, ainsi qu'aux livres de Zienkiewicz [71] et de Belytschko [5] pour des présentations orientées plus spécifiquement vers les applications de la mécanique des milieux continus et des problèmes non-linéaires.

À l'exception des matériaux élastiques en petites déformations, le système d'équations obtenu par la discrétisation par éléments finis est non-linéaire. Il est dans ce cas résolu par une méthode itérative, de type Newton. Ainsi, à partir d'une situation d'équilibre, les équations sont écrites par perturbation, de manière *incrémentale*.

Dans les situations quasi-statiques et élastiques, la variable *temps* ne joue aucun rôle. La discrétisation temporelle pourra donc y être choisie de manière arbitraire, éventuellement de manière à bien représenter les évolutions de conditions aux limites.

Par contre, dans les situations quasi-statiques visco-élastiques ou visco-plastiques, le temps intervient dans la loi de comportement. C'est le cas par exemple des déformations par *fluage*³. Contrairement au fluide, la discrétisation temporelle dans le solide ne sera pas limitée par une condition de stabilité du type CFL, le schéma d'intégration temporelle étant implicite⁴. Par contre elle sera limitée par les non-linéarités vues par la structure au cours de son évolution. Des méthodes de détermination automatique de cette discrétisation temporelle existent : la discrétisation temporelle est ainsi déterminée au cours du calcul, en fonction de la variation de certaines variables internes (déformation plastique cumulée par exemple).

1.2.4 ZéBuLoN, code *objet*

ZéBuLoN [69] est un logiciel de résolution de problèmes de mécanique non-linéaire des matériaux par la méthode des éléments finis.

Il est développé conjointement par l'École des Mines de Paris, Northwest Numerics et l'ONERA. Écrit à l'origine en Fortran, il a, depuis les années 90, été entièrement repensé de manière objet et a été réécrit en C++.

³Le fluage est le phénomène de déformation irréversible dû à l'application d'une pression pendant une longue période ; un exemple de la vie courante est la déformation progressive d'une étagère chargée de livres pendant plusieurs années.

⁴Pour les applications dynamiques, ZéBuLoN propose aussi un schéma explicite d'intégration. Pour les applications quasi-statiques envisagées ici, le schéma implicite est cependant préférable.

Les types d'éléments finis utilisés habituellement sont, dans le plan, les triangles et les quadrangles linéaires ou quadratiques, et, dans l'espace, les tétraèdres, les prismes et les hexaèdres linéaires ou quadratiques. Certaines applications nécessitent parfois d'autres éléments moins courants, tels que les pyramides ou les ressorts, qui sont aussi disponibles. L'intégration numérique des formulations variationnelles sur ce type d'éléments est faite au travers d'une règle de quadrature : elle peut être complète (par exemple 3 points d'intégrations pour le triangle linéaire) ou réduite (1 seul point d'intégration pour ce même triangle). L'intégration réduite, bien que n'intégrant pas exactement les fonctions de forme, évite dans certaines situations le phénomène de verrouillage⁵, et est surtout plus rapide que l'intégration complète. Le gain en terme de temps calcul et de mémoire peut être considérable sur les problèmes à grand nombre de variables internes.

L'architecture *objet* de ZéBuLoN présente des avantages à plusieurs niveaux : tout d'abord, elle permet la séparation claire entre objets de base, qui forment le noyau du solveur, et les développements additionnels, spécifiques à quelques problèmes. Elle permet aussi de développer de manière non intrusive de nouvelles méthodes, telles que celles de couplage dont il est question dans cette thèse.

La conception objet du logiciel est présente à tous les niveaux : par exemple, les types d'éléments disponibles de manière standard sont chacun un objet, dérivant d'une même classe de base "élément". L'ajout d'un nouveau type d'élément, ne nécessite que le développement d'un nouvel objet dérivant de cette classe. Cet objet surchargera donc les méthodes décrivant sa géométrie, ses fonctions de forme, l'intégration numérique sur l'élément, etc. . . .

La méthode de résolution est elle aussi conçue de manière objet, et est adaptable à chaque problème. Elle implique le choix de nombreuses "briques" de résolution : la méthode de gestion, d'assemblage et d'inversion de la matrice de rigidité, la méthode de résolution du problème non-linéaire, la discrétisation temporelle en sont trois exemples parmi les plus significatifs. Nous renvoyons à [8] pour une description plus détaillée des principes de développement objets appliqués à la conception d'un code de calcul, ainsi qu'à la documentation de ZéBuLoN [70] pour une description complète des capacités du logiciel.

1.3 Thermique du solide

Les problèmes de diffusion et de thermique étant analogues aux problèmes mécaniques, ils sont aussi traitables par ZéBuLoN.

Nous nous restreignons ici au phénomène de diffusion de la chaleur dans un solide. Celle-ci est régie par la loi de conduction de Fourier :

$$\rho C \frac{\partial T}{\partial t} - \text{div}(\kappa \text{grad } T) = f^{th} \quad (1.11)$$

où $T(t, x)$ est la température du solide, $\rho(t, x)$ sa densité, $C(T)$ la capacité massique (aussi appelée chaleur spécifique), $\kappa(T)$ le coefficient de conductivité (ou matrice de conductivité, dans le cas d'un matériau non isotrope) et $f^{th}(t, x)$ un éventuel terme source.

Cette équation est complétée d'une condition initiale $T(t = 0, x)$ et de conditions au bord du domaine. Ces dernières peuvent être de diverses formes, notamment :

- Dirichlet : la température est imposée ;

⁵En anglais "locking"

- Neumann : le flux de chaleur traversant la surface est imposé ;
- Fourier : une relation affine entre flux et température est imposée ;
- radiative : le flux de chaleur est fonction affine de la puissance quatrième de la température.

Dans certaines situations, dites *statiques*, le terme d’inertie peut être négligé. On considère alors que la diffusion de la chaleur est instantanée. L’équation (1.11) se simplifie dans ce cas sous sa forme quasi-statique :

$$-\operatorname{div}(\kappa \operatorname{grad} T) = f^{th} \quad (1.12)$$

Tout comme la mécanique, la thermique du solide est résolue par la méthode des éléments finis. Des méthodes similaires peuvent donc s’y appliquer, et le logiciel ZéBuLoN est aussi utilisé pour ce type de simulations.

L’équation de la chaleur est plus aisée à résoudre que celles de la mécanique, et ce pour plusieurs raisons : tout d’abord, la température est la seule inconnue, au lieu des trois déplacements. Ensuite la conductivité, qui joue ici le rôle de la rigidité dans le cas mécanique, sera dans nos situations considérée isotrope, c’est-à-dire se résumera à un seul coefficient scalaire. Nous prendrons par contre en compte une éventuelle dépendance de la conductivité et de la capacité calorifique à la température. Cette dépendance rend les systèmes (1.11) et (1.12) non-linéaires.

1.4 Problèmes couplés

Le terme *problème couplé* peut englober de nombreuses situations : un problème de mécanique du solide peut par exemple être vu comme trois problèmes de déformations suivant chacune des directions, couplées au travers du coefficient de Poisson. Les problèmes auxquels nous nous attacherons ici sont cependant bien ciblés : il s’agit de problèmes de thermo-mécanique (couplage entre la thermique d’un solide et de sa mécanique), d’aéro-thermique (couplage entre un fluide et la thermique d’un solide), d’aéro-mécanique (couplage entre un fluide et la mécanique d’un solide), et enfin de l’aéro-thermo-mécanique, regroupant l’ensemble des trois précédents.

Il est aussi intéressant de faire le rapprochement entre les questions de couplage et de décomposition de domaines, cette dernière pouvant être vue comme le couplage de deux domaines modélisés de manière identique.

L’ensemble des problèmes couplés que nous étudierons partagent un point commun : leur écriture peut se faire de manière scindée, assez naturellement. Regardons par exemple le cas du couplage aéro-mécanique.

Chacune des physiques est régie par ses propres équations. Par exemple, dans le cas d’un couplage fluide-structure, ce seront les équations de Navier-Stokes (éq. 1.1) pour le fluide, et celles de la mécanique des milieux continus (éq. 1.4) pour le solide. Nous ne nous attacherons pas ici aux difficultés de résolution spécifiques à chacun de ces domaines. Au contraire, nous nous attacherons principalement aux relations d’équilibre à l’interface entre les différentes physiques.

1.4.1 Aéro-mécanique

Dans le cas du couplage mécanique quasi-statique entre un fluide et une structure, l’interaction aux limites à prendre en compte est l’équilibre mécanique à l’interface entre les dé-

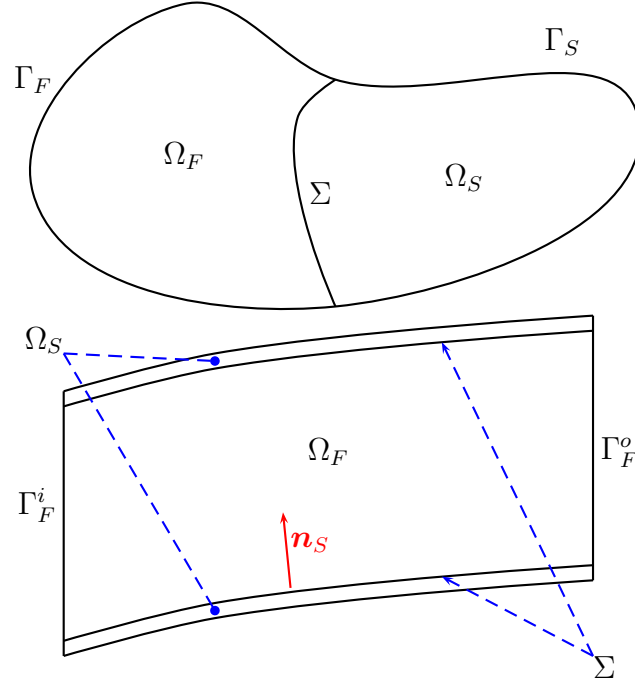


FIG. 1.1 – Problèmes multi-physiques : Ω_F et Ω_S désignent respectivement les domaines fluides et solides, Γ_F et Γ_S leurs frontières extérieures ; Σ désigne l'interface entre les deux domaines ; \mathbf{n}_S désigne la normale extérieure au domaine structure.

placements et les vecteurs des contraintes des deux milieux. Traduit sous forme d'équations, cela signifie qu'à tout instant t :

$$\begin{cases} \mathbf{u}_S \cdot \mathbf{n}_S = -\mathbf{u}_F \cdot \mathbf{n}_F \\ \boldsymbol{\sigma}_S \cdot \mathbf{n}_S = -\boldsymbol{\sigma}_F \cdot \mathbf{n}_F \end{cases} \quad \text{sur } \Sigma \quad (1.13)$$

où $\mathbf{n}_S = -\mathbf{n}_F$ désignent les normales sortantes des domaines solide et fluide respectivement.

Suivant les modèles que l'on considèrera, les conditions d'équilibre à l'interface seront simplifiées. Par exemple, si le fluide est considéré parfait, ses effets de cisaillement sont négligés et ne pourront donc pas être pris en compte à l'interface. Seule restera la contrainte normale, autrement dit la force de pression, que le fluide exerce sur la structure ; l'équation précédente sera réduite à :

$$\begin{cases} \mathbf{u}_S \cdot \mathbf{n}_S = \mathbf{u}_F \cdot \mathbf{n}_F \\ (\boldsymbol{\sigma}_S \cdot \mathbf{n}_S) \cdot \mathbf{n}_S = (\boldsymbol{\sigma}_F \cdot \mathbf{n}_F) \cdot \mathbf{n}_F \end{cases} \quad \text{sur } \Sigma \quad (1.14)$$

Par contre, dans le cas d'un fluide visqueux, les effets de paroi seront pris en compte au niveau des conditions d'interface et le fluide exercera des forces de cisaillement sur la paroi au travers du vecteur complet de contraintes. Réciproquement, les déplacements du solide exerceront un effet cisailant sur le fluide, tout en imposant ses déplacements au domaine fluide.

1.4.2 Aéro-thermique

L'aéro-thermique se concentre pour sa part sur les échanges thermiques entre un solide (fixe et indéformable) et un fluide en écoulement autour ou à l'intérieur du solide. Dans les

applications aéronautiques, il s'agit par exemple de problèmes d'échauffement de pièces telles que les aubes de turbines, les chambres de combustion ou les tuyères. Ces pièces étant soumises à des conditions thermiques sévères, elles sont souvent refroidies par injection de gaz froids ou par création d'un film de gaz froid le long de l'interface fluide-structure (film-cooling). Ces méthodes de refroidissements fournissent un deuxième exemple de problème aéro-thermique.

Nous ne considérerons que des interactions entre fluide et solide dues à la conduction. D'autres modèles d'interactions, prenant par exemple en compte le rayonnement, peuvent aussi être envisagés. Dans les phénomènes thermiques à très hautes températures, ce sont d'ailleurs ces phénomènes de rayonnement qui sont prépondérants.

À l'instar du couplage mécanique, la modélisation de chacun des domaines se fait par ses propres équations : celles de Navier-Stokes pour le fluide (éq. 1.1), et celle de la chaleur pour le solide (éq. 1.12). L'équilibre thermique à l'interface s'écrit entre les températures et les flux :

$$\begin{cases} T^S &= T^F \\ \phi^S &= -\phi^F \end{cases} \quad \text{sur } \Sigma \quad (1.15)$$

Le flux de chaleur en provenance du solide est donné par le gradient de la température :

$$\phi^S = -\kappa \text{grad } T^S$$

Le flux en provenance du fluide n'est pas aussi simple à écrire : il dépend des modèles de la loi de paroi utilisés. Nous renvoyons à [50, 17] pour son expression.

1.4.3 Thermo-mécanique

Une grande partie des applications mécaniques rencontrées dans le domaine aérospatial est soumise à des températures extrêmes. Le comportement des matériaux utilisés est fortement dépendant de ces températures, et cette dépendance induit un couplage entre l'évolution thermique et mécanique des pièces. En pratique, ce couplage est souvent asymétrique : l'influence des variations thermiques sur la mécanique est beaucoup plus importante que l'influence des déformations sur le problème de diffusion. Quoi qu'il en soit, le problème de thermo-mécanique est modélisé par les équations de la mécanique des milieux continus (éq. 1.4) et celle de la diffusion thermique (éq. 1.11) :

$$\begin{aligned} -\text{div}(\boldsymbol{\sigma}_S) &= \rho \mathbf{f} && \text{dans } \Omega_0 \\ -\text{div}(\kappa \text{grad } T) &= f^{th} && \text{dans } \Omega(t) \end{aligned}$$

où Ω_0 désigne le domaine de calcul dans sa configuration initiale, et $\Omega(t)$ ce domaine déformé au cours du temps. Ce couplage se distingue des deux précédemment présentés par deux points. Tout d'abord parce qu'il s'agit d'un couplage volumique, et non plus surfacique. Ensuite, parce que le couplage entre les deux problèmes ne se fait plus au travers d'un équilibre à l'interface.

Dans le cas de l'élasticité par exemple, les modules d'Young et coefficients de Poisson sont dépendants de la température ; dans les cas plus riches de visco-plasticité, les coefficients additionnels de la loi de comportement (viscosité, limites d'élasticité, etc...) en dépendront aussi. Les variations de température induisent aussi des déformations thermiques, notées $\boldsymbol{\xi}^{th}$ dans l'équation (1.9).

Réciproquement, l'influence de la mécanique sur le problème de diffusion se fait principalement au travers de l'évolution du domaine de calcul $\Omega(t)$. Il existe toutefois des modèles

dans lesquelles les déformations plastiques du solide sont génératrices de chaleur. Dans ces situations, le terme source f^{th} inclut donc une contribution en provenance de la mécanique⁶.

⁶C'est un phénomène d'échauffement suite à des déformations plastiques que nous avons tous eu l'occasion de mettre en évidence en tordant rapidement une fourchette par exemple.

Chapitre 2

Méthodes de couplage partitionnées

2.1 Introduction

Un problème multiphysique est mathématiquement représenté par un ensemble d'équations aux dérivées partielles en temps et en espace, telles que celles présentées dans le chapitre précédent. Dans [25], C. Felippa rappelle qu'il existe différentes méthodes permettant l'étude numérique de ces problèmes :

- les méthodes d'élimination de champs, nécessitant d'appliquer des méthodes intégrales ou de réduction de modèle à certains champs,
- les méthodes monolithiques, traitant l'ensemble du problème comme un bloc unique,
- les méthodes partitionnées, cherchant à isoler les différents champs les uns des autres.

La première méthode est particulièrement efficace puisqu'elle diminue le nombre d'inconnues du problème. Son spectre d'application n'est malheureusement pas aussi étendu que celui de ses concurrentes. Les deux autres méthodes sont utilisables dans plus de situations, et entrent en concurrence lorsqu'il s'agit d'en choisir une plutôt que l'autre pour certaines applications.

Les méthodes de couplage partitionnées ont pour point de départ l'écriture dissociée de chaque physique. Cela permet leur intégration en temps séparée, l'influence de chacune des physiques sur la (ou les) autre(s) se faisant au travers de conditions au bord.

Au contraire, les méthodes monolithiques conservent une écriture mêlée des différentes physiques, et intègrent simultanément les différentes équations. Ceci a pour principal avantage de ne pas ajouter d'instabilités supplémentaires à un système déjà complexe, ce qui, nous le verrons plus tard, est le principal inconvénient des méthodes partitionnées.

Par rapport aux méthodes monolithiques, les méthodes partitionnées présentent principalement quatre avantages :

- la flexibilité de la modélisation des différents phénomènes physiques (par exemple un modèle éléments finis côté solide, et un modèle volumes finis côté fluide, ou encore un schéma implicite côté solide, explicite côté fluide),
- l'indépendance de la modélisation de chaque physique, permettant l'utilisation de modèles non coïncidents (discrétisations de types et de finesse différentes),
- la possibilité de réutiliser des codes existant, ainsi que toute leur chaîne de pré- et post-traitement,
- l'extensibilité, c'est-à-dire la possibilité de continuer le développement de nouveaux modèles dans chaque code, indépendamment de l'aspect couplage.

Ces méthodes ont connu un essor au milieu des années 1970, simultanément dans trois groupes différents : Northwestern University (T. Belytschko et R. Mullen), Cal Tech (T. J. R. Hughes et W. K. Liu), et Lockheed Palo Alto Research Labs (J. A. DeRuntz, C. A. Felippa, T. L. Geers et K. C. Park). Ce dernier groupe travaillait alors sur des problèmes d'interactions entre ondes de chocs et sous-marins pour l'ONR (Office of Naval Research), travail pour lequel il leur a été nécessaire de coupler un modèle éléments finis du sous-marin à un modèle d'éléments frontières (BEM) pour l'acoustique externe [56, 55].

Dans un article de synthèse récent [25], les auteurs se souviennent justement qu'à cette époque, le choix de la méthode partitionnée a été motivé à la fois par la contrainte de réutilisation de code existant, et par l'impossibilité d'accéder aux sources de ces codes pour y ajouter des capacités multi-physiques. Ce sont d'ailleurs ces deux contraintes qui sont le plus souvent décisives lors du choix entre une méthode partitionnée ou monolithique.

Le titre du premier article de Park et Felippa — *Stabilization of staggered solution procedures for fluid-structure interaction analysis* [56] — fait par ailleurs déjà apparaître le principal inconvénient des méthodes partitionnées : il s'agit des difficultés de stabilité de cette technique vis-à-vis de l'intégration temporelle. En effet, la séparation des physiques entraîne forcément un décalage lors de l'intégration temporelle des deux problèmes. Ce décalage induit une incohérence entre les sous-problèmes à chaque pas de temps ; ces déséquilibres peuvent, selon la situation, se cumuler (cas défavorable) ou se compenser (cas favorable) au cours du calcul. On s'attachera plus longuement à cette question dans la suite de cette étude bibliographique (sections 2.3 et 2.4 en particulier).

Notons aussi quelques réserves à prendre en compte. La stabilité d'un problème couplé est bien entendu dépendante de la stabilité de chacun des sous-problèmes. Cependant, il n'est pas prouvé que ce soit une condition suffisante ! De nombreuses situations existent où, pris séparément, chacun des sous-problèmes est stable, mais mis ensemble le problème couplé ne l'est plus : problèmes d'entrée en résonance ou de flottements par exemple. De plus, dans la situation favorable où le processus couplé converge, il n'est pas prouvé en général que le résultat numérique obtenu soit bien la solution du problème couplé continu.

Ces problèmes de stabilité ont conduit à différents algorithmes de couplage, que l'on peut regrouper en deux familles, qualifiées respectivement de *couplage faible* et de *couplage fort* (d'autres distinctions plus parlantes sont aussi possibles, voir pour cela la section 2.2). La différence fondamentale entre ces deux familles vient de la manière d'imposer l'équilibre entre les différentes physiques, c'est-à-dire le principe d'action-réaction.

Bien entendu, ce principe est toujours présent dans les équations physiques, que ce soit dans le cas mécanique (éq. 1.13) ou thermique (éq. 1.15). Mais il n'est pas nécessairement respecté au niveau discret. Les deux discrétisations, temporelles et spatiales, sont sources de déséquilibre. Les algorithmes tenteront donc de minimiser au mieux ces sources d'erreurs.

La distinction entre algorithmes de couplages partitionnés *faibles* et *forts* apparaît au niveau de la discrétisation temporelle. Les algorithmes qualifiés de forts assurent à chaque pas de temps le principe d'action-réaction. Au contraire, les algorithmes faibles ne s'imposent pas cette contrainte. Bien entendu, si le principe n'est vraiment pas respecté, la solution obtenue ne correspondra pas à la solution du problème continu couplé. Les algorithmes faibles chercheront donc à respecter ce principe à une certaine erreur près, erreur qui peut devenir prépondérante dans certains champs d'applications.

La question analogue se pose pour le découplage spatial : étant données des discrétisations indépendantes de chacun des domaines, quelle est la méthode de transfert de champ

introduisant une erreur minimale et acceptable? Nous reviendrons sur cette question dans la section 2.6.

Nous allons maintenant voir plus en détails ces deux familles d’algorithmes, afin d’en réaliser une taxinomie. Il est important de noter dès à présent que parmi tous ces algorithmes, il n’en est malheureusement pas un qui puisse être qualifié d’universel. Chacun à ses avantages et ses inconvénients, et à chaque application devra correspondre un choix particulier d’algorithme. Ce choix sera souvent un compromis entre rapidité et précision de l’algorithme résultant, sa stabilité étant bien évidemment une condition *sine qua non* à l’obtention d’un résultat.

2.2 Terminologie couplage faible / couplage fort

Les différents articles traitant du couplage utilisent les termes “faibles” et “forts” pour signifier des idées différentes. Le terme “faible” par exemple est utilisé pour signifier :

- partitionné
- non équilibré (ne respectant pas nécessairement le principe d’action-réaction)
- explicite

Le terme “fort” pour sa part peut signifier, suivant les auteurs :

- monolithique
- algorithme respectant le principe d’action-réaction
- implicite

Ces différentes définitions sont bien entendu liées, mais ne sont pas équivalentes. Pour notre part, nous avons choisi de relier ces termes *fort* et *faible* au respect (ou non) du principe d’action-réaction, définition qui est aujourd’hui la plus communément admise. Ainsi, un algorithme monolithique est fort, alors qu’un algorithme partitionné ne l’est pas forcément. Un algorithme partitionné fort aura par contre vocation à obtenir des résultats comparables à ceux d’une méthode monolithique.

En respectant le principe d’action-réaction, les méthodes fortes permettent un gain en stabilité par rapport aux méthodes de couplage faible. Pour leur part, les méthodes faibles ne sont stables et fournissent des résultats corrects uniquement dans les situations où elles respectent quasiment le principe d’action-réaction.

2.3 Couplage faible

Historiquement, ce sont ces algorithmes qui ont été les premiers développés et utilisés. Ils ont montré une très bonne adaptation à de nombreux problèmes, d’aéro-élasticité par exemple.

Les travaux de Piperno *et al.* [61, 57] comparent les mérites respectifs d’un certain nombre de ces algorithmes dits *décalés*¹. Cette série d’algorithmes peut se présenter dans un cadre commun.

Pour décrire ces algorithmes, notons N le nombre de pas de temps du calcul. À un instant t_n , on désignera respectivement par $\mathbf{u}_\Sigma^n, \mathbf{u}_F^n, \mathbf{u}_S^n$ les déplacements de la frontière (Σ), du domaine fluide (Ω_F) et de la structure (Ω_S), et par \mathbf{v}^n, p^n la vitesse et la pression du fluide. Par commodité, on notera aussi par $*^n$ l’ensemble de l’état du système couplé à l’instant n .

¹en anglais “staggered”

L'algorithme permettant d'obtenir l'état à l'instant $n + 1$ à partir de celui à l'instant n se décompose en 5 phases :

Algorithme 1 : Algorithme partitionné séquentiel

pour $n = 1 \dots N$ faire 1 prédire le déplacement de l'interface fluide 2 en déduire le déplacement de l'ensemble du domaine fluide 3 résoudre le problème fluide dans ce domaine mobile 4 en déduire les forces (réactions) exercées sur l'interface 5 résoudre le problème solide fin	$\mathbf{u}_\Sigma^n = \mathcal{P}(*^{n-1})$ $\mathbf{u}_F^n = \mathcal{D}(\mathbf{u}_\Sigma^n, *^{n-1})$ $(\mathbf{v}^n, p^n) = \mathcal{F}(\mathbf{u}_F^n, *^{n-1})$ $f_\Sigma^n = \mathcal{R}(\mathbf{v}^n, p^n; *^{n-1})$ $\mathbf{u}_S^n = \mathcal{S}(f_\Sigma^n; *^{n-1})$
---	---

La terminologie “décalée” donnée à ce type d'algorithme vient du fait que les différentes physiques sont intégrées séparément en temps, d'où résulte un décalage entre les différents calculs.

En l'absence de meilleure possibilité, la prédiction faite à l'étape **1** pourra être simplement égale au déplacement calculé à l'itération précédente : $\mathbf{u}_\Sigma^n = \mathbf{u}_\Sigma^{n-1}$. L'algorithme résultant de ce choix de (non-)prédiction est appelé CSS (Conventional Serial Staggered) dans [61]. Cet algorithme présente un coût de calcul tout à fait raisonnable, puisqu'égal à la somme des temps de calcul fluide et structure.

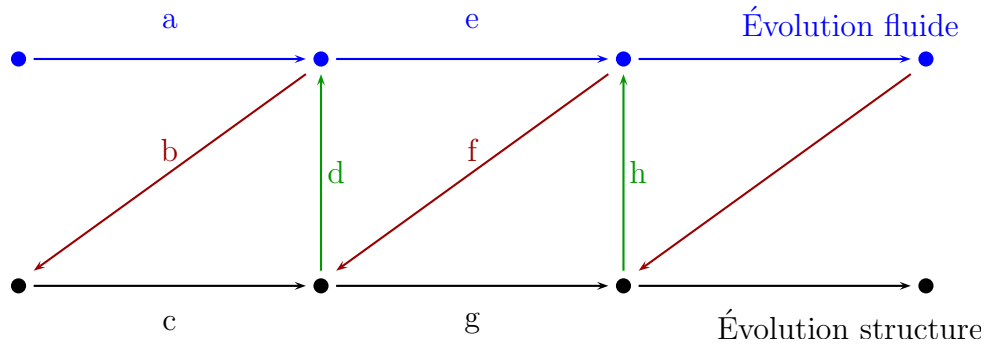


FIG. 2.1 – Algorithme partitionné séquentiel (CSS). L'étape (a) englobe les points **2** à **3** de l'algorithme **1**, l'étape (b) correspond au point **4**, l'étape (c) correspond au point **5**, et l'étape (d) correspond au point **1**.

Notons tout de suite que la prédiction faite à l'étape **1** n'est pas égale au déplacement de l'interface que l'on retrouve à l'étape **5** (sauf cas exceptionnels bien entendu). La stabilité et la précision de l'algorithme résultent directement de la qualité de cette prédiction.

Cet algorithme simple est en pratique rarement utilisé tel quel ; on lui préfère l'une des variantes ci-dessous, à peine plus complexes et présentant de bien meilleures propriétés de stabilité.

2.3.1 Algorithme sous-cyclé

La première variante de l'algorithme ci-dessus part de la constatation suivante : la taille des pas de temps doit y être à la fois convenable pour le fluide et pour la structure. Or généralement, les phénomènes fluides nécessitent des pas de temps beaucoup plus petits que

ceux de la structure (couramment d'un facteur 100 ou 1000). On se retrouve donc à résoudre le problème structure beaucoup plus fréquemment que si le problème était découplé.

Le sous-cyclage consiste à ne réaliser le couplage qu'à certains instants, par exemple à la fréquence de la physique la plus lente (Fig. 2.2). La physique la plus rapide ne sera donc couplée qu'à certains de ses pas de temps.

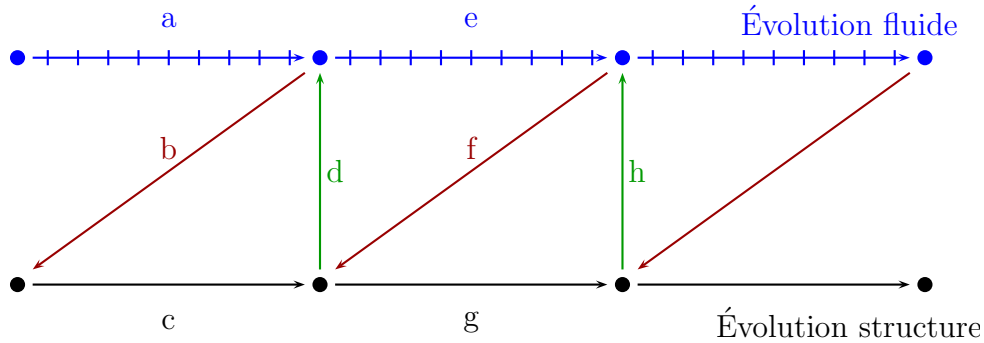


FIG. 2.2 – Algorithme partitionné séquentiel sous-cyclé (CSS sous-cyclé). Cet algorithme fait intervenir trois pas de temps : celui de couplage, du fluide et de la structure (égal à celui de couplage dans cette figure).

Entre deux instants de couplage (étape (a) sur la figure 2.2), le fluide utilise une interpolation linéaire entre les positions de la structure à deux instants de couplage.

L'algorithme résultant est, suivant le rapport entre les temps caractéristiques des deux physiques, beaucoup plus léger, et ne dégrade pas nécessairement la stabilité de la solution couplée.

On peut noter qu'en pratique, ce sous-cyclage peut être vu comme purement interne au code fluide, et que le code structure peut ne pas en avoir conscience. Du point de vue du code fluide, au cours d'un pas de temps de couplage, la structure est supposée évoluer linéairement entre les deux états couplés qui ont été transmis. Cette hypothèse est raisonnable puisque c'est celle sous-jacente au choix de discrétisation temporelle de la structure.

Le coût de cet algorithme est à nouveau la somme des temps de calcul fluide et structure, mais cette fois le nombre d'itérations nécessaire dans chaque sous-problème n'est pas surestimé par des considérations de couplage.

2.3.2 Variante parallèle

En gardant le principe précédent, il est possible de créer une variante parallèle de l'un ou l'autre des deux algorithmes précédents. Par simplicité, on n'écrira la variante que dans le cas sans sous-cyclage.

Le but de cette variante est de réaliser simultanément les résolutions fluide et structure, ce qui apportera un gain de temps sur des calculateurs à architecture parallèle. Pour cela, il faut cette fois réaliser deux prédictions, la première pour le champ de déplacement de la frontière couplée du fluide, la deuxième pour la force appliquée à la frontière couplée de la structure. L'algorithme résultant est représenté sur la figure 2.3 ; il est nommé *CPS* (Conventional Parallel Staggered).

Cette variante est particulièrement intéressante lorsque les temps de calcul des deux physiques sont similaires : son coût est en effet de l'ordre du calcul le plus cher. Par con-

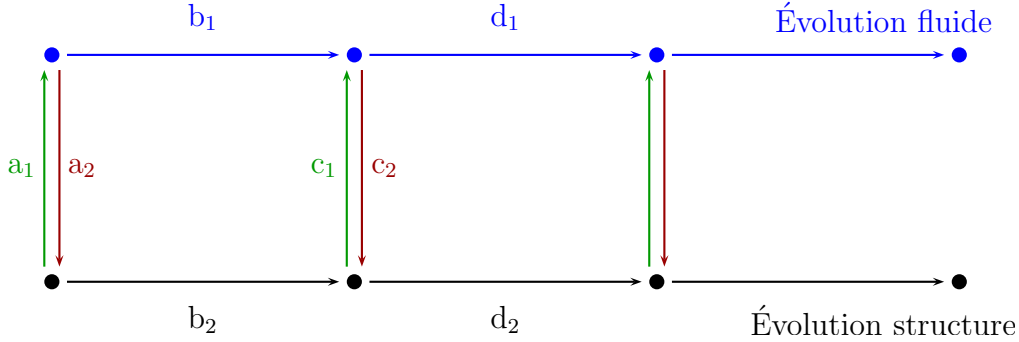


FIG. 2.3 – Algorithme partitionné parallèle (CPS). Les étapes pouvant être réalisées en parallèle portent le même numéro. Il s’agit principalement des étapes de résolution de chacune des physiques (b_1 et b_2), qui sont les étapes “chères” du calcul.

tre, si les temps de calcul des deux physiques sont très différents, le gain de la résolution simultanée par rapport à une résolution séquentielle sera moins flagrant.

En pratique, on constate malheureusement une moins bonne précision de cet algorithme par rapport aux précédents. Intuitivement, cela peut se comprendre, puisqu’il se base sur deux prédictions au lieu d’une. Dans [61], Piperno *et al.* démontrent cette baisse de stabilité dans un cas 1-D.

Le gain de temps de calcul se fait donc au détriment de la précision, voire de la stabilité du résultat. On sera tenté de compenser cela par une augmentation du nombre de pas de temps, augmentant le coup total du calcul. Un certain compromis sera donc à trouver.

Remarque

Il peut paraître superflu de prévoir un algorithme de couplage parallèle lorsque les codes sont déjà individuellement parallélisés : dans cette situation effectivement, les codes profitent séparément de l’architecture parallèle des machines de calcul.

Mais en pratique, il arrivera que les codes ne puissent pas tourner sur les mêmes machines. ZéBuLoN par exemple est bien adapté aux machines scalaires, mais a des performances limitées sur architecture vectorielle. Au contraire, MSD a été développé pour des machines vectorielles, et profite pleinement des avantages de ce type de machines. Un calcul faisant intervenir des calculateurs hétérogènes aura donc son intérêt.

Bien qu’ayant gardé ce point à l’esprit, nous n’avons pas réalisé de tels calculs sur machines multiples dans le cadre de cette thèse. Cela aurait apporté une difficulté informatique supplémentaire à celles déjà existantes.

2.3.3 Algorithme à échanges désynchronisés

Dans [61], Piperno *et al.* suggèrent de désynchroniser les échanges entre les deux codes, afin d’améliorer la précision de l’algorithme parallèle précédent.

Cette variante fait intervenir des “demi-pas de temps de couplage”, lors desquels un échange d’information supplémentaire est effectué. Plus précisément, les deux codes effectuent en parallèle la première moitié du pas de temps de couplage, puis échangent leurs champs à l’interface. Ils terminent alors le pas de temps en utilisant les données actualisées. Ce décalage permet de réaliser en quelque sorte une prédiction améliorée de l’état de la structure. Ces modifications

fournissent un algorithme de couplage parallèle, mais reproduisant la précision de l'algorithme séquentiel.

L'idée de désynchronisation sera par la suite reprise par Altmikus *et al.* pour la simulation aéro-élastique de rotors [1], puis par Farhat *et al.* [24] qui démontreront que le décalage d'un demi pas de temps entre les deux codes améliore la précision du schéma couplé en temps, pour atteindre un ordre 2. Dans le cas de codes, qui, pris indépendamment, utilisent des schémas d'intégration en temps d'ordre 2, cet algorithme évite donc la dégradation de cet ordre de précision.

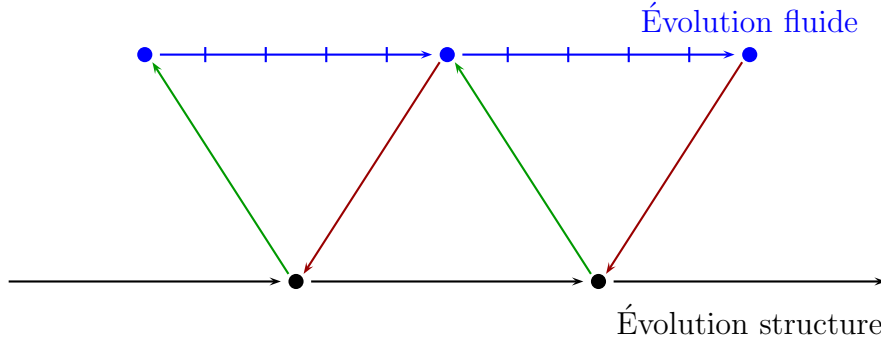


FIG. 2.4 – Algorithme partitionné à échanges désynchronisés

L'algorithme résultant, schématisé sur la figure 2.4, est appelé GSS (*Generalized Serial Staggered*) par ses auteurs.

2.3.4 Prédictions

Les trois algorithmes présentés précédemment mentionnent tous des prédictions, mais n'utilisent qu'une prédiction d'ordre 0 : $\mathbf{u}_\Sigma^n = \mathbf{u}_S^{n-1}$. Bien entendu, cette prédiction minimaliste peut être facilement améliorée, en prenant par exemple une extrapolation d'ordre 1 :

$$\mathbf{u}_\Sigma^n = \mathbf{u}_S^{n-1} + \Delta_t \dot{\mathbf{u}}_S^{n-1} \quad (2.1)$$

ou d'ordre 2 :

$$\mathbf{u}_\Sigma^n = \mathbf{u}_S^{n-1} + \frac{3}{2} \Delta_t \dot{\mathbf{u}}_S^{n-1} - \frac{1}{2} \Delta_t \dot{\mathbf{u}}_S^{n-2} \quad (2.2)$$

où Δ_t désigne le pas de temps de couplage (supposé ici régulier, afin de simplifier les notations), et $\dot{\mathbf{u}}_S^n$ représente la vitesse de l'interface à l'instant n .

À nouveau, les auteurs de [61] ont démontré le gain en stabilité et en précision de cette prédiction, d'une part théoriquement sur un modèle 1-D et d'autre part au travers d'expériences numériques sur des applications plus complexes (concernant le flottement de profils d'ailes d'avions).

Notons tout de suite que l'ajout de l'un ou l'autre de ces prédicteurs est assez aisé à mettre en œuvre : cela n'implique pas de modification importante des trois algorithmes de résolution (celui du fluide, de la structure et du problème couplé). Il est aussi intéressant de constater que ces prédictions peuvent s'appliquer aux trois algorithmes déjà présentés : l'algorithme décalé de base, le sous-cyclé et le parallèle.

2.3.5 Fluide-structure ou structure-fluide ?

On peut aussi noter que les deux algorithmes en série admettent la variante “structure-fluide”, dans laquelle la prédiction est réalisée sur le champ de forces appliquées au lieu du champ de déplacement. Le solide est alors résolu en supposant cette prédiction, et le fluide sur un déplacement actualisé.

Il n’est pas forcément clair quel ordre est préférable, mais il semble raisonnable de réaliser la prédiction soit sur le champ le plus facile à prédire correctement, soit sur celui qui a la plus faible influence sur l’autre physique.

2.4 Couplage partitionné fort

Dans certaines situations, le déséquilibre d’énergie à l’interface est trop important lorsqu’on emploie des méthodes partitionnées faibles. Il s’agit principalement de situations rencontrées lorsque les rigidités des deux physiques sont comparables. Ce phénomène instable apparaît par exemple sur des problèmes d’interactions entre fluides denses et structure [48, 43], des problèmes d’hémodynamique² [51, 12], ou encore sur des problèmes thermiques³ [13]. Les auteurs respectifs ont mis en évidence l’influence d’un terme interprétable comme une “masse ajoutée”. Ce terme provenant de l’inertie du fluide vient s’ajouter à la masse de la structure pour en modifier le comportement dynamique.

Sur ces différents problèmes, les auteurs ont donc constaté les limites des méthodes partitionnées décalées que nous avons décrites précédemment. Diverses solutions ont été avancées pour traiter ces cas délicats ; nous allons en examiner quelques unes. Il est important de noter que toutes ces méthodes cherchent à améliorer la prédiction faite à l’étape 1. En effet, dans le cas où celle-ci est exacte, l’équilibre à l’interface est respecté, et l’algorithme se stabilise. Ces algorithmes ont donc en commun la recherche du point fixe de l’application composant les différentes phases de l’algorithme de base :

$$\mathcal{S} \circ \mathcal{R} \circ \mathcal{F} \circ \mathcal{D}$$

(les notations étant reprises de la section 2.3).

Différentes méthodes usuelles de recherche de points fixes ont donc été employées dans ces situations : méthodes itératives de Jacobi, de Gauss-Seidel ou de Newton.

2.4.1 Corrections par itérations, méthode de Jacobi

L’ajout d’une phase de correction apporte une solution dans certaines situations (voir par exemple l’article de Le Tallec et Mouro [43], ou de Mok *et al.* [47]). Son principe est de vérifier, en fin de pas de temps de couplage, la qualité de la prédiction. Si elle n’est pas satisfaisante, il est possible de corriger la prédiction et de recommencer le pas de temps. Cette phase de correction permet d’assurer l’équilibre à l’interface dans les situations où cette boucle converge, et donc aussi une certaine stabilité au schéma.

À nouveau, comme pour la phase de prédiction, diverses solutions sont envisageables, présentant chacune des avantages et des inconvénients. Afin de rendre compte de l’itérativité

²l’hémodynamique étudie les problèmes de simulation d’écoulements sanguins

³dans ce cas, c’est la conductivité des matériaux qui joue le rôle de la rigidité

de la méthode, un indice k représentant l'itération en cours est ajouté aux notations précédemment employées. L'algorithme devient :

Algorithme 2 : Algorithme itératif corrigé (méthode de Jacobi)

	pour $n = 1 \dots N$ faire ;	// incrément de temps
	$k = 0$	
	répéter ;	// itération de couplage à temps fixé
1	prédire le déplacement de la frontière fluide :	
	si $k = 0$	$d_{\Sigma,(k)}^n = \mathcal{P}(*^{n-1})$
	si $k > 0$	$d_{\Sigma,(k)}^n = \mathcal{P}(*_{k-1}^n)$
2	l'étendre à l'intérieur du domaine fluide	$d_{(k)}^n = \mathcal{D}(d_{\Sigma,(k)}^n, *^{n-1})$
3	résoudre le problème fluide	$(u_{(k)}^n, p_{(k)}^n) = \mathcal{F}(d_{(k)}^n, *^{n-1})$
4	en déduire les réactions sur l'interface	$f_{(k)}^n = \mathcal{R}(u_{(k)}^n, p_{(k)}^n; *^{n-1})$
5	résoudre le problème solide	$d_{(k)}^n = \mathcal{S}(f_{(k)}^n; *^{n-1})$
	jusqu'à ce que la prédiction soit suffisamment précise	$\ d_{(k)}^n - d_{\Sigma,(k)}^n\ < \varepsilon$
	fin	

La correction la plus simple consiste à utiliser le résultat de l'étape 5 comme prédiction à l'étape 1 de l'itération suivante (ce qu'on symbolise par $d_{\Sigma,(k)}^n = \mathcal{P}(*_{k-1}^n)$), un peu à la manière des méthodes itératives de Jacobi pour l'inversion de matrices. Cette analogie ne s'arrête pas là, puisqu'on retrouve le même type de difficultés de stabilité, nécessitant parfois l'utilisation d'un paramètre de relaxation. Cette stabilisation se fait alors au détriment de la vitesse de convergence.

Dans les situations les plus sensibles, cette relaxation est parfois indispensable [51, 43, 39]. L'approche présentée par Le Tallec et Mouro, réalisant un parallèle avec les méthodes de décomposition de domaine, leur permet d'ailleurs de proposer diverses possibilités pour le choix du paramètre de relaxation.

Nous reviendrons sur cette méthode dans la section 5.1, en proposant sur des cas simples d'interaction structure-structure une analogie complète avec les méthodes matricielles.

Estimation de l'erreur

Lors de la phase de vérification de la prédiction, il est nécessaire de choisir un certain critère de convergence, c'est-à-dire une norme mesurant la différence, ainsi qu'un seuil acceptable.

La norme choisie pourra par exemple être une norme sup discrète, c'est-à-dire assurant que nœud par nœud, la différence entre prédiction et valeur reçue ne dépasse pas un certain seuil.

On pourra lui préférer une norme L^2 , permettant de donner un sens énergétique à cette erreur (on donnera une expression de cette énergie dans la section 2.5.1). Le seuil pourra ainsi être choisi en fonction des autres marges d'erreurs inhérentes aux discrétisations de chacun des sous-domaines.

2.4.2 Autres méthodes de couplage fort

Il faut tout de même garder à l'esprit que chaque étape de correction nécessite un calcul complet du fluide et de la structure. Ainsi, lorsqu'une unique correction est nécessaire (cas

des problèmes convergeant bien), le temps de calcul est néanmoins doublé. Les problèmes plus complexes, dont font partie les études d'écoulement sanguin, nécessitent un nombre important de phases de correction, se comptant parfois en centaines. Pour ces types d'applications, d'autres méthodes plus efficaces mais plus spécifiques ont donc été envisagées.

Il est en effet bien connu que la méthode de Jacobi n'est pas la méthode de recherche de point fixe la plus efficace. Mok et al. [47] comparent par exemple deux méthodes itérées avec relaxation : la méthode du gradient, et la méthode d'accélération d'Aitken. Dans la méthode du gradient, un paramètre de relaxation localement optimal est calculé. Ce calcul, malgré tout assez coûteux, permet sur certains exemples de réduire d'un facteur 2 le nombre d'itérations nécessaires. La méthode d'Aitken propose une alternative plus économique en temps de calcul, et présentant une accélération de convergence similaire.

Matthies et Steindorf [46] proposent des méthodes de type point fixe de Newton. Ces méthodes sont beaucoup moins gourmandes en nombre d'itérations, mais nécessitent le calcul de l'opérateur dérivé de $\mathcal{S} \circ \mathcal{R} \circ \mathcal{F} \circ \mathcal{D}$.

Cette dérivée pouvant être très coûteuse à déterminer, Gerbeau et Vidrascu [33] suggèrent de se restreindre au calcul d'un opérateur approché de la dérivée, plus rapide à calculer et conservant une bonne vitesse de convergence.

2.5 Critères de qualité

2.5.1 Création d'énergie à l'interface

De nombreux auteurs ont constaté que les instabilités des schémas de couplage partitionné sont liées au déséquilibre entre les énergies transmises au travers de l'interface par chacune des deux physiques. Le Tallec et Mouro [43] proposent d'ailleurs une estimation de cet écart dans le cas d'un schéma partitionné du type CSS :

$$\int_{\Sigma} \mathbf{n} \cdot \boldsymbol{\sigma}_F^{n+1} \cdot (\mathbf{u}_S^{n+1} - \mathbf{u}_F^{n+1}) \quad (2.3)$$

où \mathbf{n} est la normale à l'interface, $\boldsymbol{\sigma}_F^{n+1}$ le tenseur des contraintes dans le fluide égal à celui dans la structure par construction de l'algorithme de couplage, \mathbf{u}_S^{n+1} et \mathbf{u}_F^{n+1} les déplacements du solide et du fluide respectivement.

L'algorithme avec une prédiction constante utilise $\mathbf{u}_F^{n+1} = \mathbf{u}_S^n$, d'où un écart

$$\int_{\Sigma} \mathbf{n} \cdot \boldsymbol{\sigma}_F^{n+1} \cdot (\mathbf{u}_S^{n+1} - \mathbf{u}_S^n) \quad (2.4)$$

qui ne peut pas être nul.

Les auteurs interprètent alors ce terme comme une énergie créée artificiellement à l'interface par l'algorithme de couplage, énergie qui vient s'ajouter progressivement au système, et tend à le déstabiliser.

2.5.2 Stabilité

La stabilité d'un algorithme de couplage dépendra de sa faculté à ne pas créer d'énergie supplémentaire [57].

Cette phrase résume effectivement la source principale de dégradation de stabilité causée par le partitionnement. Plus précisément, les systèmes découplés auxquels on s'intéresse sont stables, pris séparément. L'écriture des conditions de compatibilité à l'interface impose que l'énergie extraite d'un système est exactement égale à celle reçue par l'autre domaine. Le phénomène continu couplé est donc stable.

Par contre, comme les schémas numériques de résolution choisis ne respectent pas cet équilibre, il n'y a plus de raison de conserver cette stabilité au niveau discret.

Afin d'améliorer la précision de cet équilibre à l'interface, deux principes s'affrontent : le premier va consister à appliquer des étapes de corrections, le deuxième consistera simplement à diminuer les pas de temps. Un critère de coût de calcul permettra de préférer l'une ou l'autre des solutions.

2.6 Découplage en espace

Le début de ce chapitre s'est principalement consacré au problème de la discrétisation temporelle. Cependant, l'utilisation de plusieurs codes de calculs va aussi être accompagnée de problèmes de raccord entre les différents domaines. Certains pré-traitements géométriques [34] permettent de travailler à maillages coïncidents, ce qui facilite le transfert de champs entre les domaines. Mais cela représente une contrainte importante qu'on ne peut, ou qu'on ne veut, pas toujours s'imposer.

Diverses méthodes existent pour réaliser ce transfert de champs. De manière analogue au découplage temporel, chaque méthode ajoutera sa part d'erreur, et des méthodes trop grossières risqueront de déstabiliser le processus couplé.

L'un des intérêts majeurs des méthodes partitionnées est l'utilisation de codes de calculs spécialisés dans chacune des physiques à traiter. En conséquence, il sera extrêmement rare d'utiliser des codes présentant un raccord exact à l'interface. En pratique, le code de calcul des structures utilisera par exemple des maillages de types éléments finis quadratiques, ou des modèles en "arêtes de poisson" pour modéliser les ailes d'un avion, alors que le fluide pourra utiliser un modèle volume fini. La finesse des mailles n'aura pas non plus le moindre intérêt à être identique entre les deux domaines. Au contraire, il faudra profiter de maillages bien adaptés à chaque physique étudiée.

La conséquence fondamentale du point de vue du couplage est donc la nécessité de pouvoir transférer des données pariétales d'une physique à l'autre. Comme on l'a déjà mentionné précédemment, la qualité de l'algorithme de couplage est liée à la manière dont est discrétisée la condition d'équilibre des forces à l'interface : tout déséquilibre a tendance à créer une énergie numérique, déstabilisant le processus couplé.

La méthode la plus simple de transfert reste l'interpolation directe des champs d'un domaine sur l'autre. Elle est parfaitement satisfaisante dans de nombreux cas (voir par ex. [22]). Il a néanmoins été montré par Grandmont *et al.* [40, 38] que cette méthode ajoute une erreur supplémentaire non négligeable à l'ensemble du problème. En effet, l'erreur due à l'interpolation peut être d'un ordre supérieur à celle due à la finesse de discrétisation spatiale par éléments finis.

Dans le cas de raccords surfaciques, la méthode des joints⁴ [6] permet de rendre cette erreur supplémentaire négligeable devant les erreurs dues à la discrétisation par éléments finis. Elle

⁴en anglais "mortar element method"

est en ce sens une méthode de transfert optimale. En contrepartie elle engendre un surcoût de calcul au niveau de l'interface. Bien que n'apportant pas d'amélioration significative aux cas étudiés dans [22], la méthode connaît actuellement de nombreux adeptes [3, 10], applications pour lesquelles elle semble indispensable.

2.7 Méthodes multi-domaines et conditions de raccord

Les algorithmes présentés précédemment peuvent être comparés aux méthodes de décomposition de domaines. L'intérêt est alors une nouvelle forme d'interprétation de ces algorithmes, permettant la compréhension de manière différente de leur (in-)stabilité et de leur vitesse de convergence.

Un certain nombre de résultats initialement énoncés dans le cadre des méthodes multi-domaines (différentes méthodes par domaine) ou dans le cadre des méthodes de décomposition de domaines (en vue de résolution en parallèle) trouvent des applications dans le cadre du couplage fluide-structure.

En décomposition de domaines, le type de conditions de raccord choisi pour chacun des domaines donne son nom aux algorithmes de couplage. Ainsi, le fluide se voyant ici imposer des déplacements (condition de Dirichlet) et le solide une pression (condition de Neumann), l'algorithme résultant est nommé Dirichlet→Neumann. Nous allons adopter ce type de nommage pour les algorithmes de couplage.

En se basant sur des travaux de décomposition de domaines de Quarteroni [62], Le Tallec et Mouro [43] ont proposé une interprétation "décomposition de domaines" pour le couplage d'un fluide et d'une structure. Cette interprétation repose sur une écriture commune des équations des deux physiques, dans le cadre de la mécanique des milieux continus. Les deux milieux ne se différencient en fait qu'au niveau de l'écriture des lois de comportement reliant contraintes et déformations. Leur travail, basée sur une approche par éléments finis, permet de retrouver naturellement le couplage Dirichlet→Neumann comme une conséquence du choix de décomposition des espaces de fonctions tests.

Les résultats de stabilité et convergence, déjà connus pour les méthodes multi-domaines de structures multi-matériaux trouvent dans le cadre des interactions fluide-structure une interprétation intéressante.

Divers résultats trouvent une explication naturelle dans ce cadre : par exemple, la bonne efficacité de la méthode Dirichlet→Neumann dans le cas du couplage entre une structure rigide et un fluide "léger", point sur lequel nous reviendrons dans la section 5.1.

Les méthodes utilisées dans le cadre du calcul parallèle peuvent donc trouver leur application dans les cas de calculs couplés. Les conditions de raccord mixtes de type Dirichlet→Fourier, ou Fourier→Fourier utilisées par S. Chemin [13] dans le cas de couplages thermiques ont par exemple été initialement développées pour des problèmes de décomposition de domaine [65].

Ces méthodes de raccord mixtes ont montré leur grande efficacité dans les cas de couplage thermiques. Elles commencent par ailleurs à faire leur apparition en couplage mécanique [4], pour des applications en hémodynamique, domaine où les densités du fluide et du solide sont très proches.

Deuxième partie

Couplage de codes de calcul
étude théorique

Chapitre 3

Paradigme du couplage de code

Les nombreux algorithmes de couplage partitionné issus de la littérature ont une base commune : ils reposent tous sur un échange de quantités à l'interface fluide-structure. L'algorithme utilisé est par contre toujours adapté à l'application envisagée et aux problèmes de stabilité/précision/rapidité rencontrés.

Cherchant à développer une plateforme de couplage qui puisse répondre aux divers problèmes couplés rencontrés dans le domaine aérospatial, nous faisons ici la synthèse de ces méthodes de couplage, et en déduisons les bases pour cette plateforme couplée. Nous passerons ensuite à sa mise en œuvre. Du fait des multiples possibilités et variantes de méthodes couplées, l'approche orientée objet est bien adaptée à cette mise en œuvre. Elle nous force aussi à identifier clairement les différents besoins pour cette plateforme couplée.

Afin de fixer les idées, nous baserons la discussion de ce chapitre sur les trois problèmes couplés qui nous intéresseront par la suite : le couplage aéro-mécanique, le couplage aéro-thermique et le couplage aéro-thermo-mécanique. Dans ces trois cas, il s'agira d'un couplage entre états fluides stationnaires et états solides quasi-statiques. Cette particularité nous conduira par endroits à faire certains choix qui ne seraient pas nécessairement adaptés à des cas de couplage entre des états fluides instationnaires et des états solides dynamiques.

Notre but sera ici triple :

- proposer un cahier des charges permettant le couplage à deux ou trois codes, dont l'une des caractéristiques sera de permettre un large éventail de méthodes numériques de couplage ;
- décrire, en vue d'un développement ultérieur, les différentes briques nécessaires à ce couplage ;
- étudier la faisabilité d'un couplage à trois codes.

Cette partie est à ce titre une synthèse des méthodes issues de la bibliographie, rapportées dans le contexte des applications que nous envisageons.

3.1 Le problème couplé

Commençons par rappeler (cf. chapitre 2) qu'une méthode de couplage est dite partitionnée lorsque les équations des différents phénomènes physiques pris en compte sont écrites séparément, et que l'interaction entre les différentes physiques apparaît dans ces équations comme conditions aux limites ou comme champ de paramètres extérieurs.

Ce découplage des physiques nécessite en conséquence l'utilisation d'un algorithme adéquat de (dé)couplage, permettant de retrouver la solution du problème global. L'algorithme s'attachera à la fois à la question du découplage temporel et du découplage spatial des deux

physiques. Suivant les problèmes étudiés, c’est l’une ou l’autre de ces deux questions qui s’avèrera difficile, voire les deux.

Chacune des physiques est régie par ses propres équations. Par exemple, dans le cas d’un couplage fluide-structure, ce sont les équations de Navier-Stokes (éq. 1.1) pour le fluide et celles de la mécanique des milieux continus (éq. 1.4) pour le solide. Le couplage apparaît au travers d’équations d’équilibre des déplacements et des contraintes à l’interface :

$$\begin{cases} \mathbf{u}_S = \mathbf{u}_F \\ \boldsymbol{\sigma}_S \cdot \mathbf{n}_S = -\boldsymbol{\sigma}_F \cdot \mathbf{n}_F \end{cases} \quad \text{sur } \Sigma$$

Le cas du couplage thermique est analogue : il couple la partie thermique des équations de Navier-Stokes (éq. 1.1) à celles de la diffusion thermique dans le solide (éq. 1.11). L’équilibre à l’interface s’exprime cette fois en termes de températures et de flux de chaleur :

$$\begin{cases} T^S = T^F \\ \phi^S = -\phi^F \end{cases} \quad \text{sur } \Sigma$$

Note. Il est naturel, comme on vient de le faire dans les lignes précédentes d’imaginer que le fluide exerce une pression sur la structure et, réciproquement, que la structure impose ses déplacements au fluide. Pourtant, les équations d’équilibre sont symétriques, et il est tout autant vrai que le fluide impose ses déplacements à la structure, ou que la structure impose une pression au fluide. Il est finalement plus correct de dire que les deux physiques *accordent* leurs déplacements et contraintes à l’interface.

Dans le cas des contraintes de cisaillement exercées par un fluide très visqueux sur une plaque, cet accord devient plus intuitif : la plaque est entraînée par le fluide, mais si elle se déplace elle entraîne tout autant le fluide dans son élan.

L’instinct apparaîtra tout de même comme un bon allié lorsqu’il sera question de choisir par la suite un algorithme de couplage (cf. section 5.1.3).

3.2 Coupleur pilote ou codes indépendants

La mise en données de l’algorithme de couplage est un point complexe. Nous avons besoin qu’elle soit suffisamment souple pour permettre, le cas échéant, le passage d’un algorithme à un autre, en permettant par exemple un ajout de phases de prédiction, de corrections, ou encore de recherche de points fixes du problème couplé. Notre objectif est donc de permettre l’extensibilité des algorithmes qui seront retenus. Un deuxième point sera bien entendu primordial : il s’agit de l’optimalité du résultat, en terme de coût (temps de calcul et mémoire).

Dans la formulation analytique ci-dessus, et si on s’inspire des méthodes de décomposition de domaines, chaque code apparaît comme un opérateur Dirichlet→Neumann¹ ou Neumann→Dirichlet². L’état d’équilibre se cherche sur les champs à l’interface, et il est donc agréable de pouvoir considérer les codes comme des boîtes noires réalisant ces opérateurs.

Si les codes que l’on a à disposition acceptent de se comporter ainsi, à savoir qu’ils transforment par exemple un champ de déplacement en champ de pression ou le contraire, une solution de couplage est alors obtenue en ajoutant un module “pilote”, gérant le problème à l’interface, la discrétisation en temps et l’algorithme de couplage. À la manière des méthodes

¹C’est-à-dire l’opérateur exprimant le champ de pression à l’interface en fonction de son déplacement.

²Opérateur inverse, qui exprime le champ de déplacement à l’interface en fonction du champ de pression.

multi-domaine de type condensation à l'interface, ce module part d'une solution "devinée" des champs à l'interface, et cherche ensuite par une méthode itérative à en faire un champ admissible dans chacun des deux problèmes fluide et solide. La figure 3.1 illustre cette solution.

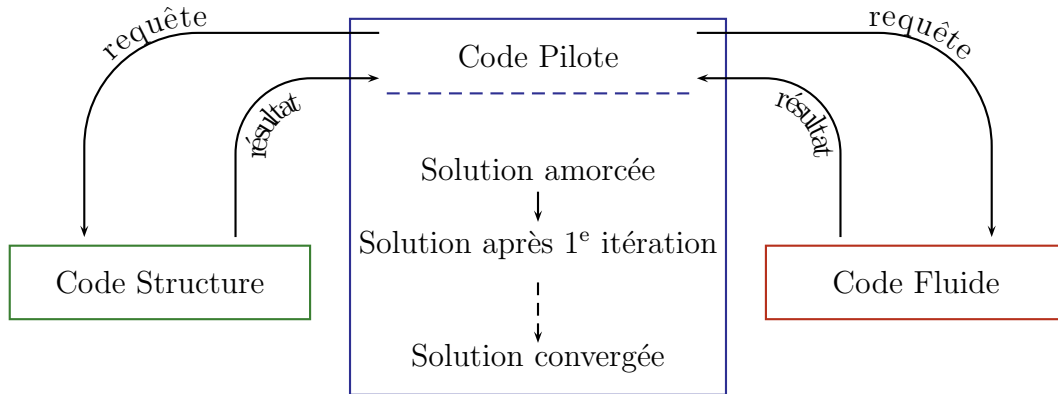


FIG. 3.1 – Couplage par code pilote

Cette solution, bien qu'élégante et agréable à décrire présente toutefois l'inconvénient d'être très intrusive et dépendante des codes. En effet, le module maître nécessite une connaissance forte de chacun des deux codes. Gérer un champ à l'interface est dépendant de la discrétisation (finesse, mais surtout type de discrétisation). Un champ de données aux nœuds ne s'interprète pas de la même manière selon qu'il est associé à un maillage de type volumes finis ou de type éléments finis (ce dernier pouvant en plus être linéaire ou d'ordre supérieur).

De plus, les codes doivent accepter ce type de pilotage. Il est peu fréquent qu'un code initialement développé pour des calculs découplés soit facilement modifiable en un code esclave³. Il apparaît donc que le code maître devient spécifique au couplage de deux codes particuliers, et nécessite un investissement non négligeable.

Mais l'inconvénient majeur de cette solution apparaîtra si l'on désire utiliser des méthodes de calcul parallèle, par exemple une méthode de décomposition de domaines (méthode utilisée couramment en mécanique des structures). Dans cette situation, le code pilote devra :

- soit être lui aussi parallélisé ; se posera alors la question de la manière de le paralléliser ;
- soit être séquentiel ; il risque alors de devenir un goulet d'étranglement du calcul.

Malgré ces quelques inconvénients, cette solution est tout à fait envisageable, et est utilisée pour des applications de grande envergure. Citons par exemple le projet Palm [53], qui fournit une interface facilitant la description de l'algorithme de couplage, et génère ensuite le code source de l'application couplée. Ce projet ne se restreint d'ailleurs pas uniquement aux problèmes couplés, et englobe aussi toute application nécessitant le dialogue et l'enchaînement de plusieurs applications, tels que les processus d'optimisation par exemple.

Ces inconvénients nous ont donc convaincus de conserver une approche autonome du couplage, et de réaliser l'investissement logiciel indépendamment dans chacun des codes. Formellement, cette approche est équivalente à la précédente puisqu'elle se base elle aussi sur la vision partitionnée, et implante les mêmes algorithmes. Mais sa mise en œuvre présente quelques avantages, que nous allons décrire ici.

³Citons toutefois l'exemple d'elsA, code de mécanique des fluides développé à l'Onera, qui a été construit dans une optique modulaire, et accepte donc ce genre de pilotage.

Il s’agit tout d’abord de l’absence d’un module pilote du couplage : rappelons que celui-ci est extrêmement dépendant de chacun des deux codes, et nécessite une forte intrusion. Le développement de ce module nécessite aussi une bonne connaissance à la fois de la mécanique des fluides et de la mécanique des structures, double compétence difficile à acquérir. En contrepartie il est nécessaire de définir de nombreuses conventions, sorte de cahier des charges à respecter par chaque code voulant prendre part au couplage.

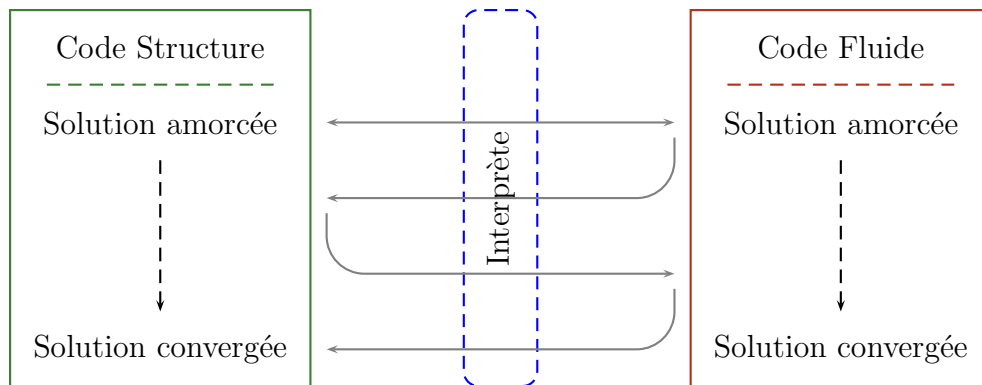


FIG. 3.2 – Couplage par bibliothèque externe

Le but de l’approche autonome est que les codes aient une indépendance maximale à tout point de vue. Pour cela, il est nécessaire de définir un certain nombre de conventions, qui n’ont pour seul objectif que de rendre le dialogue intelligible entre deux codes différents. Ces conventions décriront le comportement que l’on attend de chacun des codes dans les différentes situations d’utilisation couplée. Elles permettront la cohérence d’une implantation dissociée de l’algorithme de couplage.

Commençons par passer en revue les points qui *ne peuvent pas* être indépendants.

3.3 Quantités échangées, champs de vecteur

Chacun des codes se doit de publier un certain nombre de quantités à l’interface⁴ sous un format compréhensible soit par l’autre code, soit par un interprète intermédiaire. Une liste des quantités publiables est donc nécessaire.

Ceci implique aussi que les codes doivent pouvoir décrire la géométrie de l’interface. Un certain nombre de conventions devront être adoptées en conséquences : unités de mesure, points de référence des quantités envoyées. Ces deux points peuvent paraître anodins, mais sont problématiques s’ils n’ont pas été déterminés. Le code de mécanique des fluides MSD utilise par exemple les unités SI. Par contre, le code ZéBuLoN n’impose aucune unité ; il impose uniquement de rester cohérent entre les différentes données. Il y est d’ailleurs courant d’utiliser le millimètre comme unité de longueur (pour la géométrie et les déplacements) et par conséquent le mégapascal comme unité de pression.

Une autre habitude est de publier des résultats sous forme incrémentale, le code étant lui-même incrémental afin de traiter les non-linéarités. Il y a donc possibilité de confusion entre “température physique” (celle qu’on mesurerait avec un thermomètre) et “température relative”, telle que gérée par ZéBuLoN, c’est-à-dire l’écart à la température au début du calcul.

⁴Ici, on parle de l’interface entre physiques, qui peut par exemple être une interface surfacique ou volumique.

Dans un environnement de développement et de travail industriel, cette normalisation des quantités échangées, de leur représentation et de leurs traitements est une phase importante. Le groupe CGNS⁵ réfléchit actuellement à l'extension de leur norme pour pouvoir traiter des problèmes solides, afin de pouvoir s'étendre aux problèmes d'interactions fluide-structure.

Les quantités physiques échangées peuvent être vues comme des champs de vecteur sur la variété (surface ou volume) qu'est l'interface entre les domaines, champs qui dépendent éventuellement du temps. C'est donc au niveau de la représentation de ces champs que se poseront l'ensemble des problèmes :

- quelle est la représentation discrète des interfaces dans chacun des codes ;
- quelle est la représentation discrète des quantités dans chacun des codes ;
- comment gérer la non-coïncidence des deux représentations de l'interface commune ;
- comment gérer la non-coïncidence temporelle des deux problèmes.

Cette représentation des quantités sous forme de champs de vecteurs apporte un intérêt fondamental. Elle permet de pouvoir rendre génériques et donc identiques les différents traitements nécessaires à leur utilisation : descriptions, transferts, réinterpolations, utilisations comme conditions aux limites.

Nous verrons qu'en pratique, ces deux points (quantités et géométrie) sont les seules dépendances incontournables entre les deux codes. Nous allons maintenant détailler la manière de s'affranchir d'une dépendance des discrétisations spatiales et temporelles.

3.4 Assemblage d'un algorithme de couplage

Il ressort des différents algorithmes proposés dans la littérature, ainsi que de ceux que nous avons eu besoin de développer, qu'afin d'obtenir un algorithme efficace, c'est-à-dire à la fois stable, précis et rapide, il est important de tailler chaque algorithme sur mesure pour une application. Bien entendu, il n'est pas question de le développer en partant de rien. Au contraire, l'algorithme est un assemblage spécifique de briques élémentaires. Ces briques sont appelées *stratégies* en langage objet [30] ; nous utiliserons par la suite ce terme, dont le sens courant correspond lui aussi parfaitement à la situation présente.

Il est bien entendu impossible de donner ici une liste exhaustive de toutes ces stratégies, mais celles dont nous discuterons par la suite sont les suivantes :

- conditions de raccord multi-domaines et multi-physiques ;
- méthodes de transfert de champs entre maillages non-coïncidents ;
- méthodes de découplage temporel ;
- prédicteurs ;
- critères de convergence ;
- méthodes de corrections ;
- méthodes de recherche de points fixes.

L'*implantation* d'une stratégie correspond au choix d'une méthode particulière. Les stratégies admettent de nombreuses implantations, dont quelques exemples sont donnés sur la figure 3.3.

Si l'on veut pouvoir facilement adapter l'algorithme de couplage au problème traité, il est nécessaire de pouvoir passer d'une implantation à l'autre, au sens où cela ne doit être

⁵CFD General Notation System, groupe proposant ce type de représentation normalisée pour les calculs d'écoulements fluides. Cette normalisation permet au groupe de proposer aussi des bibliothèques de méthodes de traitement des données associées aux calculs numériques de mécanique des fluides.

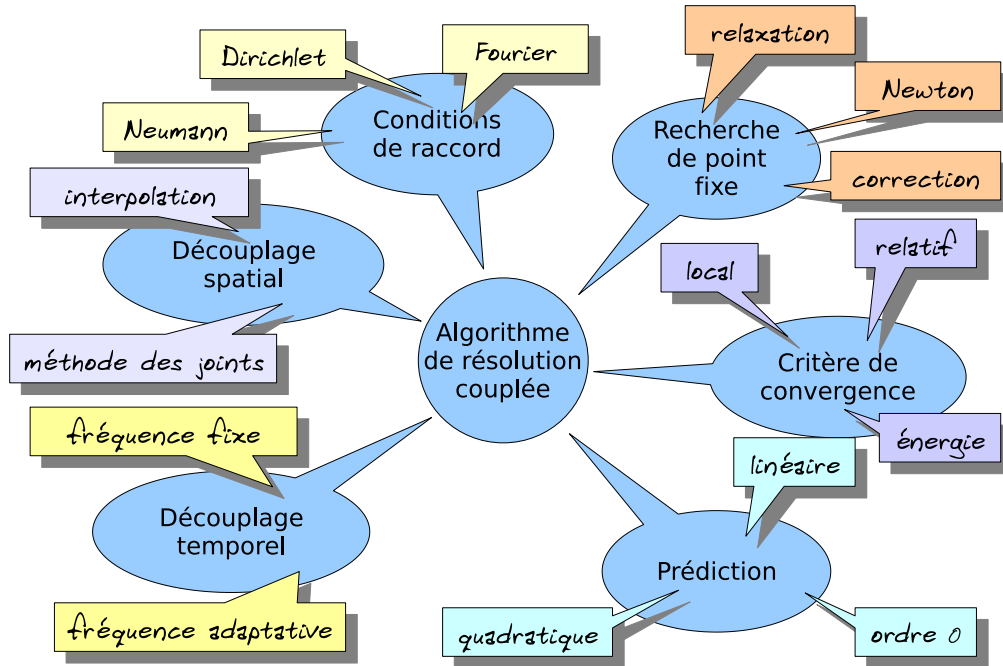


FIG. 3.3 – Quelques stratégies à assembler pour réaliser un algorithme couplé

qu’une question de mise en données. De même, l’implantation d’autres méthodes se doit aussi d’être facilitée, puisqu’il est totalement illusoire de penser pouvoir proposer aujourd’hui des méthodes adaptées à toute situation.

Nous avons mis en application ces préceptes au cas du couplage de ZéBuLoN. La conception objet de ce logiciel en a facilité l’intégration (cf. annexe C.2). Leur mise en application dans le logiciel MSD n’a par contre été réalisée que partiellement, MSD n’ayant pas été à l’origine développé dans cette optique “objet”.

La présentation des différentes implantations de chacune des stratégies serait ici fastidieuse. Celle-ci sera donc faite par la suite, à l’occasion de leur utilisation sur des cas bien précis⁶. Nous nous intéressons toutefois dès à présent à trois d’entre elles, les plus génériques : le découplage spatial, le découplage temporel ainsi que le transfert d’informations entre les codes.

3.5 Découplage spatial

L’indépendance spatiale, c’est-à-dire le fait d’autoriser chaque code à utiliser la discrétisation spatiale de son choix, est un concept indispensable. Il est en effet fréquent d’avoir des maillages du fluide beaucoup plus fins que les maillages du solide. De plus, le code MSD utilise des maillages réglés (donc des cellules hexaédriques dans le volume, et des quadrangles en surface), alors que le code ZéBuLoN utilise souvent des maillages en tétraèdres ou en prismes, plus faciles à construire, avec des triangles ou des quadrangles en surface.

⁶ Les conditions aux limites seront détaillées dans les sections 4.2, 8.1 et 8.5 ; les prédicteurs dans la section 8.2 ; les correcteurs et critères de convergence dans la section 8.3.

Les discrétisations sont d'ailleurs généralement de type différents : traditionnellement des volumes finis en mécanique des fluides et des éléments finis linéaires ou quadratiques en mécanique des structures. Dans le cas des volumes finis, les quantités sont connues au centre des éléments ou des facettes, et considérées constantes par élément ; dans le cas des éléments finis, les quantités sont connues aux nœuds du maillage, et interprétées via les fonctions de forme des éléments.

D'autres situations encore plus particulières sont parfois rencontrées : le département DADS de l'ONERA s'est par exemple intéressé au couplage aéro-mécanique dans le cas de voilures [36]. Dans cette application la structure est modélisée en "arêtes de poisson" (c'est-à-dire un modèle de l'aile comportant simplement quelques points représentatifs reliés par des éléments poutres), et le calcul des forces transmises par le fluide à l'aile doit impliquer l'intégration de ces forces sur des surfaces présentes uniquement de manière implicite dans ce maillage.

L'indépendance spatiale est entièrement liée à la question du transfert de champs entre surfaces (ou volumes) discrétisées de manières différentes. Ce transfert de champs est un travail complexe qui, dans un cas général, doit savoir gérer plusieurs points :

- la recherche de connectivité entre maillages qui, bien que coïncidant au niveau continu, ne le seront jamais parfaitement au niveau discret ;
- la "traduction" des valeurs du champ d'un maillage à l'autre ; nous avons vu que pour cela plusieurs méthodes existent : interpolation, méthodes des joints, ou méthodes intégrales par exemple.

Le problème de conservativité de ces méthodes de transferts est primordiale sur ce point.

Nous avons pris le parti de laisser cette question à la bibliothèque MpCCI [49]. Celle-ci se charge de détecter la connectivité entre les maillages décrits et d'en déduire les opérateurs de passage d'un domaine à l'autre. Elle utilise une méthode de recherche de connectivité basée sur les points les plus proches, puis transfère les champs à l'interface par interpolation. Des variantes sont disponibles, suivant que le transfert se fait entre quantités nodales ou par éléments. La documentation restant vague sur le sujet, il nous est difficile de savoir plus précisément quelles sont les méthodes de localisation spatiale et de réinterpolation utilisées.

Rappelons que la méthode de réinterpolation n'est pas parfaite. Théoriquement, elle introduit une erreur de transfert supplémentaire, qui pourrait être évitée en utilisant par exemple la méthode des joints (cf. section 2.6). Elle s'est cependant montrée satisfaisante dans les cas d'applications que nous avons envisagées.

Cette partie géométrique est identique en thermique et en mécanique, à ceci près qu'en mécanique, l'interface est potentiellement mobile et peut donc nécessiter une mise à jour des connectivités au cours du calcul. Il est donc tout à fait intéressant, comme l'a fait l'équipe développant MpCCI, de scinder la question du transfert de champ des autres problèmes liés au couplage. Cela permet, si nécessaire, de modifier cette méthode de transfert pour des applications précises⁷.

La publication de la géométrie consiste à décrire l'ensemble des nœuds et des éléments constituant l'interface. La bibliothèque MpCCI accepte la déclaration de maillages pour les éléments standards :

- triangles et quadrilatères linéaires et quadratiques en 2D ;
- tétraèdres, hexaèdres et prismes linéaires en 3D.

⁷En quelque sorte, la méthode de couplage est elle aussi partitionnée !

La déclaration du maillage est une phase amont du calcul, durant laquelle la bibliothèque calcule la connectivité entre les surfaces déclarées de part et d'autre. Tant que les maillages ne se déplacent pas l'un par rapport à l'autre, cette connectivité reste constante. Comme son calcul peut s'avérer très coûteux, il n'est effectué qu'une seule fois. En revanche, si les déplacements des maillages sont importants, que l'un des maillages glisse sur l'autre, ou que l'un des codes bénéficie de remaillage, il sera nécessaire de la recalculer en cours de calcul.

Notons tout de suite l'absence d'éléments volumiques quadratiques dans MpCCI, éléments qui sont pourtant fréquemment utilisés en mécanique du solide. Notons aussi qu'il est impossible de pouvoir gérer les maillages quelconques de CÈDRE⁸ : la bibliothèque de couplage doit connaître tous les éléments afin de pouvoir réinterpoler. Il est ainsi nécessaire de contourner ces deux problèmes ; dans le cas de ZéBuLoN nous avons ainsi réalisé le transfert via un volume intermédiaire maillé linéairement. Ceci a pour défaut d'appauvrir la description des champs échangés, et d'introduire donc une erreur supplémentaire dont on aurait pu se passer. Dans le cas de CÈDRE, une triangulation a priori de la surface couplée pourra résoudre le problème. Ces deux limitations montrent une fois de plus l'impossibilité de prévoir en amont l'ensemble des stratégies qui seront nécessaires pour mener à bien un calcul, qu'il soit couplé ou non. Nous avons été contraints de développer des méthodes permettant de contourner les limitations, alors qu'il aurait été beaucoup plus naturel, et probablement plus rapide et efficace de pouvoir enrichir la bibliothèque des capacités manquantes.

3.6 Découplage temporel

Contrairement aux habitudes, il n'est pas absolument nécessaire de réaliser des discrétisations temporelles identiques, ou compatibles entre le fluide et la structure. D'ailleurs, une contrainte équivalente sur les discrétisations spatiales n'avait jamais été envisagée. Pourquoi donc se l'imposer temporellement ?

Dans un calcul tel que celui de la tuyère, les non-linéarités sont importantes lors des phases de chauffe et lorsque les tubes entrent en contact ; elles sont plus "simples" lors des phases de fluage (longues périodes de maintien de pression). Si l'on choisissait d'imposer un pas de temps de couplage fixe, celui-ci serait choisi petit à cause des phases de déformations rapides et donc beaucoup plus fin que nécessaire pendant la majeure partie du calcul.

Il faut aussi avoir en tête que les difficultés de convergence du calcul solide ne seront pas nécessairement visibles par le fluide : le fluide n'est couplé qu'au travers du champ de déplacement et de contraintes (ou, dans le cas thermique, des champs de température et de flux de chaleur), alors que les non-linéarités pourront par exemple être dues à la forte présence de déformations plastiques.

Cette idée de s'affranchir d'une coordination temporelle apparaît en fait très tôt, par exemple dans les papiers de Piperno et al. [61]. Les auteurs partent de la constatation qu'il est illusoire et surtout inutile d'envisager des discrétisations identiques, puisqu'une discrétisation temporelle d'un problème fluide est généralement 100 fois, voire 1000 fois plus fine que celle d'un problème solide. Ils proposent en conséquence la solution dite du "sous-cyclage", qui consiste à réaliser un couplage entre fluide et structure à chaque pas de temps structure, mais tous les 100 (ou 1000) pas de temps fluide. Dans cette approche, les deux codes restent

⁸CÈDRE est un logiciel de mécanique des fluides, développé à l'ONERA, successeur de MSD. Une des différences principales est sa capacité à traiter des maillages non structurés, dont les éléments peuvent être des polygones (en 2D) ou des polyèdres (en 3D) quelconques.

néanmoins synchrones, et c'est de cela dont nous voulons ici nous affranchir complètement. Il faut aussi noter que ces approches concernent des problèmes dynamiques, et ne s'appliquent pas telles quelles aux questions qui sont les nôtres.

Un deuxième point dont nous chercherons à nous affranchir est la fréquence fixe des instants de couplage. En effet, lors de calculs de mécanique non-linéaire, il est illusoire de demander au code structure des pas de temps fixes et réguliers : ceux-ci sont calculés de manière adaptative en fonction des non-linéarités vues par la structure à l'instant considéré. Il nous est apparu naturel de chercher à mimer ce comportement, classique en mécanique des structures, et de l'appliquer au problème non-linéaire qu'est le problème couplé.

3.6.1 Pile de réception et système du *serveur*

Ainsi, pour s'affranchir de cette synchronisation des codes entre eux, nous proposons quelques modifications des habitudes oneriennes de couplage. Ces modifications se basent sur trois constatations.

Tout d'abord, il est important que les codes puissent envoyer des données quand bon leur semble, cet envoi étant par exemple décidé en fonction de certains critères d'évolution des champs à l'interface. Réciproquement, il est aussi nécessaire que les codes soient prêts à recevoir des données en permanence, afin qu'envois et réceptions soient réalisables séparément. Enfin, nous devons être conscients que le code récepteur a besoin des champs à l'interface à des instants où le code émetteur ne les calcule pas forcément.

Il faut par conséquent, qu'en plus des données, le code envoie l'*instant* auquel ce champ est valide. Le code récepteur peut ainsi faire les corrections et adaptations nécessaires pour réinterpoler les champs reçus aux instants où *lui* en a besoin. Le choix de la méthode de réinterpolation temporelle des champs fait ainsi partie des stratégies mises en œuvre dans l'algorithme de résolution couplée : ce seront par exemple des interpolations ou des extrapolations constantes, linéaires ou quadratiques. Nous verrons aussi par la suite que cette *date d'existence* des champs permettra une certaine liberté dans la réalisation de variantes des algorithmes de couplage, en particulier dans nos applications quasi-statiques.

Les données étant maintenant reçues de manière irrégulière, nous utilisons une pile de stockage des champs reçus, afin de pouvoir les utiliser aux moments nécessaires. Cette pile répond aussi à un certain nombre de besoins : elle gère tout d'abord le stock de données à l'interface. Chaque élément de ce stock est de taille relativement faible⁹ dans le cas de couplages surfaciques, mais peut devenir important dans le cas de couplages volumiques. Si la fréquence de couplage est élevée, leur nombre peut aussi devenir conséquent. Ainsi, la pile vérifie à intervalles réguliers la pertinence des champs reçus précédemment afin d'éventuellement libérer de l'espace mémoire.

Notons pour finir que nous utilisons une pile de stockage par quantité couplée. Cela rend possible, dans le cas de couplage multiple (tel que le couplage aéro-thermo-mécanique), l'utilisation de fréquences de couplage différentes pour chacun des phénomènes, ainsi que de stratégies différentes pour chacune des réinterpolations temporelles.

Afin de faciliter l'accès à cette pile de données, un *serveur* s'occupe d'une part du rangement des données dans la pile lorsqu'elles sont reçues, et d'autre part de la recherche des données et de leur éventuelle interpolation.

⁹Taille faible par rapport à celle du problème complet

3.7 Moteur de couplage

Le transfert d'information entre les différents codes est bien entendu indispensable, et diverses solutions sont envisageables. Un certain nombre de services de communications vers les codes externes sont ainsi nécessaires pour mettre en place une méthode de couplage partitionné, et ce sont ces *services* que l'on regroupe sous le terme *moteur de couplage*. Nous décrivons ici un moteur qui fournit le strict minimum pour réaliser le couplage ; le nombre de services nécessaires est finalement assez restreint. Ce moteur permet aussi de faire clairement la distinction entre transfert de données (question purement informatique), et les réinterpolations spatiales et temporelles (questions d'analyse numérique). Le concept de moteur a été mis en place dans ZéBuLoN ; c'est donc dans ce cadre, un peu plus concret, que nous allons le décrire.

3.7.1 Le moteur de couplage dans ZéBuLoN

L'unique objectif du moteur de couplage est de fournir des moyens de communication de données aux différents codes impliqués dans le calcul partitionné. À cette fin, seules trois fonctions sont réellement indispensables :

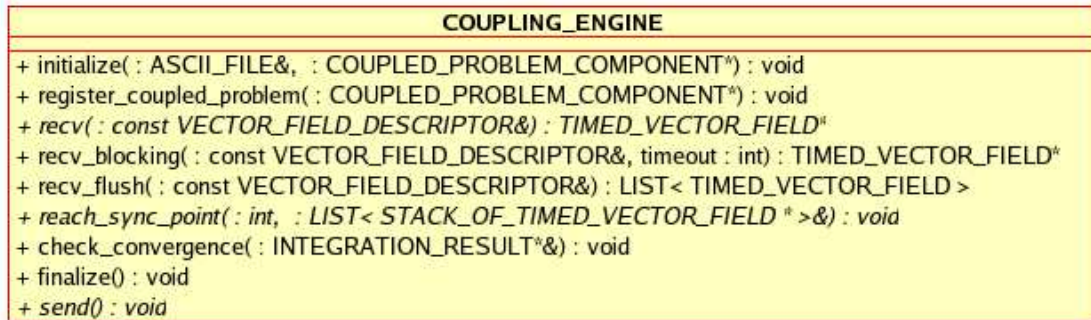
- l'envoi d'un champ de vecteur ;
- la réception d'un champ de vecteur ;
- l'échange de méta-données, telles que le statut du calcul.

Cette classe, résumée dans le diagramme 3.4, propose donc :

- une méthode de lecture du fichier de données (l'habitude dans ZéBuLoN est de la nommer `initialize`) ;
- une méthode d'initialisation (nommée ici `register_coupled_problem`) pouvant servir entre autre à mettre en place le système de messagerie ;
- une méthode d'envoi de données. Celle-ci doit être non-bloquante, au sens où elle n'a pas besoin d'attendre que le destinataire ait reçu le message pour se terminer ;
- une méthode de réception non-bloquante ;
- à partir de cette méthode de réception non-bloquante, il est aisé de dériver d'autres méthodes de réceptions : bloquante et recevant tous les messages en attente ;
- une méthode de vérification de convergence entre les différents codes de calculs, dont on discutera ci-dessous ;
- enfin, une méthode annonçant la fin du couplage pour ce code, afin d'en avertir les autres codes.

L'implantation de cet objet *moteur de couplage* peut se faire de diverses manières. La plus simple, si son développement se fait *ex nihilo*, est de réaliser les échanges de données par fichiers : l'envoi correspond à l'écriture d'un fichier ; la réception à sa lecture. L'inconvénient majeur de cette solution est l'impossibilité (ou plus précisément la grande difficulté) de réaliser des calculs sur plusieurs machines avec cette solution. Un deuxième inconvénient est la lenteur engendrée par ces écritures de fichiers. C'est toutefois une solution qui a le mérite de ne dépendre d'aucune librairie externe. C'est pourquoi nous envisageons de l'utiliser dans le cadre de la base de test. Son développement est presque réalisé, le format de fichier d'échange retenu étant le format VTK¹⁰. C'est la solution idéale dans une phase de développement,

¹⁰VTK est une librairie de visualisation graphique [67]. Son format de fichier est parfaitement adapté pour les champs issus de codes éléments finis. S'y conformer permet la visualisation de ces champs dans des logiciels tels que Paraview [54].

FIG. 3.4 – Diagramme simplifié de la classe *moteur de couplage*

durant laquelle les performances ne sont pas le point primordial.

Une deuxième stratégie pourrait consister à un passage d'informations directement par adresses ; elle n'est aussi utilisable que si les différents codes sont lancés sur une unique machine.

Une troisième stratégie consiste enfin à utiliser des bibliothèques de transfert de données, telles que MPI ou PVM. Cette solution permet des calculs multi-plateformes.

Dans le cadre de cette thèse, nous avons utilisé une quatrième solution, basée sur la bibliothèque MpCCI.

3.7.2 Le moteur MpCCI

La classe *moteur de couplage* présentée ci-dessus est une classe *abstraite*. Elle a donc été dérivée pour bénéficier de la bibliothèque MpCCI.

Cette bibliothèque est à l'origine une sur-couche de la bibliothèque de messagerie MPI, sur-couche ajoutant les méthodes de réinterpolation, ainsi que la recherche de connectivités multiprocesseurs, dans le cas de calculs parallèles. Ainsi, le moteur de couplage MpCCI regroupe en fait deux points que nous avons séparés dans notre discussion : le transfert de données et le découplage spatial.

Par rapport au canevas minimal présenté dans la figure 3.4, ce moteur nécessite donc quelques méthodes supplémentaires. Il s'agit principalement de méthodes nécessaires à l'initialisation du calcul couplé :

- déclaration géométrique des partitions couplées ;
- déclaration des quantités couplées ;
- une méthode redéfinissant les communicateurs entre codes¹¹.

Le moteur propose aussi une méthode relative au couplage par points de synchronisation. Cette méthode, bien que redondante avec celles d'envoi/réception, est celle utilisée historiquement dans MSD. Elle a donc été conservée par souci de compatibilité, bien qu'étant spécifique à MpCCI.

¹¹Dans la version 3.0.3 de MpCCI, version qui a été utilisée pendant cette thèse, et dans le cas du couplage à trois codes ou plus, les communicateurs par défaut ne sont pas utilisables. Cette limitation a semble-t-il été levée dans la version 3.0.5.

3.7.3 Mise en place d’algorithmes de couplage

La bibliothèque MpCCI apporte des solutions à de nombreuses questions posées par les problèmes couplés. Outre les questions de transfert de données mono- ou multi-processeurs, et de réinterpolations de champs sur maillages non-coïncidents, elle propose aussi des solutions rudimentaires d’implantations d’algorithmes de résolution. En effet, par le positionnement de *points de synchronisation* en divers endroits des différents codes, il est en théorie possible de réaliser le couplage simplement au travers d’une mise en donnée faisant simplement correspondre ces points pré-placés.

Cependant, autant le gain de cette bibliothèque sur les deux premiers points est considérable, autant la mise en place d’algorithmes complexes de couplage s’est avérée impossible, ou du moins extrêmement alambiquée, si elle est faite grâce à ces points. C’est pourquoi nous avons renoncé à cette méthode pour la mise en place d’algorithmes, pour pouvoir gagner en souplesse et en liberté, en utilisant de simples envois/réceptions de messages.

Contrairement aux travaux précédents, la mise en place de l’algorithme de résolution couplée ne se fera donc plus au travers de ces points de synchronisation, et n’apparaît pas dans ce moteur de couplage.

3.7.4 Vérification globale de la convergence du problème

Dans son utilisation découplée, le code ZêBuLoN utilise une classe *résultat de l’intégration* (INTEGRATION_RESULT), afin de décrire son statut de convergence. Une valeur nulle (pointeur nul) indique un calcul convergé ; toute autre valeur indique un problème, dont les sources peuvent être multiples. Les deux causes d’erreurs les plus fréquentes sont un problème d’intégration de la loi de comportement et le nombre maximal d’itérations dépassé dans la méthode de Newton-Raphson. Bien évidemment, dans le cas de calculs linéaires élastiques, ces problèmes ne peuvent pas apparaître. L’intérêt de cette classe est d’identifier l’erreur afin de pouvoir la traiter. La diminution du pas de temps est par exemple une solution très souvent efficace pour résoudre ces deux problèmes d’intégration. Dans le cas de calculs couplés, le code peut se trouver dans un nouvel état : l’intégration du problème est réussie, mais pour des raisons spécifiques au calcul couplé, le code ne s’estime pas convergé. C’est le cas par exemple si les critères de convergence couplés ne sont pas satisfaits. Ce nouveau statut nécessite de recalculer l’incrément courant, après avoir reçu des valeurs corrigées du code fluide. À ce titre, il a été nommé IR_OK_BUT_DO_IT_AGAIN.

De son côté, le code MSD peut se trouver dans deux états : calcul convergé, ou calcul divergé. Nous aurions pu envisager d’autres stratégies de couplages, basées sur des échanges d’états intermédiaires, et introduisant des états supplémentaires. Il est donc nécessaire de ne pas se restreindre au niveau de ces statuts.

L’état global du système couplé est naturellement défini comme étant celui du code le plus pessimiste. Il est par conséquent indispensable de rendre comparables les statuts des différents codes. Il est aussi intéressant d’ordonner les différents états possibles, afin d’en rendre le bilan plus simple.

À cette fin, la bibliothèque MpCCI propose quatre états possibles, que l’on peut classer par ordre croissant :

1. convergé
2. non convergé, et nécessite encore des échanges

3. convergé, mais impossible de continuer le calcul
4. divergé

Le code MSD peut donc se trouver dans l'un des états **1** ou **4**. Le code ZéBuLoN peut lui se trouver dans l'état **1** (cas d'une intégration réussie), **2** (cas d'un critère couplé non satisfait) ou **4** (cas de divergence).

Il ne nous est pas encore apparu nécessaire de définir d'autres états possibles ; ceci n'est pour l'instant pas exclu, s'il s'avère intéressant d'en utiliser d'autres. Il est aussi à noter que l'état **3** n'est en pratique pas utilisé, puisqu'aucun des codes dont nous disposons ne peut s'y trouver.

La méthode de *vérification globale de convergence* (`check_convergence`) du moteur de couplage permet donc aux différents codes, à certains instants du calcul, de se mettre d'accord sur l'état global du système couplé. À l'instar de ZéBuLoN qui propose diverses solutions pour gérer la mauvaise convergence de problèmes mécaniques non-linéaires, il devient possible, connaissant l'état global du système couplé, de proposer diverses solutions dans le cas de difficultés dues au couplage.

Chapitre 4

Développements préliminaires

Nous discutons ici d'une partie des développements qui ont été nécessaires à la réalisation d'une plateforme couplée. Il s'agit du développement d'une méthode de déplacement de maillage dans le code fluide, qui a fait l'objet de deux rapports techniques [19, 20], et du développement et de la validation d'une condition au bord couplée en pression.

4.1 Maillage mobile

4.1.1 Introduction

Les codes de mécanique des fluides (par exemple MSD ou CÈDRE) résolvent les équations de Navier-Stokes. Celles-ci sont discrétisées par la méthode des volumes finis, qui nécessite le découpage de l'espace de calcul en un maillage (structuré dans le cas de MSD, quelconque dans le cas de CÈDRE).

Lors de calculs sur des géométries fixes, les équations sont écrites sous forme Eulérienne (c'est-à-dire relativement à un domaine de référence fixé). Par contre, lorsque le domaine où a lieu l'écoulement est mobile (dans le cas de calculs couplés fluide-structure par exemple) elles sont écrites sous forme ALE (Arbitraire Lagrangienne Eulérienne), une forme à mi-chemin entre l'écriture Lagrangienne (relative à un domaine attaché aux particules en mouvement) et Eulérienne.

Notons au passage que la résolution des équations sous forme ALE doit être réalisée avec prudence. Outre les corrections de flux nécessaires pour tenir compte des déplacements des facettes du maillage, il est important que le schéma numérique respecte la *loi de conservation géométrique (GCL)* [21]. Cette loi stipule que le schéma numérique doit représenter exactement les solutions constantes sur un maillage mobile¹. Le non respect de cette loi entraîne l'apparition d'un terme d'erreur supplémentaire, pouvant déstabiliser un schéma, par ailleurs stable sur configurations fixes. Dans MSD, le calcul des vitesses de facettes est réalisé de manière à satisfaire cette loi ; nous en bénéficierons donc pour le couplage.

La formulation ALE des équations est déjà présente dans MSD, mais elle nécessite la connaissance complète des déplacements de chaque nœud du maillage. Or, dans un calcul couplé, la déformation du domaine de calcul n'est connue que sur sa frontière, en contact

¹D'autres formulations de cette loi sont possibles : Le Tallec et Mouro [43] la relie par exemple à la présence de fonctions localement constantes dans un certain espace de fonctions tests, permettant de satisfaire l'équation de conservation de la masse exactement au niveau discret, pour les écoulements uniformes.

avec une structure déformable qui impose ses déplacements. Il ne nous manque donc que la propagation de cette déformation à l'intérieur du domaine de calcul.

Ce type de problème est bien connu des mécaniciens des structures : il s'agit simplement de calculer les déformations d'une pièce, connaissant celles de sa frontière (ou d'une partie de sa frontière). Autrement dit il s'agit de résoudre un problème de Dirichlet. Les différentes méthodes de déformation de maillages que nous présenterons ci-dessous partagent toutes cette même idée, à savoir que le *maillage* du fluide est traité comme une structure déformable.

Il est aussi important de noter que l'on s'occupe ici de déformation de maillage, et non de remaillage. Ceci signifie en particulier que la topologie du domaine doit rester la même², et que la topologie de maillage est inchangée³. Seules les coordonnées des nœuds sont autorisées à être modifiées.

Les contraintes que l'on s'impose n'ont par contre pas besoin d'être réalistes d'un point de vue mécanique. Nous ne cherchons pas la solution d'un "vrai" problème mécanique, mais plutôt un maillage déformé qui garde ses bonnes propriétés (cohérence, qualité, finesse, couches limites, etc...) lors de son déplacement.

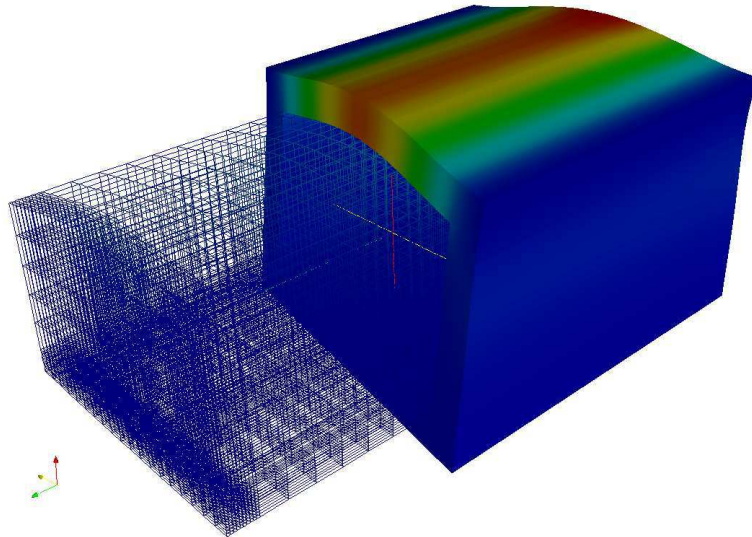


FIG. 4.1 – Gonflement d'un tube sous l'effet de la pression exercée par le fluide, et maillage fluide déformé en conséquence

La figure 4.1 présente un tel cas de calcul couplé fluide-structure (couplage mécanique) : il s'agit de l'écoulement d'un gaz dans un tube déformable. Le tube (un matériau élastique dans cet exemple) fixé à sa base, subit la pression du fluide à l'intérieur et est laissé libre sur ses autres faces. Le fluide (gaz parfait) s'écoule sous pression d'arrêt imposée en entrée et en sortie. Le long de l'interface couplée, il suit les déformations de la structure.

Rappelons brièvement l'un des algorithmes de couplage faible utilisé en interactions fluide-structure. Le code aéro et le code de structure échangent des informations le long de leur zone de contact : le code aéro envoie des pressions et reçoit des déplacements (cf. Fig. 2.1, page 24).

Après chaque réception de position de frontière actualisée, le travail du déformeur de maillage consiste à calculer des déplacements raisonnables pour les nœuds internes. Le module

²On interdit par exemple l'apparition/disparition de "trous" dans le domaine.

³On n'ajoute pas de nouveaux nœuds ou éléments au maillage.

de déformation sera donc appelé fréquemment, à chaque échange. Son coût de calcul (tant en temps CPU qu'en taille mémoire) devra donc être raisonnable par rapport au coût de calcul du fluide.

4.1.2 Comparaison de trois méthodes

Critères de qualité

Afin de comparer différentes méthodes de déformation de maillage, il nous faut tout d'abord préciser quelles (bonnes) propriétés on attend du déformeur de maillage.

On demande tout d'abord que dans des cas où la déformation de la frontière est une isométrie (rotation ou translation), l'ensemble du maillage suive la même isométrie (autrement dit, les mouvements de corps solides sont bien représentés). Dans le cas de dilatations, on désire aussi avoir cette même propriété.

Bien entendu, on demande que les déformations n'altèrent pas la validité du maillage : elles ne modifient pas la topologie, n'engendrent pas de retournement de mailles.

Les maillages que l'on utilise sont généralement resserrés à proximité des parois, afin de bien représenter les couches limites. Les déformations doivent conserver cette propriété.

La figure 4.2 illustre ces différentes propriétés.

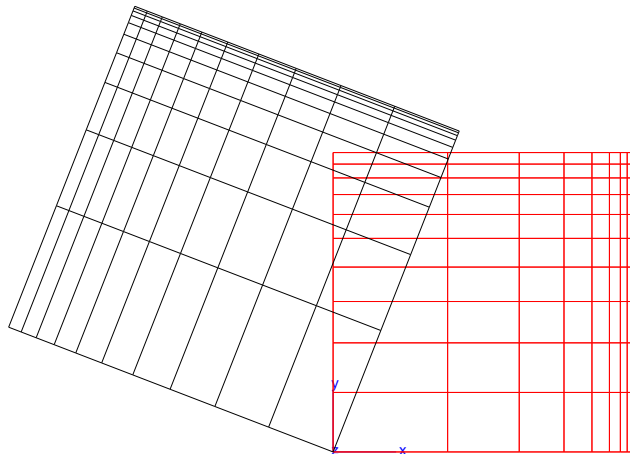


FIG. 4.2 – Conservation des propriétés du maillage lors d'une déformation par similitude (ici composée d'une rotation et d'une dilatation) : le déplacement du bord est imposé et celui des nœuds internes calculé. Le maillage de couche limite du bord de droite est conservé.

Méthode harmonique (ou du Laplacien)

Cette méthode est basée sur le Laplacien ; les équations régissant ce problème sont :

$$\Delta u_i = 0 \quad \text{dans } \Omega_F$$

$$u_i \text{ connu sur (une partie de) la frontière}$$

où u_i ($i = 1, 2, 3$) est le déplacement dans la direction i , et Ω_F est le domaine fluide.

Les solutions de ce problème sont harmoniques, et vérifient donc de bonnes propriétés : propriété de la moyenne (chaque nœud se déplace de la moyenne⁴ de ses voisins) et principe

⁴au niveau discret cette moyenne est bien entendu à interpréter prudemment

du maximum (un nœud interne ne se déplace pas plus que le maximum des déplacements de la frontière).

La méthode a aussi l'avantage de découpler les directions (les déplacements suivants x , y et z sont traités séparément), ce qui réduit d'un facteur 9 la taille en mémoire du problème, et donc aussi les coûts de résolution, par rapport à une méthode couplant les directions. De plus, dans certains cas, la matrice globale du problème est identique pour les 3 directions, et n'a donc besoin d'être factorisée qu'une seule fois.

Pour résoudre ce problème à l'aide d'un code éléments finis usuel, nous devons avoir à disposition un maillage ne contenant que des éléments de type tétraèdres, hexaèdres ou prismes (ce qui est le cas de MSD, mais pas forcément de CÈDRE). L'utilisation de cette méthode dans CÈDRE nécessitera une phase de pré-traitement du maillage, consistant par exemple à découper toutes les cellules complexes en tétraèdres.

Cette méthode satisfait les critères de qualité fixés : elle répond parfaitement aux déplacements simples (translations, rotations, dilatations) et conserve les couches limites.

Méthode élastique

L'idée de cette méthode est de dire que le maillage fluide est un matériau élastique, et est régi par les équations de l'élasticité :

$$\begin{aligned} \operatorname{div} \boldsymbol{\sigma} &= \mathbf{0} \\ \boldsymbol{\sigma} &= \underline{\underline{\mathbf{K}}} : \boldsymbol{\xi} \\ \boldsymbol{\xi} &= \frac{1}{2}(\operatorname{grad} \mathbf{u} + (\operatorname{grad} \mathbf{u})^T) \\ \mathbf{u} &\text{ connu sur une partie de la frontière} \end{aligned}$$

où \mathbf{u} est le vecteur déplacement, $\underline{\underline{\mathbf{K}}}$ une matrice de rigidité (dépendant dans le cas de l'élasticité isotrope d'un module d'élasticité et d'un coefficient de Poisson), $\boldsymbol{\xi}$ et $\boldsymbol{\sigma}$ sont respectivement les tenseurs des déformations et des contraintes.

Sur les cas testés, cette méthode présente des propriétés similaires à celles du Laplacien. Par contre, elle couple les directions. Il en résulte que la matrice globale est 9 fois plus remplie. Les temps de calculs sont donc bien plus importants. Nous verrons toutefois dans la section 4.1.3, que sur un le cas test de la languette, cette méthode fournit des résultats légèrement meilleurs que les deux autres.

Méthode par ressorts 1-D

Dans cette méthode, le domaine fluide n'est plus vu comme un domaine continu. Cette fois le maillage fluide est vu comme un treillis de ressorts, à l'équilibre dans sa position initiale. Pour être plus précis, chaque arête du maillage est considérée comme un triplet de ressorts unidimensionnels.

Ces triplets de ressorts 1-D exercent sur leurs deux extrémités la force suivante :

$$\mathbf{F}_A = \frac{1}{\|AB\|}(\mathbf{u}_B - \mathbf{u}_A) \quad \text{et} \quad \mathbf{F}_B = -\mathbf{F}_A = \frac{1}{\|AB\|}(\mathbf{u}_A - \mathbf{u}_B) \quad (4.1)$$

où A et B désignent les deux extrémités, \mathbf{u}_A et \mathbf{u}_B leurs déplacements (Fig. 4.3).

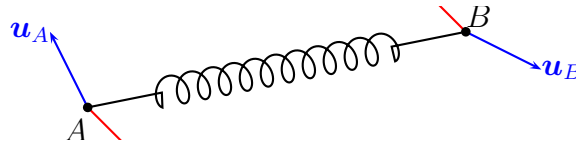


FIG. 4.3 – Modèle du triplet de ressorts 1-D

Il est éventuellement possible d’enrichir cette méthode avec une rigidité variable k pour chaque ressort, afin de rendre moins déformable certaines zones du maillage.

Le choix de force ci-dessus a été fait pour sa simplicité (les trois directions sont découplées) et sa linéarité (si la distance $\|AB\|$ est évaluée dans la configuration initiale), et non pour une quelconque ressemblance avec une situation physique. Malgré le nom de *ressort* que l’on a donné à cette méthode, la force de rappel n’est pas celle qu’appliquerait un “vrai” ressort.

Notons par ailleurs que dans le cas d’un maillage structuré régulier, il est possible de montrer que cette méthode est équivalente à celle du Laplacien [19].

Tout comme le Laplacien, la méthode par ressorts découple les directions, et présente de bonnes propriétés : les translations, rotations et dilatations sont bien représentées ; les couches limites sont préservées. La figure 4.4 présente un exemple de déformation d’une paroi d’un carré.

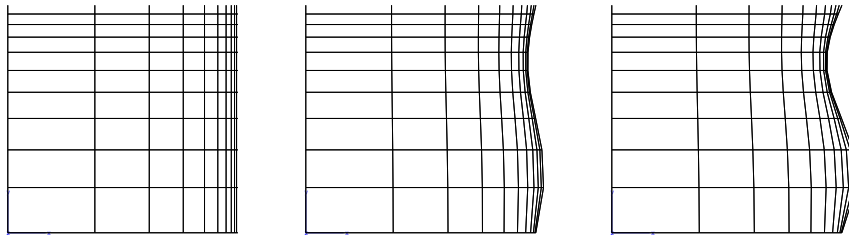


FIG. 4.4 – Déformation d’un maillage par la méthode des ressorts 1-D : préservation des couches limites. On impose un déplacement sinusoïdal à la frontière de droite. Le maillage est présenté dans sa configuration initiale, après 0,5 s et 1 s.

Une exception toutefois : dans le cas d’un maillage non structuré (par exemple un maillage en triangles) et pour des rotations, nous avons constaté l’accumulation des nœuds internes vers le bord (Fig. 4.5), un peu comme sous l’effet d’une force d’inertie. Pourtant aucune force d’inertie n’est présente dans ce calcul. Ce phénomène n’est pas présent dans les deux autres méthodes.

Le mauvais comportement des ressorts face aux rotations s’explique par le fait que la loi choisie pour ces ressorts les incite à revenir dans leur direction et longueur initiale (position d’énergie minimale). Les déplacements de type rotation augmentent donc l’énergie du système. En fait, c’est le bon comportement de cette méthode face aux maillages réglés qui est surprenante ; dans ce cas, il se trouve que, par chance, les forces de tension se compensent exactement.

Il serait possible de remplacer l’équation 4.1 par une autre équation ne créant pas de tension pour les mouvements de rotation (il suffit par exemple d’y introduire un terme faisant intervenir la variation de longueur du ressort). Cependant, cette solution couplera les 3 directions d’espace, et deviendra d’une lourdeur comparable à la méthode élastique. On ne l’a donc

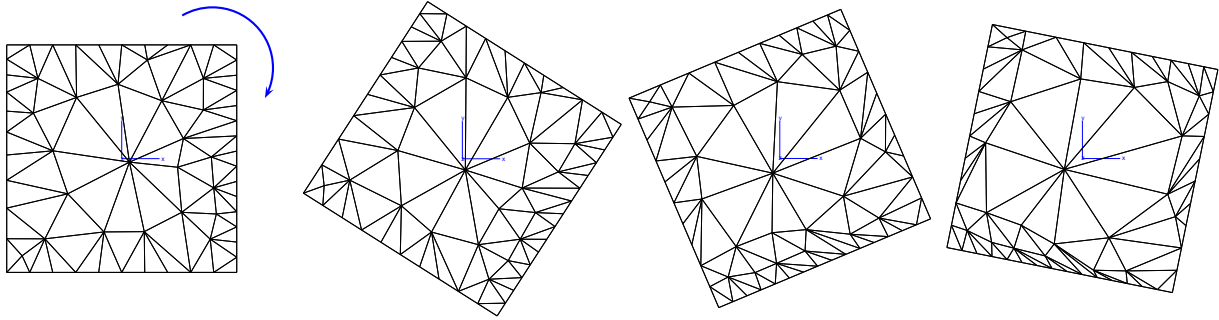


FIG. 4.5 – Ressorts 1-D : problème d’accumulation des nœuds vers le bord. On impose un mouvement de rotation à l’ensemble de la frontière. Les nœuds internes sont “centrifugés” vers le bord.

pas retenue, puisque des solutions plus légères et élégantes sont envisageables, par exemple en réalisant en amont l’isolement du mouvement de corps rigide du domaine [28, 41].

Les applications envisagées dans cette thèse ne présentant de toutes façons pas ce genre de mouvements, nous avons décidé de ne pas plus poursuivre ce type d’améliorations. La méthode est parfaitement satisfaisante dans son état actuel.

La simplicité du maillage de treillis permet à cette méthode d’être utilisable à la fois dans MSD (maillages réglés) et dans CÈDRE (maillages quelconques), sans phase de redécoupage du maillage.

4.1.3 Quelques cas tests

Ce module de déformation de maillage donne des résultats parfaitement satisfaisant sur des cas tels que celui présenté sur la figure 4.1. Il faut toutefois noter que la géométrie y est particulièrement simple (le maillage est un parallélépipède, parfois ponctué d’obstacles).

Le cas réputé difficile de la languette dans un écoulement a aussi été testé. Comme nous ne testons que le meilleur, le couplage est à sens unique : la structure impose ses déplacements au fluide, et le déplacement de la languette est imposé directement dans le code structure.

Les résultats présentés dans la figure 4.6 ont été obtenus par la méthode “élastique” (module d’Young $E = 200\text{ GPa}$, ratio de Poisson $\nu = 0,3$). On impose au bout de la languette un déplacement horizontal $u_1 = t/2$, pour t variant entre 0 et 1, et on applique ce déplacement en 5 étapes. La première déformée (correspondant donc à un déplacement de $0,1\text{ mm}$) est présentée sur la figure. La troisième déformée est encore admissible, bien que les cellules en bas à droite de la languette soient presque dégénérées. À partir du quatrième incrément, ces cellules se sont retournées (Fig. 4.7).

Le constat est un peu meilleur pour un matériau quasi-incompressible (ratio de Poisson $\nu = 0,49$) : le troisième incrément est encore correct et c’est au cinquième que les cellules se retournent. De même, si le ratio de Poisson est choisi nul, le maillage devient incorrect dès le troisième incrément.

La méthode harmonique donne des résultats similaires à celle élastique, pour un ratio de Poisson nul, c’est-à-dire produit un résultat incorrect à partir d’un déplacement supérieur à $0,3$.

La méthode des ressorts donne des résultats semblables à celle du Laplacien ; elle réussit toutefois à être encore correcte au troisième incrément, les croisements de mailles n’apparais-

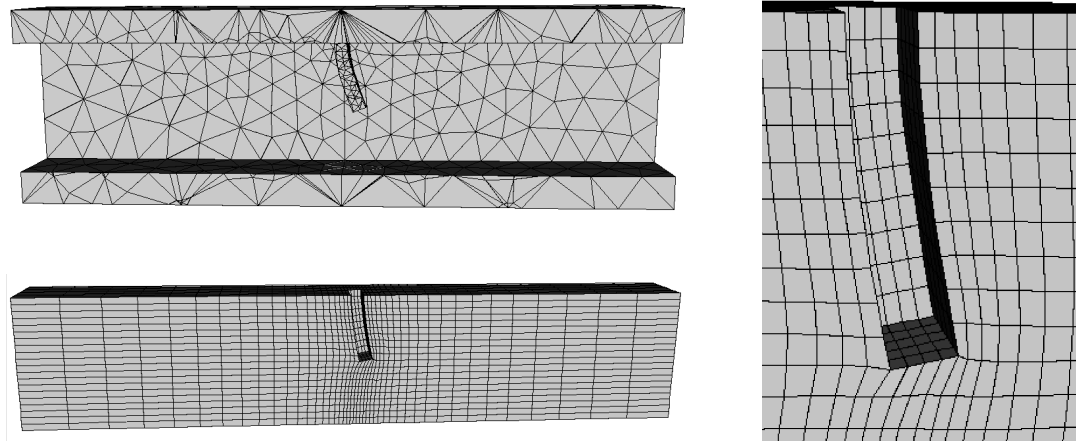


FIG. 4.6 – Cas test de la languette. En haut à gauche : coupe du domaine structure dans sa position déformée. En bas et à droite : déformations calculées pour le maillage fluide.

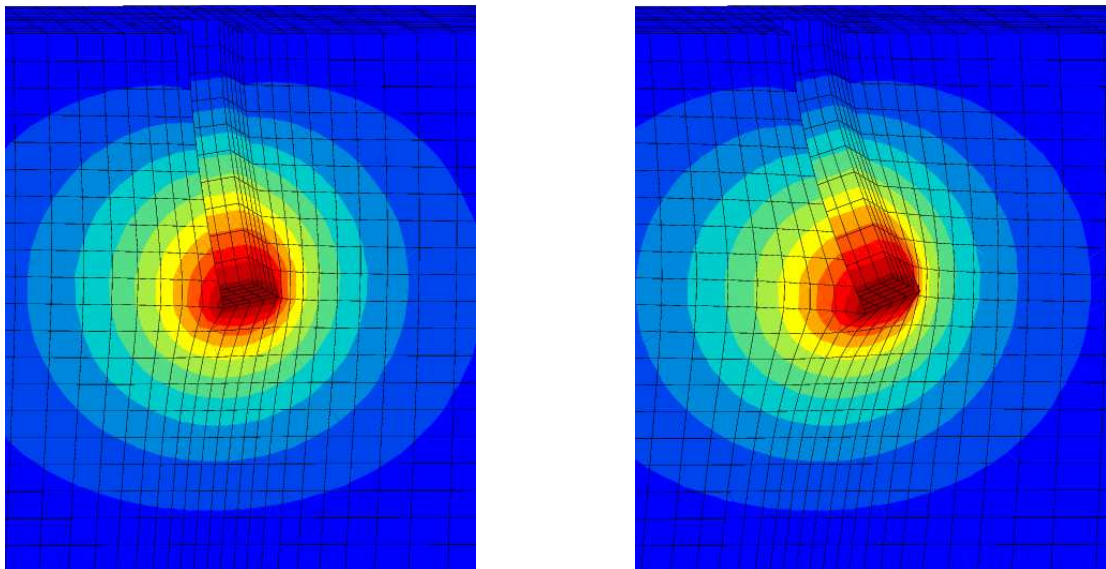


FIG. 4.7 – Cas test de la languette. À gauche : après 0,5 s ; à droite : après 1 s. Sur cette dernière figure, certaines mailles se sont retournées et le maillage est devenu incorrect. Les couleurs représentent l'intensité du déplacement horizontal.

sant qu'au quatrième.

Nous avons aussi utilisé ce module sur une géométrie plus complexe de propulseur (Fig. 4.8), fournie par Snecma Propulsion Solide (SPS, groupe Safran). La déformation du maillage a été réalisée avec la méthode des ressorts.

Un certain nombre de particularités sont apparues sur cet exemple, nécessitant entre autre l'enrichissement des conditions au bord : glissement de nœuds le long d'un axe et conditions distinctes suivant la direction considérée.

Ce cas nécessitera encore quelques vérifications, dont celles de la mise en données du couplage : il semble en effet que certaines zones s'interpénètrent, ou subissent des déplacements plus importants que ce que l'on admet ici.

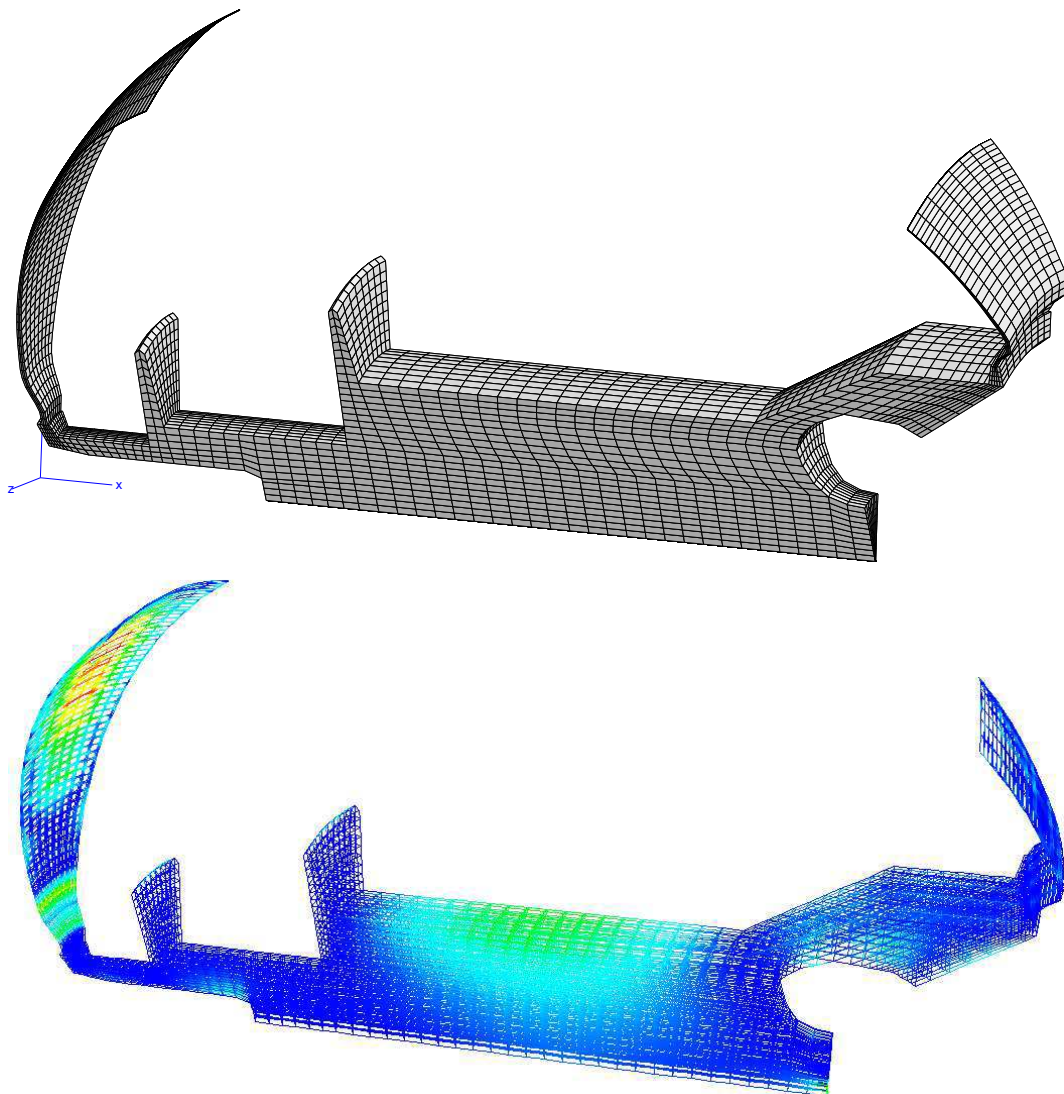


FIG. 4.8 – Cas test du propulseur : maillage initial et déformé après le premier échange. Les couleurs indiquent l'amplitude des déplacements.

4.1.4 Intégration logicielle

Ces trois méthodes ont été testées dans un premier temps sur un démonstrateur développé via le code éléments finis ZéBuLoN. Ce démonstrateur s'est tout de même révélé trop cher pour être utilisé en production. Notre implantation rapide des éléments ressorts unidimensionnels ne tire par exemple pas partie du découplage des trois directions.

C'est pourquoi l'intégration définitive de la déformation de maillage dans MSD s'est faite grâce à une autre librairie éléments finis, développée par F.X. Roux du département DTIM/CHP de l'ONERA. Cette bibliothèque intègre aussi un solveur FETI⁵ permettant un calcul parallèle des déformations par sous-domaines, dans le cas où le calcul fluide est lui-même parallélisé.

Dans le cas où la matrice du problème est identique pour les trois directions, sa factorisation n'est effectuée qu'une seule fois, les conditions au bord étant les seules différences entre les trois calculs. Cette optimisation permet de rendre le coût de calcul des plus raisonnables.

L'utilisation de cette bibliothèque a considérablement accéléré le temps de calcul des déformations par rapport à ZéBuLoN. Au final, le temps de calcul de la déformée du maillage correspond à une dizaine d'itérations MSD (estimation réalisée sur un calcul gaz parfait, taille du maillage : $15 \times 20 \times 40$). En mémoire, le programme requiert sur ce même test 50% de celle requise par MSD. Il faut toutefois noter que ces chiffres sont surestimés, puisque pris sur une version non optimisée du déformeur de maillage. Ces coûts sont tout à fait raisonnables, puisque l'appel à ces routines de déformation de maillage ne se fait que toutes les 1 000, voire 10 000 itérations du code explicite fluide, soit un coût supplémentaire inférieur à un pourcent.

4.1.5 Conclusion

Un module de déformation de maillage parfaitement fonctionnel sur les applications que nous envisageons a été développé. L'utilisation de la bibliothèque FE-FETI de F.X. Roux a permis d'en rendre le coût raisonnable face aux coûts d'un calcul fluide, et permettra de profiter d'une résolution en parallèle le cas échéant.

Certains points pourront être améliorés : tout d'abord, les conditions au bord pourront être enrichies, afin de tenir compte de lignes ou de surfaces glissantes, représentant un plan ou un axe de symétrie (comme dans le cas du propulseur).

Actuellement, le calcul des déformées est réalisé sur le maillage fluide directement. Nous pourrions envisager d'alléger encore le temps de calcul en le réalisant sur une grille plus grossière.

Enfin, pour pallier les problèmes survenant lors de grandes transformations, il serait possible d'isoler dans une première phase le mouvement rigide d'ensemble du maillage, et de calculer les déformations sur un problème ayant isolé ce mouvement rigide.

Notons pour terminer que de nombreuses autres méthodes peuvent être utilisées pour déterminer le déplacement de maillage. Nous avons retenu la méthode facilement implantable avec les outils à notre disposition.

Il est intéressant de remarquer que, comme dans notre cas, de Boer *et. al.* [16] ont proposé une solution par fonctions d'interpolation radiales, basée sur une bibliothèque de calcul de structure qu'ils avaient à disposition.

⁵Acronyme de Finite Element Tearing and Interconnecting [23].

4.2 Développement et validation d'une condition au bord couplée en pression

Afin de développer et valider les méthodes de transfert et d'application du champ de pression de MSD à ZéBuLoN, nous proposons ici de comparer le résultat d'un calcul semi-couplé à une solution analytique. Le problème considéré est celui d'une barre en flexion soumise à un champ de pression variable.

Nous réalisons cette validation en deux temps : un premier test vérifiera simplement l'application des champs de pression dans ZéBuLoN ; un deuxième test viendra ajouter la phase de transfert de ce champ entre MSD et ZéBuLoN.

4.2.1 Solution analytique

Le problème choisi est celui d'une poutre de longueur L , de section rectangulaire, encastree à une extrémité, l'autre étant laissée libre. On néglige la pesanteur et on s'intéresse à la déformation causée par l'application d'une pression non homogène (Fig. 4.9).

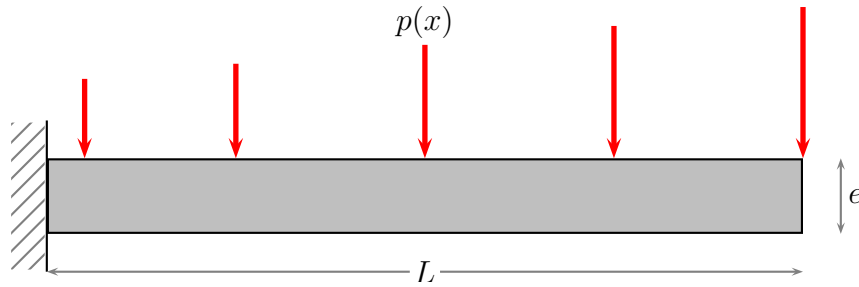


FIG. 4.9 – Cas de validation : poutre en flexion soumise à une pression non uniforme

Le chargement est orthogonal à la direction principale d'inertie Oz . Nous sommes donc dans le cas d'une poutre droite chargée dans son plan, et pouvons appliquer la théorie des poutres [35, chap. 7].

Sous des hypothèses de petites déformations et d'une loi de comportement linéaire élastique, l'équation suivante relie déplacements et chargements :

$$EI \frac{d^4 v(x)}{dx^4} = q(x) \quad (4.2)$$

Dans cette équation, v est le déplacement vertical, q la densité linéique de chargement, E le module d'Young et I le moment quadratique d'inertie. Pour une poutre de section rectangulaire, ce dernier vaut $I = \frac{be^3}{12}$, où b est la largeur et e l'épaisseur de la poutre.

Le chargement retenu est une pression variant de manière affine entre p_0 et p_L :

$$p(x) = \frac{p_L - p_0}{L} x + p_0 \quad (4.3)$$

la densité de chargement étant alors $q(x) = bp(x)$.

On utilisera aussi les conditions au bord suivantes :

– en $x = 0$ la poutre est encastree, donc $v(0) = 0$ et $\frac{dv}{dx}(0) = 0$.

4.2 Développement et validation d'une condition au bord couplée en pression 59

- en $x = L$ l'extrémité est libre, donc le moment de flexion M ainsi que le cisaillement V sont nuls. Comme $M = -EI \frac{d^2v}{dx^2}$ et $V = -EI \frac{d^3v}{dx^3}$, on a $\frac{d^2v}{dx^2}(L) = 0$ et $\frac{d^3v}{dx^3}(L) = 0$.

En prenant en compte le chargement et les conditions aux bords précédentes, l'intégration de l'équation 4.2 permet d'obtenir l'expression analytique du déplacement vertical de la poutre :

$$v(x) = \frac{bx^2}{120EI} \left[\frac{p_0 - p_L}{L} x^3 - 5p_0 x^2 + 10(p_0 + p_L)Lx - 10(p_0 + 2p_L)L^2 \right] \quad (4.4)$$

4.2.2 Comparaison avec un calcul découplé

Dans un premier temps, nous comparons la solution analytique précédente à la solution obtenue par un calcul découplé. Le champ de pression appliqué est donc directement donné par la fonction (4.3). Ceci permet de valider la méthode d'*application* des données, sans se soucier de leur *transfert* entre fluide et structure.

Les développements réalisés pour l'application d'une pression en provenance d'un code fluide sont adaptés pour accepter une forme analytique du champ de pression⁶.

L'application numérique choisie consiste en une poutre de dimensions $50 \times 5 \times 1 \text{ m}$ (Figure 4.10). Le maillage est choisi quadratique, les maillages linéaires étant mal adaptés aux problèmes de flexion. Un maillage en éléments cubiques de taille $1 \times 1 \times 1 \text{ m}$ offre une précision suffisante, des raffinements supplémentaires n'augmentant plus la précision du résultat.

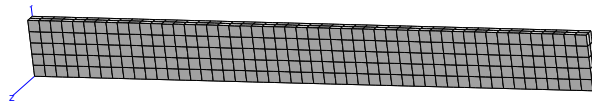


FIG. 4.10 – Maillage de la poutre utilisé pour le calcul de validation

On applique une pression variant linéairement entre 1 bar en $x = 0$ et 2 bar en $x = 50$. La comparaison entre la forme analytique et le résultat du calcul est présentée sur la figure 4.12. La concordance des deux courbes est satisfaisante : en bout de poutre, la différence entre solution analytique ($-0,722\text{m}$) et solution calculée ($-0,727\text{m}$) est inférieure à 1%.

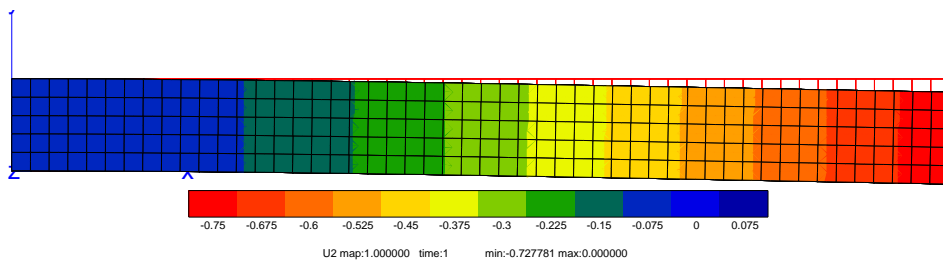


FIG. 4.11 – Résultat du calcul découplé de la poutre en flexion : les couleurs correspondent à l'intensité du déplacement vertical.

⁶Cette solution a depuis été modifiée : c'est maintenant la même condition au bord qui est utilisée pour appliquer une pression au moyen d'une fonction, ou au moyen d'un champ couplé. Seule la *source* de ce champ de pression est variable.

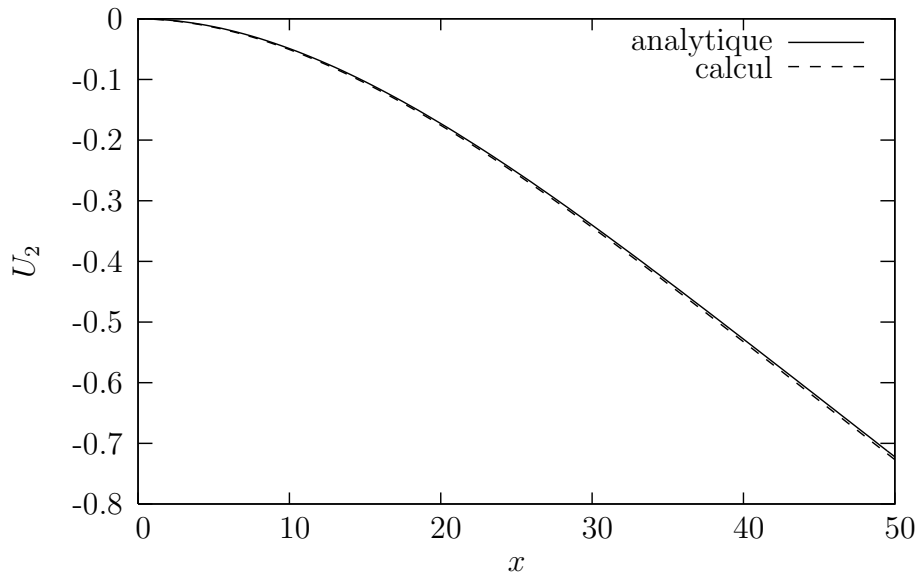


FIG. 4.12 – Comparaison entre la solution analytique et le calcul découplé

4.2.3 Confrontation entre le calcul couplé et le calcul non couplé

Afin de proposer un calcul complet de validation, on recherche un calcul fluide présentant un profil de pression donnant cette même variation linéaire, entre 1 et 2 *bar*.

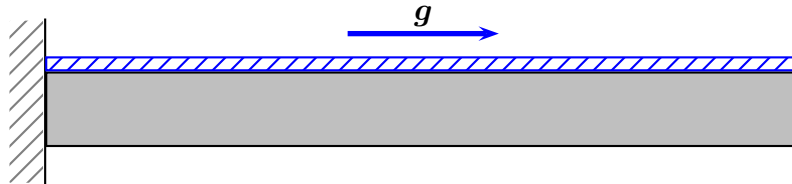


FIG. 4.13 – Représentation schématique du domaine fluide et de la barre

Une colonne d'eau soumise à son propre poids va nous fournir un tel résultat. La pression d'une telle colonne est en effet $P = P_0 + \rho gh$. Nous orientons l'axe d'application de la gravité à l'horizontale, et choisissons un coefficient de pesanteur *ad hoc* afin de retrouver une variation de pression entre 1 et 2 *bar*. Le gradient de pression ainsi obtenu est présenté sur la courbe 4.14.

Nous réalisons donc un calcul semi-couplé, dans lequel le fluide exerce sa pression sur la barre. Ce calcul ne réalise qu'une unique itération, et cherche simplement à vérifier que le champ transmis est appliqué correctement.

Le résultat de ce calcul est illustré par la figure 4.15 ; on constate sur les courbes 4.16 à nouveau un bon accord entre la solution analytique et le calcul. La différence entre les déplacements verticaux à l'extrémité de la poutre est inférieure à 1,4%.

4.2.4 Conclusion

Ces calculs nous ont donc permis de valider deux choses : tout d'abord les formules d'application d'un champ de pression variable sont correctement implantées, lorsque ce champ est

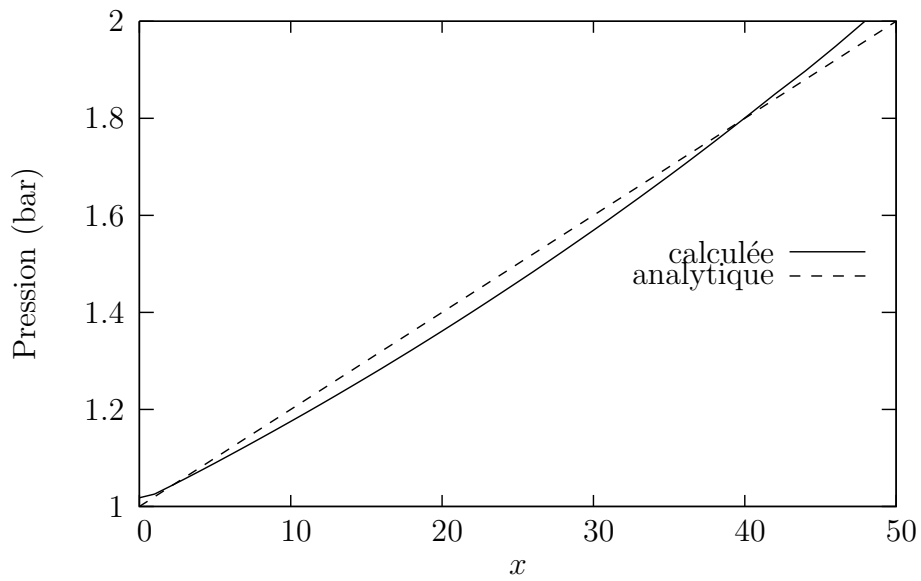


FIG. 4.14 – Profil de pression obtenu le long de la colonne

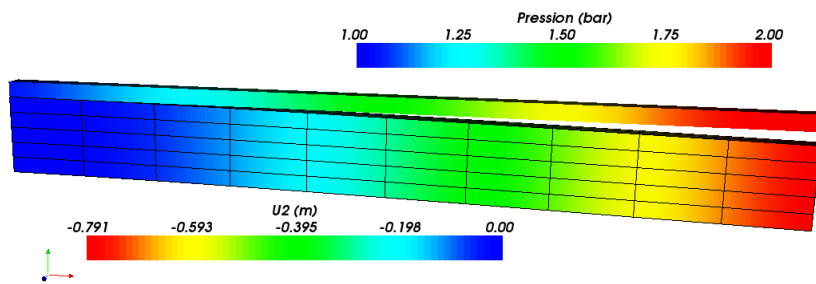


FIG. 4.15 – Visualisation du résultat du calcul semi-couplé de validation

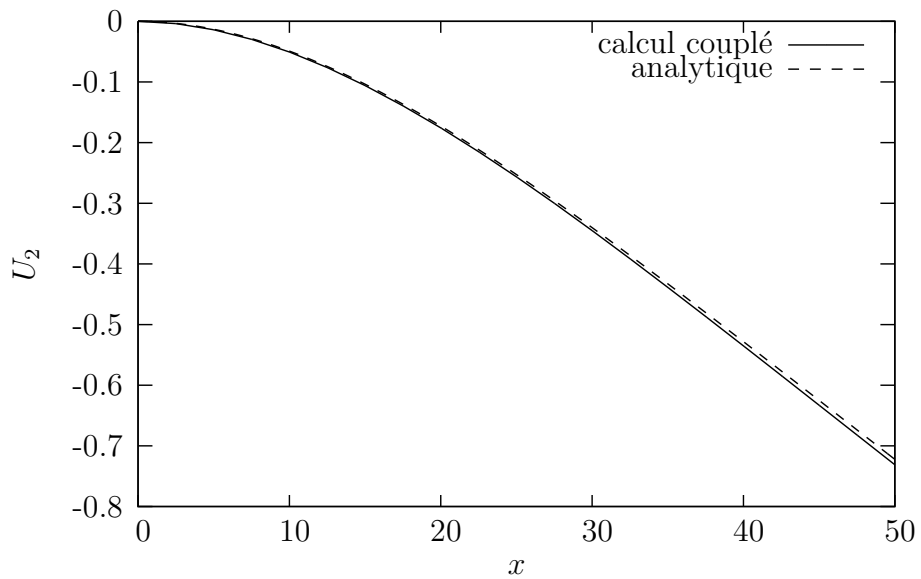


FIG. 4.16 – Résultat du calcul semi-couplé de validation

spécifié de manière analytique. Ensuite le couplage à sens unique (le fluide impose ses pressions, mais ne reçoit pas de déplacements) est lui aussi valide. Les méthodes de transfert et de réinterpolation étant utilisées pour ce calcul, elles sont aussi testées.

Ce cas de validation n'inclut par contre pas de calcul complètement couplé, afin de pouvoir rester dans un cadre pour lequel nous avons une solution analytique de référence. L'échange de données depuis la structure vers le fluide est quant à elle plus aisée à détecter, toute incohérence dans ce transfert se visualisant immédiatement par un décalage entre les géométries des deux domaines.

Chapitre 5

Le paradigme par l'exemple : couplages ZéBuLoN-ZéBuLoN

Nous présentons ici quelques séries de calculs couplant deux ou trois instances du logiciel ZéBuLoN. Le but de ces calculs est triple : illustrer les principes énoncés précédemment, montrer la faisabilité du couplage asynchrone et du couplage à trois codes et enrichir la base de tests.

Nous partons donc des algorithmes présentés dans le chapitre 2, et montrons que grâce aux développements liés aux remarques de ce chapitre, il est facile de passer d'un algorithme à un autre.

Afin de jouer avec les algorithmes de façon libre, il a été décidé de réaliser des calculs couplés entre deux ZéBuLoN, et ceci pour deux raisons de facilité : la première parce que ZéBuLoN est plus maniable qu'un logiciel de mécanique des fluides¹, la deuxième parce que les développements réalisés dans ZéBuLoN permettent de le coupler directement à lui-même.

Les exemples de couplage structure-structure présentés ici sont toutefois suffisamment riches pour présenter un certain nombre de difficultés, mais aussi pour mettre en évidence des comportements de calculs couplés que l'on retrouvera lors de couplages fluide-structure.

5.1 Dirichlet→Neumann synchrone : problème de la barre bi-matériau²

Les développements présentés au chapitre 3 permettent la mise en place des algorithmes usuels de couplage de type Dirichlet→Neumann, avec ou sans corrections. La méthode, appelée *Conventional Serial Staggered* dans [58], est mise en application dans le code ZéBuLoN pour l'étude des déformations d'une barre bi-matériaux.

L'utilisation de la méthode partitionnée n'apporte pas de nouveaux résultats, puisqu'un tel calcul est réalisable par ZéBuLoN directement³. Cet exemple sert au contraire de test de base du couplage, et nous a permis de réaliser un certain nombre de vérifications internes au logiciel. Une solution de référence est obtenue par un calcul monolithique découplé. Il permet

¹Cette maniabilité provenant à la fois de modèles et d'équations plus simples, et d'une architecture logicielle plus ouvertes aux modifications.

²Cette section est parue dans la GaZeTTe [31], le journal des utilisateurs du logiciel ZéBuLoN.

³Notons tout de même que c'est un problème mal conditionné s'il est abordé de manière monolithique. Le partitionnement que nous proposons sépare les grandes valeurs propres (associées au domaine rigide) des petites (associées au domaine le moins rigide), et couple donc deux sous-domaines bien mieux conditionnés.

aussi la comparaison de différentes méthodes de couplage (en série, parallèle, avec ou sans prédictions et corrections) sur un cas simple, et donc facile à comprendre. Nous pourrions ainsi discuter de l'efficacité de la méthode Dirichlet→Neumann en fonction des géométries et des contrastes entre les matériaux.

5.1.1 Présentation du problème

Le problème est le suivant : considérons une barre bi-matériaux chargée, découpée en deux sous-domaines.

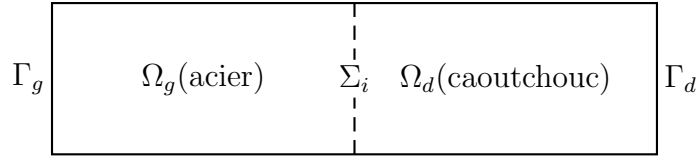


FIG. 5.1 – Barre bi-matériau décomposée en deux sous-domaines

Par simplicité, les deux matériaux sont ici considérés élastiques, en petites déformations. Les équations du problème sont écrites sur chaque sous-domaine :

$$\left\{ \begin{array}{l} \operatorname{div}(\boldsymbol{\sigma}_g) = \mathbf{0} \quad \text{dans } \Omega_g \\ + \text{conditions au bord sur } \Gamma_g \end{array} \right. \quad \left\{ \begin{array}{l} \operatorname{div}(\boldsymbol{\sigma}_d) = \mathbf{0} \quad \text{dans } \Omega_d \\ + \text{conditions au bord sur } \Gamma_d \end{array} \right. \quad (5.1)$$

et complétées de relations de compatibilité à l'interface Σ_i :

$$\mathbf{u}_g = \mathbf{u}_d \quad \text{sur } \Sigma_i \quad (5.2a)$$

$$\boldsymbol{\sigma}_g \cdot \mathbf{n}_g + \boldsymbol{\sigma}_d \cdot \mathbf{n}_d = \mathbf{0} \quad \text{sur } \Sigma_i \quad (5.2b)$$

où $\boldsymbol{\sigma}_*$ est le tenseur des contraintes dans Ω_* , \mathbf{u}_* est le vecteur déplacement du matériau, \mathbf{n}_* la normale extérieure à Ω_* (ainsi $\mathbf{n}_g = -\mathbf{n}_d$ sur Σ_i). Ces relations expriment la continuité des déplacements et des forces à l'interface entre les deux domaines.

On notera par la suite $\mathbf{f}_g = \boldsymbol{\sigma}_g \cdot \mathbf{n}_g$ et $\mathbf{f}_d = \boldsymbol{\sigma}_d \cdot \mathbf{n}_d$ les forces à l'interface.

Les contraintes sont reliées aux déformations par la loi de Hooke (cf. section 1.2.1) :

$$\boldsymbol{\varepsilon} = \underset{\approx}{\mathbf{S}} \boldsymbol{\sigma} = \frac{1+\nu}{E} \boldsymbol{\sigma} - \frac{\nu}{E} \operatorname{Tr}(\boldsymbol{\sigma}) \operatorname{Id} \quad (5.3)$$

5.1.2 Algorithme de couplage partitionné

La méthode de Dirichlet→Neumann avec corrections consiste à résoudre ce problème itérativement, en recherchant le point fixe de l'application

$$\mathbf{f}_g \longrightarrow \mathbf{u}_g|_{\Sigma_i} \longrightarrow \mathbf{u}_d|_{\Sigma_i} \longrightarrow \mathbf{f}_d \quad (5.4)$$

Un certain nombre de résultats théoriques existent quant à la convergence de la méthode de Dirichlet→Neumann⁴. En voici les grandes lignes.

⁴Voir par exemple les deux articles [29, 45]. Les auteurs ont appliqué la méthode à des problèmes de décomposition de domaines. Ils y introduisent en outre des méthodes de relaxation destinées à assurer et améliorer la convergence de l'algorithme.

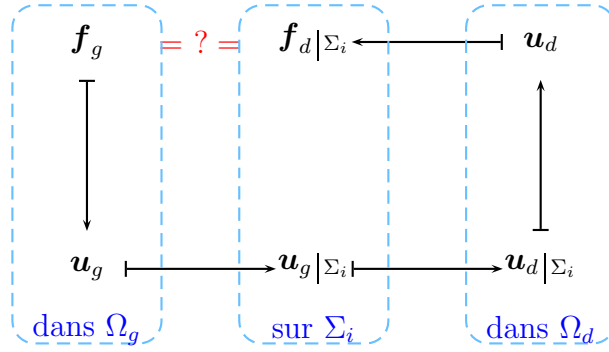


FIG. 5.2 – Une itération de l’algorithme de Dirichlet→Neumann (se lit dans le sens inverse des aiguilles d’une montre)

Commençons par regarder ce qui se passe dans Ω_g . Après discrétisation, le problème d’élasticité s’y écrit sous forme matricielle

$$\begin{pmatrix} K_{gg} & K_{gi} \\ K_{ig} & K_{ii} \end{pmatrix} \begin{pmatrix} U_g \\ U_i \end{pmatrix} = \begin{pmatrix} F_g \\ F_i^{(g)} \end{pmatrix} \quad (5.5)$$

où U_g est le déplacement de l’intérieur du domaine Ω_g (c’est le discrétisé de \mathbf{u}_g) et U_i le déplacement de l’interface Σ_i . De même, F_g et K_{gg} d’une part, F_i et K_{ii} d’autre part se rapportent respectivement à l’intérieur et à l’interface. Les termes croisés K_{gi} et K_{ig} rendent compte de l’interaction entre l’intérieur et l’interface.

Afin de simplifier les calculs qui suivent, nous supposons un instant que $F_g = 0$. Éliminer U_g de (5.5) permet d’extraire une relation entre les déplacements et les forces à l’interface :

$$(K_{ii} - K_{ig}K_{gg}^{-1}K_{gi})U_i = F_i^{(g)} \quad (5.6)$$

Cela fait apparaître le complément de Schur du domaine Ω_g : $S_g = K_{ii} - K_{ig}K_{gg}^{-1}K_{gi}$.

Le complément de Schur s’interprète de la manière suivante : c’est l’opérateur “Dirichlet vers Neumann” pour l’interface. En effet, lorsqu’il multiplie les déplacements de l’interface (U_i), nous obtenons les forces (F_i) qui ont justement été appliquées sur l’interface pour obtenir ce déplacement. Dit autrement, ce complément de Schur est le comportement que devrait avoir l’interface pour se comporter comme le domaine complet.

Le même raisonnement appliqué au domaine Ω_d nous fournit son complément de Schur $S_d = K_{ii} - K_{id}K_{dd}^{-1}K_{di}$, qui vérifie :

$$S_d U_d = F_i^{(d)} \quad (5.7)$$

Réécrite en ces termes matriciels, la méthode itérative de Dirichlet→Neumann devient :

Algorithme 3 : Algorithme partitionné avec corrections

répéter

Dans Ω_g : Prédire la force à l’interface $F_i^{(g)}$

et résoudre le problème de Neumann correspondant : $U_i^{(g)} = S_g^{-1}F_i^{(g)}$

Dans Ω_d : Appliquer ce déplacement au domaine Ω_d : $U_i^{(d)} \leftrightarrow U_i^{(g)}$

afin de résoudre le problème de Dirichlet : $F_i^{(d)} = S_d U_i^{(d)}$

jusqu’à ce que $F_i^{(d)} = F_i^{(g)}$

Une itération peut ainsi s'écrire de manière condensée : $F_i^{(d)} = S_d S_g^{-1} F_i^{(g)}$. La résolution du problème couplé devient donc la recherche du point fixe de l'application $F \mapsto S_d S_g^{-1} F$. Dans ce cas, le point fixe est bien évidemment 0, ce à quoi on pouvait s'attendre : une barre non chargée ne se déforme pas.

Dans le cas général (c'est-à-dire sans supposer $F_g = 0$ ni $F_d = 0$), le problème revient à la recherche du point fixe de $F \mapsto S_d S_g^{-1} F + C_i$, où (nous passons les détails de calcul) $C_i = K_{id} K_{dd}^{-1} F_d - S_d S_g^{-1} K_{ig} K_{gg}^{-1} F_g$.

Écrite de cette manière, la méthode s'apparente donc à la méthode itérative de Jacobi pour la résolution de systèmes linéaires que l'on sait être⁵

- convergente lorsque $\|S_g^{-1} S_d\| < 1$,
- divergente lorsque $\|S_g^{-1} S_d\| > 1$.

Ces résultats montrent que la convergence et stabilité de la méthode de couplage de Dirichlet→Neumann est entièrement liée aux normes des deux compléments de Schur, et donc aux caractéristiques élastiques relatives, ainsi qu'à la géométrie des domaines.

5.1.3 Illustrations numériques

En se rappelant que la norme du complément de Schur est liée au module d'Young du matériau, ainsi qu'à l'éclatement de la barre, on retrouve les problèmes d'instabilités mentionnés par F. Nobile [51], lorsque les matériaux sont faiblement contrastés.

Par commodité, on appellera "acier" le matériau dur, et "caoutchouc" le matériau mou, bien que dans ces expériences purement numériques, les paramètres matériaux utilisés soient fictifs.

Ces calculs sont réalisés sur maillages coïncidants. Ceci permet d'éviter le recours à la librairie MpCCI pour le transfert de champs à l'interface. Pour ces simulations, une barre telle que celle présentée sur la figure 5.3 est utilisée. Elle est constituée de deux matériaux élastiques, soumise à un champ de pression uniforme sur sa surface supérieure, fixée sur ses bords latéraux et soutenue en ses deux extrémités.

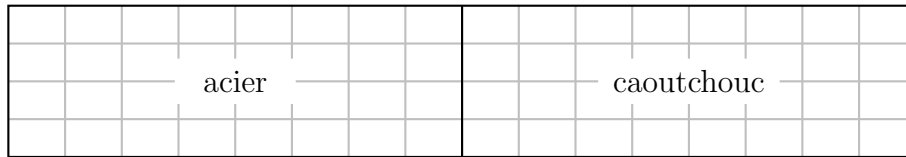


FIG. 5.3 – Modèle discret et conditions aux limites pour la barre bi-matériau

Un premier calcul est réalisé, pour une barre de dimensions $6 \times 1 \text{ mm}$, en matériaux élastiques : un module d'Young de 100 MPa un coefficient de Poisson de 0,3 sont choisis pour l'"acier", de 1 MPa et 0,49 pour le "caoutchouc". Une pression de $0,1 \text{ MPa}$ est appliquée sur la face supérieure. La solution de référence obtenue par calcul monolithique est présentée sur la figure 5.4.

Le calcul couplé est réalisé en utilisant un critère de convergence sur les efforts à l'interface entre les deux domaines ; nous réalisons des corrections jusqu'à obtenir :

$$|RU_g + RU_d| < 10^{-5}$$

⁵voir par ex. Ciarlet [14]

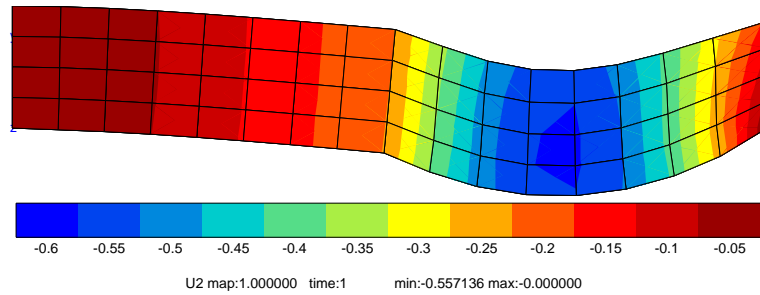


FIG. 5.4 – Solution de référence du problème de la barre bi-matériau, obtenue par un calcul monolithique découplé. Les déplacements verticaux (U_2) sont représentés.

où RU_g et RU_d désignent les réactions nodales à l'interface (vues du domaine gauche et droit respectivement), c'est-à-dire les forces appliquées ponctuellement en chaque nœud de l'interface et intégrant les efforts entre les domaines.

Les échanges de données entre les deux domaines se faisant en simple précision, cette borne correspond à une centaine de fois la précision machine de ce calcul. La méthode utilisée étant Dirichlet→Neumann, l'égalité des déplacements à l'interface est forcément satisfaite à chaque fin d'itération de couplage. La solution convergée est obtenue par cette méthode au bout de 8 itérations de couplage. Elle est, aux erreurs de précision près, identique à la solution de référence.

La figure 5.5 présente l'évolution de la solution au cours des itérations de couplage. Pour des raisons de clarté (il est plus parlant de visualiser des sauts de déplacements que des sauts de forces), nous avons réalisé le calcul en démarrant par le domaine "caoutchouc", pour un déplacement de l'interface prédit nul à la toute première itération. Le domaine transmet ensuite ses réactions à l'interface au domaine "acier". Ce dernier calcule ses déplacements sous ce chargement, puis transmet la position de l'interface. Le déséquilibre à l'interface en fin d'itération de couplage est donc sur les déplacements.

Le graphique 5.6 présente quant à lui l'évolution du saut des déplacements à l'interface au cours des itérations de calcul. Ce saut est un indicateur de l'erreur due à la méthode.

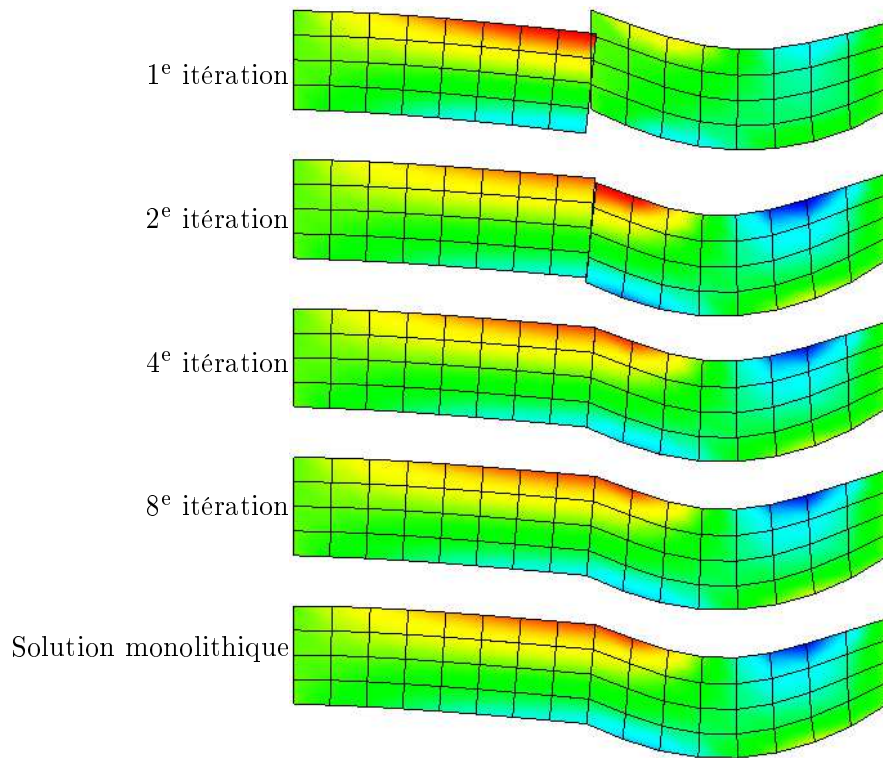


FIG. 5.5 – Évolution de la solution au cours des itérations de couplage ; les couleurs représentent les déplacements horizontaux U_1 . Pour une visualisation plus parlante, une méthode pour laquelle le déséquilibre est au niveau des déplacements plutôt que des efforts a été utilisée.

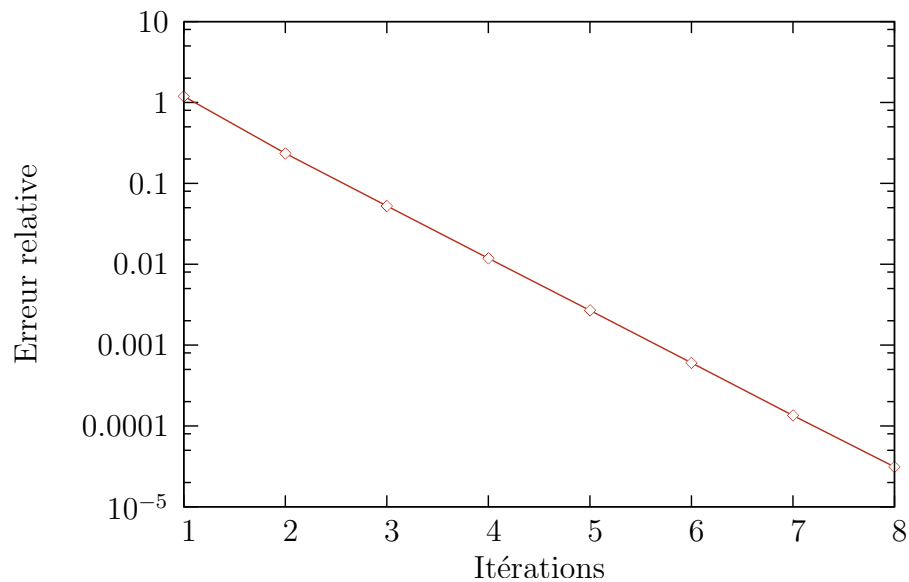


FIG. 5.6 – Évolution de l'erreur à l'interface au cours des itérations de couplage. L'erreur est donnée comme étant $\max_{\text{nœuds}} \|U_g - U_d\|$, et est normalisée par le déplacement maximal à convergence.

Influence du module d'Young

Une première série de calculs, pour différents modules d'Young a été réalisée. Les résultats de convergence sont présentés dans le tableau 5.1. Les deux premières colonnes donnent les modules d'Young utilisés pour les matériaux. La troisième colonne y présente le nombre d'itérations qui a été nécessaire afin d'atteindre une solution couplée convergée (l'écart que l'on s'est autorisé entre les solutions des deux domaines est de 10^{-6}).

Young Ω_g (MPa)	Young Ω_d (MPa)	Contraste	Itérations à convergence
200	1	200	5
100	1	100	6
50	1	50	7
25	1	25	12
12.5	1	12.5	58
11.1	1	11.1	438
11	1	11	diverge (très lentement)
10 (et moins)	1	10	diverge
200	2	100	6
100	2	50	7
50	2	25	12
25	2	12.5	58

TAB. 5.1 – Influence des modules d'Young sur la convergence. Le *contraste* est défini comme le rapport E_g/E_d des modules d'Young des deux domaines.

Ce tableau résume un certain nombre de résultats importants concernant la méthode Dirichlet→Neumann. Le premier est, comme déjà annoncé, que plus le contraste entre les deux matériaux est important, meilleure est la convergence de l'algorithme couplé. Pour des matériaux de module d'Young proches, la méthode devient par contre instable.

Ceci confirme en particulier la pertinence de cette méthode dans les cas d'interaction entre de l'air et une aile d'avion par exemple : le contraste entre les "matériaux" (air et solide) est suffisamment important pour qu'une seule correction suffise, voire même qu'aucune correction ne soit nécessaire.

Le deuxième résultat est que cette vitesse de convergence n'est non pas liée au module d'Young, mais bien au *rapport* des modules, c'est-à-dire au *contraste* entre les matériaux.

Enfin, le troisième est qu'il est indispensable, pour la convergence de l'algorithme, de choisir le matériau "dur" pour imposer ses déplacements, et le matériau "mou" pour imposer ses réactions. Autrement dit, dans les situations de couplage fluide-structure que l'on étudiera par la suite, le fluide aura des conditions de couplage de Dirichlet, alors que le solide aura des conditions de couplage de Neumann.

Influence de l'élanement

Dans une deuxième série de calculs, nous fixons les paramètres des deux matériaux (module d'Young de "l'acier" à 100 MPa, celui du "caoutchouc" à 1 MPa, coefficients de Poisson à 0.3), et jouons sur l'élanement de la barre. Les résultats obtenus sont résumés dans le tableau 5.2. On constate cette fois que la convergence de l'algorithme dépend de l'*élanement* de la barre.

Longueur	Largeur	Élancement	Itérations à convergence
1	1	1	5
3	1	3	7
9	1	9	11
2	2	1	5
6	2	3	7
18	2	9	11

TAB. 5.2 – Influence de l'élancement sur la convergence

Il faudra donc s'attendre à des difficultés de convergence des algorithmes couplés dans le cas de simulations sur des structures particulièrement élancées.

5.1.4 Raccord par conditions mixtes

Les méthodes présentées par S. Chemin [13] ont montré, dans le cas du couplage thermique, le gain de stabilité que l'on obtient par un couplage Fourier→Fourier ou Dirichlet→Fourier. Cette méthode a aussi depuis longtemps été adoptée pour les calculs parallèles par décomposition de domaines, situation s'assimilant à un problème de barre à matériaux identiques, c'est-à-dire exactement la situation pour laquelle la méthode Dirichlet→Neumann pose des problèmes de stabilité et de convergence.

L'intérêt de l'utilisation de ce type de condition mixte a aussi été très récemment mis en évidence dans des cas mécaniques de simulation en hémodynamique par Badia *et. al.* [4]. Ce sont effectivement des cas où le contraste entre le fluide (sang) et la structure (veine) est extrêmement faible.

Les applications que nous avons envisagées ne présentent que des situations où les matériaux sont fortement contrastés, bien traitées par la méthode Dirichlet→Neumann. C'est pourquoi nous n'avons pas encore implanté cette condition mixte dans le cas mécanique. Ce sera toutefois la première amélioration à envisager dès l'apparition de problèmes couplés mécaniques présentant des difficultés à converger.

5.2 Couplage asynchrone : problème de thermo-mécanique

L'étude d'un problème de thermo-mécanique va nous permettre de décrire, mettre en place et tester les méthodes de couplage asynchrones. Ce couplage implique deux instances de ZéBuLoN : la première est chargée de la résolution du problème mécanique, la deuxième du problème thermique.

5.2.1 Présentation d'un problème modèle

Le problème de thermo-mécanique que nous voulons résoudre couple les équations de la mécanique (éq. (1.4)) à celle de la thermique (éq. (1.11)) :

$$\begin{aligned} \operatorname{div}(\underline{\underline{\mathbf{K}}} : \underline{\underline{\boldsymbol{\varepsilon}}}) &= 0 & \text{dans } \Omega_0 \\ \operatorname{div}(\kappa \operatorname{grad} T) &= 0 & \text{dans } \Omega(t) \end{aligned}$$

dans lesquelles l'équation de la chaleur est écrite sur le domaine courant et les coefficients matériaux constituant le tenseur de rigidité $\underline{\underline{\mathbf{K}}}$ dépendent de la température.

Ce problème couplé est classiquement traité par la méthode partitionnée consistant à résoudre à tour de rôle chacun des deux problèmes, et d'échanger champs de déplacements et de température. Nous nous intéressons ici à une méthode de résolution asynchrone des deux sous-problèmes, dont le principal avantage est de ne pas assujettir chaque calcul aux pas de temps de l'autre physique.

Le modèle géométrique est volontairement très simple : il s'agit d'un cube unitaire. Les conditions aux limites mécaniques sont imposées afin de bloquer tout mouvement de corps rigide : U_1 est pris nul sur la face gauche ($x = 0$), U_2 est pris nul sur la face inférieure ($y = 0$) et U_3 est pris nul sur la face arrière ($z = 0$). Les déformations sont donc uniquement dues aux évolutions thermiques. Le matériau est choisi thermo-élastique sous l'hypothèse des petites déformations. Son module d'Young est fonction de la température : il varie entre $2.10^5 Pa$ à $0^\circ C$ et $2.10^4 Pa$ à $300^\circ C$ et est constant en dehors de cet intervalle. Le coefficient de Poisson est pris égal à 0,3, et le coefficient de dilatation thermique à $10^{-4} C^{-1}$.

Le problème thermique est lui aussi très simple : le bord gauche se voit imposer une variation de température au cours du temps, partant de $0^\circ C$ à $t = 0s$, pour monter à $20^\circ C$ à $t = 1s$. Le bord de droite se voit imposer ces mêmes températures, multipliées par un facteur 10. Les autres limites sont laissées à flux nul. La constante de conductivité κ est prise égale à $1 W.m^{-1}.^\circ C^{-1}$.

5.2.2 Approche partitionnée de ce problème

En mécanique non-linéaire, il devient difficile, voire impossible de réaliser des calculs en prévoyant a priori les pas de temps. Au contraire, la résolution du problème se fait grâce à des pas de temps adaptatifs (cf. section 1.2.3) afin de suivre l'évolution des non-linéarités vues par la structure.

Dans les applications que nous envisageons, nous nous trouverons fréquemment dans la situation couplant un tel problème de mécanique fortement non-linéaire avec un problème linéaire (ou faiblement non-linéaire) de thermique.

L'irrégularité des pas de temps rend impossible l'utilisation des solutions usuelles de couplage, consistant à coupler à intervalles réguliers. Ce problème peut être contourné de multiples manières : il est tout d'abord possible de forcer le code de mécanique du solide à passer par les

pas de temps choisis pour réaliser les échanges. Nous excluons toutefois cette solution, car elle implique des modifications de l'horloge de résolution du solide, et vient interférer avec elle. Ces interférences viennent ajouter un coût supplémentaire au calcul, sans y apporter d'avantages.

Une deuxième solution consiste à synchroniser le code de thermique sur celui de la mécanique. C'est à nouveau une solution envisageable, mais qui engendre des coûts de calculs supplémentaires importants⁶, en particulier dans la situation où la structure requiert de petits pas de temps, non nécessaires à la thermique. Nous utiliserons cette solution dans le cas du couplage aéro-mécanique stationnaire (cf. section 7.1), l'asynchronisme n'étant actuellement pas disponible dans le logiciel fluide.

Le problème de thermo-mécanique impliquant deux instances de ZéBuLoN, nous pouvons envisager une troisième solution ne nécessitant aucune synchronisation entre les codes.

Dans le modèle choisi ci-dessus, le couplage entre mécanique et thermique est presque à sens unique : l'influence de la thermique sur la mécanique est importante ; par contre, l'influence des (petites) déformations est minime sur le problème de conduction. Les étapes de corrections ne sont donc pas nécessaires. Le problème pourrait éventuellement être traité en deux étapes découplées : dans un premier temps résoudre le problème thermique à géométrie fixée ; puis dans un deuxième temps résoudre le problème mécanique sous champ thermique calculé précédemment. La solution que nous proposons a néanmoins l'avantage de réactualiser la géométrie, pour un coût de calcul identique.

L'algorithme en série proposé est quasiment identique à l'algorithme CSS (algorithme 1, page 24). La seule différence se trouve dans la manière de déterminer le pas de temps d'échange suivant. Afin de clarifier les notations, nous noterons t_n^T , pour $n = 0, \dots, N$ les pas de temps⁷ du calcul thermique, et t_m^S les pas de temps du solide (irréguliers, désynchronisés avec la thermique et en nombre inconnu a priori).

Algorithme 4 : Algorithme asynchrone de couplage en série

```

pour  $n = 1 \dots N$  faire
1   | Prédire le déplacement du solide à  $t_n^T$ 
2   | Résoudre le problème de diffusion thermique dans ce domaine mobile de  $t_{n-1}^T$  à  $t_n^T$ 
3   | Transférer le champ thermique au solide
4   | Sous ce champ thermique, résoudre autant d'incrémentes solide que nécessaire sans
   | dépasser  $t_n^T$ 
5   | Transférer le dernier déplacement au problème thermique.
fin

```

Cette solution est retenue principalement pour les deux qualités suivantes :

- elle évite au code solide de travailler sur des prédictions du champ thermique,
- tout en limitant au maximum les décalages en temps entre les deux problèmes.

Cet algorithme nécessite du côté thermique des prédictions des déplacements. Nous allons choisir simplement une prédiction constante (c'est-à-dire égale au dernier champ reçu), afin d'éviter qu'une prédiction linéaire ou quadratique des déplacements lors des changements de régime ne soit trop erronée.

⁶Notons toutefois que si les maillages de la mécanique et de la thermique sont d'une finesse comparable, la résolution du problème de diffusion thermique est beaucoup plus rapide que celle du problème élasto-viscoplastique.

⁷Le problème thermique choisi étant ici linéaire, on peut supposer les pas de temps réguliers et surtout prédéterminés.

Côté mécanique par contre, nous allons réaliser des interpolations linéaires entre les champs thermiques reçus, afin de disposer d'une carte de température aux instants du calcul mécanique.

5.2.3 Résultats sur maillages coïncidents

Le premier calcul est réalisé sur une situation où les deux maillages sont coïncidents, afin de pouvoir comparer les résultats à ceux obtenus par la méthode de couplage thermo-mécanique interne⁸, existant dans ZéBuLoN.

Le maillage commun retenu est simplement celui du cube unité, maillé régulièrement en $5 \times 5 \times 5$ cubes linéaires. Notons que ce maillage grossier linéaire représente parfaitement la solution du problème thermique, celle-ci étant simplement affine entre 20°C et 200°C .

La figure 5.7 présente les résultats du calcul thermo-mécanique : le champ de température varie bien linéairement entre 20 et 200°C . Les déformations consécutives à cette augmentation de température sont de l'ordre de 2%, correspondant au produit des 200°C par le coefficient de dilatation thermique.

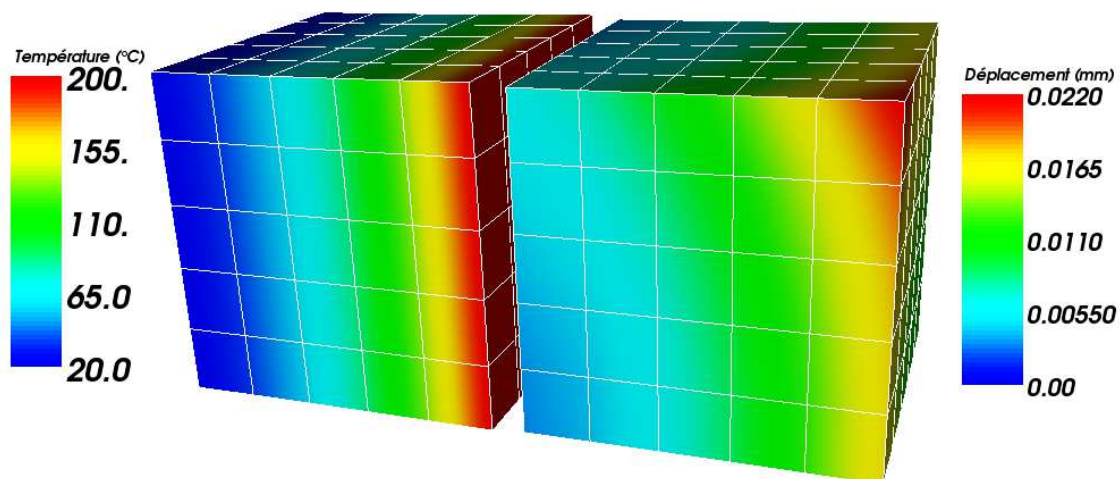


FIG. 5.7 – Calcul couplé thermo-mécanique à maillages coïncidents par couplage externe ; la température et la norme des déplacements sont représentés, à $t = 1\text{s}$.

Ce calcul est en bonne adéquation (Fig. 5.8) avec les résultats obtenus par couplage interne à ZéBuLoN, couplage qui repose lui aussi sur une méthode partitionnée faible. La légère différence entre les deux calculs peut s'expliquer par des différences d'implantation de l'algorithme.

5.2.4 Résultats sur maillages non-coïncidents

L'intérêt d'utiliser la librairie MpCCI est principalement de pouvoir réaliser des calculs sur maillages non-coïncidents. Nous reprenons donc le test précédent, en maillant différemment le cube mécanique.

⁸Cette méthode interne est elle aussi basée sur un couplage partitionné faible, implantant un algorithme de type CSS (page 24).

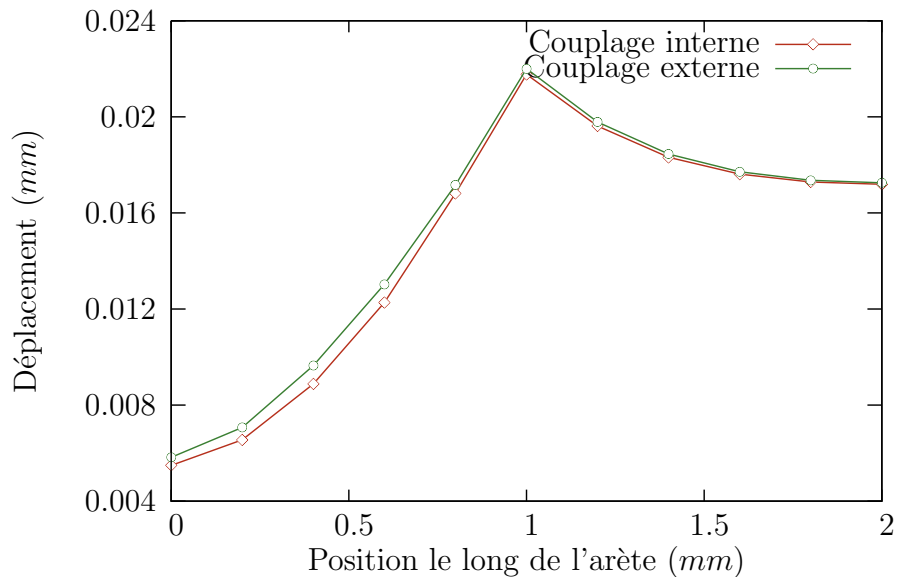


FIG. 5.8 – Comparaison des résultats couplés par la méthode interne de ZéBuLoN et le couplage externe : les déplacements sont représentés le long des arêtes supérieures et droite ($y = z = 1$ et $x = y = 1$, voir Fig. 5.9).

La figure 5.9 présente les deux maillages utilisés : côté thermique, nous conservons celui $5 \times 5 \times 5$ utilisé précédemment, et côté mécanique, nous prenons un maillage du cube en tétraèdres linéaires.

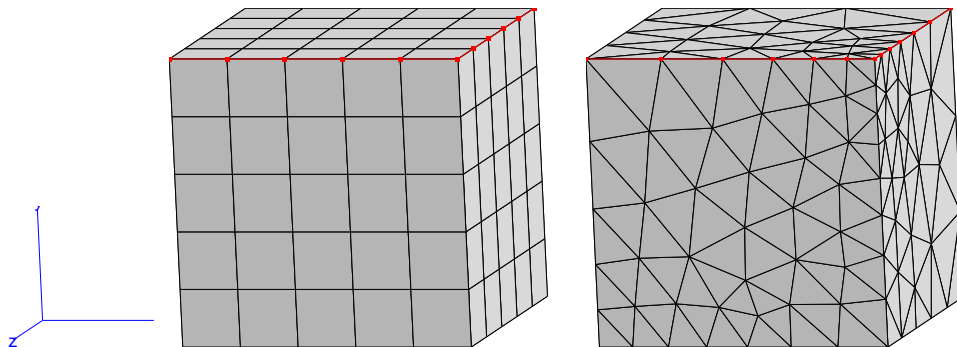


FIG. 5.9 – Maillages non-coïncidents du cube : à gauche celui utilisé pour la thermique ; à droite celui utilisé pour la mécanique. Les deux arêtes suivies dans les graphiques 5.8, 5.11 et 5.13 sont mises en évidence.

Nous obtenons les résultats illustrés par la figure 5.10 : les résultats sont semblables à ceux obtenus précédemment.

La courbe des déplacements sur les deux arêtes supérieures (Fig. 5.11) est légèrement différente de celle obtenue pour la méthode de couplage interne. Cette différence n'est pas due à une imprécision du transfert de champ, mais à un maillage mécanique trop grossier.

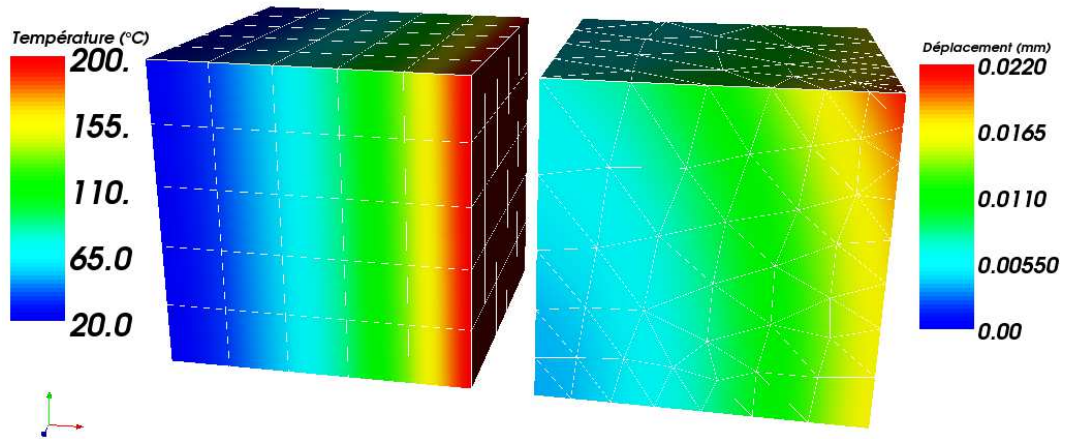


FIG. 5.10 – Calcul couplé thermo-mécanique à maillages non-coïncidents.

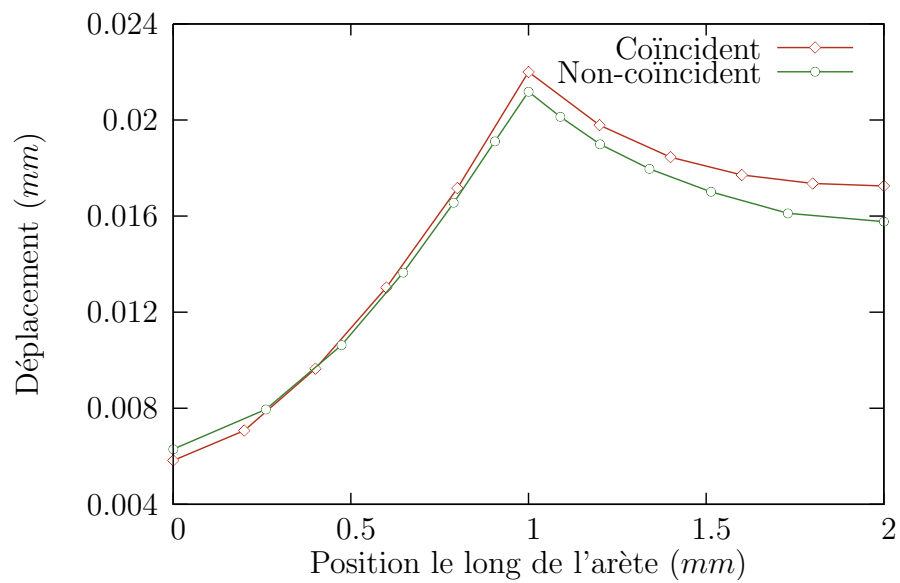


FIG. 5.11 – Comparaison des déplacements pour un calcul couplé sur maillages non-coïncidents

5.2.5 Maillage quadratique

Ce cas test est aussi l'occasion de lever une des limitations de MpCCI. En effet, cette librairie ne connaît actuellement pas les éléments volumiques quadratiques, pourtant indispensables dans les calculs mécaniques. Nous avons donc développé une couche intermédiaire au transfert, dégénérant le maillage quadratique en le maillage linéaire sous-jacent, et utilisant ce maillage linéaire pour le transfert via MpCCI.

Cette interface supplémentaire permet de réaliser le couplage entre maillages linéaires et quadratiques, tel que présenté sur la figure 5.12 : le maillage utilisé en thermique est linéaire, et celui utilisé pour le calcul mécanique est quadratique.

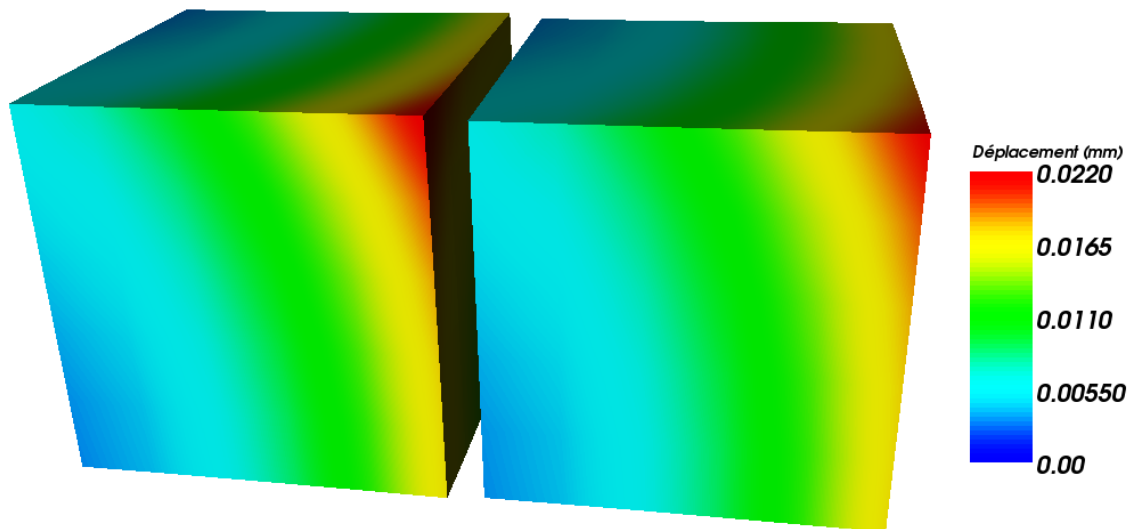


FIG. 5.12 – Comparaison du calcul couplé thermo-mécanique pour des maillages linéaire (à gauche) et quadratique (à droite)

Sur ce cas simpliste du cube, cette méthode retrouve des résultats similaires à ceux obtenus sur maillage coïncidents. Les champs sont effectivement suffisamment réguliers pour être bien représentés par le maillage linéaire sous-jacent. Nous ne cherchons à montrer ici que la validité fonctionnelle de la méthode, pas sa précision numérique.

5.2.6 Asynchronisme et corrections

L'exemple simple que nous avons choisi ne nécessite pas de phase de corrections, les déformations n'influent presque pas sur la diffusion. Ce type de couplage à sens unique est fréquent, mais il existe des cas où le couplage est fort dans les deux sens. Dans ces situations, des phases de corrections sont nécessaires.

Ces corrections sont le point difficile à gérer dans une approche asynchrone. Dans le schéma que nous avons proposé, seule la thermique travaille sur des prédictions. La mécanique, n'utilisant que des champs interpolés entre deux champs reçus, peut se passer de l'évaluation de critères d'erreurs. La partie thermique nécessite par contre des vérifications de ces prédictions, en calculant par exemple la différence entre le champ thermique calculé initialement et celui recalculé une fois les déformations mécaniques actualisées.

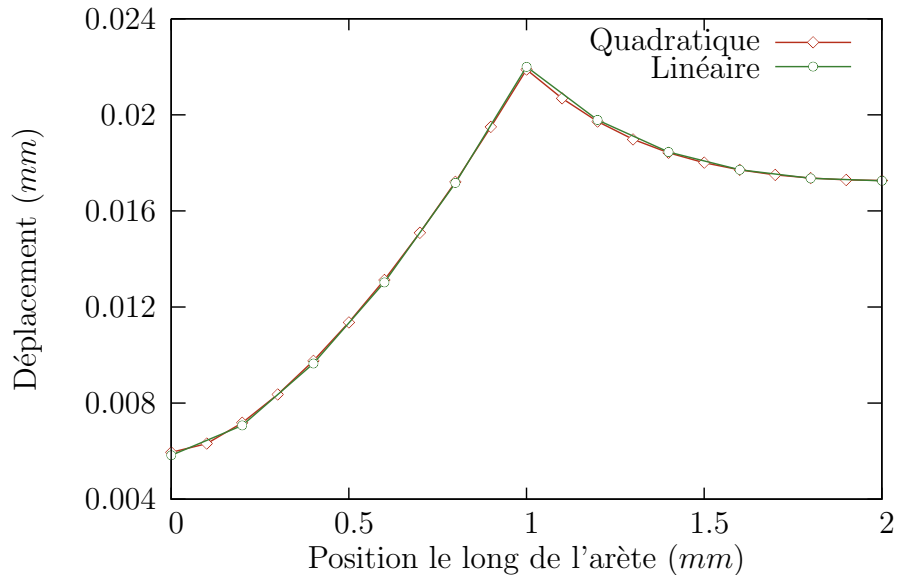


FIG. 5.13 – Comparaison des déplacements le long des arêtes supérieures et droite pour un maillage linéaire et quadratique

Le problème apparaît si cette différence est trop importante. Dans cette situation, le nouveau champ thermique doit être renvoyé au code mécanique, qui doit lui aussi entrer dans une phase de corrections. Tant que cette correction n'implique qu'un incrément de calcul, c'est une chose aisée. Mais dès qu'elle nécessite de reprendre plusieurs incréments (ce qui est le cas, puisque l'asynchronisme a été mis en place pour cette situation), les mesures à prendre deviennent complexes.

N'ayant dans le cadre de cette thèse pas d'applications précises nécessitant de telles méthodes, nous ne nous sommes pas investis dans cette direction. Lors de travaux précédents, S. Chemin a eu recours à de telles méthodes de retour en arrière sur plusieurs incréments. Les briques élémentaires nécessaires à un couplage complètement asynchrone avec corrections sont donc toutes développées. Il ne reste plus qu'à les assembler en un ensemble cohérent.

5.3 Couplage à trois codes

Le couplage de plusieurs instances de ZéBuLoN est l'occasion de tester le fonctionnement du couplage à trois codes. L'ajout du troisième code apporte plusieurs types de difficultés supplémentaires : tout d'abord conceptuelles, puisqu'il s'agit de déterminer de nouveaux algorithmes prenant en compte l'ensemble des échanges et équilibrages nécessaires. L'informatique s'est aussi révélée problématique, le logiciel MpCCI ne se comportant pas de manière identique dans son fonctionnement à 2 ou 3 codes.

Afin de se faire la main sur ce type de couplage, nous avons réalisé un démonstrateur simple, couplant trois instances de ZéBuLoN : les deux premières réalisent respectivement un calcul mécanique et un calcul thermique, comme dans la section précédente. La troisième réalise un calcul mécanique sur un cube juxtaposé, et mime le comportement d'un code fluide en recevant les déplacements du code mécanique. Par simplicité, nous allons omettre le renvoi d'un champ de pression (ou de réactions). La figure 5.14 résume ces échanges.

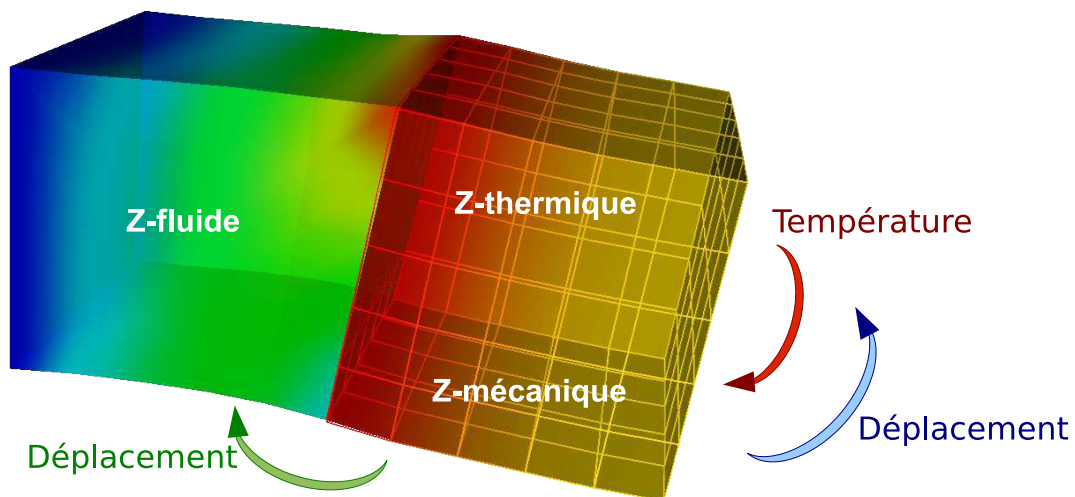


FIG. 5.14 – Communications requises pour un couplage à trois codes

5.3.1 Algorithme synchrone

Si l'on s'autorise une solution en série synchrone, le passage au couplage à trois codes se fait de manière presque transparente : il suffit simplement d'enchaîner les codes un à un,

thermique, mécanique puis fluide. Un algorithme possible est donc le suivant :

Algorithme 5 : Algorithme synchrone de couplage à 3 codes

```

pour  $n = 1 \dots N$  faire
  répéter
1   prédire déplacement du domaine thermique
2   résoudre le problème thermique sur ce domaine mobile
3   transférer le champ de température
4   résoudre le problème mécanique
5   transférer les déplacements aux deux autres codes
6   résoudre le problème fluide
  jusqu'à ce que la prédiction soit suffisamment précise
fin

```

La réalisation de ce calcul couplant trois instances de ZéBuLoN a posé quelques difficultés au niveau de MpCCI. Il est en effet apparu que les communicateurs créés par défaut ne sont pas adaptés au couplage à trois codes. Il a donc fallu en ajouter manuellement la déclaration. Cette déclaration manuelle va rendre impossible le couplage par points de synchronisation déclarés dans le fichier de mise en données. C'est donc pour cela que nous continuerons à réaliser les couplages par de simples envois/réceptions.

Nous avons utilisé une implantation générique de l'algorithme de couplage, qu'il est possible de paramétrer dans le fichier de mise en données⁹.

Ce calcul de mise au point a donc permis de montrer la faisabilité du couplage à trois codes, ainsi que le bon comportement de ZéBuLoN dans cette situation.

5.3.2 Algorithme asynchrone

Ce couplage à trois codes est aussi l'occasion de continuer les essais concernant le couplage désynchronisé. Nous réalisons donc ce même calcul en utilisant une stratégie un peu différente de celle utilisée dans l'algorithme 4 (page 72).

Cette fois, c'est le code mécanique qui démarre le calcul, et travaille prioritairement sur des prédictions. Une fois le champ de déplacements reçu, le code thermique prend la suite ; en parallèle, le code "fluide" réalise son calcul dès qu'il reçoit le champ de déplacements.

Néanmoins, le code thermique utilise une réception bloquante pendant seulement 60 secondes (environ dix incréments du code solide). Ainsi, dans les cas où la mécanique se calcule facilement c'est elle qui travaille sur extrapolations ; mais dans le cas où la mécanique subit de fortes non-linéarités, c'est le code thermique qui prend de l'avance et réalise les extrapolations.

Le fait de ne pas imposer l'ordre de résolution entre les codes n'est pas motivé par des raisons algorithmiques. Nous cherchons par contre à montrer la possibilité de laisser les codes indépendants les uns des autres, et à leur laisser la possibilité de continuer le calcul, même en l'absence de résultats des autres codes. Ce type de méthode pourra se montrer intéressante, puisqu'elle permet à un code tournant sur un processeur inactif de proposer une première prédiction de son champ, en attendant les données des autres codes.

Sur l'application retenue, nous nous trouvons effectivement dans une telle situation. Au deuxième incrément de calcul, le code mécanique souffre de difficultés à converger. Sa résolution

⁹À l'époque de la réalisation de ce calcul, cela semblait être la bonne méthode d'implantation de l'algorithme. Il est depuis apparu qu'une implantation directe et adaptée est plus aisée et lisible.

nécessitant plus de 60 secondes CPU, c'est le code thermique qui prend les devants.

Le simulation retenue, illustrée par la figure 5.14, dure une seconde. Nous avons imposé de réaliser les envois de données toutes les 0,1 s, mais sans forcer les codes à passer par ces instants précis. Ainsi, le code mécanique adaptant ses pas de temps s'est retrouvé à envoyer ses champs à $t = 0.125, 0.210, 0.336, 0.438, 0.508, 0.604, 0.725, 0.818, 0.903$ et 1. De son côté, le code thermique a réalisé ses envois à $t = 0.11, 0.2, 0.322$, etc... Le code "fluide" a pour sa part tourné à fréquence fixe, et a donc réalisé ses envois toutes les 0,1 secondes.

Les réceptions sont par contre non bloquantes : à chaque début d'incrément, les trois codes vérifient dans la pile de réception si un nouveau champ couplé a été reçu. Cela leur permet de réaliser l'incrément courant avec un champ aussi à jour que possible.

La figure 5.14 présente le résultat en milieu de calcul par cette méthode. On peut d'ailleurs y constater le léger décalage (visible par la pénétration du domaine mécanique dans le domaine fluide) entre les différents codes : la solution du calcul mécanique est connue à $t = 0,508$ s, celle du code thermique à $t = 0,548$ s et celle du code "fluide" à $t = 0,5$ s.

Dans ce test, nous supposons que le cube thermo-mécanique est en Inconel 600, et suit une loi élasto-visco-plastique. Le calcul résultant présente des non-linéarités sévères, et nécessite en conséquence l'utilisation du pas de temps automatique. Ceci a pour conséquence de désynchroniser complètement le calcul des deux autres.

Troisième partie

Étude d'un tube de tuyère

Chapitre 6

Premières approches

6.1 Présentation générale de la tuyère

La fusée Ariane 5 est équipée d'un moteur cryotechnique Vulcain 2. Nous nous intéressons ici à son divergent, partie servant à canaliser et accélérer les gaz éjectés. La partie haute de ce divergent est constituée de 288 tubes enroulés en spirale, tel que l'on peut le voir sur les photos 6.1 ou sur le schéma 6.2. Ces tubes sont refroidis par une circulation d'hydrogène. Ils sont en Inconel 600, un alliage de nickel-chrome présentant une bonne résistance à l'oxydation et à la corrosion, et qui conserve surtout ces bonnes propriétés aux températures élevées.



FIG. 6.1 – Divergent du moteur Vulcain 2, dans sa version renforcée (photos prises au salon du Bourget – juin 2005).

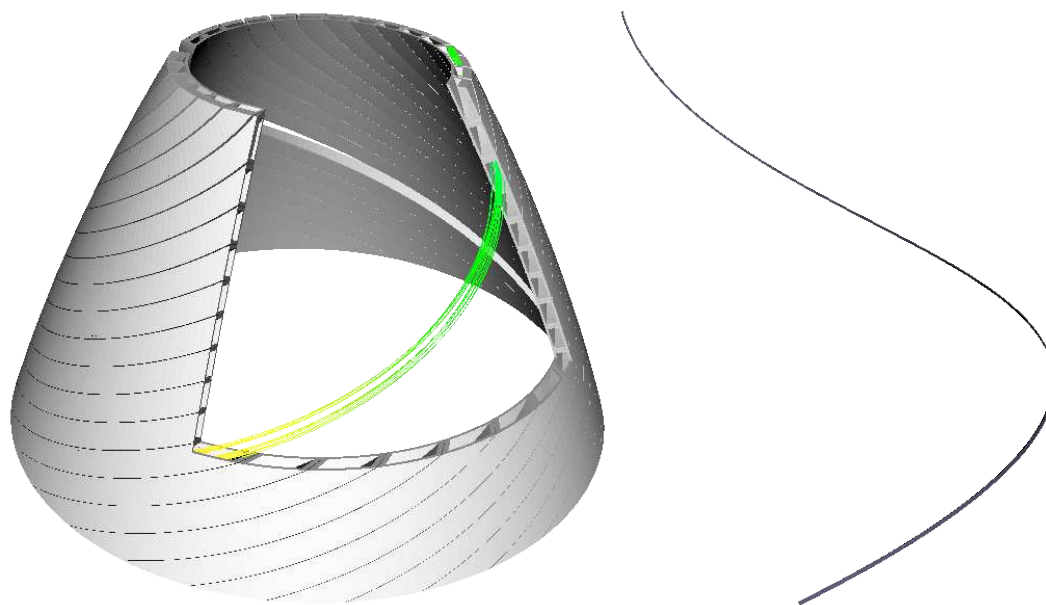


FIG. 6.2 – Schéma simplifié de la tuyère Vulcain II. Par souci de clarté, la dimension de la section des tubes a été multipliée par 10, et 28 tubes (au lieu des 288) ont été dessinés. Les lignes de courant de l'écoulement du fluide sont aussi représentées. À droite, un tube aux proportions réelles.

À la suite de l'accident du vol 157 d'Ariane 5¹ le 11 décembre 2002, la commission d'enquête a établi qu'un des scénarios possibles était une défaillance due à des déformations trop importantes de ces tubes, entraînant à terme leur rupture.

Le 12 février 2005, une version renforcée de la tuyère Vulcain 2 a permis un premier succès du lanceur dans cette configuration 10 tonnes. Les corrections apportées à la tuyère ont consisté en l'ajout d'une barrière thermique sur la face chaude de la tuyère et de raidisseurs placés verticalement à l'extérieur de la tuyère (cf. photo 6.1), ainsi qu'en l'augmentation du débit du gaz de refroidissement. Cette configuration a depuis volé treize fois, le dernier vol remontant au 23 mai 2008.

Cet incident a néanmoins motivé nos recherches concernant le couplage fluide-structure thermique *et* mécanique. En effet, au cours d'essais au sol, ou du vol proprement dit, les tubes sont soumis à différents types de sollicitations, dont voici les trois prépondérantes :

thermiques : la face chaude d'un tube est léchée par les gaz d'échappement, à une température de l'ordre de $2500\text{--}3000\text{ K}$; la face extérieure est à l'air libre, à une température ambiante ; l'intérieur des tubes est refroidi par de l'hydrogène injecté aux alentours de 50 K qui est progressivement réchauffé tout au long de son parcours des tubes.

mécaniques : les différentes faces d'un tube sont soumises aux pressions des gaz qui les entourent ; en particulier, l'hydrogène de refroidissement est injecté à une trentaine de bars de pression. Ils sont aussi soumis à d'autres sollicitations mécaniques, telles que le poids propre du divergent et les vibrations en vol.

¹Premier vol dans sa version "10 tonnes ECA" (Étage supérieur Cryotechnique A).

contact entre les différents tubes : lors de l'assemblage de la tuyère, les tubes sont soudés entre eux sur leur partie extérieure. Après plusieurs cycles d'utilisation², la zone de contact évolue, suite aux déformations des tubes. Ce contact empêche l'expansion par dilatation des tubes, et accentue donc les déformations des zones libres (partie basse de la figure 6.3).

Il est apparu lors d'études précédentes (voir en particulier [63]) que ces trois phénomènes interviennent tous significativement. Il est donc difficile de négliger l'un ou l'autre des phénomènes pour une étude représentative de ce phénomène de déformation progressive des tubes (phénomène de gonflement, aussi appelé "bulging") .

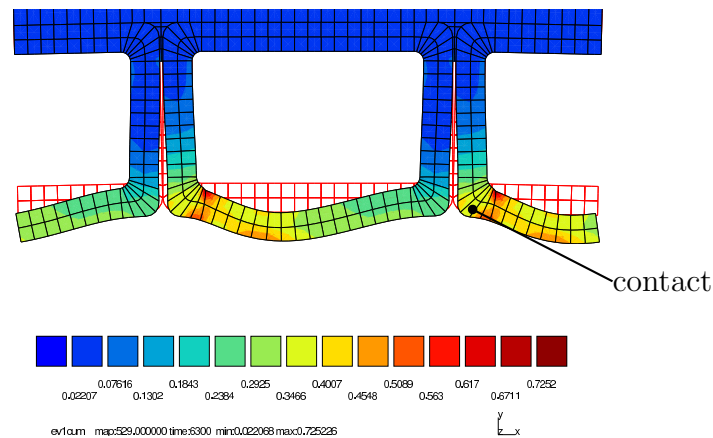


FIG. 6.3 – Gonflement (bulging) d'un tube, image extraite de [63]. Cette figure présente les déformations visco-plastiques cumulées après 10 cycles de 600 secondes, une fois les tubes refroidis. Ces déformations varient entre 2% (bleu foncé) et 70% (rouge foncé). Le champ de température appliqué à ce calcul n'étant pas symétrique, on retrouve cette asymétrie dans les déformations. Le maillage initial est représenté en traits rouges. La flèche pointe vers une zone où les tubes entrent en contact, lorsqu'ils sont chauffés.

L'effet du gradient thermique entre les faces chaude et froide du tube est primordial. Une fois dilatés, le faible espacement initial entre les tubes se referme et les tubes entrent en contact, ce qui accentue encore l'effet de gonflement.

La pression exercée par le gaz de refroidissement sur les parois des tubes engendre principalement du fluage : sa pression crée progressivement une déformation plastique, et la structure est d'autant plus sensible à cette pression que sa température est élevée.

Il est important de noter que les déformations subies par les tubes sont extrêmement complexes, puisqu'elles mettent en jeu des phénomènes de thermo-élasto-visco-plasticité. Bien que les phénomènes soient ici considérés comme quasi-statiques³, les modèles de fluage font intervenir le temps de chargement. Il est donc nécessaire de réaliser des calculs sur une échelle de temps réaliste : dans le cas présent nous choisirons un temps de 600 secondes, qui correspond au cycle d'utilisation du moteur, que ce soit lors des essais au sol ou en vol.

²Après fabrication, une tuyère est testée 2 ou 3 fois au sol, parfois plus, dans des conditions similaires à celles du vol. Des vérifications sont effectuées après chaque test. Par sécurité, la tuyère est dimensionnée pour résister à une vingtaine de tels cycles.

³Rappelons qu'en mécanique des structures, cela signifie que les termes d'inertie en $\frac{\partial^2}{\partial t^2}$ sont négligés dans les équations 1.3.

Par simplicité, nous appellerons à partir de maintenant indifféremment “vol” un des essais au sol ou le *vrai* vol de la fusée. Bien entendu, les essais au sol ne reproduisent pas l’ensemble des sollicitations vues par la tuyère. Ce sont d’ailleurs ces sollicitations supplémentaires délicates à estimer qui ont été identifiées comme l’une des causes de l’accident du vol 157 [2].

Pour cette étude néanmoins, ces deux expériences seront représentées de manière identique. Par contre, comme dit plus haut, une tuyère verra au cours de sa vie plusieurs lancements. On réalisera donc plusieurs cycles de 600 secondes, puisque les déformations (plastiques, donc permanentes) que subit la tuyère à chaque vol se trouvent cumulées.

6.2 Schématisation d’un vol

Les simulations numériques seront réalisées sur un modèle qui ne pourra représenter qu’une partie des efforts subis par la tuyère. Il est bien évident que les contraintes réelles au sol et en vol sont beaucoup plus complexes (systèmes d’accroche différents, vibrations dues aux boosters lors du vrai vol, etc. . .). Aux échelles qui nous intéressent (celle d’un tronçon d’un ou de quelques tubes), on suppose que les deux phénomènes de couplage thermique et mécanique décrits précédemment restent prépondérants ; ce sont les seuls que nous chercherons à représenter.

6.2.1 Éprouvette et discrétisation

Dans les pages qui suivent, nous présenterons quelques simulations sur un tube complet de 2 mètres de long. Cependant, ces calculs se sont révélés extrêmement coûteux en mémoire et en temps de calcul. Ainsi, dans notre première phase de mise au point de stratégies de couplage, nous avons cherché à simplifier de façon significative le problème.

Du tube de 2 mètres présenté sur la figure 6.2, nous avons extrait les 20 premiers centimètres, de la zone d’injection des gaz de refroidissement, jusqu’à 6 cm après la fin de l’insert. Cette zone a plusieurs particularités : tout d’abord, c’est celle qui est soumise aux contraintes thermiques les plus importantes, et qui, en conséquence est la plus susceptible d’être endommagée. La présence de l’insert modifie aussi fortement l’écoulement, et en rend le profil plus intéressant du point de vue du couplage. L’étude découplée de ce tronçon a été possible, le coût de ces calculs fleuretant avec les limites mémoires des machines que nous avons à disposition, sans pour autant nécessiter le recours à des calculs parallèles.

Malheureusement, lors du passage à des calculs couplés nous nous sommes heurtés à des problèmes logiciels, la version que nous avons à disposition de librairie de couplage MpCCI n’étant pas compatible avec les machines 64 bits⁴ à notre disposition. Il n’a pas été possible d’obtenir une version fonctionnant de cette librairie, et les tentatives d’allègement du calcul solide (par déaffinement de maillage) n’ont pas donné satisfaction.

Afin de contourner ce problème, il a donc été décidé de réduire encore ce tronçon aux 2 centimètres situés autour de la fin de l’insert. Il n’est donc ici plus question de prétendre les calculs réalistes par rapport à la situation du tube. Cette géométrie simplifiée permet néanmoins de conserver un certain nombre des particularités de cette situation : section et épaisseur en taille réelle et présence de l’insert. Elle permet aussi de rendre l’ensemble du calcul couplé (solide et fluide) suffisamment léger pour pouvoir tourner sur les machines à

⁴L’architecture 32 bits est limitée à 2 Go de mémoire ; les machines 64 bits peuvent en adresser beaucoup plus, 32 ou 64 Go par exemple.

notre disposition. Cet allègement a aussi eu pour avantage de réduire les temps de calculs, ce qui est toujours appréciable dans une phase de mise au point.

La figure 6.4 présente la géométrie simplifiée qui sera utilisée pour réaliser les calculs couplés.

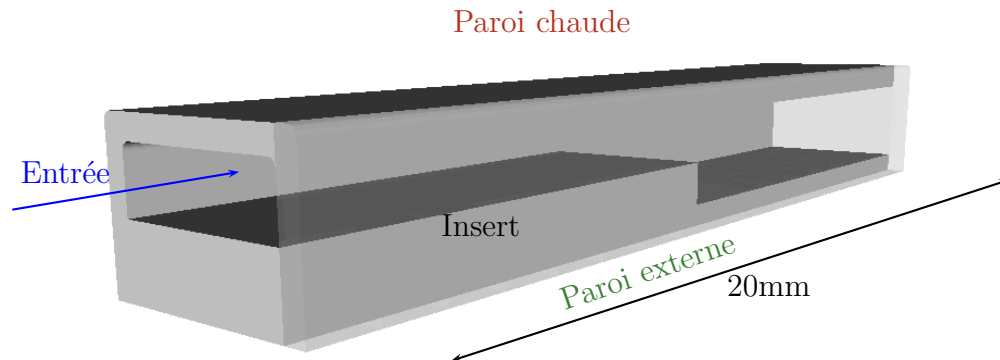


FIG. 6.4 – Géométrie simplifiée retenue pour l'étude couplée

6.2.2 Chargements appliqués

Nous avons par contre cherché à ne pas trop simplifier les différents chargements et paramètres appliqués au tube. En effet, d'un point de vue mécanique, les comportements matériaux qui nous intéressent n'apparaissent que sous fortes sollicitations et à hautes températures : à faible pression, ou à basse température le comportement de l'Inconel 600 est simplement élastique. D'un point de vue thermique, les phénomènes étudiés ne se comportent pas non plus de manière similaire à d'autres échelles. Ainsi, nous avons retenu les conditions suivantes.

Conditions pour le fluide

En début de calcul, il est important d'avoir un état initial à l'équilibre entre le fluide et le solide. Ils sont donc pris tous les deux à une même température de $273 K$. La pression initiale du fluide est de 30 bars, et il est au repos.

À l'entrée du tube (située à gauche sur la figure 6.4), les gaz sont ensuite injectés à une température descendant progressivement (durant première seconde) à $50K$ et à une pression de 33 bars. À partir de la première seconde et jusqu'à la fin du calcul, ces pression et température d'entrée sont maintenues. En sortie de tube, une pression de 27 bars est appliquée.

Enfin, sur les parois en contact avec le solide, diverses conditions seront appliquées suivant la nature du calcul réalisé : découplé, couplé thermiquement ou mécaniquement. Nous les précisons donc en temps voulu.

Conditions pour le solide

Commençons par décrire les conditions mécaniques : la base du tube (paroi externe sur la figure 6.4) est fixée suivant sa direction orthogonale. Les 3 autres faces extérieures sont laissées libres. Les mouvements de corps rigide sont bloqués en imposant un déplacement nul du nœud au centre de la base. Nous forçons aussi les faces d'entrée et de sortie à rester planes,

sans pour autant empêcher leur expansion. La paroi intérieure en contact avec le fluide sera, suivant le calcul réalisé, à pression imposée ou couplée avec le fluide. À l'instant initial, la température de la structure est suffisamment faible pour que la pression exercée par le fluide déforme la structure de manière négligeable.

Concernant les conditions thermiques, la base et les côtés extérieurs du tube voient un flux de chaleur nul : pour les côtés du tube, cela est très probablement assez réaliste, puisque le tube voisin en contact a une température proche. Pour la base du tube, qui est en contact avec l'air libre, nous avons constaté qu'une condition mixte de Fourier pour une température de référence de 273 K n'apporte qu'une différence infime par rapport à ce flux nul. C'est donc une approximation que nous nous autorisons.

La paroi supérieure est en contact avec les gaz chauds à une température de l'ordre de 3000 K . La tuyère n'est par contre pas allumée au début du calcul : cela nous permet d'avoir une situation thermique équilibrée entre le fluide et le solide. Ces gaz sont donc supposés à 273 K pendant la première seconde du calcul, pendant laquelle les tubes sont refroidis. Puis la tuyère est démarrée et chauffe progressivement pour atteindre ces 3000 K au bout de la 6^e seconde.

Une condition mixte de Fourier s'est montrée comme étant la plus adaptée pour cette paroi. La température de référence a bien entendu été prise à 3000 K ; le choix du coefficient d'échange a par contre nécessité des ajustements qui seront discutés par la suite.

Enfin, comme dans le cas mécanique, la condition au bord appliquée à la paroi interne sera précisée ultérieurement.

La figure 6.5 résume les conditions au bord que voient les problèmes thermiques fluide et solide.

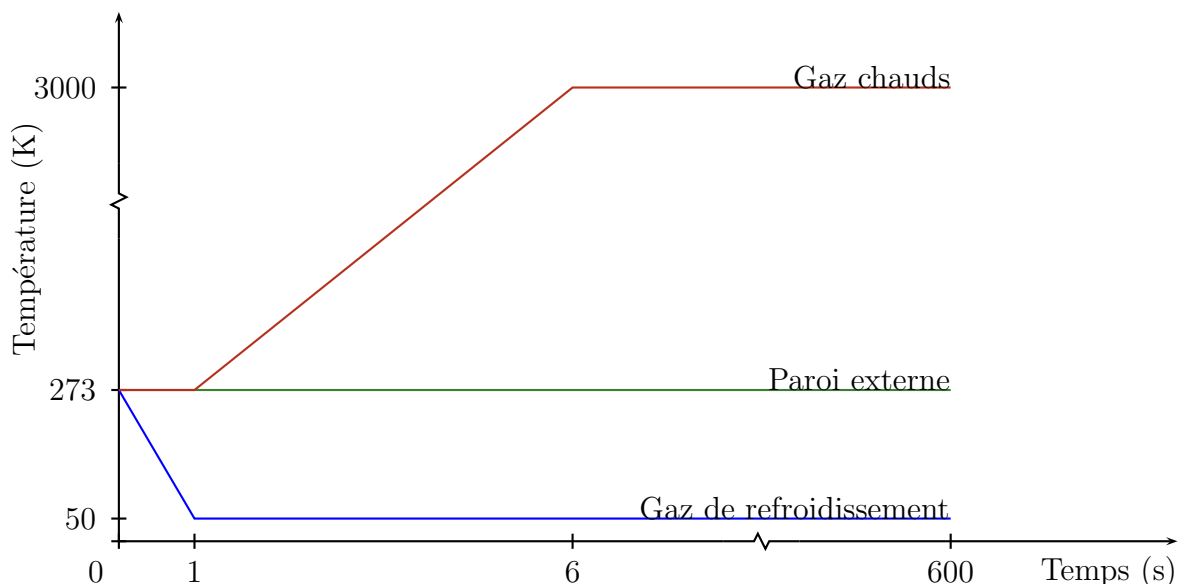


FIG. 6.5 – Schématisation de l'évolution du chargement thermique

6.3 Calculs fluides découplés

La simulation complète d'une tuyère en fonctionnement n'est pour le moment pas envisagée. Par contre, elle fournit un exemple typique des situations que l'on rencontre :

- fonctionnement à températures élevées et variantes,
- différentes sources de sollicitations mécaniques,
- mécanique non-linéaire (non-linéarités géométriques et matériau),
- dépendance de tous les paramètres matériau à la température.

Ces complexités rendent les simulations numériques parfois difficiles à réaliser, et nécessitent un certain nombre de précautions. Nous verrons aussi que les variations brusques de température nécessitent de petits pas de temps pour l'intégration des lois de comportement matériau.

Cette section et la suivante ont donc pour vocation de regarder des cas découplés représentatifs de ce type de calculs, d'en tirer des ordres de grandeur caractéristiques, et d'appréhender les difficultés numériques liées à chacune des physiques.

Bien que présentés avant les calculs couplés, ces cas tests ont été obtenus en même temps que ces derniers. Il est d'ailleurs ressorti qu'il est extrêmement difficile de retranscrire de manière découplée les chargements vus par leurs équivalents couplés.

6.3.1 Conditions de fonctionnement

Précisons les conditions d'entrée du "dump cooling" (c'est-à-dire le système de refroidissement par injection de gaz dans les tubes). Ceux-ci sont injectés à une température d'environ 50 K et une pression d'environ 30 bars. Le débit d'entrée est de $1,9\text{ kg/s}$ pour l'ensemble des 288 tubes⁵.

Les tubes sont alimentés par de l'hydrogène supercritique froid; nous avons grandement simplifié ceci en considérant le gaz comme parfait dans nos simulations. Aux vitesses considérées, son écoulement est turbulent. Nous avons respecté cela puisque c'est ainsi que les échanges et le transport de la chaleur sont le plus efficaces. La turbulence est décrite par un modèle $K - L$. Nous renvoyons à [17] pour les détails concernant ce modèle.

Cette série de calcul étant découplée, il faut aussi choisir des conditions au bord pour les parois en contact avec la structure. Cette température de paroi étant bien évidemment de nature couplée, il est très difficile d'en déterminer une valeur. Ainsi, en l'absence de mieux, nous avons choisi des valeurs constantes par parois :

- 1000 K pour la paroi supérieure (la plus chaude)
- 300 K pour la paroi inférieure
- 600 K pour les parois de côté.

Il ne reste finalement plus qu'à choisir un maillage pour le fluide. Nous avons réalisé divers essais de complexité croissante afin d'en retenir un suffisamment représentatif pour les calculs couplés.

6.3.2 Modèles fluides 1D & 2D

Nous avons envisagé quelques temps un modèle 1D de l'écoulement du fluide, bien évidemment intéressant pour sa légèreté. Ce modèle ne permet néanmoins pas de représenter l'insert qui fait tout l'intérêt de l'écoulement dans le tube. De plus, une unique maille dans l'épais-

⁵Il s'agit là du débit dans la configuration du vol 157; ce débit a été augmenté dans la configuration corrigée.

seur ne permet pas de capter le fort gradient de température entre les faces supérieures et inférieures.

Dans un deuxième temps, un modèle 2D du fluide, considérant l'écoulement uniforme dans la largeur du tube a été étudié. En l'absence des deux parois latérales (c'est-à-dire pour un écoulement infini dans cette direction), ce modèle aurait pu être représentatif. Mais les parois latérales ont justement une influence importante, tant au niveau des échanges thermiques qu'au niveau des déformations subies ; elles doivent donc être prises en compte. De plus, pour la géométries réelle tenant compte de la courbure des tubes, cette hypothèse d'uniformité n'est pas réaliste.

Les ordres de grandeur obtenus pour ces calculs 2D se sont révélés bien différents de ceux obtenus par des modèles 3D. Les calculs réalisés nous ont tout de même permis de trouver quelques ordres de grandeur caractéristiques de l'écoulement. Nous en présentons donc quelques-uns.

Pour ce modèle 2D, nous avons retenu le maillage aux caractéristiques suivantes :

- 1000 mailles dans la longueur,
- 16 mailles dans l'épaisseur,
- une maille dans la largeur.

Les dimensions et la géométrie du tube sont celles de la vraie configuration : une section de taille millimétrique, pour une longueur d'environ 2 mètres. L'insert est aussi pris en compte. Les conditions d'entrée et aux bords sont celles décrites dans la section précédente.

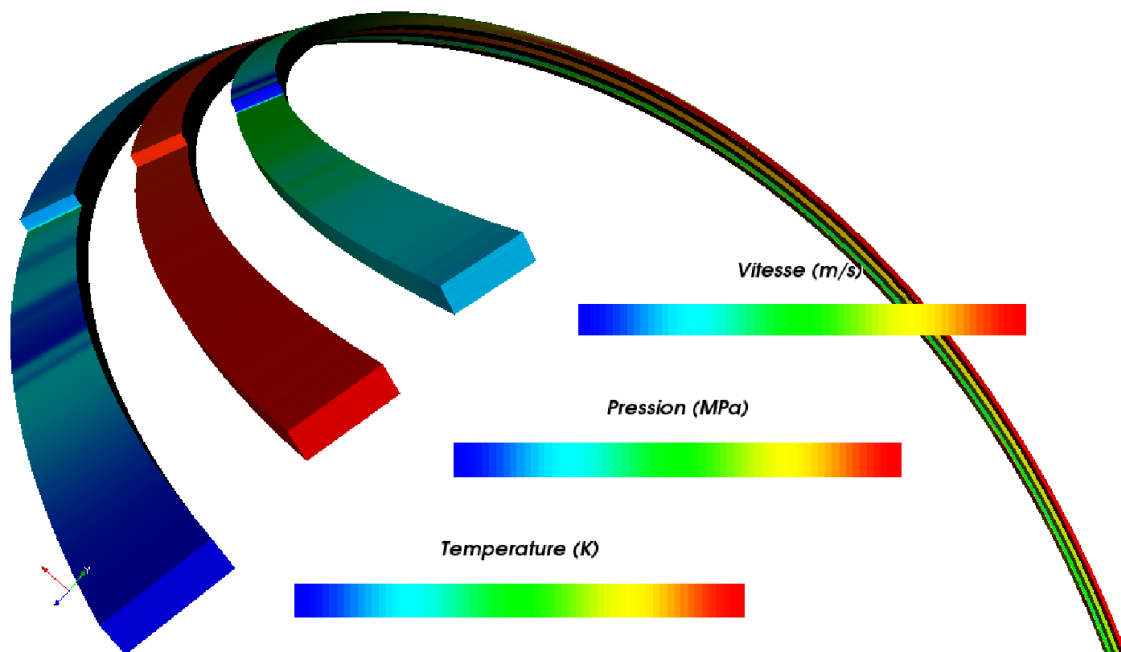


FIG. 6.6 – Température, pression et vitesse calculées sur le modèle 2D

Les champs de pression, température et vitesse obtenus sur cette configuration sont présentés sur la figure 6.6.

Les résultats obtenus sont à considérer avec prudence : la géométrie particulière du tube rend l'obtention d'un état stationnaire assez complexe. La finesse du maillage rend aussi le

calcul assez lourd. Certains points semblent néanmoins représentatifs de ce qu'il faut attendre de ce genre de calcul découplé.

Le champ de pression est beaucoup plus régulier, et présente une chute au niveau de l'insert, liée à la baisse des vitesses à cet endroit. Au-delà de cette zone, la pression diminue régulièrement jusqu'en sortie de tube.

Le profil de vitesse est grandement influencé par la présence de l'insert : au niveau du nez de marche, elle chute à des vitesses presque nulles, et un phénomène de recirculation semble se présenter. Au-delà de l'insert, la vitesse croît progressivement, en raison de l'échauffement du gaz.

6.3.3 Tube fluide 3D

Étant établi qu'un modèle 3D du fluide est indispensable, nous nous intéressons à sa discrétisation. Une discrétisation raisonnable contiendra au minimum une dizaine de mailles dans l'épaisseur et dans la largeur du tube. En raison de la courbure importante du tube, de sa petite section et la longueur du tube ($2m$), un millier de mailles dans la longueur ne sont pas superflues. Les cellules ainsi obtenues sont au nombre de 100 000.

Section élargie

Afin de pouvoir alléger cette taille de maillage, nous avons dans un premier temps réalisé un calcul similaire sur une configuration avec une section du tube multipliée par 10 (telle que présentée sur la figure 6.2). Une centaine de cellules dans la longueur est alors suffisante pour obtenir un maillage satisfaisant. La visualisation est aussi plus aisée sur cette configuration moins élancée.

Ce calcul met en évidence deux des phénomènes particuliers à l'écoulement dans ce type de tubes courbes. Tout d'abord, la présence de l'insert a une influence importante sur l'écoulement : cela est illustré ici par le champ de température ; les champs de vitesse et de pression sont aussi très irréguliers au voisinage de la fin de l'insert : ils y subissent tous les deux une chute de magnitude.

Deuxièmement, on constate une asymétrie de l'écoulement dans la section du tube : comme on pouvait s'y attendre, la température est plus élevée sur la paroi chaude que sur la paroi froide. Mais nous trouvons aussi un effet tridimensionnel dû à la géométrie courbe des tubes : la température est beaucoup plus élevée dans le coin supérieur gauche que dans les autres coins. Ces "coins chauds" ont effectivement été observés et mesurés sur les configurations réelles de la tuyère.

Il faut néanmoins rester prudent sur les interprétations de ce calcul : la géométrie élargie entraîne certainement une modification importante sur l'écoulement. En particulier, les ordres de grandeur des vitesses, pressions et températures ne devront pas être pris comme représentatifs de la configuration réelle.

Dimensions réelles

Sur la configuration réelle, il est nécessaire de placer au minimum 1000 mailles dans le sens de la longueur, afin de bien suivre la courbure du tube. Le coût du calcul devient très important : chaque itération nécessite environ 250 secondes de calcul, contre 2 secondes pour le modèle 2D présenté précédemment. Notons tout de même que ces calculs ont été réalisés sur une machine scalaire, qui n'est pas la mieux adaptée au logiciel MSD.

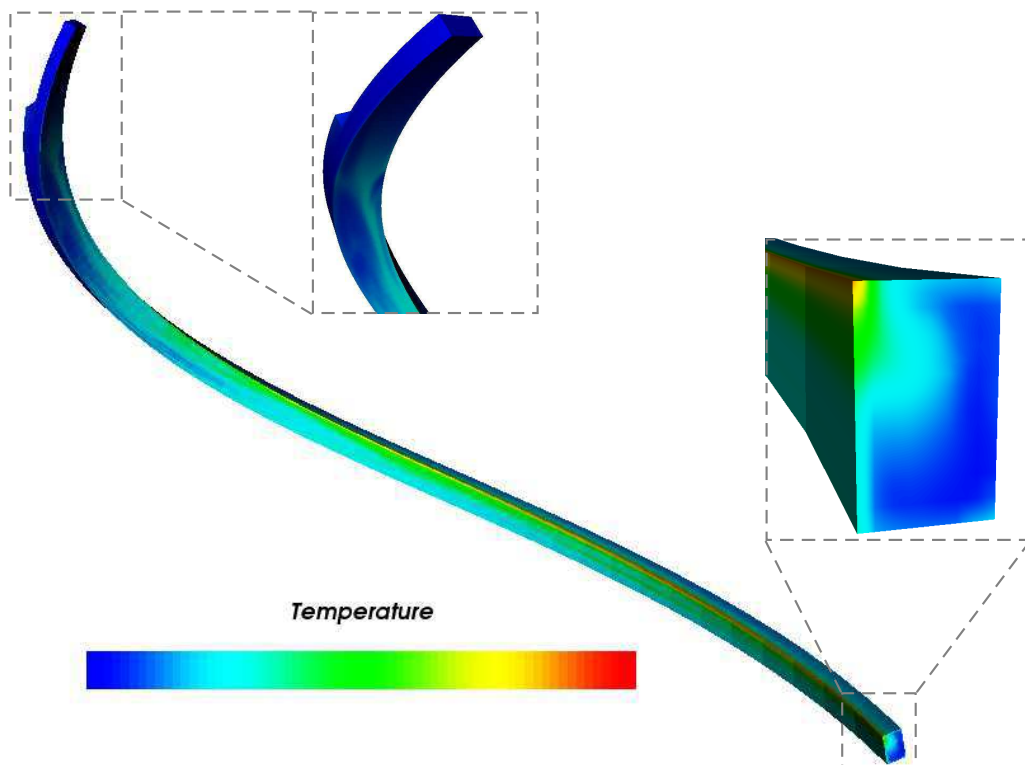


FIG. 6.7 – Écoulement fluide sur une configuration 3D élargie : pour alléger le calcul et faciliter la visualisation, la section des tubes a été multipliée par 10.

Au bout d'un mois de calcul, et 10000 itérations, le résultat n'est pas encore convergé. Nous pouvons cependant présenter les champs intermédiaires obtenus sur cette configuration réelle, sur la figure 6.8.

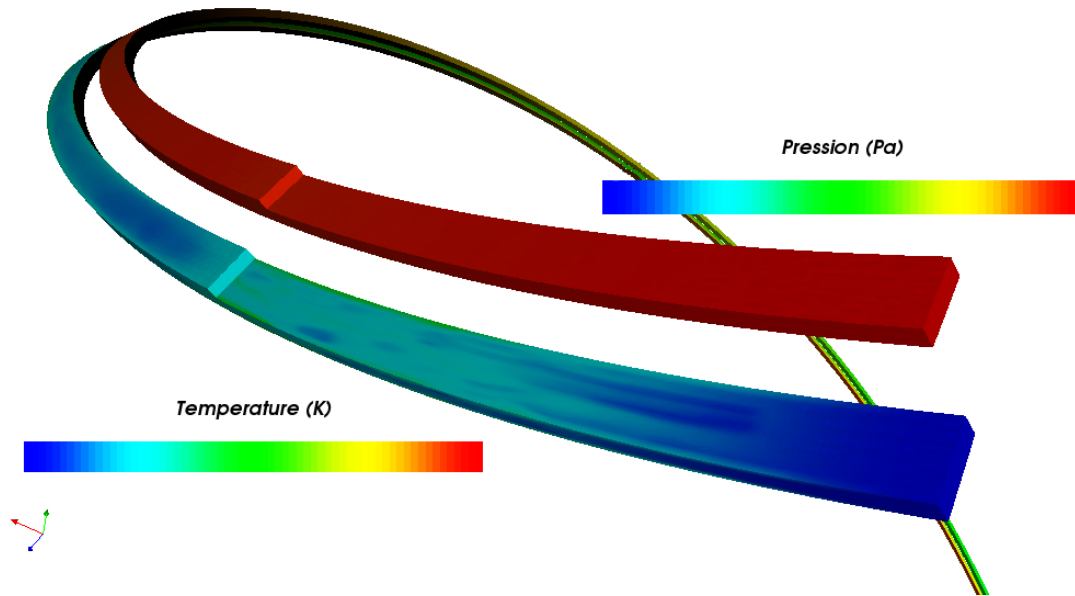


FIG. 6.8 – Écoulement dans un tube complet, aux dimensions réelles.

Ces résultats retrouvent les différentes caractéristiques des calculs précédents : les températures maximales locales correspondent à celles obtenues par le modèle 2D. Les effets de la courbure sur l'écoulement sont tout aussi visibles que dans le cas à section élargie. Cette courbure induit des différences de température importantes entre différents points d'une même section de tube.

Les différents résidus extraits de ce calcul indiquent que l'état stationnaire n'est pas encore atteint : il subsiste une différence entre les masses entrant et sortant du domaine, et les valeurs locales des champs (température, vitesse, pression) continuent d'évoluer fortement. Vu son coût, il serait nécessaire d'employer une stratégie de parallélisation afin d'en accélérer la résolution. N'ayant pas pour objectif de réaliser un tel calcul, nous nous sommes donc contentés de ces résultats partiels.

Il est pour le moment trop lourd de coupler ce calcul au solide ; pour cela, nous avons isolé les 20 premiers centimètres du tube, et réalisé un calcul aussi proche que possible en réinjectant les informations du calcul complet.

Restriction aux 20 premiers centimètres de tube

Sur ce cas, les conditions d'entrée sont inchangées, mais celles de sortie sont cette fois choisies pour une pression d'arrêt correspondant approximativement à la pression relevée sur le calcul complet.

Les figures 6.9 et 6.10 présentent respectivement les champs de température et de vitesse obtenus sur cette configuration. Les effets de la courbure sur le champ de température sont

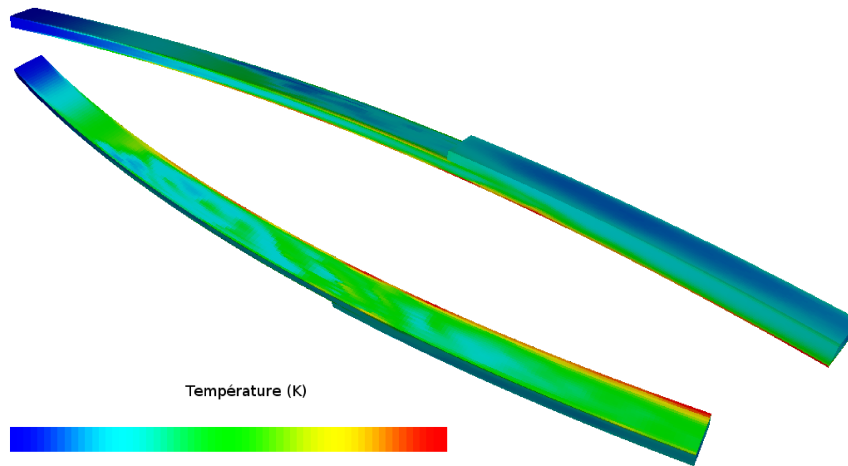


FIG. 6.9 – Écoulement dans les 20 premiers centimètres du tube : champ de température

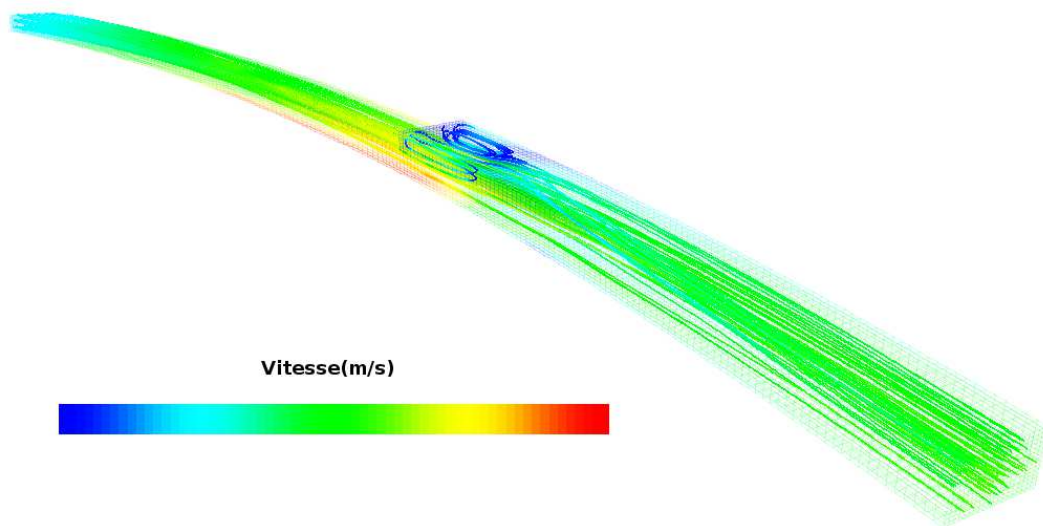


FIG. 6.10 – Écoulement dans les 20 premiers centimètres du tube : lignes de courant et vitesse du fluide.

importants : la première rangée de cellules le long de la paroi chaude, dans le coin déventé atteint des températures environ 1,5 fois plus élevées que celle dans situées dans l'autre coin.

L'effet de l'insert est flagrant sur le champ de vitesse. Une zone d'accélération est créée en amont de la marche, sous un phénomène d'aspiration. À l'abri du nez de marche se trouve par contre une zone où les vitesses deviennent presque nulles, y entraînant un phénomène de recirculation.

Le champ de pression est lui aussi directement influencé par la géométrie du tube. La courbe 6.11 présente le champ de pression relevé au milieu de la paroi chaude (paroi inférieure sur la figure 6.10).

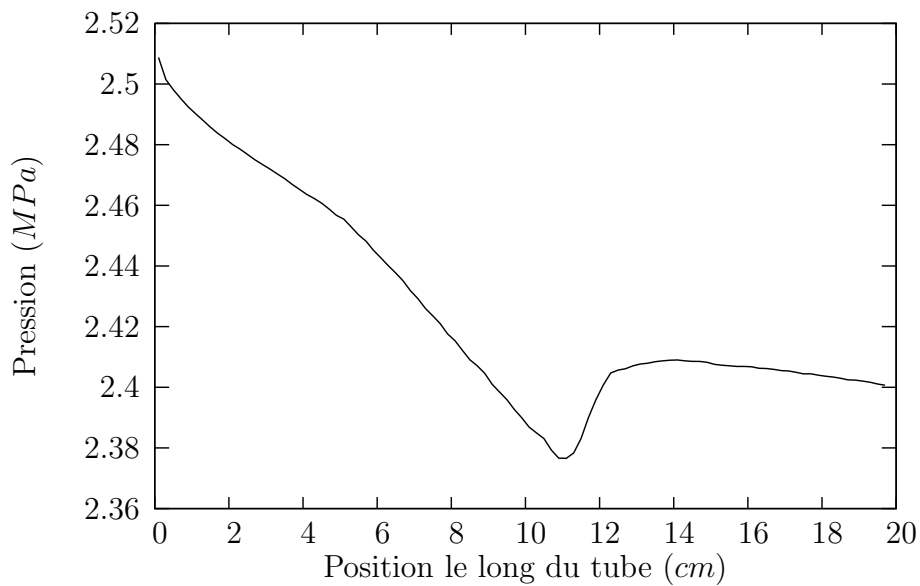


FIG. 6.11 – Profil de pression le long du tube

6.4 Calculs mécaniques découplés du solide

6.4.1 Loi de comportement de l'Inconel 600

L'Inconel 600 est l'alliage à base Nickel constituant les tubes du divergent de Vulcain 2. C'est un matériau choisi pour sa bonne résistance aux hautes températures (point de fusion à 1686 K). Il présente aussi une bonne résistance à l'oxydation et à la corrosion, qui motive son utilisation dans de nombreuses autres applications telles que le génie nucléaire et la construction de fours.

Lors de son utilisation dans le contexte de la tuyère, il est soumis à de très fortes températures et pressions pendant des périodes relativement longues, et subit par conséquent des déformations thermo-visco-plastiques. Sa loi de comportement suit le modèle décrit dans la section 1.2.1, étalonnée au cours de l'étude [63]. Nous présentons ce matériau plus en détails dans l'annexe B. Tous les calculs mécaniques (découplés, puis couplés) qui suivent utilisent cette loi de comportement.

6.4.2 Calculs plans

Cette première série de calculs sur une section du tube nous permet d'établir un choix concernant la nature et la finesse du maillage.

Ces calculs sont faits sous l'hypothèse de grandes déformations planes. Partant d'une température initiale uniforme de 21°C , celle-ci est régulièrement augmentée pendant les 6 premières secondes pour atteindre un gradient entre 1000 K et 1300 K , tel que présenté sur la figure 6.12-a. Une pression uniforme est appliquée progressivement à l'intérieur du tube, et est maintenue à partir de la 6^e seconde à 3 MPa , jusqu'à la fin du calcul à la 600^e seconde. Enfin, l'extérieur du bord inférieur est fixé afin d'empêcher les mouvements de corps rigides.

Plusieurs séries de maillages ont été éliminées pour différentes raisons ; la figure 6.12 présente un calcul typique de chacun de ces cas éliminés, mettant en évidence leurs défauts.

Tout d'abord, les éléments linéaires (Fig. 6.12-b) ont été éliminés au profit d'éléments quadratiques : les éléments linéaires ont en effet le défaut de se verrouiller en flexion, ce qui est le type de déformations que subit le tube.

Dans un deuxième temps, le maillage retenu a été affiné dans les coins subissant les déformations : la figure 6.12-c montre par exemple que la déformation plastique cumulée est sous-évaluée (0,36 au lieu de 0,6) si le maillage n'est pas suffisamment fin.

Enfin, ayant constaté sur les figures 6.12-c/d que les deux éléments les plus proches des coins sont distordus, nous avons arrondi la géométrie afin de mieux respecter la géométrie réelle des tubes et d'éviter l'apparition de concentrations de contraintes dans ces zones fortement déformées. Cette géométrie arrondie nous a naturellement conduits à abandonner les éléments quadrangulaires au profit d'éléments triangulaires, plus faciles à mailler.

Nous avons finalement retenu des éléments à intégration réduite, pour lesquels les résultats sont très semblables à ceux obtenus par intégration complète, mais qui permettent un gain important de temps de calcul. La loi de comportement étant très complexe, ce gain est notable ; il le serait moins dans des applications élastiques. Pour donner un ordre d'idées, le calcul 6.12-f nécessite 1700 secondes en intégration complète contre seulement 700 secondes en intégration réduite. Dans les deux cas, l'inversion des systèmes linéaires coûte 150 secondes.

FIG. 6.12 – Maillages et géométries éliminées : (a) champ de température imposé ; un maillage en quadrangles linéaire (b) se comporte mal en flexion (phénomène de verrouillage) par rapport aux maillages quadratiques (c-f) ; les déformations plastiques cumulées sont sous évaluées sur un maillage trop grossier (c) ; (e) maillage de "référence" quadratique ; (f) maillage "allégé" quadratique retenu pour la suite de l'étude.

Ces essais sont aussi l'occasion de regarder les parts respectives des déformations thermiques, élastiques et de fluage. Les courbes de la figure 6.13 montrent par exemple que les déformations subies pendant les premières secondes (dues principalement à la partie élastique et thermique) sont minimales par rapport à celles vues pendant le reste du calcul. Nous sommes donc dans une situation où les déformations par fluage sont prépondérantes.

Une conséquence directe de cela est la sensibilité du résultat à la pression subie. Différentes pressions appliquées (26, 28 et 30 bars, correspondant à la fourchette fournie par les calculs fluides), fournissent des flèches au milieu du tube variant d'environ $0,1\text{ mm}$ à chaque palier, ce qui représente environ 15% des déplacements subis. Cette sensibilité est directement liée à la limite d'élasticité presque inexistante à ces températures proches de la température de fusion.

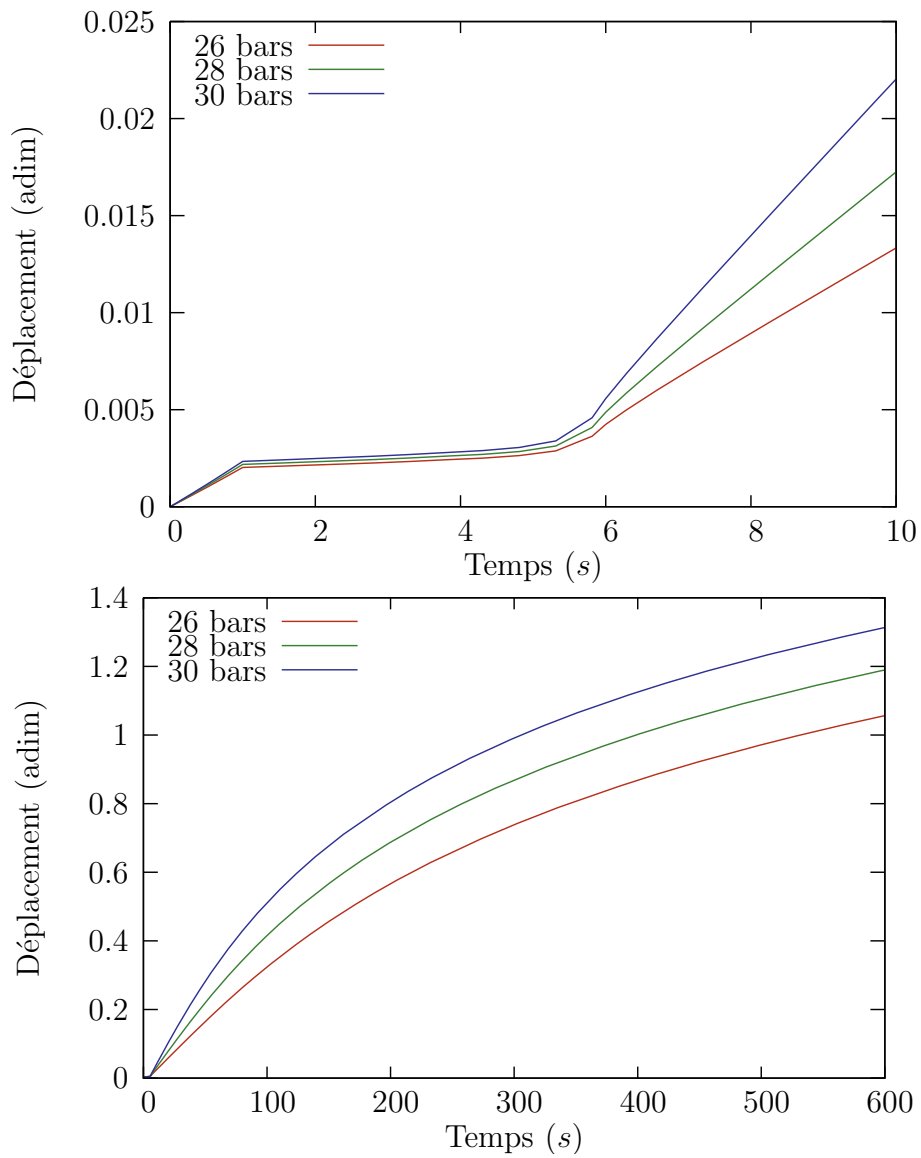


FIG. 6.13 – Déplacement (flèche) au milieu du tube, pour différentes pressions uniformes.

6.4.3 Calculs 3D

La géométrie et le maillage présentés sur la figure 6.12-f ont été finalement retenus. Ils sont simplement extrudés suivant la direction z pour fournir un maillage 3D d'un tube rectiligne. Nous aurons donc à faire à des éléments de type prisme, pour lesquels nous continuons à choisir une intégration réduite. Au vu des conditions de chargement appliquées, nous nous attendons à une dépendance en z beaucoup moins marquée des inconnues du problème (déplacements, contraintes, déformations). Cela nous autorise un fort élancement des éléments finis dans cette direction. Notons aussi que, comme dans le cas des calculs fluides, il n'aurait pas été possible d'utiliser aussi peu d'éléments dans la longueur si nous n'avions pas supposé la géométrie rectiligne.

Ainsi, pour obtenir le maillage d'un tube de 20 cm de long, seuls 100 éléments sont placés dans la longueur du tube. Le maillage résultant contient tout de même 259 506 nœuds, 86 800 éléments et 778 518 degrés de liberté. Pour donner un ordre de grandeur, le calcul résultant nécessite environ 11 Go de mémoire.

Les conditions aux bords appliquées sont similaires à celles utilisées précédemment pour les calculs plans : nous imposons en plus que les faces d'entrée et de sortie restent planes, au moyen de contraintes multi-points, c'est-à-dire une relation linéaire entre degrés de liberté. À l'intérieur du tube, une pression de 30 bars est appliquée. La mise en données complète est présentée dans l'annexe D.1.

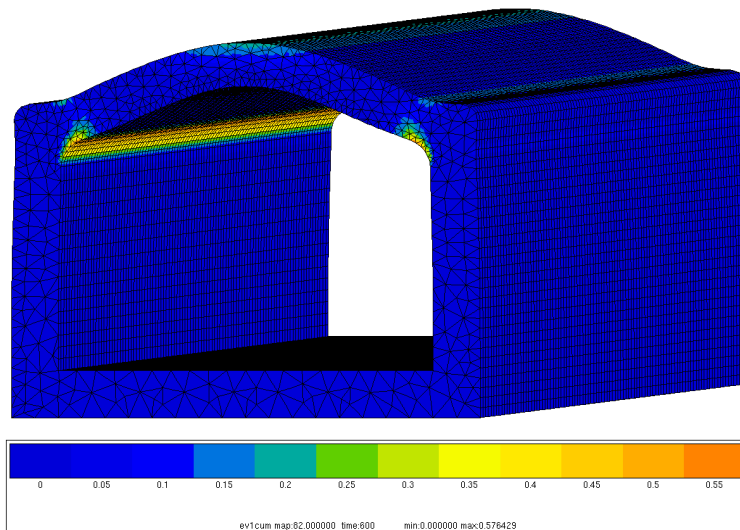


FIG. 6.14 – Calcul 3D sur un tube de 20 cm de long : les déformations plastiques cumulées sont représentées.

Les résultats obtenus sont tout à fait comparables à ceux obtenus dans le cas 2D.

6.4.4 Calculs 3D sur un tube raccourci

Dans la phase de développement de méthodes couplées, il n'est pas envisageable de considérer un calcul à l'échelle précédente. En effet, pour réaliser en un temps raisonnable le calcul précédent, il sera nécessaire de mettre en place des stratégies de parallélisation par décomposition de domaines, à la fois du côté solide et du côté fluide. Ces méthodes seront

particulièrement bien adaptées à cette géométrie très simple. Cependant, elles apportent des difficultés supplémentaires, que nous ne voulons pas gérer dans l'état actuel.

C'est pourquoi nous raccourcissons encore le tube, cette fois pour atteindre une longueur de 3 centimètres. Seulement 16 éléments seront placés dans la longueur du tube. Cela fera à nouveau des éléments très allongés, mais au vu de la faible variation des inconnues dans cette direction, cela ne sera pas handicapant. Les conditions aux bords restent inchangées par rapport au cas précédent.

Les résultats obtenus sont présentés sur la figure 6.15. Nous retrouvons des résultats conformes aux modèles 2D : les déplacements au centre du tube sont de l'ordre de l'épaisseur de la paroi. Les déformations plastiques cumulées sont toujours concentrées dans les coins du tube où elles atteignent 60%, ainsi qu'au sommet du tube, où elles sont de l'ordre de 20%.

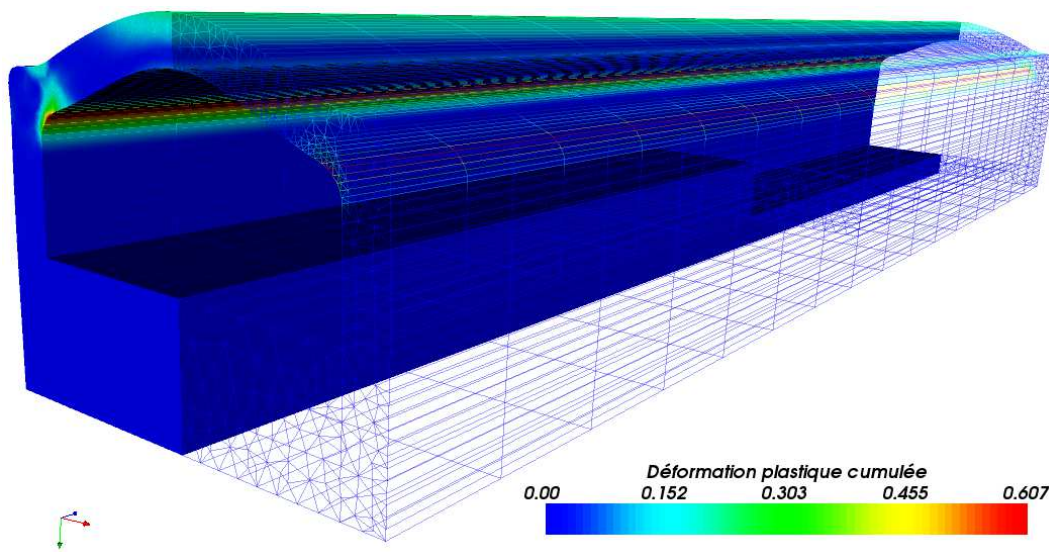


FIG. 6.15 – Tube déformé sous une pression de 30 bars, à la 600^e seconde.

Il est intéressant de constater sur la courbe 6.16 que la présence de l'insert dans la première moitié du tube rigidifie légèrement cette première zone.

6.5 Calculs thermiques découplés du solide

6.5.1 Loi de comportement thermique

Les paramètres de comportement thermique de l'Inconel 600 sont tirés de [66]. La conductivité et la capacité calorifique sont dépendantes de la température, rendant le problème thermique non-linéaire. Ces non-linéarités sont cependant peu importantes, et ne compliquent pas énormément le comportement par rapport à une loi de diffusion à coefficients constants.

Les détails de ce comportement sont regroupés dans l'annexe B.

6.5.2 Calculs 2D

Rappelons les conditions au bord utilisées pour le calcul thermique. Pour la paroi externe et les deux parois latérales, une condition de flux nul a été choisie.

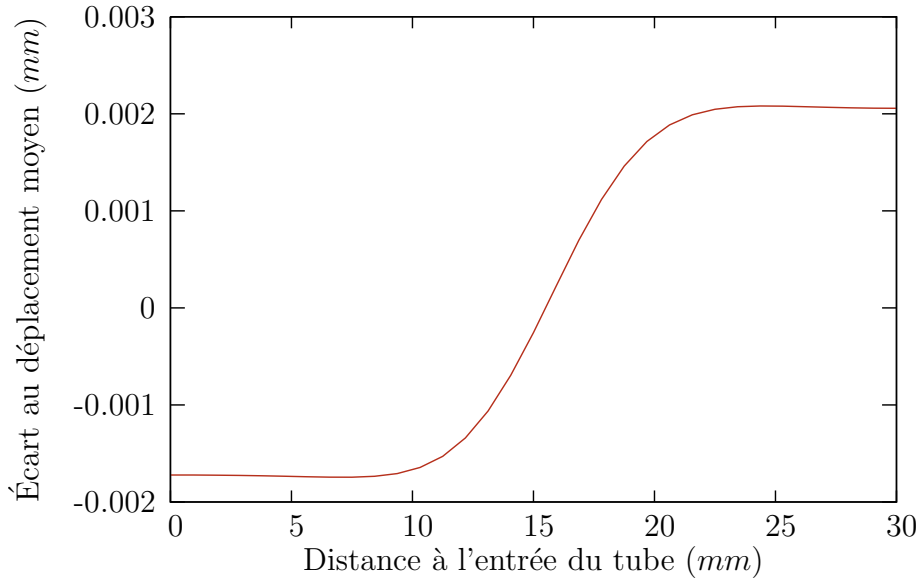


FIG. 6.16 – Profil du déplacement vertical le long de l’axe principal du tube, sous une pression de 30 bars, à la 600^e seconde. Dans la première moitié du tube, où se trouve l’insert, le tube se déforme légèrement moins que dans la deuxième moitié. Pour une meilleure clarté, la courbe présente l’écart au déplacement moyen.

La paroi en contact avec les gaz brûlés est modélisée par une condition de Fourier avec une température de référence de 3000 K et un coefficient d’échange que nous recalons afin d’obtenir des températures maximales de l’ordre de 1300 K . Des essais avec une condition de Dirichlet imposant ces 1300 K n’ont pas donné satisfaction, de même que des essais imposant le flux de chaleur sur cette paroi.

Enfin, la paroi interne est en contact avec le fluide. Pour ces calculs découplés, nous la représenterons par une condition de Neumann, avec des flux en paroi tirés des calculs fluides découplés de la section 6.3.

Il nous faut aussi déterminer un maillage ; au vu des gradients de température importants attendus, celui-ci devra être suffisamment fin. Par contre, les problèmes thermiques ne souffrant pas de problème de verrouillage et nous pourrions utiliser un maillage linéaire. Vu les limitations de la bibliothèque MpCCI (celle-ci ne gère pas le couplage volumique par éléments quadratiques), il est en pratique préférable de se restreindre à des éléments linéaires.

La géométrie devant être coïncidente avec celle du calcul mécanique, nous devons conserver la majeure partie de ses particularités. Par commodité, nous allons repartir des maillages testés pour le calcul mécanique, et les adapter au cas thermique. Ainsi, le maillage quadratique “de référence” suffisamment fin sera comparé au maillage linéarisé retenu pour la mécanique, afin de voir si ce maillage linéarisé est acceptable pour le calcul thermique.

La figure 6.18 compare les résultats obtenus sur ces deux maillages, pour les conditions aux bords décrites ci-dessus. Une condition de Fourier est à la fois appliquée pour la paroi chaude (paroi supérieure extérieure sur la figure) et pour la paroi intérieure. Pour la paroi chaude, la température de référence est de 3000 K ; pour la paroi froide, elle est de 50 K . Dans les deux cas, le coefficient d’échange retenu est de $1000\text{ W}\cdot\text{m}^{-2}\cdot\text{K}^{-1}$.

On constate que les deux calculs fournissent des résultats quasiment identiques. La dif-

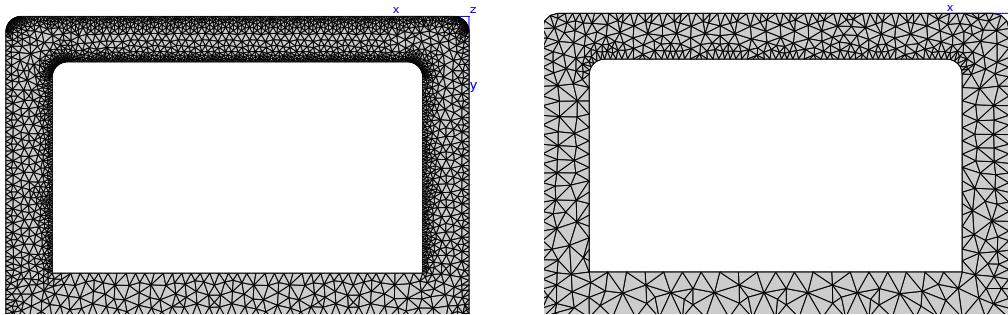


FIG. 6.17 – Maillages 2D de référence et retenu d'une section de tube : le premier a 10 608 nœuds et 4844 éléments quadratiques, le deuxième a 569 nœuds et 868 éléments linéaires.

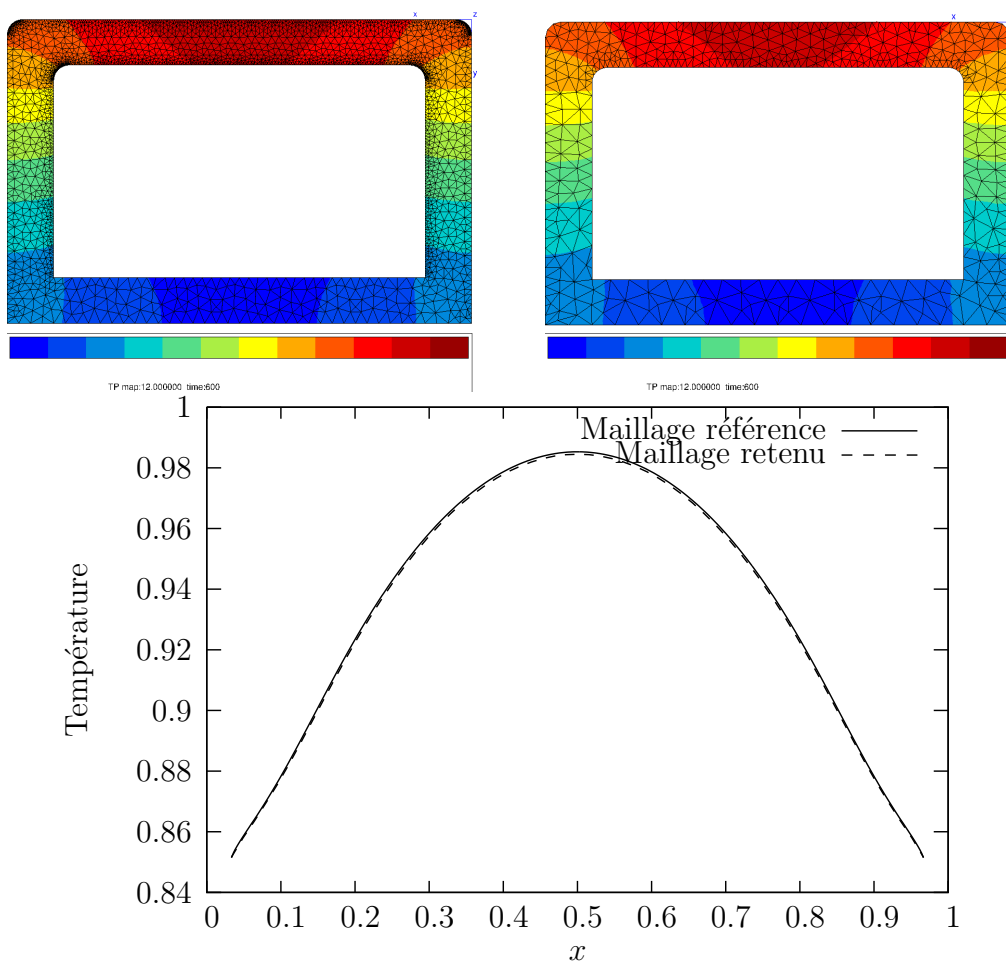


FIG. 6.18 – Comparaison des champs thermiques obtenus pour les deux maillages précédents. Le graphique compare les deux températures sur la paroi chaude (paroi supérieure) à $t = 600\text{ s}$; on constate que la différence entre les deux résultats y est inférieure à 2 K .

férence entre les deux résultats est inférieure à $2 K$ sur l'ensemble du domaine, ce que l'on peut considérer comme une erreur acceptable.

6.5.3 Calculs 3D

Le passage au calcul 3D se fait simplement par extension du maillage 2D à la 3D ; Les éléments résultants sont donc des prismes linéaires, à intégration complète. L'insert est ajouté dans la première moitié du tube.

Les résultats obtenus sont présentés sur la figure 6.19. Les conditions appliquées sont simplement l'extension des conditions appliquées dans le calcul 2D précédent : une condition mixte est appliquée sur la paroi supérieure ($T_{\text{ref}} = 3000 K$, $h = 1000 W.m^{-2}.K^{-1}$), de même que sur la paroi interne ($T_{\text{ref}} = 50 K$, $h = 1000 W.m^{-2}.K^{-1}$). On applique une condition de flux nul sur les autres parois.

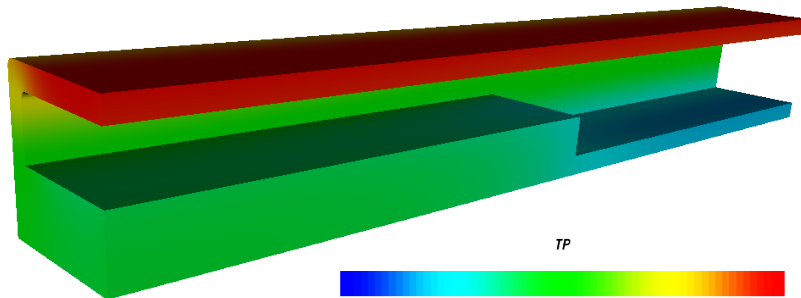


FIG. 6.19 – Champ thermique 3D obtenu par un calcul découplé

On constate que le champ obtenu dans la deuxième moitié du tube est très similaire à celui obtenu par le calcul plan : la température maximale le long de la paroi chaude est de l'ordre de $1200 K$, alors que la température minimale est autour de $300 K$. Par contre, dans la première moitié du tube, l'insert fournit un surcroît de matière à refroidir, et la température moyenne dans cette zone est donc plus importante, aux alentours de $600 K$.

Ce calcul découplé montre un phénomène exactement à l'inverse de la situation réelle : l'insert est placé afin d'accélérer le fluide, et donc d'en améliorer les capacités de refroidissement. Si l'on voulait représenter cela dans ce calcul découplé, il serait nécessaire de modifier les conditions au bord, en utilisant par exemple une température locale de référence du fluide (au lieu de ces $50 K$, moyenne grossière de la température dans les 20 premiers centimètres du tube), ou encore un coefficient d'échange local reflétant le meilleur refroidissement dans les premiers centimètres du tube.

6.6 Bilan

Ces quelques exemples découplés ont permis de mettre en données les différents calculs que nous couplerons par la suite. Un calcul sur un tube complet montre un coût important, que ce soit en temps calcul pour le fluide, ou en mémoire pour le solide. Sans avoir recours à des méthodes de parallélisation, il est toutefois possible de réaliser des calculs acceptables sur une vingtaine de centimètres, sur les machines 64 bits actuelles. Cependant, n'ayant pas eu la possibilité de lancer de calculs couplés sur de telles machines, nous nous sommes restreints

à des calculs sur une géométrie réduite aux 2 centimètres autour de la fin de l'insert. Cette géométrie restreinte s'est finalement révélée très bien adaptée aux calculs de mise aux points.

Les calculs fluides ont montré l'impossibilité de se restreindre à des modèles 1D ou 2D : l'écoulement voit des variations suffisamment importantes dans la largeur du tube pour nécessiter de les prendre en compte par un maillage dans cette direction.

Les calculs mécaniques du solide ont permis de déterminer un maillage raisonnable pour la suite du projet, compromis entre légèreté et précision. Nous retrouvons aussi les différents principes de ce type de simulation : le comportement de l'Inconel est très sensible à la température. Aux hautes températures auxquelles il est soumis, ce sont principalement des déformations visco-plastiques de fluage qui entrent en jeu.

Concernant enfin les calculs thermiques, nous avons pu calibrer une condition de paroi chaude, permettant de retrouver des températures de fonctionnement de l'ordre de celles effectivement relevées dans la situation réelle. Cela reste cependant une représentation très grossière des conditions réelles, beaucoup plus complexes puisqu'à la fois couplées aux gaz de refroidissement et aux gaz chauds d'échappement.

Chapitre 7

Calculs couplés mécaniques

Dans ce chapitre, nous allons spécialiser l'algorithme de Dirichlet→Neumann présenté précédemment au cas de la simulation mécanique d'un tube constituant le divergent de la tuyère. Nous verrons que dans cette situation, il est possible de tenir compte des particularités des phénomènes mis en jeu, afin de simplifier et donc d'accélérer l'algorithme de couplage.

7.1 Calcul couplé mécanique des sollicitations d'un tube de tuyère

Avant de pouvoir définir un algorithme de couplage mécanique, il est primordial de bien définir les objectifs de la simulation. En effet, suivant les phénomènes qui nous intéressent, diverses questions entrent en jeu. C'est une fois de plus l'illustration d'"un algorithme de couplage pour chaque calcul couplé".

Dans le cas des sollicitations mécaniques subies par les tubes de la tuyère, nous avons pu voir, au travers des simulations découplées, que les principaux phénomènes entrant en jeu sont de deux natures. Tout d'abord, durant les 6 premières secondes du calcul, le matériau est à une température suffisamment froide pour avoir une réponse thermo-élastique ; autrement dit, les seules déformations subies sont liées aux variations de pression et de température vues par le tube. Puis, pendant les 594 secondes restantes, les conditions aux bords s'étant stabilisées, les déformations sont principalement dues au fluage du matériau. Ce sont donc ces deux phases que l'on désire capter précisément au cours du calcul couplé. Notons que bien qu'il s'agisse d'un calcul mécanique, les résultats sont dépendants de la température, intervenant comme paramètre de la loi de comportement.

Dans son utilisation de gaz de refroidissement, la principale qualité du fluide est sa capacité d'extraction de chaleur, capacité non étudiée pour le moment dans ce couplage uniquement mécanique. D'un point de vue mécanique, notons toutefois l'importance des variations de la pression sur les déformations observées. Les courbes 6.13 ont en effet montré qu'une variation d'une dizaine de pourcents de la pression entraîne des différences de déplacements d'une vingtaine de pourcents. Les résultats des calculs fluides découplés ont justement montré que de tels écarts de pression sont observés en différents points du tube (Fig. 6.11), ainsi qu'au cours de l'évolution des conditions d'entrée (Fig.6.5).

Ainsi, il apparaît important de bien représenter d'une part la mise en régime du moteur et d'autre part l'ensemble des 600 secondes de calcul. L'évolution des différents phénomènes

est brutale pendant la première partie du calcul. Elle nécessitera donc une discrétisation temporelle relativement fine, ceci indépendamment de toute question de couplage. La deuxième partie a une évolution beaucoup plus importante, mais aussi beaucoup plus lente. Elle peut donc être discrétisée de manière plus grossière.

En pratique, ayant à disposition une méthode de pas de temps adaptatif dans ZéBuLoN, nous nous fierons à celle-ci. Étant donné que les pas de temps sont choisis afin de limiter les variations trop importantes des variables du problème mécanique, ils sont relativement bien adaptés pour le couplage. Ils sont même plus conservatifs que nécessaires pour le couplage. En effet, les quantités matériaux internes telles que les déformations plastiques cumulées subissent de fortes variations, qui n'engendrent que de faibles variations des déplacements de l'interface couplée.

La discrétisation temporelle résultante pour le calcul couplé sera donc très similaire à celle nécessaire pour le calcul découplé : des pas de temps entre 0,1 et 0,5 s pendant les 6 premières secondes, puis des pas de temps pouvant aller jusqu'à 10s dans la deuxième partie du calcul.

Côté fluide, il sera donc au minimum nécessaire de produire des champs pariétaux à ces différents pas de temps de couplage. Une simulation complètement explicite nécessitant des pas de temps de l'ordre de 10^{-3} s, le coût d'une simulation complète des 600 secondes de vol deviendrait prohibitif ; nous ne pouvons actuellement pas nous la permettre. De toutes façons, elle ne semble pas nécessaire. En effet, les mouvements des parois couplées sont suffisamment lents, et le solide est peu sensible aux variations instantanées du fluide. Nous nous autoriserons donc l'hypothèse de stationnarité des états fluides lors des différents points de couplage. Cette hypothèse permet l'utilisation d'algorithmes de résolution simplifiés et plus rapides que ceux présentés au chapitre 2. Entre deux états stationnaires calculés, les champs fluides seront au besoin interpolés linéairement.

Sous cette forme, le calcul fluide n'impose aucune contrainte quant à la discrétisation temporelle couplée. Nous la déterminerons donc entièrement à partir de celle du calcul mécanique.

7.2 Modèles retenus

Les calculs mécaniques découplés présentés dans le chapitre précédent nous fournissent une base aux calculs couplés que l'on veut réaliser ici. L'intérêt du couplage partitionné est que l'on peut reprendre pratiquement telles quelles les mises en données précédentes et simplement les enrichir des commandes de couplage.

En pratique, ce passage est bien entendu plus complexe : les maillages déterminés séparément pour le solide et le fluide ne sont pas suffisamment coïncidents pour que la bibliothèque MpCCI puisse déterminer une correspondance propre entre les maillages. C'est au niveau des coins légèrement arrondis côté solide qu'il y a quelques problèmes. Il a donc été nécessaire d'arrondir le maillage fluide pour mieux épouser ces coins.

Par contre, la modélisation de chacune des physiques, les conditions au bord et les paramètres ont pu être conservés à l'identique, à l'exception bien entendu des surfaces couplées.

À cause de limitations sur la taille mémoire maximale autorisée pour un calcul couplé, nous avons dû nous limiter au calcul sur le tube de 30 mm, extension du maillage présenté sur la figure 6.12f.

7.3 Algorithme de couplage

Toujours dans le chapitre précédent, nous avons constaté la faible influence des déplacements de la structure sur le résultat fluide, même lorsque ceux-ci sont importants. L'algorithme le plus simple Dirichlet→Neumann est donc un bon choix pour réaliser ce calcul couplé.

La figure 7.1 schématise un tel algorithme couplant un fluide stationnaire à la structure. L'algorithme se retrouve indirectement piloté par la structure : c'est elle qui choisit les instants de couplage, et donne la main au code fluide pour démarrer son calcul stationnaire.

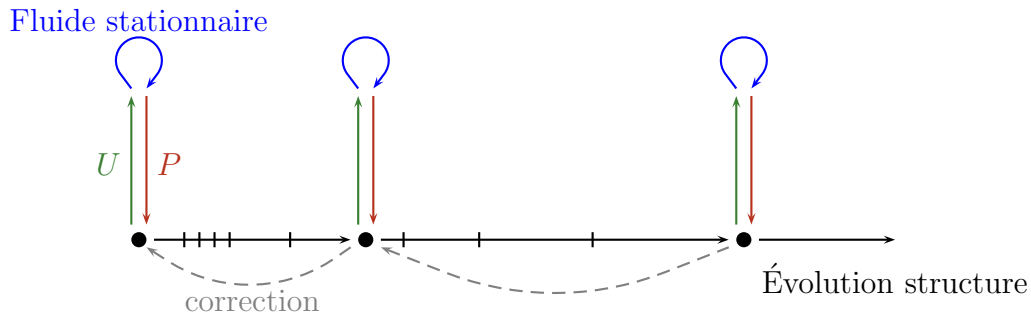


FIG. 7.1 – Algorithme choisi pour un couplage entre une structure quasi-statique et un fluide stationnaire. U désigne le transfert du champ de déplacement et P celui du champ de pression. Dans ce schéma, le couplage entre fluide et structure n'est pas nécessairement réalisé à chaque incrément de la structure.

Le schéma présente une situation où la structure ne déclenche pas le couplage automatiquement à chacun de ses pas de temps. Bien qu'en pratique, nous l'ayons réalisé à chaque pas de temps. Quelques essais nous ont en effet montré que le temps de calcul d'un état stationnaire du fluide est négligeable devant le calcul d'un pas de temps du solide. Cela va aussi en faveur de la version séquentielle de l'algorithme Dirichlet→Neumann, plutôt qu'une version parallèle. Si les temps de calculs avaient été du même ordre de grandeur, il aurait été préférable de minimiser les échanges, en limitant la fréquence ou en basant sur un senseur surveillant les variations des quantités échangées (ici le déplacement de l'interface).

Notons donc une fois de plus que cet algorithme a été adapté tout particulièrement à ce calcul, c'est-à-dire à ce modèle particulier du solide (non-linéaire et maillé finement, donc cher à calculer), du fluide (stationnaire et maillé assez grossièrement, donc rapide à calculer). Dans toute autre situation, un algorithme différent aurait pu se montrer intéressant.

7.4 Mise en place du calcul

Partant des mises en données découplées que nous avons sous la main, la mise en place du calcul couplé nécessite un certain nombre d'interventions.

Tout d'abord, l'algorithme de couplage est à intégrer dans les différents codes. Du côté du code fluide, cela se fait de manière directe : en début de calcul nous ajoutons une réception bloquante des données du solide ; celle-ci est suivie du calcul d'un état stationnaire autour de cette configuration pariétale. Une fois cet état déterminé, le champ de pression est renvoyé à la structure, et le code attend un nouveau champ de déplacement à l'interface (cf. Fig 7.2).

Du côté du solide, cela serait tout aussi simple dans les cas de calculs linéaires sans cor-

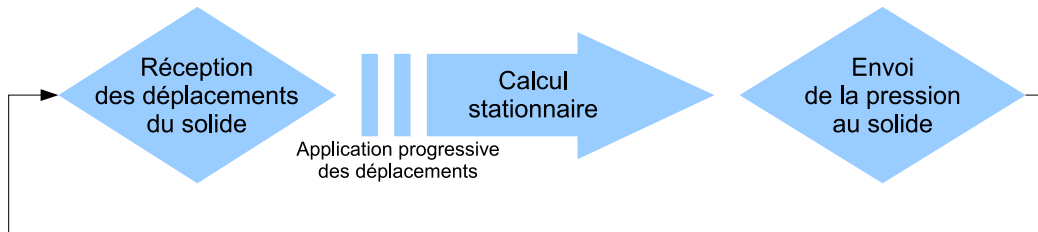


FIG. 7.2 – Implémentation de l'algorithme de couplage dans le code fluide

rections : en début d'incrément, faire une réception bloquante du champ de pression. Nous laissons ensuite le code réaliser normalement son incrément, la seule modification étant que lors de l'application des conditions aux bords couplées, la prédiction du champ de pression est réalisée. En fin d'incrément¹, il suffit d'envoyer le champ de déplacements de l'interface. En tout début de calcul, il est aussi nécessaire de réaliser un envoi du champ de déplacement, afin d'initialiser le calcul fluide.

Cependant, dans le cas de calculs non-linéaires, il nous faut prendre en compte d'éventuelles non-convergences de la résolution du problème solide, en plus de celles du processus couplé. Il n'est pas possible de conserver un envoi en début d'incrément, ceux-ci pouvant correspondre à des étapes de correction. Nous allons donc déplacer les envois en fin d'incrément, juste après sa validation.

L'algorithme ainsi retenu pour le code solide est le suivant. En tout début de calcul, il est toujours nécessaire de réaliser un envoi du champ de déplacement, afin d'initialiser le calcul fluide². En début d'incrément, aucune action particulière n'est nécessaire. Le code réalise ensuite son incrément, et les prédictions des champs de pression. C'est en fin d'incrément que les choses se compliquent : si le calcul mécanique n'a pas convergé, aucune action couplée n'est faite : nous laissons le code gérer la non-linéarité. S'il a convergé, le déplacement de l'interface est transmis au fluide, et le code se met en attente du champ de pression. Une fois ce champ reçu, les différents critères de convergence du calcul couplé sont vérifiés. Dans le cas présent, nous vérifions simplement que la prédiction réalisée sur le champ de pression est correcte. Si les critères sont vérifiés, l'incrément est validé. Dans le cas contraire, l'incrément est recalculé en modifiant le prédicteur.

Comme nous venons de le constater, la politique de couplage devient rapidement sujette à de nombreux petits détails. Nous avons initialement essayé d'écrire un unique *composant de problème couplé* qui puisse servir à implanter n'importe quel algorithme de couplage, le passage d'un algorithme à un autre se faisant simplement par la mise en données du fichier `.inp`. Cela s'est révélé une très mauvaise idée. Au bout de trois cas d'utilisation (les cas de couplage structure-structure, de couplage élastique puis élasto-visco-plastique sur le tube), l'objet en question est devenu ingérable.

Nous avons donc décidé que la mise en données de l'algorithme de couplage se ferait au travers d'un *plugin*, c'est-à-dire un fichier C++. Le *composant de problème couplé* développé précédemment est ainsi devenu une classe de base dont sont dérivées les implantations des

¹ ou toutes les n fins d'incrément si on désire une fréquence plus faible de couplage

² En pratique, les conditions initiales étant choisies à l'équilibre, l'état initial du fluide est connu : fluide au repos, pression de 30 bars, température de 50 K. On peut donc éventuellement se passer de cet échange initial, en spécifiant ce champ initial.

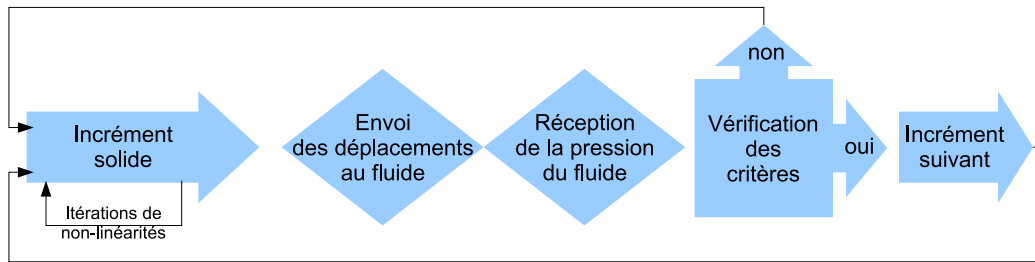


FIG. 7.3 – Implémentation de l’algorithme de couplage dans le code solide

algorithmes réellement utilisés. La classe de base fournit des routines de couplage de haut de niveau (par exemple le déclenchement de l’envoi ou de la réception des données, la vérification des critères de convergence), afin de rendre le plus simple possible ses dérivations. Cette solution s’est avérée au final beaucoup plus lisible, facile à gérer et surtout extensible aux cas du couplage thermique et du couplage à trois codes qui ont suivi. Nous renvoyons à l’annexe C.2.1, pour une discussion plus approfondie sur ces différentes questions d’implantation.

La figure 7.4 présente un exemple de mise en données de l’algorithme de couplage, au travers d’un fichier C++. Nous n’intervenons qu’à deux endroits : en début et fin d’incrément. En début d’incrément, nous passons la main aux critères de convergence, pour qu’ils puissent au besoin s’initialiser (ligne 14-15). En fin d’incrément, si l’incrément de la partie mécanique n’a pas convergé (lignes 20-22), nous quittons immédiatement la fonction, puisqu’aucune action couplée n’est envisagée. Les lignes 24-26 permettent un couplage suivant une certaine fréquence (soit à certains incrément, à certains instants ou toutes les quelques secondes : nous avons ici accès à toute la panoplie de description des fréquences dans ZéBuLoN). Si ces deux conditions sont satisfaites, nous transférons le champ de pression, recevons le champ de déplacement et enfin vérifions les critères de convergence. Ces derniers peuvent modifier la variable `ir` afin de déclarer la non-convergence de l’algorithme couplé. Les mesures à prendre dans cette dernière situation sont précisées ailleurs, et ne sont pas spécifiques au problème couplé. Elles dépendent par contre du type de *résultat d’intégration* émis. Nous reviendrons sur ce point dans la section 8.4.

```
1 #include <Coupled_problem_component.h>
2
3 class CPC_MECHANICAL_NOZZLE: public COUPLED_PROBLEM_COMPONENT
4 {
5 public:
6     virtual void start_increment();
7     virtual void end_increment(INTEGRATION_RESULT*& ir);
8 };
9
10 void CPC_MECHANICAL_NOZZLE::start_increment()
11 {
12     if (is_very_first_increment) do_transfers(Time_ini);
13
14     for(int i=0;i<convergence_criteria.size();i++)
15         convergence_criteria[i]->start_increment();
16 }
17
18 void CPC_MECHANICAL_NOZZLE::end_increment(INTEGRATION_RESULT*& ir)
19 {
20     if (ir) {
21         if (! ir->areYouA("IR_OK_BUT_DO_IT_AGAIN") ) return;
22     }
23
24     if (coupling_frequency) {
25         if (! coupling_frequency->ok()) return;
26     }
27
28     do_transfers(Time);
29     do_receives() ;
30     do_check_convergence(ir);
31 }
```

FIG. 7.4 – Exemple de mise en données de l'algorithme de couplage, pour l'application au calcul du tube. L'utilisation des routines de haut niveau permet d'ordonner simplement les différentes tâches à effectuer.

7.5 Résultats

L'algorithme de couplage simple présenté ci-dessus s'est montré particulièrement efficace sur l'application de la tuyère. Ceci s'explique principalement par le contraste très important entre les deux domaines. D'un côté, la structure reste relativement rigide, malgré les fortes températures. De l'autre côté, le fluide a une densité pratiquement négligeable en regard de la structure. Les échelles de temps des deux phénomènes sont aussi très différentes. Les modifications de la structure n'apparaissent qu'après une longue exposition aux forces de pression ; les fluctuations instantanées du fluide n'ont donc que peu d'effet sur elle. Enfin, le coût presque négligeable³ d'un calcul fluide nous permet une fréquence importante de couplage, sans pour autant pénaliser le coût total du calcul. Nous ne nous sommes donc pas privés de cette facilité.

Le calcul a été réalisé pour un tube de 30 mm de long, pour une durée de simulation de 600 s. Il est illustré sur la figure 7.5. Nous avons représenté d'une part l'intensité de la déformation plastique cumulée sur le solide ; d'autre part, nous avons représenté les lignes de courant ainsi que la pression calculée par le fluide.

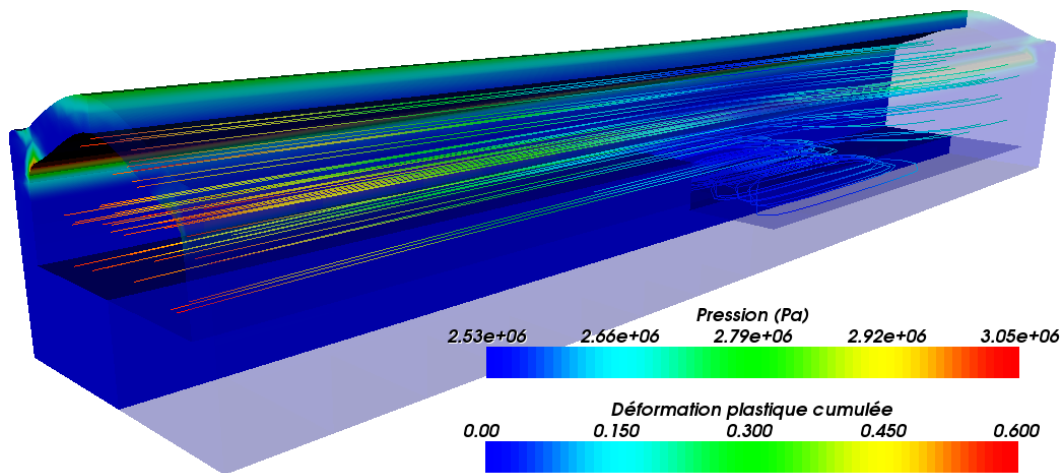


FIG. 7.5 – Résultat du calcul mécanique couplé

Les profils de pression observés se rapprochent de ceux des calculs découplés : la pression, imposée en entrée, chute progressivement jusqu'au niveau de l'insert, pour remonter jusqu'à sa pression de sortie. Les courbes 7.6, présentant le profil de pression le long du tube en début et fin de calcul montrent l'assez faible dépendance de ce champ vis à vis des déformations du tube.

Du côté solide, les effets déjà mentionnés lors des calculs découplés se retrouvent ici : les déplacements le long du milieu du tube sont toujours de l'ordre de l'épaisseur du tube. Par contre, nous constatons qu'ils ne sont pas uniformes le long du tube : ils sont plus marqués en entrée de tube et en sortie, où la pression est la plus importante. Au niveau de l'insert, là où la pression subit une chute d'intensité, les déplacements verticaux sont environ 20% inférieurs à ce qu'ils sont dans le premier tiers du tube (cf. Fig. 7.7). Ces variations montrent l'intérêt d'un calcul couplé au fluide. Les déformations plastiques cumulées se retrouvent concentrées dans les angles arrondis du tube ; elles y atteignent les 60%.

³ par rapport au coût d'un incrément solide

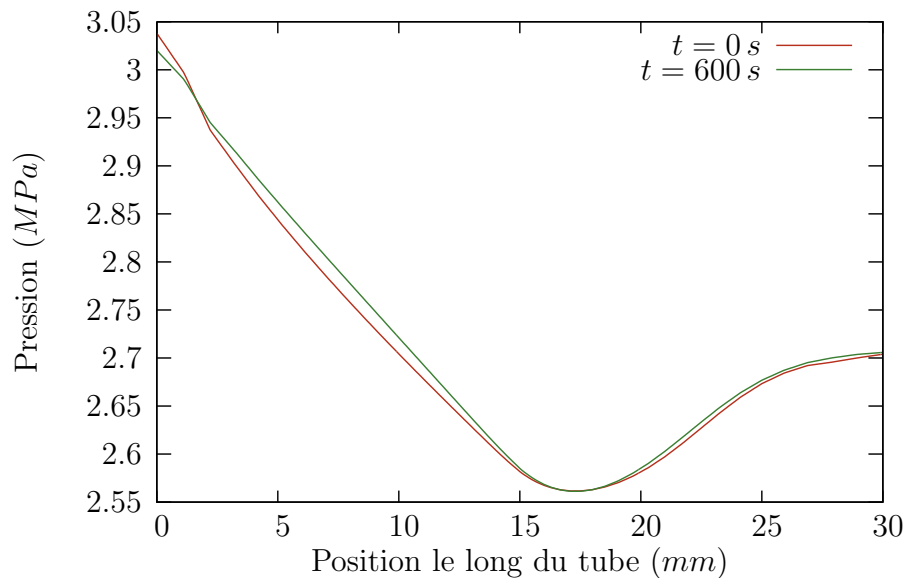
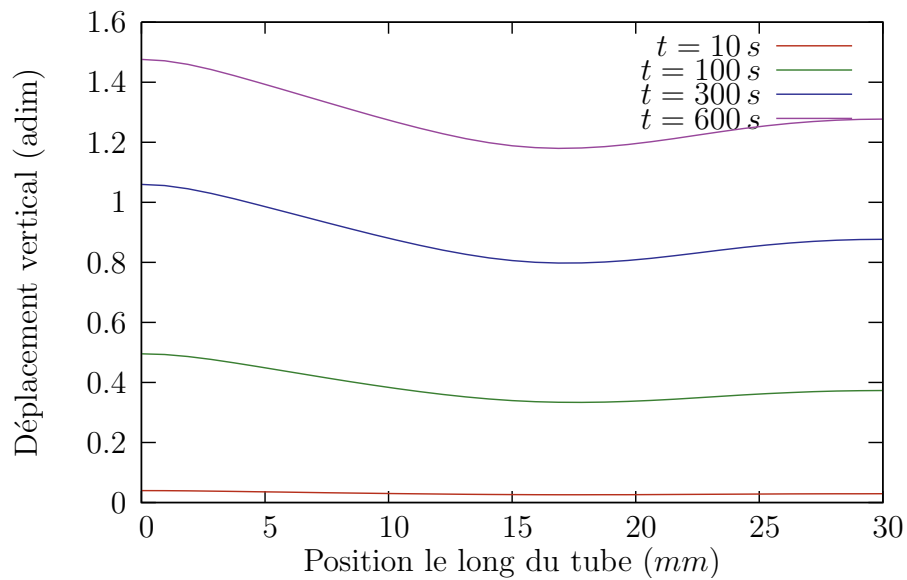
FIG. 7.6 – Pression le long de l'axe principal en début et fin de calcul (à $t = 0\text{ s}$ et $t = 600\text{ s}$)

FIG. 7.7 – Déplacement vertical le long du tube, pour différents instants

L'évolution des déformations au cours du temps (Fig. 7.8) met bien en évidence l'importance des déformations par fluage. Ce comportement étant relié à la pression subie, les déformations sont plus importantes en entrée et sortie de tube qu'en son milieu (Fig. 7.8). Les différences de déformations sont suffisamment notables pour justifier la nécessité d'appliquer un champ de pression réaliste au tube.

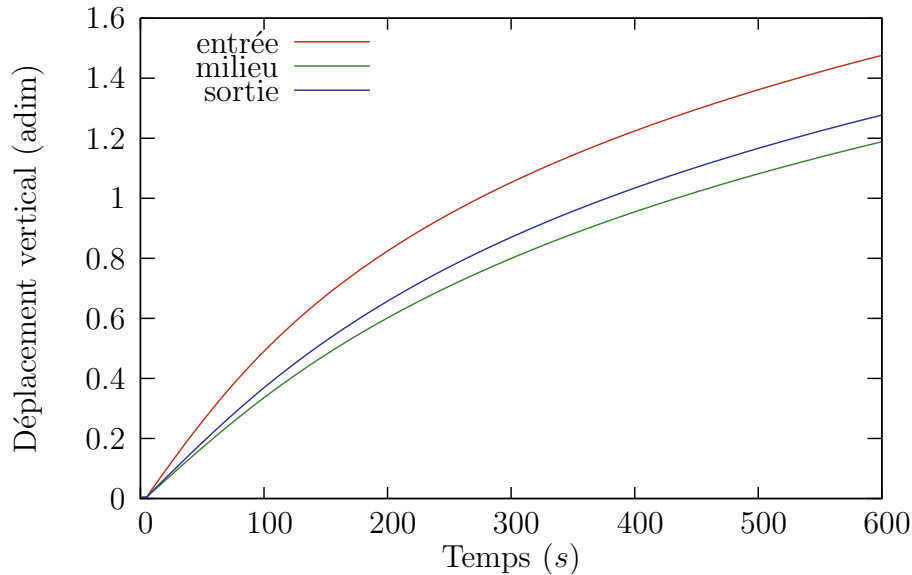


FIG. 7.8 – Évolution du déplacement vertical en trois points du tube (entrée, milieu et sortie)

Par contre, il semble a posteriori que le calcul complètement couplé ne soit pas indispensable. En effet, il suffirait d'appliquer le champ de pression variant en espace au tube pour l'ensemble du calcul pour obtenir un résultat tout à fait satisfaisant ; sa réactualisation au cours de calcul n'est pas forcément nécessaire. Un simple report manuel du champ de pression aurait pu suffire.

Néanmoins, ne perdons pas de vue que le cas qui nous intéresse concerne un problème couplant à la fois mécanique et thermique. Le couplage est assez faible du point de vue mécanique ; il n'en est pas de même du couplage aéro-thermique, ni du couplage thermo-mécanique. Le couplage aéro-mécanique se trouvera renforcé par ces couplages annexes. Il est donc important dès à présent de se soucier de solutions de couplage permettant un couplage automatique et fréquent de l'aéro-mécanique.

Chapitre 8

Calculs couplés thermiques

Ce chapitre décrit différents calculs couplant le solveur thermique de ZéBuLoN à MSD. Ces calculs vont permettre de mettre en place et de comparer différentes méthodes et algorithmes de couplage. La situation du tube de tuyère présente déjà de nombreuses difficultés, et nous espérons que ces difficultés sont suffisamment représentatives du calcul complet.

Cette situation a mis en défaut les méthodes que nous avons employées précédemment, le couplage thermique étant ici beaucoup plus fort que celui que nous avons constaté pour les calculs mécaniques. Pour aborder ces problèmes nous allons devoir utiliser de nombreuses stratégies de couplage.

8.1 Couplage par la méthode Dirichlet→Neumann

8.1.1 Première approche

Dans le cas thermique, la méthode de couplage Dirichlet→Neumann signifie simplement un couplage par échange de la température et du flux de chaleur à la paroi. C'est, comme dans le cas mécanique, la méthode la plus simple à mettre en place. Nous nous sommes donc intéressés à elle en premier.

La première série de test que nous présentons correspond à un couplage Dirichlet→Neumann, sans corrections. Son but est d'étudier l'influence de la fréquence de couplage sur le résultat. Théoriquement, les pas de temps de couplage doivent être de l'ordre du temps de diffusion de la chaleur dans le solide. Ce temps caractéristique est donné par

$$\tau_S = \frac{\rho_S C_S d_S^2}{k_S} \quad (8.1)$$

où ρ_S est la densité du matériau¹ ($8,47.10^3 \text{ kg.m}^{-3}$), C_S sa chaleur spécifique ($444 \text{ J.kg}^{-1}.K^{-1}$), k_S sa conductivité ($14,9 \text{ W.m}^{-1}.K^{-1}$) et d_S une longueur caractéristique. Un choix raisonnable de cette longueur est la taille de maille du maillage solide (6.10^{-5} m).

Le temps caractéristique ainsi calculé vaut :

$$\tau_S \simeq \frac{8470 \times 444 \times (6.10^{-5})^2}{14,9} \simeq 9.10^{-4} \text{ s} \quad (8.2)$$

¹cf. annexe B, valeurs prises à 20°C

soit environ une milliseconde.

Le calcul réalisé avec cette fréquence de couplage (Fig. 8.1 et 8.2) a été passé avec succès pour les 3,6 premières secondes du calcul (le calcul s'est arrêté pour un problème de fichier de taille trop importante). Cependant, son coût est rédhibitoire (environ une semaine de calcul a été nécessaire pour en arriver à ce point), et une telle fréquence n'est pas envisageable pour réaliser l'ensemble des 600 secondes de la simulation. C'est de toutes façons une fréquence très surévaluée dans les parties monotones du calcul (en particulier une fois les conditions aux bords stabilisées). Ce calcul nous servira néanmoins de référence dans la suite de ce chapitre.

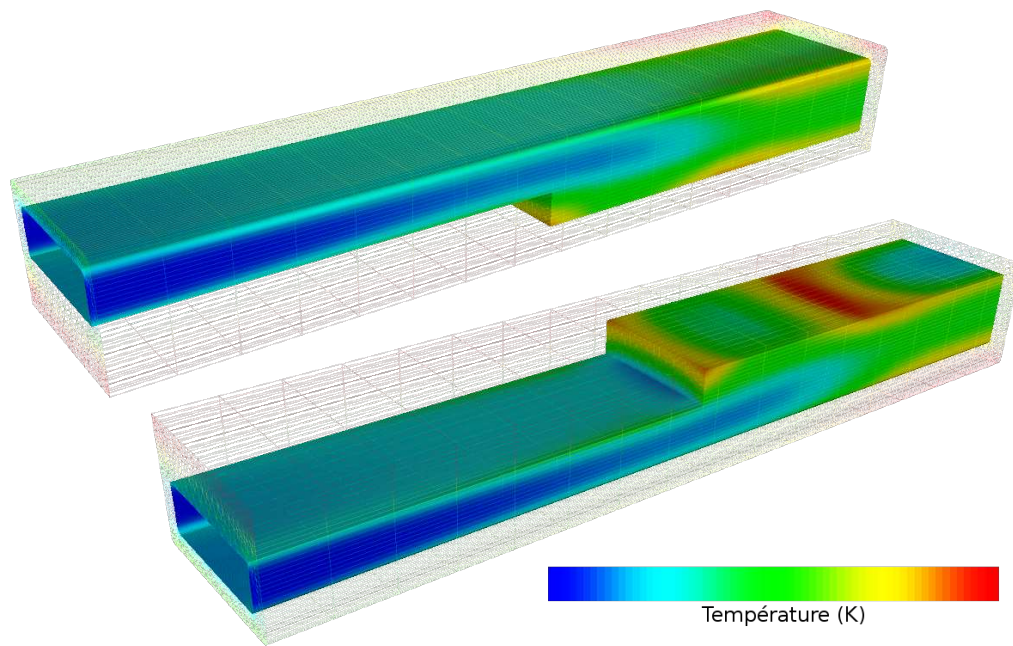


FIG. 8.1 – Solution de référence sur la surface couplée, pour $\Delta_t = 1/1000^e s$: champ de température à $t = 1s$.

Notons aussi la présence d'une instabilité dans ce calcul à $t = 0,35s$, à peine visible sur la courbe 8.2 au nœud 832², mais flagrante dans les coins du tube, à l'approche de la sortie (autour du nœud 920 par exemple). Nous n'en avons pour l'instant pas trouvé la cause. Nous avons retrouvé cette instabilité dans toutes les autres méthodes que nous employons, et au même instant ; l'instabilité n'est donc probablement pas due à un problème informatique. Il semblerait qu'elle soit plutôt d'origine physique ou géométrique, et donc à prendre en compte dans nos simulations. Elle reste toutefois d'une amplitude assez faible, et se dissipe rapidement.

D'autres instabilités isolées sont visibles au nœud 920, à $t = 0,83s$ et $t = 0,85s$. Elles sont suivies d'un comportement plus chahuté au-delà de la première seconde dans cette zone du tube. Le nœud 832 montre une dernière instabilité plus importante à $t = 3,425s$.

Afin d'appréhender ce que peut être une fréquence raisonnable dans cette situation précise, d'autres essais, à fréquences de couplage plus faibles ont été réalisés : $10/s$, $50/s$; les résultats sont présentés sur la figure 8.4.

²L'emplacement des nœuds est précisé sur la figure 8.3.

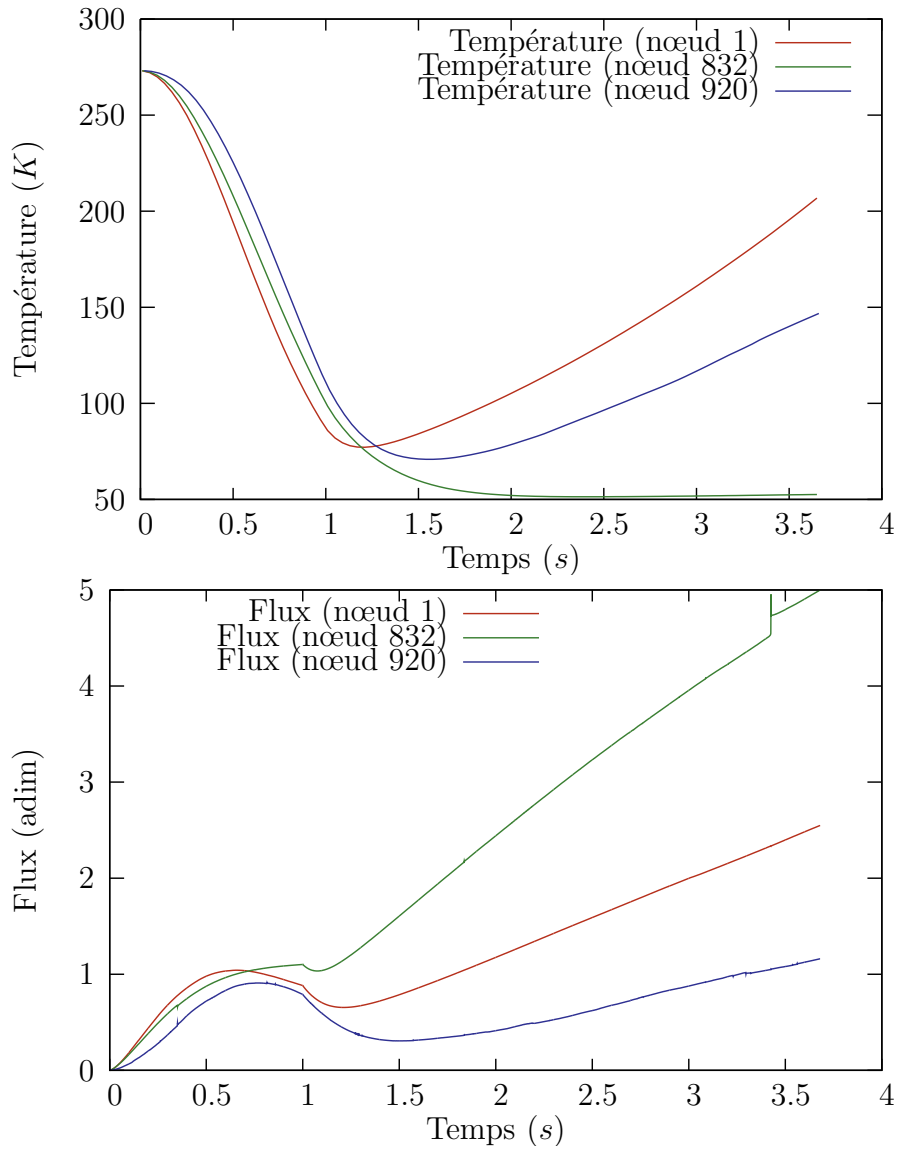


FIG. 8.2 – Solution de référence (pour $\Delta_t = 1/1000^e s$) : évolution de la température et du flux de chaleur en trois nœuds de la paroi couplée (voir la figure 8.3 pour la position de ces nœuds)

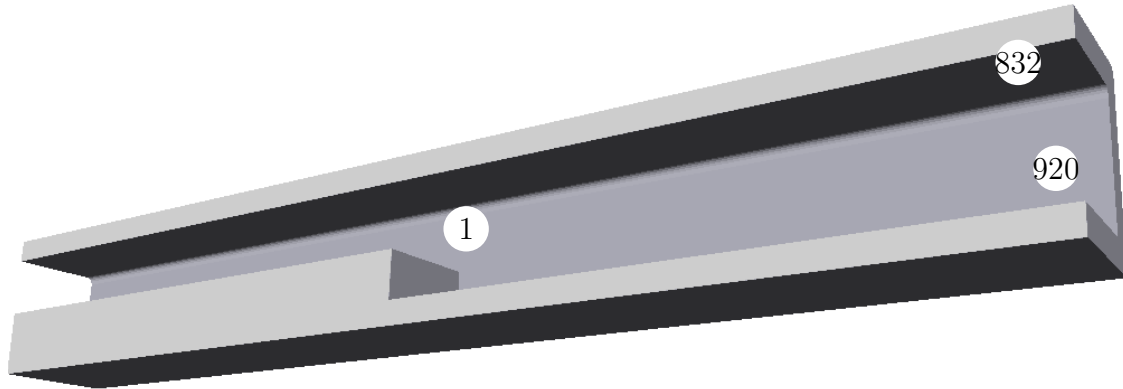


FIG. 8.3 – Emplacements des nœuds suivis dans les calculs de ce chapitre : le nœud 1 est situé à mi-longueur du tube sur la paroi latérale ; le nœud 832 est situé sur la face supérieure près de la sortie ; le nœud 920 est situé sur la face latérale près de la sortie.

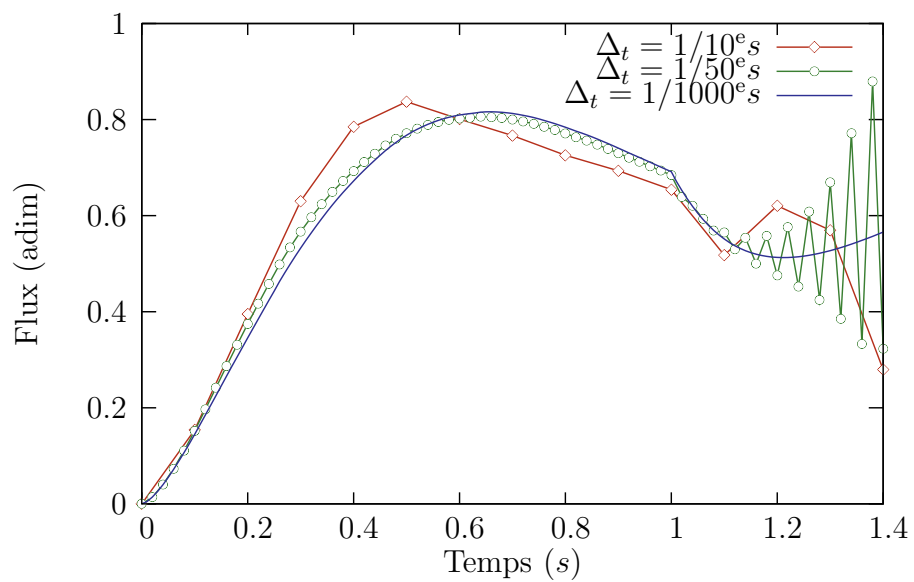


FIG. 8.4 – Flux reçus par la méthode Dirichlet→Neumann pour différents pas de temps de couplage ($\Delta_t = 0.1s$, $\Delta_t = 0.02s$ et $\Delta_t = 10^{-3}s$), au nœud n°1.

On y constate qu'une fréquence de 10 couplages par seconde est largement trop faible pour obtenir un résultat satisfaisant ; elle conduit à un résultat instable au-delà de la première seconde.

Une fréquence de 50/s donne par contre un résultat que l'on peut considérer acceptable, du moins durant la première seconde de calcul. Malheureusement, après cette première seconde (qui, rappelons-le, correspond au démarrage de la tuyère), le processus se déstabilise, et diverge assez rapidement.

D'autres calculs ont été réalisés à des fréquences intermédiaires : pour $\Delta_t = 1/100^e s$ le résultat est tout à fait comparable à la courbe de référence (Fig 8.5). La différence entre les 2 courbes est inférieure à 10^{-2} , ce qui est de l'ordre du pourcent, au vu des flux constatés.

Pour $\Delta_t = 1/200^e s$, le résultat est, à 10^{-3} près, confondu avec la courbe de référence, à l'exception de l'intervalle $[0,95; 1]$ pour lequel de petites oscillations d'amplitude 6.10^{-3} surgissent.

Ces deux derniers calculs ont été menés jusqu'à la 6^e seconde, la fréquence étant toujours trop importante pour espérer pousser le calcul jusqu'à la 600^e seconde.

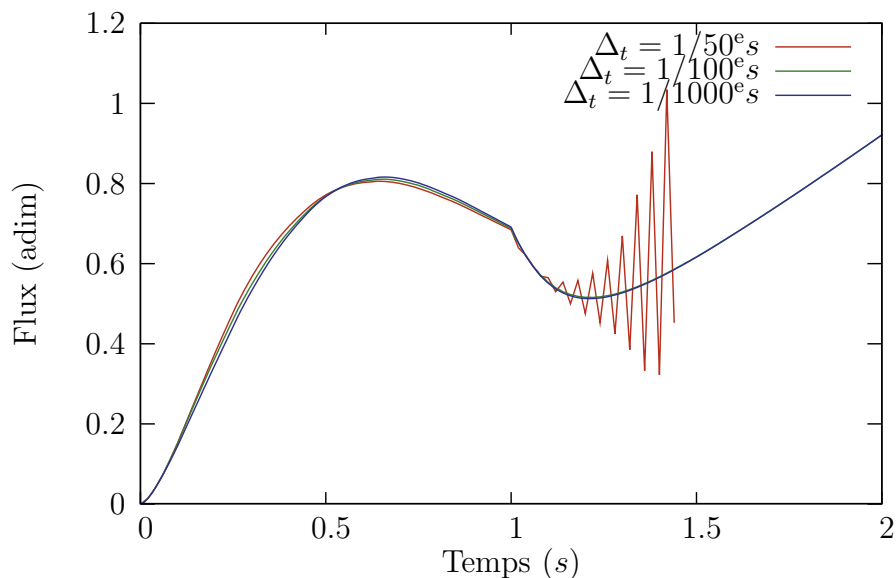


FIG. 8.5 – Flux reçus par la méthode Dirichlet→Neumann pour des fréquences de 50/s, 100/s et 1000/s au nœud n°1 de la paroi couplée (les courbes pour les fréquences de 100/s et 1000/s sont confondues au-delà de la première seconde).

8.1.2 Calcul de l'erreur

La méthode que l'on vient d'utiliser (Dirichlet→Neumann sans corrections) n'assure pas le principe d'action-réaction à l'interface : c'est une méthode de couplage faible. Plus précisément, nous nous reposons sur le fait que plus le pas de temps est fin, meilleure est la prédiction. Il est donc important de regarder a posteriori l'erreur commise entre la prédiction réalisée sur le flux de chaleur et la valeur effectivement reçue de l'autre code, afin de savoir dans quelle mesure le principe d'action-réaction est respecté. La différence entre flux reçu ϕ^r et flux prédit

ϕ^p est normalisée de la manière suivante :

$$\text{Erreur} = \left| \frac{\phi^r - \phi^p}{\phi^r} \right| \quad (8.3)$$

Cette formule nous sert d'indicateur d'erreur.

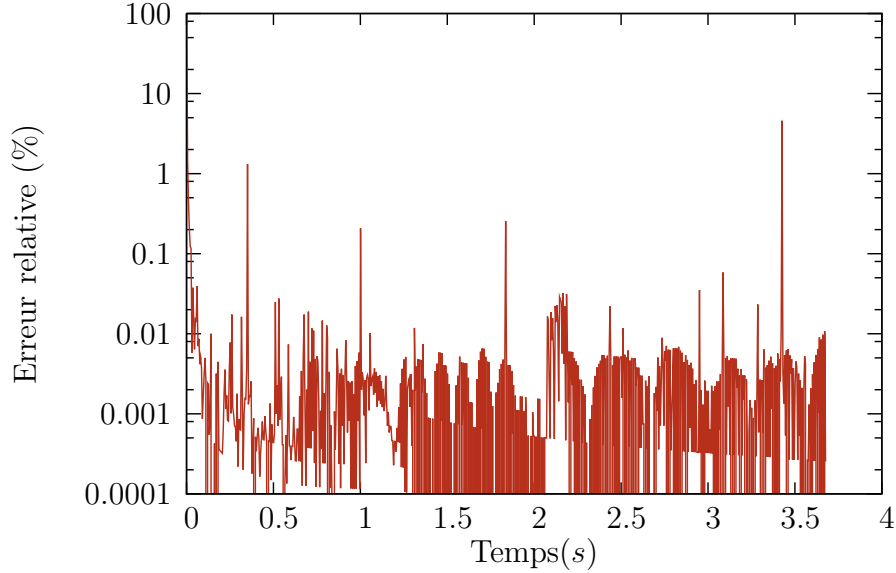


FIG. 8.6 – Comparaison des flux prédits et reçus dans la méthode Dirichlet→Neumann, pour une fréquence de 1000/s, au nœud 832. L'erreur relative est définie comme $\left| \frac{\phi^r - \phi^p}{\phi^r} \right|$.

La figure 8.6 compare le flux utilisé lors de la phase de prédiction avec celui reçu du code fluide en fin de pas de temps. On constate que l'écart est très faible, généralement en dessous de 0,01%. Les deux petites instabilités que l'on avait constatées sur la figure 8.2 ressortent bien à $t = 0,35s$ avec une erreur de l'ordre du pourcent, et à $t = 3,42s$ avec une erreur de l'ordre de 5%. Cet indicateur d'erreur ne nous donne par contre pas d'information supplémentaire quant à la source de l'erreur.

Deux autres pics d'erreur apparaissent sur cette courbe, à $t = 1s$ et à $t = 1,83s$. Le premier est tout à fait normal, puisqu'il correspond à l'instant de changement de conditions d'entrée dans le fluide. Le deuxième reste par contre inexplicable, comme les deux pics précédents. L'erreur reste tout de même inférieure au pourcent, et est donc tout à fait acceptable. Il reste enfin le problème du début du calcul, où, pendant les 20 premiers incréments, la différence entre flux prédits et reçus est du même ordre que ces flux.

Les deux courbes de la figure 8.7 fournissent deux mesures de l'erreur dans le cas d'une fréquence de couplage de 50/s. Ces erreurs sont présentées au nœud 1, qui est représentatif d'une bonne partie de la surface couplée.

Durant la première seconde, le calcul se comporte correctement, avec une erreur inférieure au pourcent. Notons toutefois à nouveau l'erreur relative importante pendant les huit premiers incréments, comme dans le cas d'une fréquence de 1000/s. Au vu de la faible amplitude des flux en ce début de calcul, cette erreur relative peut tout de même être considérée comme acceptable, influençant peu la suite du calcul. La comparaison entre les flux prédits et reçus montre bien la réaction en chaîne d'instabilités à partir de $t = 1s$.

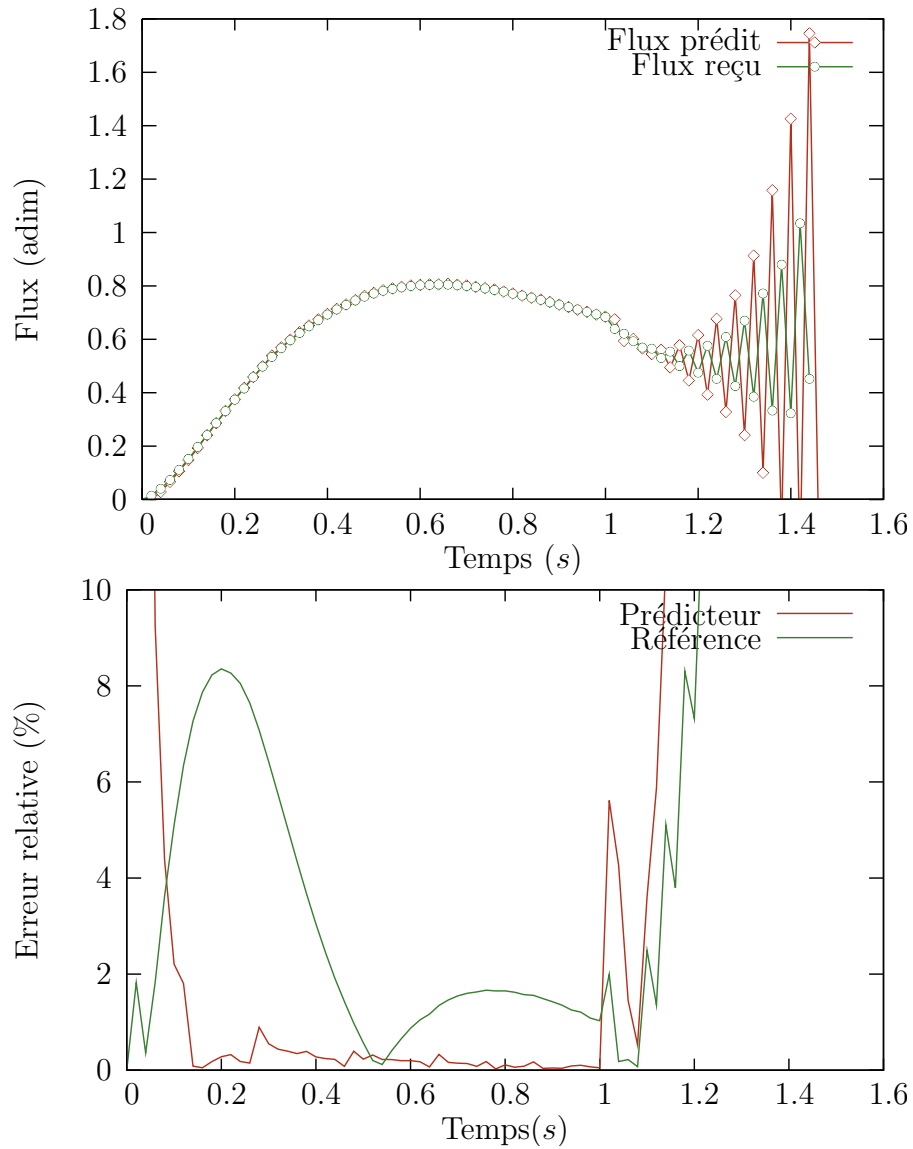


FIG. 8.7 – Comparaison des flux prédits et reçus dans la méthode Dirichlet→Neumann, pour une fréquence de 50 couplages par seconde, au nœud n°1 de la paroi couplée. Le premier graphique compare les flux prédits et reçus. Le deuxième graphique montre deux indicateurs d'erreur : l'erreur relative entre les deux flux prédits et reçus (1^e courbe, erreur définie par (8.3)) ainsi qu'entre le flux reçu et le calcul de référence (2^e courbe, erreur définie par (8.4)).

Ayant à disposition un calcul de référence, nous avons aussi trouvé intéressant de comparer les flux reçus pour une fréquence de 50/s à ceux reçus pour une fréquence de 1000/s. Cette erreur est elle aussi normalisée de la manière suivante :

$$\text{Erreur} = \left| \frac{\phi_{1000}^r - \phi_{50}^r}{\phi_{1000}^r} \right| \quad (8.4)$$

Cette première série de calculs montre un certain nombre de choses. Tout d'abord, en tout début de calcul, la méthode présente une erreur importante : cela vient du fait que les prédictions initiales (nulle dans cette situation) sont complètement erronées. Par contre la bonne nouvelle est la faculté qu'a le problème couplé de retrouver une solution correcte après quelques incréments. Modérons toutefois ces erreurs : elles ont lieu pour des flux qui sont finalement faibles par rapport à ce que subira la structure par la suite.

Deuxièmement, une fréquence de 50 couplages par seconde semble suffisante pendant la première seconde du calcul (qui, rappelons le, correspond au refroidissement de la tuyère). Les sauts constatés en certains nœuds à $t = 0,35s$ et $t = 0,85s$ sont rapidement compensés. Cette compensation est néanmoins spécifique à ce calcul et il serait très hasardeux de s'y fier. Au delà de la première seconde, cette fréquence se révèle par contre insuffisante.

Troisièmement, une erreur du prédicteur inférieure à 1% peut tout de même correspondre à une erreur par rapport à la courbe de référence de 8% (Fig 8.7). Cela nous donnera donc une fourchette raisonnable pour mettre en place nos critères de convergence (cf. section 8.3).

Enfin, la fréquence de 50 couplages par seconde est insuffisante pour bien appréhender le changement de régime entre la première seconde et la suite du calcul. Il semble qu'une fréquence d'au moins 100 couplages par seconde soit nécessaire. Mais raffiner constamment le pas de temps n'est pas forcément la méthode la mieux adaptée à notre situation.

À ce stade, plusieurs solutions s'offrent à nous :

- déterminer a priori une fréquence de couplage adaptée au calcul ;
- améliorer la prédiction faite pour le flux à l'interface ;
- mettre en place une phase de correction, assujettie ou non à une vérification de la qualité du prédicteur ;
- modifier le type de condition couplée, pour passer à un couplage Dirichlet→Fourier.

Nous ne nous étendrons pas plus sur la première solution : le choix de la bonne³ fréquence de couplage devient dans ce cas extrêmement difficile et nécessite de nombreuses tentatives. Quelques essais nous ont tout de même permis de poursuivre le calcul jusqu'à la sixième seconde, et conduits aux résultats présentés sur les figures 8.8 et 8.9. Cette méthode reste hasardeuse : il nous est arrivé de trouver des séquences de couplage semblant stabiliser le calcul, mais dont des raffinements redevenaient instables : dans ces situations, la compensation des erreurs constatée par exemple dans le cas d'une fréquence de 50/s n'ayant plus lieu. Cette solution étant difficilement transposable à d'autres situations, nous l'avons momentanément écartée. Elle reprendra par contre son intérêt dans le cadre d'un choix de pas de temps adaptatif (section 8.4).

Regardons maintenant de plus près les trois autres solutions envisagées.

³au sens de compromis entre stabilité et coût du calcul

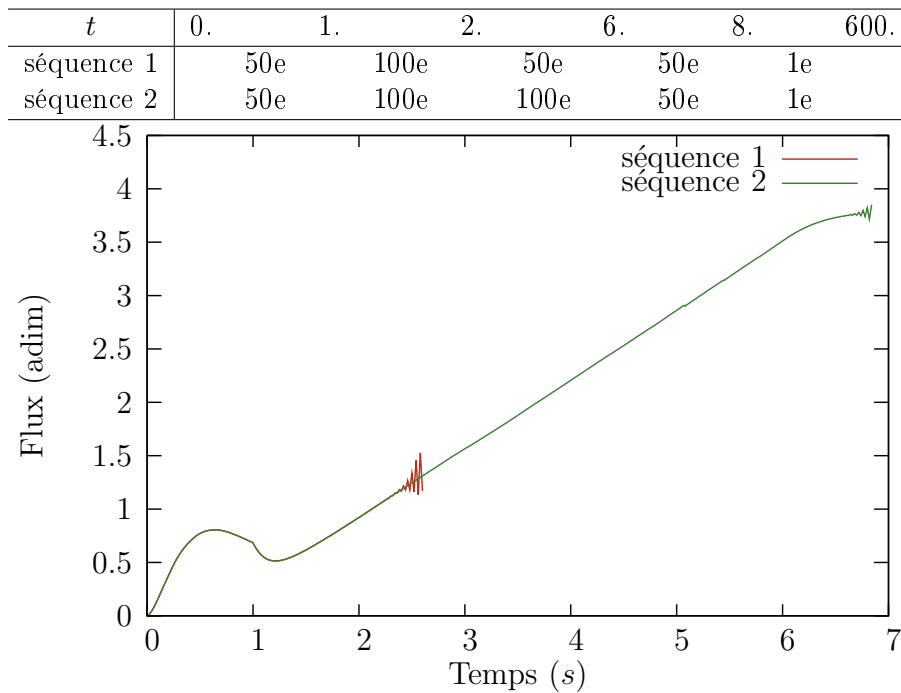


FIG. 8.8 – Ajustement de la séquence de couplage : flux de chaleur au nœud 1. De nombreuses tentatives permettent de trouver une fréquence de couplage adaptée à ce calcul. Le premier essai commence à osciller à partir de $t = 2s$, les oscillations s'amplifient ensuite jusqu'à divergence du calcul. Le deuxième essai commence à osciller visiblement à partir de $t = 6,5s$, mais la source des oscillations remonte certainement au changement de régime de la 6^e seconde.

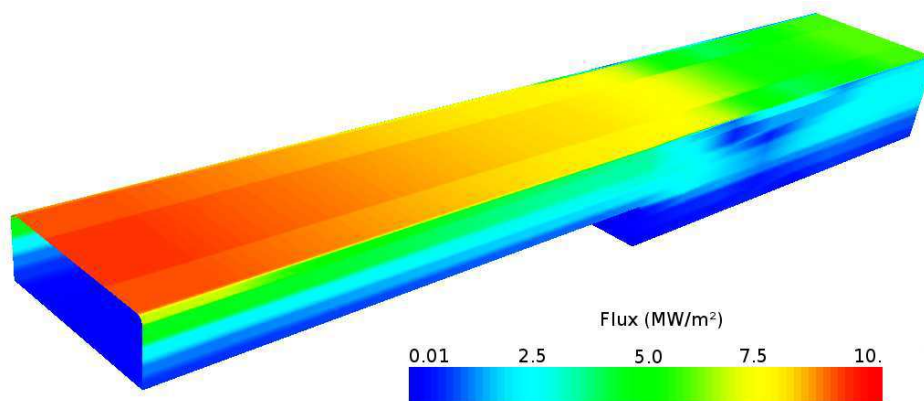


FIG. 8.9 – Flux de chaleur pariétal à $t = 6s$, obtenu pour la deuxième séquence de couplage décrite dans la figure 8.8.

8.2 Prédicteurs

Les calculs de la section précédente ont été réalisés en utilisant un prédicteur linéaire. Or, le terme d'erreur se fait important aux alentours de la première seconde, en une phase du calcul où le flux de chaleur sort d'une évolution quasiment linéaire, pour prendre une certaine courbure. Par ailleurs, l'allure de la courbe du flux de chaleur (Fig. 8.4) ne laisse presque aucun doute quant au gain d'une prédiction quadratique.

Modifier ce prédicteur est peu intrusif, et est donc relativement facile à mettre en place. Le flux utilisé à l'incrément de calcul $n + 1$ est calculé à partir des 3 champs précédents par interpolation de Lagrange (plutôt que via la formule 2.2 page 27, plus difficile à écrire pour des pas de temps non réguliers). Notons t_{n-2}, t_{n-1}, t_n les 3 instants où sont connus les champs précédents. Le champ extrapolé est alors donné par

$$\phi_{n+1} = L_{n-2}(t_{n+1}) \phi_{n-2} + L_{n-1}(t_{n+1}) \phi_{n-1} + L_n(t_{n+1}) \phi_n \quad (8.5)$$

où L_k désigne les polynômes de base de Lagrange. L_n par exemple est défini par

$$L_n(t) = \frac{t - t_{n-2}}{t_n - t_{n-2}} \cdot \frac{t - t_{n-1}}{t_n - t_{n-1}}$$

L_{n-1} et L_{n-2} sont définis de manière similaire.

Forts de ce nouveau prédicteur, reprenons un des calculs précédents. Nous repartons de celui à une fréquence de 100 couplages par seconde. Le flux de chaleur calculé au nœud 1, pour les trois prédicteurs — constant, linéaire et quadratique — à disposition est présenté sur la courbe 8.10.

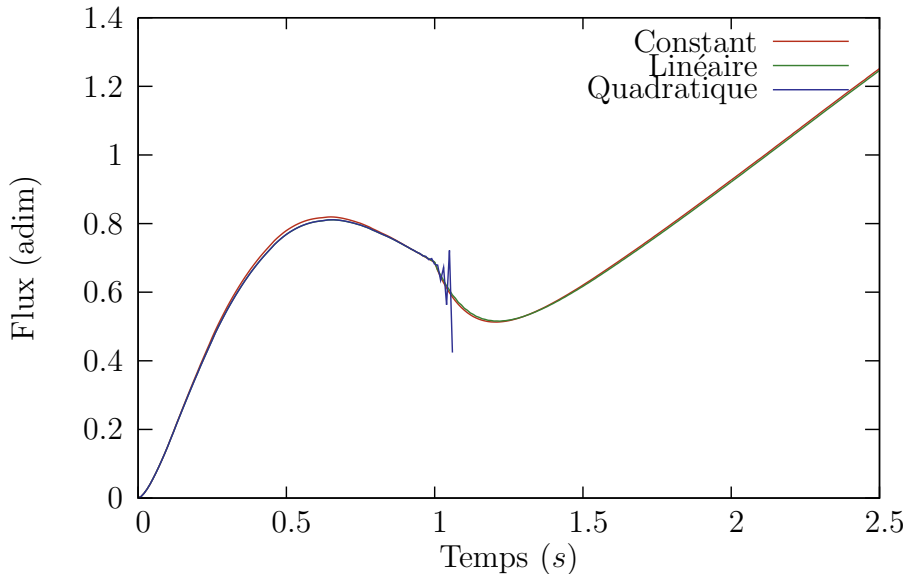


FIG. 8.10 – Comparaison des prédicteurs constant, linéaire et quadratique. Le résultat pour le prédicteur quadratique commence ses oscillations autour de la seconde 0,9 (tout juste visibles sur la courbe), oscillations qui gagnent en amplitude après le changement de régime de la fin de première seconde.

Contrairement à nos attentes, le prédicteur quadratique n'apporte pas d'amélioration significative. En fait, il se comporte de manière assez particulière : comparé au prédicteur linéaire, il n'aide ni à la précision, ni à la stabilité de l'algorithme. Par contre, lors d'apparitions des petites instabilités autour de la seconde 0,9, il a la particularité de faire diverger le calcul beaucoup plus rapidement qu'un prédicteur linéaire ou constant.

Le prédicteur constant fournit des résultats d'une stabilité comparable au prédicteur linéaire, à une exception près : au-delà de la sixième seconde, lorsque le calcul tend à se stabiliser, ce prédicteur se montre plus stable que le prédicteur linéaire. L'erreur entre flux prédit et flux reçu est par contre plus importante, celle-ci restant au-dessus des 10^{-2} , alors que l'erreur due aux deux autres prédicteurs (linéaire ou quadratique) oscille entre 10^{-3} et 10^{-2} .

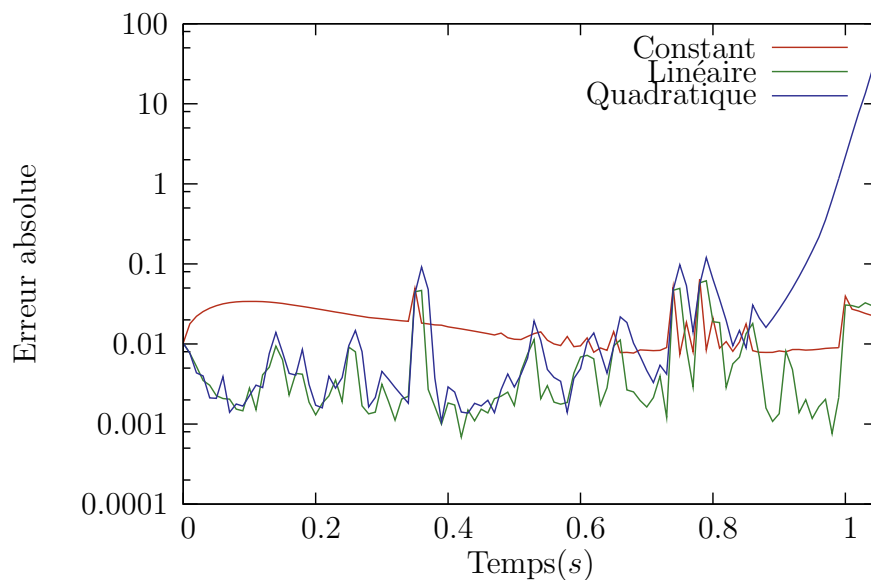


FIG. 8.11 – Comparaison des erreurs absolues pour les trois prédicteurs. L'erreur absolue est définie comme étant simplement $|\phi^r - \phi^p|$.

Ces différents prédicteurs seront retestés par la suite, en conjonction avec des critères de convergence et des conditions de couplage mixte. Les constatations seront similaires : le prédicteur quadratique n'apporte pas de gain significatif, et se montre plus sensible aux instabilités. En revanche, nous avons constaté que le prédicteur constant est le choix le plus "passe-partout" : il ne détériore que très légèrement le calcul dans ses premières phases, et est parfait en fin de calcul. Les deux prédicteurs d'ordre supérieur se montreront efficaces en début de calcul, mais malheureusement trop instables à partir de la sixième seconde.

8.3 Critères de convergence et phase de correction

La mise en place de différents prédicteurs n'a pas suffi pour diminuer l'erreur réalisée dans le calcul thermique couplé. Elle reste par moments supérieure à ce que l'on veut accepter, et conduit à des résultats instables. Il est donc nécessaire de mettre en place cette phase de correction permettant d'assurer le principe d'action-réaction, et donc d'améliorer la stabilité du calcul.

Nous mettons donc en place un système de vérification de l'équilibre à l'interface afin de rendre l'algorithme couplé fortement. Si cet équilibre n'est pas respecté, nous avons alors la possibilité de réitérer le pas de temps courant du calcul couplé, en corrigeant la prédiction précédente. Le calcul couplé se déroule ainsi de la manière suivante :

Algorithme 6 : Algorithme itératif de résolution du problème thermique

```

pour  $n = 0 \dots nb$  d'incrément faire
   $k = 0$ 
  répéter
     $k++$  Prédire le flux  $\phi_{n+1}^{(k)}$  à l'interface :
      si  $k = 1$ ,  $\phi_{n+1}^{(1)}$  est la prédiction d'ordre 2, donnée par (8.5)
      si  $k \neq 1$ , utiliser le flux reçu à l'itération précédente :  $\phi_{n+1}^{(k)} = \phi_{n+1}^{(k-1)}$ 
    Résoudre le problème thermique du solide
    Envoyer le champ pariétal de température  $T_{n+1}^k$  au fluide
    Résoudre le problème fluide
    Renvoyer le flux de chaleur pariétal  $\phi_{n+1}^k$ 
  jusqu'à  $\|\phi_{n+1}^k - \phi_{n+1}^{(k)}\| < \varepsilon$ 
fin

```

Dans cet algorithme, la notation entre parenthèses désigne les champs prédits, tels que $\phi_{n+1}^{(k)}$, et sans parenthèses les champs issus d'un calcul fluide, tels que ϕ_{n+1}^k .

Cet algorithme est en quelque sorte la traduction au cas du couplage de la méthode de Jacobi de recherche de point fixe. Notons tout de suite la possibilité d'en réaliser des variantes avec un coefficient de relaxation $\vartheta \in [0; 1]$, pour lesquelles on utilise

$$\phi_{n+1}^{(k)} = \vartheta \phi_{n+1}^{(k-1)} + (1 - \vartheta) \phi_{n+1}^k$$

Ce coefficient a vocation à stabiliser le processus itératif, comme très souvent dans ce genre de méthodes. Dans certains cas, l'ajout de cette relaxation ralentit la convergence de l'algorithme ; ce n'est donc pas une méthode qu'il faut adopter en toute situation.

Dans la situation de simulation du tube, nous avons choisi dans un premier temps une erreur de type norme infinie :

$$\|\phi_{n+1}^k - \phi_{n+1}^{(k)}\|_{\infty} = \max_{\text{nœuds}} |\phi_{n+1}^k - \phi_{n+1}^{(k)}| \quad (8.6)$$

Autrement dit, l'écart (nœud par nœud) entre le champ prédit et celui recalculé doit être inférieur à une certaine borne ε . Afin d'avoir une borne raisonnable, nous avons choisi $\varepsilon = 10^{-3}$, qui correspond à une erreur relative inférieure à 0,1%. Ce choix nous a paru plus adapté qu'un critère relatif local (erreur inférieure à un certain pourcentage en chaque nœud) : en effet, une partie importante du tube voit passer un flux relativement faible (c'est le cas par exemple du nez de marche, cf. Fig 8.9). Nous ne voulons pas pénaliser l'ensemble du couplage par les petites variations qui auront lieu sur ces parties.

Une norme en énergie, plus physique, sera par contre envisagée par la suite. Dans cette phase de mise au point, il est plus difficile d'en choisir un seuil raisonnable, et c'est pour cela qu'une norme infinie a été initialement retenue.

Lorsque les itérations sont réalisées jusqu'à satisfaction du critère décrit ci-dessus, on obtient les résultats de la figure 8.12. Ces résultats sont à mettre en regard de ceux sans correction (Fig. 8.4).

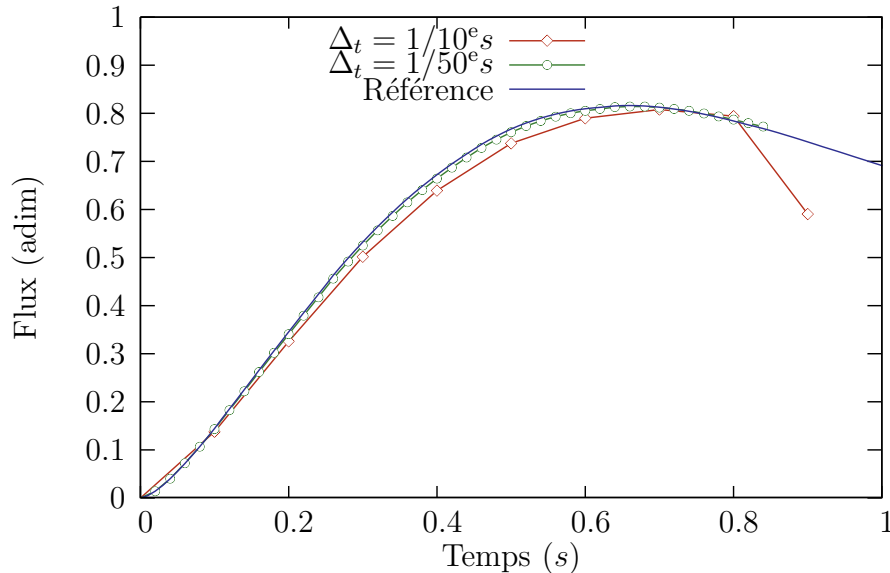


FIG. 8.12 – Résultats obtenus par la méthode Dirichlet→Neumann avec corrections, pour des périodes de couplage $\Delta_t = 1/10^e s$ et $\Delta_t = 1/50^e s$. Le résultat de référence *sans correction* est aussi représenté. Le calcul pour $\Delta_t = 1/10^e s$ s'est arrêté à $t = 0,9s$, le calcul devenant instable. Celui pour $\Delta_t = 1/50^e s$ s'est arrêté à $t = 0,84s$: les itérations de correction ne parviennent alors plus à faire décroître l'erreur.

À une fréquence de 10 couplages par seconde, le résultat est bien plus proche de la solution de référence. Le calcul ne parvient toutefois pas à atteindre la première seconde : il diverge à $t = 0,9s$. Ceci ne doit pas être interprété comme un désavantage par rapport à la méthode sans correction : bien que le calcul couplé y ait fourni une réponse jusqu'à $t = 1,2s$, nous avons bien vu que celle-ci était totalement erronée. L'ajout de cette phase de correction permet donc d'accroître la précision du résultat. Elle n'autorise pas pour autant des pas de temps trop importants.

Pendant la première partie du calcul, l'évolution de l'erreur au cours des itérations est décroissante de manière monotone (Fig. 8.13), diminuant environ de moitié à chaque itération.

Ces phases de correction ont toutefois un coût non négligeable : le calcul a nécessité environ une dizaine d'itérations par pas de temps. Plus précisément, ce calcul a nécessité au total 73 itérations pour atteindre la seconde 0,7. Son coût est donc comparable à celui du calcul non corrigé à une fréquence de 100 couplages par seconde, calcul qui fournissait de meilleurs résultats (tant en stabilité qu'en précision).

Aux autres fréquences testées (50, 100 et 200 couplages par seconde), les résultats fournissent des résultats semblables, quasiment confondus avec la courbe de référence. Les trois calculs ont néanmoins stoppé aux alentours de $t = 0,84s$. Le calcul arrive alors dans une phase où les résidus oscillent autour de $2 \cdot 10^{-3}$, sans arriver à descendre sous le seuil de 10^{-3} .

La méthode de correction complète que nous avons choisie y est à chaque fois mise en défaut, à cause de quelques points problématiques mis en évidence sur la figure 8.14. Quelques

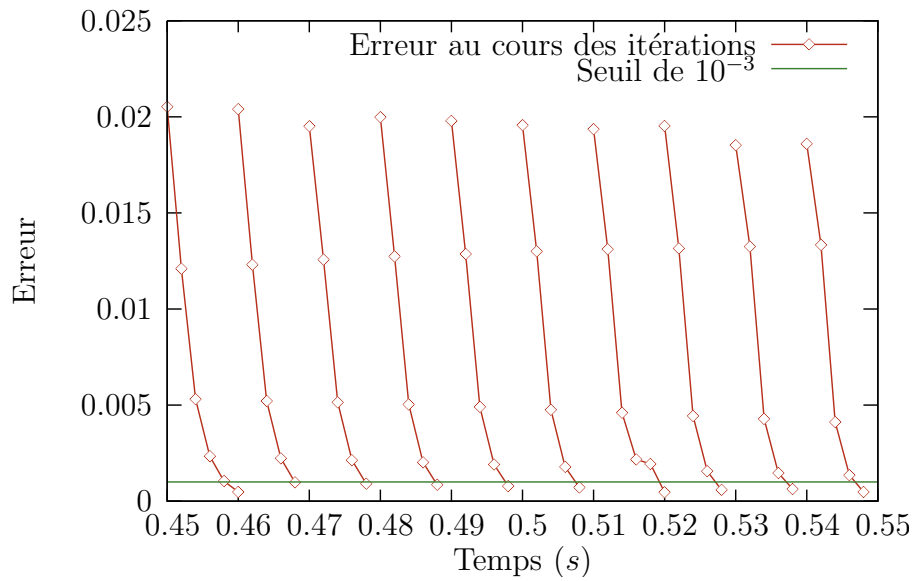


FIG. 8.13 – Évolution de l'erreur au cours des itérations, autour de $t = 0,5s$. Le calcul est réalisé à une fréquence de 100 couplages par seconde.

points situés sur le flanc du tube à proximité de la paroi froide produisent des erreurs absolues importantes. Le calcul fluide dans ces zones est relativement perturbé, par la présence de l'insert ; ce ne sont pourtant pas des zones critiques du calcul, et il est assez dommageable qu'elles empêchent le bon déroulement du calcul.

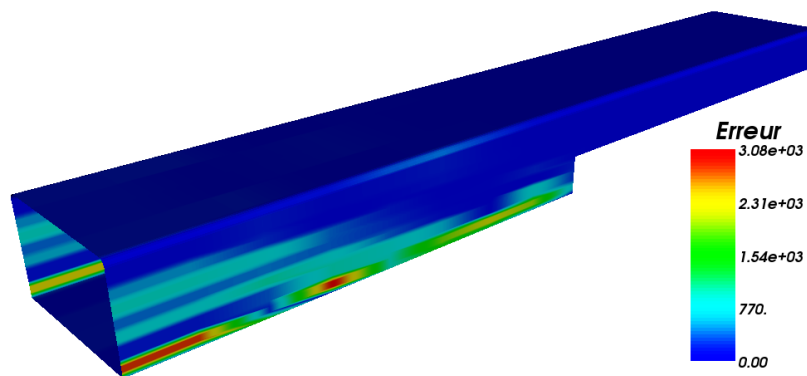


FIG. 8.14 – Localisation de l'erreur : seuls quelques nœuds situés sur le flanc du tube empêchent la convergence de l'algorithme, en conservant une erreur supérieure au seuil de 10^{-3} (figure obtenue pour une fréquence de 200 couplages par seconde, et représentant la différence entre flux prédit et flux reçu pour la 95^e correction, à $t = 0,835 s$).

Coût de la méthode

Si on oublie pour un instant cette 0,83^e seconde fatidique, nous pouvons évaluer le coût de ces quatre calculs (en nombre d'itérations de couplage⁴). Ce coût est présenté sur la figure 8.15. Il est à comparer avec celui des calculs sans corrections, qui est bien évidemment égal à la fréquence de couplage.

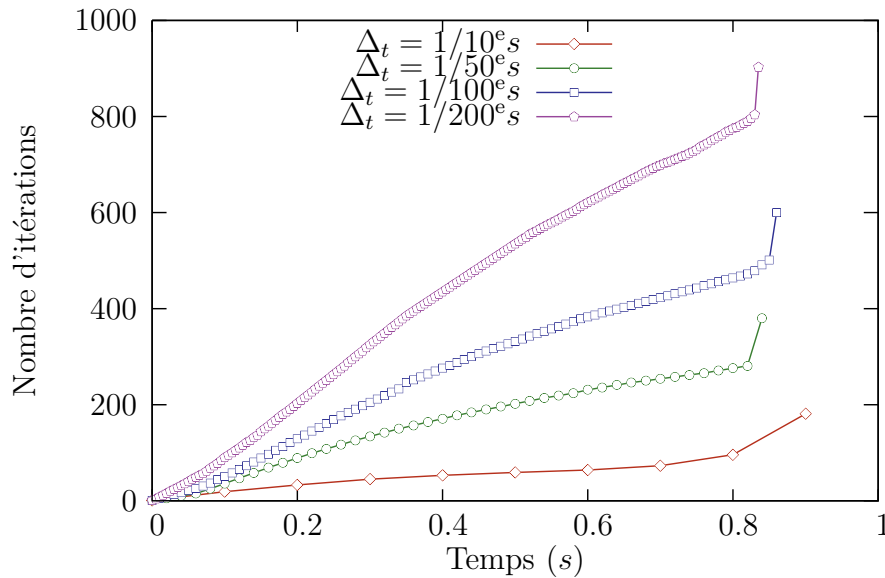


FIG. 8.15 – Nombre cumulé d'itérations, pour les méthodes Dirichlet→Neumann avec corrections.

Le calcul pour $1/50^e s$ a par exemple nécessité en moyenne 7 corrections par pas de temps, pour un total d'environ 276 itérations cumulées à $t = 0,8s$. Il a donc eu un coût comparable à un calcul sans correction à une fréquence de 350 couplages par seconde.

De même le calcul corrigé pour une fréquence de 200/s nécessite environ 5 corrections par seconde, pour un total de 699 itérations à $t = 0,7s$. Il est donc à comparer (Fig. 8.16) avec le calcul pour une fréquence de 1000/s que nous avons qualifié de calcul de référence. Les deux résultats sont quasiment identiques. Notons tout de même que durant les premières 0,02s du calcul, le calcul corrigé présente une allure plus raisonnable que celui sans corrections. Bien entendu aussi, le calcul corrigé assure une erreur entre flux prédit et flux reçu inférieure à 10^{-3} tout au long du calcul, alors que pour le calcul non corrigé, nous avons vu que l'erreur atteint par moments $20 \cdot 10^{-3}$. Le calcul avec correction a bien entendu le désavantage de s'arrêter à $t = 0,835s$, instant auquel notre méthode de point fixe ne parvient plus à réduire le terme d'erreur sous le seuil de 10^{-3} .

⁴ Cette méthode d'évaluation du coût des algorithmes est discutable : chaque incrément du code thermique solide a un coût constant ; par contre, le temps d'obtention d'un état stationnaire dans le code fluide est lié à l'amplitude de la perturbation qui lui est transmise. Ainsi, la première itération de couplage est toujours plus chère que ses corrections. En pratique, le nombre d'itérations de couplage reste tout de même un bon indicateur de l'efficacité de nos algorithmes.

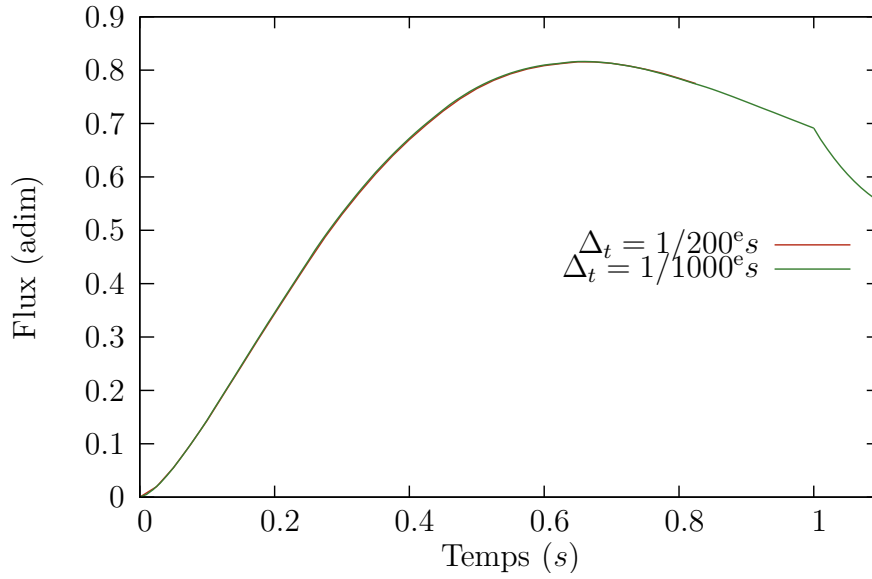


FIG. 8.16 – Comparaison des résultats obtenus pour des méthodes avec correction ($\Delta_t = 1/200^e s$, $\varepsilon_\infty = 10^{-3}$) ou sans correction ($\Delta_t = 1/1000^e s$). Avant l'arrêt du calcul corrigé à $t = 0,83s$, les deux calculs sont quasiment identiques.

Critère relatif local

Revenons justement sur ce problème survenant autour de $t = 0,83s$. Les difficultés sont liées à des températures aberrantes du côté fluide : quelques cellules estiment la température aux alentours de $170 K$, alors que les cellules avoisinantes sont plutôt de l'ordre de $100 K$. Les flux pariétaux déduits sont donc négatifs sur ces quelques cellules, et déstabilisent totalement le processus couplé.

Un critère de convergence local permet de contourner ce problème. Il définit l'erreur de la manière suivante :

$$\text{Erreur}_{\text{rel}} = \max_{\text{nœuds}} \left| \frac{\phi^k - \phi^{(k)}}{\phi^k} \right| \quad (8.7)$$

Autrement dit, nous normalisons la différence entre flux reçu et flux prédit, afin d'avoir une erreur plus significative. Quelques essais nous ont montré qu'une borne $\varepsilon_{\text{rel}} = 0,1$ permet d'obtenir des résultats satisfaisants. Ce critère s'est en pratique révélé moins restrictif que le critère absolu à 10^{-3} . Les problèmes rencontrés à certains instants sont considérés comme acceptables par ce critère, les oscillations de flux de $2 \cdot 10^{-3}$ étant localement négligeables par rapport aux flux en ces points.

Regardons les résultats obtenus avec ce nouveau critère de convergence, pour les deux fréquences de couplage $50/s$ et $100/s$. Les flux obtenus au nœud 1 sont représentés sur la figure 8.17.

Le calcul pour une fréquence de 100 couplages par seconde ne parvient pas à converger à la seconde 0,86, de la même manière que pour les calculs avec critère absolu. Par contre, le calcul à une fréquence de 50 couplages par seconde, bien qu'oscillant sensiblement dans la première partie du calcul, parvient à passer ce cap difficile. Alors qu'il semble stabilisé, le calcul recommence à avoir des difficultés à la seconde 1,28, l'empêchant lui aussi de converger. Si l'on resserre le critère de convergence à $\varepsilon_{\text{rel}} = 0,02$, les oscillations disparaissent ; par contre

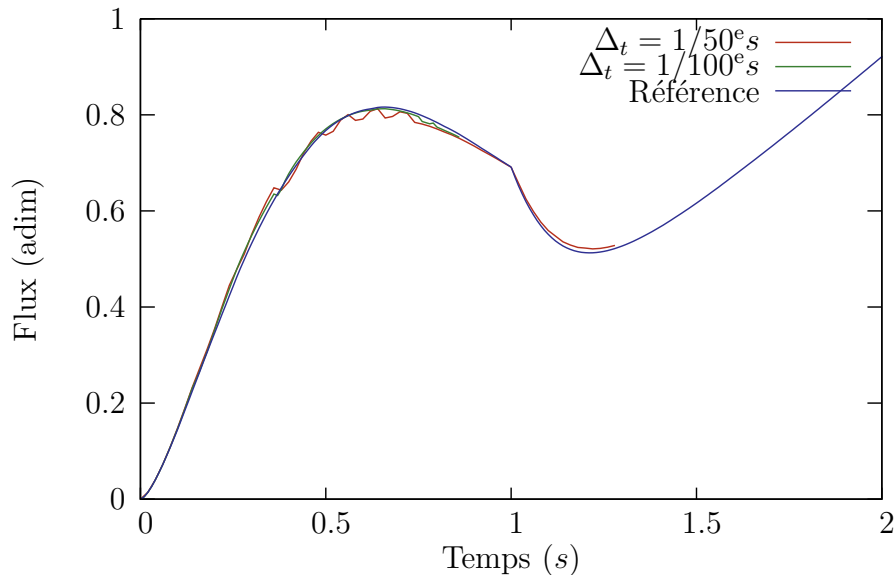


FIG. 8.17 – Résultats obtenus par la méthode corrigée, avec un critère de convergence relatif local ($\varepsilon_{\text{rel}} = 0,1$), pour différentes fréquences de couplage.

les instabilités autour de la seconde 0,86 réapparaissent, confirmant à nouveau les difficultés de ce passage.

8.4 Pas de temps adaptatif

Nos derniers calculs montrent deux phases : celles qui se passent bien (entre $t = 0,1$ et $t = 0,7$ par exemple), et celles qui se passent mal (mise en route du calcul, autour de $t = 0,8$ et démarrage de la condition de paroi chaude après $t = 1$).

Ces alternances de passages simples et complexes sont fréquemment rencontrées en mécanique du solide, lors du traitement de non-linéarités. Cela est généralement géré par un pas de temps adapté à la complexité du calcul en cours. La taille des pas de temps est alors déterminée en cours de calcul, en fonction de divers indicateurs des non-linéarités subies par la structure. Ces indicateurs peuvent par exemple suivre les variations d'une variable critique, ou plus globalement, suivre le nombre d'itérations nécessaires à la méthode itérative de résolution du problème non-linéaire.

Notre problème couplé présente un comportement tout à fait similaire à celui des problèmes de mécanique non-linéaires. Il semble donc naturel de lui appliquer cette même méthode de pas de temps adaptatif.

8.4.1 Description de la méthode

Nous avons retenu un pas de temps lié au nombre d'itérations nécessaires à l'obtention du point fixe. D'autres méthodes pourront être par la suite envisagées, méthodes par exemple liées à la variation des champs couplés.

Nous nous sommes inspirés de ce qui est actuellement utilisé dans ZéBuLoN : l'utilisateur spécifie un nombre N_{opt} d'itérations de couplage qu'il considère optimal, ainsi qu'une borne s

maximale d’agrandissement du pas de temps. Le code ajuste le pas de temps en tentant de rester autour de ce nombre optimal : si l’incrément précédent a nécessité plus que les N_{opt} itérations, le pas de temps est diminué ; s’il en a nécessité moins, le pas de temps est augmenté. Nous résumons cela par

$$\Delta_t = r \cdot \Delta_t^{\text{prev}}$$

Heuristiquement, il avait été déterminé que le coefficient r donné par

$$r = \begin{cases} 1 + \frac{s-1}{10} & \text{si } N_{\text{prev}} = N_{\text{opt}} \\ \min\left(\sqrt{\frac{N_{\text{opt}}}{N_{\text{prev}}}}, s\right) & \text{si } N_{\text{prev}} \neq N_{\text{opt}} \end{cases}$$

convient pour les calculs mécaniques [70]. Dans cette formule, N_{prev} correspond au nombre d’itérations effectivement nécessaires au pas de temps précédent. Nous reprenons cette formule au compte du couplage, formule dont la principale vertu est d’être supérieure à 1 quand $N_{\text{prev}} \leq N_{\text{opt}}$ et inférieure à 1 sinon. La racine carrée permet de modérer la croissance du pas de temps, par rapport à un simple $r = \frac{N_{\text{opt}}}{N_{\text{prev}}}$.

Le développement complet de la gestion de ces pas de temps adaptatifs a été complexe : il a fallu à la fois tenir compte de ce que ce pas de temps de couplage indique, ainsi que de ce qu’indiquent d’éventuels autres pas de temps adaptatifs basés sur les non-linéarités mécaniques ou thermiques.

Se pose d’ailleurs la question de savoir s’il est préférable de dissocier complètement le pas de temps de couplage du pas de temps physique, ou s’il faut les combiner en un unique. La première solution est la plus élégante. En effet, la fréquence de couplage est dépendante de la variation des quantités couplées : pression, déplacements, flux ou température de la paroi. Le pas de temps de la physique est lui lié aux variables mécaniques ou thermiques : par exemple aux déformations plastiques cumulées, ou encore à la dépendance de la conductivité à la température. Il est donc courant que de fortes non-linéarités soient subies par la structure sans pour autant que les quantités pariétales varient de manière importante. L’échange de ces quantités pariétales n’apporte donc rien d’un point de vue couplage.

La deuxième solution, consistant en pratique à retenir le plus petit des pas de temps proposés par les divers pas de temps automatiques, est donc plus onéreuse. Elle a néanmoins l’avantage d’être conservatrice, au sens où elle forcera toujours une fréquence assez fine pour satisfaire à la fois les contraintes de chacune des physiques et les contraintes couplées.

Il n’est donc pas évident de déterminer a priori la meilleure solution : c’est au final la manière dont l’algorithme est implanté qui déterminera quel choix est le mieux adapté.

Dans l’application que nous envisageons, nous avons choisi la deuxième solution, principalement pour sa plus grande facilité de mise en œuvre, ainsi que son intégration dans l’algorithme de couplage. Les calculs thermiques que nous avons jusque-là testés ne subissent de toute manière que de faibles non-linéarités. Le coût d’une itération du calcul thermique étant relativement faible en comparaison du calcul fluide, augmenter la fréquence des calculs thermiques n’a que peu d’incidence sur la performance globale.

Le premier choix sera par contre plus intéressant dans l’approche totalement asynchrone que nous décrivons dans la section 5.2. Pour les calculs mécaniques, beaucoup plus coûteux et pour lesquels les pas de temps physiques sont par moments beaucoup plus fins qu’en thermique, cette première méthode aura aussi son intérêt.

8.4.2 Résultats obtenus

Forts de cette méthode automatique de choix du pas de temps de couplage, nous reprenons les calculs précédents. Les réserves que nous avons exprimées concernant d'éventuels interférences avec un pas de temps de la thermique ne s'appliquent pas à notre problème de calcul d'un tube de tuyère. Les non-linéarités sont relativement faibles dans le problème thermique : elles ne sont dues qu'à la dépendance des coefficients de conductivité par rapport à la température. En d'autres termes, le pas de temps de couplage sera toujours plus petit que le pas de temps que requerrait la thermique.

Nous reprenons donc la situation précédente :

- un critère absolu de convergence (de 10^{-3}) ;
- un prédicteur linéaire ;
- un pas de temps initial de couplage de $1/10^6$ s ;

et effectuons deux tests : le premier pour une valeur de N_{opt} à 5, le deuxième pour une valeur de N_{opt} à 10.

Ces deux calculs fournissent des résultats analogues aux méthodes à pas de temps fixes, comme on peut le constater sur les courbes de la figure 8.18.

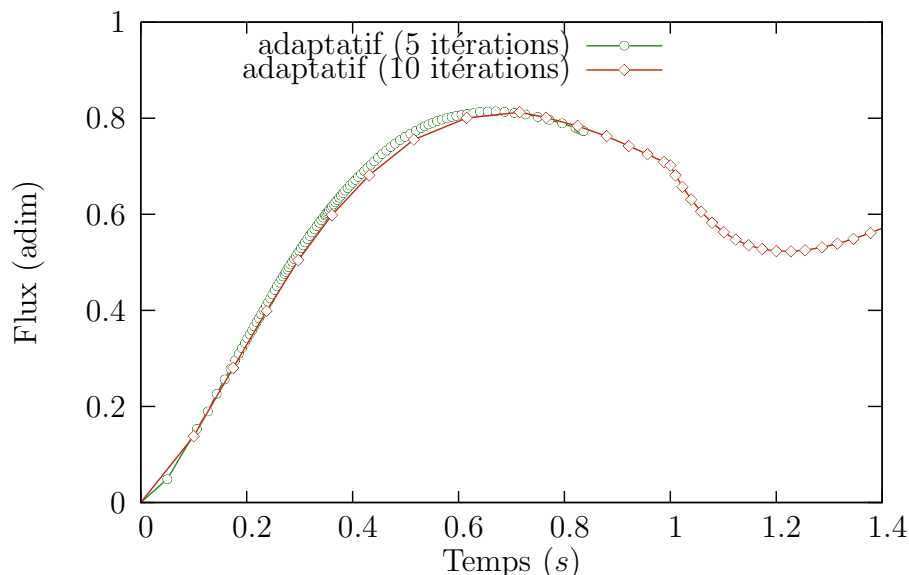


FIG. 8.18 – Résultats de la méthode du pas de temps adaptatif. La première courbe est obtenue pour un nombre optimal d'itérations $N_{\text{opt}} = 5$, la deuxième pour $N_{\text{opt}} = 10$.

Rappelons que le calcul à fréquence fixe ne parvient pas à trouver d'état convergé au-delà la 0,83^e seconde. Le calcul pour $N_{\text{opt}} = 5$ souffre du même problème. Par contre, celui pour $N_{\text{opt}} = 10$ passe par chance par-dessus la difficulté ; il s'arrêtera à la seconde 2,7, ne parvenant pas à trouver d'état convergé.

Ces courbes mettent aussi en évidence la différence de pas de temps choisis dans les deux situations : dans le cas $N_{\text{opt}} = 5$, ils diminuent très rapidement vers une taille de l'ordre de $5 \cdot 10^{-3}$ s. Par contre, pour $N_{\text{opt}} = 10$, le calcul se déroule beaucoup mieux, et les pas de temps arrivent à rester de l'ordre de 0,1 s.

Le bon comportement de la méthode pour $N_{\text{opt}} = 10$, ou plutôt le mauvais comportement de $N_{\text{opt}} = 5$, s'explique par le fait que le nombre d'itérations nécessaires pour atteindre un

état convergé est peu sensible à la taille des pas de temps, et est généralement compris entre 5 et 10 itérations.

Sans surprise, la différence entre les méthodes se trouve donc au niveau des coûts de chacun des calculs (Fig. 8.19). Si l'on compare le nombre d'itérations nécessaires pour atteindre un état convergé, le calcul pour $N_{\text{opt}} = 10$ ne nécessite qu'une centaine d'itérations pour atteindre la 0,8^e seconde. Par contre, les deux autres calculs (à fréquence fixe, et pour $N_{\text{opt}} = 5$) en nécessitent respectivement 450 et 580.

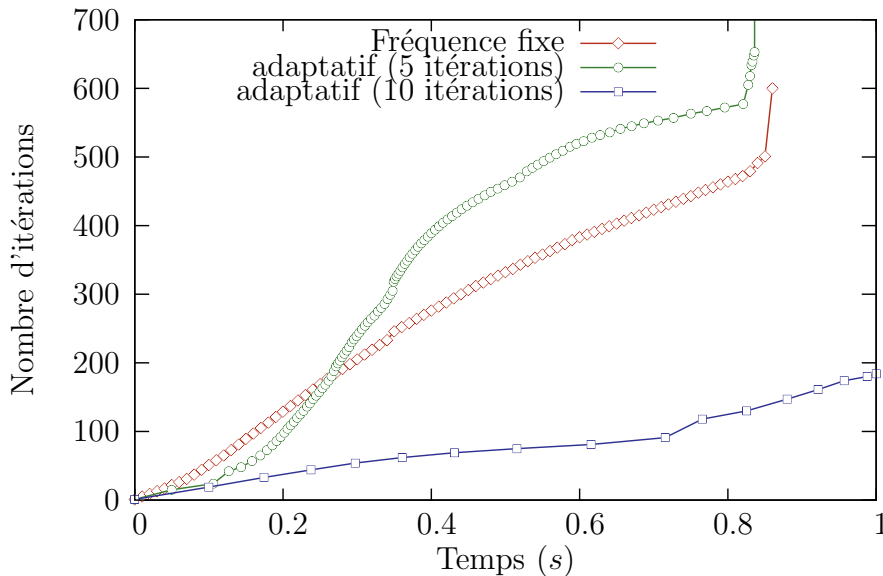


FIG. 8.19 – Comparaison des coûts des méthodes à pas de temps fixe et adaptatif

Le gain en temps de calcul est obtenu par l'augmentation de la taille des pas de temps. On constate en effet (Fig. 8.20) que le nombre d'itérations nécessaires quand $N_{\text{opt}} = 10$ n'est pas beaucoup plus important que dans l'autre cas. Il est aussi intéressant de constater sur cette même courbe le pic d'itérations aux alentours de la 0,8^e seconde : il indique que le calcul a bien rencontré la difficulté à cet instant, mais a réussi à faire converger l'algorithme couplé.

Pour conclure sur cette méthode de détermination automatique des pas de temps de couplage, il nous faut remarquer que la méthode peut se montrer très efficace si elle est bien paramétrée. En revanche, si nous n'autorisons pas un nombre d'itérations suffisant la méthode se révèle pénalisante.

Le critère que nous avons proposé, basé sur un nombre d'itérations est, en ce sens, global. Il englobe les différents aspects qui peuvent apparaître dans le problème couplé : instabilités de chacun des codes autour d'un point de couplage, instabilités du couplage lui-même. Des critères plus ciblés, sur les variations des champs couplés par exemple, se montreront peut-être plus faciles à gérer d'un point de vue physique.

Il subsiste des difficultés, en particulier à $t = 2,73$ s, qui nécessitent encore d'autres améliorations. Nous allons donc nous intéresser à une troisième stratégie, consistant à passer à des conditions de raccord mixte à l'interface couplée.

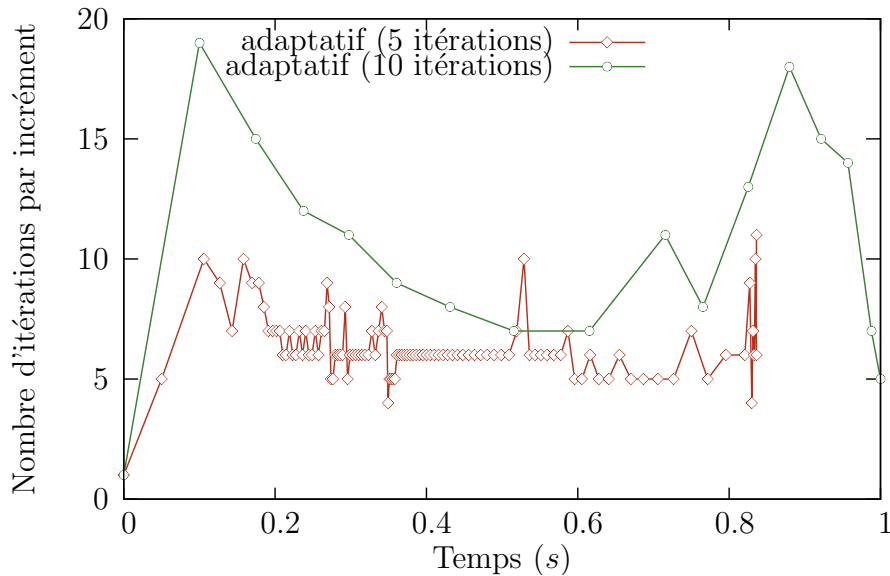


FIG. 8.20 – Nombre d’itérations par incrément pour chacun des deux jeux de paramètres de la méthode de pas de temps adaptatif

8.5 Couplage par la méthode Dirichlet→Fourier

Le raccord de domaines par des conditions mixtes, dites de Fourier, est devenu habituel dans les méthodes de décomposition de domaines. Il permet une vitesse de convergence accrue. Si nous poussons notre analogie entre ces méthodes de décomposition et les méthodes de couplage partitionnées, il est naturel d’espérer un gain à l’utilisation de ce type de conditions de raccord.

La condition de Fourier est une condition mixte : plutôt que d’imposer des degrés de liberté (condition de Dirichlet) ou d’imposer un flux (condition de Neumann), on impose une relation affine entre flux et température :

$$\phi + h(T - T_{\text{ref}}) = 0$$

Nous nous sommes limités à une température de référence constante égale à 50 K (donc mal adaptée à la première seconde de calcul), et à un coefficient d’échange déduit du flux de chaleur reçu (plutôt qu’un coefficient d’échange directement évalué par le fluide).

8.5.1 Loi de comportement flux–température

L’étude du comportement précis du calcul sur le tube nous apporte un argument supplémentaire en faveur de ce type de conditions de raccord. Regardons en effet les courbes de réponse flux–température au niveau de la paroi couplée (Fig. 8.21).

Pendant la première seconde, correspondant à la baisse de température du fluide, la relation flux température est difficilement identifiable. Par contre, au delà de cette première seconde, les courbes montrent que l’ensemble des points de la paroi couplée présentent un comportement quasiment affine entre le flux et la température. L’abscisse à l’origine de toutes ces courbes est très proche des 50 K correspondant à la température d’entrée du gaz, mais aussi à la

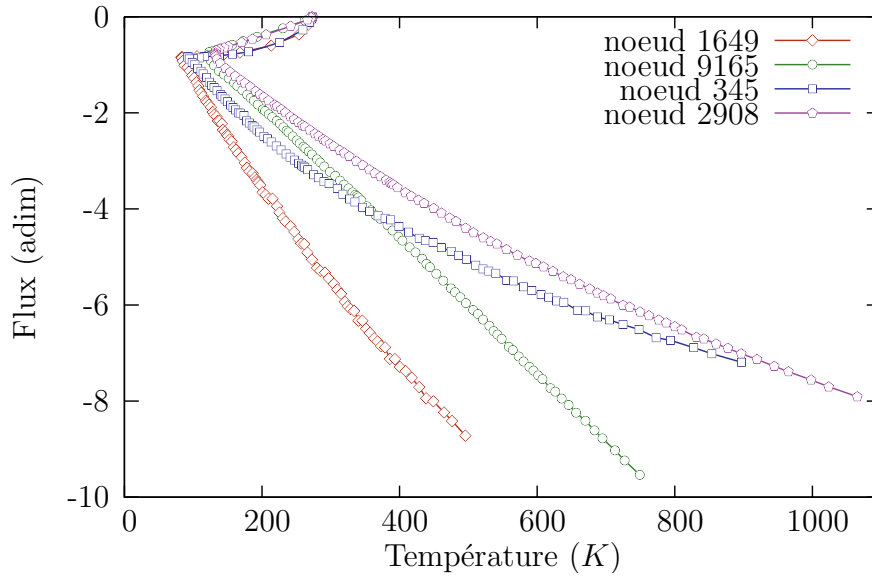


FIG. 8.21 – Relation flux–température en différents nœuds de la paroi couplée. Ces courbes sont obtenues à partir des résultats de la séquence 2 de la figure 8.8, pour t compris entre 0 s et 6 s.

température adiabatique de frottement. Autrement dit, en tout point de la paroi couplée nous avons bien une relation :

$$\phi = \alpha(T - T_{\text{ref}})$$

pour $T_{\text{ref}} = 50K$. Les pentes α des droites sont par contre très différentes selon le point considéré. Elles correspondent effectivement à un comportement linéarisé du fluide à la paroi.

Ce caractère local et très variable de la pente nous laisse prédire que la méthode d'évaluation locale d'une approximation du complément de Schur sera plus efficace qu'une méthode prescrivant un coefficient de Fourier global et constant pour la paroi couplée. Nous vérifierons cette bonne adaptation du coefficient dans la suite de ce chapitre.

Il est important de noter que ces relations flux–température sont constatées sur des résultats de calcul couplés par une méthode Dirichlet→Neumann : nous ne sommes donc pas en train d'observer des conséquences de la méthode Dirichlet→Fourier .

8.5.2 Résultats

Nous remplaçons la condition de flux couplé à l'interface par une condition de Fourier. Nous pouvons ainsi reprendre l'ensemble des calculs précédents, pour voir l'apport de cette nouvelle condition de raccord.

Nous commençons par reprendre les premiers calculs de la section 8.1. Il s'agit donc de calculs à fréquence fixe, sans vérification de convergence ni critère de correction.

Le premier calcul (Fig. 8.22), présentant un résultat pour une fréquence de 10 couplages par seconde, arrive à obtenir un résultat jusqu'à la seconde 1,4, bien qu'il présente des oscillations à partir de la première seconde.

Durant cette première seconde, l'erreur sur le flux est raisonnable, inférieure à 10%. Cette situation est à comparer avec l'instabilité dès la seconde 0,8 de la méthode Dirichlet→Neumann (cf. Fig. 8.4).

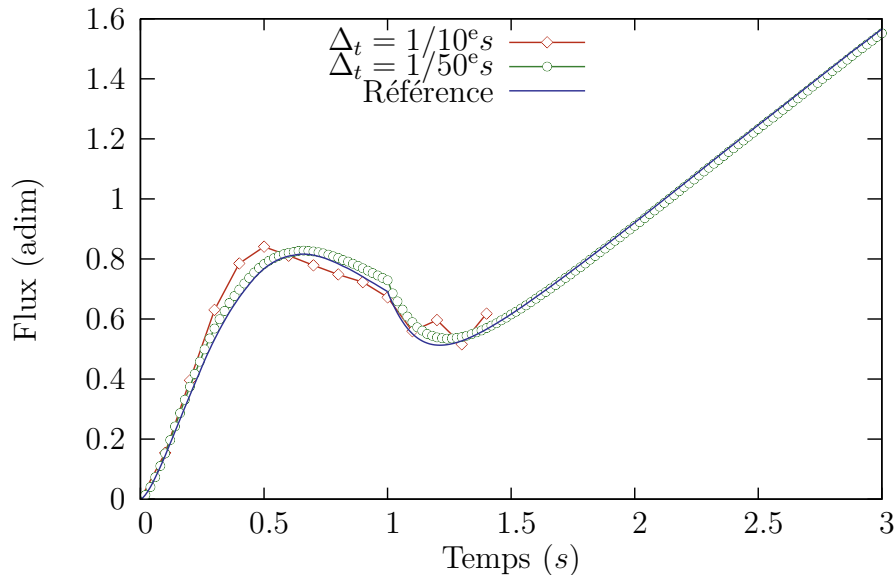


FIG. 8.22 – Résultats obtenus par la méthode Dirichlet→Fourier pour différentes fréquences de couplage ($\Delta_t = 1/10^e s$ et $\Delta_t = 1/50^e$).

Pour une fréquence de 50 couplages par seconde et sans corrections, le calcul est stable et permet d'obtenir des résultats très proches de la solution de référence ; rappelons que par la méthode Dirichlet→Neumann, cette fréquence conduisait à des instabilités au-delà de la première seconde (cf. Fig. 8.5). Ceci souligne donc l'apport en stabilité de cette condition de raccord mixte. Enfin, pour des fréquences de couplage plus importantes, le calcul par Fourier ou Neumann présentent des résultats similaires.

Une deuxième série de calculs, utilisant de plus la méthode du pas de temps adaptatif, nous a permis de dépasser la sixième seconde, en des temps de calcul raisonnables. Au-delà de cette sixième seconde, les prédicteurs linéaires et quadratiques se sont montré instables, alors même que l'ensemble du calcul tend à se stabiliser. Nous avons donc continué les simulations avec le prédicteur d'ordre 0, et avons finalement pu mener à bien l'ensemble des 600 secondes de simulation.

Cette série de calculs nous permet de comparer les coûts respectifs de différentes solutions. Comparons tout d'abord les méthodes à coefficient α fixé (Fig. 8.23). Nous constatons que durant la première seconde, l'ensemble des méthodes présente un coût comparable. En effet, nous avons bien constaté sur la figure 8.21 que la relation affine entre flux et température n'est pas encore établie pendant cette première seconde. En fait, cela est aussi lié à l'erreur que nous faisons quant au choix de la température de référence à cet instant du calcul : nous la prenons simplement égale à $50K$, alors que la température d'entrée est encore en train d'évoluer. Il sera à terme nécessaire de modifier la condition de couplage, afin qu'elle puisse accepter une température de référence variant au cours du temps, ou, ce qui serait encore préférable, que cette température de référence soit estimée par le solveur fluide en cours de calcul.

Au-delà de cette première seconde, le coût de calcul est visiblement minimal pour $\alpha = 10^5$. Pour des valeurs de α entre 0 et 1000, le nombre total d'itérations nécessaires est à peu près comparable.

L'utilisation d'un coefficient de Fourier local permet d'améliorer encore ces performances

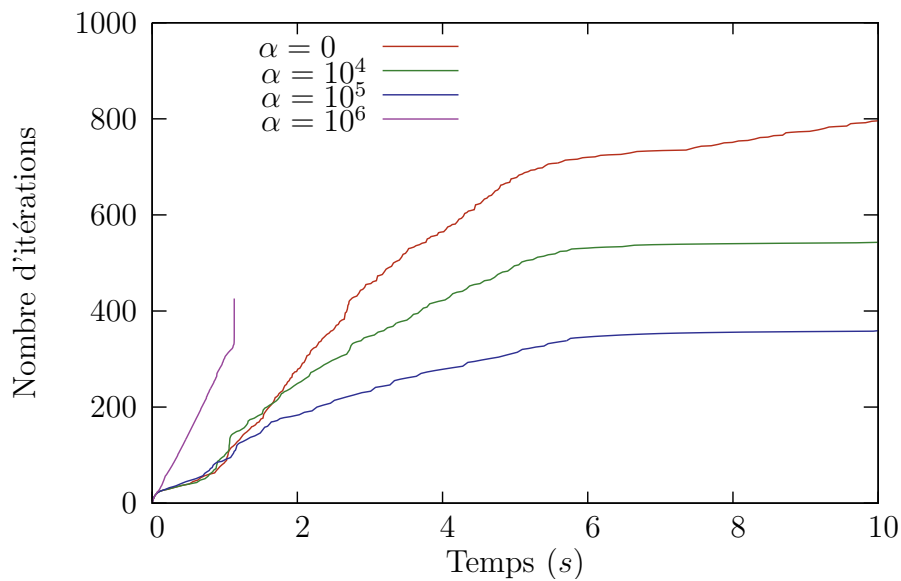


FIG. 8.23 – Coût de la méthode Dirichlet→Fourier pour différentes valeurs du coefficient d'échange α . Le coefficient est choisi constant : 0, 10^4 et 10^5 , le cas $\alpha = 0$ étant rigoureusement équivalent à la méthode Dirichlet→Neumann. Durant la première seconde de calcul, les différents choix possibles pour α ont des coûts similaires. Ce n'est qu'une fois les conditions d'entrée du fluide stabilisées que la méthode de Fourier montre son efficacité. Les courbes pour α valant 10, 100 et 1000 ne sont pas représentées ici : elles sont toutes les trois extrêmement proches de celle pour $\alpha = 0$. Une valeur de α trop importante, par exemple 10^6 , déstabilise le calcul.

(Fig. 8.24). Le résultat est obtenu en deux fois moins d'itérations que par la méthode de Dirichlet→Neumann. Contrairement à nos attentes, son coût n'est par contre que légèrement moins important que pour une méthode à coefficient d'échange fixé bien choisi (en l'occurrence $\alpha = 10^5$). Le principal avantage de cette détermination locale du coefficient d'échange est donc qu'il nous épargne une évaluation de α . Cette évaluation a priori est basée soit sur des calculs analytiques sur modèles simplifiés, soit sur des essais numériques (comme dans la figure 8.23) ; dans un cas comme dans l'autre, c'est une évaluation non triviale.

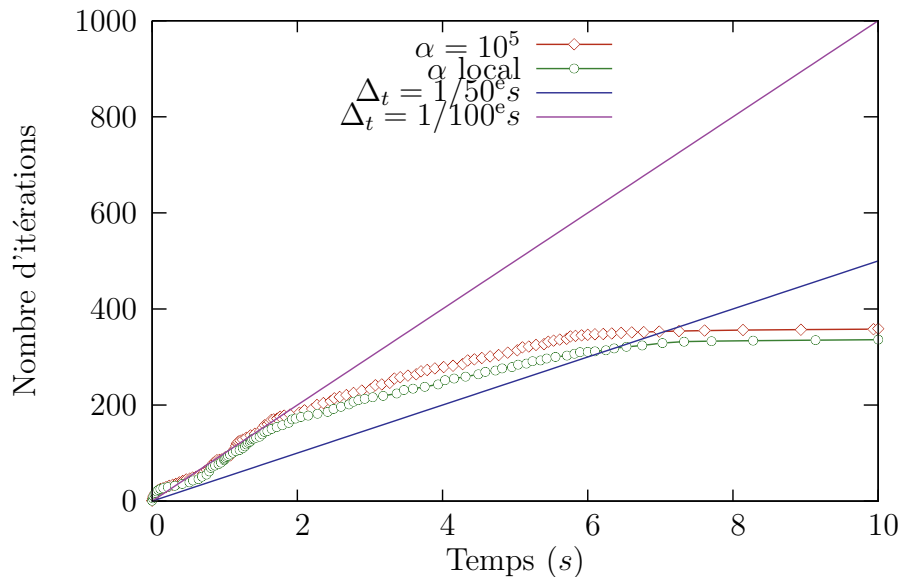


FIG. 8.24 – Coût de la méthode Dirichlet→Fourier pour un coefficient α approximant localement le complément de Schur du fluide. Pour comparaison, les coûts des méthodes sans corrections à fréquence fixe ont aussi été représentées (rappelons qu'à une fréquence de 50 couplages par seconde, le calcul est instable après la 1^e seconde).

8.6 Conclusion

La combinaison des deux méthodes, pas de temps adaptatif et couplage mixte, nous permet finalement de réaliser la simulation complète de cette thermique du tube en un temps de calcul que nous pourrions considérer comme raisonnable.

Nous avons été contraints d'utiliser un prédicteur d'ordre 0, afin de pouvoir augmenter librement la taille des pas de temps au-delà de la sixième seconde. Le gain d'un prédicteur linéaire ou quadratique n'est pas apparu flagrant sur cette application.

En fait, la géométrie du tube avec son insert complique de manière importante le couplage thermique : plusieurs zones géométriques apparaissent, pour lesquelles le comportement couplé est radicalement différent. Certaines zones voient un flux de chaleur à la paroi variant de manière lisse, alors que d'autres sont victimes des turbulences de l'écoulement. Bien que cela risque de compliquer énormément ce problème couplé, il faudrait idéalement envisager un traitement local des prédicteurs pour pallier ces problèmes.

Il semble en tout cas que sur cette application, le meilleur algorithme de couplage à notre disposition soit l'assemblage :

- d’une condition de raccord Dirichlet→Fourier avec coefficient local ;
- d’un critère de convergence absolu ;
- d’un pas de temps adaptatif ;
- d’un prédicteur constant.

Un certain nombre de points restent problématiques, la seconde 0,85 en particulier. Les instabilités rencontrées sont causées par quelques cellules fluides, surestimant leur température. Nous n’avons pas réussi à remonter à la source de ce problème, pour trouver la cause de cette surchauffe. Par chance, les méthodes de couplage par conditions mixtes se sont révélées moins sensibles à ces écarts de température, et ont permis de les supporter.

Un certain nombre de points pourraient encore être améliorés. Les quelques tests que nous avons jusqu’ici réalisés sur les critères de convergence n’ont pas permis de trancher clairement entre ceux que nous avons testés. Il est fort probable qu’un critère de convergence en énergie sera plus satisfaisant d’un point de vue physique, et se révélera plus aisé à calibrer.

Concernant les prédicteurs, il n’est pas non plus satisfaisant de ne pas pouvoir envisager une prédiction quadratique. Dans les zones de calcul où l’évolution temporelle du flux est d’allure quadratique, on s’attend à des bénéfices en stabilité, précision et donc en temps calcul. Son mauvais comportement en un nombre réduit de zones critiques (à proximité des angles par exemple), alors même que l’ensemble des conditions au bord sont stabilisées, nous en dénie malheureusement l’accès. D’autres prédicteurs plus évolués, par exemple une régression sur les cinq derniers champs reçus, pourraient éventuellement être un bon compromis dans ces situations.

Enfin, la méthode de correction que nous utilisons est brusque : nous réalisons une correction complète des flux reçus. Dans la situation présente, la relaxation n’est pas nécessaire, et serait même pénalisante puisque nos courbes de convergence sont monotones. Mok *et al.* [47] ont d’ailleurs suggéré d’utiliser la méthode d’accélération d’Aitken, particulièrement efficace pour accélérer la convergence de suites monotones ou oscillantes.

Nous pouvons aussi noter que des méthodes de recherche de points fixes plus efficaces (telles que les méthodes de Newton) pourraient fournir d’encore meilleurs résultats. Il reste cependant assez difficile dans notre cas d’évaluer un opérateur tangent au problème couplé. C’est donc une méthode qu’il sera plus complexe à mettre en œuvre.

Chapitre 9

Calculs couplés aéro-thermo-mécaniques

Nous avons précédemment montré la faisabilité de calculs couplés à deux codes, aéro-mécanique et aéro-thermique, sur la situation de la tuyère. Il est donc désormais possible de s'atteler à la question du couplage complet aéro-thermo-mécanique sur un tube. Celui-ci nécessite l'ensemble des échanges présentés précédemment : un couplage par température-déplacements entre les solveurs mécanique et thermique du solide, un couplage par pression-déplacements entre le solveur mécanique et le solveur fluide et enfin un couplage par flux-température entre le solveur thermique et le solveur fluide. Ces échanges sont résumés sur la figure 9.1.

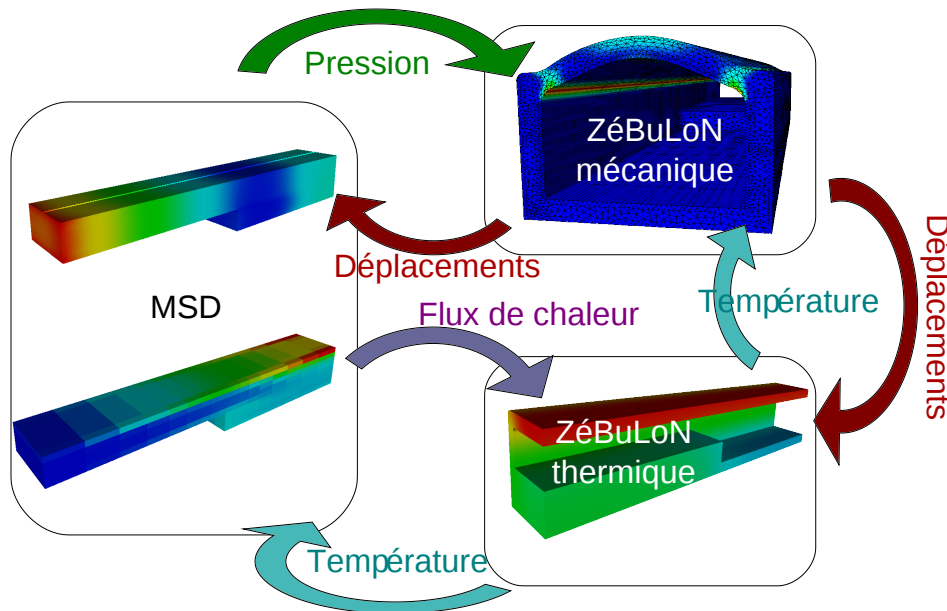


FIG. 9.1 – Différentes interactions, dans le cas du couplage aéro-thermo-mécanique

Nous avons vu que parmi ces trois couplages, certains sont plus forts que d'autres : le couplage aéro-thermique est particulièrement marqué, mais aussi l'influence de la thermique sur la mécanique du solide. L'influence du champ de pression sur les déformations du tube est

aussi importante, mais à sens unique : nous avons vu que les déformations du tube n'ont que de faibles conséquences sur l'évolution du fluide dans le tube. De même, les déformations n'ont que peu d'influence sur la thermique du tube, du moins tant que les conditions appliquées du côté de la paroi chaude ne tiennent pas compte de l'effet "sous le vent"¹ que subissent réellement les tubes.

La prise en compte des trois types d'interactions a aussi pour effet de renforcer certains couplages qui étaient jusque-là relativement faibles : la mécanique par exemple, dépendante du champ de pression reçu du fluide, est aussi fortement liée à l'évolution de sa température qui est elle liée au fluide. L'influence du calcul fluide sur la partie mécanique s'en trouve donc fortement renforcé.

Nous commencerons l'étude de ce couplage à trois codes par un premier cas de couplage simplifié, dans lequel les axes du couplage les moins marqués sont négligés. Le cas complet nous a posé des difficultés de synchronisation des différents codes ; nous l'aborderons donc dans un deuxième temps.

9.1 Calcul aéro-thermo-mécanique simplifié

9.1.1 Présentation des simplifications

Comme nous le précisons précédemment, l'influence des déformations du tube sur le calcul aéro et sur le calcul thermique est très faible. Ce sont donc ces deux flèches que nous nous permettons de négliger dans ce calcul simplifié, tout en conservant une bonne représentativité du modèle. Le schéma d'influence résultant est présenté sur la figure 9.2.

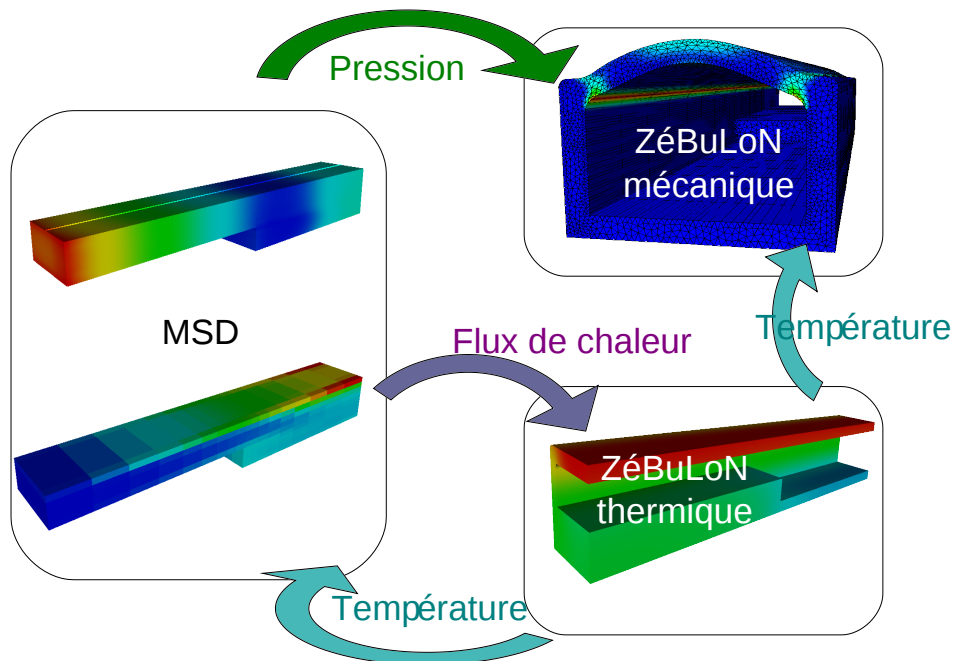


FIG. 9.2 – Couplage simplifié à trois codes

¹ Le gonflement des tubes rend la paroi interne du divergent ondulée. Les gaz chauds viennent impacter ces rugosités, et augmentent notablement la température sur la moitié la plus exposée du tube.

Ce cas conserve tout de même le couplage aéro-thermique complet, ainsi qu'une double influence du code aéro et du code thermique sur le calcul mécanique. Les simplifications permettent néanmoins de simplifier toute une branche du couplage, puisque le code mécanique se retrouve entièrement assujéti aux deux autres.

Le problème ainsi simplifié pourrait s'aborder en considérant dans un premier temps le couplage aéro-thermique (problème résolu au chapitre précédent), à la suite duquel nous enchaînerions un calcul mécanique seul, avec application des paramètres (pression sur la paroi interne et champ de température) issus du calcul précédent. Cependant, notre objectif est d'aboutir à un calcul couplé complet, et nous ne nous autorisons donc pas cette facilité. Nous désirons conserver la résolution simultanée des trois problèmes, au travers du système d'échange de données.

9.1.2 Algorithme de couplage retenu

Le couplage aéro-thermique étant au centre de notre problème simplifié, et étant surtout le point problématique du calcul, nous allons dériver le nouvel algorithme de couplage à trois codes de celui utilisé pour le cas thermique (cf. section 8.6). Rappelons que celui-ci intégrait :

- une condition de raccord Dirichlet→Fourier avec coefficient local ;
- un critère de convergence absolu ;
- un pas de temps adaptatif ;
- un prédicteur constant.

Nous y ajoutons simplement les deux transferts vers la partie mécanique : transfert du champ de pression et transfert de la carte de température. Les transferts étant peu coûteux par rapport aux temps de calcul, nous choisissons de les effectuer à chaque pas de temps du couplage thermique.

L'avancée de la résolution mécanique va être liée à l'avancée des deux autres codes. En effet, les simplifications que nous avons faites rendent inutile toute phase de prédiction du côté solide. En contrepartie, celui-ci doit simplement attendre que les deux autres codes aient avancé suffisamment pour lui fournir les champs couplés. Par contre, étant donné les petits pas de temps nécessaires au couplage thermique, nous conservons une horloge de résolution indépendante. Ainsi, le code mécanique utilise son propre système de pas de temps automatique (limitant l'augmentation des déformations plastiques cumulées). Le système de pile de réception permet ce genre de désynchronisation, comme nous l'avons vu lors de la mise en place du couplage thermo-mécanique (section 5.2).

Du point de vue du code thermique, il nous faudra d'ailleurs différencier les envois de champs au code fluide (qui sont des champs en cours de validation), de ceux à envoyer au code solide (qui ne devront être que des champs valides).

L'ensemble de ces considérations nous conduisent à l'algorithme présenté sur la figure 9.3. Son implantation dans chacun des codes est détaillée dans les figures suivantes. Le code fluide (Fig. 9.4) se contente de rajouter une phase d'envoi de son champ de pression, en même temps que le flux de chaleur à l'interface. Par facilité, tous les champs calculés sont envoyés (et pas uniquement ceux correspondant à une situation aéro-thermique convergée). Ce sera au code mécanique de faire la distinction.

Dans le code thermique (Fig. 9.5), nous rajoutons en fin d'incrément une phase d'envoi du champ de température, lorsque l'état convergé aéro-thermique est atteint.

Dans le code mécanique (Fig. 9.6), nous mettons en place en début d'incrément une boucle réceptionnant le champ de température. Cette boucle continue, jusqu'à ce que le champ reçu

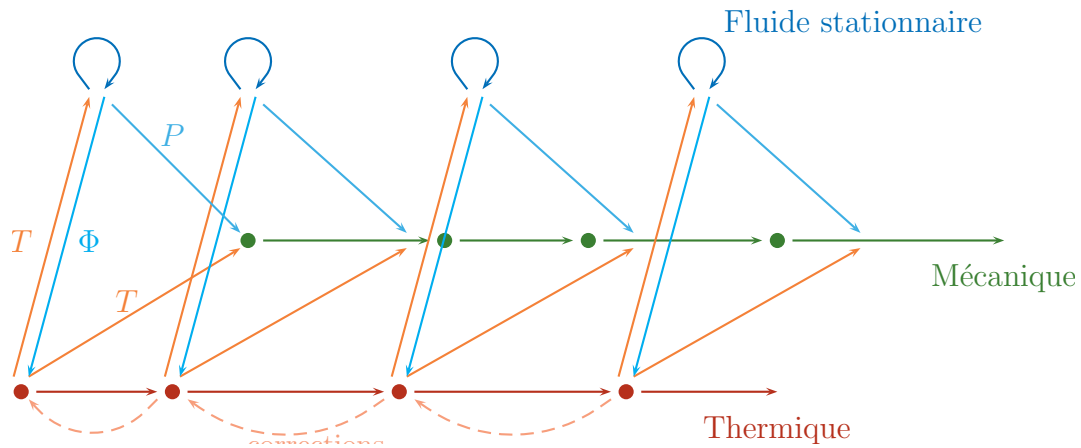


FIG. 9.3 – Algorithme de couplage simplifié à 3 codes : T désigne les envois du champ de température pariétale, P celui du champ de pression, et Φ celui du flux de chaleur.

permette de réaliser l'incrément courant sans avoir besoin d'extrapolations de ce champ. Autrement dit, le code attend de recevoir un champ thermique ayant une date ultérieure à sa fin d'incrément. Une fois ce champ reçu, le code reçoit l'ensemble des champs de pression qui ont été envoyés par le code fluide. Les champs correspondant à des états intermédiaires seront ainsi écrasés par le dernier champ reçu, solution fonctionnelle bien que peu élégante de contourner le problème cité ci-dessus. Une fois ces réceptions réalisées, les champs sont au besoin interpolés en début et fin d'incrément, afin de permettre la complète désynchronisation des codes. Pour finir, l'incrément est réalisé de manière usuelle.

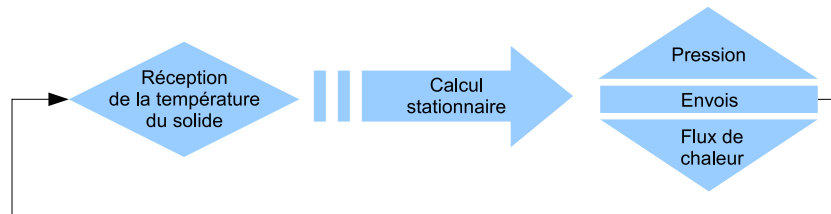


FIG. 9.4 – Mise en place de l’algorithme de couplage dans le code fluide

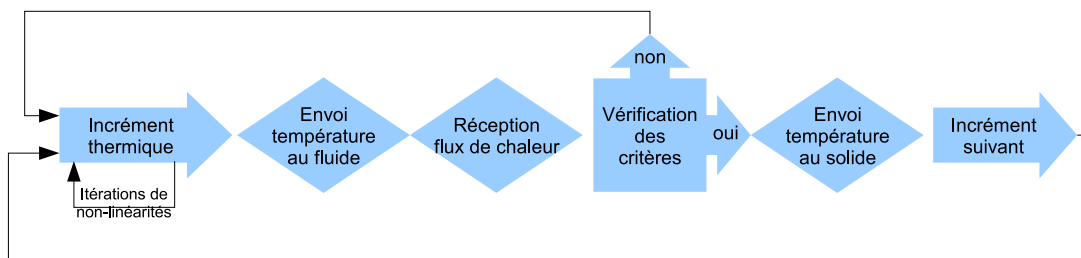


FIG. 9.5 – Mise en place de l’algorithme de couplage dans le code thermique

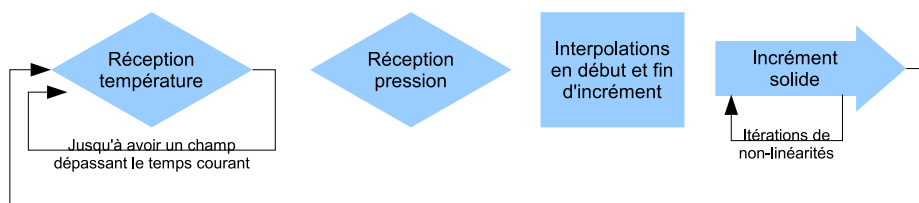


FIG. 9.6 – Mise en place de l’algorithme de couplage dans le code mécanique

9.1.3 Résultats

Le calcul couplé utilisant l'algorithme que nous venons de décrire, ainsi que les différentes mises en données déjà présentées lors des couplages à deux codes a été mené jusqu'à la 27^e seconde. À cet instant, le code mécanique a atteint une phase pour laquelle il n'arrive plus à converger. Pour le moment, nous n'avons pas mis en évidence la cause de ce problème. Les champs reçus sont corrects, et d'ailleurs, le problème thermique a atteint depuis la 7^e seconde un état parfaitement stabilisé, et les valeurs atteintes sont raisonnables. Il ne semble donc pas que la cause du problème soit physique ; elle est probablement due à une erreur informatique.

Nous allons donc baser notre discussion sur les résultats que nous avons à disposition à la 20^e seconde. La figure 9.7 présente une image représentative de chacun des trois codes mis en jeu : les déplacements verticaux, le flux de chaleur à l'interface et la température de la structure.

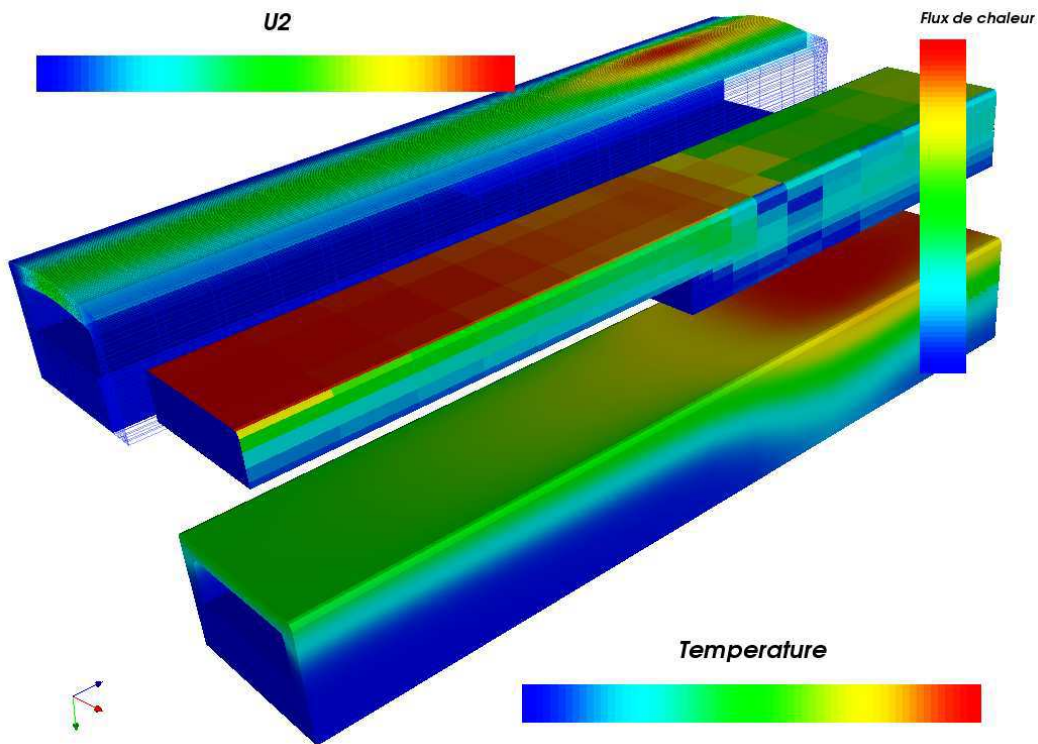


FIG. 9.7 – Solution du calcul couplé à trois codes, obtenue à la 20^e seconde. À cet instant du calcul, le problème couplé thermique s'est bien stabilisé ; en revanche, le problème mécanique commence vraiment à évoluer, sa température ayant atteint un état assez chaud pour rendre le métal suffisamment mou.

Le couplage mécanique n'étant qu'à sens unique, nous retrouvons exactement les résultats thermiques déjà obtenus précédemment. Le champ de température ainsi que les flux pariétaux sont stabilisés à partir de la 7^e seconde, et atteignent au point le plus chaud une température de 1386 K.

Du point de vue mécanique par contre, nous gagnons ici une carte thermique plus riche que dans les cas de couplage purement mécanique traités au chapitre 7. Les déformations observées sont donc bien différentes.

Les courbes de la figure 9.8 montrent par exemple que les déplacements mesurés au sommet du tube sont ici 30 fois moins importantes que dans le cas de couplage mécanique en entrée de tube, et 10 fois moins importantes en sortie de tube.

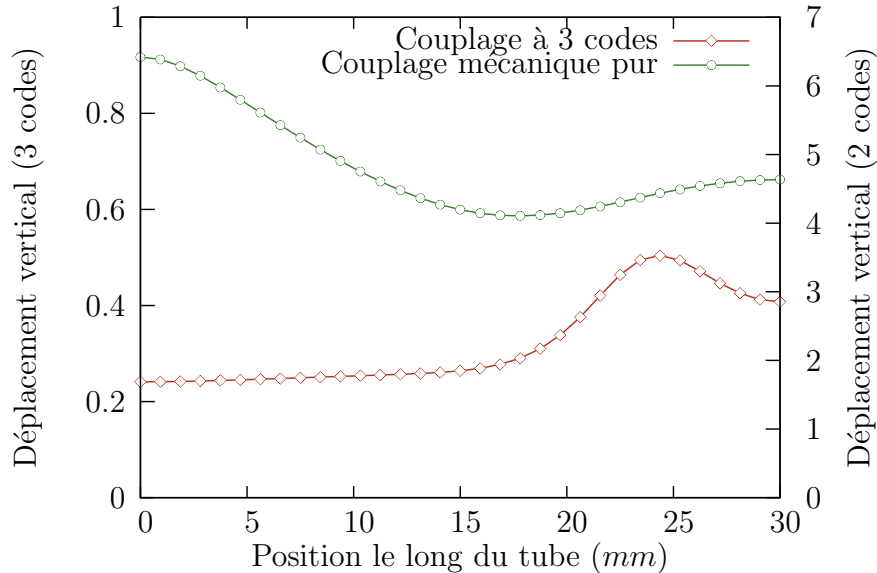


FIG. 9.8 – Déplacement le long de l’axe principal du tube, autour de la 20^e seconde. La première courbe est issue du calcul à 3 codes, alors que la deuxième est issue du calcul uniquement couplé mécaniquement. Les deux calculs fournissent des résultats différents d’un ordre de grandeur ; nous les présentons donc à deux échelles différentes.

Cette énorme différence s’explique par la différence entre les températures vues par la structure dans les deux calculs : nous l’avions choisie uniforme le long du tube et variant entre 1000 K et 1300 K dans l’épaisseur ; ici, le calcul thermique nous fournit un champ de température variant entre 50 K et 1386 K , les variations se faisant à la fois dans l’épaisseur et dans la longueur (Fig. 9.9).

Dans la première moitié du tube, où la température ne dépasse pas les 1000 K , le matériau est suffisamment froid pour n’être que très peu sensible au fluage. La deuxième moitié du tube voit par contre des températures plus élevées, au-delà de 1000 K . Le profil de déformations s’en ressent : les déplacements en sortie de tube sont deux fois plus importants qu’en entrée.

La courbe de température le long du tube (Fig. 9.10) a une allure très similaire à celle des déplacements (première courbe de 9.9). Cela nous montre que dans cette simulation, les déformations du tube sont très liées à sa température : rappelons que les coefficients matériau de la loi de comportement utilisée en dépendent fortement. Attention toutefois à ne pas se méprendre : les déformations sont toujours causées par le fluage, mais celui-ci n’est important que dans les zones chaudes.

9.2 Calcul aéro-thermo-mécanique complet

Le couplage aéro-thermo-mécanique complet est celui représenté sur la figure 9.1. Il fait intervenir la totalité des interactions que nous avons jusqu’ici considérées :

- l’interaction mécanique entre le fluide et le solide,

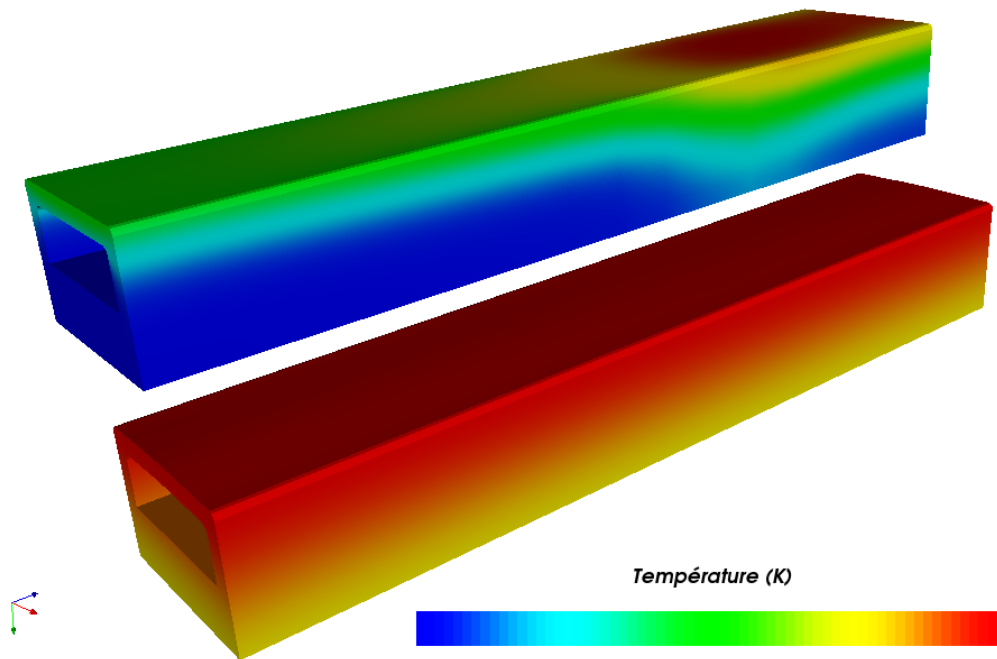


FIG. 9.9 – Comparaison des champs de température appliqués dans le calcul couplé mécanique (en bas) et le calcul couplé à trois codes (en haut)

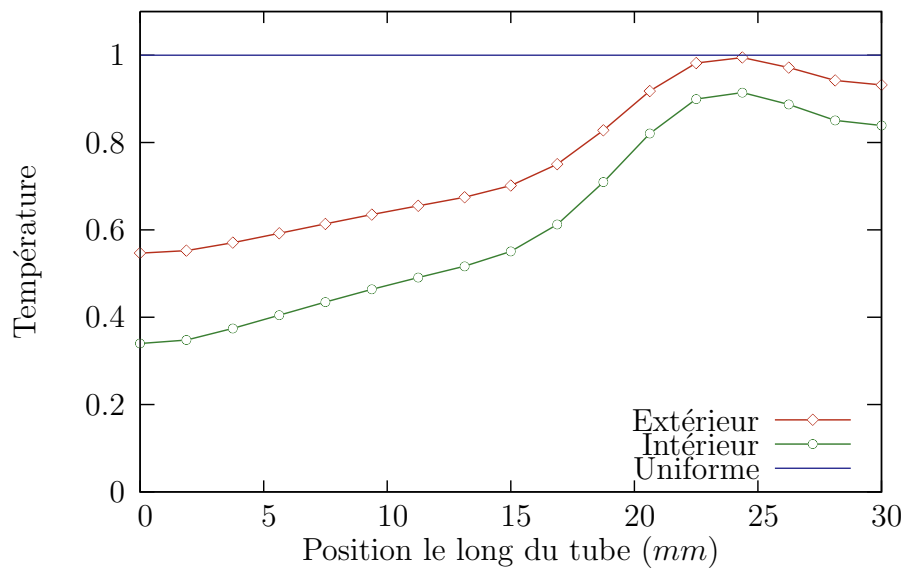


FIG. 9.10 – Température le long de l'axe principal du tube à $t = 20s$, au niveau de la paroi chaude extérieure et de la paroi chaude intérieure. Pour le calcul couplé mécaniquement, la température était uniforme, représentée ici par la ligne bleue.

- l'interaction thermique entre le fluide et le solide,
- l'interaction thermo-mécanique dans le solide.

Cette fois-ci, nous ne négligeons plus l'influence du calcul mécanique sur les deux autres.

9.2.1 Algorithmes envisagés

Plusieurs approches ont été tentées pour résoudre ce problème : tout d'abord, nous avons essayé de mettre en place une stratégie de résolution totalement asynchrone. Repartant de l'algorithme simplifié précédent (Fig. 9.3-9.6), celle-ci va incorporer en plus la publication des données du code mécanique.

Reprenons donc le schéma de couplage 9.3, et ajoutons-y les envois issus du code mécanique. Nous obtenons l'algorithme de la figure 9.11.

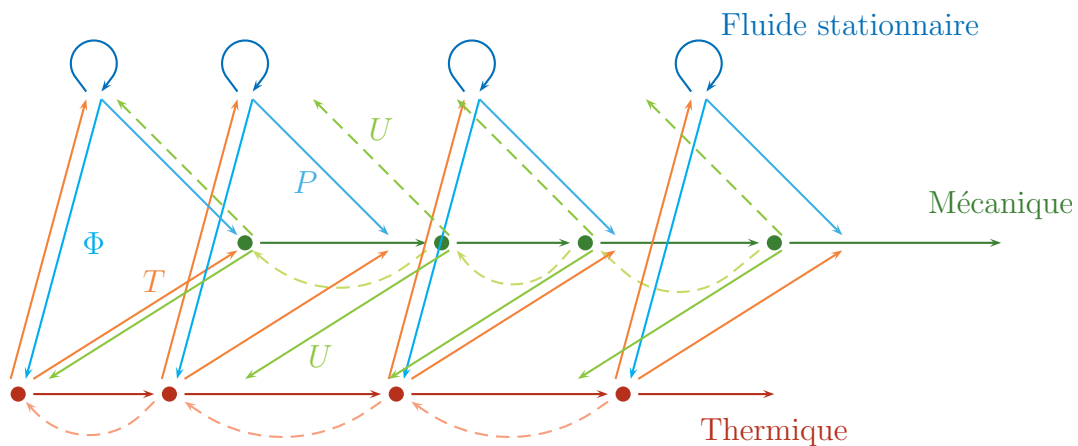


FIG. 9.11 – Algorithme de couplage à 3 codes. Que faire des champs de déplacements désynchronisés reçus par le fluide ?

La principale difficulté de ce nouvel algorithme apparaît immédiatement : les champs de déplacements envoyés par le solide au fluide ne correspondent pas à des instants où le fluide est calculé. Il faut donc prendre des mesures.

Une première solution (Fig. 9.12) consiste à ajouter des calculs d'états fluides stationnaires autour de ces instants choisis par l'horloge de la mécanique. Autrement dit, nous complétons l'algorithme précédent avec l'équivalent mécanique de ce qui avait été fait en thermique. Cette solution présente principalement deux désavantages. Premièrement, elle augmente le nombre d'états fluides calculés, ce qui n'est pas forcément nécessaire, surtout si les nouveaux états sont calculés à des instants très proches des anciens. Deuxièmement, elle nécessite dans le code fluide un système de réinterpolations des champs analogue à celui développé dans ZéBuLoN. Il faut en effet avoir à disposition un champ thermique pour calculer l'état stationnaire à partir d'un champ mécanique reçu. C'est un investissement non négligeable, qui n'a pas encore été envisagé. Toutefois, il semble qu'il faille en passer par là si l'on veut pouvoir s'attaquer à ce type de problèmes.

Si l'on avait à disposition un tel système, une deuxième solution devient envisageable. Elle consiste à ne plus considérer le fluide comme assujéti aux deux autres codes. C'est donc dorénavant le code fluide qui détermine à quels instants il va calculer son état stationnaire. Ceci permettrait de rendre le couplage parfaitement symétrique : plus aucun code ne se retrouve

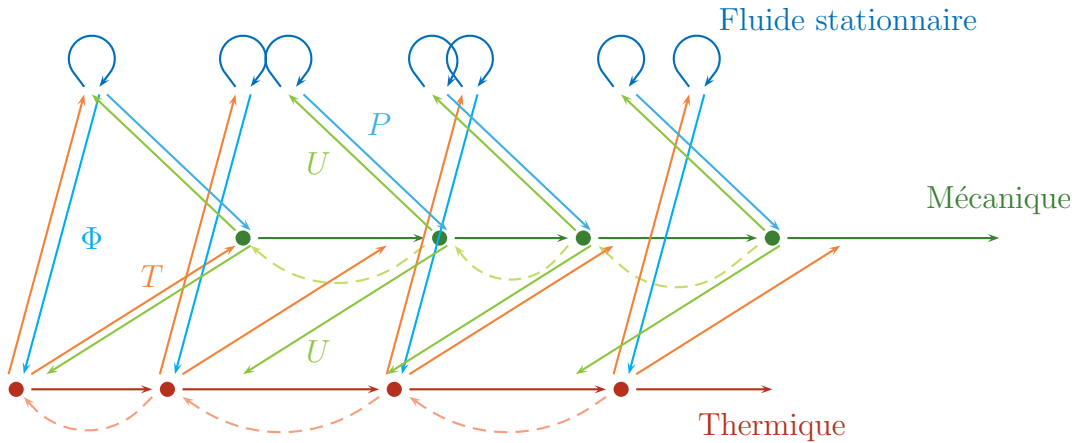


FIG. 9.12 – Algorithme de couplage à 3 codes : première solution. Cette solution augmente le nombre d'états fluides calculés, et nécessite un système de désynchronisation dans le code fluide.

pilote ou esclave des autres, et l'algorithme de couplage s'en retrouve simplifié (Fig. 9.13). Nous avons déjà testé cette méthode dans le cas du démonstrateur couplant trois instances de ZéBuLoN (voir section 5.3), situation pour laquelle elle donnait des résultats satisfaisants.

Pour pouvoir mettre en place un tel calcul, il sera cependant nécessaire de développer dans MSD les méthodes permettant la détermination d'instantanés judicieux de calcul d'états stationnaires. Par judicieux, nous entendons ici un nombre suffisant d'états pour que l'interpolation temporelle linéaire entre deux états calculés rende compte de manière satisfaisante de l'évolution temporelle du système. C'est à nouveau un investissement de développement conséquent, auquel nous ne nous sommes pas attaqués. C'est aussi une question qui doit être vue de manière indépendante au couplage. Une telle méthode de simulation de l'évolution du système par suite d'états stationnaires peut en effet avoir d'autres applications que celles du couplage.

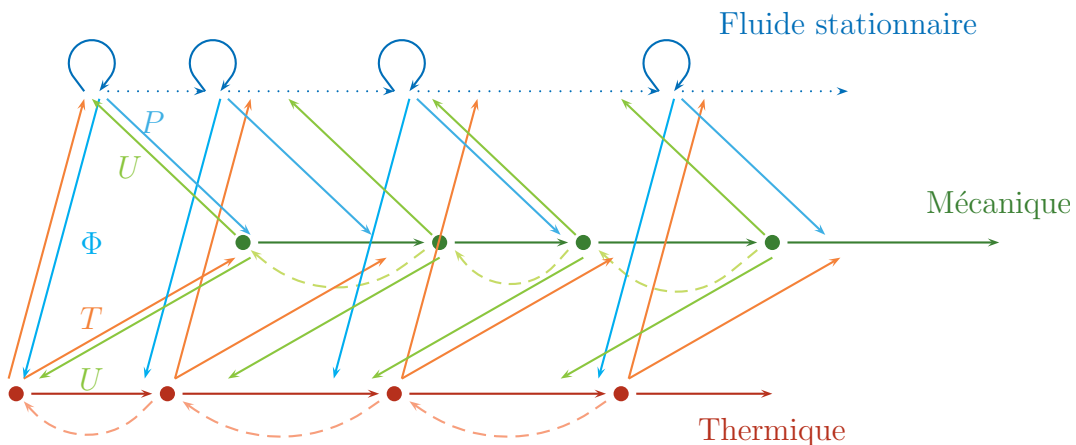


FIG. 9.13 – Algorithme de couplage à 3 codes : deuxième solution. Cette solution rend au code fluide son indépendance vis à vis des autres codes.

Cette solution consistant à désynchroniser tous les codes facilite la description de l'algo-

rithme : les codes envoient les champs aux instants où ils en disposent, et reçoivent tous les champs éventuellement disponibles de l'extérieur. En contrepartie, la gestion des critères de convergence devient plus complexe. En effet, la transmission de champ doit être accompagnée d'une information complémentaire, précisant si le champ reçu est un champ validé, ou en cours de validation. Chaque incrément de calcul se retrouve soumis à cette variable : un champ calculé à partir de données en cours de validation devra être lui aussi validé. Le calcul traînera donc dans sa suite un arbre de validation qu'il faudra confirmer de proche en proche.

Cette situation semble traitable en théorie ; cependant nous sommes conscients que le passage à l'implantation d'une telle solution fera certainement apparaître de nouvelles difficultés.

9.2.2 Solution utilisable dans l'état actuel

Dans l'état actuel, il est toutefois possible d'envisager une solution ne nécessitant pas de développements importants supplémentaires dans le code fluide. Il s'agit de synchroniser les trois codes sur le code le plus lent. En pratique, cela nécessite donc un échange des horloges de calcul entre les codes thermique et mécanique, afin qu'ils s'accordent sur le pas de temps suivant.

Cette solution présente un défaut flagrant : elle est inutilement chère. En début de calcul, la partie mécanique encore froide ne nécessite pas de pas de temps aussi fins que la thermique ; en fin de calcul la situation est inversée : c'est la mécanique qui nécessite de plus petits pas de temps que la thermique. Le coût total du calcul en est donc largement augmenté.

Malgré ce défaut, c'est cette méthode que nous avons mise en place pour réaliser le couplage complet. Précisons l'ordonnancement des tâches :

- le code thermique commence le premier, sur une prédiction des champs de déplacements et du flux de chaleur à l'interface ;
 - le code de mécanique des fluides prend la suite, sur une prédiction du champ de déplacements ;
 - le code mécanique du solide tourne en dernier, sans avoir besoin de réaliser de prédictions.
- L'ensemble des prédictions réalisées sont d'ordre 0. Les vérifications de convergence sont faites au niveau du code thermique : il vérifie à la fois la précision suffisante du prédicteur de flux de chaleur et du prédicteur en déplacements. Le code fluide devrait lui aussi effectuer une vérification de sa prédiction du déplacements de l'interface. Mais comme le code thermique réalise déjà cette même vérification, nous pouvons nous en passer. Enfin, le code mécanique ne travaillant qu'avec des champs à jour, il ne nécessite pas de critères de convergence.

Cette méthode nécessite une stratégie de synchronisation. Un nouvel objet dérivant de la classe `COUPLED_AUTOMATIC_TIME` est donc développé. Son but est d'assurer la synchronisation du code mécanique sur le code thermique. Quand il est interrogé, l'objet va simplement vérifier dans la pile de réception des champs thermiques quel est le dernier pas de temps connu, et proposer cet instant comme prochain pas de temps mécanique. Le *système de pas de temps automatique multiple* se charge ensuite de concilier cette proposition avec le pas de temps automatique assujetti aux non-linéarités matériau.

Pour réaliser effectivement les premières secondes du calcul, nous n'avons besoin que de synchroniser le code mécanique sur le code thermique, ce dernier nécessitant les plus petits pas de temps. Afin de simplifier la mise en données, nous omettons provisoirement d'ajouter l'objet équivalent du côté thermique ; celui-ci sera nécessaire au-delà de la sixième seconde, une fois la thermique stabilisée.

Pour des raisons actuellement inconnues, ce calcul diverge très rapidement, après deux incréments, à la 0,2^e seconde. Le code mécanique a des difficultés à faire converger son point fixe, alors que la température du matériau est basse (nous sommes dans la phase de refroidissement) et le champ de pression est autour de 30 bars, soit une pression tout à fait supportable à cette température. L'origine du problème n'est donc ni conceptuelle, ni physique. Elle est très probablement due à une erreur de gestion au niveau de l'algorithme, qui nous échappe encore. Peut-être une mauvaise interaction justement entre les différents codes ? Une mauvaise compréhension des états non convergés intermédiaires ?

Conclusions
et
Perspectives

Conclusion

Nous avons apporté au travers de ce travail de thèse trois grands axes de réponse concernant le couplage fluide-structure. Il s'agit tout d'abord d'une synthèse des méthodes existantes, et d'une réflexion sur la conception d'une plateforme de simulations couplées. Cette plateforme, associant les deux logiciels MSD pour le fluide et ZéBuLoN pour le solide, reprend les concepts de couplage partitionné : en particulier, elle a été développée en conservant au maximum l'indépendance entre les codes, permettant aujourd'hui d'assurer le remplacement du logiciel MSD par son successeur CÈDRE. Toujours motivés par cette idée d'interchangeabilité, le moteur de couplage utilisé dans ce travail, basé sur MpCCI pour la prise en charge de la réinterpolation des champs ainsi que du transfert de données, a été abstrait pour pouvoir être facilement remplacé ultérieurement par d'autres solutions de couplage. Ces différents points nous ont d'ailleurs conduits à mettre en évidence un cahier des charges à respecter pour coupler plusieurs codes de calcul entre eux.

Mais le principal apport de cette réflexion conceptuelle restera le point de vue *objet* du couplage, permettant une grande souplesse dans la mise au point des algorithmes de couplage. Cette souplesse est indispensable, chaque application présentant ses propres difficultés et nécessitant donc une résolution taillée sur mesure. En conséquence, nous avons développé l'ensemble des stratégies de couplage comme autant de briques assemblables pour pouvoir adresser un grand panel de problèmes couplés.

Ces différentes briques forment maintenant une base, à disposition pour les développements de futures méthodes de couplage. Le fait d'avoir considéré l'ensemble du couplage comme un échange de champs de vecteur, et surtout d'avoir transcrit informatiquement ce concept de champ nous a permis une cohérence de l'ensemble des stratégies de couplage.

La réalisation du moteur de couplage répond à la fois au problème de l'indépendance du couplage vis-à-vis de la méthode de transfert de données, mais surtout propose des solutions pratiques au problème de désynchronisation des codes. Ce moteur est aussi au cœur des solutions proposées pour supporter les non-linéarités, gérées par pas de temps adaptatifs de la résolution du solide.

Une autre brique fondamentale est constituée des stratégies retenues pour l'application des champs reçus. De nombreuses conditions au bord couplées ont été testées : tout d'abord les plus simples, conditions de Neumann et de Dirichlet, appliquant directement les champs reçus de l'autre code. Cette solution est parfaitement satisfaisante dans notre situation de couplage mécanique ; un calcul du contraste entre les comportements des deux domaines nous le laissait d'ailleurs penser a priori.

Dans le cas du couplage thermique, ces conditions de raccord ont toutefois montré leurs limites. Le passage à des conditions de raccord mixtes, ou conditions de Fourier, souvent utilisées par les méthodes de décomposition de domaines, a permis un gain significatif en

stabilité et en rapidité de résolution.

Mais ce cas de couplage thermique a nécessité des efforts supplémentaires pour être résolu de manière satisfaisante en un temps de calcul raisonnable. En nous inspirant des méthodes de choix automatique de pas de temps, classiques en résolution de mécanique non-linéaire des matériaux, nous avons développé une horloge dédiée au couplage, adaptative en fonction des non-linéarités dues au couplage. Cette solution a permis un deuxième gain significatif en temps de calcul ; elle a surtout permis une simulation complète des 600 secondes constituant un vol.

Tous ces développements nous ont finalement permis de proposer trois solutions pour aborder le problème couplé à trois codes. La première, consistant à négliger l'influence des déformations mécaniques sur les deux autres domaines, s'est montrée parfaitement adaptée au cas de l'étude d'un tube de divergent. Les calculs préalables nous avaient en effet montré que cette influence est négligeable. C'est toutefois une simplification qui ne pourra pas être toujours envisagée. La deuxième solution proposée, consistant à synchroniser complètement les trois codes entre eux, a, pour des raisons encore inconnues, conduit à la divergence du calcul dès ses premiers instants. Par manque de temps, nous n'avons pas pu remonter à la source de problème. La dernière solution, consistant au contraire à exploiter complètement une méthode asynchrone n'a été testée avec succès que sur des cas tests simplistes. La simulation d'un tube se montre bien évidemment plus complexe, et nécessitera encore quelques efforts pour être complètement exploitable.

Rappelons, pour finir ce bilan sur les développements réalisés, qu'il serait inconscient de prétendre ces stratégies exhaustives ; nous conservons donc toujours à l'esprit la possibilité de les étendre, afin d'adresser de nouveaux problèmes.

De nombreuses simulations sont venues illustrer ces différents développements : on retiendra tout d'abord quelques cas simples de validation, permettant de vérifier indépendamment le bon fonctionnement des briques de couplage, et surtout d'assurer la non-régression des développements.

Mais les principales simulations ont été concentrées sur le problème des tubes de refroidissement du divergent du moteur Vulcain 2. Malgré les simplifications, parfois grossières, auxquelles nous avons été contraints pour pouvoir réaliser les simulations sur les machines à notre disposition, nous avons pu simuler les trois problèmes couplés apparaissant dans cet exemple : le couplage aéro-mécanique, le couplage aéro-thermique et enfin le couplage complet aéro-thermo-mécanique.

Chacun de ces trois cas a apporté ses difficultés propres : dans le cas aéro-mécanique, la difficulté principale a été de tenir compte des non-linéarités importantes de la loi de comportement du matériau. Les résultats issus de cette situation ont montré une sensibilité non négligeable des déformations subies par le tube, en fonction de la pression imposée par le fluide. Nous avons aussi pu constater que, dans cette situation fortement contrastée, la solution de couplage par échange de pression-déplacements est parfaitement adaptée.

Le cas aéro-thermique s'est montré beaucoup plus délicat à traiter : bien que ne présentant quasiment pas de non-linéarités, les instabilités dues au couplage se sont montrées beaucoup plus présentes. Plusieurs stratégies, rappelées ci-dessus, ont donc été mises en place pour traiter ce problème. Nous avons ainsi pu mesurer les gains respectifs de chacune de ces stratégies, et tailler sur mesure un algorithme de résolution adapté à ce calcul. Le résultat de ce calcul nous montre une carte thermique beaucoup plus complexe que le simple gradient de température que nous avons choisi dans les situations découplées. La présence de l'insert en début de tube

modifie sensiblement l'écoulement du fluide, et donc ses capacités réfrigérantes.

La conjonction des deux couplages précédents nous a enfin permis de réaliser le couplage aéro-thermo-mécanique sur ce cas du tube. Négliger l'influence du calcul mécanique sur les deux autres codes nous a permis de réaliser une simulation complète des 600 secondes de vol. Nous retiendrons particulièrement la très grande influence de la carte thermique sur les déformations observées. Les déformations obtenues ici sont en effet très différentes de celles calculées par un couplage purement mécanique. Cela souligne encore une fois la complexité des phénomènes apparaissant dans cette situation de la tuyère.

L'ensemble de ces réalisations n'aurait pas été possible sans la coopération entre les trois départements impliqués dans le plan de recherche fédérateur CoCoMAT. Les deux départements DSNA et DMSM apportent chacun leur code de calcul. Le département DTIM apporte quant à lui son expérience sur les questions de décomposition de domaines, et plus généralement sur les méthodes numériques. L'exemple le plus représentatif de cette coopération a concerné la réalisation du module de déformation de maillages : ce problème est apparu pour la simulation d'écoulements fluides sur domaine mobile, la méthode de déformation a été inspirée par un calcul de structure, et la solution finalement implantée a utilisé une librairie développée au DTIM.

Mais les exemples de transfert de compétences entre départements ont été nombreux : les instabilités de schémas numériques d'ordre élevés sont par exemple habituels en mécanique des fluides ; nous avons retrouvé ce même type de difficultés concernant les prédictions de champs couplés. Les méthodes de pas de temps adaptatifs qui ont été utilisées pour le couplage sont elles empruntées au code de calcul de structure ZéBuLoN. Enfin, l'utilisation d'une condition de raccord mixte, qui s'est montrée indispensable dans le cas de couplage thermique, est quant à elle directement inspirée de ce qui se fait en calcul multi-domaines.

La recherche et la simulation de tels problèmes couplés nécessite des compétences dans chacun de ces domaines. Il sera donc important de conserver par la suite cette coopération transverse, pour pouvoir proposer des solutions à la fois bien adaptées et efficaces.

Perspectives

Nous avons ainsi mené à bien la simulation d'un tube de la tuyère Vulcain 2. Il serait maintenant temps de défaire les simplifications géométriques : repasser à un tube complet, de 2 mètres de long dans sa géométrie courbe, puis à terme de rajouter les tubes voisins en contact, contact que l'on sait avoir une importance significative dans les déformations finales. Mais ces extensions vont impliquer des calculs de grande taille, nécessitant un recours à des méthodes de calcul parallèle. L'ensemble des développements ont été faits en tenant compte de ce point, le calcul parallèle étant aujourd'hui un passage incontournable pour toute application de grande taille. Il restera donc à voir si la mise en pratique se révèle transparente, ou si elle apporte son lot de difficultés. Si l'on désire approfondir cette étude de la tuyère, il faudra aussi envisager d'enrichir la description du fluide, à la fois en terme de comportement (l'hydrogène de refroidissement est dans un état supercritique, bien loin du modèle gaz parfait que nous avons choisi), de modèles de turbulence et de lois de parois.

À plus court terme, le passage au couplage entre ZéBuLoN et CÈDRE devra être envisagé. Le développement du logiciel MSD a en effet été arrêté depuis quelques années maintenant au profit de son successeur. Le couplage thermique avait déjà été effectué, mettant en pratique l'interchangeabilité des codes. Il reste à rajouter les capacités de couplage mécanique, sujet qui est en cours d'implantation ; en particulier, l'intégration du module de déformation de maillage et de prise en compte de la formulation Lagrangienne-Eulerienne dans CÈDRE, prérequis indispensable au couplage mécanique, est réalisée en ce moment même. D'autres types de couplages commencent aussi à être envisagés au sein de l'ONERA : citons en particulier un couplage radiatif avec le logiciel Astre.

Ces extensions seront d'ailleurs l'occasion d'aborder de nouvelles applications couplées. Elles permettront de voir dans quelle mesure les développements et méthodes présentés dans cette thèse se transposent à d'autres applications. Nous les avons conçus dans cette optique de généralité et de réutilisabilité, mais seule leur utilisation sur des exemples concrets permettra de trancher sur cet aspect. Ces nouvelles applications devront aussi être l'occasion de confronter les calculs couplés à des expériences de validation de ce type de calcul couplé.

D'un point de vue plus applicatif, de nombreux points peuvent encore être améliorés : les critères de convergence que nous avons utilisés sont trop simplistes, et l'introduction d'un critère en énergie permettrait de rendre plus physique la détection de convergence du calcul couplé.

La condition de raccord mixte est actuellement bridée : par simplicité, nous nous sommes restreints à une température de référence constante. Il faudra rapidement envisager de pouvoir la spécifier variable en temps et en espace : nous avons effectivement remarqué qu'elle n'apportait aucun gain significatif pendant la mise en régime du moteur. Introduire une température variable, mais spécifiée manuellement, est un développement presque trivial. Mais idéalement,

et c'est là un développement plus conséquent, cette température devra être évaluée directement par le code fluide puisque c'est lui qui a l'ensemble des informations nécessaires pour évaluer cette température de référence.

La réinterpolation des champs par la bibliothèque MpCCI s'est montrée relativement efficace mais présente tout de même quelques inconvénients, à la fois informatiques (difficultés à chaque changement de version et de machine), et numériques (les méthodes de transfert ne sont pas nécessairement conservatives). Nous envisageons donc une solution développée en interne à l'ONERA, basée sur la méthode des joints pour assurer un transfert optimal des quantités pariétales. La bibliothèque MpCCI répond aussi aux besoins de transfert d'information ; une solution devra donc être proposée pour traiter ce point.

Il nous faut pour finir reconnaître que la solution proposée pour le couplage à trois codes, traitant la mécanique à part, s'est certes montrée efficace pour le problème du tube. Mais cette solution ne sera pas utilisable dans toutes les situations de couplage multi-physiques. Les solutions asynchrones exposées n'ont malheureusement pas pu montrer leur adéquation à ce problème. Elle reste à mon avis la solution qu'il faudra approfondir, en rendant disponible cette option dans le code fluide. Le développement de critères de convergence adaptés à cette situation particulière sera aussi un des points clefs de cette méthode.

Annexes

Annexe A

Notations

Conventions typographiques

Dans l'ensemble du manuscrit, les vecteurs sont notés en gras : \mathbf{v} par exemple ; les tenseurs d'ordre 2 sont soulignés d'un ondulé : $\underline{\underline{\mathbb{I}}}$ désigne par exemple le tenseur identité ; les tenseurs d'ordre 4 d'un double ondulé : $\underline{\underline{\underline{\underline{\mathbb{K}}}}}$.

Nous avons aussi essayé autant que possible de noter en minuscule les vecteurs continus (les déplacements u par exemple), et en majuscule leur équivalent discrétisé dans la base d'éléments finis U .

Notations communes

t	temps	s
\mathbf{x}	position	m
\mathbf{n}	vecteur normal	
Ω	domaine de calcul	
Σ	interface fluide-structure	
div	divergence	
grad	gradient	
$\boldsymbol{\xi}^T$	transposée du vecteur $\boldsymbol{\xi}$	
\otimes	produit tensoriel	

Mécanique des fluides

ρ	densité	$kg.m^{-3}$
p	pression	Pa
\mathbf{v}	vitesse	$m.s^{-1}$
\mathbf{g}	accélération de la pesanteur	$m.s^{-2}$
$\boldsymbol{\sigma}_F$	tenseur des contraintes dans le fluide	Pa
$\boldsymbol{\tau}$	tenseur des contraintes visqueuses	Pa
μ	viscosité moléculaire	$Pa.s$
r	constante de Mariotte	$Pa.m^3.kg^{-1}.K^{-1}$

Mécanique et thermique du solide

u	déplacement	m
σ	tenseur des contraintes	Pa
ε	tenseur des déformations	
ε^e	déformations élastiques	
ε^p	déformations plastiques	
ε^{th}	déformations thermiques	
ν	coefficient de Poisson	
E	module d'Young	Pa
ρ	densité	$kg.m^{-3}$
T	température	K
ϕ	flux de chaleur	$W.m^{-2}$
κ	conductivité	$W.m^{-1}.K^{-1}$
C	chaleur spécifique	$J.kg^{-1}.K^{-1}$
h et α	coefficients d'échange physique et numérique	$W.m^{-2}.K^{-1}$

Annexe B

Inconel 600

L’Inconel 600 est l’alliage constituant le divergent de la tuyère des moteurs Vulcain. C’est un super-alliage à base nickel, incorporant aussi du chrome et du fer, ainsi que divers autres composants en faible quantité [66]. Il est généralement utilisé pour sa bonne résistance à la corrosion et aux haute température.

Dans le cas de la tuyère, cet alliage a été choisi pour son bon comportement à haute température. Son point de fusion est aux alentours de $1350\text{ }^{\circ}\text{C}$, soit 1650 K environ, ce qui en fait un des rares matériaux à pouvoir supporter les très hautes températures atteintes par une tuyère.

B.1 Comportement thermique

La courbe B.1 présente la variation du coefficient de dilatation thermique en fonction de la température.

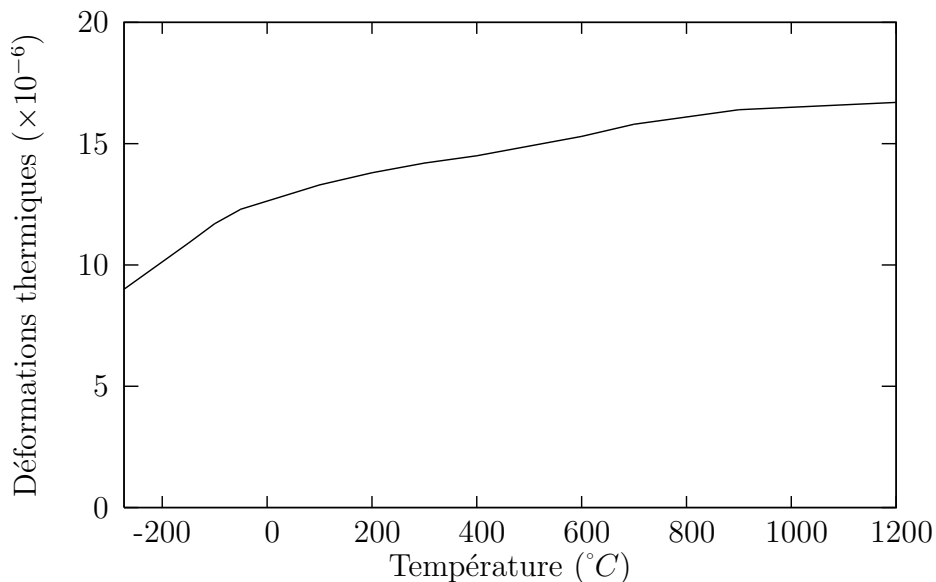


FIG. B.1 – Déformations thermiques en fonction de la température

Le tableau B.1 montre quant à lui la forte dépendance des autres paramètres inter-

venant dans le comportement thermique de l’Inconel 600 : conductivité et chaleur spécifique. Toutes ces valeurs sont tirées de données publiées en ligne par la société commercialisant le matériau [66].

Température [°C]	Conductivité [W.m ⁻¹ .°C ⁻¹]	Chaleur spécifique [J.kg ⁻¹ .°C ⁻¹]
-150	12,5	310
-100	13,1	352
-50	13,6	394
20	14,9	444
100	15,9	465
200	17,3	486
300	19,0	502
400	20,5	519
500	22,1	536
600	23,9	578
700	25,7	595
800	27,5	611
900	–	628

TAB. B.1 – Conductivité et chaleur spécifique de l’Inconel 600 ; valeurs issues de [66].

B.2 Comportement mécanique

Ce même document fournit aussi la dépendance du module d’Young en fonction de la température ; celle-ci est présentée sur le graphique B.2.

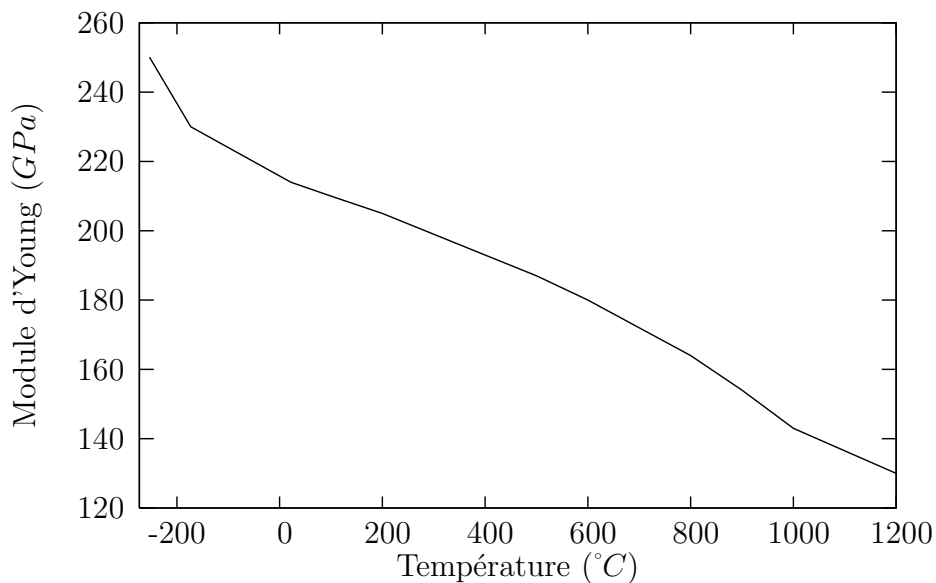


FIG. B.2 – Module d’Young en fonction de la température

Les autres paramètres intervenant dans l’équation 1.10 ont été identifiés au cours de l’é-

tude [63], par identification sur essais à différentes températures. Nous avons repris dans cette thèse cette loi de comportement. Pour des raisons de confidentialité industrielle, nous ne pouvons pas les reproduire ici. Il faut cependant garder à l'esprit que l'ensemble des paramètres (écrouissage, paramètres de viscosité) sont très fortement dépendants de la température.

Annexe C

Précisions logicielles

C.1 Brève description des diagrammes UML

Le logiciel ZéBuLoN est développé en C++, langage de programmation objet.

L'UML (Unified Modeling Language) est un langage de modélisation de données et de traitement. C'est une formalisation de la modélisation objet utilisée en génie logiciel¹.

Dans cette thèse, on en utilise les *diagrammes de classes* afin de représenter de façon claire et concise les différents objets développés, ainsi que les liens existant entre eux. Les conventions nécessaires à la compréhension de ces diagrammes sont expliquées au travers de l'exemple de la figure C.1.

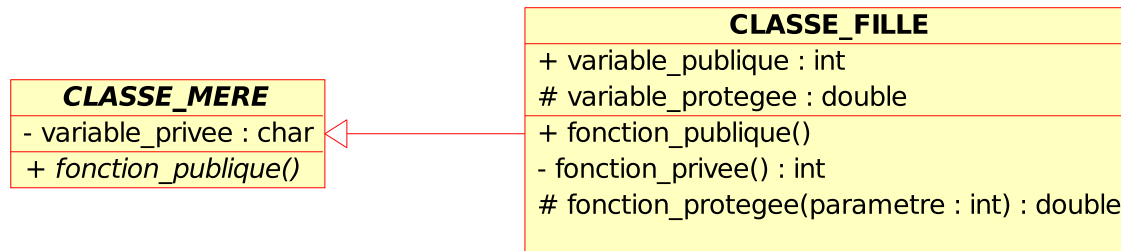


FIG. C.1 – Exemple de diagramme UML

Chaque boîte représente une *classe*. Une flèche reliant deux boîtes représente la *dérivation* d'un objet en un autre, ou, prise d'en l'autre sens, la relation héritier → donateur, l'objet "parent" étant pointé par la flèche. Outre le nom de la classe, la boîte contient tout d'abord l'ensemble des *attributs* (autrement dit les variables), puis l'ensemble des *méthodes* (ou fonctions) de la classe. Les signes '+', '-' et '#' devant le nom de la variable ou de la méthode représentent sa *visibilité* : publique, privée et protégée respectivement.

Les méthodes en italique représentent des méthodes *virtuelles pures*, c'est le cas dans l'exemple de la méthode `fonction_publice` de la classe mère. Toute classe qui contient une méthode virtuelle pure est une *classe virtuelle* (c'est-à-dire ne pouvant être instanciée). Elle sera elle aussi mise en italique.

Les attributs et méthodes statiques sont soulignées.

Le type et le nom des paramètres des méthodes sont donnés dans l'ordre inverse du C++ :

¹Définition extraite de [68]

d’abord le nom du paramètre, puis son type. Dans le cas particulier de paramètres dont on spécifie le type mais pas le nom, il n’apparaîtra que `:type`. Les pointeurs et les références sont notées comme en C++, par une étoile et une esperluette respectivement. Les *modèles* (*templates* en anglais) sont eux aussi notés comme en C++, entre les signes `<` et `>`.

C.2 Architecture du *plugin* de couplage ZéBuLoN

La méthode usuelle et surtout la plus pratique pour enrichir le logiciel ZéBuLoN de nouvelles fonctionnalités est à l’aide de *plugins*. Ainsi il n’est pratiquement pas nécessaire de modifier le cœur du logiciel, pour lui ajouter comme ici des capacités de couplage.

Les points de branchements se font au niveau d’objets particuliers, principalement les “composants du problème” (`PROBLEM_COMPONENT`) et les conditions au bord (`BC`). Le premier type d’objet permet de modifier et d’enrichir l’algorithme de résolution du calcul, en donnant accès à différents endroits stratégiques de l’algorithme de résolution : début/fin du problème, début/fin des incréments, etc . . .

Les objets dérivés de la classe `BC` permettent de définir les conditions de couplage à l’interface fluide-structure.

Un “noyau de couplage” a été identifié à partir des réflexions du chapitre 3. Ce noyau contient donc un certain nombre d’objets de base, réalisant les besoins qui avaient été exprimés dans ce précédent chapitre :

- le “composant de problème couplé” (classe `COUPLED_PROBLEM_COMPONENT`), dont dériveront les différents algorithmes que l’on envisagera ;
- les transferts de données (classe `EXTERNAL_DATA_TRANSFER`) ;
- les conditions au bord couplées, ainsi que les paramètres couplés (la différence entre les deux étant que les conditions au bord agissent directement sur les inconnues du problème, alors que les paramètres ne sont pas des inconnues du problème considéré² ;
- les objets permettant de réaliser le système du “postier” ;
- le “moteur de couplage”.

Autour de ce noyau gravite le reste des développements liés au couplage : cela comprend

- l’interface avec la librairie MpCCI ;
- les méthodes spécifiques au couplage mécanique ;
- les méthodes spécifiques au couplage thermique.

Nous avons en plus isolé l’ensemble des algorithmes développés spécifiquement pour une application dans un répertoire particulier. Ceux-ci pourront soit servir tel quel pour de nouvelles applications, soit servir d’exemple de base pour en dériver de nouveaux.

C.2.1 Classe “composant de problème couplé”

La classe `COUPLED_PROBLEM_COMPONENT` identifie et donne accès aux différents points à modifier l’algorithme usuel de calcul. Par exemple, dans la phase d’initialisation, le composant va se charger d’initialiser les paramètres de couplage : détection des zones couplées, déclaration de leur géométrie, initialisation d’éventuels critères de convergence, etc . . .

Selon l’algorithme choisi, le début d’incrément sera le bon endroit pour réaliser les envois ou les réceptions de quantités couplées.

²Ce sont par contre généralement les inconnues du problème *auquel* on est couplé.

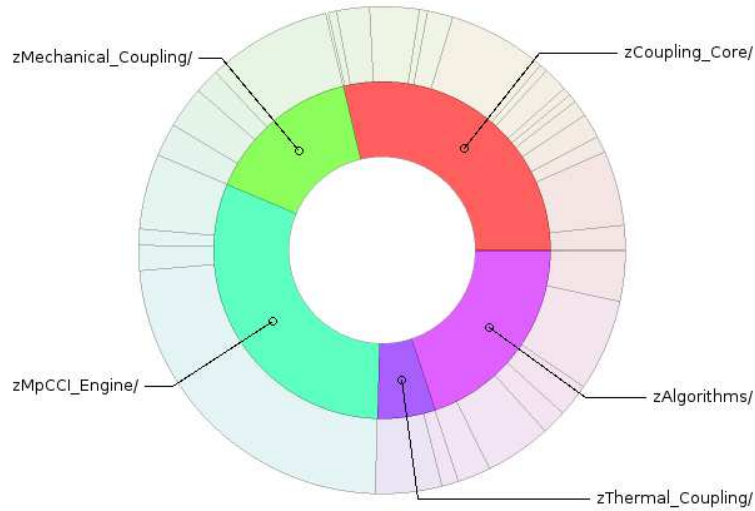


FIG. C.2 – Structure du plugin de couplage

En fin d'incrément, on réalisera aussi certains échanges de quantités ; c'est aussi le bon endroit pour vérifier les critères de convergence et d'échanger cette information avec les autres codes.

La classe mère `COUPLED_PROBLEM_COMPONENT` propose déjà un certain nombre de routines pour les échanges d'information³ afin que l'implantation d'un algorithme ne soit plus qu'une simple question de mise dans l'ordre de ces différentes routines de base.

Ils seront présentés dans la section [C.2.4](#).

C.2.2 Classes “conditions au bord et paramètres couplés”

Les conditions au bord couplées et les paramètres couplés dérivent tous les deux d'une classe mère `COUPLED_OBJECT`, qui regroupe certains comportements communs à tous les objets se servant de données couplées. Il s'agit :

- d'une méthode qui leur permet de se faire connaître du *composant de problème couplé* ;
- d'un accesseur à ce problème couplé ;
- de la méthode d'accès aux quantités couplées (par exemple “extrapolation”, ou “le plus récent”) ;
- d'une valeur par défaut du champ couplé au début du calcul.

Une fois ces objets détectés par le *composant de problème couplé*, ce dernier se charge de réaliser les réceptions de données suivant la manière définie dans l'algorithme de couplage.

Il reste ensuite à la charge des objets spécifiques d'utiliser les champs reçus pour les transformer en conditions au bord ou en paramètres. Ceci est un développement qui peut se faire de manière indépendante de tout couplage, et correspond souvent à la réutilisation de code déjà écrit dans d'autres situations. On illustrera cette situation dans la section suivante ([C.2.3](#)).

³C'est-à-dire échanges de quantités ou d'états de convergence.

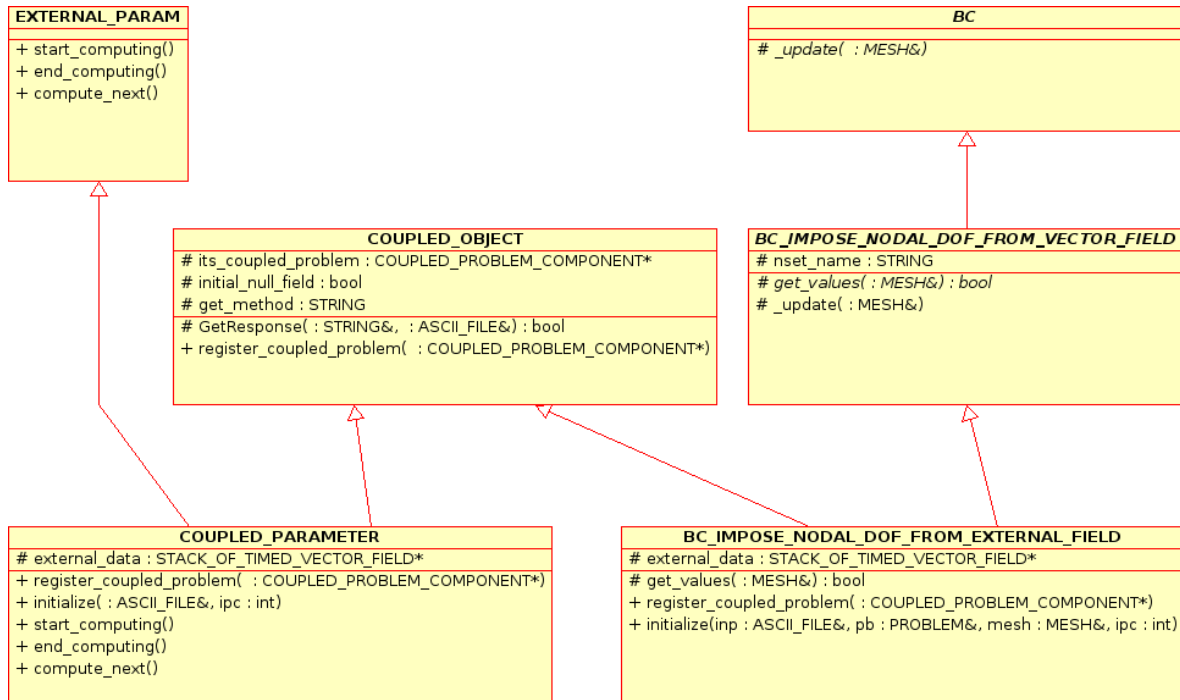


FIG. C.3 – Exemple d’objets *conditions au bord* et *paramètres* couplés. On peut y voir la classe commune à ces objets couplés. Par souci de clarté, seules les méthodes et variables importantes des objets sont présentées.

C.2.3 Factorisation des classes imposant la valeur de degrés de liberté

La programmation objet permet l’héritage, et donc la factorisation de certaines méthodes dans des classes mères génériques. Ce principe a été utilisé dans le développement d’une famille de classes de type *conditions au bord* servant à imposer les valeurs de certains degrés de liberté. En fait, toutes ces classes partagent un même noyau, qui est d’effectuer la recherche des nœuds auxquels on doit imposer ces valeurs, d’en extraire les bons degrés de liberté et d’imposer la valeur en fonction des instants auxquels on connaît et on veut imposer ces valeurs. Par contre, suivant la situation, ces valeurs proviendront de différentes sources :

- spécifiées par une fonction (de l’espace et/ou du temps) ;
- issues d’un fichier (qui aura été créé par un autre logiciel, ou issu d’un calcul préliminaire) ;
- reçues par une bibliothèque d’échanges de données, dans le cas d’un calcul couplé ;
- spécifié de manière dérivée, par un terme de pression variable.

Ce dernier exemple est un peu particulier, puisqu’il correspond à l’application de réactions nodales, plutôt que de valeurs nodales.

On a ainsi développé une classe mère abstraite, implantant les différentes méthodes communes aux classes filles. Il ne reste finalement aux classes filles que de surcharger la méthode de récupération des données. Suivant la situation, ces données seront :

- calculées à partir de la fonction spécifiée par l’utilisateur ;
- lues dans le fichier ;
- reçues du moteur de couplage ;

- calculées par intégration de la pression sur les éléments au travers des fonctions de forme.

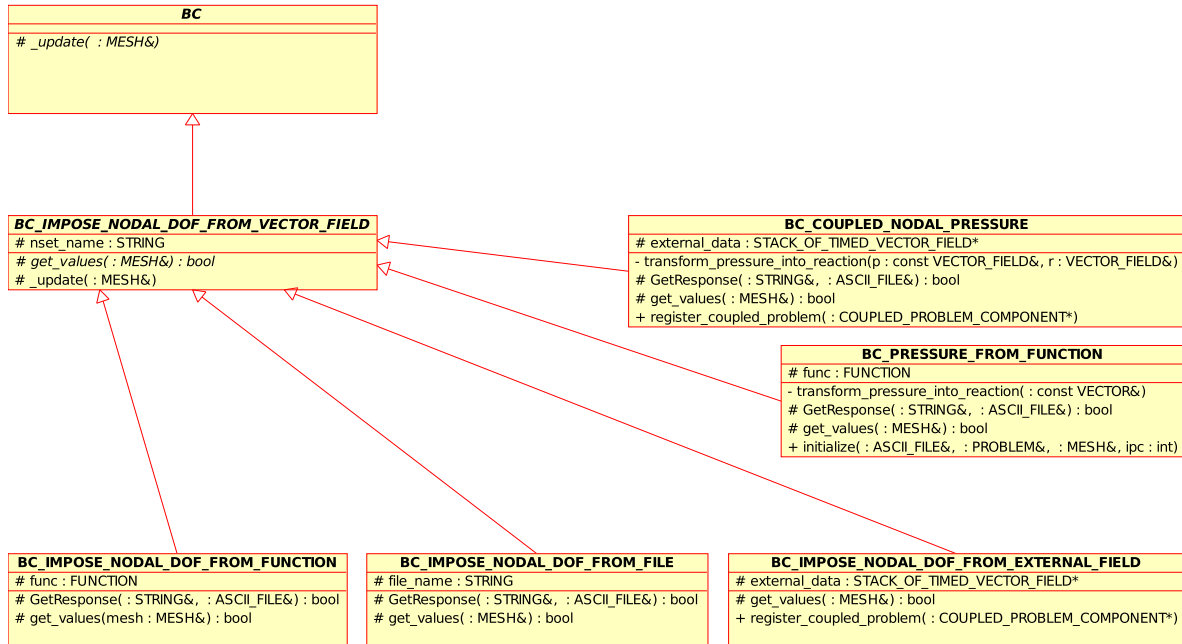


FIG. C.4 – Présentation de différents objets permettant d'imposer les degrés de liberté ou les réactions associées (à partir de fonctions, fichiers, ou de calculs couplés). Cette famille montre les possibilités d'héritage de la programmation objet. Notons qu'il aurait été possible de réaliser une classe intermédiaire regroupant les deux conditions aux bords imposant les pressions.

C.2.4 Algorithmes de couplage développés

Un certain nombre d'algorithmes de couplage ont été testés et implantés. Plutôt que de développer une unique classe paramétrable qui puisse être utilisée dans toutes les situations⁴, il a été décidé de développer une classe de base, puis d'en dériver différentes sous-classes adaptées à chaque algorithme que l'on désire utiliser.

La classe mère, déjà présentée ci-dessus, se nomme `COUPLED_PROBLEM_COMPONENT`, et propose des versions par défaut des routines d'envoi et de réception, ainsi que de la gestion de messages (système du postier). Ainsi, les algorithmes de couplage de type Dirichlet→Neumann tels que proposés par S. Piperno dans [57] nécessitent simplement le choix des fréquences de couplage, ainsi que leur placement dans la boucle de résolution (au choix : au début ou à la fin des incréments).

D'autres algorithmes, tels que ceux cherchant à conserver l'asynchronisme des codes sont plus spécifiques, et redéfinissent donc ces politiques d'envoi/réception.

La figure C.5 en présente quelques uns, disponibles dans le répertoire `MpCCI-3/zAlgorithms`.

⁴ Cette classe "tout-terrain" paramétrable a pourtant été le choix initial de développement. Elle s'est révélée complexe à maintenir dans toutes les situations. Nous nous sommes rapidement retrouvé devant des mises en données absurdes, ne reflétant plus le choix d'algorithme, mais plutôt cherchant à "plier" l'algorithme existant aux différentes situations.

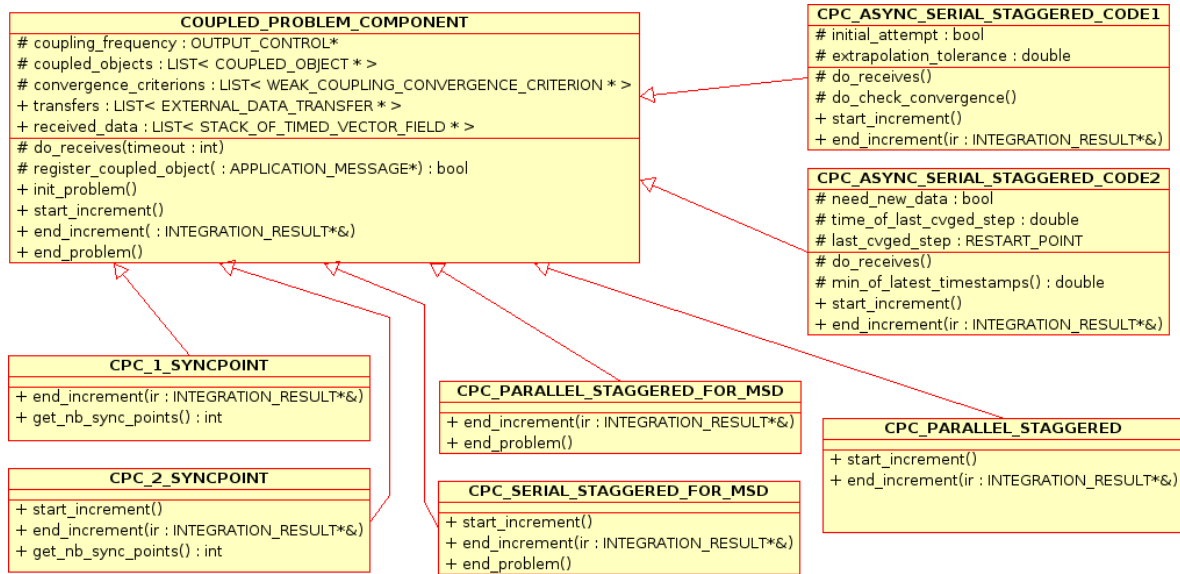


FIG. C.5 – Présentation de quelques algorithmes de couplage développés

Il est toutefois important de garder à l'esprit que l'algorithme de couplage n'est pas uniquement déterminé par l'objet `COUPLED_PROBLEM_COMPONENT`, mais aussi par le choix des conditions aux bords ou des paramètres couplés, des critères de correction et de vérification de convergence, ainsi que des fréquences de couplage.

Annexe D

Mises en données

D.1 Mises en données mécanique découplée pour ZéBuLoN

Ce fichier de mise en données correspond au calcul présenté dans la section 6.4. Il s'agit d'un calcul mécanique découplé, sous pression constante.

```
****calcul mechanical
***mesh classic_updated_lagrangian
**file tube20cm.geof

***resolution newton
**sequence
*time      1. 6. 600.
*increment 10
*iteration 10
*ratio 1.e-1
*algorithm p1p2p3

**automatic_time global 5
*divergence      2.0 10
*security        1.5
**init_d_dof 1.0 sequence

***parameter
**function temperature % (degres Celsius)
*function 21. + (t/6.) * (t<=6.) * (1100. - 300. * y / 4.e-3)
          + (t>6.) * (1100. - 300. * y / 4.e-3) ;
*tables CONSTANTE
*node

***bc
**pressure          % (Pa)
  Interieur -30.e+5 plateau
```

```
**impose_nodal_dof
  Froid-ext          U2 0.
  point_fixe_entree U1 0.
  point_fixe_sortie U1 0.
  point_fixe_centre U3 0.

***equation
**mpc1 face_entree U3
**mpc1 face_sortie U3

***table
**name plateau
*time 0. 1. 600.
*value 0. 1. 1.
**name CONSTANTE
*time 0. 600.
*value 1. 1.

***material
*file inco600_SI.mat

***output
**save_parameter
**output_first
***output ensight_gold
**save_parameter
**output_first

****return
```

D.2 Mises en données mécanique couplée pour ZéBuLoN

Le passage de la mise en données précédente à une mise en données couplée se fait en intervenant à deux endroits. Le premier est l'ajout d'un bloc décrivant le composant de problème couplé. Celui-ci précise le moteur de couplage, les envois et les critères de convergence :

```
***cpc_customizable
**coupling_engine MpCCI_engine

**recv_timeout -1
**maximum_iterations 20

**frequency
*dtime 0.1

**do_initial_send false
**do_initial_receive false
```

```
**external_transfer position
*nset Interieur *coord x
**external_transfer position
*nset Interieur *coord y
**external_transfer position
*nset Interieur *coord z
```

Le deuxième est l'ajout de la condition au bord couplée en pression :

```
***bc
**coupled_pressure_at_node
*nset Interieur
```

D.3 Mises en données thermique couplée ZéBuLoN

Ce fichier de mise en données correspond au calcul couplé thermique, avec condition de raccord mixte, présenté dans la section 8.5. On notera aussi le composant de problème couplé dédié spécifiquement à cette simulation (`cpc_thermo_nozzle`) et la présence d'un pas de temps de couplage automatique.

```
****calcul thermal_transient
***mesh
**file tube3D.geof

***resolution
**sequence
*time      1.  6.  10. 100.  600.
*iteration 10
*ratio     0.1
*algorithm p1p2p3
**automatic_time multiple global 10
*first_dtime 0.1 0.01 0.1 0.1 10.
*max_dtime   0.1 1.   1.  10. 100.
*divergence  2.0 10
*security    1.5

***cpc_thermo_nozzle
**coupling_engine MpCCI_engine
**maximum_iterations 20

**automatic_time std_coupled global 10
*first_dtime 0.1 0.01 0.1 0.1 10.
*max_dtime   0.1 1.   1.  10. 100.
*divergence  2.0 10
*security    1.5
```



```
**convergence_criterion Delta_flux
*nset Interieur
*ratio absolu 1.e-3

**external_transfer Temperature % vers MSD
*nset Interieur

***bc
**convection_heat_flux Chaud-ext
  h 4800.
  Te 2731.5 plateau_chaud

**coupled_convection_heat_flux
*nset Interieur
*Tref 50.
*initial value 0.
*get_method extrapolation_0

***table
**name plateau_chaud
*time 0. 1. 6. 600.
*value 0.1 0.1 1. 1.

***init_dof_value TP uniform 273.15

***material
*file inco600_thermal_SI.mat

***output Z7
**output_first
***output ensight_gold
**output_first

****return
```

Bibliographie

Bibliographie

- [1] A. ALTMIKUS, S. WAGNER, T. HABLOWETZ, AND K. WELL, *On the accuracy of modular aeroelastic methods applied to fixed and rotary wings*, in 18th AIAA Applied Aerodynamics Conference, AIAA 2000-4224, Aug. 2000.
- [2] ARIANESPACE, *Communiqué de presse : Vol 157, résultats de la commission d'enquête*, Jan. 2003. http://www.arianespace.com/site/fr/actualite/p07_01_03.htm.
- [3] F. P. T. BAAIJENS, *A fictitious domain/mortar element method for fluid-structure interaction*, International Journal for Numerical Methods in Fluids, 35 (2001), pp. 743–761.
- [4] S. BADIA, F. NOBILE, AND C. VERGARA, *Fluid-structure partitioned procedures based on Robin transmission conditions*, Tech. Rep. MOX-Report 11/2007, Politecnico di Milano, 2007.
- [5] T. BELYTSCHKO, W. K. LIU, AND B. MORAN, *Nonlinear finite elements for continua and structures*, Wiley, 2000.
- [6] C. BERNARDI, Y. MADAY, AND A. T. PATERA, *A new non conforming approach to domain decomposition : The mortar element method*, in Nonlinear partial differential equations and their applications, H. Brezis and J.-L. Lions, eds., Pitman, 1994.
- [7] J. BESSON, G. CAILLETAUD, J.-L. CHABOCHE, AND S. FOREST, *Mécanique non linéaire des matériaux*, Hermès, 2001.
- [8] J. BESSON AND R. FOERCH, *Large scale object-oriented finite element code design*, Computer Methods in Applied Mechanics and Engineering, (1997), pp. 165–187.
- [9] J. BESSON, R. LE RICHE, R. FOERCH, AND G. CAILLETAUD, *Application of object-oriented programming techniques to the finite element method – Part II : Application to material behaviors*, Revue Européenne des Éléments Finis, 7 (1998), pp. 567–588.
- [10] S. BRUNSSSEN AND B. WOHLMUTH, *An elastoplastic coupling algorithm for the simulation of incremental metal forming processes*, in Oñate et al. [52], pp. 133–136.
- [11] S. CANDEL, *Mécanique des fluides*, Dunod, 2 ed., 2001.
- [12] P. CAUSIN, J.-F. GERBEAU, AND F. NOBILE, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 4506–4527.
- [13] S. CHEMIN, *Étude des interactions thermiques fluide-structure par un couplage de codes de calcul*, PhD thesis, Université de Reims / ONERA, 2006.
- [14] P. G. CIARLET, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Dunod, 1982.
- [15] *Computer Methods in Applied Mechanics and Engineering Special issue on Fluid-Structure Interaction*, vol. 190, Mar. 2001.

- [16] A. DE BOER, A. VAN ZUIJLEN, AND H. BIJL, *Mesh movement based on radial basis function interpolation*, in Oñate et al. [52], pp. 669–672.
- [17] D. DUTOYA, *Modélisation de la turbulence au voisinage des parois : étude numérique des performances des différents modèles.*, Tech. Rep. RT 48/3473 EY, ONERA, 1992.
- [18] A. ERN AND J.-L. GUERMOND, *Éléments finis : théorie, applications, mise en œuvre*, Springer, 2002.
- [19] M. ERRERA, G. CHAINERAY, F.-X. ROUX, AND J.-D. GARAUD, *Déformation de maillage pour le couplage fluide/structure dans les propulseurs à propergol solide*, Tech. Rep. RT 1/12189 DSNA, ONERA, Dec. 2006.
- [20] ———, *Travaux de mise en place du maillage mobile dans le code CEDRE*, Tech. Rep. RT 4/10813 DSNA, ONERA, Nov. 2006.
- [21] C. FARHAT, P. GEUZAINÉ, AND C. GRANDMONT, *The discrete geometric conservation law and its effects on nonlinear stability and accuracy*, Journal of Computational Physics, 174 (2001), pp. 669–694.
- [22] C. FARHAT, M. LESOINNE, AND P. LE TALLEC, *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*, Computer Methods in Applied Mechanics and Engineering, 157 (1998), pp. 95–114.
- [23] C. FARHAT AND F.-X. ROUX, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, International Journal for Numerical Methods in Engineering, 32 (1991), pp. 1205–1227.
- [24] C. FARHAT, K. G. VAN DER ZEE, AND P. GEUZAINÉ, *Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1973–2001.
- [25] C. A. FELIPPA, K. C. PARK, AND C. FARHAT, *Partitioned analysis of coupled mechanical systems*, in Computer Methods in Applied Mechanics and Engineering [15], pp. 3247–3270.
- [26] M. A. FERNÁNDEZ, J.-F. GERBEAU, AND C. GRANDMONT, *A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid*, Rapport de recherche RR-5700, INRIA, 2005. Submitted to International Journal for Numerical Methods in Engineering.
- [27] S. FOREST, M. AMESTOY, S. CANTOURNET, G. DAMAMME, S. KRUCH, AND V. MAUREL, *Mécanique des milieux continus*. ENSMP, 2007. http://mms2.ensmp.fr/mmc_paris/poly/MMC.pdf.
- [28] B. FRAEJIS DE VEUBEKE, *The dynamics of flexible bodies*, International Journal of Engineering Science, 14 (1976).
- [29] D. FUNARO, A. QUARTERONI, AND P. ZANOLLI, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM Journal on Numerical Analysis, 6 (1988), pp. 1213–1236.
- [30] E. GAMMA, R. HELM, R. JOHNSON, AND J. M. VLISSIDES, *Design Patterns : Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [31] J.-D. GARAUD, *Une utilisation du couplage faible dans ZéBuLoN*, La GaZeTTe, 21 (2005). <http://www.mat.ensmp.fr/Produits/LaGaZeTTe/index.html>.

- [32] J.-D. GARAUD, A. ROOS, F. FEYEL, F.-X. ROUX, M. ERRERA, AND Y. MADAY, *Coupled simulation of a nozzle cooling system*, in Oñate et al. [52], pp. 421–424.
- [33] J.-F. GERBEAU AND M. VIDRASCU, *A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows*, Tech. Rep. RR-4691, INRIA, 2003.
- [34] J.-F. GERBEAU, M. VIDRASCU, AND P. FREY, *Fluid-structure interaction in blood flows on geometries based on medical imaging*, *Computers & Structures*, 83 (2005), pp. 155–165.
- [35] J. M. GERE AND S. TIMOSHENKO, *Mechanics of Materials*, D. Van Nostrand Company, 1972.
- [36] P. GIRODROUX LAVIGNE, *Progress in steady/unsteady fluid-structure coupling with Navier-Stokes equations*, in International Forum on Aeroelasticity and Structural Dynamics, Munich, Germany, 28 June – 1 July 2005. Disponible sur <http://publications.onera.fr/exl-doc/342923.pdf>.
- [37] M. GLÜCK, M. BREUER, F. DURST, A. HALFMANN, AND E. RANK, *Computation of fluid-structure interaction on lightweight structures*, *Journal of Wind Engineering and Industrial Applications*, 89 (2001), pp. 1351–1368.
- [38] C. GRANDMONT, *Analyse mathématique et numérique de quelques problèmes d'interaction fluide-structure*, PhD thesis, Université Paris 6, 1998.
- [39] C. GRANDMONT, V. GUIMET, AND Y. MADAY, *Numerical analysis of some decoupling techniques for the approximation of the unsteady fluid structure interaction*, *Mathematical Models and Methods in Applied Sciences*, 11 (2001), pp. 1349–1377.
- [40] C. GRANDMONT AND Y. MADAY, *Nonconforming grids for the simulation of fluid-structure interaction*, *Contemporary Mathematics*, 218 (1998).
- [41] C. GRANDMONT, Y. MADAY, AND P. MÉTIER, *Modeling and analysis of an elastic problem with large displacements and small strains*, *Journal of Elasticity*, 87 (2007), pp. 29–72.
- [42] P. HUERRE, *MMC3 : Mécanique des fluides*, Éditions de l'École Polytechnique, 1992.
- [43] P. LE TALLEC AND J. MOURO, *Fluid structure interaction with large structural displacements*, in *Computer Methods in Applied Mechanics and Engineering* [15], pp. 3039–3067.
- [44] J. LEMAITRE AND J.-L. CHABOCHE, *Mécanique des matériaux solides*, Dunod, 2004.
- [45] L. D. MARINI AND A. QUARTERONI, *A relaxation procedure for domain decomposition methods using finite elements*, *Numer. Math.*, 55 (1989), pp. 575–598.
- [46] H. G. MATTHIES AND J. STEINDORF, *Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction*, *Computers & Structures*, 80 (2002), pp. 1991–1999.
- [47] D. P. MOK, W. A. WALL, AND E. RAMM, *Accelerated iterative substructuring schemes for instationary fluid structure interaction*, in *Proceedings of the First MIT Conference on Computational Fluid and Solid Mechanics*, 2001.
- [48] J. MOURO, *Interaction fluide structure en grands déplacements. Résolution numérique et application aux composants hydrauliques automobiles*, PhD thesis, École polytechnique - Peugeot, 1996.
- [49] MPCCI, *Mesh-based Parallel Code Coupling Interface*. <http://www.scai.fraunhofer.de/mpcci.html>.
- [50] *MSD 3.1, notice d'utilisation*, 2003. MSDH/NU 001 - 4.0.

- [51] F. NOBILE, *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*, PhD thesis, École polytechnique fédérale de Lausanne, 2001.
- [52] E. OÑATE, M. PAPADRAKAKIS, AND B. SCHREFLER, eds., *Computational Methods for Coupled Problems in Science and Engineering II – Coupled problems 2007*, May 2007.
- [53] PALM, *Projet d'assimilation par logiciel multi-méthodes*. <http://www.cerfacs.fr/~palm>.
- [54] *Parallel visualization application*. <http://www.paraview.org/>.
- [55] K. C. PARK AND C. A. FELIPPA, *Partitioned analysis of coupled systems*, in *Computational methods for transient analysis*, T. Belytschko and H. T. J. R., eds., North-Holland Pub. Co., 1983, pp. 157–219.
- [56] K. C. PARK, C. A. FELIPPA, AND J. A. DERUNTZ, *Stabilization of staggered solution procedures for fluid-structure interaction analysis*, in *Computational methods for fluid-structure interaction problems*, T. Belytschko and T. Geers, eds., vol. 26 of AMD, American society of mechanical engineers, 1977, pp. 95–124.
- [57] S. PIPERNO, *Simulation numérique de phénomènes d'interaction fluide-structure*, PhD thesis, ENPC, France, 1995.
- [58] S. PIPERNO, *Explicit/implicit fluid-structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulation*, *International Journal for Numerical Methods in Fluids*, 25 (1997), pp. 1207–1226.
- [59] S. PIPERNO AND C. FARHAT, *Design of efficient partitioned procedures for the transient solution of coupled aeroelastic*, *Revue Européenne des Éléments Finis*, 9 (2000), pp. 655–680.
- [60] —, *Partitioned procedures for the transient solution of coupled aeroelastic problems – part 2 : energy transfer and 3-dimensional applications*, in *Computer Methods in Applied Mechanics and Engineering* [15], pp. 3147–3170.
- [61] S. PIPERNO, C. FARHAT, AND B. LARROUTUROU, *Partitioned procedures for the transient solution of coupled aeroelastic problems – part 1*, *Computer Methods in Applied Mechanics and Engineering*, 124 (1995), pp. 79–112.
- [62] A. QUARTERONI, *Domain decomposition method for the numerical solution of partial differential equations*, Tech. Rep. UMSI90/246, University of Minnesota, 1990.
- [63] A. ROOS AND J.-L. CHABOCHE, *Vulcain II cooling tubes – final report*, Tech. Rep. RT 2/08500, ONERA, 2005.
- [64] J. SALENÇON, *Mécanique des milieux continus, Tomes I à III*, Éditions de l'École Polytechnique, 2002.
- [65] L. SERIES, F. FEYEL, AND F.-X. ROUX, *Une méthode de décomposition de domaine à deux multiplicateurs de Lagrange*, tech. rep., ONERA, 2003. 6^e congrès national en calcul de structures. Disponible sur <http://publications.onera.fr/exl-doc/333140.pdf>.
- [66] SPECIAL METALS, *Properties of Inconel 600 alloy*. <http://www.specialmetals.com/products/inconelalloy600.php>.
- [67] *Vtk, the visualization toolkit*. <http://public.kitware.com/VTK/>.
- [68] WIKIPEDIA, *UML*. http://fr.wikipedia.org/wiki/Unified_Modeling_Language.
- [69] *Z-SeT, finite element solver and material constitutive model library*. <http://www.nwnumerics.com/>.

-
- [70] *Z-Set*, user manual. <http://www.nwnumerics.com/Downloads/>.
- [71] O. C. ZIENKIEWICZ AND R. L. TAYLOR, *The finite element method : fluid dynamics*, vol. 3, Butterworth-Heinemann, 5th ed., 2000.
- [72] S. ZURBACH AND L. BALLESTER, *Cryogenic flows in rocket engines*, 2005. Snecma, SAFRAN Group, Vernon, France. <http://www.fluent.com/about/news/newsletters/05v14i3/pdfs/s4.pdf>.

Jean-Didier GARAUD

DÉVELOPPEMENT DE MÉTHODES DE COUPLAGE AÉRO-THERMO-MÉCANIQUE POUR LA PRÉDICTION D'INSTABILITÉS DANS LES STRUCTURES AÉROSPATIALES CHAUDES

Résumé :

Dans cette thèse, on étudie numériquement, par la méthode partitionnée, trois types de couplage issus du domaine aérospatial : l'aéro-mécanique, l'aéro-thermique et l'aéro-thermo-mécanique. Un moteur de couplage est développé pour gérer les aspects logistiques. Outre l'indépendance spatiale et temporelle des différents codes, il permet de mettre en place rapidement un algorithme de couplage taillé sur mesure pour chaque application.

L'étude d'une tuyère du moteur Vulcain 2, refroidie par écoulement de gaz, sert de fil conducteur applicatif. Modélisée à haute température par un comportement non-linéaire élasto-visco-plastique, la mécanique couplée est résolue par un algorithme simple. Au contraire, la thermique se montre problématique, et nécessite l'utilisation conjointe de deux méthodes, originales dans le contexte du couplage : un pas de temps automatique de couplage, et des conditions de raccord mixtes. Ces deux cas sont finalement assemblés pour résoudre le problème complètement couplé, à trois codes.

Mots-clés : interaction fluide-structure, algorithmes de couplage partitionné, aérothermique, aéromécanique, couplage à 3 codes, tuyère Vulcain 2.

COUPLED AERO-THERMO-MECHANICAL ALGORITHMS FOR NON-LINEAR SIMULATIONS OF CRITICALLY HOT AEROSPACE STRUCTURES

Abstract :

A partitioned approach is used to study numerically three types of coupled problems arising in aerospace engineering : aero-mechanics, aero-thermics and aero-thermo-mechanics. A coupling engine is specifically designed to handle data exchange and algorithmic control signals between structural mechanics and fluid mechanics codes. Each code runs with its own space and time discretisation, and the engine provides a toolbox to quickly develop coupling algorithms tailored to a given application.

The hydrogen-cooled nozzle extension of the Vulcain 2 rocket engine serves as a test case. The mechanical coupling is relatively straightforward due to the high contrast between stiffness of the fluid and the elasto-viscoplastic solid. On the other hand, the thermal coupling requires the joint use of two methods, innovative in this context : an automatic coupled time-stepping, and Robin boundary conditions at the coupled interfaces. Both mechanical and thermal coupling methods are finally assembled to address the complete coupled problem involving three codes.

Keywords : fluid-structure interaction, partitioned coupling algorithms, aerothermics, aeromechanics, 3-codes coupling, Vulcain 2 nozzle extension.