



HAL
open science

Prédiction par essais virtuels de l'amortissement dans les structures spatiales

Alain Caignot

► **To cite this version:**

Alain Caignot. Prédiction par essais virtuels de l'amortissement dans les structures spatiales. Mécanique [physics.med-ph]. École normale supérieure de Cachan - ENS Cachan, 2009. Français. NNT : . tel-00422291

HAL Id: tel-00422291

<https://theses.hal.science/tel-00422291>

Submitted on 6 Oct 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.



ENSC-2009 n° 159

**THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN**

Présentée par
ALAIN CAIGNOT

pour obtenir le grade de
DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Domaine :
MÉCANIQUE - GÉNIE MÉCANIQUE - GÉNIE CIVIL

Sujet de la thèse :
**Prédiction par essais virtuels de l'amortissement
dans les structures spatiales**

Thèse soutenue le 16 juin 2009 à l'ENS de Cachan devant le Jury composé de
MM. :

CHRISTOPHE BARDET	EADS-Bordeaux	Examineur
DAVID DUREISSEIX	LMGC Montpellier	Rapporteur
LUC GONIDOU	CNES Évry	Examineur
MOHAMED ICHCHOU	École Centrale de Lyon	Rapporteur
PIERRE LADEVÈZE	ENS de Cachan	Directeur de thèse
VINCENT LE GALLO	ASTRIUM-ST	Examineur
DAVID NÉRON	ENS de Cachan	Co-encadrant de thèse
ROGER OHAYON	CNAM	Examineur

LMT-Cachan
ENS Cachan/CNRS/UMPC/PRES UniverSud Paris
61, avenue du Président Wilson, F-94235 CACHAN CEDEX (France)

Je tiens à remercier tout particulièrement l'ensemble des membres du jury pour avoir participé à ma soutenance et particulièrement MM. Dureisseix et Ichchou pour avoir accepté d'être rapporteur de mon travail.

Je remercie Pierre Ladevèze pour la qualité du sujet qu'il m'a proposé et David Néron pour toutes les relectures, parfois douloureuses, de mes différents travaux. Je remercie également ASTRIUM-ST, le CNES et la DGA pour l'intérêt et le soutien apporté à ce travail.

Je remercie également très chaleureusement Laurent Champaney qui m'a permis de démarrer ce travail en passant des heures à dépouiller divers résultats de calcul. Une mention toute particulière à mes amis David Violeau, avec qui nous avons passé de très très longs moments à développer ce logiciel et Hugo Leclerc qui est à l'origine de tous ces développements.

Ces longues années m'ont permis de cotoyer de nombreuses personnes notamment au centre de calcul. Merci à toi Frisou pour avoir toujours su mettre de l'ambiance et nous remonter le moral pendant les périodes difficiles. Merci à tous mes camarades, amis dont je ne saurais citer tous les noms : Ludo, Julien, Nico, Manu, Gilles, Sébastien, Jérémie...

Enfin mes derniers mots iront à Delphine et mes enfants Charlotte et Benjamin pour l'infini patience dont ils ont fait preuve durant cette très longue dernière ligne droite.

Table des matières

Table des matières	i
Introduction	1
1 Amortissement dans les structures spatiales	7
1 Présentation du support industriel	7
2 Sources de dissipation dans les lanceurs	8
3 Modélisations de l'amortissement	9
3.1 Rappels de vibrations non amorties	9
3.1.1 Modes propres normaux de vibration	9
3.1.2 Orthogonalité des modes propres	10
3.1.3 Développement modal d'un vecteur	10
3.1.4 Réponse à un chargement extérieur	10
3.2 Vibration amortie	11
3.2.1 Problème à résoudre	11
3.2.2 Modélisation de l'amortissement	11
4 Détermination des coefficients d'amortissement modaux	13
5 Approche expérimentale	14
5.1 Description du matériel	14
5.1.1 Éprouvettes	14
5.1.2 Montage expérimental	15
5.2 Principaux résultats	17
5.2.1 Éprouvette composite	17
5.2.2 Liaisons collées	19
5.2.3 Liaison boulonnée SSS	19
5.3 Limitations	21
2 Outils disponibles pour concevoir une stratégie d'essais virtuels	23
1 Problème type à résoudre	23
2 Outils numériques	25

2.1	Logiciels	25
2.2	Algorithmes de résolution d'un problème de contact	26
2.2.1	Comparaison entre Cofast3d et Abaqus avec ses options par défaut	27
2.2.2	Comparaison des algorithmes Abaqus	29
2.3	Calcul d'une liaison avec un boulon	31
2.4	Calcul du problème type	32
2.4.1	Comparaison entre Cofast3d et Abaqus avec ses options par défaut	33
2.4.2	Comparaison entre Cofast3d et Abaqus avec le meilleur coefficient de pénalisation	34
3	Conclusions - limitations	36
3	Méthode LATIN multiéchelle en espace	37
1	Présentation de la méthode LATIN dans sa version monoéchelle	37
1.1	Décomposition de domaine - Problème à résoudre	38
1.2	Comportements d'interface particuliers	40
1.2.1	Interface parfaite	40
1.2.2	Interface de type contact avec frottement	40
1.3	Stratégie de résolution itérative	41
1.3.1	Étape locale à l'itération $n+1/2$	42
1.3.2	Étape linéaire à l'itération $n+1$	42
1.3.3	Contrôle des itérations	43
2	Ajout des aspects multiéchelles en espace	44
2.1	Introduction des aspects multiéchelles	44
2.1.1	Séparation des échelles	44
2.1.2	Admissibilité des quantités macro	46
2.2	Modification de l'étape linéaire	46
2.2.1	Définition du comportement homogénéisé	48
2.2.2	Algorithmes de résolution	49
3	Calcul des énergies imposée et dissipée	51
4	Implantation numérique	51
5	Premiers exemples simples	52
5.1	Contact entre deux poutres	52
5.2	Retour sur la résolution du problème type	53
4	Prédiction de l'amortissement dans les liaisons	57
1	Vérification de l'approche : simulation de l'essai de flexion de la liaison SSS	57
1.1	Modélisation de la liaison SSS	57
1.2	Vérification du calcul	59

1.2.1	Détermination du coefficient d'amortissement . . .	59
1.2.2	Influence du niveau du critère d'arrêt	60
1.2.3	Influence du numéro du cycle sur le calcul de l'amortissement	61
1.2.4	Influence de la discrétisation temporelle	62
1.2.5	Influence de la discrétisation spatiale	62
2	Validation de la méthode par comparaison avec les essais expérimentaux	64
2.1	Coefficient de frottement	64
2.2	Identification du coefficient de frottement	65
3	Résultats et influence des paramètres physiques du modèle	68
3.1	Carte de glissement et de dissipation	68
3.2	Influence des précharges dans les boulons	69
4	Autres sollicitations	69
5	Calcul d'autres liaisons types des lanceurs spatiaux	72
5.1	Bride à 90°	72
5.2	Bride à 45°	74
5	Parallélisation de la méthode	79
1	Parallélisation de la méthode	79
1.1	Motivations	79
1.2	Implantation dans le logiciel	80
1.2.1	Mise en données	81
1.2.2	Algorithme de résolution LATIN	83
1.2.3	Visualisation des résultats	84
1.3	Gains obtenus grâce aux parallélisme	84
1.3.1	Extensibilité	84
1.3.2	Speed-up	87
1.4	Conclusions	88
2	Un exemple de calcul	90
	Conclusion	95
	Bibliographie	97
A	Notice d'utilisation du logiciel JointVirt	105
1	Construction du maillage	105
2	Fabrication du fichier de données	106
2.1	Section mesh	106
2.2	Section materials	107
2.3	Section proprietes_interfaces	108
2.4	Section CL	109
2.5	Section parametres	111

2.6	Section parametres_temporels	112
2.7	Section parametres_affichage	112
2.8	Section direction_recherche	113
3	Post-traitement des résultats	113
4	Interactivité	114
5	Utilisation du logiciel	115
5.1	Description de l'archive	115
5.2	Utilisation standard	116
5.3	Utilisation en parallèle	116
5.4	Remarques	117

Introduction

La modélisation de la réponse dynamique des structures est une des questions majeures liées au dimensionnement et à la conception. La prédiction des modes propres par des moyens numériques ou expérimentaux est aujourd'hui bien maîtrisée, notamment en basses fréquences, mais la caractérisation des niveaux reste une question ouverte car elle dépend de la connaissance de l'amortissement. La prédiction de celui-ci est donc une problématique industrielle importante, en particulier dans les domaines de pointe comme l'industrie spatiale ou aéronautique.

En effet, l'amélioration des technologies employées pour la construction des structures se caractérise par une augmentation de la raideur et une diminution de la masse des matériaux ou des liaisons. Ces changements s'accompagnent dans la plupart des cas d'une diminution de l'amortissement, ce qui augmente les niveaux de réponse de la structure et doit donc être parfaitement maîtrisé. C'est pourquoi, dans le cadre d'une vraie démarche prédictive de dimensionnement, la connaissance *a priori* de l'amortissement de chaque élément constitutif des sous-structures (matériaux, liaisons...) est indispensable.

Actuellement, l'impossibilité pour l'ingénieur de connaître l'amortissement dès la conception d'une structure implique de devoir réaliser des campagnes d'essais longues et coûteuses afin de pouvoir l'identifier *a posteriori*. L'information obtenue est alors globale par sous-structure et par mode et l'influence du niveau d'excitation est rarement connue, alors que l'on sait que l'amortissement peut varier fortement en fonction de ce niveau ([Ungar 1964 - 1973, Ibrahim et Pettit 2005] pour les dissipations issues de contact avec frottement).

L'objectif industriel de cette étude est donc la prédiction de l'amortissement dans les structures, dès leur phase de conception en fonction du type de sollicitation, aussi facilement que cela est réalisé pour les coefficients d'élasticité.

Le travail présenté ici se restreint à l'étude de structures faiblement amorties et vibrant à basse fréquence, avec une application directe aux lanceurs spatiaux. Il fait l'objet d'un programme « Recherche et Technologie Lanceurs » en collabo-

ration avec le CNES Evry et ASTRIUM-ST Les Mureaux et a comme support le lanceur Ariane 5. L'ensemble des résultats présentés ici est applicable aux liaisons de tous les types de lanceurs civils ou militaires, qui possèdent globalement les mêmes caractéristiques. La méthode proposée pour prédire l'amortissement structural dans le lanceur passe par différentes étapes. La première est d'effectuer un premier calcul modal, sans amortissement, permettant de déterminer les modes et les niveaux auxquels la structure doit répondre. Ensuite, l'amortissement correspondant est déterminé en sommant la dissipation des différentes parties de la structure à l'aide d'une base de données contenant la valeur de l'amortissement des liaisons en fonction de tous les paramètres retenus (coefficient de frottement, serrage des boulons, type de chargement, niveau de chargement...). Enfin, le calcul prédictif final peut être réalisé avec les valeurs de l'amortissement précédemment déterminées.

Dans le lanceur, les principales sources de dissipation recensées sont les matériaux et les liaisons entre les sous-structures et entre les sous-structures et le milieu extérieur (écoulement des fluides). La caractérisation de l'amortissement dans les matériaux a fait l'objet de nombreux travaux : [Zener 1948] pour les métalliques, [Nashif *et al.* 1985] pour les polymères, [Adams et Maheri 1994, Maheri et Adams 1994] pour les composites stratifiés à l'échelle du pli, [Saravanos et Chamis 1990, Chandra *et al.* 2002 - 2003] et [Adams et Maheri 1993, Maheri et Adams 1994] pour les âmes composites nid d'abeille en aluminium utilisées dans les structures spatiales, [Galucio *et al.* 2007, Deü *et al.* 2008] pour des matériaux de type composite avec peaux en matériau piezoélectrique. L'amortissement des matériaux, qu'ils soient métalliques, polymères, composites, est donc assez bien connu pour les matériaux qui nous intéressent et sa valeur très faible. [Gaul et Nitsche 2001, Ibrahim et Pettit 2005] ont ainsi montré que la dissipation est principalement localisée dans les liaisons et notamment dans celles avec frottement comme les liaisons boulonnées ou rivetées.

Ce travail fait suite à des travaux [Leloch 2003, Maisonneuve 2004, Caignot *et al.* 2005], dans lesquels un premier pas dans la création de cette base de données d'amortissement passait par une approche expérimentale. Les techniques expérimentales avaient déjà montré leur efficacité pour les matériaux [Adams et Maheri 1993, Maheri et Adams 1995, Pourroy *et al.* 1998, Esteban et Rogers 1999, Ichchou *et al.* 2008], mais la mesure dans les liaisons restait encore une question largement ouverte. Depuis, elle a fait l'objet de plusieurs travaux, parmi lesquels on peut citer [Oldfield *et al.* 2005, Jalali *et al.* 2007, Zang *et al.* 2008]. L'identification de l'amortissement dans les liaisons nécessite en effet de se placer dans des gammes proches de la réalité en terme de niveau de déformation, fréquence et déformées modales, mais aussi que les résultats ne soient pas perturbés par le moyen d'essai. Les travaux expérimentaux menés ont conduit à réaliser un montage d'essai original permettant de mesurer l'amortissement dans les matériaux et les liaisons pour des niveaux proches de ceux auxquels est soumis la structure en conditions réelles,

pour une sollicitation de type flexion. L'amortissement dans les matériaux avait ainsi pu être quantifié expérimentalement et comparé à des résultats de référence (numériques ou expérimentaux), sa valeur restant très faible, environ 0,2%. Un certain nombre de liaisons types avait également été testé, dont des liaisons collées et des liaisons boulonnées. Les principaux résultats étaient les suivants : les liaisons collées avec ou sans renfort n'apportent aucun amortissement supplémentaire alors que les liaisons boulonnées apportent de l'amortissement de manière significative. L'amortissement dans les liaisons boulonnées testées est non linéaire en fonction du niveau d'excitation, variant de 0,2 à plus de 10%, et il est indépendant de la fréquence. Cependant, le problème majeur de cette approche expérimentale est qu'elle nécessite la mise au point d'un essai pour chaque type de sollicitation et pour chaque liaison, ce qui conduit à des études très coûteuses.

L'objectif de ce travail est donc d'aller encore plus loin en proposant une méthode pour prédire l'amortissement numériquement et de manière systématique. Le calcul numérique de l'amortissement passera par la prise en compte dans un calcul de type éléments finis des sources de dissipation. On parlera de « Virtual Testing » .

La modélisation des mécanismes dissipatifs dans les liaisons est encore une question ouverte. Toutefois, la principale source d'amortissement dans les liaisons de type boulonnées ou rivetées est le frottement entre les différentes pièces en contact [Ibrahim et Pettit 2005]. Les liaisons sont dimensionnées de façon à ne pas glisser à une échelle dite « macro », c'est-à-dire pour qu'il n'y ait pas de glissement complet entre les surfaces de contact. Cependant, du fait des déformations, il y a du « micro-glissement » (à l'échelle de la rugosité) et du « méso-glissement » (échelle intermédiaire entre le micro et le macro) [Gropper et Hemmye 1983a -b]. La modélisation de ces phénomènes peut aller des lois de frottement de Coulomb à des lois beaucoup plus complexes comme les modèles de Dahl [Dahl 1976] ou de LuGre [Barabanov et Ortega 2000, Hensen 2002], voire même à une description atomistique du phénomène [Müser 2002, Hirano 2006]. Bien entendu, les modèles atomistiques ne sont pas envisageables pour réaliser des calculs à l'échelle d'une structure complète. Les modèles de Dahl ou de LuGre ont également un nombre de paramètres élevé, mais leur implantation numérique aurait pu être envisagée dans la méthode de calcul retenue. **Cependant, les déplacements constatés dans les exemples étudiés sont cependant suffisamment grands par rapport à la rugosité des surfaces en contact pour considérer que le contact peut être modélisé par les lois de Coulomb. Ainsi le calcul de l'amortissement ne dépend que de la connaissance du coefficient de frottement et de son éventuelle évolution au cours des cycles.**

L'amortissement doit être calculé pour les sollicitations basses fréquences, et étant données les longueurs d'onde mises en jeu, il est légitime d'effectuer le calcul en quasi-statique. La principale difficulté du calcul à réaliser est de calculer des quantités locales (forces tangentes et vitesses de glissement sur les zones de contact) très précisément afin obtenir une approximation correcte de la dissipation. De plus, la dissipation est une quantité souvent très faible, ce qui nécessite des techniques de calcul où la pollution numérique peut être contrôlée. Le calcul numérique des problèmes de contacts à grand nombre de degrés de liberté est un problème encore assez ouvert. En effet, un certain nombre de logiciels industriels, tels Abaqus, permettent d'effectuer des simulations multi-contacts complexes, mais souvent avec un niveau de précision insuffisant pour effectuer le calcul de la dissipation. En effet dans Abaqus, pour obtenir un résultat satisfaisant, les différents algorithmes proposés conduisent soit à un résultat imprécis sur la dissipation, soit à des coûts prohibitifs pour générer la solution. C'est pourquoi un certain nombre de travaux dans lesquels on simule précisément par éléments finis les liaisons ou dans lesquels on met en place un modèle d'interface pour simuler les liaisons ont été proposés [Mackerle 2003, Crocombe *et al.* 2006, Wentzel et Olsson 2008, Miller et Quinn 2009].

Dans ce travail, la solution retenue pour effectuer le calcul de la dissipation est l'utilisation d'un algorithme de calcul non incrémental : la méthode LATIN multiéchelle [Ladevèze 1999]. Cette méthode permet en effet de construire à chaque itération une solution complète sur l'espace et le temps, qui augmente en qualité au cours des itérations. Il suffit alors de choisir un critère d'arrêt, basé dans notre cas sur la qualité du calcul de la dissipation, pour stopper l'algorithme et avoir une solution de qualité donnée. Ainsi, cette méthode permet de converger vers la solution recherchée, sans que le calcul se bloque comme c'est le cas parfois pour des méthodes traditionnelles. **Un des apports de ce travail a été d'adapter la méthode au problème en mettant notamment en place les critères d'arrêts, ainsi que la méthode de calcul de la dissipation sur les interfaces dans un code de calcul de recherche développé au LMT-Cachan.** De plus, la méthode LATIN, associée à une technique d'homogénéisation en espace pour prendre en compte les aspects multiéchelles, est particulièrement bien adaptée pour les problèmes sous-structurés.

Le calcul correct de la dissipation nécessite des maillages très fins au niveau des zones de contact, ce qui implique de devoir traiter des problèmes avec de très nombreux degrés de liberté (de 0,5 à 1 million de degrés de liberté pour des liaisons simples voire beaucoup plus). Ces problèmes peuvent bien entendu être résolus sur un seul processeur, la limitation se situant au niveau de la taille de la mémoire et au temps que l'on souhaite accorder à la génération de la solution. Jusqu'il y a deux ans, l'évolution des processeurs était purement basée sur des améliorations de leur architecture et sur l'augmentation du nombre d'opérations réalisables à la seconde,

aussi les codes de calcul standards profitaient directement des évolutions en terme de rapidité. Cependant, l'évolution actuelle est tout autre. En effet, les fabricants de processeurs se sont tournés vers une nouvelle stratégie qui consiste à augmenter le nombre d'unités physiques sur un même processeur. Il y a actuellement jusqu'à 8 cœurs par processeur mais d'ici 2012, les fabricants annoncent jusqu'à 16 cœurs et plus encore par la suite. Même si les compilateurs actuels et futurs permettent de paralléliser « automatiquement » quelques opérations très élémentaires, force est de constater que les codes de calcul ne pourront plus bénéficier des améliorations des processeurs sans passer par la parallélisation de la méthode elle-même.

La méthode LATIN associée à la décomposition mixte est parallélisable [Cognard *et al.* 1996, Champaney *et al.* 1997] et les aspects multiéchelles de la méthode la rendent extensible [Ladevèze et Dureisseix 2000, Ladevèze *et al.* 2001]. **Un nouvel apport de ce travail a été d'adapter le code de calcul recherche développé initialement par H. Leclerc et D. Violeau [Ladevèze *et al.* 2006] pour résoudre le problème posé et le paralléliser en utilisant les bibliothèques MPI.** L'utilisation du standard MPI plutôt qu'une vision *multithreading* plus classique permet au code de pouvoir travailler indifféremment sur des architectures à mémoire partagée (machines SMP) ou sur des clusters de PC à mémoire distribuée. Le travail effectué sur la parallélisation a pris soin de minimiser au maximum les besoins en mémoire ainsi que les redondances mémoires entre les différents processeurs afin de maximiser les performances.

Ce mémoire est composé de cinq chapitres :

- le chapitre 1 rappelle les différentes modélisations de l'amortissement utilisées dans l'analyse dynamique des structures ainsi que les différents résultats expérimentaux publiés dans [Leloch 2003, Maisonneuve 2004, Caignot *et al.* 2005]. Ces résultats serviront de référence pour le calcul numérique des liaisons ;
- le chapitre 2 présente le problème à résoudre ainsi que les différentes stratégies actuelles permettant de le résoudre. Il montre notamment les limitations des outils actuels et la nécessité du développement d'un outil dédié ;
- le chapitre 3 présente la stratégie de calcul retenue : la méthode LATIN dans sa version multiéchelle pour la résolution de problème de contact avec frottement en quasi-statique ;
- le chapitre 4 présente les résultats obtenus à l'aide du nouvel outil de calcul sur la liaison de référence ainsi que sur d'autres liaisons types du lanceur Ariane 5. Il est l'occasion de vérifier puis de valider les calculs qui sont menés ;
- le chapitre 5 présente la parallélisation de l'outil et les gains qui y sont associés.

Amortissement dans les structures spatiales

Dans ce chapitre, on présente le support de l'étude et on détaille les sources de dissipation dans les lanceurs. Puis, on rappelle les différentes modélisations de l'amortissement utilisées dans le calcul de structures. On rappelle également les principaux développements et résultats expérimentaux [Leloch 2003, Maisonneuve 2004, Caignot et al. 2005], notamment ceux sur les liaisons boulonnées qui serviront de référence pour la comparaison avec l'approche numérique.

1 Présentation du support industriel

Cette étude a été réalisée en collaboration avec le CNES-Evry et ASTRIUM-ST Les Mureaux afin d'augmenter la connaissance sur l'amortissement des différents éléments constitutifs des lanceurs utilisés. Le support de l'étude est le lanceur européen Ariane 5 (Figure 1.1) muni de son SYstème de Lancement Double (SYLDA5 Figure 1.1) qui lui permet de mettre en orbite deux satellites lors d'un seul et même lancement. Malgré ce système, les coûts d'un lancement restent très élevés et les industriels cherchent à réduire les marges de sécurité prises lors des dimensionnements pour diminuer les coûts drastiquement. Pour cela, la connaissance de l'amortissement est primordiale, car il conditionne les efforts maximaux que la structure va devoir supporter aux fréquences de résonance.

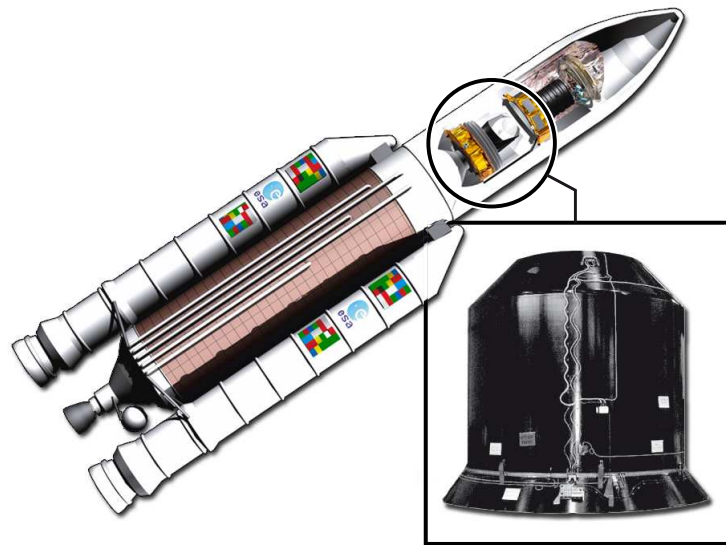


Figure 1.1: Lanceur Ariane 5 et SYLDA5 (©EADS)

Le SYLDA5 est un assemblage de cônes et de cylindres réalisé avec des panneaux en composite sandwich (peaux en composite stratifié carbone/epoxy et âme en nid d'abeille d'aluminium). Les liaisons entre les différents panneaux composites sont variées. Ce sont essentiellement des liaisons collées avec éventuellement des zones de renfort, mais on trouve aussi la liaison du système de séparation, qui est une liaison boulonnée (liaison SSS). Elle contient le cordon pyrotechnique qui permet de séparer le SYLDA5 du lanceur avant le lancement de la charge utile basse.

Afin de caractériser l'amortissement du SYLDA5, une vaste campagne d'essais expérimentaux a été menée en 1998 sur la structure réelle. L'ensemble du dispositif mesurait 10 mètres de haut pour un diamètre de 5 mètres. Il était instrumenté avec environ 200 capteurs et 5 excitateurs. Ces essais ont notamment permis de vérifier que les modes propres calculés étaient corrects et de recalibrer les coefficients d'amortissement modal.

2 Sources de dissipation dans les lanceurs

Le cahier des charges fonctionnel de l'étude en cours [Le Gallo 2005] recense de manière exhaustive les sources de dissipation présentes dans le lanceur.

On a de la dissipation dans les matériaux due à différents mécanismes comme le mouvement des dislocations ou la plastification dans les métalliques ou encore due à la micro-fissuration, décohésion fibre/matrice, délaminage local dans les matériaux

composites. Il y a aussi de la dissipation dans les matériaux polymères mais ceux-ci sont présents en très faible quantité dans le lanceur.

Pour ce qui est des liaisons, on retrouve dans les liaisons collées des mécanismes dissipatifs comme la plastification de la colle ou la rupture des interfaces collées. Dans les liaisons boulonnées, on a de la dissipation essentiellement due au frottement entre les différentes pièces en contact mais aussi due à la plastification des matériaux.

D'autres sources de dissipation peuvent enfin être citées même si elles ne font pas l'objet d'une étude ici : frottements dus à l'écoulement de l'air sur le lanceur, ainsi que l'écoulement des poudres dans les deux boosters.

3 Modélisations de l'amortissement

3.1 Rappels de vibrations non amorties

On rappelle dans ce paragraphe la démarche de résolution des problèmes de vibrations qui conduit à faire des choix sur la modélisation de l'amortissement comme cela est détaillé dans [Géradin et Rixen 1997].

3.1.1 Modes propres normaux de vibration

De manière générale, un problème de vibrations libres traité par éléments finis se ramène à :

$$\mathbf{M}\ddot{\underline{x}} + \mathbf{K}\underline{x} = \underline{0} \quad (1.1)$$

où \underline{x} est le vecteur du champ des déplacements discrétisés, \mathbf{M} est la matrice de masse et \mathbf{K} la matrice de raideur. On notera n le nombre de degrés de liberté.

On cherche la solution sous la forme $\underline{x} = \underline{\phi}g(t)$ où $\underline{\phi}$ est un vecteur de constantes constituant la forme propre du mouvement. Cela conduit à résoudre :

$$\ddot{g}(t)\mathbf{M}\underline{\phi} + g(t)\mathbf{K}\underline{\phi} = \underline{0} \quad (1.2)$$

Comme $\mathbf{K}\underline{\phi}$ et $\mathbf{M}\underline{\phi}$ sont non nuls (sauf pour $\underline{\phi} = 0$), on en déduit que $-\frac{\ddot{g}}{g}$ est une constante et l'on peut montrer qu'elle est strictement positive. Elle sera notée ω^2 .

On est amené à résoudre un problème aux valeurs propres $(\mathbf{K} - \omega^2\mathbf{M})\underline{\phi} = \underline{0}$ dont la solution sera composée des fréquences propres ω_i et des vecteurs propres $\underline{\phi}_i$.

3.1.2 Orthogonalité des modes propres

Les modes propres calculés précédemment sont \mathbf{K} -orthogonaux et \mathbf{M} -orthogonaux. C'est-à-dire que pour deux pulsations propres distinctes, on a :

$$\underline{\phi}_i^T \mathbf{K} \underline{\phi}_j = 0 \quad (1.3)$$

$$\underline{\phi}_i^T \mathbf{M} \underline{\phi}_j = 0 \quad (1.4)$$

et pour deux pulsations propres identiques, la forme quadratique est constante :

$$\underline{\phi}_i^T \mathbf{K} \underline{\phi}_i = k_i \quad (1.5)$$

$$\underline{\phi}_i^T \mathbf{M} \underline{\phi}_i = m_i \quad (1.6)$$

où k_i est appelée raideur généralisée et m_i masse généralisée du mode i .

Couramment, les modes propres sont normés par rapport à la masse, c'est-à-dire que :

$$\underline{\phi}_i^T \mathbf{M} \underline{\phi}_j = \delta_{ij} \quad (1.7)$$

$$\underline{\phi}_i^T \mathbf{K} \underline{\phi}_j = \omega_i^2 \delta_{ij} \quad (1.8)$$

où δ_{ij} est le symbole de Kronecker et ω_i^2 est la fréquence propre associée au mode i .

3.1.3 Développement modal d'un vecteur

La famille des modes propres $(\underline{\phi}_1, \dots, \underline{\phi}_n)$ forme une base de l'espace vectoriel associé à la représentation de la matrice de masse \mathbf{M} et de la matrice de raideur \mathbf{K} . Aussi, tout vecteur quelconque, comme celui des déplacements, peut être décomposé sur cette base :

$$\underline{x} = \sum_{i=1}^n q_i \underline{\phi}_i \quad (1.9)$$

3.1.4 Réponse à un chargement extérieur

On souhaite maintenant résoudre le problème suivant :

$$\mathbf{M}\ddot{\underline{x}} + \mathbf{K}\underline{x} = \underline{f} \quad (1.10)$$

où \underline{f} est le vecteur correspondant au chargement.

Dans un cadre éléments finis standard, on pourrait résoudre ce problème avec un algorithme explicite ou implicite en résolvant une succession de problèmes linéaires.

L'approche modale consiste à chercher la solution en projection sur la base modale du système calculée précédemment : $\underline{x} = \Phi \underline{q}$ avec Φ la matrice composée des modes propres $\Phi = (\underline{\phi}_1, \dots, \underline{\phi}_n)$ et $\underline{q} = (q_1, \dots, q_n)$. En injectant la solution sous cette forme dans l'équation (1.10) et en prémultipliant par Φ^T , on obtient le système suivant à résoudre :

$$\mathbf{m}\ddot{\underline{q}} + \mathbf{k}\dot{\underline{q}} = \Phi^T \underline{f} = \underline{z} \quad (1.11)$$

où $\mathbf{m} = \Phi^T \mathbf{M} \Phi$ est la matrice des masses généralisées et $\mathbf{k} = \Phi^T \mathbf{K} \Phi$ la matrice des raideurs généralisées. \mathbf{m} et \mathbf{k} sont des matrices diagonales du fait de l'orthogonalité des vecteurs propres, dont les coefficients diagonaux sont respectivement les (m_1, \dots, m_n) pour la matrice \mathbf{m} et les (k_1, \dots, k_n) pour la matrice \mathbf{k} .

Ce système est un système découplé de n équations différentielles du second ordre à coefficients constants pour la résolution duquel on peut utiliser les techniques classiques (résolution directe, transformée de Laplace, résolution numérique).

L'avantage de la résolution modale est de passer d'un système d'équations couplées à un système d'équations découplées qui est beaucoup moins coûteux à résoudre.

3.2 Vibration amortie

3.2.1 Problème à résoudre

On cherche maintenant à résoudre le problème des petites oscillations suivant :

$$\mathbf{M}\ddot{\underline{x}} + \mathbf{C}\dot{\underline{x}} + \mathbf{K}\underline{x} = \underline{f} \quad (1.12)$$

où $\mathbf{C}(\omega)$ est la matrice d'amortissement structural, qui peut dépendre de la fréquence.

Si on cherche la solution de ce problème avec une décomposition sur les modes propres du système conservatif associé, on obtient un système de n équations à résoudre :

$$\underbrace{\Phi^T \mathbf{M} \Phi}_{\mathbf{m}} \ddot{\underline{q}} + \underbrace{\Phi^T \mathbf{C} \Phi}_{\mathbf{c}(\omega)} \dot{\underline{q}} + \underbrace{\Phi^T \mathbf{K} \Phi}_{\mathbf{k}} \underline{q} = \Phi^T \underline{f} = \underline{z} \quad (1.13)$$

qui n'est pas découplé du fait du caractère non diagonal de \mathbf{c} .

L'approche modale perd son avantage principal qui consistait à aboutir à la résolution d'un système d'équations découplées. Cependant, quand on cherche la solution sur une base tronquée des modes propres, on doit résoudre un système couplée mais qui reste de taille raisonnable.

3.2.2 Modélisation de l'amortissement

Les techniques expérimentales d'identification de la dynamique des structures étant basées principalement sur de l'analyse modale, il est nécessaire d'effectuer des

hypothèses sur la répartition de l'amortissement dans la structure. Classiquement, on suppose que la matrice d'amortissement est diagonale dans la base des modes propres, ce qui n'a aucune justification physique mais ce qui permet de revenir à un système d'équations découplées. Voici les deux principales hypothèses, couramment utilisées et qui ne sont pas forcément vérifiées en même temps, permettant de rendre la matrice d'amortissement diagonale.

Hypothèse de Basile : on suppose que l'amortissement est très faible $\|\mathbf{C}\| \ll 1$ et que les modes sont éloignés en terme de fréquence. L'hypothèse de Basile consiste à considérer que la contribution en terme d'amortissement est déterminée uniquement par le terme diagonal de la matrice. Pour des modes couplés en revanche, la partie extradiagonale peut intervenir. Si on suppose n'avoir que des modes découplés, on peut considérer la matrice \mathbf{c} diagonale.

Hypothèse d'amortissement proportionnel : on la matrice d'amortissement comme étant proportionnelle aux matrices de masse et de raideur de la structure. C'est une vision globale de la dissipation où l'on suppose que chaque zone dissipe proportionnellement de la même manière. La matrice d'amortissement peut alors se mettre sous la forme :

$$\mathbf{C}(\omega) = \alpha(\omega)\mathbf{K} + \beta(\omega)\mathbf{M} \quad (1.14)$$

$$\mathbf{c}(\omega) = \alpha(\omega)\mathbf{k} + \beta(\omega)\mathbf{m} \quad (1.15)$$

$$c_i(\omega) = \alpha(\omega)k_i + \beta(\omega)m_i \quad \forall i \quad (1.16)$$

En injectant cette forme dans l'équation (1.13) et en recherchant \underline{q} sous forme harmonique, on obtient :

$$-\mathbf{m}\omega^2 \underline{q} + j\omega \mathbf{c}(\omega) \underline{q} + \mathbf{k} \underline{q} = \underline{f} \quad (1.17)$$

$$\Leftrightarrow m_i \omega^2 q_i + j\omega c_i(\omega) q_i + k_i q_i = \underline{f}_i \quad \forall i \quad (1.18)$$

$$\Leftrightarrow \omega^2 q_i + j\omega \frac{c_i(\omega)}{m_i \omega_i} \omega_i q_i + \omega_i^2 q_i = \frac{\underline{f}_i}{m_i} \quad \forall i \quad (1.19)$$

$$\Leftrightarrow \omega^2 q_i + 2j \frac{\omega}{\omega_i} \xi_i q_i + \omega_i^2 q_i = \frac{\underline{f}_i}{m_i} \quad \forall i \quad (1.20)$$

où $\xi_i = \frac{1}{2} \frac{c_i(\omega)}{m_i \omega_i}$ est le coefficient d'amortissement modal réduit du mode i .

Le choix des coefficients de proportionnalité $\alpha(\omega)$ et $\beta(\omega)$ conduit à différents types de modélisation. Les plus couramment utilisés sont le modèle d'amortissement visqueux et le modèle d'amortissement hystérétique.

a) Modèle visqueux Dans ce modèle, les coefficients $\alpha(\omega)$ et $\beta(\omega)$ sont supposés indépendants de la fréquence. On peut donc en déduire que $\underline{\phi}^T \mathbf{C} \underline{\phi}$, la matrice d'amortissement modale, est une matrice diagonale, donc les coefficients sont notés \mathbf{c}_i .

On en déduit donc que pour ce modèle, l'amortissement modal ξ_i vaut :

$$\xi_i = \frac{1}{2} \left(\alpha \omega_i + \frac{\beta}{\omega_i} \right) \quad (1.21)$$

Ce modèle est associé au modèle rhéologique de Kelvin-Voight comportant un ressort et un amortisseur visqueux en parallèle. L'avantage de ce modèle est de pouvoir résoudre analytiquement le problème à résoudre.

b) Modèle hystérétique Dans ce modèle, communément appelé « modèle à rigidité complexe », les coefficients α et β sont supposés dépendre de la fréquence par les relations $\alpha = \frac{\eta}{\omega}$ et $\beta = 0$. La proportionnalité est exprimée en fonction du facteur de perte η .

On a alors :

$$c_i = \frac{\eta}{\omega} k_i \quad (1.22)$$

$$\xi_i = \frac{1}{2} \eta \quad (1.23)$$

4 Détermination des coefficients d'amortissement modaux

N'ayant, le plus souvent, aucune connaissance sur la valeur de l'amortissement, les coefficients d'amortissement modaux ξ_i sont identifiés à partir de résultats expérimentaux. Naturellement, une fois la matrice d'amortissement modale identifiée, l'ensemble des calculs sur la structure qui seront réalisés, seront prédictifs. Le problème est que l'identification conduit à réaliser des études très longues et coûteuses, et **l'objectif de ce travail est de proposer une méthode de détermination prédictive des coefficients d'amortissement ne nécessitant plus d'essais expérimentaux sur la structure finale.**

La méthode proposée dans le cadre de cette étude est la suivante. Pour un mode i donné, on calcule les efforts et les moments dynamiques sur les différents éléments du maillage de la structure en faisant un calcul sans amortissement. À partir de ces torseurs d'efforts dynamiques locaux, on calcule la dissipation apportée sur ces éléments à partir d'une base de données de dissipation des matériaux et des liaisons. Pour cela, on somme la dissipation apportée par chaque type de sollicitations principales considérées (traction, cisaillement, moment fléchissant) et on obtient ainsi une énergie dissipée par élément de liaison notée $\Delta U_i^{\text{éléments}}$.

On peut ensuite calculer la dissipation totale de la structure, notée ΔU_i , en faisant la somme des contributions élémentaires :

$$\Delta U_i = \sum_{\text{éléments}} \Delta U_i^{\text{éléments}} \quad (1.24)$$

On en déduit ensuite l'amortissement modal ξ_i par le rapport de la dissipation avec l'énergie élastique du mode, notée U_i :

$$\xi_i = \frac{1}{4\pi} \frac{\Delta U_i}{U_i} \quad (1.25)$$

Le calcul des efforts dynamiques d'un mode donné sont en fait dépendants de l'amortissement du mode. Il est donc nécessaire d'effectuer un processus itératif afin d'obtenir un amortissement modal correct.

La véritable difficulté de cette méthode est l'élaboration de la base de données des dissipations des matériaux et des liaisons.

Notons que lorsqu'on parle d'amortissement, plusieurs définitions sont couramment utilisées :

- la capacité d'amortissement spécifique ou coefficient d'absorption : $\Psi = \frac{\Delta U}{U}$
où U est l'énergie élastique maximale sur le cycle et ΔU l'énergie dissipée sur le cycle ;
- le facteur de perte : $\eta = \frac{1}{2\pi} \Psi$;
- le coefficient d'amortissement modal : $\xi = \frac{1}{2} \Psi$.

5 Approche expérimentale

Une première méthode pour construire cette base de données est d'utiliser des techniques expérimentales. Cela a fait l'objet d'une première étude menée au LMT-Cachan, qui a permis de trouver la valeur des coefficients d'amortissement pour quelques éléments constitutifs du lanceur [Leloch 2003, Maisonneuve 2004, Caignot *et al.* 2005]. Les matériaux utilisés ont pour la plupart déjà fait l'objet d'études expérimentales ou analytiques dans [Adams et Maheri 1993 - 1994, Maheri et Adams 1994], en revanche pour ce qui est des liaisons, il n'y avait pas d'informations. C'est pourquoi un dispositif expérimental a été développé pour mesurer l'amortissement dans celles-ci. Il a été validé en mesurant l'amortissement dans le composite, qui est le matériau principalement utilisé dans le SYLDA5 et en vérifiant la corrélation avec les résultats de [Adams et Maheri 1993 - 1994, Maheri et Adams 1994].

5.1 Description du matériel

5.1.1 Éprouvettes

Afin de caractériser l'amortissement dans les différents éléments constitutifs du SYLDA5, EADS avait mis à la disposition du LMT-Cachan un certain nombre d'éprouvettes représentatives de chacun d'entre eux : des éprouvettes en matériaux

composites (peaux en composite stratifié carbone/époxy et âme en nid d'abeille d'aluminium (Figure 1.2(a)), des liaisons collées simples (Figure 1.2(b)) permettant de relier les différents panneaux de composites entre eux, des liaisons collées avec un renfort composite (Figure 1.2(c)) permettant de relier les panneaux composites entre la zone cylindrique et la zone conique du SYLDA5, et enfin, des liaisons boulonnées, dites SSS (Figure 1.2(d)) permettant le passage d'un cordon pyrotechnique nécessaire à la séparation des charges utiles avec le lanceur. Les principales dimensions des éprouvettes sont données sur la Figure 1.3.

5.1.2 Montage expérimental

La sollicitation choisie dans cette étude était la flexion pure, sollicitation prépondérante pour les basses fréquences. Le montage d'essai retenu, permettant de réaliser ces conditions, était un montage de flexion quatre points, la liaison étant située entre les deux noeuds centraux pour avoir une sollicitation de flexion pure (Figure 1.4). Afin de minimiser au maximum la dissipation du montage, les liaisons pivots avaient été spécialement étudiées pour ne pas dissiper. La solution technique retenue était l'utilisation de liaison souple avec un rétrécissement de matière (Figure 1.4), qui ne dissipait que très peu d'énergie comparée à des solutions standards avec des roulements ou des coussinets.

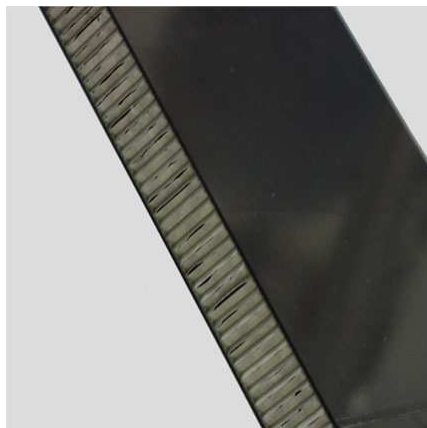
De plus, afin d'être prédictif, on souhaitait mesurer l'amortissement dans des conditions similaires à la réalité. Pour cela, il a été nécessaire de rajouter des masses afin de faire vibrer le montage dans la bonne gamme de fréquence et dans la bonne gamme de niveau d'effort.

Les principales difficultés d'une telle campagne d'essais étaient :

- de mesurer la dissipation de l'éprouvette et pas du montage en n'apportant pas de dissipation supplémentaire, notamment avec les conditions limites.
- de ne pas perturber les résultats par les moyens d'excitation ou de mesures.

Afin d'avoir une mesure fiable de l'amortissement dans les liaisons, un système de double pesée était utilisé : l'amortissement de la liaison était obtenu en retranchant l'amortissement obtenu pour l'éprouvette composite seul à l'amortissement brut obtenu avec les éprouvettes avec liaison, ainsi l'amortissement de liaison ne prenait en compte ni la dissipation du montage, ni la dissipation du composite. Enfin, l'ensemble du montage était placé sur une plateforme de vibration permettant de ne pas avoir de pollution due à l'environnement extérieur.

L'excitation choisie était un chargement de type lâché permettant de mesurer les vibrations libres de l'éprouvette. Le chargement était imposé à l'aide d'un câble d'acier sur le plateau supérieur (Figure 1.5) amené à rupture ce qui permettait d'obtenir une excitation sans introduire de choc. Différents diamètres de câble étaient utilisés, permettant d'injecter un niveau d'effort plus ou moins élevé. Le montage avait également été soumis à des vibrations forcées sous l'action d'une machine



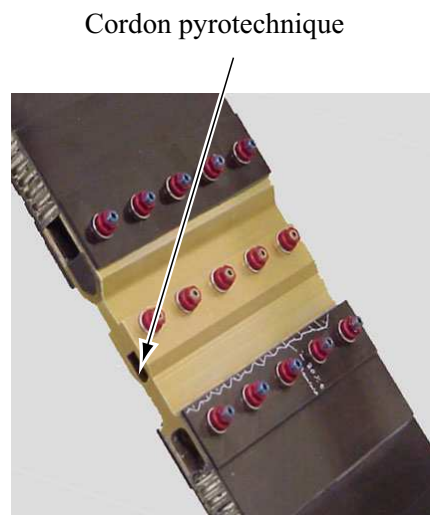
(a) Matériau composite



(b) Liaison collée



(c) Liaison collée avec renfort



Cordon pyrotechnique

(d) Liaison boulonnée SSS

Figure 1.2: Éprouvettes représentatives des éléments constitutifs du SYLDA5

de traction hydraulique afin d'effectuer un grand nombre de cycles à haut niveau de déformation et sous l'action d'un pot vibrant afin d'observer l'influence de la fréquence sur l'amortissement dans les liaisons étudiées.

Les mesures étaient réalisées à l'aide de jauge de déformation collées sur la peau de composite. Elles étaient placées dans une zone de déformation homogène et permettaient de mesurer la déformation de traction et l'angle de flexion global. L'ex-

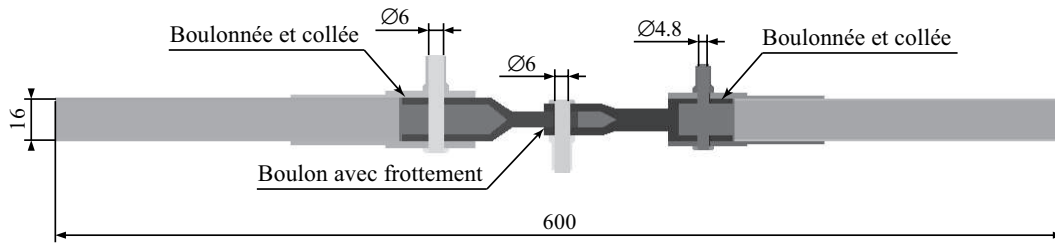


Figure 1.3: Dimensions principales des éprouvettes (ici avec la liaison SSS)

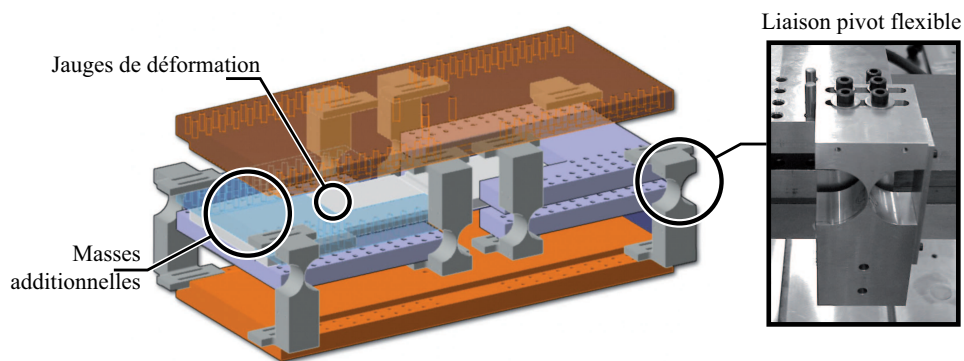


Figure 1.4: Montage d'essai avec l'éprouvette et les masses additionnelles en place

plottage des résultats temporels était réalisée à l'aide d'une méthode de décrémentation logarithmique.

Les mesures étaient de bonne qualité et reproductibles. La Figure 1.6 présente la déformation mesurée par les jauges de déformation en fonction du temps lors d'un essai de lâché. L'amortissement mesuré avait été comparé après plusieurs montages/démontages de l'éprouvette et pour différentes éprouvettes ; les résultats étaient tout à fait reproductibles.

5.2 Principaux résultats

5.2.1 Éprouvette composite

Un modèle théorique a été développé dans [Adams et Maheri 1994, Maheri et Adams 1995] pour déterminer l'amortissement dans les matériaux composites sandwich à partir des caractéristiques des constituants de base. D'après ce modèle, dans notre cas, l'amortissement devait être de 0,2%. L'amortissement mesuré à l'aide de l'essai expérimental est de 0,3%. Ce test de référence a permis de montrer que la

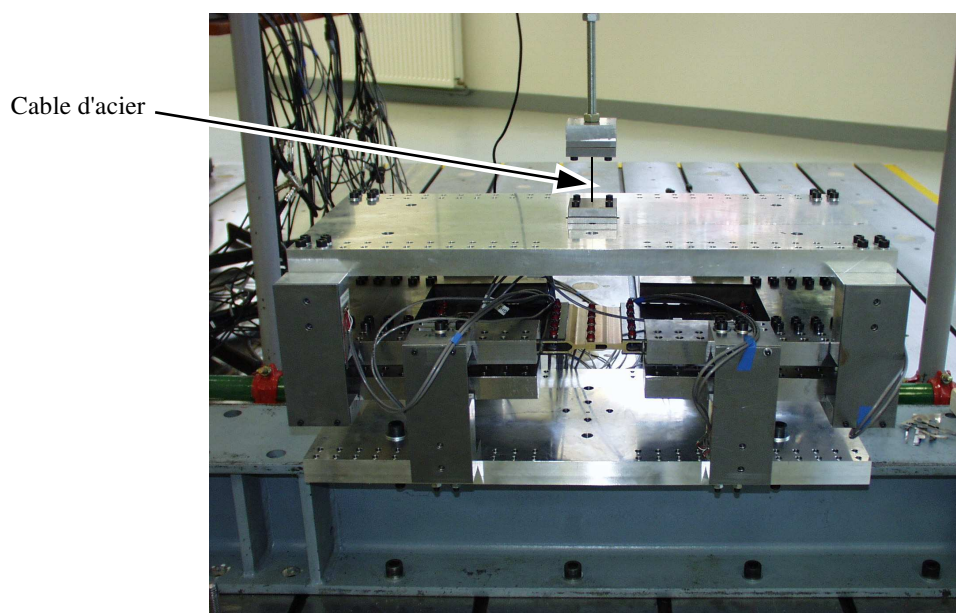


Figure 1.5: Montage avec le câble d'acier pour l'essai de lâché

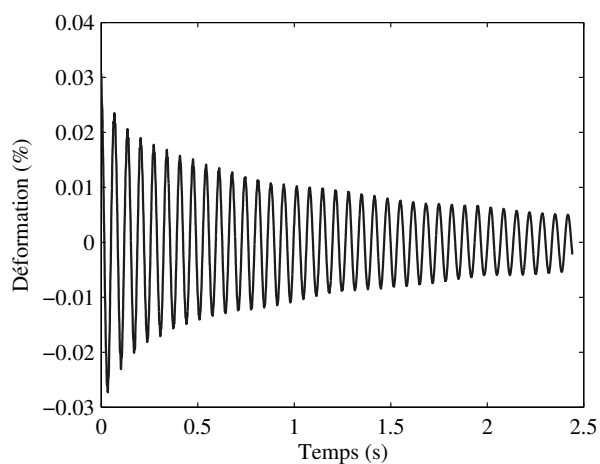


Figure 1.6: Déformation mesurée en fonction du temps pour la liaison SSS

dissipation du montage expérimentale était très faible (environ 0,1%) et que l'amortissement dans le matériau était constant en fonction du niveau d'excitation.

5.2.2 Liaisons collées

Les résultats étaient tout à fait similaires à ceux obtenus pour le matériau seul et montraient que l'amortissement ne dépendait pas du niveau d'excitation et surtout que les liaisons collées avec ou sans renfort n'apportent pas de dissipation supplémentaire. L'amortissement reste très faible, de l'ordre de 0,3% et il reste constant en fonction du niveau de déformation (Figure 1.7).

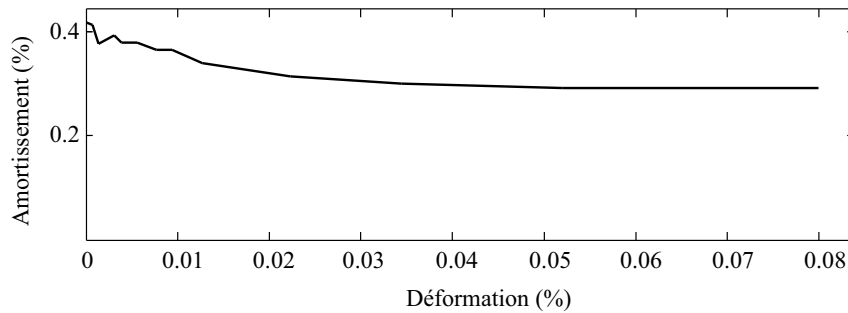


Figure 1.7: Liaison collée

5.2.3 Liaison boulonnée SSS

Non linéarité de l'amortissement Le fait que l'amortissement est non linéaire en fonction du niveau d'excitation est maintenant bien connu ([Ibrahim et Pettit 2005]) et a été confirmé par ces essais. L'amortissement variait considérablement de 0,5% à plus de 11% en fonction du niveau de déformation (Figure 1.8).

Influence du nombre de cycles L'utilisation d'un vérin hydraulique a permis de mettre en évidence l'influence du nombre de cycles comme cela a été montré dans [Tajima 1964, Chesson et Munse 1964, Gropper *et al.* 1966]. La Figure 1.8 montre que l'amortissement variait considérablement en fonction du nombre de cycles, ce qui a révélé un phénomène de rodage de l'éprouvette dû à la modification de l'état des surfaces en contact et donc à la modification du coefficient du frottement.

Influence de la fréquence d'excitation L'influence de la fréquence d'excitation pour l'essai sous vibration forcée à l'aide du pot vibrant a permis de mettre en évidence que l'amortissement dépendait très peu de la fréquence d'excitation (Figure 1.9).

Le montage et la démarche expérimentale [Leloch 2003, Maisonneuve 2004, Caignot *et al.* 2005], ont été validés en retrouvant les résultats de référence sur le matériau composite. Ensuite, les différentes liaisons présentes dans le SYLDA5 ont pu être testées en flexion pour caractériser l'amortissement en fonction du niveau d'effort imposé et de la fréquence d'excitation.

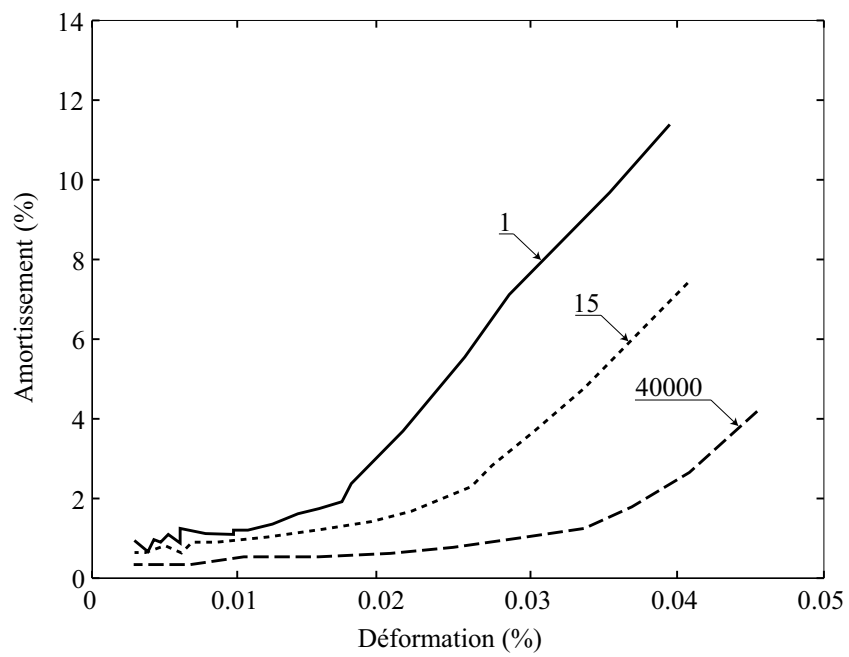


Figure 1.8: Évolution de l'amortissement en fonction de la déformation et du nombre de cycles

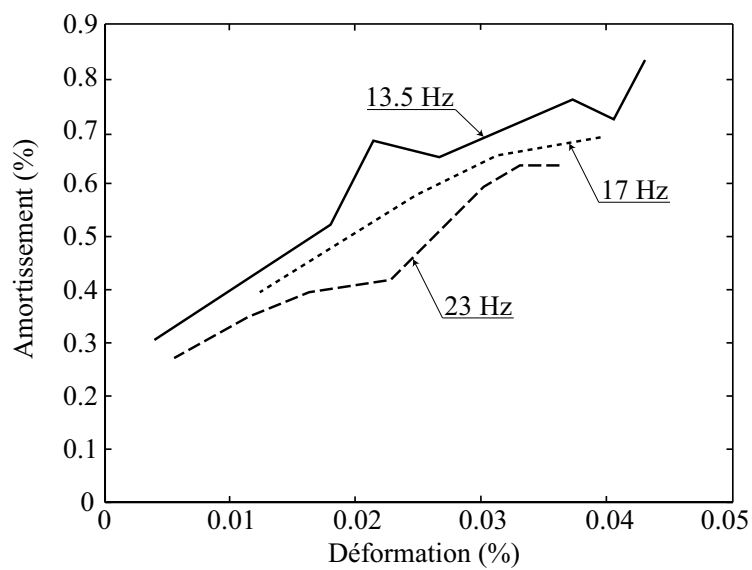


Figure 1.9: Influence de la fréquence d'excitation sur l'amortissement

Les lois d'amortissement ainsi caractérisées ont permis de réaliser une prédiction de l'amortissement complet du SYLDA5, en supposant que toutes les autres liaisons boulonnées suivent la même loi que celle de la liaison SSS. Les coefficients d'amortissement modaux ainsi prédits ont pu être comparés aux valeurs utilisées par ASTRIUM-ST, valeurs obtenues par recalage à partir d'un essai complet de la structure. La prédiction était tout à fait correcte pour les premiers modes de flexion et le premier mode de torsion. Toutefois, pour le premier mode longitudinal et les seconds modes de flexion, la prédiction n'était pas bonne. En effet, pour ces modes, la liaison SSS ne travaille pas en flexion et de plus l'énergie de déformation est concentrée dans des zones contenant d'autres liaisons qui n'ont pas été caractérisées et pour lesquelles la loi de la liaison SSS n'est probablement pas la bonne.

5.3 Limitations

La première limitation importante est que l'amortissement dans les liaisons peut dépendre fortement du type de sollicitation. Ainsi pour caractériser l'amortissement pour toutes les sollicitations, il faut concevoir autant de montages expérimentaux, définir éventuellement des éprouvettes différentes pour chaque sollicitation, ce qui conduit à des coûts prohibitifs en terme de temps et d'argent.

La seconde limitation est que dans le cadre d'une démarche de conception prédictive, il est nécessaire de connaître l'amortissement des liaisons en fonction de différents paramètres comme la géométrie, les coefficients de frottement, les serrages des boulons, les matériaux utilisés. Dans ce cadre, l'approche expérimentale devient pratiquement impossible à mettre en œuvre, c'est la raison pour laquelle, une approche numérique doit être mise en place afin de pouvoir réaliser les études paramétriques sur différentes liaisons et pour différentes sollicitations. C'est ce qui a fait l'objet du travail présenté ici.

Outils disponibles pour concevoir une stratégie d'essais virtuels

Dans ce chapitre, une analyse des outils disponibles pour résoudre le problème posé est présentée. Les possibilités et limitations de chacun d'entre eux sont discutées. On montre notamment qu'avec les options par défaut du logiciel industriel utilisé, les résultats peuvent être erronés. On est alors amené à modifier les paramètres par défaut ce qui conduit souvent au blocage de l'algorithme et il est alors nécessaire de faire une longue étude paramétrique pour trouver les options optimales qui permettent d'obtenir un résultat correct, ce qui rend cette méthode non robuste. Le logiciel de recherche utilisé permet d'obtenir une bonne solution de manière robuste mais dans des temps de calcul prohibitif.

1 Problème type à résoudre

La solution retenue pour prédire numériquement l'amortissement dans les liaisons est d'effectuer un calcul par éléments finis d'un assemblage de sous-structures en contact avec frottement. Pour simplifier l'étude, les sous-structures sont supposées avoir un comportement élastique (isotrope ou non). On se place en quasi-

statique car dans la gamme de fréquences de l'étude, les longueurs d'onde sont largement supérieures à la dimension des liaisons, n'entraînant donc pas d'effet dynamique.

Supposons qu'on souhaite calculer numériquement la dissipation d'une liaison simple constituée de deux plaques assemblées avec deux boulons qui sera chargée en traction (Figure 2.1). Les conditions de symétrie font que l'on peut simplifier le problème en ne prenant en compte qu'un quart de cette liaison (Figure 2.2) et en ajoutant des conditions limites de déplacements normaux nuls sur les plans de symétrie ainsi que sur l'extrémité gauche de la plaque inférieure. Un effort normal est imposé sur la partie droite de la plaque supérieure (Figure 2.3). On a des interfaces de contact avec frottement entre les deux plaques, ainsi qu'entre les têtes de vis et les plaques, et entre les tiges des vis et les plaques. L'effort imposé suit l'évolution temporelle suivante : il est nul entre 0 et 1 s, puis il suit un demi-sinus entre 1 et 2 s (Figure 2.3). La répartition de pression permettant d'obtenir l'effort normal souhaité de 5 MPa. La zone à effort imposé nul correspond à la mise en précharge des boulons.

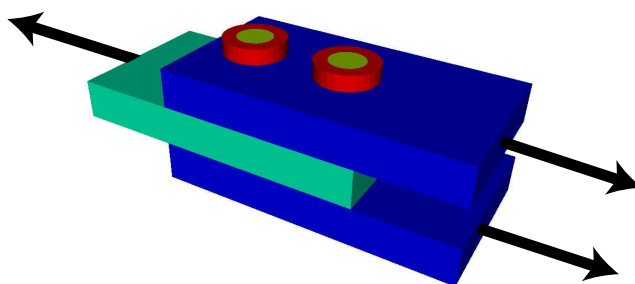


Figure 2.1: Assemblage boulonné complet

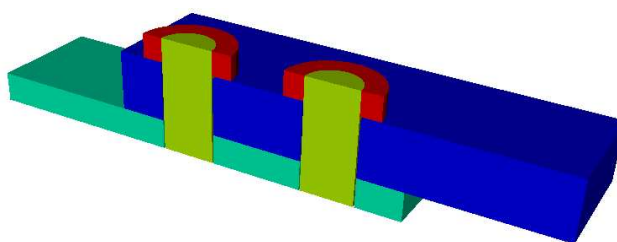


Figure 2.2: Assemblage boulonné avec les conditions de symétrie

Sur la Figure 2.3, les caractéristiques géométriques de l'assemblage sont données. Le diamètre des boulons est de 18,8 mm et le diamètre des trous de passage est

de 19 mm, donc il n'y a pas de contact entre la tige des boulons et le trou de passage. La précharge des boulons est exprimée en millimètre correspondant au déplacement dû au serrage entre la tige de la vis et l'écrou : 0,12 mm pour le boulon de gauche, 0,09 mm pour le boulon de droite. Le comportement mécanique des différentes pièces est élastique isotrope avec pour module d'Young $E = 200\,000$ MPa et coefficient de poisson $\nu = 0,3$. Le coefficient de frottement des différentes zones en contact est pris égal à 0,3. La discrétisation temporelle utilisée est de 1 pas de temps entre 0 et 1 s et de 20 pas de temps entre 1 et 2 s.

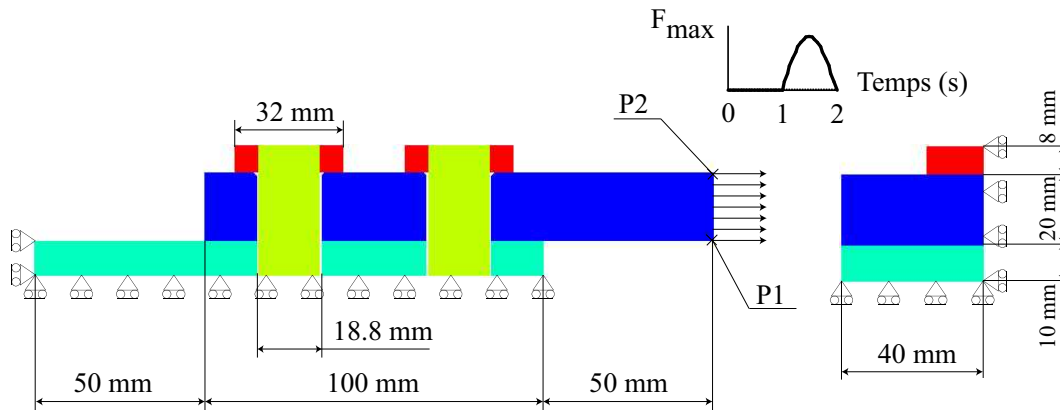


Figure 2.3: Conditions limites et principales dimensions de l'assemblage

2 Outils numériques

2.1 Logiciels

On souhaite savoir si les logiciels à notre disposition sont capables de résoudre le problème posé, à savoir calculer la dissipation par frottement dans l'assemblage sous les conditions limites imposées. On a choisi de tester deux logiciels, le premier est Abaqus [Hibbitt *et al.* 2006], qui est très utilisé dans l'industrie et réputé pour résoudre des problèmes de contact avec frottement. L'algorithme utilisé est une méthode incrémentale. Pour passer d'une itération à une suivante, il faut que toutes les équations du problème soient vérifiées et que les zones de contact restent dans un état stable (c'est-à-dire qu'il ne doit pas y avoir de changement d'état des zones de contact entre deux itérations). Le second logiciel est un code de recherche, Co-fast3d [Champaney *et al.* 1997], dédié au calcul d'assemblages de structures avec

contact et frottement. Il est basé sur la méthode LATIN [Ladevèze 1999] dans sa version monoéchelle, dont le détail est présenté dans le chapitre 3 section 1.

2.2 Algorithmes de résolution d'un problème de contact

Il y a plusieurs méthodes pour prendre en compte le frottement dans Abaqus. La méthode par défaut est une méthode par pénalisation dont on peut régler le coefficient, ce qui revient à adoucir la non linéarité du modèle de Coulomb en autorisant un glissement élastique. Une des autres méthodes implantées dans le logiciel est une méthode par multiplicateur de Lagrange qui conduit à prendre en compte le frottement de Coulomb de manière exacte.

On va comparer ces deux méthodes sur un exemple académique afin de mettre en évidence les différences de comportement de ces deux algorithmes. Pour cela, on s'intéresse au contact entre deux carrés de 50 mm de côté en deux dimensions, les conditions limites étant données sur la Figure 2.4. On impose tout d'abord une précharge sur le haut du carré supérieur puis on impose un effort en demi-sinus sur son coté tout en maintenant l'effort sur la partie haute.

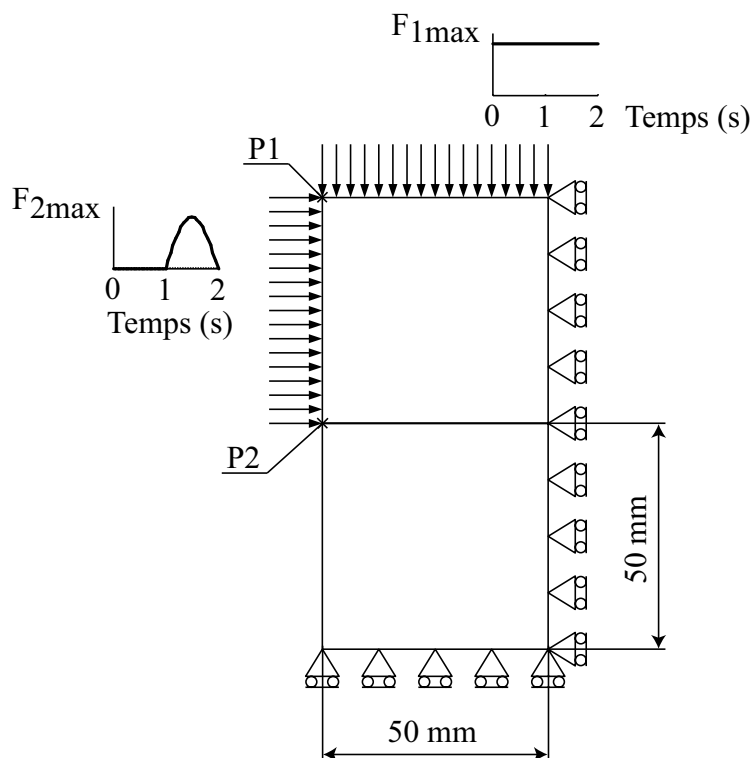


Figure 2.4: Conditions limites et dimensions du problème

La pression maximale imposé est de $F_{1max} = 10$ MPa sur la partie supérieure et de $F_{2max} = 100$ MPa sur le côté. La discrétisation temporelle utilisée est de 1 pas de temps entre 0 et 1 s et de 20 pas de temps entre 1 et 2 s.

2.2.1 Comparaison entre Cofast3d et Abaqus avec ses options par défaut

On effectue tout d'abord la comparaison entre les résultats donnés par Cofast3d et par Abaqus avec l'option par défaut (pénalisation).

Sur la Figure 2.5, on compare le déplacement du point P1 (Figure 2.4) obtenu par Abaqus avec l'option par défaut et Cofast3d : les résultats sont tout à fait similaires.

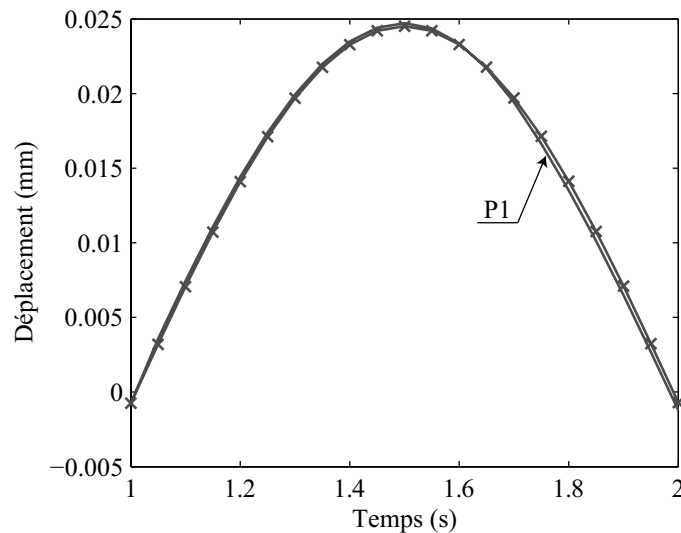
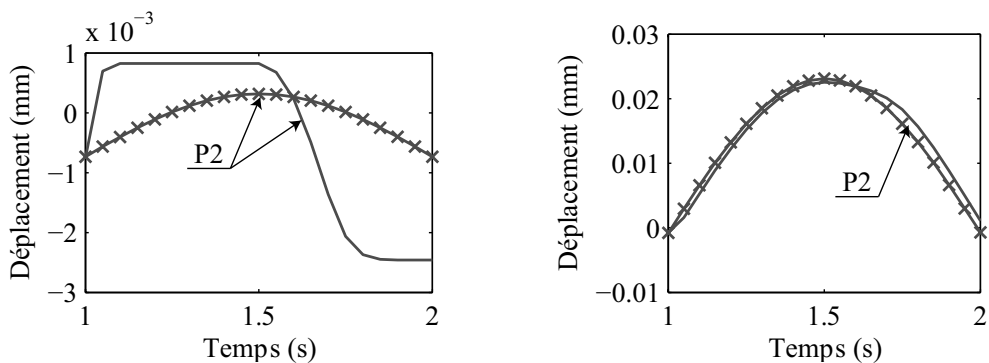


Figure 2.5: Déplacement du point P1 (résultat Abaqus option défaut (-x-) et résultat Cofast3d (—))

En revanche, quand on compare le déplacement du point P2 (Figure 2.4) appartenant à la plaque inférieure sur la Figure 2.6(a) et du point P2 appartenant à la plaque supérieure sur la Figure 2.6(b) dans les mêmes conditions, on observe que les déplacements de ces points de la zone de contact sont très différents. Notamment, la solution que génère Abaqus est complètement symétrique et revient à zéro en fin de chargement, alors que la solution attendue ne devrait pas être symétrique du fait des non linéarités du contact.

Enfin sur la Figure 2.7, on s'intéresse à l'énergie imposée et à l'énergie dissipée. Les valeurs calculées par Abaqus et par Cofast3d sont complètement différentes. On remarque surtout que la dissipation calculée avec Abaqus est nulle, ce qui n'est



(a) Déplacement de P2 appartenant à la plaque inférieure

(b) Déplacement de P2 appartenant à la plaque supérieure

Figure 2.6: Déplacement du point P2 (résultat Abaqus option défaut (-x-) et résultat Cofast3d (—))

pas possible pour ce type de problème, car s'il y a glissement, il doit y avoir de la dissipation.

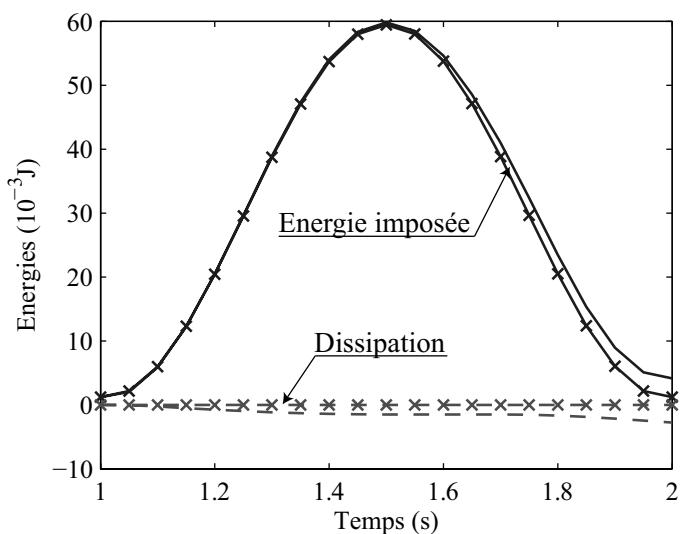


Figure 2.7: Énergies dissipée et imposée (résultat Abaqus option défaut (-x-) et résultat Cofast3d (—))

On peut donc observer que les options par défaut d'Abaqus ne sont pas bien

adaptées dans notre cas. On a donc testé les différents algorithmes avec différents paramètres afin de comparer les résultats et voir s'il est possible de converger vers une même solution entre Abaqus et Cofast3d.

2.2.2 Comparaison des algorithmes Abaqus

Pour l'algorithme de pénalisation, l'option par défaut dans Abaqus, on a effectué le calcul pour différentes valeurs de la tolérance de glissement (SLIP TOLERANCE), qui permet de gérer le coefficient de pénalisation. On fait le calcul pour 10^{-3} , 10^{-4} et 10^{-5} , la valeur par défaut étant de $5 \cdot 10^{-3}$. On a également effectué le calcul avec l'option LAGRANGE, qui permet d'obtenir la solution éléments finis exacte, qui servira de référence, nommé simplement Abaqus sur les courbes suivantes.

Le déplacement du point P1 était déjà correct, il n'y a donc aucune différence quel que soit l'algorithme choisi (Figure 2.8). Pour le déplacement du point P2

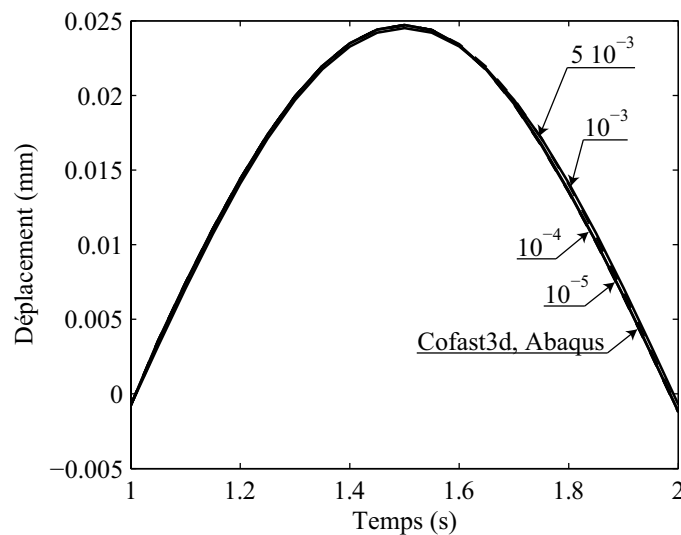
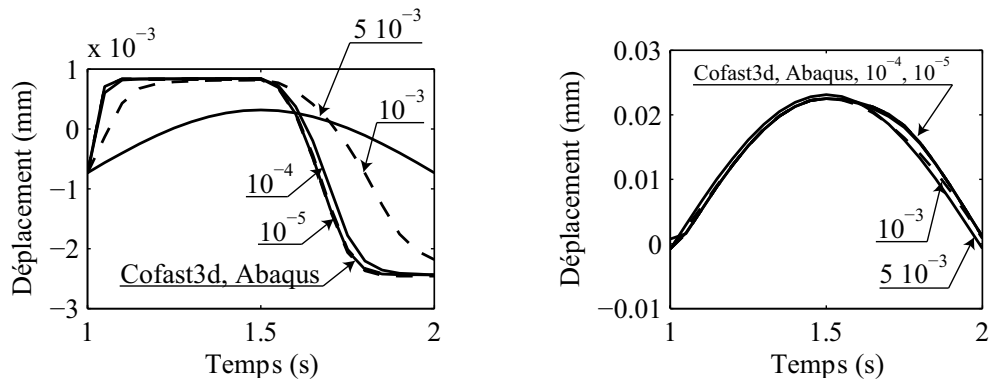


Figure 2.8: Déplacement du point P1 : comparaison des algorithmes

appartenant à la plaque inférieure, il faut avoir un coefficient de tolérance au glissement de 10^{-5} pour obtenir la même solution que Cofast3d et Abaqus avec l'option LAGRANGE (Figure 2.9(a)). Pour le déplacement du point P2 appartenant à la plaque supérieure, il faut également un coefficient de tolérance au glissement de 10^{-5} (Figure 2.9(b)).

La convergence pour les énergies est obtenue à partir d'un coefficient de tolérance au glissement de 10^{-4} (Figure 2.10). Encore une fois, les résultats obtenus avec Co-



(a) Déplacement de P2 appartenant à la plaque inférieure

(b) Déplacement de P2 appartenant à la plaque supérieure

Figure 2.9: Déplacement du point P2 : comparaison des algorithmes

fast3d et Abaqus avec l'option LAGRANGE sont tout à fait similaires.

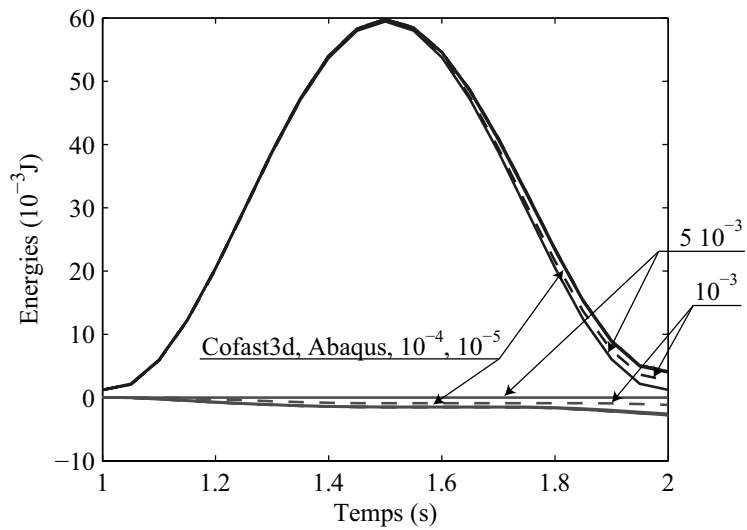


Figure 2.10: Énergie imposée et dissipée : comparaison des algorithmes

On peut également visualiser ce qui se passe en observant la déformée pour le dernier pas de temps. Notre référence est Abaqus avec l'option LAGRANGE (Figure 2.11(a)). On peut voir que la déformée obtenue avec l'option par défaut

est complètement différente (Figure 2.11(b)). En fait, il n'y a eu aucun glissement entre les deux surfaces de contact, ce qui explique le fait que l'énergie dissipée soit complètement nulle pour cette option. La déformée obtenue avec Cofast3d est tout à fait comparable à celle obtenue avec Abaqus avec l'option LAGRANGE (Figure 2.11(c)).

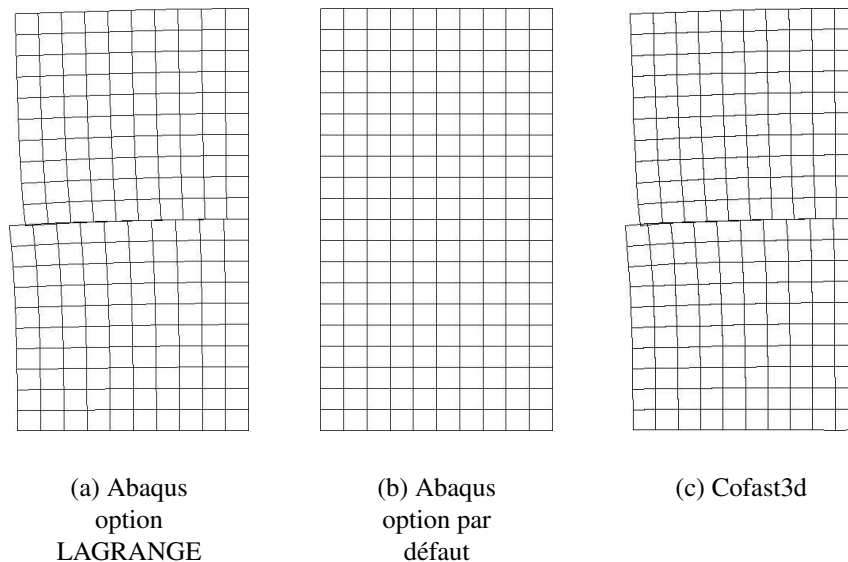


Figure 2.11: Comparaison des déformées pour le dernier pas de temps

2.3 Calcul d'une liaison avec un boulon

Le problème à résoudre est identique à notre problème type présenté dans la section 1, sauf que l'on ne considère qu'un seul boulon. Les conditions limites en effort et déplacement restent inchangées. Ce problème a pu être traité par Abaqus avec l'option LAGRANGE et on peut donc comparer les résultats obtenus par Cofast3d et Abaqus.

La Figure 2.12 présente la comparaison des déplacements du point P1 et du point P2 (définis sur la Figure 2.3) en fonction du temps. On peut voir que les résultats sont tout à fait similaires. Si on s'intéresse aux énergies imposée et dissipée, on peut voir sur la Figure 2.13, qu'elles sont également très similaires.

La méthode LATIN dans sa version monoéchelle, à l'aide du logiciel Cofast3d, permet donc de résoudre le problème de contact et de calculer la dissipation dans une liaison simple.

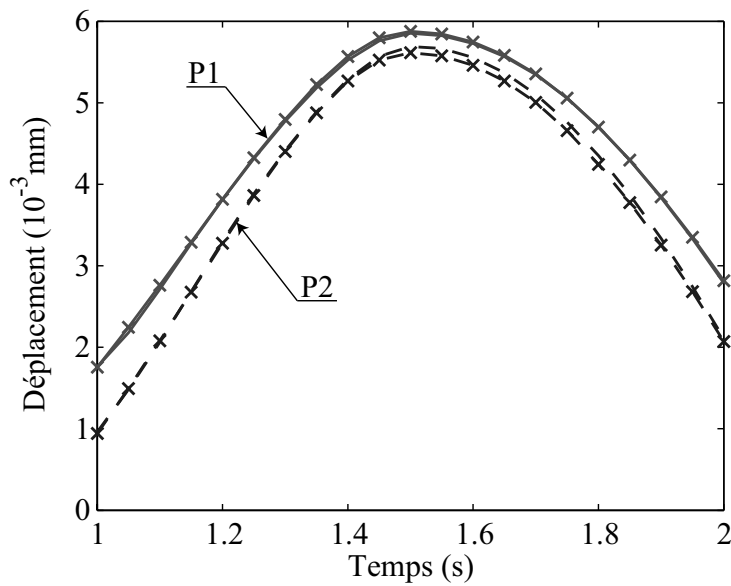


Figure 2.12: Déplacements des points P1 et P2 (résultat Abaqus option LAGRANGE (-x-) et résultat Cofast3d (—))

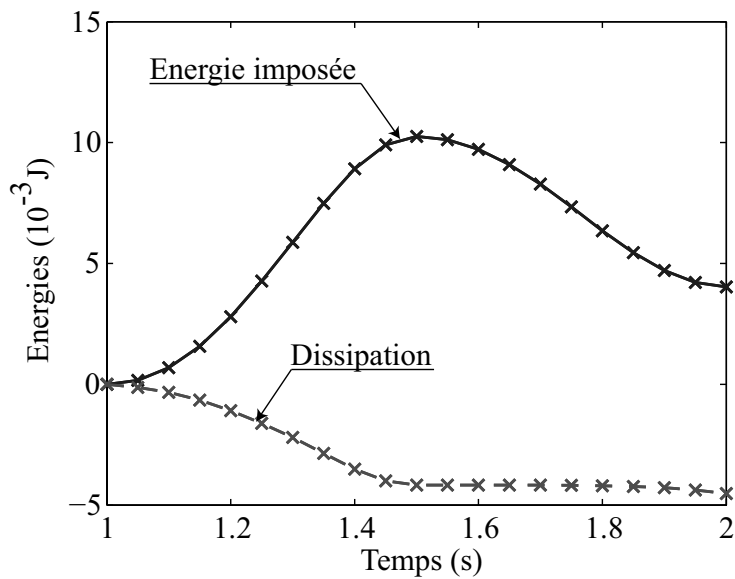


Figure 2.13: Énergies dissipée et imposée (résultat Abaqus option LAGRANGE (-x-) et résultat Cofast3d (—))

2.4 Calcul du problème type

On va maintenant résoudre le problème type défini dans la section 1. Compte-tenu de ce que l'on vient de montrer dans les sections précédentes, la solution de

référence devrait être obtenue avec Abaqus et l'option LAGRANGE. Malheureusement, avec cette option, il a été impossible de générer une solution et on n'a pas réussi à trouver la bonne combinaison de paramètres permettant de faire converger Abaqus (en effet, on peut gérer tous les paramètres de l'algorithme : le coefficient de diminution du pas de temps, le nombre de tentatives de chaque sous-itération, le nombre de fois où l'on décrémente le pas de temps, ...).

On va présenter la comparaison des résultats du calcul dans le cas où la solution est générée par Abaqus avec ses options par défaut et dans le cas où l'on a essayé de trouver le meilleur coefficient de pénalisation permettant d'obtenir une solution.

2.4.1 Comparaison entre Cofast3d et Abaqus avec ses options par défaut

Abaqus utilise un algorithme de pénalisation dont le coefficient, exprimé en terme de tolérance au glissement, vaut $5 \cdot 10^{-3}$.

La Figure 2.14 présente la comparaison du déplacement du point P1 et du point P2 (définis sur la Figure 2.3) en fonction du temps, on peut voir que les résultats sont très différents. Si on s'intéresse aux énergies imposée et dissipée, on peut voir sur la Figure 2.15, qu'elles sont également très différentes. Cependant, il faut remarquer qu'Abaqus donne une énergie dissipée nulle, ce qui n'est physiquement pas possible.

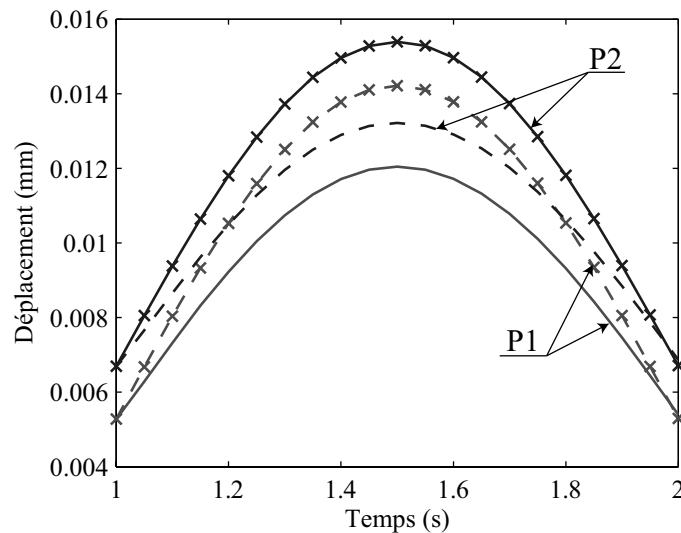


Figure 2.14: Déplacements des points P1 et P2 (résultat Abaqus option défaut (-x-) et résultat Cofast3d (—))

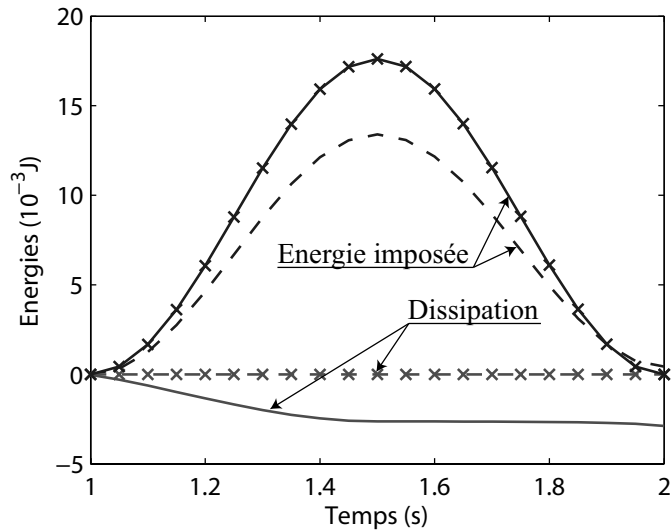


Figure 2.15: Énergies dissipée et imposée (résultat Abaqus option défaut (-x-) et résultat Cofast3d (—))

2.4.2 Comparaison entre Cofast3d et Abaqus avec le meilleur coefficient de pénalisation

Comme l'option LAGRANGE conduit à bloquer l'algorithme de résolution, la meilleure solution qu'Abaqus pourra générer sera pour la plus petite valeur du coefficient de tolérance au glissement. On n'a pas pu descendre en dessous de 10^{-5} pour le coefficient de tolérance au glissement, cependant les déplacements des points P1 et P2 sont maintenant tout à fait comparables (Figure 2.16) ainsi que l'énergie imposée et l'énergie dissipée (Figure 2.17).

Avec ces paramètres, on donne les résultats en terme de temps de calcul et nombre d'itérations dans le tableau 2.1. Le gain en terme de temps de calcul est de 3,8 ce qui commence à être non négligeable. Cela permet de faire un calcul à l'échelle d'une semaine en seulement deux jours par exemple.

	Abaqus	Cofast3d
Mémoire (Mo)	256	75
Nombre d'incréments	31	21
Temps CPU (s)	676	175

Tableau 2.1: Comparaison Abaqus convergé - Cofast3d

Le gain en terme de mémoire est de 3,4, ce qui est important. Cependant, il n'est

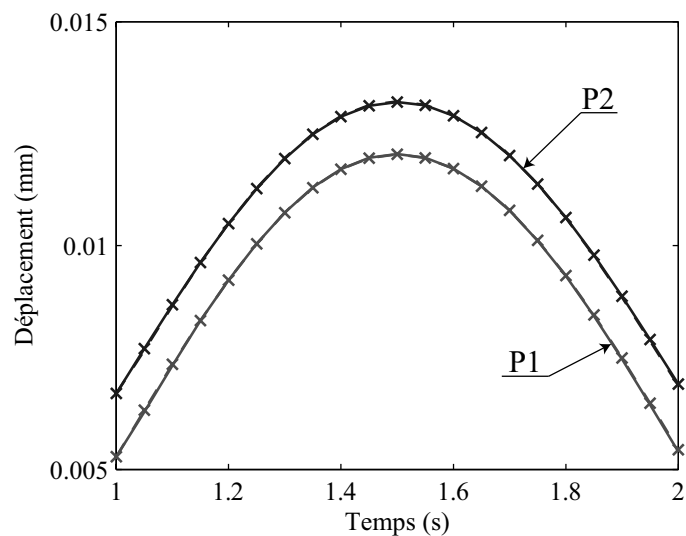


Figure 2.16: Déplacements des points P1 et P2 (résultat Abaqus coefficient de pénalisation optimal (-x-) et résultat Cofast3d (—))

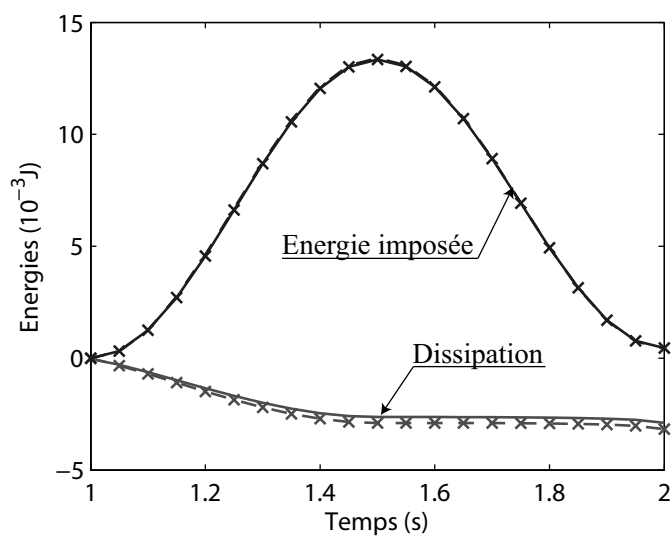


Figure 2.17: Énergies dissipée et imposée (résultat Abaqus coefficient de pénalisation optimal (-x-) et résultat Cofast3d (—))

pas représentatif de la réalité car le problème traité ne comporte que peu d'inconnues (25 000 degrés de liberté).

3 Conclusions - limitations

On a vu ici que les logiciels Abaqus et Cofast3d sont capables de résoudre le problème du calcul de la dissipation dans une liaison. Abaqus ne permet malheureusement pas de contrôler la qualité de la solution générée. En effet, si le coefficient de tolérance au glissement minimal, permettant d'assurer la convergence, n'est pas suffisamment bas, la solution obtenue peut très bien être erronée. Même si on ne recherche la dissipation qu'à 10% près, on ne peut pas se permettre de donner un résultat sans savoir s'il approche la réalité ou s'il est complètement faux. De plus, la recherche du meilleur coefficient de tolérance au glissement ne peut pas être automatisée, ce qui implique un surcoût en temps de calcul non négligeable.

La solution basée sur la méthode LATIN dans sa version monoéchelle, présente ici deux avantages. Le premier est que comme on génère une solution sur l'espace-temps tout entier à chaque itération, il suffit d'avoir un critère basé sur la convergence de la dissipation pour arrêter l'algorithme comme on le verra dans le chapitre 3. Le second avantage, c'est qu'elle permet de générer la solution quatre fois plus rapidement, ce qui est un avantage non négligeable en milieu industriel.

Cependant, la solution Cofast3d présente des limites. Malgré un gain conséquent en terme de temps de calcul par rapport à Abaqus, on est loin de pouvoir prétendre calculer une liaison complète avec plusieurs dizaines de milliers de degrés de liberté pour tous les cas de charges, pour tous les efforts dans un temps raisonnable. La solution pour palier ce problème est d'effectuer le calcul avec la méthode LATIN dans sa version multiéchelle en espace.

Méthode LATIN multiéchelle en espace

Dans ce chapitre, la stratégie de calcul retenue pour résoudre le problème présenté dans le chapitre 2 est détaillée. On rappelle tout d'abord la stratégie dans sa version monoéchelle. Puis les aspects multiéchelles, nécessaires pour assurer l'extensibilité numérique de la méthode, sont présentés. Le gain apporté par les aspects multiéchelles, notamment en terme de temps de calcul, seront illustrés sur divers exemples.

1 Présentation de la méthode LATIN dans sa version monoéchelle

La stratégie monoéchelle a été implantée dans le logiciel Cofast3d [Champaney 1996] et a fait l'objet de plusieurs travaux. L'extension multiéchelle de la méthode est apparue avec les travaux de [Ladevèze et Dureisseix 2000, Ladevèze *et al.* 2001] dans le cadre de la statique linéaire. Les travaux de [Ladevèze et Nouy 2003] ont apporté de nombreuses avancées, non seulement dans la manière de traiter l'aspect multiéchelle à la fois en espace et en temps mais aussi dans l'amélioration de plusieurs points clés de la stratégie. Récemment, les travaux de [Guidault *et al.* 2007] ont permis de coupler la stratégie de calcul multiéchelle à la XFEM pour le traitement de macrofissures dans une structure. Les principes de la stratégie ont été

repris pour le traitement des systèmes de tenségrité par [Nineb *et al.* 2007] dans la perspective d'une extension aux milieux granulaires. Les travaux de [Ladevèze *et al.* 2006] ont permis d'effectuer le calcul de la dégradation des composites par décohésion fibre-matrice grâce, notamment, à l'ajout d'une troisième échelle. On notera également les travaux de [Néron et Dureisseix 2008a -b] dans lesquels la méthode est appliquée au traitement de problèmes couplés fluide-structure. Dernièrement, la méthode a été étendue pour prendre en compte des aspects multi-échelle en temps [Ladevèze *et al.* 2009] et la rendre ainsi extensible en temps.

1.1 Décomposition de domaine - Problème à résoudre

Le problème à traiter est celui d'un assemblage de pièces, naturellement sous-structuré, soumis à un chargement quasi-statique sur une de ses frontières, notée $\partial\Omega_F$, et des déplacements imposés sur une autre partie, notée $\partial\Omega_U$. Le comportement de la structure peut être non linéaire, toutefois la méthode sera présentée dans le cadre d'un comportement élastique linéaire. Les non linéarités du problème se situent aux interfaces entre les sous-structures qui auront un comportement représenté par les lois de Coulomb. Le cadre de l'étude est celui des petites perturbations.

L'assemblage est donc composé d'un ensemble de sous-structures Ω_E noté \mathbf{E} qui sont reliées entre elles par des interfaces $\Gamma_{EE'}$ (Figure 3.1). Le premier point clé de la méthode est de considérer les interfaces comme des entités mécaniques à part entière, qui possèdent leur propre jeu de variables et un comportement. On peut considérer un grand nombre de comportements mais dans ce travail, on s'intéressera essentiellement à des comportement d'interface de type contact avec frottement ainsi que des comportements d'interface parfaite. De manière plus générale, les conditions limites sont également prises en compte par l'intermédiaire d'un comportement d'interface de type effort imposé, déplacement imposé, symétrie...

Les variables associées aux interfaces sont des champs d'efforts $\underline{F}_{EE'}$ et $\underline{F}_{E'E}$ et les variables cinématiques duales. En statique, il s'agit des champs de déplacement $\underline{W}_{EE'}$ et $\underline{W}_{E'E}$ et en quasi-statique des champs de vitesse $\dot{\underline{W}}_{EE'}$ et $\dot{\underline{W}}_{E'E}$ (Figure 3.2).

Si on considère une sous-structure Ω_E entourée de n sous-structures $\Omega_{E'_1}, \dots, \Omega_{E'_n}$, elle est soumise à des conditions mixtes force-déplacement que l'on suppose compatible pour assurer l'existence et l'unicité de la solution du problème. On note σ_E et \underline{u}_E les contraintes et déplacements de la sous-structure Ω_E appartenant respectivement aux espaces \mathcal{S}_E et \mathcal{U}_E . On note \underline{F}_E (répartition d'effort sur tous les bords de Ω_E décrivant $\underline{F}_{EE'_1}, \dots, \underline{F}_{EE'_n}$) et $\dot{\underline{W}}_E$ (répartition des vitesses sur tous les bords de Ω_E décrivant $\dot{\underline{W}}_{EE'_1}, \dots, \dot{\underline{W}}_{EE'_n}$) appartenant respectivement aux espaces \mathcal{F}_E et \mathcal{W}_E .

Le problème sous-structuré à résoudre s'exprime de la façon suivante :

Trouver $\mathbf{s} = \{\mathbf{s}_E\}_{E \in \mathbf{E}}$ avec $\mathbf{s}_E = (\underline{\dot{u}}_E, \dot{\underline{W}}_E, \sigma_E, \underline{F}_E) \in \mathcal{S}_E = \mathcal{U}_E \times \mathcal{W}_E \times \mathcal{S}_E \times \mathcal{F}_E$ qui vérifient les équations :

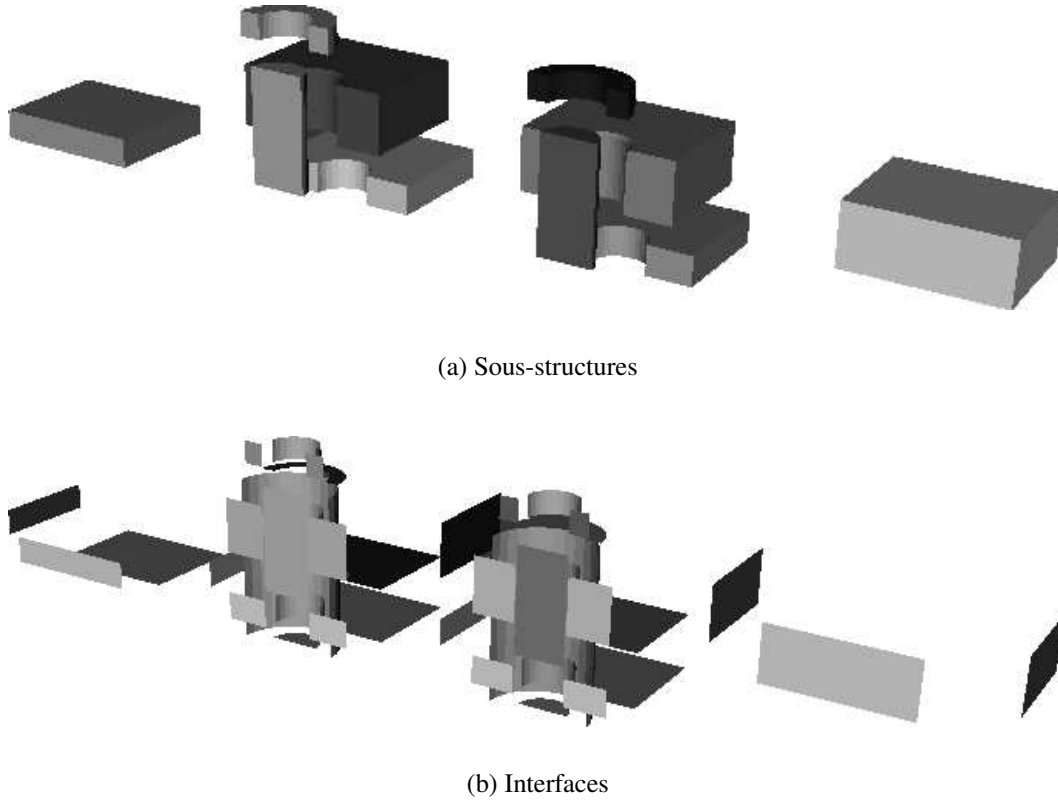


Figure 3.1: Décomposition de domaine de l'assemblage boulonné

- Admissibilité cinématique des sous-structures :

$$\forall t \in [0, T], \forall M \in \partial\Omega_E, \quad \underline{u}_E|_{\partial\Omega_E} = \underline{W}_E \quad (3.1)$$

- Admissibilité statique des sous-structures :

$$\forall t \in [0, T], \forall \underline{u}^* \in \mathcal{U}_E^*, \quad \int_{\Omega_E} \text{Tr}(\underline{\sigma}_E \underline{\varepsilon}(\underline{u}^*)) \, d\Omega - \int_{\Omega_E} \underline{f}_d \cdot \underline{u}^* \, d\Omega - \int_{\partial\Omega_E} \underline{F}_E \cdot \underline{u}^* \, dS = 0 \quad (3.2)$$

où \mathcal{U}_E^* est l'espace homogène associé à l'espace \mathcal{U}_E .

- Relation de comportement des sous-structures :

$$\forall t \in [0, T], \quad \underline{\sigma}_E = \mathbf{K}(\underline{\varepsilon}(\underline{u}_E)) \quad (3.3)$$

où \mathbf{K} est l'opérateur de Hooke.

- Équation d'équilibre des interfaces :

$$\forall t \in [0, T], \forall M \in \Gamma_{EE'}, \quad \underline{F}_{EE'} + \underline{F}_{E'E} = 0 \quad (3.4)$$

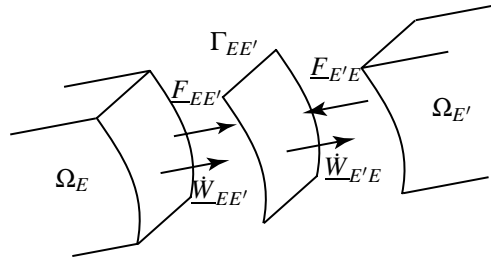


Figure 3.2: Échange sous-structures / interfaces

- Relation de comportement des interfaces :

$$\forall t \in [0, T], \forall M \in \Gamma_{EE'}, \quad \mathcal{R}(\underline{\dot{W}}_{EE'}, \underline{F}_{EE'}, \underline{\dot{W}}_{E'E}, \underline{F}_{E'E}) = 0 \quad (3.5)$$

où \mathcal{R} est un opérateur linéaire caractérisant le comportement de l'interface.

1.2 Comportements d'interface particuliers

1.2.1 Interface parfaite

Les interfaces parfaites transmettent les efforts et vitesses entre les sous-structures.

- Équilibre :

$$\underline{F}_{EE'} + \underline{F}_{E'E} = 0$$

- Comportement :

$$\underline{\dot{W}}_{EE'} - \underline{\dot{W}}_{E'E} = 0$$

1.2.2 Interface de type contact avec frottement

Les interfaces de contact avec frottement ont un comportement différent selon la normale \underline{n}_E et la tangente à l'interface. On note f le coefficient de frottement. $\Pi_{EE'}$ est l'opérateur de projection orthogonale à \underline{n}_E .

- Equilibre :

$$\underline{F}_{EE'} + \underline{F}_{E'E} = 0$$

- Contact :

$$\text{si } \underline{n}_E \cdot (\underline{W}_{E'E} - \underline{W}_{EE'}) > 0,$$

$$\text{alors } \underline{n}_E \cdot \underline{F}_{EE'} = \underline{n}_E \cdot \underline{F}_{E'E} = 0$$

(décollement),

$$\text{si } \underline{n}_E \cdot (\underline{W}_{E'E} - \underline{W}_{EE'}) = 0,$$

$$\text{alors } \underline{n}_E \cdot \underline{F}_{EE'} + \underline{n}_E \cdot \underline{F}_{E'E} = 0$$

(contact),

- Frottement :

$$\text{si } \|\Pi_{EE'} \underline{F}_{EE'}\| < f |\underline{n}_E \cdot \underline{F}_{EE'}|,$$

$$\begin{array}{ll}
 \text{alors} & \Pi_{EE'}(\dot{W}_{EE'} - \dot{W}_{E'E}) = 0 & (\text{adhérence}), \\
 \text{si} & \|\Pi_{EE'} \underline{F}_{EE'}\| = f |\underline{n}_E \cdot \underline{F}_{EE'}|, \\
 \text{alors} & \exists \lambda > 0 \quad | \quad \Pi_{EE'}(\dot{W}_{E'E} - \dot{W}_{EE'}) = -\lambda \underline{F}_{EE'} & (\text{glissement}).
 \end{array}$$

On s'est restreint ici à présenter les comportements de deux types d'interface. Les conditions limites sont également introduites par l'intermédiaire d'une interface en considérant une sous-structure voisine virtuelle qui imposera par exemple l'effort imposé ou le déplacement imposé souhaité.

L'introduction d'interfaces avec un comportement complexe ne pose aucun problème, étant donné le caractère local de leur comportement. On peut noter que des interfaces de type contact avec jeu, interfaces cohésives ont déjà été mises en place.

1.3 Stratégie de résolution itérative

Le deuxième point clé de la méthode est d'utiliser une stratégie de résolution itérative non incrémentale, la LATIN, dont l'origine date des années 80 [Ladevèze 1985] et qui est détaillée dans [Ladevèze 1999]. Cette méthode permet de construire des approximations successives de la solution sur tout le domaine espace-temps et est très bien adaptée pour les problèmes non linéaires d'évolution. Elle repose sur trois principes et seuls deux d'entre eux ont été mis en œuvre dans ce travail.

■ Le premier point consiste à séparer les difficultés de la résolution en séparant les équations en deux groupes distincts. Un groupe, noté \mathbf{A}_d , est constitué des équations linéaires, éventuellement globales (3.1 et 3.2). L'autre groupe, noté Γ , est constitué des équations locales, éventuellement non linéaires (3.4 et 3.5) ainsi que du comportement non linéaire des sous-structures. Ici, comme le comportement des sous-structures est linéaire, l'équation (3.3) est placée dans le groupe \mathbf{A}_d afin de faciliter la résolution.

■ Le second point consiste à appliquer une stratégie itérative de résolution à deux pas, en cherchant une solution vérifiant successivement les équations des deux groupes \mathbf{A}_d et Γ . On passe de la solution d'un groupe d'équations à un autre à l'aide de directions de recherche \mathbf{E}^+ et \mathbf{E}^- qui conditionnent la convergence de la méthode, comme cela est schématisé sur la Figure 3.3. Une itération de la méthode se décompose en deux étapes, appelées « étape locale » et « étape linéaire ».

Les directions de recherche couplent les efforts et les vitesses entre deux sous-structures et le choix de celles-ci est primordial pour assurer une convergence efficace de la méthode. Le choix optimal de ces directions de recherche ainsi que leur mise en œuvre pratique a été rappelé dans [Violeau 2003].

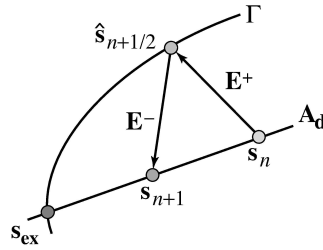


Figure 3.3: Schéma itératif de la méthode LATIN

1.3.1 Étape locale à l'itération n+1/2

L'étape locale consiste à trouver $\hat{s}_{n+1/2} \in \Gamma$ connaissant $s_n \in A_d$ en utilisant une direction de montée E^+ . Ainsi sur chaque interface, $\hat{s}_{n+1/2}$ doit vérifier :

$$\begin{aligned} \forall t \in [0, T], \forall M \in \Gamma_{EE'}, \quad & \left(\hat{F}_{EE'} - \underline{F}_{EE'} \right) - k^+ \left(\hat{W}_{EE'} - \underline{W}_{EE'} \right) = 0 \\ \forall t \in [0, T], \forall M \in \Gamma_{E'E}, \quad & \left(\hat{F}_{E'E} - \underline{F}_{E'E} \right) - k^+ \left(\hat{W}_{E'E} - \underline{W}_{E'E} \right) = 0 \end{aligned}$$

où k^+ est un opérateur défini positif.

On notera que l'étape locale ne pose aucune difficulté et que les directions de recherche combinée avec l'équilibre de l'interface (3.4) et la relation de comportement (3.5) constituent un système d'équation de quatre équations à quatre inconnues, qui peut être résolu et implanté analytiquement dans le code de calcul.

1.3.2 Étape linéaire à l'itération n+1

L'étape linéaire consiste à trouver $s_{n+1} \in A_d$ connaissant $\hat{s}_{n+1/2} \in \Gamma$ en utilisant une direction de descente E^- . Ainsi sur chaque interface, s_{n+1} doit vérifier :

$$\forall t \in [0, T], \forall M \in \Gamma_{EE'}, \quad \left(\underline{F}_{EE'} - \hat{F}_{EE'} \right) + k^- \left(\underline{W}_{EE'} - \hat{W}_{EE'} \right) = 0 \quad (3.6)$$

$$\forall t \in [0, T], \forall M \in \Gamma_{E'E}, \quad \left(\underline{F}_{E'E} - \hat{F}_{E'E} \right) + k^- \left(\underline{W}_{E'E} - \hat{W}_{E'E} \right) = 0 \quad (3.7)$$

où k^- est un opérateur défini positif.

Ces directions de recherche sont injectées dans les équations (3.2) et (3.3) pour obtenir le problème à résoudre suivant :

$$\begin{aligned} \text{Trouver } (\underline{u}_E, \underline{W}_E) \in \mathcal{U}_E \times \mathcal{W}_E \text{ tel que : } & \quad \forall t \in [0, T], \forall \underline{u}^* \in \mathcal{U}_E^*, \\ \int_{\Omega_E} \text{Tr}(\varepsilon(\underline{u}_E) \mathbf{K} \varepsilon(\underline{u}^*)) d\Omega + \int_{\partial\Omega_E} k^- \underline{W}_E \cdot \underline{u}^* dS = & \\ \int_{\Omega_E} \underline{f}_d \cdot \underline{u}^* d\Omega + \int_{\partial\Omega_E} (\widehat{\underline{F}}_E + k^- \widehat{\underline{W}}_E) \cdot \underline{u}^* dS & \end{aligned}$$

Cette équation est une équation différentielle linéaire du premier ordre que l'on résoud avec un schéma de type Euler implicite. Seules les vitesses sur les interfaces sont à sauvegarder pour passer d'un pas de temps à un autre.

Après avoir déterminé \underline{W}_E , on calcule \underline{F}_E à l'aide de la direction de recherche.

1.3.3 Contrôle des itérations

Dans les méthodes itératives de décomposition de domaine, plusieurs indicateurs sont possibles. Pour les méthodes primales, c'est naturellement le résidu sur les déplacements qui pilote l'algorithme et réciproquement pour les méthodes duales, c'est le résidu en effort.

Dans notre cas, l'indicateur d'erreur construit dans la méthode LATIN est basé sur la vérification des directions de recherche, choisi en pratique tel que $k^- = k^+ = k$, sur le principe qu'à convergence la solution vérifie les équations du groupe \mathbf{A}_d et du groupe Γ . L'erreur sur les directions de recherche est calculée de la manière suivante :

$$\eta^2 = \frac{\sum_{E \in \mathbf{E}} [\|\underline{W}_E - \widehat{\underline{W}}_E\|_{\partial\Omega_E}^{W,k} + \|\underline{F}_E - \widehat{\underline{F}}_E\|_{\partial\Omega_E}^{F,k}]}{\sum_{E \in \mathbf{E}} [\|\underline{W}_E\|_{\partial\Omega_E}^{W,k} + \|\widehat{\underline{W}}_E\|_{\partial\Omega_E}^{W,k} + \|\underline{F}_E\|_{\partial\Omega_E}^{F,k} + \|\widehat{\underline{F}}_E\|_{\partial\Omega_E}^{F,k}]} \quad (3.8)$$

$$\text{où } \|\underline{W}\|_{\partial\Omega_E}^{W,k} = \int_{\partial\Omega_E} \underline{W} \cdot k \underline{W} dS \text{ et } \|\underline{F}\|_{\partial\Omega_E}^{F,k} = \int_{\partial\Omega_E} \underline{F} \cdot k^{-1} \underline{F} dS.$$

Pour assurer la convergence de la stratégie pour un grand nombre de comportements, l'étape linéaire est modifiée en incluant une phase de relaxation. Ainsi, en notant \tilde{s}_{n+1} la solution issue de l'étape linéaire directe, on définit la nouvelle solution relaxée $s_{n+1} \in \mathbf{A}_d$, par la relation :

$$s_{n+1} = \mu \tilde{s}_{n+1} + (1 - \mu) s_n$$

où le paramètre μ est choisi en pratique à 0,8.

En pratique, il est nécessaire d'atteindre un critère d'erreur de 10^{-4} en 3D pour obtenir une solution globalement correcte et pour obtenir une solution localement correcte, il est souvent nécessaire d'aller jusqu'à 10^{-5} , notamment pour avoir une solution fine dans les zones de contact qui nous intéressent tout particulièrement dans cette étude.

Cette méthode dans sa version monoéchelle, est celle implantée dans le logiciel Cofast3d.

2 Ajout des aspects multiéchelles en espace

2.1 Introduction des aspects multiéchelles

La méthode de décomposition de domaine mixte présentée précédemment permet de résoudre des problèmes de très grande taille mais, on ne peut pas assurer l'extensibilité numérique de la méthode. Avec une itération, on transmet uniquement une information d'interface d'une sous-structure à une autre, donc plus le nombre de sous-structures est grand, plus il faudra d'itérations pour le résoudre. L'ajout d'un problème global défini à une échelle macroscopique permet de palier ce défaut.

L'aspect multiéchelle nécessite de définir des quantités macro et micro ainsi qu'un dialogue entre les échelles. Une fois ces choix effectués, la vérification des contraintes d'admissibilité des quantités macro conduit à la mise en place d'un problème macroscopique ([Dureisseix et Ladevèze 1998]).

2.1.1 Séparation des échelles

La séparation des échelles est effectuée uniquement sur les quantités d'interface, qui sont écrites sous la forme $\underline{F}_E = \underline{F}_E^m + \underline{F}_E^M$ et $\underline{\dot{W}}_E = \underline{\dot{W}}_E^m + \underline{\dot{W}}_E^M$, ce qui diffère des procédures d'homogénéisation classique. Sur une interface $\Gamma_{EE'}$ entre les sous-structures Ω_E et $\Omega_{E'}$, les vitesses macro et les efforts macro sont choisis dans des espaces de dimension finie $\mathcal{W}_{EE'}^M$ et $\mathcal{F}_{EE'}^M$ avant toute discrétisation de l'interface.

Une fois les espaces macro choisis, on définit la partie macro $\underline{\dot{W}}_E^M$ d'un champ $\underline{\dot{W}}_E \in \mathcal{W}_E$ par :

$$\forall F^* \in \mathcal{F}_{EE'}^M, \quad \int_{\Gamma_{EE'}} (\underline{\dot{W}}_E^M - \underline{\dot{W}}_E) \cdot F^* dSdt = 0 \quad (3.9)$$

et la partie macro \underline{F}_E^M d'un champ $\underline{F}_E \in \mathcal{F}_E$ par :

$$\forall \dot{W}^* \in \mathcal{W}_{EE'}^M, \quad \int_{\Gamma_{EE'}} (\underline{F}_E^M - \underline{F}_E) \cdot \dot{W}^* dSdt \quad (3.10)$$

Les efforts et vitesses micro complémentaires sont simplement donnés par :

$$\underline{F}_E^m = \underline{F}_E - \underline{F}_E^M \quad \text{et} \quad \underline{\dot{W}}_E^m = \underline{\dot{W}}_E - \underline{\dot{W}}_E^M \quad (3.11)$$

De ces trois relations découlent la relation de découplage entre les parties macro et les parties micro qui s'écrit :

$$\forall t \in [0, T], \quad \int_{\Gamma_{EE'}} \underline{F}_E \cdot \underline{\dot{W}}_E d\Gamma = \int_{\Gamma_{EE'}} \underline{F}_E^M \cdot \underline{\dot{W}}_E^M d\Gamma + \int_{\Gamma_{EE'}} \underline{F}_E^m \cdot \underline{\dot{W}}_E^m d\Gamma \quad (3.12)$$

Ici, le choix qui est fait pour l'espace des efforts macro est d'extraire les résultantes et les moments des efforts. D'après le principe de Saint-Venant, les parties micro, qui sont alors à résultantes et moments nuls n'ont qu'un effet localisé et c'est la propagation des quantités macro dans l'ensemble de la structure qui va permettre de recouvrir les interactions entre les sous-structures. Un choix identique est fait pour l'espace des déplacements macro. En pratique, la partie macro représente la meilleure partie affine des déplacements et efforts d'interface (cf. Figure 3.4).

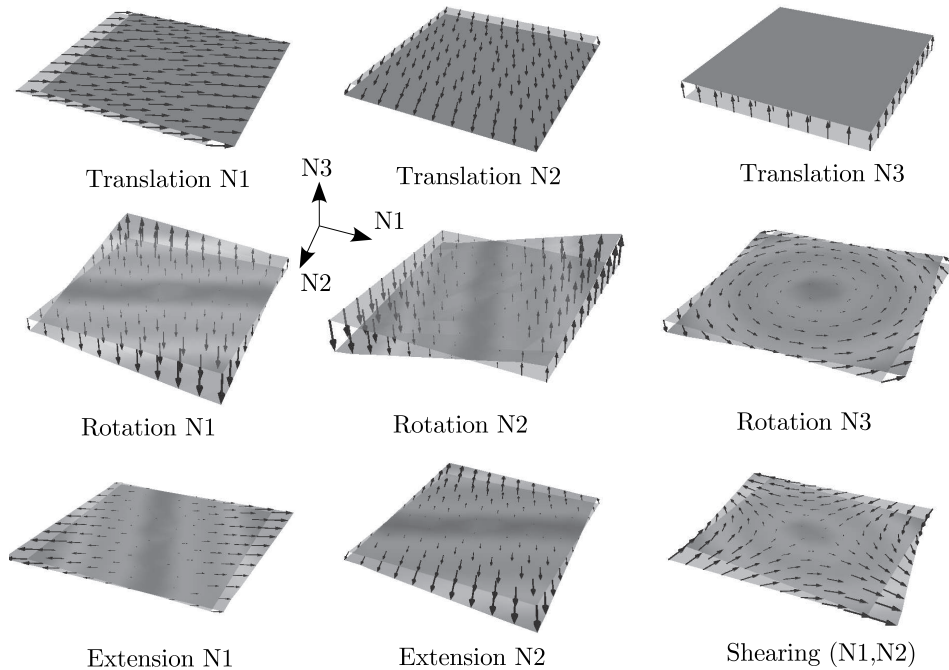


Figure 3.4: Fonctions de base macro sur une interface plane

2.1.2 Admissibilité des quantités macro

L'ajout des nouvelles inconnues (parties micro et macro) nécessite l'ajout d'équations supplémentaires. On choisit d'imposer la continuité des efforts macroscopiques à toutes les étapes de la résolution. Ce choix est justifié car il est valable pour tous les types d'interface. Dans le cas d'interfaces parfaites par exemple, il est possible d'imposer également la continuité des vitesses macroscopiques ce qui accélère fortement la convergence mais n'est valable que pour ces interfaces.

Les efforts macro sont donc recherchés dans l'espace \mathcal{F}_{ad}^M défini par :

$$\mathcal{F}_{ad}^M = \{ \underline{F}^M \in \mathcal{F}^M \mid \forall E \in \mathbf{E}, \forall E' \in \mathbf{V}_E, \quad \underline{F}_{EE'}^M + \underline{F}_{E'E}^M = 0 \} \quad (3.13)$$

où \mathbf{V}_E est l'ensemble des sous-structures voisines de la sous-structure E .

2.2 Modification de l'étape linéaire

La séparation micro-macro est réalisée uniquement au niveau de l'étape linéaire. L'introduction d'un multiplicateur de Lagrange permet de proposer un formalisme très proche de celui présenté en monoéchelle.

La prise en compte de l'admissibilité de \underline{F}^M ($\underline{F}^M \in \mathcal{F}_{ad}^M$) couple les équations (3.6) et (3.7) de toutes les interfaces. La direction de descente \mathbf{E}^- doit donc être écrite de manière globale pour être compatible avec la contrainte d'admissibilité imposée aux efforts macro. On définit donc la direction de descente sous la forme :

$$\forall \underline{F}_E^* \in \mathcal{F}^m \cup \mathcal{F}_{ad}^M, \quad \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E} \{ k^{-1} (\underline{F}_E - \widehat{\underline{F}}_E) + (\underline{\dot{W}}_E - \widehat{\underline{\dot{W}}}_E) \} \cdot \underline{F}_E^* dS = 0$$

Pour obtenir des problèmes linéaires restreints à chaque sous-structure et couplés uniquement par le problème macro, [Ladevèze et Nouy 2003] propose d'introduire un multiplicateur pour prendre en compte l'admissibilité des efforts macro. Cette écriture permet de séparer les problèmes linéaires et rend leur écriture plus simple mais ne change pas le caractère global de la direction de recherche.

On définit l'espace $\mathcal{W}_{ad,0}^M$ des vitesses macro continues aux interfaces et nulles sur les bords. On obtient alors les relations suivantes exprimant l'admissibilité des efforts macro et la direction de recherche globale :

$$\forall \underline{F}_E^* \in \mathcal{F}, \quad \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E} \{ k^{-1} (\underline{F}_E - \widehat{\underline{F}}_E) + (\underline{\dot{W}}_E - \widehat{\underline{\dot{W}}}_E) \} \cdot \underline{F}_E^* dS = \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E} \widetilde{\underline{\dot{W}}}_E^M \cdot \underline{F}_E^* dS \quad (3.14)$$

$$\forall \underline{\tilde{W}}_E^{M^*} \in \mathcal{W}_{ad,0}^M, \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E} \underline{\tilde{W}}_E^{M^*} \cdot \underline{F}_E dS = \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E \cap \partial\Omega_F} \underline{\tilde{W}}_E^{M^*} \cdot \underline{F}_d dS \quad (3.15)$$

Les directions de recherche peuvent ainsi être utilisées dans les problèmes par sous-structures. Par opposition au problème macro défini sur les interfaces et ne portant que sur les quantités macro, on appelle problèmes micro, les problèmes locaux définis par sous-structures.

Les problèmes à résoudre à l'étape linéaire sont alors donnés par :

– **Problème micro** pour toute sous-structure $E \in \mathbf{E}$:

Trouver $(\underline{u}_E, \underline{W}_E) \in \mathcal{U}_E \times \mathcal{W}_E$ qui vérifient :

$$\forall t \in [0, T], \forall \underline{u}^* \in \mathcal{U}_E^*, \int_{\Omega_E} \text{Tr}(\mathbf{K}\varepsilon(\underline{u}_E)\varepsilon(\underline{u}^*)) d\Omega + \int_{\partial\Omega_E} k^- \underline{W}_E \cdot \underline{u}^* dS = \int_{\Omega_E} \underline{f}_d \cdot \underline{u}^* d\Omega + \int_{\partial\Omega_E} (\widehat{\underline{F}}_E + k^- \widehat{\underline{W}}_E) \cdot \underline{u}^* dS + \int_{\partial\Omega_E} k^- \underline{W}_E^M \cdot \underline{u}^* dS \quad (3.16)$$

– **Problème macro** sur l'ensemble des interfaces :

Trouver $\underline{F}_E \in \mathcal{F}$ qui vérifie :

$$\forall \underline{\tilde{W}}_E^{M^*} \in \mathcal{W}_{ad,0}^M, \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E} \underline{\tilde{W}}_E^{M^*} \cdot \underline{F}_E dS = \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E \cap \partial\Omega_F} \underline{\tilde{W}}_E^{M^*} \cdot \underline{F}_d dS \quad (3.17)$$

Le problème microscopique (3.16) est un problème linéaire, et \mathbf{K} et k^- étant définis positifs, il admet une solution unique. D'après le principe de superposition, la solution de ce problème peut être décomposée en deux parties :

– $(\underline{u}_{E1}, \underline{W}_{E1})$ solution du problème micro (dénommé **problème micro 1**) dont le second membre est entièrement connu au début de l'étape linéaire :

Trouver $(\underline{u}_{E1}, \underline{W}_{E1}) \in \mathcal{U}_E \times \mathcal{W}_E$ qui vérifient :

$$\forall t \in [0, T], \forall \underline{u}^* \in \mathcal{U}_E^*, \int_{\Omega_E} \text{Tr}(\varepsilon(\underline{u}_{E1})\mathbf{K}\varepsilon(\underline{u}^*)) d\Omega + \int_{\partial\Omega_E} k^- \underline{W}_{E1} \cdot \underline{u}^* dS = \int_{\Omega_E} \underline{f}_d \cdot \underline{u}^* d\Omega + \int_{\partial\Omega_E} (\widehat{\underline{F}}_E + k^- \widehat{\underline{W}}_E) \cdot \underline{u}^* dS \quad (3.18)$$

- $(\underline{u}_{E2}, \underline{\dot{W}}_{E2})$ solution du problème micro (dénommé **problème micro 2**) dont le second membre n'est pas connu.

Trouver $(\underline{u}_{E2}, \underline{\dot{W}}_{E2}) \in \mathcal{U}_E \times \mathcal{W}_E$ qui vérifient :

$$\forall t \in [0, T], \forall \underline{\dot{u}}^* \in \mathcal{U}_E^*, \int_{\Omega_E} \text{Tr}(\boldsymbol{\varepsilon}(\underline{u}_{E2}) \mathbf{K} \boldsymbol{\varepsilon}(\underline{\dot{u}}^*)) d\Omega + \int_{\partial\Omega_E} k^- \underline{\dot{W}}_{E2} \cdot \underline{\dot{u}}^* dS = \int_{\partial\Omega_E} k^- \underline{\tilde{W}}_E^M \cdot \underline{\dot{u}}^* dS \quad (3.19)$$

En déterminant la vitesse sur les interfaces puis par la direction de recherche sur les efforts, on en déduit que la solution sur les interfaces entourant une sous-structure est telle que :

$$\underline{F}_E^M = \mathbf{L}_E^F(\underline{\tilde{W}}_E^M) + \underline{F}_{E,d}^M \quad (3.20)$$

$$\underline{\dot{W}}_E^M = \mathbf{L}_E^W(\underline{\tilde{W}}_E^M) + \underline{\dot{W}}_{E,d}^M \quad (3.21)$$

où $\underline{F}_{E,d}^M$ et $\underline{\dot{W}}_{E,d}^M$ ne dépendent que des données sur les sous-structures (efforts volumiques \underline{f}_d) et des quantités $\widehat{\mathbf{s}}_E$ aux interfaces. Ces quantités sont déterminées en résolvant le problème micro 1.

Remarque : Les problèmes micro 1 et micro 2 sont indépendants d'une sous-structure à l'autre et peuvent être résolus en parallèle. Seul le problème macro nécessite un assemblage.

2.2.1 Définition du comportement homogénéisé

\mathbf{L}_E^F (resp. \mathbf{L}_E^W) est un opérateur linéaire de \mathcal{W}_E^M dans \mathcal{F}_E^M (resp. de \mathcal{W}_E^M dans \mathcal{W}_E^M), symétrique et positif. Ces opérateurs peuvent être considérés comme des opérateurs homogénéisés sur la sous-structure Ω_E et sont déterminés avant toute résolution itérative.

Pour chaque sous-structure Ω_E , on impose successivement des valeurs nulles aux multiplicateurs macro pour toutes les interfaces sauf pour une composante macro d'une interface donnée. On résout ensuite le problème micro 2 soumis à la distribution de multiplicateurs $\underline{\tilde{W}}_E^M$ pour déterminer les vitesses $\underline{\dot{W}}_E^M$. Par la direction de recherche, on extrait les efforts macro \underline{F}_E^M correspondant. Chaque calcul fournit une colonne des opérateurs homogénéisés. Si n_M est la dimension de l'espace $\mathcal{W}_{EE'}^M$ pour une interface $\Gamma_{EE'}$ et que Ω_E est entourée de n_i interfaces, alors $n_M \times n_i$ calculs micro sont nécessaires pour déterminer ces opérateurs.

Remarque : La technique de construction des opérateurs ressemble à celle utilisée dans les méthodes d'homogénéisation cinématique et statique. Dans ces approches, on impose soit un champ de déformation, soit un champ de contrainte macroscopique, on résout le problème sur un volume élémentaire de référence donné, on en déduit ensuite les contraintes ou déformations résultantes et en extrayant la moyenne, on détermine le comportement macroscopique. La stratégie est différente ici car on travaille uniquement sur les interfaces et l'opérateur engendré est bien plus riche que ceux obtenus par homogénéisation cinématique ou statique.

La relation $\underline{F}_E^M = \mathbf{L}_E^F(\tilde{\underline{W}}_E^M) + \underline{F}_{E,d}^M$ est introduite dans le problème macroscopique (3.17). Ce problème est ensuite résolu de manière directe et permet d'extraire les multiplicateurs sur chaque interface. L'écriture sous forme faible de l'admissibilité statique (3.15) et la relation de comportement homogénéisé aboutissent au problème macro en multiplicateur suivant :

$$\begin{aligned} \text{Trouver } \tilde{\underline{W}}^M = \{ \tilde{\underline{W}}_E^M \}_{E \in \mathbf{E}} \in \mathcal{W}_{ad,0}^M \text{ qui vérifie :} \\ \forall \tilde{\underline{W}}_E^{M*} \in \mathcal{W}_{ad,0}^M, \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E} \tilde{\underline{W}}_E^{M*} \cdot \left(\mathbf{L}_E^F(\tilde{\underline{W}}_E^M) + \underline{F}_{E,d}^M \right) dS = \\ \sum_{E \in \mathbf{E}} \int_{\partial\Omega_E \cap \partial\Omega_F} \tilde{\underline{W}}_E^{M*} \cdot \underline{F}_d dS \quad (3.22) \end{aligned}$$

Ce problème admet une et une seule solution si $mes(\partial\Omega_U) \neq 0$.

Les multiplicateurs étant connus, le problème micro 2 peut être résolu à son tour pour déterminer les vitesses et efforts d'interface complets.

2.2.2 Algorithmes de résolution

Un résumé de la stratégie est proposé sous forme d'algorithme à la fois pour la méthode monoéchelle (Algorithme 1) et pour la méthode multiéchelle (Algorithme 2) en précisant les différentes étapes. On note \mathbb{K} la raideur obtenue par assemblage éléments finis, \mathbf{k} la matrice de direction de recherche obtenu par assemblage de celles-ci sur chaque interface, \mathbb{L} la matrice macro résultant de l'assemblage du problème macro. On note $[U^i]$ la solution aux problèmes micro 1 ou 2. On note N le nombre d'incrément de temps et dt le pas de temps.

On peut remarquer que la méthode monoéchelle est entièrement parallélisable contrairement à la méthode multiéchelle qui nécessite de résoudre un problème globale sur toutes les interfaces. Cependant, en pratique, pour les calculs d'assemblage qui nous intéressent, la taille du problème global est suffisamment faible pour rendre son temps résolution insignifiant par rapport à la résolution d'un seul des problèmes sur une sous-structure.

Algorithme 1 Stratégie de calcul monoéchelle (// : phase parallélisable)**Initialisation :** Assignment de \widehat{F} et \widehat{W} pour les interfaces extérieures**Itérations :****boucle** pour $iter = 0$ à $iter_{max}$ **faire**

1. Étape locale : Calcul de $\widehat{\mathbf{s}}_{iter+1/2} \in \Gamma$
// Résolution de problèmes locaux sur les interfaces $\Gamma_{EE'}, \forall n \in [0, N]$,
2. Étape linéaire : Calcul de $\mathbf{s}_{iter+1} \in \mathbf{A}_d$
// Résolution des problèmes linéaires :
boucle pour $n = 1$ à N **faire**
 $(\mathbb{K} + \frac{\mathbf{k}}{dt})[U_n] = [\widehat{F}_n + k\widehat{W}_n + k\frac{\widehat{W}_{n-1}}{dt}]$
fin boucle
// Relaxation
3. Indicateur d'erreur η : **Si** $\eta < \eta_c$ **Sortie**

fin boucle**Algorithme 2** Stratégie de calcul multiéchelle (// : phase parallélisable)**Initialisation :** Assignment de \widehat{F} et \widehat{W} pour les interfaces extérieures**Itérations :****boucle** pour $iter = 0$ à $iter_{max}$ **faire**

1. Étape locale : Calcul de $\widehat{\mathbf{s}}_{iter+1/2} \in \Gamma$
// Résolution de problèmes locaux sur les interfaces $\Gamma_{EE'}$,
2. Étape linéaire : Calcul de $\mathbf{s}_{iter+1} \in \mathbf{A}_d$
boucle pour $n = 1$ à N **faire**
// Résolution du problème micro 1 :
 $(\mathbb{K} + \frac{\mathbf{k}}{dt})[U_n^1] = [\widehat{F}_n + k\widehat{W}_n + k\frac{\widehat{W}_{n-1}}{dt}]$
// Construction du problème macro : $\widehat{F}_{E,d}^M$ par sous-structure + assemblage second membre macro $[\widetilde{F}_d]$
■ Résolution du problème macro $\mathbb{L}[\widetilde{W}] = [\widetilde{F}_d]$
// Résolution du problème micro 2 : $(\mathbb{K} + \frac{\mathbf{k}}{dt})[U_n^2] = [k\widetilde{W}]$
// Reconstruction des solutions : $[U_n] = [U_n^1] + [U_n^2]$
fin boucle // Relaxation
3. Indicateur d'erreur η : **Si** $\eta < \eta_c$ **Sortie**

fin boucle

3 Calcul des énergies imposée et dissipée

Le calcul de l'énergie imposée s'écrit comme l'intégrale sur le temps de la puissance injectée sur les différentes interfaces de conditions limites :

$$\sum_{\text{Interface CL}} \int_0^t \int_{\Gamma_{EE'}} \widehat{\underline{F}}_E \cdot \widehat{\underline{W}}_E \, dS d\tau \quad (3.23)$$

Le calcul de l'énergie dissipée cumulée sur le temps s'écrit comme l'intégrale sur le temps de la puissance dissipée sur les différentes surfaces de contact :

$$\sum_{\text{Interface de contact}} \int_0^t \int_{\Gamma_{EE'}} \widehat{\underline{F}}_E \cdot (\widehat{\underline{W}}_E - \widehat{\underline{W}}_{E'}) \, dS d\tau \quad (3.24)$$

Numériquement, ces calculs d'intégrale peuvent poser un problème s'ils sont mal effectués. Par exemple, si l'on prend un système masse-ressort où l'on injecte un effort sinusoïdale, la vitesse récupérée est en cosinus :

$$F = F_0 \sin(\omega_0 t)$$

$$\dot{W} = \frac{F_0 \omega_0}{k} \cos(\omega_0 t)$$

et le produit force-vitesse est en sinus de fréquence deux fois plus élevée :

$$F \cdot \dot{W} = \frac{F_0^2 \omega_0}{k} \sin(2\omega_0 t)$$

Or les schémas d'intégration numérique vont engendrer naturellement un déphasage entre l'effort et la vitesse, qui va conduire à ne plus obtenir la bonne solution lors de l'intégration si on ne fait pas attention.

La méthode retenue pour effectuer l'intégration est d'effectuer le produit force-vitesse en prenant la force au demi pas de temps. Le résultat est défini par la formule suivante :

$$\mathcal{P}(t = nh) = \sum_{i=1}^n \dot{W}_i \frac{F_i + F_{i-1}}{2} dt \quad (3.25)$$

4 Implantation numérique

L'implantation numérique de la méthode a été réalisée dans le cadre d'une nouvelle politique logicielle du LMT-Cachan basée sur une plateforme modulaire maintenue par un ingénieur de recherche H. Leclerc. Les différentes bibliothèques de calcul

scientifique permettent de travailler avec une syntaxe proche de celle de Matlab pour ce qui est du calcul matricielle tout en ayant toute la puissance du C++. Le logiciel, nommé JointVirt, a ainsi été développé en collaboration avec D. Violeau, H. Leclerc et moi même et il sert maintenant de base à plusieurs thèses et post-doc sur l'extension à la dynamique de la LATIN multiéchelle, l'implantation des modèles d'endommagement dans les composites, l'interaction fluide-structure, l'erreur en relation de comportement...

Le code a été développé uniquement pour effectuer la mise en données nécessaire à la méthode, le solveur, et la sortie des résultats. Aussi, il est pour l'instant nécessaire d'effectuer le pré-traitement (création des maillages) dans le logiciel de son choix. Le code va ensuite venir lire les maillages pour faire la mise en données (trouver les interfaces, les voisins, contruire les opérateurs...), effectuer le calcul et créer des fichiers résultats lisible par le logiciel paraview, qui permet notamment d'effectuer la lecture de très gros modèles en parallèle.

5 Premiers exemples simples

5.1 Contact entre deux poutres

L'objet de ce premier exemple simple est d'illustrer le gain apporté par le multi-échelle pour un problème de contact avec frottement. Le problème à résoudre est celui de deux poutres de section carrée posées l'une sur l'autre, découpées chacune en huit sous-structures. On impose des conditions de symétrie sur les plans (xOy), (xOz) et (yOz) pour bloquer les mouvements de corps rigide. On applique sur la partie supérieure un effort de 30 MPa tout en appliquant sur l'extrémité restante avec un effort de 100 MPa (cf. Figure 3.5).

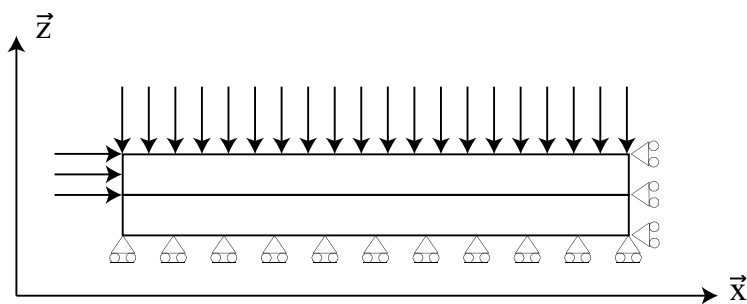


Figure 3.5: Modélisation des deux poutres en contact

Cet exemple académique permet de bien se rendre compte du gain engendré par le multi-échelle grâce à la propagation rapide de l'information macro. Sur la

Figure 3.6, on montre le gain en terme de nombre d'itérations : on passe de 386 itérations nécessaires à la convergence (erreur LATIN de 10^{-5}) en monoéchelle à seulement 19 en multiéchelle, soit un gain d'un facteur 20. Ce gain se traduit en terme de temps de calcul par un gain d'un facteur 10, en effet en multiéchelle on résout à chaque itération, deux fois les systèmes linéaires associés à chacune des sous-structures.

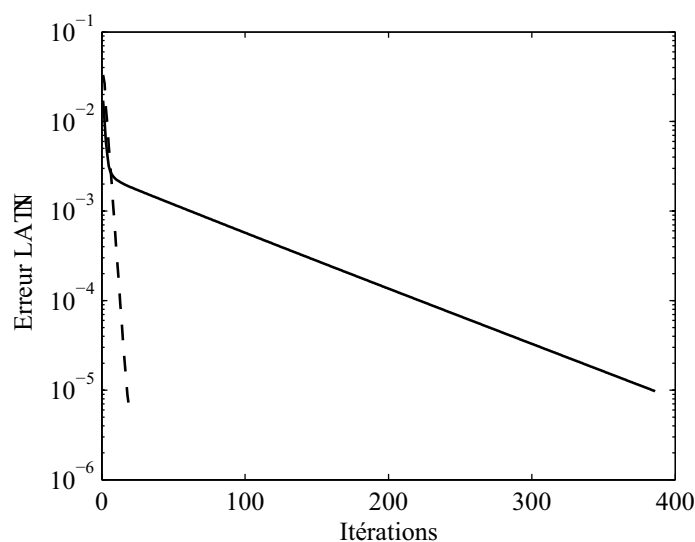


Figure 3.6: Gain apporté par le multiéchelle

Les différents gains en terme de mémoire, temps CPU sont synthétisés dans le tableau 3.1.

	Abaqus	Cofast3d	JointVirt
Mémoire (Mo)	153	38	42
Nombre d'incrément	21	21	21
temps CPU (s)	10	60	6

Tableau 3.1: Comparaison Abaqus convergé - Cofast3d - JointVirt

5.2 Retour sur la résolution du problème type

On revient à présent sur le calcul de la liaison générique présentée dans le chapitre précédent dont les conditions limites sont sur la Figure 2.3. Sur la Figure 3.7,

on compare le déplacement des points P1 et P2 obtenu par Abaqus, Cofast3d et JointVirt et on peut constater que les résultats sont similaires. De même, les énergies calculées par les différentes méthodes (Figure 3.8) sont tout à fait comparables.

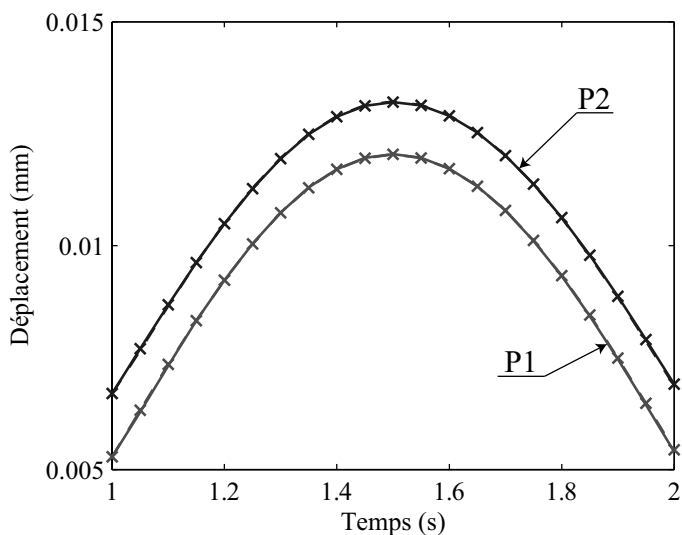


Figure 3.7: Déplacements des points P1 et P2 (résultat Abaqus coefficient de pénalisation optimal (-x-), résultat Cofast3d (—) et résultat JointVirt (-.-.-))

En terme de temps de calcul (tableau 3.2), cette fois le gain est très important. La LATIN dans sa version multiéchelle permet de générer la solution 18 fois plus rapidement qu'Abaqus. L'apport du multiéchelle dans ce cas permet d'avoir un gain de 4,7.

	Abaqus	Cofast3d	JointVirt
Mémoire (Mo)	256	75	80
Nombre d'incréments	31	21	21
temps CPU (s)	676	175	37

Tableau 3.2: Comparaison Abaqus convergé - Cofast3d - JointVirt

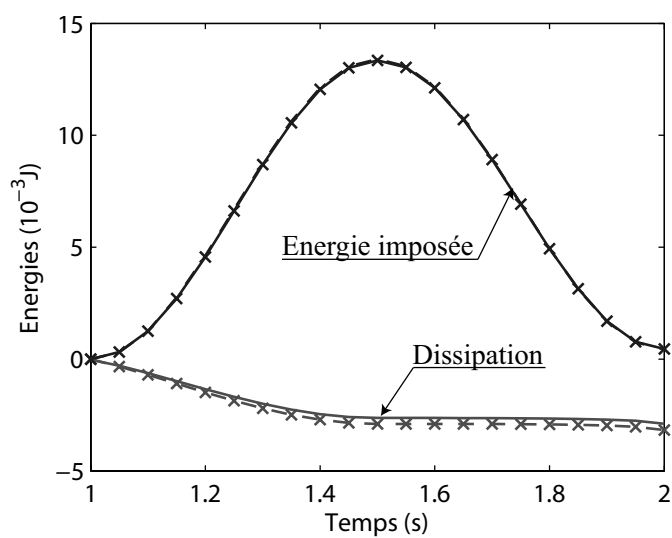


Figure 3.8: Énergies dissipée et imposée (résultat Abaqus coefficient de pénalisation optimal (-x-), résultat Cofast3d (—) et résultat JointVirt (-.-.-))

Prédiction de l'amortissement dans les liaisons

Dans ce chapitre, on applique la méthode présentée au chapitre précédent au calcul de la liaison SSS en flexion et on compare les résultats numériques aux résultats expérimentaux présentés dans le chapitre 1. La matrice d'amortissement relative à toutes les sollicitations est calculée et la méthode est appliquée à d'autres liaisons types du lanceur.

1 Vérification de l'approche : simulation de l'essai de flexion de la liaison SSS

1.1 Modélisation de la liaison SSS

L'objectif principal de cette partie est de calculer la liaison SSS en flexion pure avec le logiciel JointVirt. Le maillage complet de la liaison, qui contient 1,2 million de degrés de liberté, est représenté sur la Figure 4.1(a). Pour diminuer le coût de calcul, un maillage simplifié qui prend en compte les symétries est utilisé. Ce nouveau maillage, qui contient 120 000 degrés de liberté, est représenté sur la Figure 4.1(b).

La simulation de ce problème est très coûteuse car le modèle compte 60 interfaces, dont 16 de contact avec frottement, et 21 sous-structures. Les boulons des

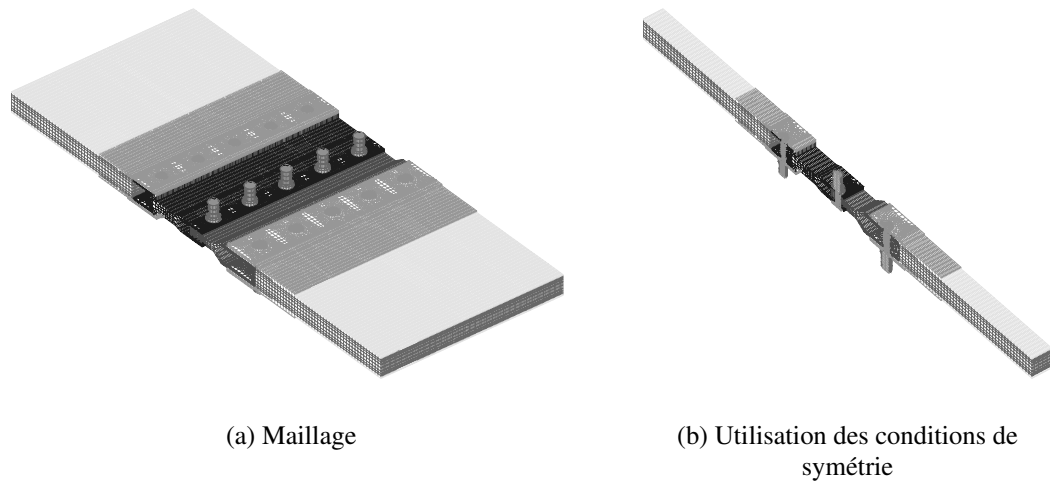


Figure 4.1: Modélisation de la liaison SSS

extrémités sont collés (voir Figure 4.2 et Figure 4.3). Toutes les interfaces entre les différentes pièces sont alors considérées comme parfaites excepté dans la partie centrale, où les interfaces sont de type contact avec frottement. Ce type d'interfaces est également présent au niveau de toutes les têtes de vis, entre les écrous et les différentes plaques et également au niveau des tiges de vis dans les trous des plaques.

Une des extrémités de la liaison est encadrée et sur l'autre est appliqué un chargement permettant d'imposer de la flexion pure. Ce moment de flexion suit l'évolution suivante : de 0 à 1 s, il est nul, ce qui permet de mettre la précharge dans les boulons, et de 1 à 7 s, il suit une évolution en triple sinus (voir Figure 4.2), ce qui permet d'obtenir trois valeurs d'amortissement (une par période du sinus). La discrétisation temporelle retenue est suffisamment fine pour obtenir un résultat convergé comme on le montrera en 1.2.4. La précharge dans les boulons est découpée en 101 pas de temps et chaque sinus en 400 pas de temps pour un total de 1301 pas de temps.

Le calcul sera mené pour différentes amplitudes du sinus, que l'on peut faire correspondre aux différentes valeurs de la déformation des peaux du composite mesurées lors de la campagne d'essai expérimental. On cherche à déterminer l'amortissement pour une déformation variant de 0 à $4 \cdot 10^{-4}$.

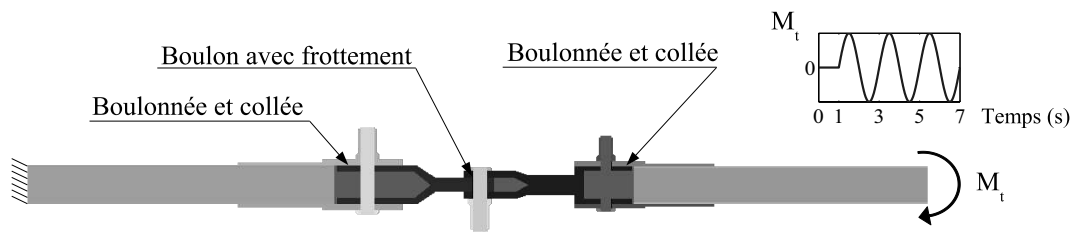


Figure 4.2: Conditions limites

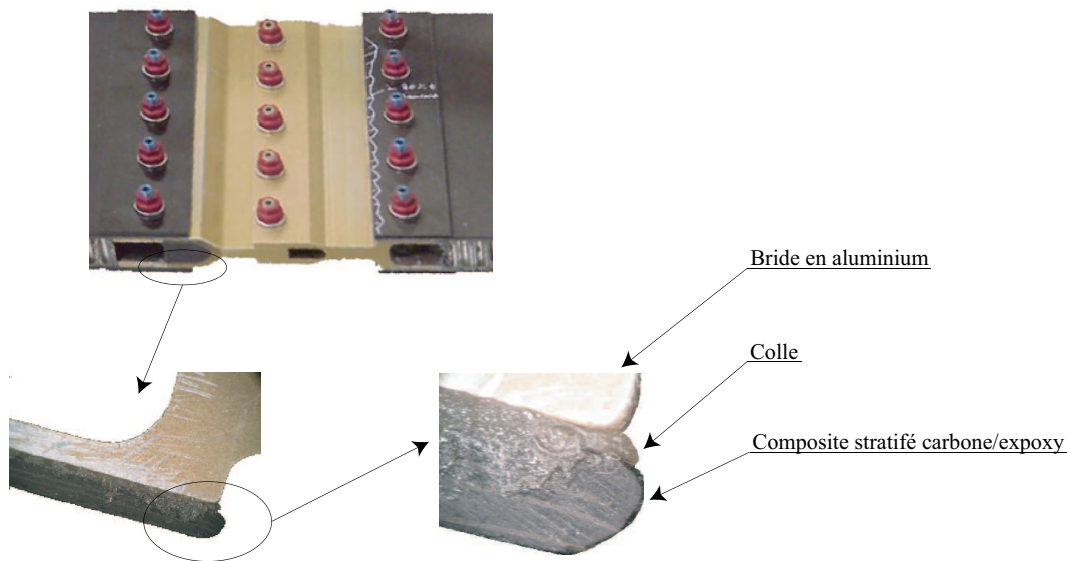


Figure 4.3: Interface entre le composite et l'aluminium

1.2 Vérification du calcul

1.2.1 Détermination du coefficient d'amortissement

On peut proposer deux manières de déterminer le coefficient d'amortissement η . Par définition, η est égal à $\eta = \frac{1}{2\pi} \frac{\Delta U}{U}$ avec U l'énergie imposée et ΔU l'énergie dissipée par frottement durant le cycle.

La première méthode consiste à calculer les énergies imposées et dissipées comme étant des intégrales de surface des puissances imposées et des puissances

dissipées :

$$\eta = \frac{1}{2\pi} \frac{\Delta U}{U} = \frac{1}{2\pi} \frac{\sum_{\text{interface contact}} \int_{[t_1, t_1+T] \times S_c^i} \hat{F}_E \cdot (\hat{W}_{E'} - \hat{W}_E) dS dt}{\max_{\tau \in [0, T]} \sum_{\text{interface effort imposé}} \int_{[t_1, t_1+\tau] \times S_i^j} \hat{F}_E \cdot \hat{W}_E dS dt}$$

avec t_1 l'instant correspondant au début d'un cycle, T la période du cycle, S_c^i les surfaces de contact et S_i^j les surfaces à effort imposé.

La seconde méthode consiste à tracer sur un cycle, le moment appliqué en fonction de l'angle de la section droite où les efforts sont imposés. Cette courbe est une ellipse, d'autant plus large que la liaison dissipe, dont l'aire correspond à l'énergie dissipée sur le cycle considéré. L'énergie imposée est égale à l'aire du « triangle » sous l'ellipse (voir la Figure 4.4).

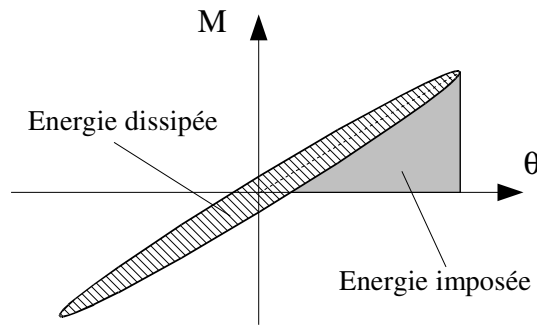


Figure 4.4: Énergie dissipée et énergie imposée

1.2.2 Influence du niveau du critère d'arrêt

Le paramètre qui permet de stopper l'algorithme quand on a atteint une solution correcte est l'erreur LATIN. Pour étudier l'influence de ce critère, différents calculs de la liaison SSS sont lancés pour une valeur de déformation de $3 \cdot 10^{-4}$, avec des niveaux d'erreur à atteindre différents.

La Figure 4.5 montre l'évolution de l'énergie imposée et de l'énergie dissipée calculées sur la 3^e période du sinus avec les deux méthodes présentées en 1.2.1 en fonction du niveau de l'erreur LATIN atteint. On constate que l'énergie imposée converge plus rapidement avec la seconde méthode alors que l'énergie dissipée converge plus rapidement avec la première.

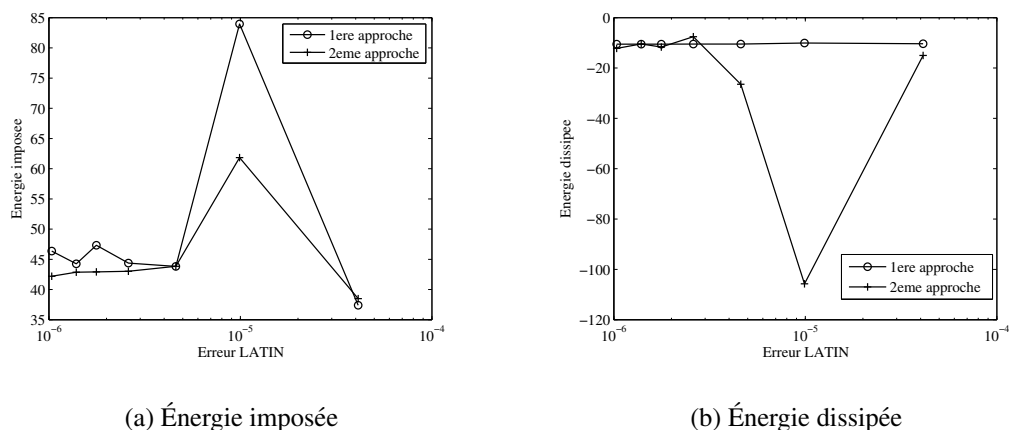


Figure 4.5: Caculs des énergies avec les deux approches en fonction du niveau de l'erreur LATIN

Le but étant de calculer le plus rapidement possible, on va chercher à diminuer le nombre d'itérations de la méthode pour obtenir un résultat correct en terme d'amortissement. Compte tenu de l'influence du niveau de l'erreur LATIN sur les résultats des calculs d'énergies imposée et dissipée par les deux méthodes, l'amortissement va être calculé à l'aide d'une méthode mixte. L'énergie imposée sera calculée à partir de l'ellipse du moment fléchissant en fonction de la rotation et l'énergie dissipée sera calculée à partir du calcul des intégrales de la puissance dissipée sur les surfaces de contact.

D'autre part, on choisit à présent de prendre un niveau d'erreur LATIN à atteindre de $2,5 \cdot 10^{-6}$ pour effectuer tous les calculs, ce critère donnant une valeur stabilisée de l'énergie imposée et de l'énergie dissipée d'après la Figure 4.5.

1.2.3 Influence du numéro du cycle sur le calcul de l'amortissement

Un autre paramètre du calcul de l'amortissement est de savoir sur quel cycle le calculer. Pour cela, un calcul de la liaison SSS a été effectué pour une valeur de déformation de $3 \cdot 10^{-4}$, avec une évolution de 6 périodes sinusoïdales sur l'effort imposé. La Figure 4.6 montre le coefficient d'amortissement en fonction du numéro du cycle sur lequel il est calculé. On peut en conclure qu'à partir du 3^e cycle la valeur de l'amortissement est stabilisée, c'est la raison pour laquelle tous les calculs d'amortissement seront à présent effectués sur le 3^e cycle.

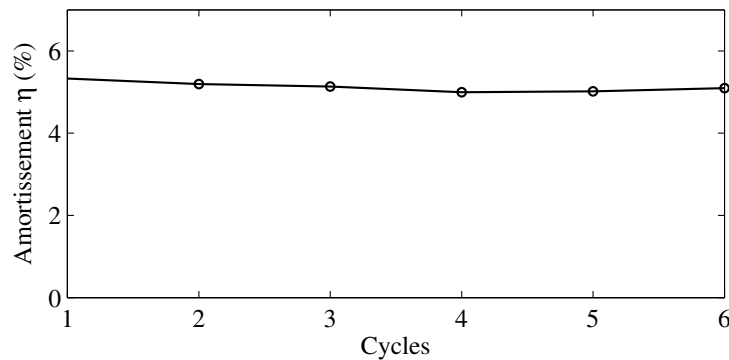


Figure 4.6: Influence du numéro du cycle sur lequel l'amortissement est calculé

1.2.4 Influence de la discrétisation temporelle

La discrétisation temporelle des sinus est un paramètre qu'il est également important d'analyser. Pour cela, plusieurs calculs de la liaison SSS sont lancés pour une valeur de déformation de $3 \cdot 10^{-4}$, ceci pour différentes discrétisations temporelles. Chaque sinus est discrétisé en 100, 200, 250, 400, 500, 800 pas de temps.

La Figure 4.7 montre l'évolution de l'amortissement calculé sur le 3^e cycle en fonction du nombre de pas de temps par sinus. On constate qu'à partir d'une discrétisation de 250 pas de temps par sinus, la solution est stabilisée. Pour la suite des calculs, on choisit de prendre 400 pas de temps par sinus.

1.2.5 Influence de la discrétisation spatiale

Le dernier paramètre de calcul à étudier est la taille de maille utilisée pour la discrétisation spatiale. Sur la Figure 4.1, la taille des éléments du maillage des surfaces de contact de la liaison SSS était de l'ordre 1 mm. Un nouveau calcul, pour une déformation de $3 \cdot 10^{-4}$, est effectué avec des tailles de maille de l'ordre de 0,5 mm ce qui conduit à un problème à 320 000 degrés de liberté, puis un second avec des tailles de maille de l'ordre de 0,3 mm, ce qui conduit à un problème à 580 000 degrés de liberté. Les maillages pour les différentes tailles de maille d'une interface sont proposés sur la Figure 4.8.

Le tableau 4.1 résume les différents calculs menés ainsi que les résultats. On constate que les résultats en terme d'amortissement pour les deux discrétisations spatiales sont tout à fait similaires et conduisent à un écart de 1,4%, ce qui étant donnée la précision recherchée dans le calcul de l'amortissement, est très faible. En revanche, le calcul avec le maillage grossier (taille de maille de 1 mm) conduit à un écart de 18% ce qui est trop élevé. Les calculs seront désormais réalisés avec des tailles de maille d'environ 0,5 mm.

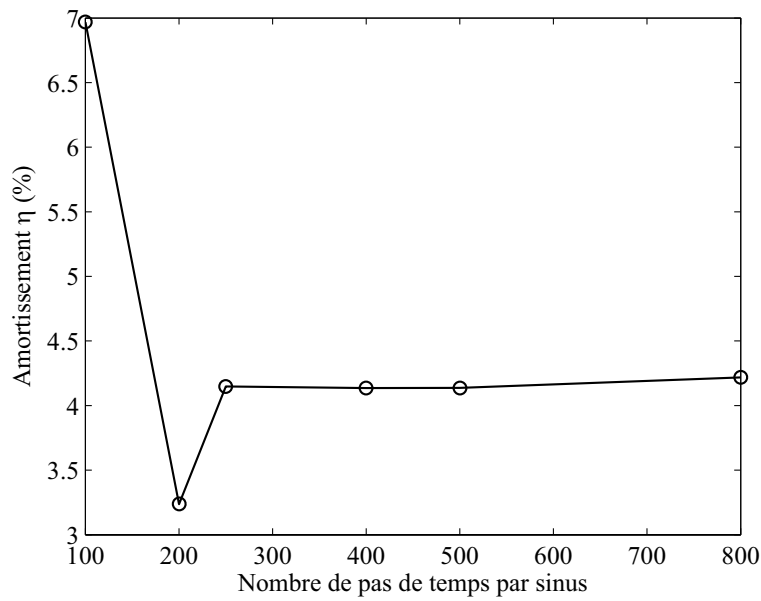


Figure 4.7: Influence de la discrétisation temporelle sur le calcul de l'amortissement

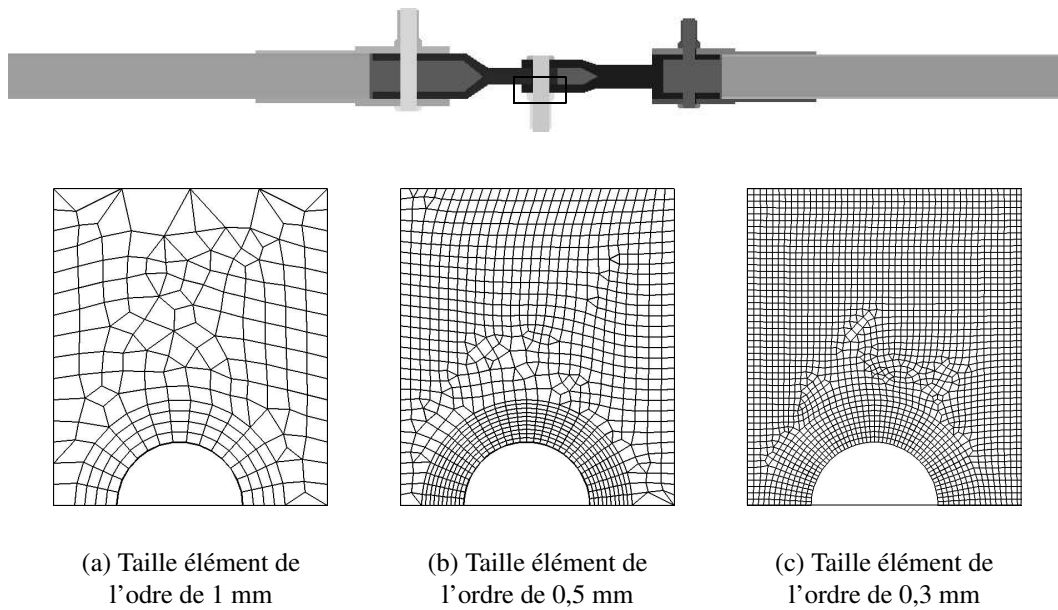


Figure 4.8: Différentes discrétisations spatiales étudiées

Taille de maille	Degrés de liberté	Temps de calcul	Amortissement η	Erreur
1 mm	120000	6,2h	4,13%	18%
0,5 mm	320000	45,9h	5,03%	référence
0,3 mm	580000	92,8h	5,01%	1,4%

Tableau 4.1: Influence de la discrétisation spatiale

2 Validation de la méthode par comparaison avec les essais expérimentaux

2.1 Coefficient de frottement

Dans un assemblage, les surfaces en contact s'usent au cours des cycles, ce qui entraîne la modification des surfaces en contact, et fait que le coefficient de frottement ne peut pas être constant.

Dans le cas de la liaison SSS, les matériaux en contact sont en alliage d'aluminium recouvert d'une couche de protection anti-corrosion en Alodine 1200. Cette protection est une chromatisation de la surface d'aluminium dont le procédé chimique est commercialisé par la société Henkel. La rugosité R_a , mesurée sur les pièces alodinisées, est de l'ordre de 0,2 μm .

Le couple chrome-chrome est couramment utilisé pour son très faible coefficient de frottement. La valeur retenue est classiquement 0,12 (adhérence et frottement). La société Henkel (site de Boulogne-Billancourt) a été contactée pour connaître plus précisément le coefficient de frottement du couple Alodine-Alodine [Julvez 2007]. Elle nous a mis en relation avec les chimistes du site américain qui nous ont annoncé une valeur variant entre 0,1 et 0,2. Pour les applications classiques, la valeur retenue est proche de 0,1 (mais peut être inférieure si adjonction de cire synthétique dans la formule). Les valeurs hautes sont obtenues par adjonction de silice pour des cas spécifiques (notamment pour des toits sur lesquels on veut pouvoir marcher sans glisser).

L'usure des surfaces de contact va se traduire par une usure de cette couche de protection. Ainsi, le coefficient de frottement va varier avec le nombre de cycles. On choisit donc de faire les calculs avec des coefficients de frottement compris entre 0,08 et 0,14 environ.

L'amortissement, calculé sur le 3^e et dernier cycle de chargement pour différentes valeurs maximales de moment, est présenté sur la Figure 4.9 pour différentes valeurs de coefficient de frottement (0,08, 0,10, 0,12 et 0,14).

Le temps de calcul est de 15 jours pour générer la courbe complète sur sept processeurs d'un cluster d'OpteronTM.

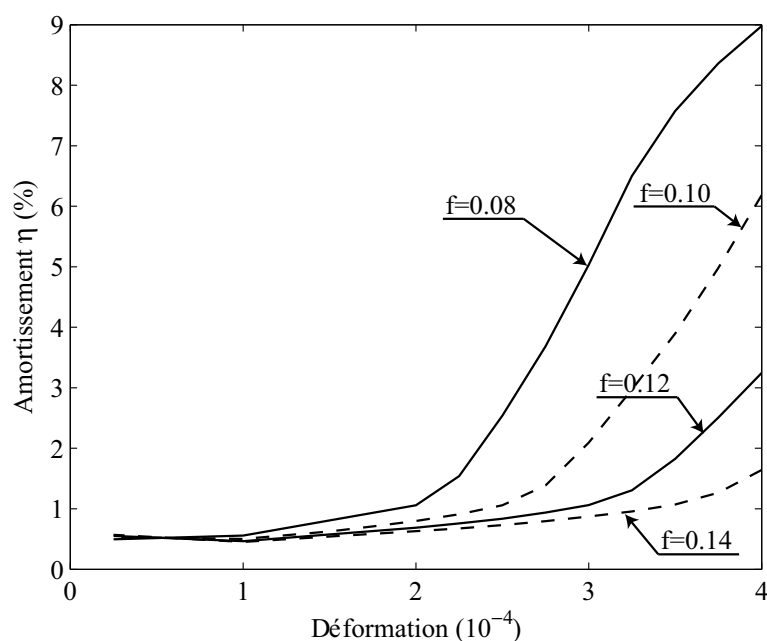


Figure 4.9: Amortissement de la liaison SSS pour différents coefficients de frottement

On observe que lorsque le coefficient d'amortissement augmente, l'amortissement diminue. D'une manière générale, il est difficile de prévoir l'influence du coefficient de frottement sur l'amortissement, car il dépend du produit de l'effort tangentiel et de la vitesse tangentielle. Or quand l'effort normal est élevé, l'effort tangentiel l'est aussi, cependant la vitesse est plus faible. Quand l'effort normal est faible, l'effort tangentiel l'est aussi, mais cette fois la vitesse peut être plus élevée.

La forme des résultats numériques est très proche des résultats expérimentaux comme le montre la Figure 4.10. Le modèle de Coulomb est donc suffisant pour modéliser le contact dans cette structure, la seule difficulté étant de le caractériser. L'évolution précise du coefficient de frottement en fonction des cycles doit être donnée par des essais tribologiques préalables à l'étude numérique. La connaissance de cette évolution permet la connaissance complète de l'amortissement dans la liaison. Dans ce cas précis, étant donné que tous les résultats d'essais réalisés dans l'étude expérimentale sont disponibles, le coefficient de frottement peut être identifié.

2.2 Identification du coefficient de frottement

On se propose donc d'identifier le coefficient de frottement à partir des différents résultats expérimentaux obtenus lors de la campagne d'essais expérimentaux. Pour

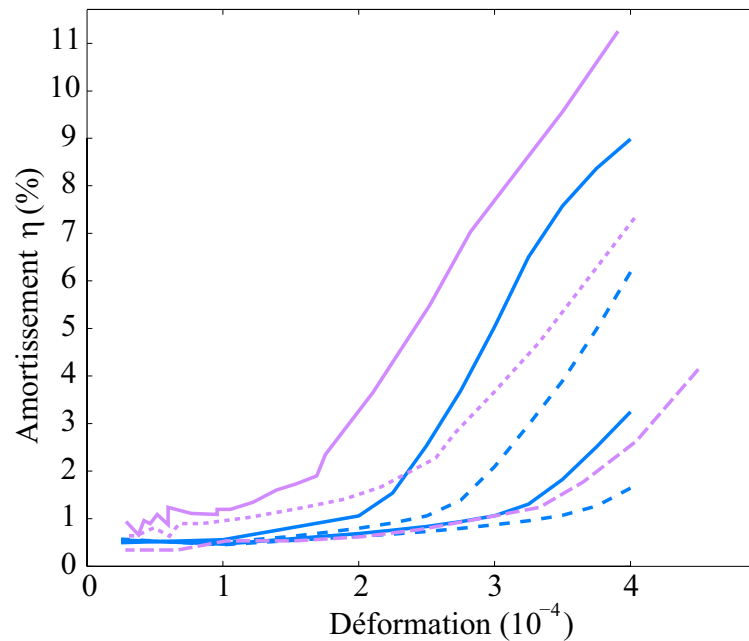
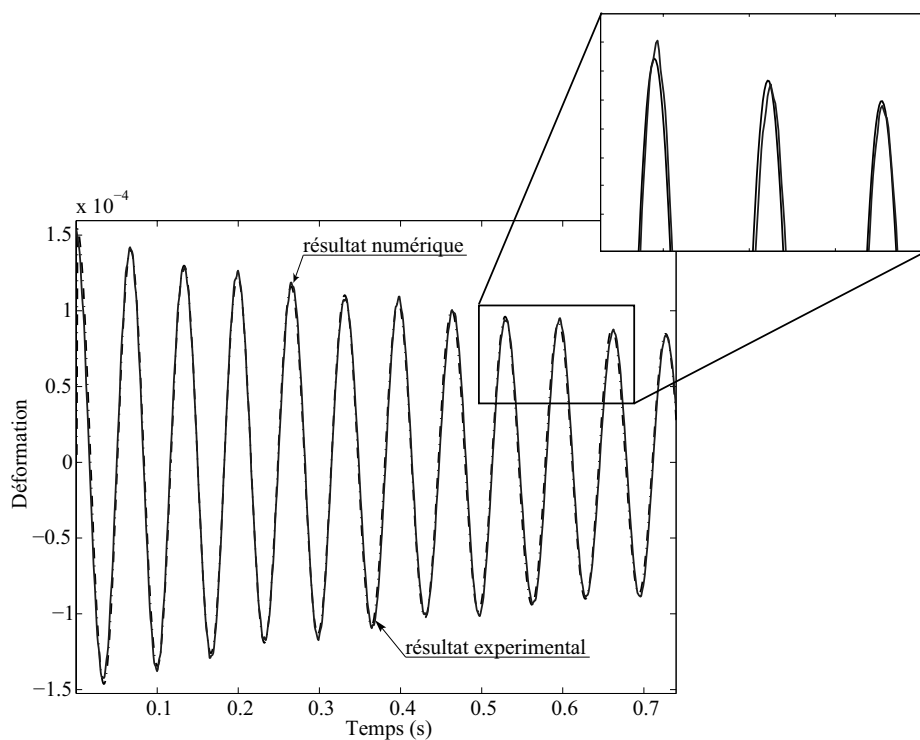


Figure 4.10: Comparaison des résultats numériques avec les résultats expérimentaux (rose = résultats numériques, bleu = résultats expérimentaux)

cela, on va simuler la réponse de chaque essai de lâché. Cette réponse dépend de l'évolution du coefficient d'amortissement en fonction de la déformation maximale sur le cycle. Cette évolution est obtenue en fixant une valeur du coefficient de frottement à partir d'une interpolation des résultats numériques de la Figure 4.9. Il suffit ensuite de trouver la meilleure valeur du coefficient de frottement permettant de minimiser l'écart entre les résultats de l'essai de lâché expérimental et ceux de l'essai simulé. On trace sur la Figure 2.2 la déformation de la peau du composite de la liaison SSS issue des essais expérimentaux et des essais numériques en fonction du temps. On observe que pour le coefficient de frottement optimal, les courbes simulées concordent tout à fait avec les résultats d'essais expérimentaux.

Les différentes valeurs de coefficient de frottement ainsi obtenues sont présentées sur la Figure 4.11 pour l'ensemble des résultats d'essais obtenus lors des tests sur deux éprouvettes SSS. L'abscisse (nombre de cycle) correspond aux cycles où le niveau a dépassé les $0,5 \cdot 10^{-4}$ en déformation (l'allure reste identique pour des coupures à des niveaux différents). Les cycles à très basse déformation sont supposés ne pas roder les éprouvettes car il n'y a pas assez d'énergie en jeu. On constate que le coefficient de frottement ainsi identifié varie entre 0,08 et 0,12 (un seul point pour 0,14), ce qui représente tout à fait l'intervalle de valeurs attendu et corrobore



les données du fabricant concernant le traitement de surface.

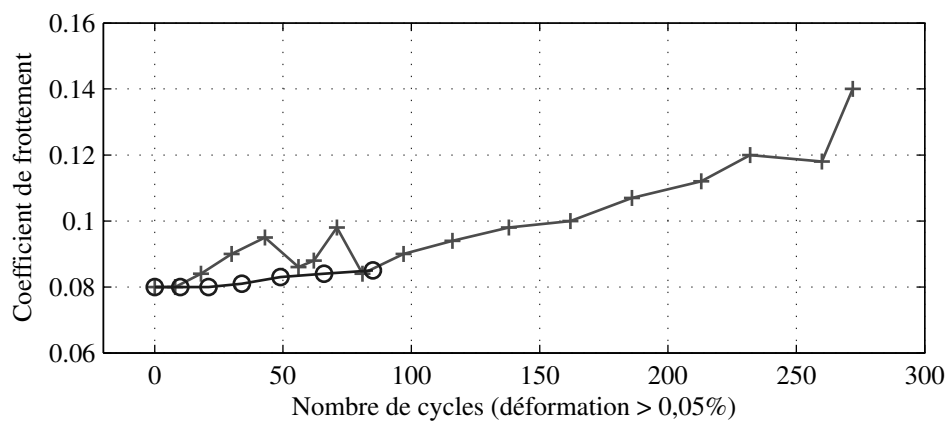


Figure 4.11: Identification du coefficient de frottement en fonction du nombre de cycles sur les deux éprouvettes testées expérimentalement ((+) liaison SSS 1,(o) liaison SSS 2)

3 Résultats et influence des paramètres physiques du modèle

Dans le cadre de la détermination de l'amortissement, il est nécessaire d'effectuer un certain nombre de calcul permettant de caractériser l'influence des différents paramètres physiques constituant la liaison. La plupart des valeurs prises par les différents paramètres physiques des liaisons (coefficients matériau, géométrie, chargement, jeu, serrage...) ont été consignées dans le cahier des charges confidentiel associé à cette étude [Le Gallo 2005].

3.1 Carte de glissement et de dissipation

La Figure 4.12, à droite, montre les zones de glissement à l'instant $t = 2,5$ s. On constate que les zones de glissement important se situent sur les bords des plaques, ce qui s'explique naturellement par le fait que les boulons sont serrés très fort, provoquant ainsi des décollements dans ces zones. Le chargement de flexion va avoir ensuite tendance à ouvrir et fermer alternativement ces zones en les faisant glisser. La valeur du glissement varie entre 0,02 et 0,1 mm ce qui est suffisamment important pour qu'il soit raisonnable d'utiliser le modèle de Coulomb.

La Figure 4.12 de gauche montre les zones de dissipation. Les niveaux les plus élevés ne correspondent pas forcément aux zones de glissement maximal car c'est le produit effort tangentiel-déplacement relatif qui influe.

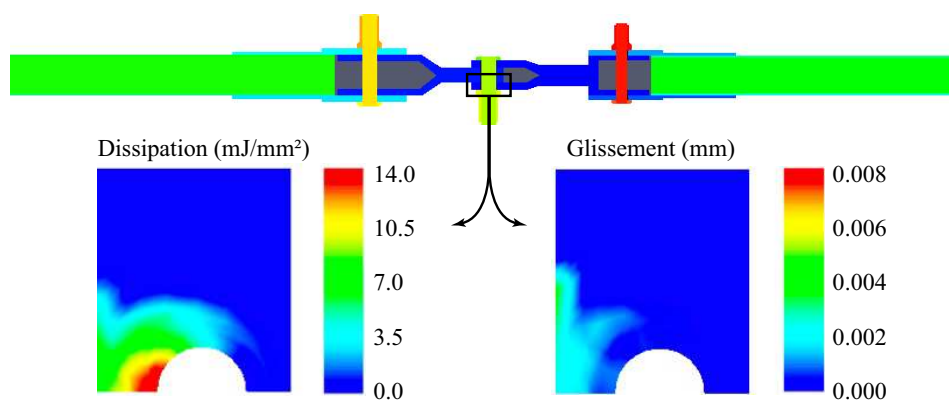


Figure 4.12: Carte de dissipation et de glissement

3.2 Influence des précharges dans les boulons

La précharge dans les différents boulons est, elle aussi, un paramètre essentiel pour la détermination de l'amortissement. Le serrage est déterminé initialement par extensométrie mais il peut évoluer au cours du temps en fonction du chargement. La Figure 4.13 représente l'évolution de l'amortissement pour une précharge de 80% et 90% de la précharge initiale. On constate que, dans cette gamme de serrage, l'amortissement augmente quand le serrage diminue.

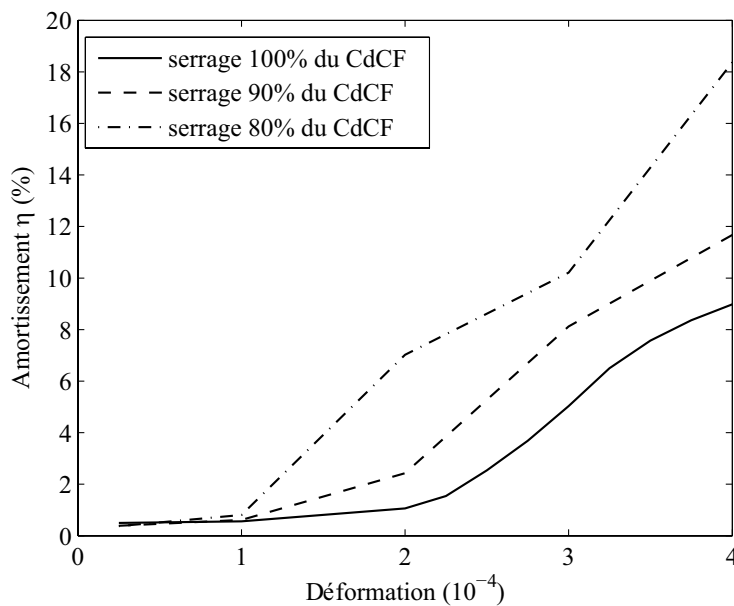


Figure 4.13: Influence de la précharge des boulons

4 Autres sollicitations

Précédemment, on s'est intéressé uniquement à la simulation de l'essai de flexion quatre points. On a montré qu'il était possible de calculer l'amortissement pour une liaison complexe. On s'intéresse maintenant à la connaissance de l'amortissement pour d'autres sollicitations simples.

La mise en forme de la base de données peut être réalisée avec une vision matricielle dont les lignes et colonnes correspondront aux sollicitations de base du modèle plaque utilisé. Les termes de la matrice, notée \mathcal{D} , représenteront la dissipation pour chaque sollicitation. La diagonale comportera les termes relatifs aux

sollicitations simples, et les termes extradiagonaux feront apparaître les dissipations relatives aux sollicitations couplées. Dans un premier temps, on suppose le découplage des sollicitations, aussi la matrice sera \mathcal{D} diagonale :

$$\mathcal{D} = \begin{bmatrix} d_{\text{traction}} & 0 & 0 & 0 \\ 0 & d_{\text{effort tranchant}} & 0 & 0 \\ 0 & 0 & d_{\text{cisaillement}} & 0 \\ 0 & 0 & 0 & d_{\text{moment flexion}} \end{bmatrix}$$

La matrice \mathcal{D} dépend naturellement de nombreux paramètres tels que les coefficients de frottement des différentes surfaces en contact, les serrages des différents boulons, le niveau des excitations. La dissipation totale, notée $\Delta U_{\text{liaison}}$, due à la liaison est obtenue par la formule :

$$\Delta U_{\text{liaison}} = T^T \mathcal{D} T \quad \text{avec} \quad T^T = [N \quad T_z \quad T_y \quad M_t]$$

où N est l'effort de traction, T_z est l'effort tranchant radial, T_y est l'effort tranchant transverse et M_t est le moment fléchissant.

La dissipation due à un effort tranchant conduit à solliciter en flexion la liaison, c'est pourquoi elle peut être déduite de l'étude précédente.

La traction est une sollicitation de base que l'on retrouve lors de l'analyse modale de la structure pour des modes longitudinaux. Le même genre d'étude que pour la flexion a été réalisée, c'est à dire un calcul de l'amortissement de la liaison pour un coefficient de frottement moyen de 0,08, et ce, pour des niveaux d'efforts imposés par le cahier des charges du lanceur. Il en résulte que l'amortissement de la liaison est de l'ordre de 0,3%, c'est-à-dire très faible pour ces niveaux d'efforts.

Le cisaillement transverse est également une sollicitation de base que l'on retrouve lors de l'analyse modale de la structure pour les modes de torsion. L'amortissement a également été calculé pour un coefficient de frottement moyen de 0,08, pour des niveaux d'efforts imposés par le cahier des charges du lanceur. On constate que l'amortissement de la liaison est encore plus faible ; il est inférieur à 0,1%.

Le calcul de l'amortissement pour cette sollicitation a nécessité de prendre en compte des conditions de périodicité sur les flancs et donc de supprimer la condition de symétrie (Figure 4.14). Le problème à résoudre présente ainsi plus de 640000 degrés de liberté.

La difficulté est de relier les dissipations tabulées dans la base de données (et donc obtenues par des calculs de liaisons avec des conditions limites particulières) au calcul structural du lanceur. Pour cela, la méthodologie proposée est la suivante. On effectue un calcul de dynamique sur une modélisation éléments finis en élément coque assez grossière du lanceur. Le calcul est effectué avec un amortissement for-

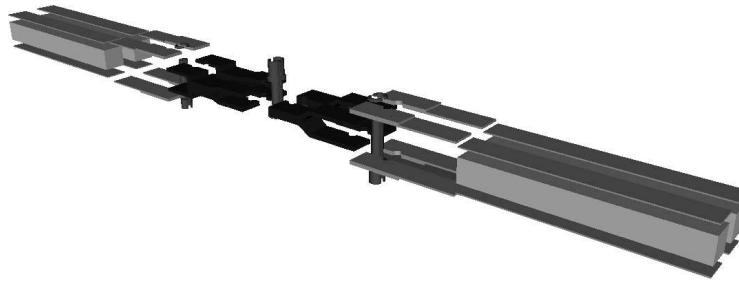


Figure 4.14: Décomposition de domaine de la liaison SSS sans conditions de symétrie

faitaire, le but étant simplement de déterminer les niveaux des efforts aux niveaux des liaisons.

On s'intéresse à l'élément de coque qui « contient » la liaison et on récupère dans les résultats du calcul les éléments du tenseur des actions mécaniques extérieures sur cet élément en chaque nœud (moment fléchissant, effort longitudinal, effort tranchant...). En pratique on découple les différentes sollicitations, donc on cherche à déterminer l'énergie dissipée pour chacune des sollicitations élémentaires.

Par exemple, pour une sollicitation de type flexion (simple et/ou composée), le rayon du lanceur étant suffisamment grand devant la taille de l'élément coque, on peut adopter une modélisation poutre dont les conditions limites sont données sur Figure 4.15(a). On détermine ainsi qu'au centre de l'élément en O, le moment fléchissant vaut $M_O = \frac{M_B - M_A}{2}$ (on néglige les effets dynamiques).

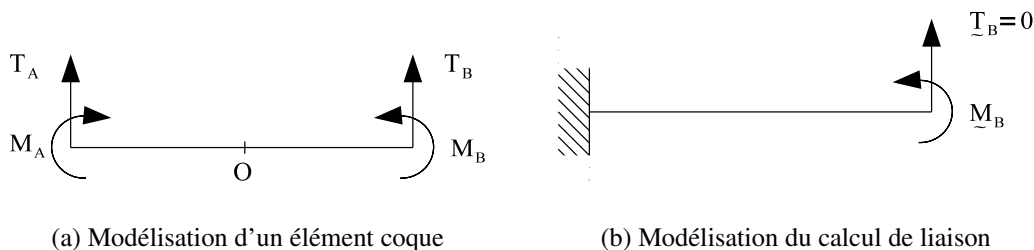


Figure 4.15: Équivalence entre le calcul de la structure et le calcul des liaisons

D'autre part pour le calcul de la liaison, en adoptant également une modélisation poutre (voir Figure 4.15(b)), on est capable de déterminer l'énergie de déformation en fonction de la composante du tenseur qui nous intéresse au centre de la liaison. Dans le calcul de la liaison qui a été effectué, on suppose que le moment fléchissant est constant dans la poutre et vaut \underline{M}_B , d'où $\underline{M}_O = \underline{M}_B$ et l'énergie de déformation

vaut $E_{def} = \frac{1}{2} \frac{L}{EI} M_O^2$.

Pour relier les deux modèles, on suppose que $\underline{M}_O = M_O$, ce qui revient à considérer que l'on se place dans un cas où la liaison va être sollicitée « localement » de la même manière. On détermine ainsi numériquement la valeur de l'énergie de déformation dans la liaison et il suffit ensuite de prendre dans la base de données, l'énergie dissipée correspondante.

5 Calcul d'autres liaisons types des lanceurs spatiaux

La méthode ayant été validée, le code JointVirt a permis de traiter d'autres liaisons : la bride à 90°, notée T90, et la bride à 45°, notée T45 (la Figure 4.16 recense la liste des liaisons existantes sur Ariane 5). Les résultats ont été fournis à ASTRIUM-ST mais ces résultats ne sont pas présentés ici pour des raisons de confidentialité.

5.1 Bride à 90°

La bride à 90° est une liaison qui permet de relier deux cylindres en aluminium par l'intermédiaire de deux bordures à 90° accueillant les boulons. L'axe des boulons est ainsi décalé de la fibre neutre des cylindres, contrairement à la bride chapelle (notée BCH).

La liaison doit être traitée pour les différents cas de charge qui nous intéressent, à savoir, la traction (T_x), le cisaillement radial (T_r), le cisaillement transversale (T_t) et le moment toroïdale (M_t) représentés sur la Figure 4.17. La modélisation retenue pour cette liaison a été de prendre une bande de liaison contenant un boulon. Des conditions limites de périodicité sont ainsi présentes sur les deux flancs de la liaison. La section inférieure est encastree et les conditions limites en effort sont imposées sur la section supérieure (voir Figure 4.17).

Comme pour la liaison SSS, la taille de maille retenue est de l'ordre de 0,5 mm, ce qui conduit à un problème contenant environ 275 000 degrés de liberté. Il y a 56 sous-structures ayant chacune environ le même nombre de degrés de liberté, 126 interfaces dont 16 interfaces de contact avec frottement, 8 interfaces de contact avec frottement et jeu physique. La décomposition de domaine de cette bride à 90° est donnée sur la Figure 4.18.

Chaque type de sollicitation est décomposée en deux parties : une partie statique (qui peut être nulle) et une partie dynamique. Les chargements suivent l'évolution suivante : entre 0 et 1 s, ils sont nuls, ce qui correspond à la mise en charge des boulons. Entre 1 et 7 s, ils suivent une évolution en triple sinus, la hauteur du sinus correspondant à la valeur de la partie dynamique (voir Figure 4.17). Une partie statique de type compression constante est également rajoutée. Pour chaque type de sollicitation, le calcul est effectué pour plusieurs valeurs de partie dynamique afin

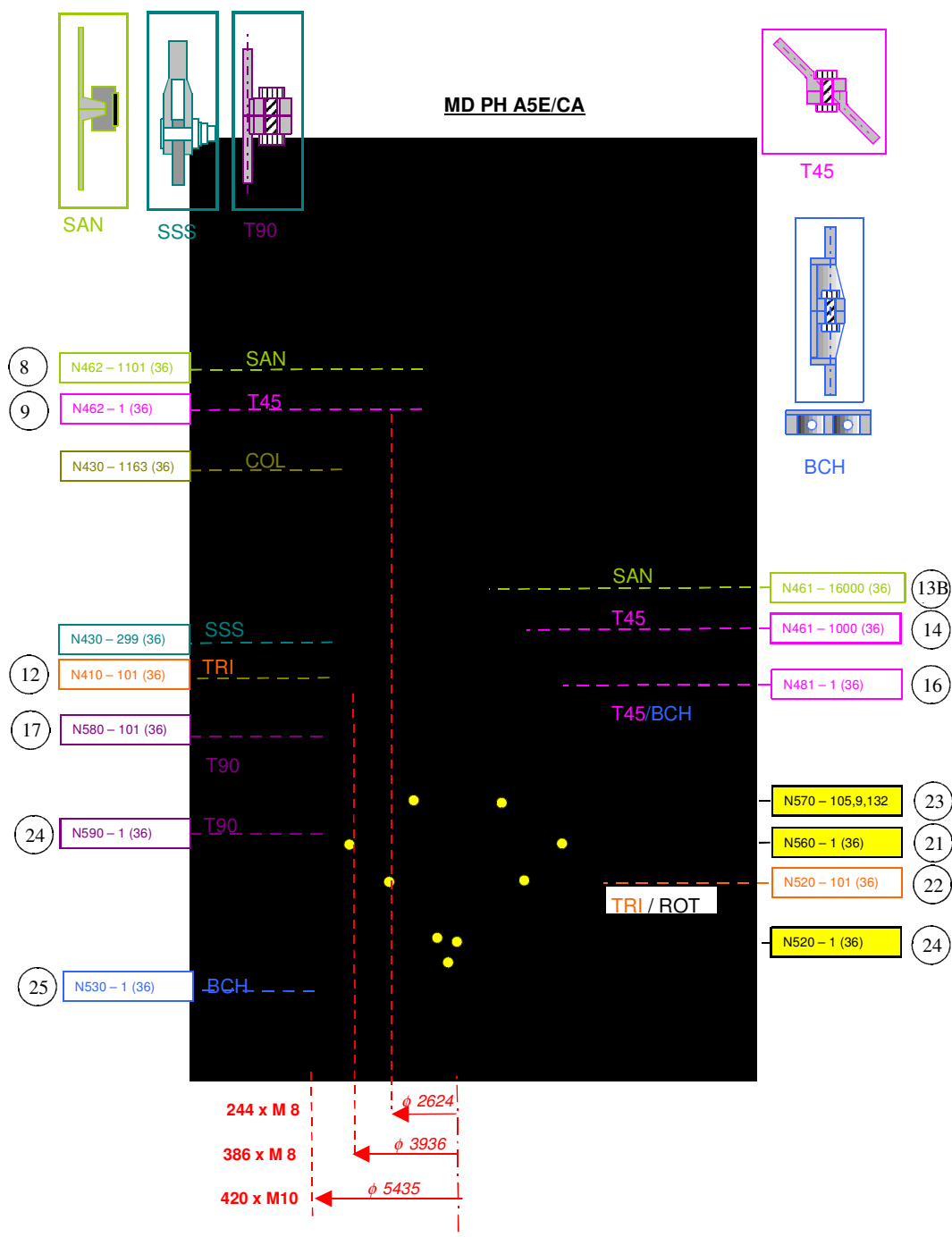


Figure 4.16: Liste des liaisons présentes sur le lanceur Ariane 5

de parcourir la gamme de valeur des efforts d'excitation et étudier l'influence du niveau de l'excitation.

Le calcul d'une sollicitation complète prend environ une semaine sur 16 processeurs d'un cluster d'Opteron™.

La Figure 4.19 présente l'amortissement dans la bride à 90° en fonction du flux de chargement pour les sollicitations de type traction (T_x) et de cisaillement transversale (T_t).

5.2 Bride à 45°

La bride à 45° est une liaison qui relie deux cônes en aluminium. L'axe des boulons est ainsi décalé de 45° par rapport à l'axe de la fibre neutre des cônes.

De même que pour la bride à 90°, la liaison est traitée pour les différents cas de charge qui nous intéressent, à savoir, la traction (T_x), le cisaillement radial (T_r), le cisaillement transversale (T_t) et le moment toroïdale (M_t) représentés sur la Figure 4.21. La modélisation retenue pour cette liaison a été de prendre une bande de liaison contenant un boulon. Des conditions limites de périodicité sont ainsi présentes sur les deux flancs de la liaison. La section inférieure est encastree et les conditions limites en effort sont imposées sur la section supérieure (voir Figure 4.21).

Comme pour la bride à 90°, la taille de maille retenue est de l'ordre de 0,5 mm, ce qui conduit à un problème contenant environ 275000 degrés de liberté. Il y a 56 sous-structures ayant chacune environ le même nombre de degrés de liberté, 126 interfaces dont 16 interfaces de contact avec frottement, 8 interfaces de contact avec frottement et jeu physique. La décomposition de domaine de cette bride à 45° est donnée sur la Figure 4.22.

Le chargement à imposer sur la bride à 45° est du même type que celui de la bride à 90° (voir Figure 4.21). Le calcul d'une sollicitation complète prend également environ une semaine sur 16 processeurs d'un cluster d'Opteron™.

La Figure 4.20 présente l'amortissement dans la bride à 45° en fonction du flux de chargement pour les sollicitations de type traction (T_x) et de cisaillement transversale (T_t).

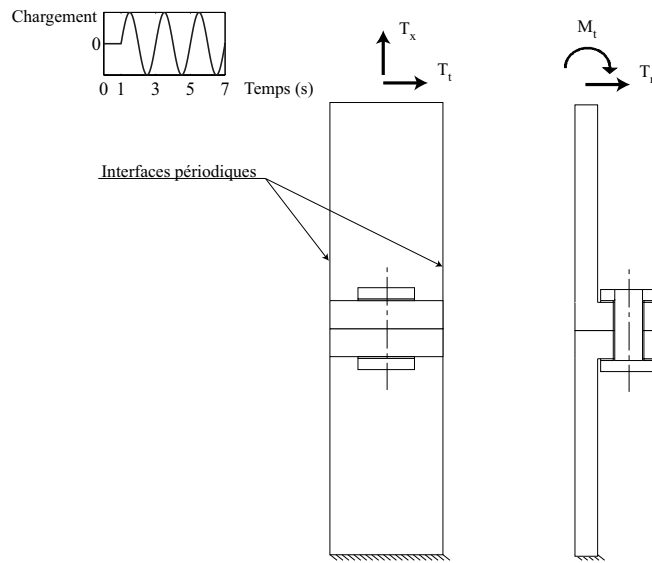


Figure 4.17: Modélisation et conditions limites de la bride à 90°

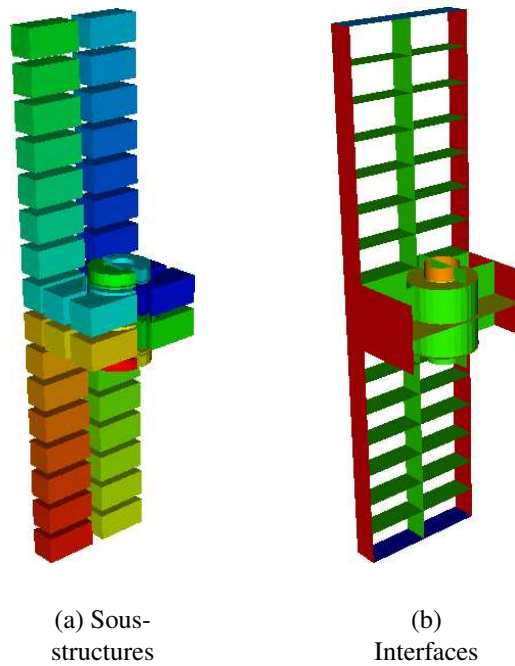


Figure 4.18: Décomposition de domaine de la bride à 90°

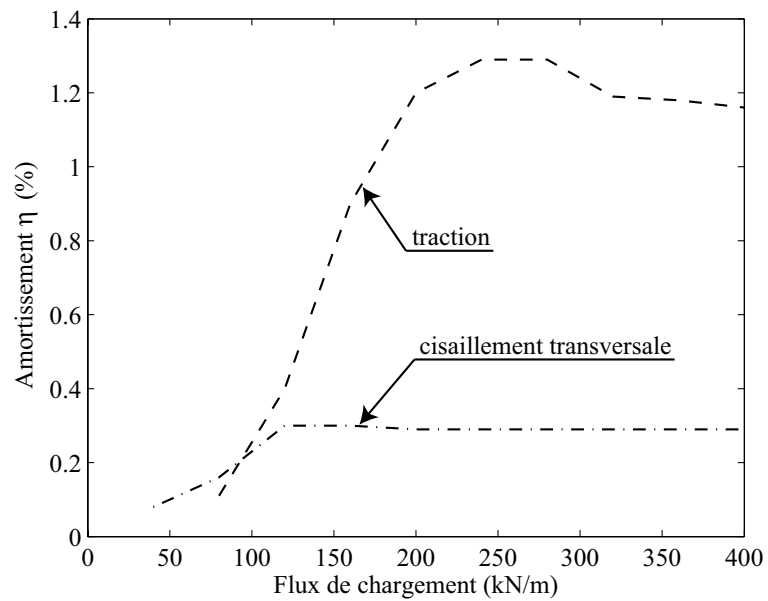


Figure 4.19: Amortissement dans la bride à 90° en traction et cisaillement transversale

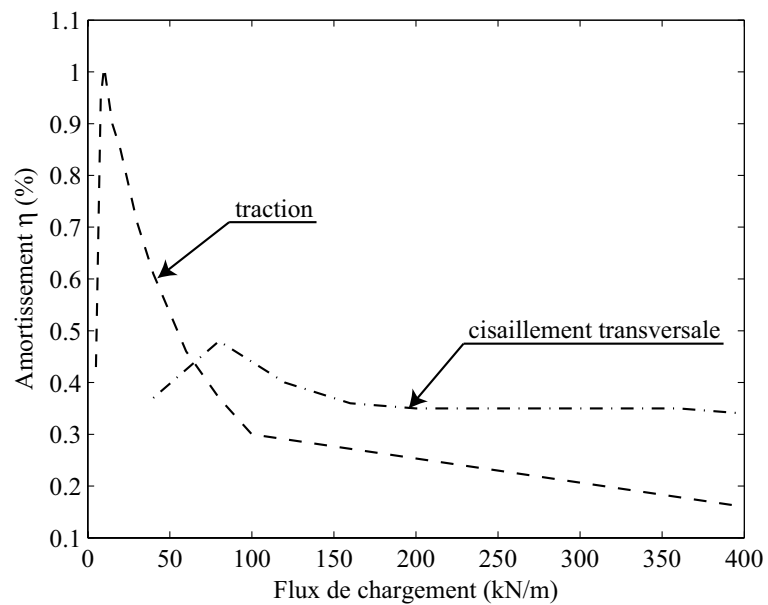


Figure 4.20: Amortissement dans la bride à 45° en traction et cisaillement transversale

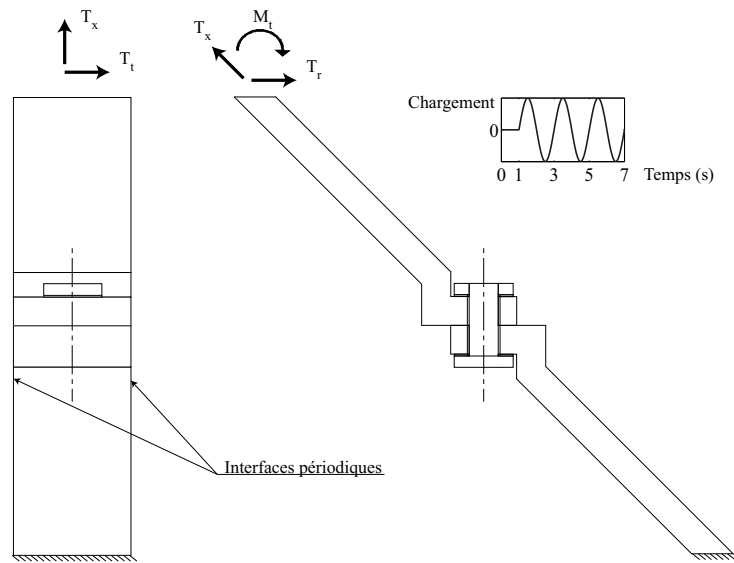


Figure 4.21: Modélisation et conditions limites de la bride à 45°

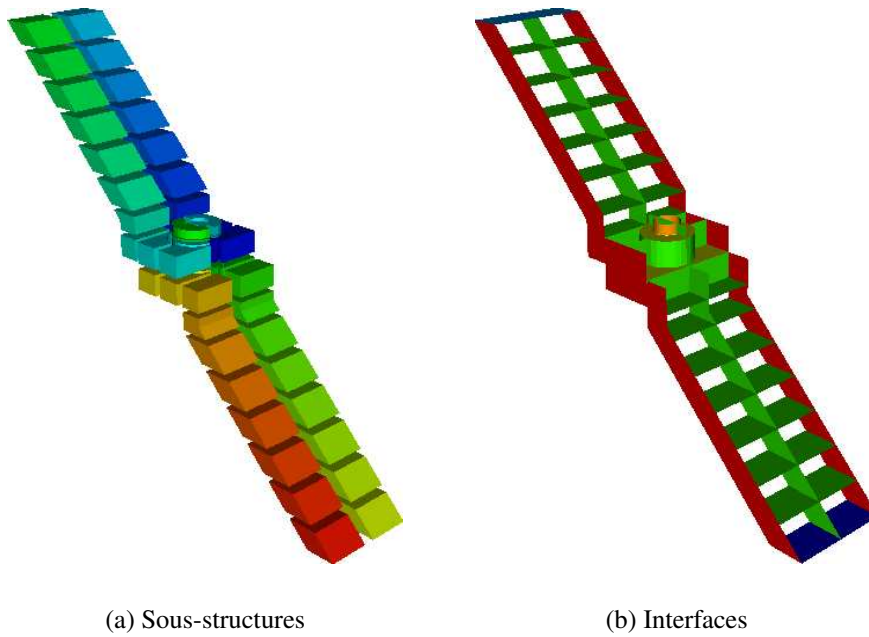


Figure 4.22: Décomposition de domaine de la bride à 45°

Dans ce chapitre, la parallélisation effective du logiciel de calcul est présentée ainsi que les gains qui en résultent.

1 Parallélisation de la méthode

1.1 Motivations

Les calculs nécessaires à la prédiction de la dissipation nécessitent des discrétisations possédant un très grand nombre de degrés de liberté (de 0,5 à 1 million de degrés de liberté pour des liaisons simples voir beaucoup plus). Ces calculs peuvent bien entendu être réalisés sur un seul processeur, la limitation se situant au niveau de la taille de la mémoire et du temps que l'on souhaite accorder à la génération de la solution. Les challenges industriels actuels visent à traiter des problèmes allant jusqu'au milliard de degrés de liberté, notamment pour faire des calculs de composites ou d'assemblages complexes de structures.

À l'heure actuelle, il y a un tournant dans l'évolution des technologies des processeurs. En effet, jusqu'en 2005, la puissance des processeurs augmentait par des montées des fréquences de fonctionnement et des changements d'architecture tels que l'augmentation des performances étaient directement disponibles dans les codes de calcul. La tendance actuelle n'est plus d'augmenter les performances brutes des

processeurs mais d'augmenter le nombre d'unités de calculs (appelées cœurs) sur un même processeur. Il y a actuellement jusqu'à 4 cœurs par processeur mais d'ici 2012, les fabricants annoncent jusqu'à 16 cœurs, voir bien plus par la suite. Même si les compilateurs, actuels et futurs, permettent de paralléliser « automatiquement » quelques opérations très élémentaires, force est de constater que les codes de calcul ne pourront plus bénéficier des améliorations des processeurs sans passer par la parallélisation de leurs méthodes, et les méthodes qui ne sont pas parallélisables seront certainement vouées à disparaître.

La méthode LATIN associée à la décomposition mixte est nativement parallélisable [Cognard *et al.* 1996, Champaney *et al.* 1997] et les aspects multiéchelles de la méthode la rendent extensible [Ladevèze et Dureisseix 2000, Ladevèze *et al.* 2001]. Il a été décidé de paralléliser le code de calcul, développé en C++, pour bénéficier des performances des nouveaux processeurs et pour pouvoir utiliser des architectures de type cluster. Il existe pour cela plusieurs solutions.

La première est de faire du *multithreading* qui consiste à exécuter plusieurs sous-programmes en même temps sur une même machine. Cette solution présente l'avantage de ne pas à avoir à dupliquer les données en mémoire et de limiter les communications. En revanche, elle présente le désavantage de devoir sécuriser l'accès en écriture aux données communes afin de ne pas créer de conflits et surtout elle ne permet pas de travailler sur des architectures de type cluster.

La seconde solution couramment utilisée est d'utiliser des bibliothèques comme MPI (Message Passing Interface) qui permet de gérer la transmission d'informations au travers de *sockets*. Les bibliothèques MPI permettent donc de transférer des informations entre des machines appartenant à un réseau internet ou intranet, ou bien de les transférer sur une même machine. Le principal avantage de cette méthode est ainsi de pouvoir paralléliser le code sur des architectures de type cluster et elle permet également de travailler sur des architectures à mémoire partagée (machine SMP). Le désavantage est que sur une machine à mémoire partagée, on crée des redondances de données, ce qui pénalise l'encombrement de la mémoire. Il faut donc chercher à les diminuer.

1.2 Implantation dans le logiciel

La modification du code a été réalisée de manière non intrusive, c'est-à-dire en limitant les modifications dans les fonctions de base de l'algorithme. Pour cela, il a suffi dans la mise en données de créer, à la place des vecteurs complets d'interface et de sous-structures, des sous-vecteurs ne contenant que les informations nécessaires à la résolution par processeur. La vraie difficulté consiste en la parallélisation de la mise en données, de manière à ne garder en mémoire que les informations nécessaires au déroulement du calcul.

1.2.1 Mise en données

Il s'agit ici de lire les paramètres du calcul et de répartir la charge sur les différents processeurs. La mise en données suit les étapes suivantes :

- lecture des maillages des sous-structures ;
- création des interfaces entre les sous-structures (par intersection des maillages des sous-structures, conditions limites) ;
- modification des propriétés des interfaces (contact, frottement,...) ;
- création des opérateurs : sur les interfaces, sur les sous-structures, « macro ».

Toutes ses étapes ont été parallélisées de façon à ne pas perdre de temps sur la mise en données et d'obtenir un maximum de performance sur le résultat final.

Une fois les interfaces créées, la répartition des sous-structures sur les différents processeurs est réalisée avec les bibliothèques METIS, ce qui permet de minimiser le nombre de données qui doivent transiter entre les différents processeurs afin de ne pas faire trop baisser le speed-up (le speed-up est l'accélération effective quand on augmente le nombre de processeurs, c'est-à-dire que quand on passe de 1 à 2 processeurs, on espère avoir un speed-up de 2). L'utilisation de ce genre de bibliothèque se justifie par le fait qu'on ne peut pas faire une répartition hasardeuse « à la main » des sous-structures au risque de n'avoir pratiquement aucune frontière commune entre les sous-structures communes à un même processeur.

Par exemple pour un calcul de composite fibre-matrice, on se retrouve très vite avec un nombre très important de sous-structures et il devient impossible de réaliser une répartition intelligente à la main. En effet, la Figure 5.1 représente le numéro brut de chacune des sous-structures d'un tel calcul où chaque cercle est une fibre et chaque polygone est un morceau de la matrice du composite. On constate que le numéro des sous-structures peut être aléatoire ce qui nécessite l'utilisation de ce genre d'algorithme de répartition « intelligent ». La répartition obtenue pour ce problème est illustrée sur la Figure 5.2. On constate que les sous-structures vont être réparties sur les processeurs par bloc contigu de façon à minimiser le nombre d'interface à envoyer.

Il suit une étape de libération de la mémoire, qui permet de ne garder que les données nécessaires à la poursuite du calcul dédié à chaque processeur ; le gain obtenu est d'environ 2 sur le maximum de mémoire nécessaire pour effectuer le calcul.

Il suit l'étape de création de tous les opérateurs nécessaires pour effectuer le calcul (matrice de rigidité, matrice de masse...). Chaque processeur crée uniquement les opérateurs qui seront nécessaires pour effectuer sa portion de calcul. On constate qu'il y a des redondances au niveau des interfaces qui peuvent être créées sur deux processeurs différents quand les deux sous-structures voisines sont réparties sur deux processeurs différents. Ce sont les quantités de ces interfaces là qui devront être échangées entre les processeurs concernés lors du calcul.

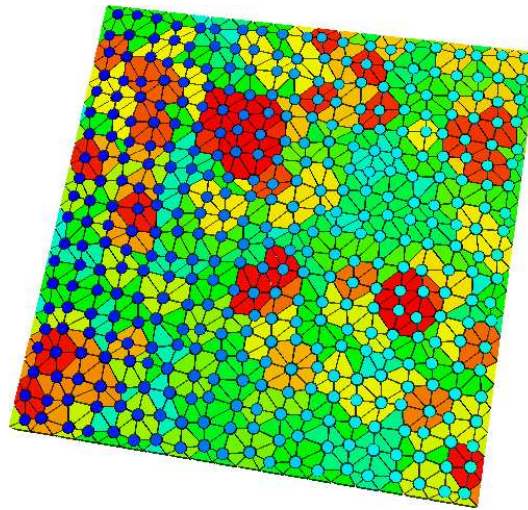


Figure 5.1: Numérotation des sous-structures d'un calcul fibre-matrice

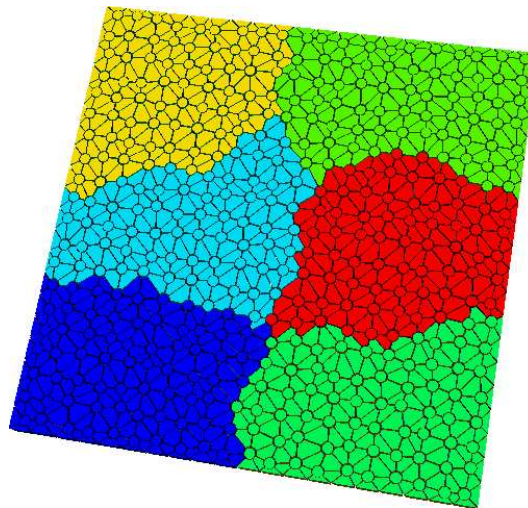


Figure 5.2: Répartition des sous-structures d'un calcul fibre-matrice sur 8 processeurs

Un processeur unique est dédié à la résolution du problème « macro » (couplage multiéchelle entre toutes les sous-structures) afin de ne pas avoir de limitation de ce côté car il devient coûteux en mémoire quand le nombre d'interfaces devient très grand comme dans les problèmes de calcul de composites où leur nombre peut atteindre plusieurs milliers. Cette limitation n'est généralement pas visible pour un problème d'assemblage, car le nombre d'interfaces est en général assez raisonnable

et reste de l'ordre de quelques centaines.

1.2.2 Algorithme de résolution LATIN

On reprend les différentes étapes présentées dans l'algorithme 2 de la section 3 du chapitre 3 et on présente les modifications apportées pour la parallélisation, notamment dans le transfert des informations.

1.2.2.1 Étape locale À partir des quantités calculées à l'étape linéaire sur chaque côté de l'interface, on calcule des nouvelles quantités d'interface qui vérifient les relations de comportement des interfaces. De part la répartition effectuée lors de la mise en données, seules les interfaces présentes sur un processeur seront calculées.

Il n'est pas nécessaire d'envoyer d'informations entre les processeurs à l'issue de cette étape car les interfaces qui sont reliées à deux processeurs auront été calculées sur chacun d'entre eux, ce qui constitue en quelque sorte une redondance dans le calcul.

1.2.2.2 Étape linéaire À partir des quantités calculées à l'étape locale sur chaque côté de l'interface, on calcule des nouvelles quantités d'interface qui vérifient les relations de comportement des sous-structures.

Pour cela, on résout le problème « micro 1 » et on en déduit l'assemblage partiel du second membre « macro ». De même que pour l'étape locale, cette opération est réalisée automatiquement pour les sous-structures nécessaires grâce à la répartition effectuée lors de la mise en données.

À ce stade, il est nécessaire que chaque processeur transmette sa partie de second membre « macro » au processeur qui gère le problème « macro ». Ce processeur va résoudre ce problème et ensuite, il va transmettre le multiplicateur « macro » aux autres processeurs.

On peut ensuite résoudre sur chaque processeur le problème « micro 2 » puis effectuer la reconstruction de la solution complète par sous-structure.

1.2.2.3 Étape de relaxation Chaque processeur relaxe les solutions par sous-structure qu'il a préalablement calculées à l'étape linéaire.

La trace sur les interfaces de la solution relaxée est ensuite calculée. Puis s'en suit une étape d'envoi des quantités d'interface, pour celle qui sont entre deux sous-structures réparties sur deux processeurs différents. On peut remarquer, l'on envoie l'ensemble des données de l'interface pour tous les pas de temps en une seule fois, ce qui limite les appels de connexion répétés.

1.2.2.4 Calcul d'erreur On calcule la contribution du numérateur et du dénominateur de l'erreur par processeur et il suffit ensuite de faire la somme de toutes les contributions puis d'en faire le rapport pour obtenir le calcul de l'erreur.

1.2.3 Visualisation des résultats

Un des problèmes majeurs quand on fait des calculs avec beaucoup d'inconnues est la visualisation des résultats. En effet, il faut pouvoir gérer l'affichage du maillage ainsi que des différents champs que l'on souhaite visualiser. Le code de calcul développé utilise Paraview pour le post-traitement. Paraview a été développé dans l'optique de la visualisation parallèle et permet ainsi de visualiser des gros volumes de données.

À cela, on a rajouté une fonctionnalité qui permet de visualiser uniquement les champs sur les peaux des maillages, qui sont bien souvent suffisant. On peut également les visualiser sous-structures par sous-structures et pas de temps par pas de temps. Il suffit ensuite d'effectuer la combinaison nécessaire à la visualisation souhaitée, comme un champ sur le maillage complet ou un champ sur une sous-structure pour tous les pas de temps.

1.3 Gains obtenus grâce aux parallélisme

1.3.1 Extensibilité

On va tout d'abord illustrer l'extensibilité numérique de la méthode [Ladevèze et Dureisseix 2000, Ladevèze *et al.* 2001] sur deux exemples simples. Cela consiste à montrer que la méthode permet de converger dans le même nombre d'itérations vers la solution donnée quel que soit le nombre de sous-structures.

1.3.1.1 Premier exemple : calcul linéaire On effectue le calcul simple d'une poutre en traction avec un comportement élastique isotrope. Elle est encastée à une extrémité et un effort constant est appliqué à l'autre extrémité (voir Figure 5.3).

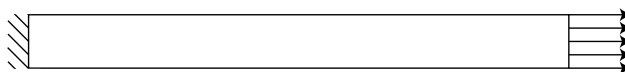


Figure 5.3: Conditions limites de la poutre en traction

Le calcul de cette poutre en traction est réalisée pour différentes décompositions : 2, 4, 8, 16 et 32 sous-structures (Figure 5.4). L'erreur LATIN tracée sur la Figure 5.5, montre que la convergence de la méthode est indépendante du nombre de sous-structures.

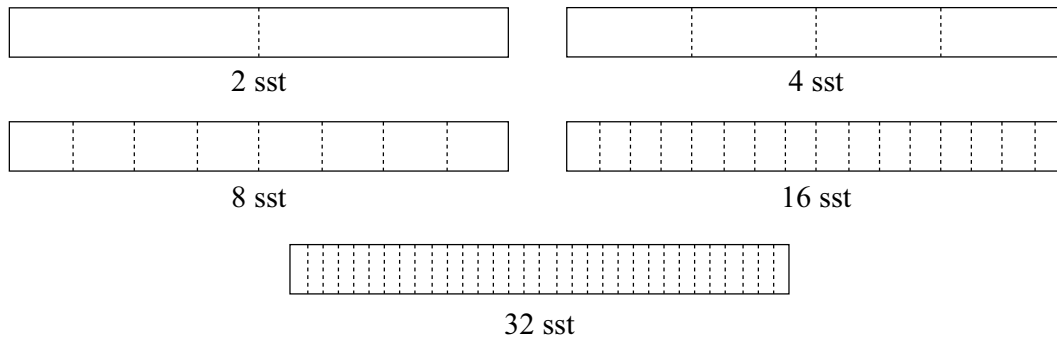


Figure 5.4: Décomposition de la poutre

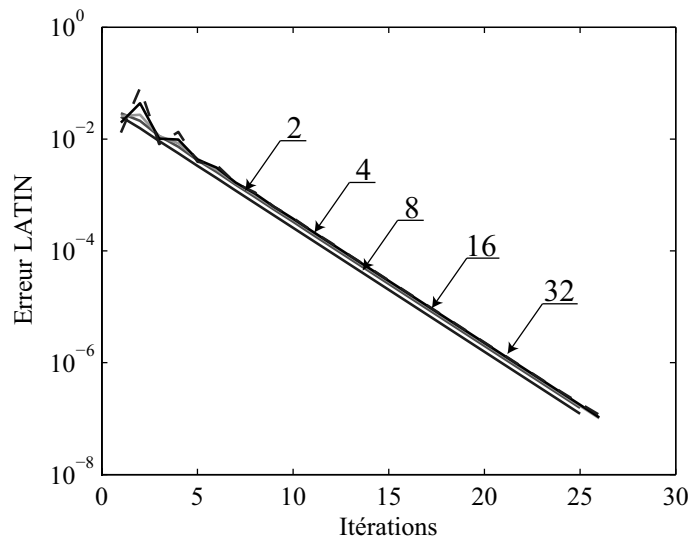


Figure 5.5: Convergence de la méthode LATIN multiéchelle en fonction du nombre de sous-structures pour la poutre en traction

En revanche, si on effectue la même série de calcul avec l'algorithme LATIN dans sa version monoéchelle, on constate sur la Figure 5.6 que sans les aspects multiéchelle, la méthode est non extensible et est beaucoup moins performante.

1.3.1.2 Second exemple : calcul non linéaire On reprend le calcul des deux poutres en contact présenté dans le chapitre 3 dont les conditions limites sont rappelés sur la Figure 5.7. Le problème à résoudre est celui de deux poutres de section carrée posées l'une sur l'autre. On impose des conditions de symétrie sur les plans (xOy) , (xOz) et (yOz) pour bloquer les mouvements de corps rigide. On applique

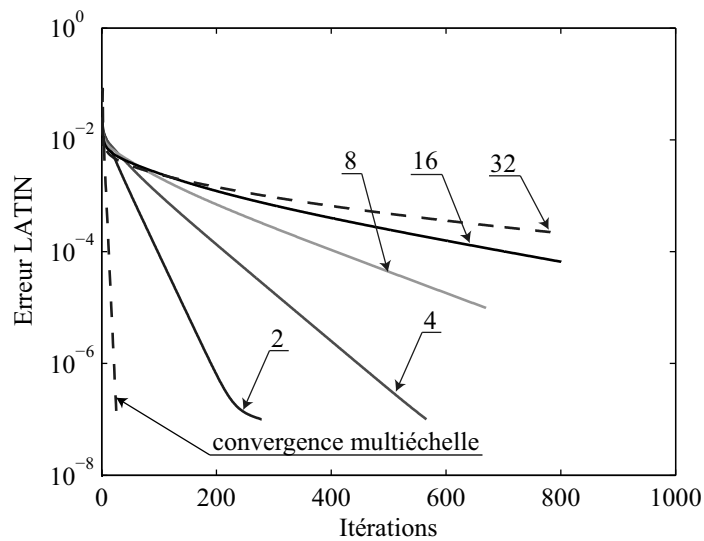


Figure 5.6: Convergence de la méthode LATIN monoéchelle en fonction du nombre de sous-structures pour la poutre en traction

sur la partie supérieure un effort de 30 MPa tout en appliquant sur l'extrémité restante avec un effort de 100 MPa.

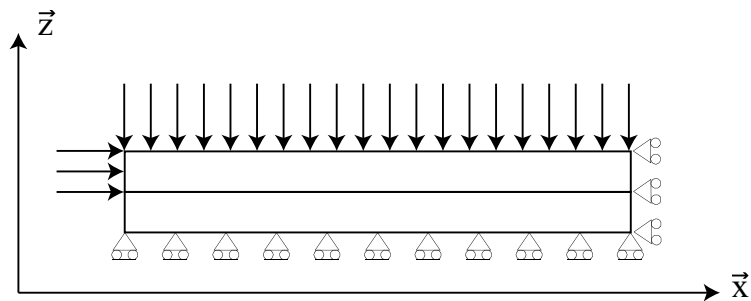


Figure 5.7: Modélisation des deux poutres en contact

On a effectué le calcul en faisant varier le nombre de sous-structures composant les deux poutres. Le calcul a ainsi été effectué pour 2, 4, 8, 16 et 32 sous-structures par poutre (Figure 5.8). L'erreur LATIN tracée sur la Figure 5.9 pour chacun de ses calculs, montre que la convergence de la méthode est indépendante du nombre de sous-structures pour les problèmes non linéaires également.

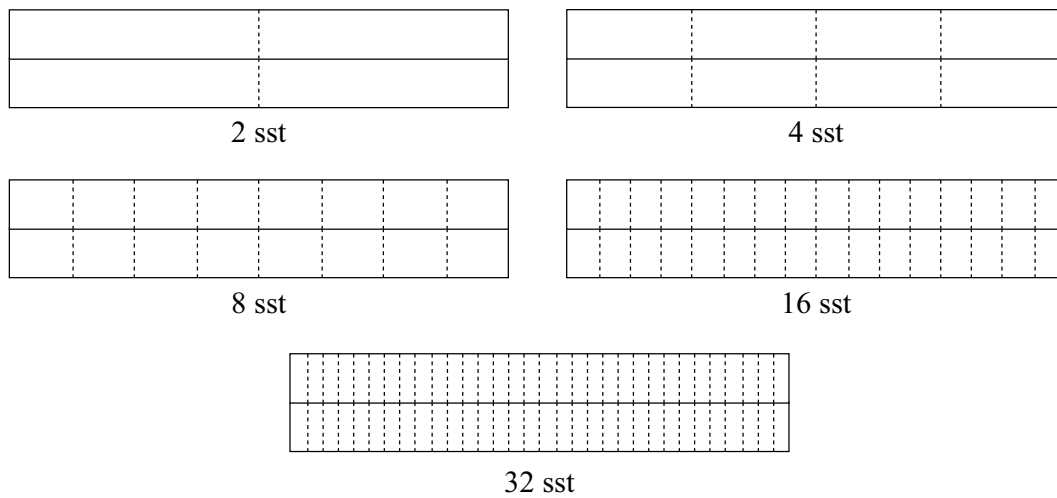


Figure 5.8: Décomposition des deux poutres

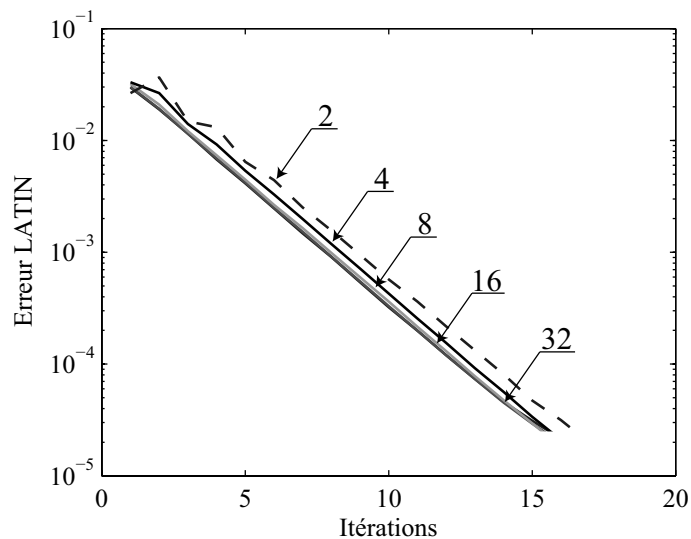


Figure 5.9: Convergence de la méthode LATIN en fonction du nombre de sous-structures pour les deux poutres en contact

1.3.2 Speed-up

Le speed-up représente la gain apporté par le parallélisme en fonction du nombre de processeurs. La limite théorique et visée est que si on prend n processeurs, le calcul va être effectué n fois plus vite. Dans la pratique, il peut y avoir des pertes

de temps dues à des envois répétés d'informations ou dues tout simplement à une mauvaise répartition des données. L'avantage de la méthode LATIN est que les données à échanger sont localisées uniquement sur les interfaces et sont donc d'une taille très faible. À titre d'exemple, le temps passé sur un calcul à échanger des données représente moins de 1% du temps complet, l'essentiel du temps, environ 97%, étant dédié à la résolution des problèmes linéaires « micro 1 » et « micro 2 ».

La Figure 5.10 représente le speed-up obtenu pour le problème de la poutre en traction, sachant qu'en pratique, il ne dépend que de la bonne répartition des sous-structures et interfaces sur les différents processeurs.

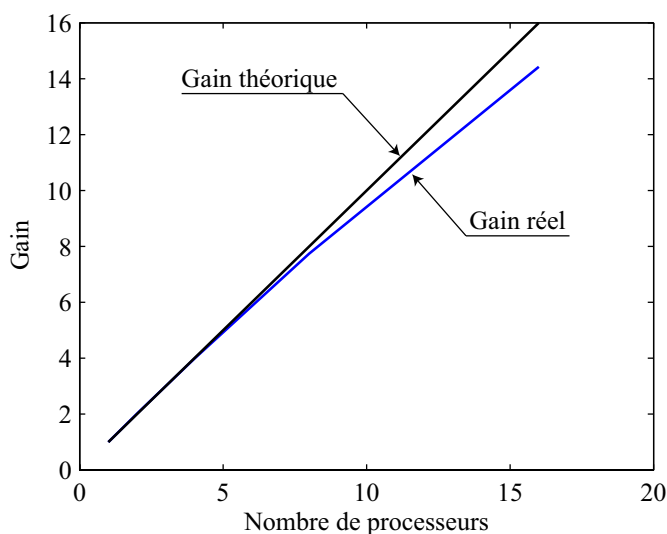


Figure 5.10: Speed-up pour le problème de la poutre en traction

On peut remarquer que pour un calcul sur 16 processeurs, le gain est de pratiquement 15 ce qui montre que la méthode est réellement parallélisable de manière efficace et que le code est bien optimisé.

1.4 Conclusions

C'est grâce à la parallélisation du logiciel que le calcul de la dissipation a été rendu possible dans un temps de calcul raisonnable. Il fallait 3 mois de calcul pour obtenir les résultats présentés dans la section 4 alors qu'après la parallélisation ces calculs sont possibles en moins de 15 jours sur sept processeurs.

Bien que le logiciel actuel soit performant, de nombreuses améliorations sont encore possibles dans différents domaines.

D'un point de vue méthode de calcul, la LATIN n'est pas encore implantée avec toutes les spécificités qui font sa puissance. Il reste à implanter l'approximation espace-temps [Ladevèze 1999] qui consiste à représenter l'ensemble des inconnues du problème par des produits de fonctions de l'espace et du temps, ce qui conduit à diminuer les coûts de stockage et les coûts de calculs. Cette approximation espace-temps peut être couplée avec des aspects multiéchelle en temps, ce qui permet de rendre la méthode extensible en temps de façon à avoir les performances maximales [Ladevèze *et al.* 2009].

D'un point de vue programmation, des améliorations sont aussi envisageables. Dans la mise en données notamment, des approches alternatives peuvent être utilisées. À l'heure actuelle, la recherche des interfaces se fait en parcourant les éléments de peaux des sous-structures et en détectant les éléments de peaux des sous-structures voisines. Cela implique notamment que l'on ne peut travailler qu'avec des maillages compatibles et que pour les interfaces avec des jeux physiques, il est nécessaire de donner les maillages d'interface. Une des approches alternatives proposées consisteraient, à partir d'un maillage donné, à trouver les surfaces élémentaires qui le constitue (cône, cylindre, plan) afin de travailler ensuite sur les intersections des géométries. La gestion de la détection des maillages incompatibles serait ainsi automatiquement réalisée. À terme, cette évolution permettra également de construire un maillage avec des éléments rationnels afin de travailler sur des géométries quasi-exactes.

Un autre point d'amélioration se situe dans la gestion des formulations sur les sous-structures. Une formulation est un outil de génération de code, qui génère entre autres, la fabrication des matrices de rigidité associées à une formulation variationnelle donnée. À partir de ces matrices, on vient actuellement lui rajouter les directions de recherche avant de factorisation l'opérateur complet. L'évolution consisterait à effectuer une formulation qui génère l'opérateur complet avec l'optimisation des directions de recherche.

Le dernier point notable d'amélioration de la façon de programmer se situe au niveau des comportements d'interface. À l'heure actuelle, les comportements sont écrit « à la main » élément par élément du fait de la discrétisation P0 des interfaces. L'objectif serait de créer des formulations d'interface, qui à partir d'un comportement d'interface générerait la résolution qui permet de trouver les quantités de l'étape « locale ». Cette vision sous forme de formulation permettra de développer beaucoup plus facilement de nouveaux comportements d'interfaces et surtout cela permettra de gérer également les maillages incompatibles.

D'un point de vue informatique, il y a également de nombreuses pistes en voie d'exploration. La plus notable est basée sur le concept « d'évaluation paresseuse ». Cela permettra notamment d'explorer le programme de manière analytique, de faire

toutes les simplifications possibles, et surtout de calculer les quantités seulement au moment où il est vraiment nécessaire (pour un affichage par exemple). Cela permettra également, de paralléliser plus ou moins automatiquement une grande partie des tâches comme [Leclerc 2007] l'a montré, ce qui apparaît indispensable avec l'évolution actuelle des processeurs, surtout quand on parle d'avoir une centaine de cœurs par processeur d'ici une dizaine d'année.

2 Un exemple de calcul

Cet exemple a été réalisé dans le cadre d'un projet de TER (travail encadré de recherche) en M1 au sein du LMT-Cachan par MM. Luc LAURENT et Julien BERTHE sous la direction du Pr. Pierre-Alain BOUCARD et du doctorant David ODIEVRE en partenariat avec M. Alain FANGET, ingénieur à la DGA. L'objectif de ce travail était de montrer la possibilité de simuler numériquement le comportement d'une cellule de matériau typique d'un propergol sous l'effet d'une forte charge statique ou sous d'un petit choc.

Suite à l'analyse - réalisée par le Centre d'Étude de Gramat - de ces matériaux par micro-tomographie rayons X, il s'avère que leur composition repose sur une architecture grains-matrice comme un matériau composite (voir la Figure 5.11) avec un pourcentage volumique de grains très élevé (au moins 70%). L'analyse numérique du comportement d'une cellule d'un tel matériau est très complexe à cause, d'une part, de la mise en données du problème (maillage) et, d'autre part, du nombre très important d'interfaces de contact avec frottement entre les grains et la matrice.

Dans une première étape, afin de simuler le comportement d'une cellule de ce matériau, chaque grain est d'abord assimilé à un hexaèdre. Le problème est alors de positionner au sein de la matrice (dont on suppose qu'elle est d'épaisseur constante) des hexaèdres de tailles aléatoires. L'ensemble est ensuite maillé en imposant des maillages compatibles sur toutes les interfaces. Notons qu'en pratique la matrice qui entoure un grain est elle-même décomposée en 6 sous-structures. Ainsi, pour un grain et sa matrice environnante, il faut 7 sous-structures (voir la Figure 5.12).

Le calcul qui est présenté a été réalisé ici sur une cellule cubique de matériau comprenant 8000 grains hexaédriques. Cette cellule est encadrée sur une des faces et soumise à de la traction ou de la compression sur la face opposée. Ce chargement n'est pas réaliste vis-à-vis des chargement que la DGA souhaite réaliser mais cela permet d'illustrer les capacités du logiciel.

Ainsi, le problème se compose donc de 56 000 sous-structures. Au total il comporte 167 000 interfaces dont 48 000 sont des interfaces de contact-frottement (entre les hexaèdres et la matrice qui les entoure) pour un total de 75,8 millions de degrés de liberté. Sur la Figure 5.13 est représentée la répartition par processeur (calcul

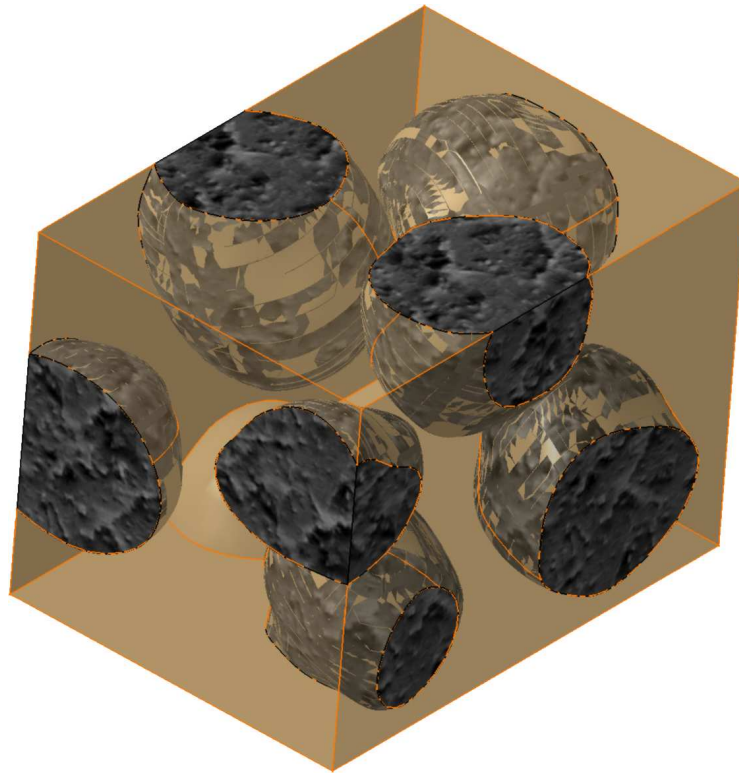


Figure 5.11: Architecture grains-matrice

réalisé sur 54 processeurs + 1 pour le traitement du problème macroscopique) des sous-structures « grains » pour la cellule de matériau composée de 8000 grains.

Avec un tel nombre d'interfaces, la taille du problème « macro » est très importante et la factorisation de la matrice prend 3 h, la factorisation des opérateurs sur les sous-structures a quant à lui pris 50 min. Le calcul converge en 59 itérations en 35 min seulement sur les 55 processeurs. Au total, pour résoudre ce problème de statique non linéaire, il aura fallu 6h15. Notons que le passage en quasi-statique - indispensable pour la prise en compte plus réaliste du contact-frottement - nécessite une discrétisation du chargement en quelques pas de charges. Par exemple pour une dizaine de pas de charges et en estimant à 35 minutes le calcul de chaque pas on obtiendrait alors la solution du problème en quasi-statique non linéaire en environ : 5h50 de calcul + 5h40 de préparation (lecture des données, assemblage et factorisation des opérateurs) soit un peu moins de 12 h au total.

La visualisation des résultats d'un tel calcul est un nouveau problème à résoudre en raison du nombre très important d'informations à traiter : le fichier résultat fait plus de 7 Go pour un seul pas de chargement. Paraview permet la visualisation en parallèle d'une telle quantité d'informations mais son utilisation dans ce mode n'est

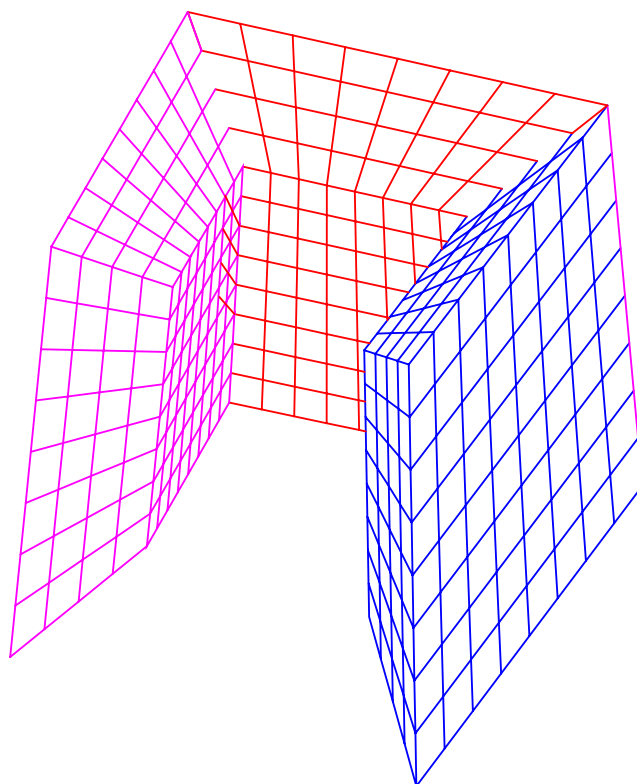


Figure 5.12: Visualisation de 3 des sous-structures composant la matrice autour d'un grain

pas des plus aisée. En revanche, à partir du code de calcul, on peut avoir accès à un certain nombre d'informations comme le déplacement d'un point, l'énergie imposée, l'énergie dissipée...

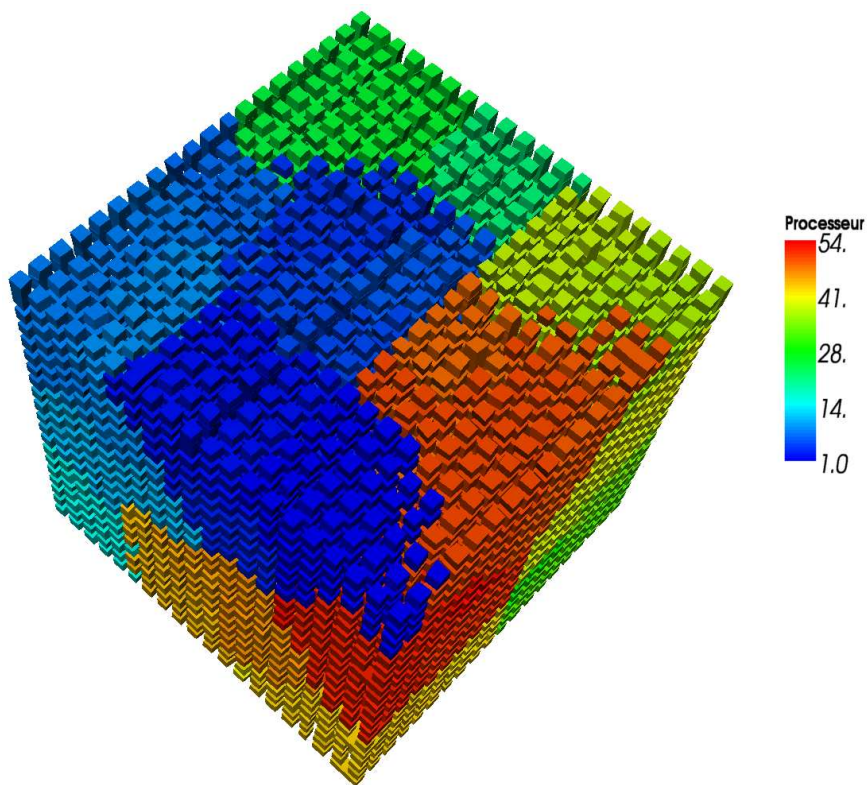


Figure 5.13: Matériau composé de 8000 grains

Conclusion

Ce travail a permis de franchir une étape importante dans la détermination de l'amortissement dans les structures complexes, comme les lanceurs spatiaux. Il s'insère dans une logique de conception industrielle globale chez ASTRIUM-ST Les Mureaux sur la prédiction de l'amortissement, dont le principe est de déterminer l'amortissement à partir d'une base de données des différentes sources dissipatives. Ce travail permet d'alimenter cette base de données pour les liaisons.

Le calcul de la dissipation passe par la prise en compte des sources de dissipation dans un calcul éléments finis. On s'est limité ici aux dissipations provoquées par le frottement des différentes pièces en contact, qui sont largement prépondérantes dans les problèmes considérés. On a montré que ce calcul n'était pas réalisable avec des méthodes de calcul standards, et que seule une stratégie adaptée permet de résoudre convenablement le problème. La méthode employée ici est la méthode LATIN dans sa version multiéchelle. Les calculs permettant d'obtenir la dissipation sont plus coûteux que les calculs globaux standards, car on se s'intéresse ici essentiellement aux quantités locales.

On a montré que le calcul numérique mené permettait de représenter tout à fait le comportement réel des liaisons en comparant les résultats numériques aux résultats expérimentaux, ce qui a validé toute la démarche.

Un des résultats majeurs de ce travail est que l'utilisation d'un modèle de Coulomb simple pour les interfaces de contact avec frottement, est suffisant pour calculer la dissipation dans les liaisons. Ainsi, la connaissance du coefficient de frottement sur les différentes interfaces et de son évolution au cours du temps permet, à elle seule, de déterminer l'amortissement dans les liaisons. Aussi, avant tout calcul de dissipation numérique, il est nécessaire d'effectuer des études de tribologie sur les matériaux en présence. Ces études, qui permettront d'alimenter le simulateur d'amortissement, sont largement moins coûteuses que des études expérimentales sur des structures complètes comme cela est le cas actuellement.

Le calcul de la dissipation a pu être réalisé, d'une part grâce à la méthode mais

d'autre part grâce à un développement logiciel assez poussé. Le logiciel actuel est développé en C++ avec des bibliothèques, qui permettent d'allier performance et facilité d'implantation. Ce code sert maintenant de base à plusieurs thèses sur des domaines assez différents (composites à l'échelle micro, à l'échelle méso, calcul d'erreur, contact, dynamique, ...). Un des points clés de la réussite du calcul est la parallélisation de celui-ci. Cette parallélisation a été rendue possible assez facilement car la méthode LATIN est naturellement sous-structurée. La parallélisation a été effectuée avec les bibliothèques MPI qui permettent au logiciel final de pouvoir fonctionner sur toutes les architectures disponibles : machine à mémoire partagée (SMP) et machine à mémoire distribuée (cluster).

Les développements en cours concernent la généralisation de l'utilisation de la génération de code à l'aide de formulation pour avoir une approche logicielle encore plus modulaire. Le concept de formulation d'interface permettra notamment de pouvoir gérer les maillages incompatibles. La gestion des maillages incompatibles est une nécessité dans le cas de l'utilisation de la méthode dans un cadre industriel, car il facilite énormément la discrétisation des problèmes.

Enfin, il faut noter que le logiciel développé a déjà été transféré chez ASTRIUM-ST en avril 2007 et des calculs de nouvelles liaisons ont pu être ainsi effectués. Cela a permis notamment d'effectuer les premières évaluations de l'amortissement sur des structures complexes qui ont donné des résultats remarquables sur certains modes.

Bibliographie

[Adams et Maheri 1993]

R. D. Adams et M. R. Maheri. The dynamic shear properties of structural honeycomb materials. *Composites Science and Technology*, 47 : 15–23, 1993.

[Adams et Maheri 1994]

R. D. Adams et M. R. Maheri. Dynamic flexural properties of anisotropic fibrous composite beams. *Composites Science and Technology*, 50 : 497–514, 1994.

[Barabanov et Ortega 2000]

N. Barabanov et R. Ortega. Necessary and sufficient conditions for passivity of the lugre model. Dans *IEEE Transactions on Automatic Control*, 2000.

[Caignot et al. 2005]

A. Caignot, P. Ladevèze, D. Néron, S. Le Loch, V. Le Gallo, K.M. Ma et T. Ro-meuf. Prediction of damping in space launch vehicles using a virtual testing strategy. Dans *Proc. of 6th International Symposium on Launcher Technologies*, 2005.

[Champaney 1996]

L. Champaney. *Une Nouvelle Approche Modulaire pour l'Analyse d'Assemblages de Structures Tridimensionnelles*. Thèse de doctorat, ENS de Cachan, LMT-Cachan, 1996.

[Champaney et al. 1997]

L. Champaney, J.-Y. Cognard, D. Dureisseix et P. Ladevèze. Large scale applications on parallel computers of mixed domain decomposition method. *Computational Mechanics*, 19 : 253–263, 1997.

[Chandra et al. 2002]

R. Chandra, S.P. Singh et K. Gupta. Micromechanical damping models for fiber-reinforced composites : a comparative study. *Composites*, 333 : 787–796, 2002.

[Chandra *et al.* 2003]

R. Chandra, S.P. Singh et K. Gupta. A study of damping in fiber-reinforced composites. *Journal of Sound and Vibration*, 262 : 475–496, 2003.

[Chesson et Munse 1964]

E. Chesson et W.H. Munse. Studies of the behavior of high-strength bolted joints. Rapport technique Bulletin 469, University of Illinois, Experimental Station, 1964.

[Cognard *et al.* 1996]

J.-Y. Cognard, D. Dureisseix, P. Ladevèze et Ph. Lorong. Expérimentation d'une approche parallèle en calcul de structures. *Revue Européenne des éléments Finis*, 5(2) : 197–220, 1996.

[Crocombe *et al.* 2006]

A.D. Crocombe, R. Wang, G. Richardson et C.I. Underwood. Estimating the energy dissipated in a bolted spacecraft at resonance. *Computer and Structures*, 84 : 340–350, 2006.

[Dahl 1976]

P. Dahl. Solid friction damping of mechanical vibrations. *AIAA Journal*, 14 : 1675–1682, 1976.

[Deü *et al.* 2008]

J.F. Deü, A.C. Galucio et R. Ohayon. Dynamic responses of flexible-link mechanisms with passive/active damping treatment. *Computers and Structures*, 86 : 258–265, 2008.

[Dureisseix et Ladevèze 1998]

D. Dureisseix et P. Ladevèze. *A multi-level and mixed domain decomposition approach for structural analysis*, volume 218 de *Contemporary Mathematics, Domain Decomposition Methods 10*, pages 246–253. AMS, 1998.

[Esteban et Rogers 1999]

J. Esteban et C.A. Rogers. Wave localization due to material damping. *Computer Methods in Applied Mechanics and Engineering*, 177(1-2) : 93–107, July 1999.

[Galucio *et al.* 2007]

A.C. Galucio, J.F. Deü et R. Ohayon. Hybrid active-passive damping treatment of sandwich beams in non-linear dynamics. *Journal of Vibration and Control*, 13(7) : 851–881, 2007.

[Gaul et Nitsche 2001]

L. Gaul et R. Nitsche. The role of friction in mechanical joints. *Applied Mechanics Reviews*, Vol 54 : 93–106, 2001.

[Géradin et Rixen 1997]

M. Géradin et D. Rixen. *Théorie des Vibrations, Application à la Dynamique des Structures, 2nde édition*. Collection Physique Fondamentale et Appliquée. Masson, France, Paris, 1997.

[Gropper et al. 1966]

M. Gropper, V. Bausic et G. d'Albon. Measuring and control method for the prestressing force and its variation in time of high tensile bolts. *Studii si cercetari*, INCRC, Bucaresti, Romania, 1966.

[Gropper et Hemmye 1983]

M. Gropper et J. Hemmye. The dissipation of energy in high strength friction grip bolted joints. Dans *SESA Spring Conference, Cleveland, OH*, 1983.

[Gropper et Hemmye 1983]

M. Gropper et J. Hemmye. Partial slip damping in high strength friction grip bolted joints. Dans *Fourth International Conference of Mathematical Modeling, New York*. Pergamon Press, 1983.

[Guidault et al. 2007]

P.-A. Guidault, O. Allix, L. Champaney et J.-P. Navarro. A two-scale approach with homogenization for the computation of cracked structures. *Computers and Structures*, 85 : 1360–1371, 2007.

[Hensen 2002]

R.H.A. Hensen. *Controlled Mechanical Systems with Friction*. Thèse de doctorat, Technische Universiteit Eindhoven, 2002.

[Hibbitt et al. 2006]

Hibbitt, Karlson et Sorensen, éditeurs. *Abaqus/Standard - User's Manual*. 2006.

[Hirano 2006]

M. Hirano. Atomistics of friction. *Surface Science Reports*, 60 : 159–201, 2006.

[Ibrahim et Pettit 2005]

R. A. Ibrahim et C. L. Pettit. Uncertainties and dynamic problems of bolted joints and other fasteners. *Journal of Sound and Vibration*, 279 : 857–936, 2005.

[Ichchou et al. 2008]

M. Ichchou, O. Bareille et J. Berthaut. Identification of effective sandwich structural properties via an inverse wave approach. *Engineering Structures*, 30(10) : 2591–2604, 2008.

[Jalali et al. 2007]

H. Jalali, H. Ahmadian et J. E. Mottershead. Identification of nonlinear bolted lap-joint parameters by force-state mapping. *Solids and Structures*, 44 : 8087–8105, 2007.

[Julvez 2007]

J. Julvez. Coefficient de frottement pour revêtement alodine. Rapport technique, Société Henkel, 2007.

[Ladevèze 1985]

P. Ladevèze. Sur une famille d'algorithmes en mécanique des structures. *Comptes-Rendus de l'Académie des Sciences*, 300(2) : 41–44, 1985.

[Ladevèze 1999]

P. Ladevèze. *Nonlinear Computational Structural Mechanics – New Approaches and Non-Incremental Methods of Calculation*. Springer Verlag, 1999.

[Ladevèze et Dureisseix 2000]

P. Ladevèze et D. Dureisseix. A micro/macro approach for parallel computing of heterogeneous structures. *International Journal for Computational Civil and Structural Engineering*, 1 : 18–28, 2000.

[Ladevèze et al. 2001]

P. Ladevèze, O. Loiseau et D. Dureisseix. A micro-macro and parallel computational strategy for highly heterogeneous structures. *International Journal for Numerical Methods in Engineering*, 52(1-2) : 121–138, 2001.

[Ladevèze et al. 2006]

P. Ladevèze, G. Lubineau et D. Violeau. A computational damage micromodel of laminated composites. *International Journal of Fracture*, 137 : 139–150, 2006.

[Ladevèze et al. 2007]

P. Ladevèze, D. Néron et P. Gosselet. On a mixed and multiscale domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 196 : 1526–1540, 2007.

[Ladevèze et al. 2009]

P. Ladevèze, D. Néron et J.-C. Passieux. Bridging the scales in science and engineering. *Section On multiscale computational mechanics with time-space homogenization*, pages 18–28, A paraitre. 2009.

[Ladevèze et Nouy 2003]

P. Ladevèze et A. Nouy. On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 192 : 3061–3087, 2003.

[Le Gallo 2005]

V. Le Gallo. Étude de l'amortissement des lanceurs. *Cahier des charges fonctionnelles ASTRUM-ST*, 2005.

[Leclerc 2007]

H. Leclerc. Lazy evaluation for automatic parallelisation and pertinent use of deep memory hierarchies. Dans Proc. of 9th US National Congress on Computational Mechanics, éditeur, *Actes du Septième Colloque National en Calcul des Structures, Giens, France*, 2007.

[Leloch 2003]

S. Leloch. *Modélisation et Identification de l'Amortissement dans les Structures Spatiales*. Thèse de doctorat, ENS de Cachan, LMT-Cachan, 2003.

[Mackerle 2003]

J. Mackerle. Finite element analysis of fastening and joining : A bibliography (1990–2002). *Pressure Vessels and Piping*, 80 : 253–271, 2003.

[Maheri et Adams 1994]

M. R. Maheri et R. D. Adams. Steady-state flexural vibration damping of honeycomb sandwich beams. *Composites Science and Technology*, 52 : 333–347, 1994.

[Maheri et Adams 1995]

M. R. Maheri et R. D. Adams. Finite-element prediction of modal response of damped layered composite panels. *Composites Science and Technology*, 55 : 13–23, 1995.

[Maisonneuve 2004]

P. Maisonneuve. Contribution à l'industrialisation d'un montage sur l'amortissement des modes propres de lanceur spatial, mémoire d'ingénieur, CNAM Paris. 2004.

[Miller et Quinn 2009]

J. D. Miller et D. D. Quinn. A two-sided interface model for dissipation in structural systems with frictional joints. *Journal of Sound and Vibration*, 321 : 201–219, 2009.

[Müser 2002]

M. H. Müser. Towards an atomistic understanding of solid friction by computer simulations. *Computer Physics Communications*, 146 : 54–62, 2002.

[Nashif et al. 1985]

A.D. Nashif, D.I.G. Jones et J.P. Henderson. *Vibration Damping*. John Wiley & Sons, Inc, U.S.A, Braun-Brumfield, Inc, 1985.

[Néron et Dureisseix 2008]

D. Néron et D. Dureisseix. A computational strategy for poroelastic problems with a time interface between coupled physics. *International Journal for Numerical Methods in Engineering*, 73(6) : 783–804, 2008.

[Néron et Dureisseix 2008]

D. Néron et D. Dureisseix. A computational strategy for thermo-poroelastic structures with a time-space interface coupling. *International Journal for Numerical Methods in Engineering*, 75(9) : 1053–1084, 2008.

[Nineb et al. 2007]

S. Nineb, P. Alart et D. Dureisseix. Domain decomposition approach for nonsmooth discrete problems, example of a tensegrity structure. *Computers and Structures*, 85(9) : 499–511, 2007.

[Oldfield et al. 2005]

M. Oldfield, H. Ouyang et J. E. Mottershead. Simplified models of bolted joints under harmonic loading. *Computers and Structures*, 84 : 25–33, 2005.

[Pourroy et al. 1998]

F. Pourroy, S. Shakhesi et P. Trompette. Coupling of flexural and longitudinal damped vibration in a two-layered beam. *Shock and Vibration*, 5 : 337–341, 1998.

[Saravanos et Chamis 1990]

D.A. Saravanos et C.C. Chamis. Unified micromechanics of damping for unidirectional and off-axis fiber composites. *Journal of Composites Technology and Research*, 12(1) : 31–40, Spring 1990.

[Tajima 1964]

J. Tajima. Effects of relaxation and creep on the slip load of the high strength bolted joints. *Japanese National Railways*, 1964.

[Ungar 1964]

E. E. Ungar. Energy dissipation at structural joints : mechanisms and magnitudes. Technical Documentary Report FDL-TDR-64-98, Air-Force Flight Dynamics Lab, Wright-Patterson Air Force Base, 1964.

[Ungar 1973]

E. E. Ungar. The status of engineering knowledge concerning the damping of built-up structures. *Journal of Sound and Vibration*, 26 : 141–154, 1973.

[Violeau 2003]

D. Violeau. Sur une amélioration de la stratégie de calcul avec homogénéisation.

Rapport du DEA techniques avancées en calcul des structures, ENS Cachan, 2003.

[Wentzel et Olsson 2008]

H. Wentzel et M. Olsson. Mechanisms of dissipation in frictional joints – influence of sharp contact edges and plastic deformation. *Wear*, 265 : 1814–1819, 2008.

[Zang et al. 2008]

C. Zang, C.W. Schwingshackl et D.J. Ewins. Model validation for structural dynamics analysis : An approach to the sandia structural dynamics challenge. *Computer methods in applied mechanics and engineering*, 197 : 2645–2659, 2008.

[Zener 1948]

C. Zener. *Elasticity and Anelasticity of Metals*. University of Chicago press édition, 1948.

Notice d'utilisation du logiciel JointVirt

Le logiciel JointVirt mis au point permet de traiter notamment des problèmes d'assemblage avec contact et frottement avec des comportements matériaux élastiques linéaires en statique ou quasi-statique.

Nous allons détailler les différentes étapes nécessaire pour mettre en œuvre un calcul de contact avec frottement.

1 Construction du maillage

La première étape est de construire un maillage avec son mailleur habituel, en sous-structurant les pièces de manière assez homogène pour que la stratégie soit efficace en parallèle.

Prenons, l'exemple d'une bride à 90° sur la Figure A.1(a) comportant un seul boulon. On la décompose en plusieurs sous-structures (Figure A.1(b)) (outre les différentes pièces) et interfaces (Figure A.1(c)).

Les contraintes :

- l'ensemble des sous-structures doivent être enregistrées sous un nom générique (par exemple "V") suivi d'un numéro commençant à 0 et finissant au nombre de sous-structure moins un, par exemple, V0.avs à V10.avs si on a onze sous-structures,
- les maillages de toutes les interfaces de contact entre les sous-structures doivent être compatibles,
- dans le cas d'interface avec un jeu physique, comme une tige de boulon par rapport au trou, il est nécessaire de créer et d'enregistrer le maillage des deux interfaces. Elles devront être nommées, dans le cas d'une interface entre la sous-structure i et la sous-structure j : Ii-i-j.avs (correspondant à l'interface de la sous-structure i) et Ij-i-j.avs (correspondant à l'interface de la sous-structure j) si l'on a choisi "I" comme nom générique.

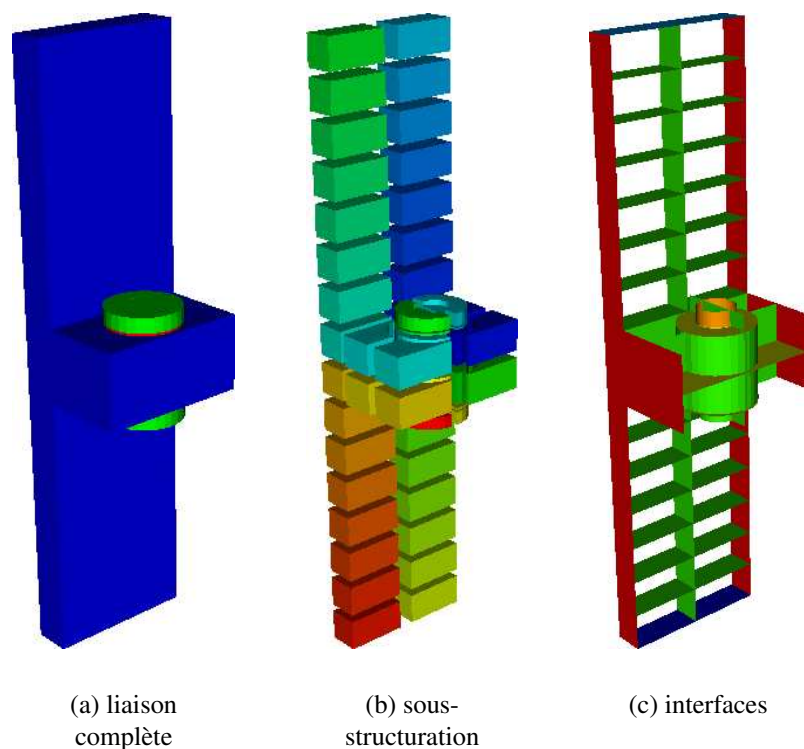


Figure A.1: Modélisation d'une bride à 90°

Les formats d'entrée de maillage disponible sont : AVS (issu de CAST3M) et NASTRAN.

2 Fabrication du fichier de données

Le fichier de données est au format xml, il doit donc contenir les balises `<?xml version="1.0" ?>` pour indiquer le format, `<root>` et `</root>` pour délimiter le fichier.

2.1 Section mesh

La section mesh du fichier permet de donner les informations sur les maillages d'entrée. Pour la bride à 90°, cette section est remplie comme ceci :

```
<mesh repertoire_des_maillages="./EXEMPLES/ALAIN/liaisons_eads/bride_90/"  
      nom_fichier_qualification_materiaux="Vqualif.py"
```

```

nom_des_maillages="V" nb_de_maillages="56" extension=".avs"
jeu_physique="1" nom_des_maillages_jeu="I"
inter_jeu="0 13;11 13;14 27;25 27;28 41;39 41;42 55;53 55" />

```

La signification des différents paramètres est donnée dans le tableau A.1.

Paramètre	Signification
repertoire_des_maillages	chemin d'accès pour les fichiers du maillage
nom_des_maillages	préfixe des noms de fichiers
nb_de_maillages	nombre de maillages à lire
extension	extension des fichiers à lire
nom_fichier_qualification_materiaux	nom du fichier de qualification des matériaux
jeu_physique	présence de jeu physique (0 ou 1)
nom_des_maillages_jeu	préfixe des interfaces
inter_jeu	liste des sous-structures en contact avec jeu physique

Tableau A.1: Paramètres de lecture des maillages

Aussi dans notre exemple, on a :

- les maillages sont dans ./EXEMPLES/ALAIN/liaisons_eads/bride_90/,
- le préfixe de ces maillages est V,
- l'extension de ces maillages est .avs
- le nombre de maillages à lire est de 56 (on va donc lire V0.avs à V55.avs)
- le fichier de qualification se nomme Vqualif.py et est situé dans le même répertoire que les maillages
- il y a des jeux physiques
- le préfixe de noms de maillages des interfaces est I
- il y a des interfaces avec jeu physique entre les sous-structures : 0 et 13, 11 et 13, 14 et 27, 25 et 27, 28 et 41, 39 et 41, 42 et 55, 53 et 55.

2.2 Section materials

La section `materials` permet de déclarer les différents matériaux disponibles pour le calcul. La variable `nom_fichier_qualification_materiaux` de la section `mesh` permet de définir le fichier de qualification, qui contient autant de lignes que le nombre de sous-structures et sur chaque ligne on note le numéro d'identification du matériau correspond à la sous-structure visée.

Les matériaux sont déclarés dans les balises suivantes :

```

<materials>
  <coefficients type="isotrope" identificateur="0" name="sst"

```

```
        elastic_modulus="69850" poisson_ratio="0.32" unit="MPa"  
        alpha="2e-6" f_vol="0;0;0"/>  
<coefficients type="isotrope" identificateur="1" name="vis"  
        elastic_modulus="210e3" poisson_ratio="0.3" unit="MPa"  
        alpha="2e-6" f_vol="0;0;0"/>  
</materials>
```

La signification des différents paramètres est donnée dans le tableau A.2.

Paramètre	Signification
type	type de formulation (isotrope)
identificateur	numéro d'identification
name	nom
elastic_modulus	module d'Young
poisson_ratio	coefficient de poisson
unit	unité
alpha	coefficient de dilatation thermique
f_vol	effort volumique à appliquer suivant les axes (x ;y ;z)

Tableau A.2: Paramètres des matériaux

Dans le cas du calcul de liaison, seules les informations `identificateur`, `elastic_modulus`, `poisson_coefficient` et éventuellement `f_vol` sont à renseigner.

Aussi dans notre exemple, on a deux matériaux différents au comportement élastique isotrope. Le fichier de qualification des matériaux permettra d'assigner à chaque sous-structure un des deux matériaux définis.

2.3 Section proprietes_interfaces

Cette section permet de renseigner les comportements d'interface particuliers. Par défaut, le logiciel va détecter et construire toutes les interfaces entre chaque sous-structures et leurs assigner un comportement de type parfait. Pour avoir, des comportements particuliers, il faut les indiquer explicitement dans le fichier.

Les comportements disponibles sont :

- `jeu_impose` pour définir un jeu imposé entre deux interfaces, comme par exemple le jeu entre un écrou et la vis qui permet de simuler le serrage,
- `contact_sst` pour définir une interface de contact entre deux sous-structures données,
- `contact_jeu_physique` pour définir une interface de contact avec jeu physique entre deux sous-structures données.

Les interfaces sont déclarées à l'aide des balises suivantes :

```
<proprietes_interfaces>
  <coefficients type="jeu_impose_sst" num_sst="12 13" jeu="0;0;0.14"
                                     name="" />
  <coefficients type="contact_jeu_physique" coeffrottement="0.08"
                                     num_sst="53 55" name="" />
  <coefficients type="contact_sst" coeffrottement="0.08" num_sst="0 11"
                                     name="" />
</proprietes_interfaces>
```

La signification des différents paramètres est donnée dans le tableau A.3.

Paramètre	Signification
type	type d'interface
num_sst	numéro des sous-structures en contact
coefficienfrottement	coefficient de frottement (si besoin)
jeu	valeur du jeu imposé entre les deux sous-structures (si besoin)

Tableau A.3: Paramètres des matériaux

Aussi dans notre exemple, on a :

- une interface de type `jeu_impose` entre les sous-structures 12 et 13. Le jeu est de 0.14 mm entre les deux sous-structures,
- une interface de type `contact_jeu_physique` entre les sous-structures 53 et 55, le coefficient de frottement est de 0.08,
- une interface de type `contact` entre les sous-structures 0 et 11, le coefficient de frottement est de 0.08.

2.4 Section CL

Cette section permet de déclarer les conditions limites du calcul. Les conditions limites disponibles sont les suivantes : déplacement imposé, effort imposé, symétrie, condition périodique.

Les conditions limites sont déclarées à l'aide des balises suivantes :

```
<CL>
  <parametres box="-100 -100 -80 100 100 -80" unit="mm" comp="depl" />
  <fct_spatiale fonction="0;0;0" />
  <fct_temporelle intervalle="0. 10000." fonction="0" />
</CL>
<CL>
  <parametres box="-100 -20 -100 100 -20 100"
```

```

        box1="-100 20 -100 100 20 100" unit="mm" comp="periodique" />
</CL>
<CL>
  <parametres box="-100 -100 80 100 100 80" unit="mm" comp="effort" />
  <fct_spatiale fonction="1/8;0;1/8" />
  <fct_temporelle intervalle="0. 0.1" fonction="0;0;0" />
  <fct_temporelle intervalle="0.1 10000."
        fonction="400*sin(3.14159265358979*(t-0.1));0;-400" />
</CL>

```

La signification des différents paramètres de la section `parametres` est donnée dans le tableau A.4.

Paramètre	Signification
<code>comp</code>	comportement d'interface
<code>box</code>	boîte définie par deux points contenant les surfaces des CL
<code>box1</code>	boîte qui permet de définir l'autre surface périodique
<code>unit</code>	unité

Tableau A.4: Paramètres des conditions limites : section `parametres`

Les interfaces de type effort et déplacement imposés sont définies en fonction du temps et de l'espace d'où la description adoptée avec des fonctions de l'espace et du temps. La fonction espace est décrite vectoriellement sur les 3 axes principaux. La fonction du temps est décrite soit de manière scalaire, soit de manière vectorielle.

La signification des différents paramètres de la section `fct_spatiale` est donnée dans le tableau A.5.

Paramètre	Signification
<code>fonction</code>	fonction spatiale $f(x,y,z)$ vectorielle $(x ; y ; z)$

Tableau A.5: Paramètres des conditions limites : section `fct_spatiale`

La signification des différents paramètres de la section `fct_temporelle` est donnée dans le tableau A.6.

Dans notre exemple, on a donc :

- des interfaces périodiques entre les faces définies par la boîte `-100 -20 -100 100 -20 100` et la boîte `-100 20 -100 100 20 100`
- des déplacements imposés sur la face définie par la boîte `-100 -100 -80 100 100 -80`, avec une fonction spatiale nulle et une fonction temporelle nulle, il s'agit donc d'une encastrement.

Paramètre	Signification
intervalle	intervalle de définition
fonction	fonction temporelle f(t) vectorielle ou scalaire

Tableau A.6: Paramètres des conditions limites : section fct_temporelle

- des efforts imposés sur la face définie par la boîte $-100 \ -100 \ 80 \ 100 \ 100$ 80 , avec une fonction spatiale égale à $1/8 ; 0 ; 1/8$ et une fonction temporelle nulle jusqu'à 0.1 s puis égale à la fonction $400 * \sin(3.14159 * (t - 0.1)) ; 0 ; -400$. C'est-à-dire que l'on impose $-400/8$ suivant \vec{z} ce qui correspond à une pré-charge statique longitudinale et $400 * \sin(\pi * (t - 0.1))/8$ suivant \vec{x} ce qui correspond à un effort tranchant sinusoidale déphasée pour débiter à 0.1 s.

2.5 Section parametres

Cette section permet de définir les différents paramètres du calcul. Un grand nombre d'options sont disponibles, nous définiront seulement celles qui peuvent être modifiées, les autres devant conserver leur valeur par défaut afin que les calculs convergent.

Les paramètres de la méthode sont déclarés à l'aide des balises suivantes :

```
<parametres sous_integration="0" type_sous_integration="p" type_base_macro="3"
opti_multi="0" erreur_macro="1e-6" multiechelle="1" blocage_modes_rigides="0"
mvts_bloques="Ty Tx Rz" nbitermax="5000" facteur_relaxation="0.8"
critere_erreur="1e-4" type_erreur="ddr" nb_threads="1" nom_calcul="latin"
deltaT="0" save_depl_SST="1" save_data="1" critere_erreur_diss="0.01"
read_data="0" reprise_calcul="0"/>
```

La signification des différents paramètres de la section parametres est donnée dans le tableau A.7.

Paramètre	Signification
nbitermax	nombre d'itérations maximales autorisés
critere_erreur	valeur de l'erreur LATIN
critere_erreur_diss	valeur de l'erreur en dissipation relative
save_data	1=sauvergarde des resultats en fin de calcul (0 sinon)
read_data	1=lecture de resultat de sauvergarde (0 sinon)
reprise_calcul	1=relance le calcul avec la sauvergarde en valeur initiale

Tableau A.7: Paramètres de l'algorithme de résolution

Dans notre exemple, on a donc :

- 5000 itérations maximales autorisées ;
- erreur LATIN vaut 10^{-4} ;
- erreur en dissipation vaut 0.01 ;
- on sauvergarde les resultats à la fin du calcul

2.6 Section `parametres_temporels`

Cette section permet de définir les différents paramètres du calcul. Un grand nombre d'options sont disponibles, nous définiront seulement celles qui peuvent être modifiées, les autres devant conserver leur valeur par défaut afin que les calculs convergent.

Les paramètres temporels sont déclarés à l'aide des balises suivantes :

```
<parametres_temporels type_de_calcul="Qstat"  
    nbpastemps="122" pasdetemps="0.05" />
```

La signification des différents paramètres de la section `parametres_temporels` est donnée dans le tableau A.8.

Paramètre	Signification
<code>type_de_calcul</code>	stat pour statique, ou Qstat pour quasi-statique
<code>nbpastemps</code>	nombre de pas de temps
<code>pasdetemps</code>	pas de temps fixe

Tableau A.8: Paramètres temporels

Dans notre exemple, on effectue donc un calcul quasi-statique avec 122 pas de temps, avec un pas de temps de 0.05 s.

2.7 Section `parametres_affichage`

Cette section permet de définir les différents paramètres d'affichage et de post-traitement.

Les paramètres d'affichage sont déclarés à l'aide des balises suivantes :

```
<parametres_affichage interactivite="1" affich_mesh="1"  
    affich_resultat="1" type_affichage="Inter" save_or_display="save"  
    display_fields="dep qtrans num_proc_skin numsst_skin sigma_skin  
    epsilon_skin num dissipation dWt dWn F" list_error="1" display_error="1"  
    repertoire_save="./tmp/bride_90/cisaillementR/f_0.08/stat_3/precharge_3/"  
    name_data="data1" command_file="bride_90_cisaillementR_stat_3" />
```

La signification des différents paramètres de la section `parametres_affichage` est donnée dans le tableau A.9.

Paramètre	Signification
<code>interactivite</code>	activation du module interactif en fin de calcul (0/1)
<code>affich_mesh</code>	affichage du maillage au début du calcul (0/1)
<code>affich_resultat</code>	affichage des resultats en fin de calcul (0/1)
<code>type_affichage</code>	Type d’affichage : Inter pour les interfaces, Sinterieur pour les SST
<code>save_or_display</code>	sauvergarde ou affichage des maillages/résultats (save/display)
<code>repertoire_save</code>	répertoire de sauvegarde
<code>display_fields</code>	liste des champs à afficher dans les résultats
<code>name_data</code>	préfixe des fichiers de résultats
<code>command_file</code>	nom du fichier de commande pour scripter l’interactivité

Tableau A.9: Paramètres d’affichage

Dans notre exemple, on a donc :

- l’interactivité est activé et on va lire le fichier de commande :
`bride_90_cisaillementR_stat_3,`
- on sauvegarde sans les afficher le maillage et les resultats des interfaces
- le prefixe de sauvegarde est `data1`
- le répertoire de sauvegarde des fichiers est
`./tmp/bride_90/cisaillementR/f_0.08/stat_3/precharge_3/`

2.8 Section `direction_recherche`

Cette section permet de définir des directions de recherche, paramètres fondamentaux de la méthode LATIN.

Les paramètres de direction de recherche sont déclarés à l’aide des balises suivantes :

```
<direction_recherche ktype="scalaire_auto_CL" kfact="1" copydirection="0" />
```

Il n’est pas nécessaire de les modifier a priori.

3 Post-traitement des résultats

Le post-traitement des résultats s’effectue par le logiciel Paraview. Le code sauvegarde les fichiers résultats au format `vtu` que peut lire ce logiciel. Pour l’utilisation du logiciel, veuillez vous reporter à la documentation officielle sur le site <http://www.paraview.org>.

La liste des champs disponibles dans les sous-structures sont définies dans le tableau A.10 et la liste de ceux disponibles sur les interfaces dans le tableau A.11.

Champ	Signification
dep	champ de déplacement
qtrans	champ de translation permettant de faire un éclaté
epsilon	champ de déformation
sigma	champ de contrainte
numsst	numéro de la sous-structure
num_proc	numéro du processeur traitant la sous-structure
typemat	numéro de l'identificateur matériau
epsilon_skin	champ de déformation sur la peau
sigma_skin	champ de contrainte sur la peau
numsst_skin	numéro de la sous-structure sur la peau
num_proc_skin	numéro du processeur sur la peau

Tableau A.10: Champs disponibles sur les sous-structures

Les sorties des sous-structures sont en fait tracées uniquement sur la peau pour alléger fortement la taille des fichiers résultats et leur temps de lecture, c'est la raison pour laquelle il y a les champs « skin » de définis.

Champ	Signification
type	type de comportement de l'interface
num	numéro de l'interface
F	effort F sur l'interface
W	déplacement (ou vitesse en quasi-statique) sur l'interface
dWn	saut de déplacement normal
dWt	saut de déplacement tangentiel
dissipation	dissipation de l'interface

Tableau A.11: Champs disponibles sur les interfaces

La liste des comportements d'interface et le numéro associé est donné dans le tableau A.12.

4 Interactivité

Le module d'interactivité permet de réaliser des opérations à la fin du calcul comme l'affichage de l'évolution du déplacement d'un point, le maillage des sous-

Type	Signification
0	interface de type déplacement
1	interface de type effort
2	interface de type symetrie
3	interface de type deplacement normal
4	interface de type parfait
5	interface de type contact ou contact avec jeu ou contact physique
6	interface de type jeu imposé
7	interface périodique
8	interface d'un autre type

Tableau A.12: Définitions des types d'interface

structures ou des interfaces, modifier les paramètres d'une interface, de relancer le calcul...

Le principe est une invite de commande devant laquelle on peut rentrer un certain nombre de commandes élémentaires. La liste des commandes disponibles est donnée dans le tableau A.13.

L'ensemble de ses commandes peuvent être placées dans un fichier de commande, qui permettra de « batcher » un certain nombre d'opération et permettra notamment d'effectuer des calculs multirésolution. Le fichier de commande doit être renseigné dans la section `parametres_affichage` argument `command_file`.

5 Utilisation du logiciel

Pour décrire l'utilisation du logiciel, on va se baser sur un exemple de calcul sur une liaison composé de un boulon avec des conditions de symétrie.

5.1 Description de l'archive

L'archive `exemple_1boulon.tar.bz2` contient l'ensemble des fichiers nécessaires, il suffit de la décompresser dans le répertoire où le logiciel JointVirt a été installé, les fichiers seront décompressés dans le dossier `EXEMPLES/1boulon/`.

Dans ce dossier, on trouvera :

- le fichier de mise en donnée relatif à ce calcul : `1boulon.xml`,
- les fichiers maillage de chaque sous-structures : `V0.avs`, `V1.avs`, ... et les fichiers maillage des interfaces avec jeu physique : `I0-0-1.avs`, `I1-0-1.avs`, ...

Commande	Signification
help	donne la liste des commandes
help cmd	donne l'aide relative à la commande cmd
evol x y z	trace le déplacement du point de coordonnée x y z
trac_error	trace l'évolution de l'erreur LATIN
mesh_inter	affiche le maillage des interfaces
mesh_sst	affiche le maillage des sous-structures
trac_inter_temps	affiche les résultats sur les interfaces
trac_sst_temps	affiche les résultats sur les sous-structures
trac_ener 0 0	trace l'énergie dissipée à l'étape n
trac_ener 0 1	trace l'énergie dissipée à l'étape n+1/2
trac_ener 1 0	trace l'énergie imposée à l'étape n
trac_ener 1 1	trace l'énergie imposée à l'étape n+1/2
modif_param_inter num	modification des paramètres de l'interface num
modif_param_inter num temps	modification des paramètres temporels
calcul	relance le calcul
exit	sort du programme

Tableau A.13: Liste des commandes du module d'interactivité

- le fichier `Vqualif.py` qui assigne le numéro du matériau de chaque sous-structure relatif à la déclaration du fichier `1boulon.xml`,
- le fichier de commande `commandfile`, qui contient un certain nombre de commande (evolution du déplacement d'un point, affichage de l'énergie imposé, énergie dissipée, ...),
- un dossier `resultats`, dans lequel tous les résultats seront sauvegardés après avoir lancé le calcul.

5.2 Utilisation standard

Une fois que le fichier de donnée au format xml est réalisée, on peut lancer le calcul en utilisant la commande suivante `./jointvirt 3 1boulon.xml` où 3 est la dimension du problème.

5.3 Utilisation en parallèle

Pour lancer le logiciel en parallèle, il est nécessaire de créer un fichier contenant le nom des ordinateurs sur lesquels le calcul va être lancé, ainsi que le nombre de processeur associé. Le fichier, nommé `lamhosts` par exemple, ressemblera à celui-ci :

```
node0 cpu=2  
node1 cpu=2  
node2 cpu=2
```

Il suffira ensuite de lancer le serveur MPI, qui permet de gérer le transit des informations en parallèle, à l'aide de commande : `lamwipe -v lamhosts ; lamboot -v lamhosts`. Puis de lancer le calcul en parallèle avec la commande : `mpirun -np 4 ./jointvirt 3 lboulon.xml mpi` pour lancer le calcul sur 4 processeurs.

Note : il faut remarque qu'il y a un processeur qui est dédié au problème multi-échelle pour accélérer la convergence des calculs, donc dans l'exemple précédent, les sous-structures seront réparties sur 3 processeurs.

5.4 Remarques

Dans la pratique, il est nécessaire de lancer le logiciel pour créer proprement son fichier de donnée xml. En effet, on créera tout d'abord un fichier sans les conditions limites et interfaces particulières, pour uniquement visualiser (option `affich_mesh="1"`) si toutes les sous-structures sont bien lues (option `type_affichage="Sinterieur"`) puis si toutes les interfaces sont bien détectées (option `type_affichage="Inter"`).

Ensuite, on pourra mettre en place les comportements d'interface particuliers ainsi que les conditions limites et on visualisera le résultat (option `affich_mesh="1"`) sur les interfaces (option `type_affichage="Inter"`) en vérifiant que les numéro de type correspondent à ce que l'on souhaitait...

L'étape suivante sera d'effectuer un calcul en statique, pour mettre en place le serrage souhaité dans les boulons. Notamment si le serrage est exprimé en MPa, il faudra mettre un premier serrage et mesurer la contrainte moyenne dans la tige de la vis. Ensuite il suffira d'utiliser la linéarité du comportement des matériaux (si c'est le cas) pour en déduire la valeur du serrage optimale par un produit en croix.

Il ne restera plus qu'à effectuer le calcul complet désiré avant de post-traiter les résultats.