

THÈSE

CALCUL MULTI-DOMAINES ET APPROCHES MULTI-ÉCHELLES POUR LA SIMULATION NUMÉRIQUE DE CRASHS AUTOMOBILES

Présentée devant
l'Institut National des Sciences Appliquées de Lyon

pour obtenir
le **GRADE DE DOCTEUR**

École doctorale :
Mécanique, Énergétique, Génie Civil, Acoustique

Spécialité :
MÉCANIQUE - GÉNIE MÉCANIQUE - GÉNIE CIVIL

par
Benjamin BOUREL
Ingénieur

Thèse soutenue le 19 décembre 2006 devant la Commission d'examen

Jury

JEZEQUEL LOUIS	Professeur	Président
DRAZETIC PASCAL	Professeur	Rapporteur
PESEUX BERNARD	Professeur	Rapporteur
WINKELMULLER GÉRARD	Ingénieur	Examineur
COMBESCURE ALAIN	Professeur	Examineur
DI VALENTIN LAURENT	Ingénieur	Examineur

Remerciements

A l'heure du bilan de ces trois années de thèse réalisées dans le cadre d'un contrat CIFRE entre le LaMCoS et PSA PEUGEOT-CITROËN[®], je débute comme à l'accoutumé ce mémoire en témoignant ma reconnaissance envers toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail de recherche.

Je tiens tout d'abord à exprimer ma profonde gratitude au professeur Alain COMBES-CURE, mon directeur de thèse, pour le soutien sans faille et l'encadrement hors pair dont il a fait preuve, malgré les longues heures de transport qui nous séparaient. Je témoigne également toute ma reconnaissance à Laurent DI VALENTIN, pour son suivi quotidien et tout aussi précieux au sein de PSA PEUGEOT-CITROËN[®]. Je tiens avant tout à les remercier de m'avoir confié ce sujet de recherche ô combien captivant qui a su me tenir en haleine durant ces trois ans, mais aussi pour la confiance qu'ils m'ont accordée durant tout ce temps. Leur esprit scientifique, leur clairvoyance et leur disponibilité à toute épreuve font d'eux deux personnes que j'admire et respecte beaucoup. Au-dessus des liens professionnels, leurs qualités humaines m'ont particulièrement marqué et m'ont fait apprécier très sincèrement ces années de travail en leur collaboration.

J'exprime ici aussi toute ma gratitude à Louis JEZEQUEL pour l'honneur qu'il m'a fait en étant président du jury, ainsi qu'à Pascal DRAZETIC et Bernard PESEUX pour avoir accepté d'être rapporteurs de ce travail.

Je tiens à remercier Gérard WINKELMULLER responsable du code de calcul RADIOSS pour son implication sur le sujet mais aussi pour avoir accepté de participer à la soutenance en temps qu'examineur. Un grand merci à Maciek WRONSKI, développeur du même code de calcul, à qui je dois le déploiement de plusieurs beta-versions indispensables à l'avancement de mes travaux.

Mes remerciements vont également à l'ensemble des personnes présentes et passées de l'équipe « Modélisation Dynamique et Sécurité Véhicule » de PSA PEUGEOT-CITROËN[®] pour l'atmosphère de travail tout aussi rare que plaisante qu'ils se sont efforcés de maintenir... Merci à Fabien dont l'œil d'expert crash m'a été d'un grand recours aux débuggages de modèles. Merci à Kiwan pour la patience, le dévouement et la grande sympathie dont il a fait preuve pour répondre à mes interrogations les plus diverses. Merci à tous les autres dont la liste serait tout aussi longue qu'impersonnelle et risquerait de conduire à d'impardonnables oublis.

Je terminerai par remercier, ceux qui m'ont supporté « au quotidien » pendant cette thèse ; merci à Nico, Sam, Christophe, Lionel, Cédric et Amandine dont la compagnie réconfortait mon univers durant cette période si particulière. Merci enfin à ma famille et mes parents grâce à qui j'ai pu m'accomplir dans cette voie. Un merci très spécial à ceux qui ont accepté de me relire alors que leur ignorance du calcul numérique était tout aussi certaine que mes lacunes en orthographe.

Table des matières

Table des matières	vii
Table des figures	xi
Liste des tableaux	xv
Introduction	1
1 Positionnement de la recherche	5
1.1 Le schéma de dynamique explicite	6
1.1.1 Principe des travaux virtuels en formulation Lagrangienne	6
1.1.1.1 Problème de référence - Équations du mouvement	6
1.1.1.2 Description Lagrangienne du mouvement	8
1.1.1.3 Formulation faible : principe des travaux virtuels	9
1.1.2 Discrétisation en espace - Méthode éléments finis	10
1.1.2.1 Approximation éléments finis	10
1.1.2.2 Équations discrètes	11
1.1.3 Discrétisation en temps - Schéma d'intégration	12
1.1.3.1 Schéma de Newmark	12
1.1.3.2 Étude de convergence	14
1.1.3.3 Schémas aux différences centrées	16
1.1.4 Algorithme de dynamique explicite	17
1.2 Méthode de décomposition de domaines	20
1.2.1 Bref état de l'art sur les méthodes multi-domaines	20
1.2.1.1 Méthode de <i>Schwarz</i>	21
1.2.1.2 Méthode de <i>Schur primale</i>	22
1.2.1.3 Méthode de <i>Schur duale</i>	24
1.2.1.4 Méthodes mixtes	25
1.2.2 Méthode multi-domaines en dynamique explicite	26
1.2.2.1 Définition du problème couplé	26
1.2.2.2 Décomposition <i>free-link</i>	26
1.2.2.3 Généralisation de l'algorithme à n_{ssd} sous-domaines	27
1.2.3 Collage des maillages incompatibles	28
1.2.3.1 Notion d'interface et de collage	28
1.2.3.2 Construction des matrices de contraintes	29
1.2.3.3 Choix de l'espace des multiplicateurs de Lagrange	30
1.2.3.4 Collage de modélisations incompatibles	32
1.2.4 Plusieurs échelles de temps	34

1.2.4.1	Motivation de la démarche	35
1.2.4.2	Algorithme multi-échelles en temps	35
1.2.4.3	Gestion des pas de temps variables	37
1.3	Implantation dans RAD2RAD	39
1.3.1	Organisation du calcul couplé	39
1.3.1.1	Mécanisme de communication	40
1.3.1.2	Structuration des données RAD2RAD	41
1.3.1.3	Données échangées	42
1.3.2	Exemple de calcul couplé	43
1.3.2.1	Corps creux en compression	43
1.3.3	Traitement du problème d'interface	45
1.3.3.1	Optimisation de l'algorithme multi-échelles en temps	45
1.3.3.2	Coût de résolution du problème condensé aux interfaces	46
1.3.3.3	Tests de solveurs	48
1.3.4	Calculs couplés et parallélisme	49
1.3.4.1	Brefs rappels sur le calcul parallèle	49
1.3.4.2	Parallélisation des calculs couplés	51
2	Gestion des non-linéarités de contact	53
2.1	Étude du contact mono-domaine	55
2.1.1	Définition du contact	55
2.1.1.1	Notion d'interface de contact	55
2.1.1.2	Condition de contacts	56
2.1.1.3	Loi de frottement	57
2.1.2	Formulation du problème de contact	58
2.1.2.1	Prise en compte des contraintes	58
2.1.2.2	Formulation faible du problème de contact par pénalité	61
2.1.2.3	Formulation discrète	62
2.1.2.4	Remarques sur la régularisation des lois de contact	63
2.1.3	Traitement numérique du contact dans les codes explicites	64
2.1.3.1	Modélisation de l'interface de contact	65
2.1.3.2	Réduction de la pénétration	66
2.1.3.3	Contact en dynamique explicite	68
2.1.3.4	Nouvelles conditions de stabilité	69
2.2	Développement d'une méthode de contact inter-domaine	71
2.2.1	Bref état de l'art sur le contact multi-domaines	71
2.2.1.1	La méthode FETI-C	71
2.2.1.2	La méthode CONTRAST	72
2.2.1.3	Algorithme de Newton-Schur	74
2.2.2	Méthode de la <i>peau artificielle</i>	76
2.2.2.1	Stratégie de recouvrement/duplication	76
2.2.2.2	Liaison rigide entre les noeuds <i>esclaves</i> et <i>copiés</i>	77
2.2.2.3	Démarche générale de calcul du contact inter-domaines	79
2.2.3	Extension aux liaisons permanentes	80
2.2.3.1	Interface de collage rigide	80
2.2.3.2	Comparaison des méthodes <i>primale/duale</i>	80
2.3	Collage temporel dans la méthode de couplage dynamique	83
2.3.1	Communications à pas de temps unique	83

2.3.1.1	Pas de temps	83
2.3.1.2	Continuité cinématique à l'interface	83
2.3.1.3	Masses et raideurs nodales des noeuds <i>copiés</i>	84
2.3.2	Gestion des pas de temps asynchrones	84
2.3.2.1	Interpolation en temps	84
2.3.2.2	Problème de continuité cinématique	86
2.3.2.3	Correction des accélérations	87
2.3.3	Implémentation à RAD2RAD	88
2.3.3.1	Algorithme d'avancé en temps généralisé à n domaines	88
2.3.3.2	Schéma de communications RAD2RAD-RADIOSS	90
2.4	Exemples numériques	92
2.4.1	Chute d'une poutre	92
2.4.1.1	Description du cas test	92
2.4.1.2	Étude cinématique	93
2.4.2	Impact d'un corps creux	94
2.4.2.1	Description du cas test	94
2.4.2.2	Résultats du calcul	95
2.4.3	Simulation d'un choc piéton	97
2.4.3.1	Description du cas test	97
2.4.3.2	Validation des résultats	97
2.4.3.3	Gain en temps CPU	99
3	Calculs dynamiques à discrétisation variable	101
3.1	État de l'art des approches multi-échelles	103
3.1.1	L'approche multi-grilles	103
3.1.1.1	Procédure multi-grille classique	103
3.1.1.2	Adaptation aux calculs multi-domaines	105
3.1.2	La méthode <i>Arlequin</i>	106
3.1.2.1	Principes de la méthode	106
3.1.2.2	Formulation <i>Arlequin</i>	108
3.1.2.3	Formulation discrète	108
3.1.3	L'approche adaptative	109
3.1.3.1	Éléments finis hiérarchiques	109
3.1.3.2	Méthode de raffinement local automatique	111
3.2	Méthode de basculement de maillage en linéaire	114
3.2.1	Principe du basculement de maillage	114
3.2.1.1	Cadre de l'étude - Limites de l'approche proposée	114
3.2.1.2	Problématique	115
3.2.2	Opérateurs de changement d'échelles	115
3.2.2.1	Techniques de transfert de champs	115
3.2.2.2	Transformation isoparamétrique inverse	116
3.2.2.3	Calcul de l'opérateur de projection	118
3.2.3	Insuffisance de la projection	119
3.2.3.1	Étude de stabilité de l'algorithme à discrétisation variable	119
3.2.3.2	Interprétation du déséquilibre	121
3.2.4	Méthode de rééquilibrage dynamique	122
3.2.4.1	Recouvrement d'échelles	123
3.2.4.2	Cycle d'équilibrage « implicite »	123

3.3	Extension au cas non-linéaire	125
3.3.1	Méthode de rééquilibrage itérative	125
3.3.1.1	Résolution du pas de temps d'équilibre	125
3.3.1.2	Initialisation de la procédure	127
3.3.2	Critère de changement de maillage	127
3.3.2.1	Critère de raffinement	128
3.3.2.2	Critère de déraffinement	128
3.3.3	Implémentation de la méthode	128
3.3.3.1	Algorithme général en grands déplacements et grandes rotations	129
3.3.3.2	Adaptation aux calculs couplés RAD2RAD-RADIOSS	130
3.4	Exemples numériques	133
3.4.1	Cas 2D : Poutre en flexion	133
3.4.1.1	Validation de la démarche d'équilibrage	134
3.4.1.2	Influence du recouvrement	136
3.4.1.3	Influence du critère de raffinement	137
3.4.1.4	Influence du maillage	138
3.4.2	Cas 3D : Flambage de prolonge	139
3.4.2.1	Validation du calcul avec raffinement	140
3.4.2.2	Confrontation précision - gain de temps CPU	141
	Conclusions et perspectives	145
	Bibliographie	149

Table des figures

1.1	Définition du problème de référence.	7
1.2	Configuration non-déformée et déformée du système.	8
1.3	Discrétisation du domaine en éléments finis.	10
1.4	Algorithme de dynamique explicite.	18
1.5	Décomposition de domaine - Méthode de <i>Schwarz</i>	21
1.6	Décomposition de domaine - Méthode de <i>Schur primale</i>	23
1.7	Liaison de collage permanent.	29
1.8	Calcul des intégrales de collage (cas d'une interface linéique).	30
1.9	Méthode des multiplicateurs de Lagrange localisés.	31
1.10	Choix de l'espace des multiplicateurs de Lagrange.	32
1.11	Couplage coque-poutre.	33
1.12	Principe du calcul à pas de temps multiples.	34
1.13	Échelles de temps locales et absolue.	35
1.14	Principe du couplage RADIOSS-RADIOSS via le module RAD2RAD.	40
1.15	Illustration des entités de calcul PART, LINK et INTER.	41
1.16	Exemple de structures de données RAD2RAD.	42
1.17	Corps creux en compression : comparaison des déformées.	44
1.18	Corps creux en compression : déplacements absolus à l'interface.	44
1.19	Corps creux en compression : vitesses absolues à l'interface.	45
1.20	Évolutions du nombre d'opérations élémentaires.	47
1.21	Schémas d'architectures parallèles.	50
2.1	Problématique de gestion du contact pour les calculs multi-domaines.	54
2.2	Définition de l'interface de contact.	55
2.3	Fonction d'interpénétration de Ω_2 dans Ω_1	56
2.4	Illustration de la loi de <i>Coulomb</i>	58
2.5	Régularisation des lois de contact.	64
2.6	Modélisation de l'interface de contact.	65
2.7	Calcul des efforts de contacts normaux.	67
2.8	Principe de la méthode CONTRAST.	72
2.9	Procédure itérative de la méthode CONTRAST.	73
2.10	Algorithme de Newton-Schur pour les problèmes de contact frottant.	75
2.11	Principe du recouvrement par duplication des noeuds d'interfaces.	76
2.12	Décomposition du domaine de calcul.	77
2.13	Comparaison des collages avec et sans multiplicateurs de Lagrange.	80
2.14	Corps creux en compression : comparaison des déformées.	81
2.15	Corps creux en compression : vitesse résultante à l'interface.	82
2.16	Échange des accélérations en mono-échelle.	83

2.17	Échange des accélérations en multi-échelles : calcul coté <i>maître</i>	85
2.18	Échange des accélérations en multi-échelles : calcul coté <i>esclave</i>	85
2.19	Impact d'une masse sur un plan.	86
2.20	Déplacements normaux des noeuds <i>esclave</i> et <i>copié</i>	86
2.21	Vitesses normales des noeuds <i>esclave</i> et <i>copié</i>	87
2.22	Accélérations normales des noeuds <i>esclave</i> et <i>copié</i>	87
2.23	Algorithme générale de collage en temps.	89
2.24	Algorithme général de calcul du contact inter-domaines.	91
2.25	Chute d'une poutre : décomposition du domaine de calcul.	93
2.26	Évolution du déplacement vertical à l'extrémité de la poutre.	93
2.27	Évolution de la vitesse verticale à l'extrémité de la poutre.	93
2.28	Évolution de l'accélération verticale à l'extrémité de la poutre.	94
2.29	Impact d'un corps creux : décomposition du domaine de calcul.	95
2.30	Écrasement du tube sur une plaque encastree.	95
2.31	Évolution de la vitesse résultante au noeuds N	96
2.32	Évolution de l'effort de contact normal entre le TUBE et la PLAQUE.	96
2.33	Évolution du pas de temps de stabilité de chaque modèle.	96
2.34	Simulation d'un choc piéton : décomposition du domaine de calcul.	97
2.35	Comparaison des accélérations au centre de l'impacteur tête.	98
2.36	Comparaison des efforts de contact à l'interface « Bras-Grille ».	98
2.37	Comparaison des énergies internes.	99
3.1	Principe de l'approche multi-grilles par sous-domaine.	105
3.2	Méthode <i>Arlequin</i> : superposition de deux modèles	107
3.3	Exemple d'élément finis hiérarchique unidimensionnel.	110
3.4	Méthode de raffinement automatique local.	111
3.5	Changement de maillage en calcul multi-domaines.	114
3.6	Schéma d'évolution entre les instants t^s et t^{s+1}	115
3.7	Construction des opérateurs de projection	116
3.8	Découpage d'un quadrangle linéaire.	118
3.9	Précontrainte des noeuds projetés intermédiaires.	121
3.10	Exemple 1D : Poutre encastree/libre en flexion.	122
3.11	Schéma d'évolution entre les instants t^s et t^{s+1}	123
3.12	Recouvrement des échelles de temps avant raffinement.	123
3.13	Algorithme général du calcul à discrétisation variable.	129
3.14	Enrichissement de l'espace des multiplicateurs de Lagrange.	131
3.15	Schéma simplifié du calcul multi-domaines à discrétisation variable.	132
3.16	Cas test de poutre en flexion.	133
3.17	Changement de discrétisation de la poutre.	134
3.18	Évolution de la vitesse transverse à l'extrémité de la poutre.	134
3.19	Évolution de l'énergie de déformation.	135
3.20	Comparaison des distributions de contraintes équivalentes.	135
3.21	Convergence de la procédure itérative de Newton-Raphson.	135
3.22	Influence du paramètre de recouvrement	136
3.23	Influence du critère de raffinement	137
3.24	Évolution de l'erreur relative avec le critère de raffinement.	137
3.25	Courbes précision/temps de calcul pour différents maillages.	139
3.26	Changement de discrétisation de la prolonge.	139

3.27	Évolution de la vitesse axiale au point P	140
3.28	Évolution de l'énergie de déformation.	141
3.29	Comparaison des distributions de contraintes équivalentes.	141
3.30	Évolution du déplacement axial pour différents critères de raffinement. . .	142
3.31	Évolution du temps CPU avec le critère de raffinement.	142

Liste des tableaux

1.1	Corps creux en compression : paramètres de calcul.	43
1.2	Comparaison des tailles et pas de temps des différents modèles.	44
1.3	Détail des temps CPU consommés par les différents modèles.	45
1.4	Nombre d'opérations pour une résolution directe de Cholesky.	47
1.5	Comparaison des temps CPU pour différents solveurs d'interface.	49
2.1	Détail des temps CPU consommés par les différents modèles.	81
2.2	Simulation d'impact d'une masse sur un plan.	86
2.3	Chute d'une poutre : paramètres de calcul.	92
2.4	Impact d'un corps creux : paramètres de calcul.	94
2.5	Comparaison des tailles et pas de temps des différents modèles.	97
2.6	Comparaison des temps CPU.	99
3.1	Cas test de poutre en flexion : caractéristiques matériaux.	133
3.2	Temps de calcul pour différentes valeurs du critère de raffinement.	138
3.3	Cas test de flambage de prolonge : caractéristiques matériaux.	140

Introduction

AU cours des 30 dernières années, les statistiques sur les accidents de la route élaborées au niveau Européen, font état d'une réduction de moitié du taux de mortalité. Cette réduction, résultat d'une progression très significative de la sécurité, est d'autant plus remarquable que le nombre de véhicules en circulation a été multiplié par près de 2,5 sur cette zone. Aujourd'hui et plus que jamais, les efforts de R&D en matière de sécurité sont poursuivis avec la même efficacité notamment par les constructeurs automobiles qui font d'une de leur principale préoccupation la sécurisation des véhicules tant vis-à-vis des passagers que vis-à-vis des piétons. En dehors des tests d'homologation indispensables à la commercialisation, un certain nombre d'organismes consommateurs incitent alors les constructeurs à soumettre leurs produits à des sollicitations de plus en plus sévères. Tel est le cas sur le marché européen des tests EuroNCap dont les fameuses étoiles jugent du comportement de la structure habitable en cas de choc (frontal ou latéral) et donc du degré de sécurité qu'elle procure à ses usagers. Au delà de son caractère officiel, ce label constitue aujourd'hui pour les marques un atout commercial et marketing de première importance auprès des clients pour lesquels, la qualité et le niveau de sécurité s'avèrent être un des critères d'achat de plus en plus décisif.

Face à cette nécessité en matière de santé publique, PSA PEUGEOT-CITROËN[®] s'emploie à intégrer ce que nous appellerons sécurité passive très tôt dans les études de conception. Il s'appuie pour cela sur ce que l'on appelle un cycle de développement en « V » dont la branche descendante est consacrée aux étapes de spécifications et la branche ascendante aux étapes de validations du produit. Le but est donc de prendre en compte ce facteur déterminant très en amont du cycle, dès l'expression du besoin (dans ce cas, l'obtention de 4 ou 5 étoiles EuroNCAP par exemple). On en déduit les critères spécifiques à atteindre pour chaque sous-système et les choix techniques susceptibles d'y répondre de manière à concevoir chacune des pièces du véhicule.

La simulation numérique lors de cette phase de conception amont est aujourd'hui devenue indispensable pour tenir les délais et coûts de développement dictés par la concurrence du marché. Son rôle est alors primordial puisqu'il permet une validation progressive du produit et évite les remises en questions tardives de solutions technologiques. Bien que les récents progrès réalisés dans le domaine permettent aujourd'hui de prédire le comportement d'une structure avec une grande précision, elle ne dispense évidemment pas d'essais physiques. Elle permet néanmoins de mieux les préparer en optimisant les prototypes et donc d'en réduire considérablement leur nombre durant la phase de validations et de spécifications. On mesure alors l'intérêt du numérique dans le domaine du crash où les essais réels, nécessitant la réalisation et la perte du prototype, sont longs et extrêmement

onéreux. Si il y a encore quelques années, la majorité des constructeurs utilisait 3 phases de lancement de prototypes réels, ils n'en délivrent désormais plus qu'une seule appelée SRU (Support Représentatif Unique) en fin de cycle de développement. Naturellement, les moyens consacrés au numérique n'ont fait qu'augmenter au cours de ces mêmes dernières années, et si dans les années 90 une dizaine d'ingénieurs PSA PEUGEOT-CITROËN[®] se consacraient aux crashes-tests numériques, ils sont aujourd'hui près de 140 personnes à utiliser des modèles de plus en plus fidèles, mais aussi de plus en plus lourds. L'exigence à l'égard des modèles numériques croît en même temps que l'intérêt et la confiance que les ingénieurs souhaitent leur accorder. En effet, la méthode des éléments finis sur laquelle repose traditionnellement le calcul des structures emploie aujourd'hui des modèles de véhicule comprenant de 500 000 à 1 000 000 éléments alors qu'ils n'en comprenaient à peine 100 000 à l'avènement de cette discipline.

Pour promouvoir cette montée en puissance du numérique, les constructeurs doivent faire face à un problème majeur qui se pose en terme de ressources de calcul. Parmi les différentes prestations qu'explorent les projets automobiles, la simulation de crash constitue sans doute le plus gros consommateur de ressources, avec près de la moitié du potentiel qui lui est dédié. Évidemment l'augmentation du nombre d'utilisateurs mais aussi de la taille des modèles s'accompagne des évolutions de solutions matérielles, notamment avec l'essor des supercalculateurs parallèles. Le groupe PSA PEUGEOT-CITROËN[®] a multiplié par 30 sa puissance de calcul dans les cinq dernières années et dispose désormais d'une capacité telle que, le plus long des calculs ne dépasse pas les 48 heures. Pourtant, la charge de calcul est telle que cela reste insuffisant devant l'incessante volonté d'explorer le maximum de voies techniques dans des délais de plus en plus courts. C'est pourquoi, on se tourne aujourd'hui vers d'autres solutions, en proposant notamment de nouvelles méthodes numériques moins coûteuses. Parmi celles-ci, il s'en s'illustre une certaine catégorie que l'on qualifie de méthodes multi-échelles.

Notre problématique s'inscrit donc dans le caractère fortement multi-échelles des simulations de crash. Les phénomènes observés y sont très complexes et peuvent être très différents dans le temps et dans l'espace. Au cours d'un choc frontal par exemple, les phénomènes observés (contraintes, déformations, énergie dissipée...) à l'avant et à l'arrière du véhicule ne s'évaluent pas sur les mêmes échelles. De même, l'échelle de temps représentative de l'impact est très différente de celle représentative du retour élastique et des vibrations qui en résultent. L'approche traditionnelle du calcul des structures ignore bien souvent cette coexistence d'échelles pour se focaliser sur l'évaluation des réponses, la plus détaillée possible, en espace et en temps. La méthode des éléments finis classique permet certes en utilisant une discrétisation très raffinée d'obtenir un niveau de détail très fin mais cela se fait au détriment des temps CPU et de l'allocation mémoire nécessaire. D'autre part, il convient de rappeler que l'analyse dynamique de la structure requiert une intégration temporelle, qui se traduit par une discrétisation de l'échelle de temps. Le schéma explicite utilisé pour les problèmes de crash, a la particularité d'être conditionnellement stable, ce qui lui confère un pas de temps critique à ne pas dépasser. L'utilisation d'un tel schéma entraîne le couplage des échelles de temps et d'espace puisque cette valeur critique dictée par la condition de CFL (Courant-Friedrich-Levy) est imposée par le plus petit élément du modèle. Ainsi le raffinement même très localisé du maillage, pénalise l'ensemble du calcul par l'utilisation d'un pas de temps très faible. Ceci constitue la véritable problématique de la simulation numérique de crash, où les

modèles de véhicules, résultant de l'assemblage de pièces aux tailles de maille souvent très hétérogènes, n'évoluent jamais avec leur pas de temps optimal.

Dans ce contexte, la question que l'on se pose aujourd'hui est la suivante : comment faire évoluer les méthodes numériques pour réduire le temps de calcul de la solution dynamique sur des modèles détaillés sans en altérer la qualité ? L'objectif de ce travail est donc de fournir une stratégie de calcul efficace et optimale permettant d'appliquer à l'échelle de chaque phénomène mis en jeu la modélisation la plus pertinente. Cette stratégie sera basée sur une méthode de sous-structuration où chaque sous-structure dispose de la discrétisation la plus adéquate. L'idée sous-jacente est de pouvoir concentrer la puissance de calcul sur les zones du véhicule les plus sollicitées, en adoptant pour celles-ci un maillage nettement plus fin et par conséquent un pas de temps nettement plus faible. L'une des principales difficultés est alors de tenir compte des éventuelles incompatibilités de maillages aux interfaces, mais aussi des incompatibilités temporelles puisque chacun des sous-domaines dispose désormais de ces propres instants d'équilibre discret. Pour parvenir à cette approche de calcul multi-domaines, nous poursuivrons les travaux entrepris par B. Herry [Her02], lequel s'était tout particulièrement intéressé à la gestion des interfaces dites de « liaisons permanentes » entre sous-domaines. Outre un certain nombre de travaux liés à la validation et à l'optimisation de l'algorithme proposé, la présente thèse de doctorat s'articule autour de deux principaux axes de recherche :

La gestion du contact inter-domaines : Le premier volet de la thèse a pour but de développer une méthode assurant la gestion du contact unilatéral (avec ou sans frottement) entre les sous-domaines issus de la sous-structuration. L'intérêt est de permettre à deux sous-domaines distincts de se rencontrer (s'impacter) au cours d'un calcul multi-domaines. Ce développement est capital pour l'industrialisation de l'approche multi-domaines liée aux applications crash, où l'on ne peut exclure a priori des contacts de ce type. Les conditions de contact unilatéral, dictées par les lois de *Signorini*, seront intégrées aux modèles numériques par la méthode de pénalité, très prisée des codes de dynamique explicite.

Le calcul explicite à discrétisation variable : Le second volet de la thèse concerne la mise en place d'une stratégie de calcul à discrétisation variable, dite multi-échelles, pour la dynamique explicite. Ce second sujet proche des méthodes de raffinement adaptatif s'inscrit toujours dans la même problématique de réduction des temps de calcul. Le but sera ici d'alterner l'utilisation de maillages plus ou moins fins afin d'adopter la discrétisation du sous-domaine la plus adaptée aux différentes phases de la simulation. La difficulté réside alors dans la façon de transférer les grandeurs statiques et cinématiques entre les différents maillages et surtout dans la manière de rééquilibrer la solution avant la reprise du calcul sur la nouvelle discrétisation.

Ce mémoire de thèse comprendra trois parties. Dans le premier chapitre nous nous intéresserons au cadre de la recherche et aux principaux concepts indispensables à la bonne compréhension de ce document. Après avoir rappelé les fondements de la dynamique explicite, nous reviendrons sur les différents aspects de la méthode de décomposition de domaine développée par B. Herry. Cette dernière sera notamment illustrée par la présentation du programme de couplage RAD2RAD dans lequel elle est implémentée. Dans le

second chapitre, nous nous intéresserons à la problématique du contact unilatéral dans le cadre des approches de calculs multi-domaines. Après avoir introduit les conditions de contact, et la manière dont elles sont généralement prises en compte dans un code de dynamique explicite, nous développerons la stratégie de contact inter-domaines mise en oeuvre. Nous nous appuierons ensuite sur quelques exemples d'application pour valider cette approche et montrer le gain du calcul multi-domaines qu'elle apporte par rapport à une méthode de calcul mono-domaine traditionnelle. Enfin, le troisième chapitre sera consacré à la méthode de calcul à discrétisation variable. Bien que cette méthode soit destinée à l'approche multi-domaines, nous nous focaliserons ici essentiellement au « basculement » de maillage propre à un sous-domaine. Après avoir dressé un bref état de l'art sur les différentes approches multi-échelles, nous détaillerons la méthode mise en oeuvre dans le cadre des problèmes linéaires, puis non-linéaires. Comme pour les deux précédents chapitres, nous terminerons par illustrer nos travaux sur quelques exemples d'application.

Chapitre 1

Positionnement de la recherche

Sommaire

1.1	Le schéma de dynamique explicite	6
1.1.1	Principe des travaux virtuels en formulation Lagrangienne . . .	6
1.1.2	Discretisation en espace - Méthode éléments finis	10
1.1.3	Discretisation en temps - Schéma d'intégration	12
1.1.4	Algorithme de dynamique explicite	17
1.2	Méthode de décomposition de domaines	20
1.2.1	Bref état de l'art sur les méthodes multi-domaines	20
1.2.2	Méthode multi-domaines en dynamique explicite	26
1.2.3	Collage des maillages incompatibles	28
1.2.4	Plusieurs échelles de temps	34
1.3	Implantation dans RAD2RAD	39
1.3.1	Organisation du calcul couplé	39
1.3.2	Exemple de calcul couplé	43
1.3.3	Traitement du problème d'interface	45
1.3.4	Calculs couplés et parallélisme	49

L'analyse éléments finis apparaît souvent comme une composante essentielle à la simulation numérique dont les nombreux avantages ont déjà largement été évoqués en introduction de ce mémoire. On la retrouve naturellement dans les problèmes de dynamique rapide, discipline particulière où les échelles de temps mises en jeu sont relativement courtes et les comportements généralement fortement non-linéaires. De nombreux efforts de recherche ont été faits dans ce domaine pour améliorer les performances des algorithmes mais aussi pour les adapter aux nouvelles architectures informatiques sur lesquelles elles reposent. Parmi les progrès réalisés, un des plus importants concerne sûrement l'apparition des schémas d'intégration en temps explicite dont la mise en oeuvre informatique affiche des spécificités très attrayantes pour ce type de problème. Un certain nombre de logiciels commerciaux, tel que RADIOSS, ont alors vu le jour et sont maintenant largement utilisés pour les études de dynamique non-linéaire.

Ce premier chapitre a pour principal but de positionner les recherches présentées dans le deuxième et troisième chapitre. Nous reviendrons pour cela sur un certain nombre de

travaux existants, notamment ceux réalisés dans [Her02], pour lesquels nos recherches constituent un aboutissement et un prolongement logique.

Dans la première partie, nous commencerons par introduire les bases de la formulation éléments finis du principe des puissances virtuelles conduisant à la discrétisation spatiale de l'équation d'équilibre. Nous nous focaliserons alors sur les schémas d'intégrations temporelles et notamment sur le formalisme explicite. Dans une seconde partie nous dresserons un bref état de l'art sur les techniques de décompositions de domaine, pour parvenir à la formulation duale à la base de notre approche multi-domaines dynamique. Nous aborderons ensuite les récents développements sur la gestion des incompatibilités spatiales et temporelles qui constitue un premiers pas vers le calcul multi-échelles. Enfin, nous terminerons ce chapitre par quelques considérations sur l'implantation de la méthode dans RAD2RAD, code sur lequel repose la plupart des travaux développés dans ce mémoire. Après avoir illustré son fonctionnement par quelques premiers exemples d'application, nous discuterons des pistes envisagées pour optimiser le problème d'interface qu'il est supposé traiter.

1.1 Le schéma de dynamique explicite

De manière générale, le système mécanique analysé est idéalisé en adoptant le point de vue de la mécanique des milieux continus puis discrétisé par la méthode des éléments finis. Par ailleurs, l'étude des phénomènes dynamiques transitoires exigent en général une approche incrémentale pour intégrer sur le temps le système d'équations différentielles issues du problème discrétisé. Ceci est d'autant plus vrai dans le cadre non-linéaire, tel que celui des grandes déformations où la complexité des chemins de déformations rencontrés et des lois de comportement utilisées, rend l'approche incrémentale indispensable.

1.1.1 Principe des travaux virtuels en formulation Lagrangienne

Revenons tout d'abord sur les concepts fondamentaux de l'analyse éléments finis en mécanique non-linéaire et notamment sur le principe des travaux virtuels sur laquelle elle repose.

1.1.1.1 Problème de référence - Équations du mouvement

Considérons un problème structural classique portant sur un système solide occupant un domaine Ω de frontière Γ dans l'espace. Ce système, vu comme milieu continu, est soumis à une densité d'effort volumique b , à une densité d'effort surfacique \bar{t} sur la partie Γ_t de sa frontière Γ , ainsi qu'à un déplacement imposé \bar{u} sur la partie restante Γ_u de Γ (Figure 1.1).

De manière générale, le comportement mécanique du milieu peut être décrit par les variables dépendantes que sont : le champ de déplacement $u(X, t)$, le champ de contrainte de Cauchy $\sigma(X, t)$, et le champ de densité $\rho(X, t)$. Toutes ces variables sont fonctions des coordonnées matérielles X du point considéré de Ω et du temps $t \in [0, T]$. Nous supposons ici une transformation isotherme. Sous cette hypothèse la résolution d'un problème de mécanique non-linéaire revient à chercher le champ de déplacement $u(X, t)$ définis sur $\Omega \times [0, T]$ et vérifiant :

⇒ Conditions limites

$$u(X, t) = \bar{u} \quad \forall (X, t) \in \Gamma_u \times [0, T] \quad (1.1.1)$$

$$\sigma(X, t)n(X, t) = \bar{t} \quad \forall (X, t) \in \Gamma_f \times [0, T] \quad (1.1.2)$$

⇒ Conditions initiales

$$u(X, 0) = u_0(X), \quad \frac{\partial u}{\partial t}(X, 0) = v_0(X) \quad \forall X \in \Omega \quad (1.1.3)$$

⇒ Équation d'équilibre dynamique

$$\rho(X, t) \frac{\partial^2 u}{\partial t^2}(X, t) = b + \text{div}[\sigma(X, t)] \quad \forall (X, t) \in \Omega \times [0, T] \quad (1.1.4)$$

⇒ Relation de comportement

$$\sigma(X, t) = S[u(X, t), \frac{\partial u}{\partial t}(X, t), \dots] \quad \forall (X, t) \in \Omega \times [0, T] \quad (1.1.5)$$

Remarquons d'ores et déjà que cette formulation du problème, appelée formulation forte, peut faire intervenir des non-linéarités de différentes natures. On distingue généralement les non-linéarités matérielles (elasto-plasticité, viscoplasticité, endommagement...), les non-linéarités géométriques (grand déplacement, grande déformation) et les non-linéarités d'interfaces (conditions limites variables, chocs, contact, frottements...). Nous ignorons pour le moment ce dernier type de non-linéarité qui sera largement discuté au second chapitre.

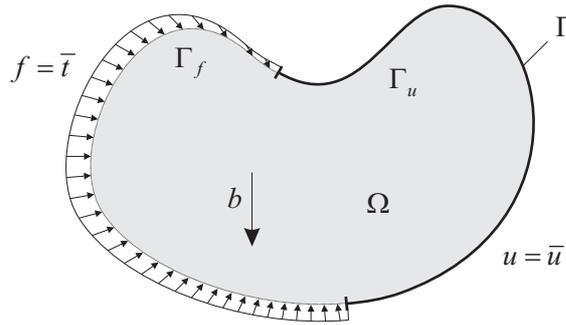


Figure 1.1: Définition du problème de référence.

En grandes déformations, le volume qu'occupe la structure subit des variations non négligeables au cours du temps. Le système d'équation (1.1.1) à (1.1.5) doit donc être exprimé sur la configuration courante (ou déformée) Ω à l'instant t plutôt que sur la configuration d'origine Ω_0 .

1.1.1.2 Description Lagrangienne du mouvement

Comme souvent en mécanique du solide, nous adopterons une approche Lagrangienne pour décrire le mouvement du système. Les coordonnées matérielles (indépendantes du temps), que nous avons notées X correspondent au vecteur position sur la configuration de référence Ω_0 , et les coordonnées spatiales x au vecteur position sur la configuration courante Ω . On introduit la fonction Φ tel que :

$$x = \Phi(X, t) \quad (1.1.6)$$

En formulation Lagrangienne, on s'attache à décrire le mouvement des points matériels en fonction des deux variables indépendantes X et t . Le déplacement d'un point s'écrit alors comme la différence entre sa position courante et sa position originale :

$$u(X, t) = \Phi(X, t) - \Phi(X, 0) = x - X \quad (1.1.7)$$

La déformation du milieu est caractérisée par le gradient déformation, noté F , qui s'exprime par :

$$F = \frac{\partial \Phi}{\partial X} = \frac{\partial x}{\partial X} \quad (1.1.8)$$

On peut alors soit toujours se référer à la configuration initiale, c'est l'approche de type Lagrangien total, soit se référer à la dernière configuration courante, c'est l'approche dite Lagrangien réactualisé [BLM00]. Cette seconde approche est souvent utilisée en dynamique explicite, où la configuration courante est connue au moment de la résolution de l'équilibre. L'intérêt de ce formalisme réside dans sa simplicité et dans la possibilité de prendre directement en compte les non-linéarités en faisant l'hypothèse des petites perturbations.

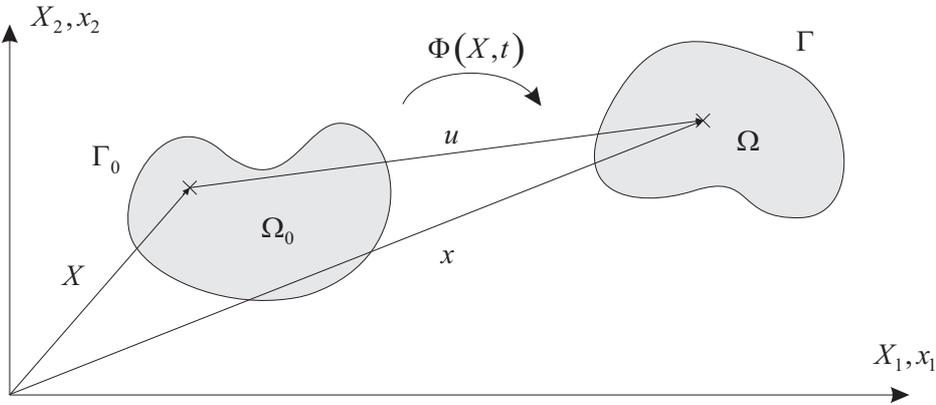


Figure 1.2: Configuration non-déformée et déformée du système.

Remarque 1 : Différentes mesures des déformations et contraintes peuvent être utilisées suivant que l'on utilise l'approche Lagrangien total ou Lagrangien réactualisé [Bel83]. La formulation Lagrangienne totale utilise souvent le tenseur de déformations de Green-Lagrange E avec le tenseur de contraintes associé, le second tenseur de Piola-Kirchoff Π .

$$E = \frac{1}{2} (F^t F - I) \quad (1.1.9)$$

$$\Pi = \det(F) F^{-1} \cdot \sigma \cdot F^{-t} \quad (1.1.10)$$

Par contre, en formulation Lagrangienne réactualisé, on introduit le tenseur de déformation d'Almansi-Euler E_{AE} qui est l'équivalent du tenseur de Green Lagrange calculé sur la configuration courante. La contrainte est alors directement mesuré par le tenseur de Cauchy σ :

$$E_{AE} = (F^{-1})^t \cdot E \cdot F^{-1} \quad (1.1.11)$$

Le passage à la formulation faible présenté ci-dessous sera établi en adoptant l'approche Lagrangienne réactualisée. Nous verrons alors que dans ce cas, la dérivation se fait par rapport aux coordonnées spatiales x et l'intégration a lieu sur la configuration déformée.

1.1.1.3 Formulation faible : principe des travaux virtuels

L'équation d'équilibre (1.1.4) ne peut pas être discrétisée directement par la méthode des éléments finis. Afin de discrétiser cette équation, une formulation faible, aussi appelée forme variationnelle, est nécessaire. On considère ici un champ de déplacement $u(X, t)$ cinématiquement admissible vérifiant les conditions limites (1.1.1). De plus, ce champ doit être suffisamment continu pour que ses dérivées spatiales dans l'équation d'équilibre soient correctement définies. L'espace fonctionnel des solutions $u(X, t)$ vérifiant ces deux conditions est un espace de Sobolev [Hug01] noté \mathcal{U} :

$$\mathcal{U} = \{u(X, t) | u(X, t) \in C^0(X), u = \bar{u} \text{ sur } \Gamma_u\} \quad (1.1.12)$$

Le passage à la formulation recourt à l'introduction des fonctions tests $\delta u(X)$, encore appelées déplacements virtuels. Ces fonctions sont aussi définies sur un espace fonctionnel \mathcal{U}_0 similaire à \mathcal{U} excepté pour ce qui est de la condition limite ($\delta u = 0$) :

$$\mathcal{U}_0 = \{\delta u(X) | \delta u(X) \in C^0(X), \delta u = 0 \text{ sur } \Gamma_u\} \quad (1.1.13)$$

Le tenseur des contraintes de Cauchy σ , fonction du tenseur des déformations (lui même issue de la dérivation en espace des déplacements) via la relation de comportement (1.1.5), est quant à lui supposé $C^1(X)$ en espace.

Afin d'alléger les écritures, nous omettrons de préciser par la suite la dépendance des fonctions scalaires, vectorielles et de tensorielles aux variables d'espace et de temps (X, t) . Nous utiliserons également la notation simplifiée des dérivées temporelles, comme par exemple pour le déplacement u :

$$\dot{u} = \frac{\partial u}{\partial t} \quad ; \quad \ddot{u} = \frac{\partial^2 u}{\partial t^2} \quad (1.1.14)$$

La formulation faible est obtenue en multipliant l'équation d'équilibre (1.1.4) par la fonction test δu puis en intégrant le produit sur l'ensemble du domaine dans la configuration courante Ω . On obtient en utilisant la notation indicielle et la convention de sommation des indices répétés :

$$\int_{\Omega} \delta u_i \left(\frac{\partial \sigma_{ji}}{\partial x_j} + b_i - \rho \ddot{u}_i \right) d\Omega = 0 \quad (1.1.15)$$

En utilisant la formule d'intégration par partie, le premier terme de (1.1.15) peut se réécrire sous la forme :

$$\int_{\Omega} \delta u_i \frac{\partial \sigma_{ji}}{\partial x_j} d\Omega = \int_{\Omega} \frac{\partial}{\partial x_j} (\delta u_i \sigma_{ji}) d\Omega - \int_{\Omega} \frac{\partial (\delta u_i)}{\partial x_j} \sigma_{ji} d\Omega \quad (1.1.16)$$

Puisque les fonctions tests s'annulent sur le contour Γ_u , la première intégrale du second membre (1.1.16) devient en appliquant le théorème de Gauss :

$$\int_{\Omega} \frac{\partial}{\partial x_j} (\delta u_i \sigma_{ji}) d\Omega = \int_{\Gamma} \delta u_i n_j \sigma_{ji} d\Gamma = \int_{\Gamma_t} \delta u_i \bar{t}_i d\Gamma \quad (1.1.17)$$

En substituant (1.1.16) et (1.1.17) dans (1.1.15), on obtient finalement :

$$\int_{\Omega} \frac{\partial(\delta u_i)}{\partial x_j} \sigma_{ji} d\Omega - \int_{\Omega} \delta u_i b_i d\Omega - \int_{\Gamma_t} \delta u_i \bar{t}_i d\Gamma + \int_{\Omega} \delta u_i \rho \ddot{u}_i d\Omega = 0 \quad (1.1.18)$$

Cette formulation faible équivalente à l'équation d'équilibre et à la condition limite en effort est souvent appelée principe des travaux virtuels.

Remarque 2 : On peut reformuler l'équation (1.1.18) comme un problème de stationnarité de la fonctionnelle \mathcal{W} correspondante à l'énergie potentielle totale du système. L'équilibre émanant du principe de stationnarité s'écrit :

$$\delta \mathcal{W}(u) = \delta \mathcal{W}_{int}(u) - \delta \mathcal{W}_{ext}(u) + \delta \mathcal{W}_{kin}(u) = 0 \quad \forall \delta u \in \mathcal{U}_0 \quad (1.1.19)$$

où chacun des termes $\delta \mathcal{W}_{int}$, $\delta \mathcal{W}_{ext}$ et $\delta \mathcal{W}_{kin}$ désignent respectivement les travaux des forces internes, externes et d'inerties dues au déplacement virtuel δu .

1.1.2 Discrétisation en espace - Méthode éléments finis

La résolution analytique du problème (1.1.18) étant généralement impossible, il est nécessaire d'approximer le problème continu par un modèle approché en espace et en temps. Nous nous intéresserons dans un premier temps à la discrétisation spatiale que propose la méthode des éléments finis [ZTJ05].

1.1.2.1 Approximation éléments finis

La discrétisation consiste à approcher le domaine continu Ω par un ensemble d'éléments finis Ω_e disjoint (cf. Figure 1.3). Nous noterons $\tilde{\Omega}$ le domaine approché ainsi formé ($\tilde{\Omega} \equiv \cup \Omega_e$) et $\tilde{\Gamma}$ sa frontière.

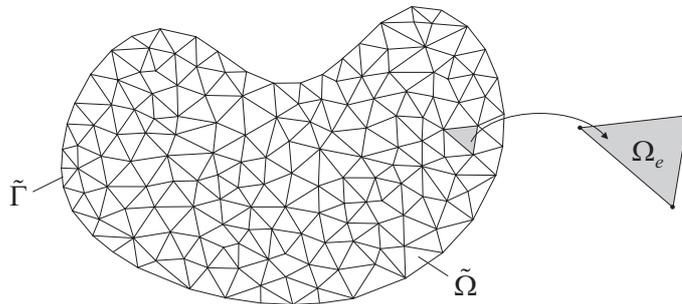


Figure 1.3: Discrétisation du domaine en éléments finis.

On distingue généralement deux types d'interpolation lors de la construction d'un élément fini : l'interpolation géométrique et l'interpolations des variables. Toutefois, nous nous limiterons ici au cas des éléments isoparamétriques, où les fonctions utilisées pour

l'interpolation des coordonnées et des variables sont identiques. Les champs inconnus deviennent alors des champs discrets définis aux noeuds de la discrétisation $\tilde{\Omega}$. Le champ de déplacement est approximé de la manière suivante :

$$u_i(X, t) = \sum_{I=1}^{n_{noeuds}} N_I(X) u_{iI}(t) \quad (1.1.20)$$

où n_{noeuds} désigne le nombre de noeuds, N_I les fonctions d'interpolation, encore appelées fonctions de forme. Le terme u_{iI} , fonction du temps t , désigne la composante de déplacement suivant i au noeud I . De manière similaire, on peut approximer les fonctions tests :

$$\delta u_i(X) = \sum_{I=1}^{n_{noeuds}} N_I(X) \delta u_{iI} \quad (1.1.21)$$

1.1.2.2 Équations discrètes

En introduisant les formules de discrétisations (1.1.20) et (1.1.21) dans le principe des travaux virtuels (1.1.18) , on obtient un système discret en espace :

$$\delta u_{iI} \left(\int_{\Omega} \frac{\partial N_I}{\partial x_j} \sigma_{ji} d\Omega - \int_{\Omega} N_I f_i^v d\Omega - \int_{\Gamma_f} N_I f_i^s d\Gamma + \int_{\Omega} \rho N_I N_J d\Omega \ddot{u}_{iJ} \right) = 0 \quad (1.1.22)$$

Les différentes intégrales de (1.1.22) sont explicitées par les grandeurs matricielles et vectorielles suivantes :

⇒ Matrice de masse

$$M_{ijIJ} = \delta_{ij} \int_{\Omega} \rho N_I N_J d\Omega \quad (1.1.23)$$

⇒ Vecteur force interne

$$f_{int,iI} = \int_{\Omega} \frac{\partial N_I}{\partial x_j} \sigma_{ji} d\Omega = \int_{\Omega} B_{jI}^t \sigma_{ji} d\Omega \quad (1.1.24)$$

⇒ Vecteur force externe

$$f_{ext,iI} = \int_{\Omega} N_I f_i^v d\Omega + \int_{\Gamma_f} N_I f_i^s d\Gamma \quad (1.1.25)$$

Ces trois opérateurs sont calculés par intégration des contributions sur chacun des éléments finis. Une procédure d'assemblage permet l'écriture du système global en sommant les contributions élémentaires au niveau des noeuds. En utilisant ces notations, l'équation (1.1.22) devient :

$$\delta u_{iI} (f_{int,iI} - f_{ext,iI} + M_{ijIJ} \ddot{u}_{jJ}) = 0 \quad \forall \delta u_{iI} \notin \tilde{\Gamma}_u \quad (1.1.26)$$

où M_{ijIJ} désigne les composantes de la matrice masse et $f_{int,iI}$, $f_{ext,iI}$ désignent respectivement les composantes des vecteurs forces internes et forces externes. L'équation (1.1.26) devant être vérifiée quel que soit le champ de déplacement virtuel choisi, on peut la réécrire sous la forme matricielle :

$$\{f_{int}\} - \{f_{ext}\} + [M]\{\ddot{u}\} = 0 \quad \text{sur } \tilde{\Omega} \setminus \tilde{\Gamma}_u \quad (1.1.27)$$

A cela s'ajoutent les conditions limites (1.1.1) et les conditions initiales (1.1.3) discrètes :

$$\{u\} = \{\bar{u}\} \quad \text{sur} \quad \tilde{\Gamma}_u \quad (1.1.28)$$

$$\{u\} = \{u_0\}, \quad \{\dot{u}\} = \{v_0\} \quad \text{à} \quad t = 0 \quad (1.1.29)$$

Afin d'alléger les écritures, nous abandonnerons les notations $\{\cdot\}$ et $[\cdot]$ pour désigner les grandeurs vectorielles et matricielles. Sauf lorsque la distinction sera absolument nécessaire, nous confondrons les champs discrets définis sur Ω avec les champs continus définis sur $\tilde{\Omega}$. Nous écrirons simplement (1.1.27) et (1.1.28) de la manière suivante :

$$f_{int} - f_{ext} + M\ddot{u} = 0 \quad \text{sur} \quad \tilde{\Omega} \setminus \tilde{\Gamma}_u \quad (1.1.30)$$

$$u = \bar{u} \quad \text{sur} \quad \tilde{\Gamma}_u \quad (1.1.31)$$

A noter que cette forme finale de l'équation dynamique (1.1.30) est dite semi-discrète puisqu'elle n'est pas discrétisée en temps. C'est cette discrétisation en temps que l'on propose de présenter dans le paragraphe suivant.

Remarque 1 : Les conditions limites peuvent être traitées de différentes manières. On peut par exemple utiliser la méthode des multiplicateurs de Lagrange, laquelle sera développée plus en détail au paragraphe 2.1.2.1. Dans ce cas, la condition (1.1.31) est prise en compte en adjoignant un terme $C^t \lambda$ aux forces externes du système (1.1.30), où C est une matrice de contrainte telle que $Cu = \bar{u}$.

1.1.3 Discrétisation en temps - Schéma d'intégration

L'équation d'équilibre (1.1.30), discrétisée en espace, forme un système d'équation différentielle du second ordre en temps. Différentes techniques peuvent être utilisées pour résoudre de tels systèmes à partir des conditions initiales (1.1.3). Nous nous intéresserons ici aux méthodes de discrétisation par pas (ou méthode d'intégration directe) qui consiste à transformer le problème différentiel en un problème approché, où les solutions sont calculées de manière discrète à différent instant t^n , $0 < n < N$, de l'intervalle $[0, T]$. Le pas de temps entre deux instants de calculs successifs, noté Δt , sera dans un premier temps considéré comme étant constant. Le problème d'équilibre discrétisé devient donc à l'instant t^n :

$$M\ddot{u}^n = f_{ext}^n - f_{int}^n \quad (1.1.32)$$

1.1.3.1 Schéma de Newmark

Dans le cadre de nos travaux nous ne considérerons que les schémas d'intégrations les plus classiques de la famille de Newmark, dont fait partie le schéma explicite qui nous intéressera plus particulièrement par la suite. De manière générale, le schéma de Newmark [New59] s'écrit :

$$u^{n+1} = p_u u^{n+1} + \beta \Delta t^2 \ddot{u}^{n+1} \quad (1.1.33)$$

$$\dot{u}^{n+1} = p_{\dot{u}} \dot{u}^{n+1} + \gamma \Delta t \ddot{u}^{n+1} \quad (1.1.34)$$

où les termes $p_u u^{n+1}$ et $p_{\dot{u}} \dot{u}^{n+1}$, appelés prédicteurs, regroupent les variables connues à l'instant t^n :

$$p_u u^{n+1} = u^n + \Delta t \dot{u}^n + \frac{\Delta t^2}{2} (1 - 2\beta) \ddot{u}^n \quad (1.1.35)$$

$$p_{\dot{u}} \dot{u}^{n+1} = \dot{u}^n + (1 - \gamma) \Delta t \ddot{u}^n \quad (1.1.36)$$

Comme nous allons le voir, les paramètres β et γ conditionnent la stabilité du schéma. Le paramètre β détermine le type de schéma. Si $\beta = 0$ le schéma est dit explicite, il est alors soumis à une condition de stabilité sur le pas de temps Δt appelée condition CFL (1.1.57). Par contre, si $\beta \neq 0$, le schéma est dit implicite. Dans ce cas, il est généralement inconditionnellement stable, ce qui permet l'utilisation d'un nombre réduit d'instant de calcul par intervalle d'étude. Le paramètre γ contrôle quant à lui la viscosité artificielle. Il s'agit d'un amortissement numérique introduit de manière à réduire le bruit sur la solution. Lorsque $\gamma = 1/2$, aucun amortissement n'est introduit dans le schéma alors que pour $\gamma > 1/2$ un amortissement proportionnel à $\gamma - 1/2$ est introduit. Cette dissipation numérique parfois utilisée en implicite peut-être un moyen de forcer la stabilité non-linéaire.

Convergence, consistance et stabilité Les formules (1.1.33) et (1.1.34) résultent du développement en série de Taylor des champs de déplacement et de la vitesse au voisinage du temps t^n . L'erreur de troncature inévitablement introduite doit donc être contrôlée pour éviter qu'elle ne s'amplifie au cours des pas de temps successifs et n'entraîne ainsi la divergence de l'algorithme de résolution.

Généralement, l'analyse de convergence des méthodes numériques se base sur l'étude de systèmes linéaires. L'extrapolation au cas non-linéaire, comme c'est ici le cas, se fait par la linéarisation du problème (1.1.32) entre deux instants de calcul successifs. La convergence doit alors être vérifiée à chaque incrément. Cela suppose que l'on peut ne retenir que les termes linéaires du tenseur des déformations, et que la relation de comportement soit linéaire entre σ et E . Le modèle linéarisé devient :

$$M\ddot{u} + K_{int}u = f_{ext} \quad \text{avec} \quad K_{int} = \frac{\partial f_{int}}{\partial u} \quad (1.1.37)$$

où la matrice K_{int} est appelée matrice de raideur tangente.

De manière générale, un schéma d'intégration est dit convergent si, à instant t^n fixé, la solution approchée u^n tend vers la solution réelle $u(t^n)$ lorsque le pas de discrétisation Δt tend vers 0.

$$\lim_{\Delta t \rightarrow 0} (e^n = u^n - u(t^n)) = 0 \quad \forall t^n \in [0, T] \quad (1.1.38)$$

L'erreur globale notée e^n prend en compte deux éléments a priori indépendants que sont l'erreur de troncature $\tau(t^n)$ due à la discrétisation d'une part (due au développement en série de Taylor), et l'amplification des erreurs (même initialement petites) due à la répétition du schéma. Il est évident que pour qu'il y ait convergence, il convient d'étudier séparément les notions de consistance (ou précision) et de stabilité, relatives à ces deux sources d'erreur. Le théorème d'équivalence de Lax établit la convergence discrète du schéma lorsque celui-ci est à la fois consistant et stable.

Remarque 1 : Le schéma est dit *consistant* si l'erreur de troncature locale $\tau(t)$ tend vers 0 lorsque le pas de temps tend vers 0, c'est-à-dire si il existe c tel que :

$$\tau(t^n) \leq c\Delta t^k, \quad k > 0, \quad \forall t \in [0, T] \quad (1.1.39)$$

Dans ce cas le schéma est dit précis à l'ordre k .

Remarque 2 : Le schéma est dit *stable* si toute perturbation de la donnée initiale n'entraîne qu'un faible changement de la solution. De manière plus formelle, en notant \hat{u}^0 l'état initiale u^0 perturbé, la condition de stabilité s'écrit :

$$\|u^n - \hat{u}^n\| \leq c\epsilon \quad \forall u^0 \quad \text{tel que} \quad \|u^0 - \hat{u}^0\| \leq \epsilon \quad (1.1.40)$$

1.1.3.2 Étude de convergence

On se propose ici d'étudier la convergence du schéma de Newmark dans sa forme la plus générale. Plusieurs techniques peuvent être utilisées pour analyser la convergence de l'algorithme. Nous présenterons ici une approche modale aussi appelée analyse spectrale. Une autre alternative, basée sur une approche énergétique, est présentée dans [Hug01]. Elle sera développée plus précisément dans le paragraphe 3.2.3.

La première étape consiste à décomposer le système d'équations (1.1.37) en n_{ddl} équations scalaires indépendantes (où n_{ddl} est le nombre de degrés de liberté du système). D'après [Hug01], on montre que la convergence du problème ainsi formé implique la convergence du problème matriciel original. Afin de procéder à cette décomposition, on utilise les vecteurs propres ψ_I du système associés :

$$K_{int}\psi_I = \omega_I^2 M\psi_I \quad (1.1.41)$$

où ω_I représente la pulsation associée à chacun des modes propres. Les vecteurs propres formant une base de $\mathbb{R}^{n_{ddl}}$, chaque vecteur $u \in \mathbb{R}^{n_{ddl}}$ peut s'écrire comme étant une combinaison linéaire de ces vecteurs propres :

$$u(t) = \sum_{J=1}^{n_{ddl}} \alpha_J(t)\psi_J \quad (1.1.42)$$

En appliquant la décomposition modale (1.1.42) à l'équation discrète (1.1.37) préalablement multipliée par ψ_I^t , et en utilisant l'orthogonalité des vecteurs propres par rapport aux matrices M et K , on obtient notre système de n_{ddl} équations indépendantes :

$$\ddot{\alpha}_I^{n+1} + \omega_I^2 \alpha_I^{n+1} = f_I^{n+1} \quad \text{avec} \quad f_I^{n+1} = \psi_I^t f_{ext}^{n+1} \quad (1.1.43)$$

De même, la décomposition modale peut être appliquée aux relations (1.1.33) et (1.1.34) du schéma de Newmark. Pour simplifier les écritures, nous omettrons par la suite l'indice relatif au mode I . On obtient finalement :

$$\ddot{\alpha}^{n+1} + \omega^2 \alpha^{n+1} = f^{n+1} \quad (1.1.44)$$

$$\alpha^{n+1} = \alpha^n + \Delta t \dot{\alpha}^n + \frac{\Delta t^2}{2} [(1 - 2\beta)\ddot{\alpha}^n + 2\beta\ddot{\alpha}^{n+1}] \quad (1.1.45)$$

$$\dot{\alpha}^{n+1} = \dot{\alpha}^n + \Delta t [(1 - \gamma)\ddot{\alpha}^n + \gamma\ddot{\alpha}^{n+1}] \quad (1.1.46)$$

La seconde étape de l'analyse consiste à transformer chaque équation scalaire du second ordre (1.1.44) en un système du premier ordre, mais de dimension 2. On peut ainsi ré-écrire les équations (1.1.44-1.1.46) sous la forme de la relation de récurrence :

$$y^{n+1} = Ay^n + L^n \quad (1.1.47)$$

où la matrice A , appelé matrice d'amplification, est définie par $A = A_1^{-1}A_2$ tel que :

$$A_1 = \begin{bmatrix} 1 + \Delta t^2 \beta \omega^2 & 0 \\ \Delta t \gamma \omega^2 & 1 \end{bmatrix} \text{ et } A_2 = \begin{bmatrix} 1 + \Delta t^2 (1 - 2\beta) \omega^2 & 0 \\ -\Delta t (1 - \gamma) \omega^2 & 1 \end{bmatrix} \quad (1.1.48)$$

Le vecteur chargement L^n ainsi que le vecteur d'état y^n sont eux définis par :

$$L^n = A_1^{-1} \left\{ \begin{array}{l} \frac{\Delta t^2}{2} [(1 - 2\beta)f^n + 2\beta f^{n+1}] \\ \Delta t [(1 - \gamma)f^n + \gamma f^{n+1}] \end{array} \right\} \text{ et } y^n = \left\{ \begin{array}{l} \alpha^n \\ \dot{\alpha}^n \end{array} \right\} \quad (1.1.49)$$

En remplaçant y^n et y^{n+1} par leurs valeurs exactes continues dans l'équation approchée (1.1.47), on fait apparaître le vecteur d'erreurs de troncatures locales $\tau(t^n)$:

$$y(t^{n+1}) = Ay(t^n) + L^n + \Delta t \tau(t^n) \quad (1.1.50)$$

En utilisant le développement limité de Taylor de y^{n+1} autour de t^n dans (1.1.50), on montre que $\tau(t^n) = \mathcal{O}(\Delta t^k)$ pour tout $t^n \in [0, T]$, avec $k = 2$ si $\gamma = 1/2$ et $k = 1$ sinon. Le schéma de Newmark est donc précis à l'ordre 2 ou à l'ordre 1 suivant la valeur du paramètre γ . Dans tous les cas, la condition (1.1.39) restant vérifiée, *le schéma est bien consistant*.

La stabilité du schéma peut ensuite être étudiée en examinant les propriétés spectrales de la matrice d'amplification A . On note alors $\lambda_i(A)$ les valeurs propres de A calculées en résolvant l'équation :

$$\det(A - \lambda(A)I) = \lambda(A)^2 - 2A_I \lambda(A) + A_{II} = 0 \quad (1.1.51)$$

où A_I et A_{II} sont les invariants principaux de A :

$$A_I = \frac{1}{2} \text{trace} A \quad \text{et} \quad A_{II} = \det A \quad (1.1.52)$$

En substituant (1.1.47) à (1.1.50), on obtient la relation sur l'erreur globale :

$$e(t^n) = A^n e(0) - \Delta t \sum_{i=0}^{n-1} A^i \tau(t^{n-1-i}) \quad (1.1.53)$$

Le premier terme du second membre étant nul si $y^0 = y(0)$, on montre que l'erreur ne s'amplifie pas tant que $\|A^n\| \leq Cste$. D'après [Hug83] cette condition de stabilité est vérifiée si (i) le module de toutes les valeurs propres de A est inférieur ou égal à 1 (i.e $\rho(A) = \max |\lambda_I(A)| \leq 1$) et si (ii) l'ordre de multiplicité de chaque valeur propre tel que $|\lambda_I| = 1$ n'est pas plus grand que 1. La déclinaison de (i) et (ii) sur les invariants principaux de A donne :

$$-(A_{II} + 1)/2 \leq A_I \leq (A_{II} + 1)/2 \quad (1.1.54)$$

$$-1 \leq A_{II} \leq 1 \quad (1.1.55)$$

D'après (1.1.52), on en déduit les conditions de stabilités classiques pour le schéma de Newmark :

- ⇒ Si $2\beta \geq \gamma \geq \frac{1}{2}$, le schéma est inconditionnellement stable.
- ⇒ Si $2\gamma \geq \frac{1}{2}$ et $\beta < \frac{\gamma}{2}$, le schéma est stable pour :

$$\Delta t \leq \frac{1}{\omega \sqrt{\frac{\gamma}{2} - \beta}} \quad (1.1.56)$$

En cas de stabilité conditionnelle, la condition sur le pas de temps vaut pour tous les modes ω_I du système. Le pas de temps critique Δt_c sera donc celui imposé par la pulsation maximal ω_{max} . En pratique, on ne cherche pas à calculer de manière exacte cette pulsation, notamment car elle nécessite l'assemblage de la matrice de raideur tangente, pourtant inutile à la résolution. On montre alors que cette pulsation est majorée par la pulsation ω_e maximale calculée indépendamment sur chacun des éléments issus de la discrétisation spatiale [Iro70, HPT79]. Ceci conduit à la célèbre condition CFL (Courant-Friedrich-Levy) :

$$\Delta t \leq \Delta t_c = \min_{e \in \Omega} \frac{1}{\omega_e \sqrt{\frac{\gamma}{2} - \beta}} \quad (1.1.57)$$

1.1.3.3 Schémas aux différences centrées

Par la suite, nous nous intéresserons tout particulièrement au schéma explicite, encore appelé schéma aux différences centrées obtenu avec les paramètres $\beta = 0$ et $\gamma = 1/2$:

$$u^{n+1} = u^n + \Delta t \dot{u}^n + \frac{\Delta t^2}{2} \ddot{u}^n \quad (1.1.58)$$

$$\dot{u}^{n+1} = \dot{u}^n + \frac{\Delta t^2}{2} (\ddot{u}^n + \ddot{u}^{n+1}) \quad (1.1.59)$$

D'après (1.1.58), on constate que les déplacements (et par conséquent les nouvelles coordonnées spatiales) sont déterminés explicitement à partir de l'état précédent. Ceci confère aux schémas explicites un traitement aisé de l'équilibre en rejetant au second membre les principales sources de non-linéarité. Contrairement aux schémas implicites, ils ne souffrent généralement d'aucune dissipation numérique, mais sont fortement pénalisés par la forte contrainte sur la taille du pas de temps de stabilité utilisé. A noter que dans le cas d'un schéma au différence centré, la condition (1.1.57) se décline sous la forme :

$$\Delta t = \alpha \Delta t_c, \quad \Delta t_c = \min_{e \in \Omega} \frac{2}{\omega_e} = \min_{e \in \Omega} \frac{\ell_e}{c_e} \quad (1.1.60)$$

où ℓ_e est la longueur caractéristique de l'élément e , et c_e est la vitesse de propagation du son dans ce même élément. Le facteur α est un coefficient sécurité prenant en compte les effets déstabilisant des non-linéarités ($0.8 \leq \alpha \leq 0.98$).

Le grand nombre de pas de temps utilisé en explicite peut donc devenir fortement rédhibitoire en terme de temps de calcul. L'emploi d'un tel schéma est donc particulièrement adapté en cas de forte non-linéarité, en particulier pour les problèmes de contact/impact, mais aussi pour les problèmes de dynamique rapide, lorsque la représentation temporelle des conditions limites impose une grande finesse dans la discrétisation temporelle (chocs, explosions...). Le pas souhaitable pour la bonne prise en compte des phénomènes devient alors de l'ordre du pas de temps de stabilité, ce qui lève le principal inconvénient de ces schémas.

1.1.4 Algorithme de dynamique explicite

La figure 1.4 schématise les différentes étapes intervenant lors d'un calcul de dynamique explicite. L'algorithme débute par l'initialisation des différentes variables (grandeurs cinématiques, des variables internes...) ainsi que par le calcul de la matrice masse M .

Comme la plupart des code de dynamique explicite, RADIOSS utilise une formulation Lagrangienne réactualisée (cf. §1.1.1) en prenant la configuration courante comme configuration de référence. Cette formulation est bien adaptée à ce type de schéma où le déplacement u^{n+1} (et par conséquent la configuration finale Ω^{n+1}) est parfaitement connue dès le début du pas. Dès lors, nous appellerons configuration initiale la configuration d'équilibre déjà calculé à l'instant t^n et configuration finale, la configuration correspondant au temps de calcul courant t^{n+1} .

Ainsi, pour résoudre l'équilibre dynamique à l'instant t^{n+1} , on commence par calculer la configuration Ω^{n+1} , grâce à l'incrément de déplacement Δu^{n+1} fourni explicitement par le prédicteurs du schéma (1.1.58). Nous noterons x^{n+1} le vecteur de coordonnées nodales représentative de la configuration géométrique à l'instant t^{n+1} .

$$x^{n+1} = x^n + \Delta u^{n+1} \quad (1.1.61)$$

Les forces nodales peuvent être déterminées en évaluant les forces internes et les forces externe de manière à ce que le second membre de (1.1.32) soit complètement connu.

Calcul des forces internes Le calcul des forces internes nécessite l'évaluation de l'incrément de déformation ΔE_{AE}^n entre les instants t^n et $t^n + 1$. On en déduit grâce à la loi de comportement (loi incrémentale comme souvent en non-linéaire) l'incrément de contrainte $\Delta \sigma^n$ subit entre ces deux instants. On dispose donc de contrainte, noté Π_n^{n+1} calculé à l'instant t^{n+1} mais sur la configuration Ω^n .

$$\Pi_n^{n+1} = \sigma^n + \Delta \sigma^n \quad (1.1.62)$$

Afin d'exprimer la contrainte sur la configuration courante, à savoir Ω^{n+1} , il convient d'après (1.1.10) de lui appliquer la transformation :

$$\sigma^{n+1} = \frac{1}{\det(F)} F \Pi_n^{n+1} F^t \quad (1.1.63)$$

où $F = I - \text{grad}[\Delta u^n]$ est le gradient de déformation entre la configuration initiale Ω^n et la configuration courante Ω^{n+1} . Les forces internes $f_{int,e}^{n+1}$ sont ensuite évaluées au niveau de l'élément en intégrant le produit de la matrice $B = \partial N / \partial x$ et de la contrainte de Cauchy σ^{n+1} sur le domaine élémentaire.

$$f_{int,e}^{n+1} = \int_{\Omega_e} B^t \sigma^{n+1} d\Omega \quad (1.1.64)$$

L'intégrale (1.1.64) est calculée par la méthode de Gauss, ce qui en pratique nécessite l'évaluation des contraintes (et des incréments de déformation) aux points de quadrature de l'élément.

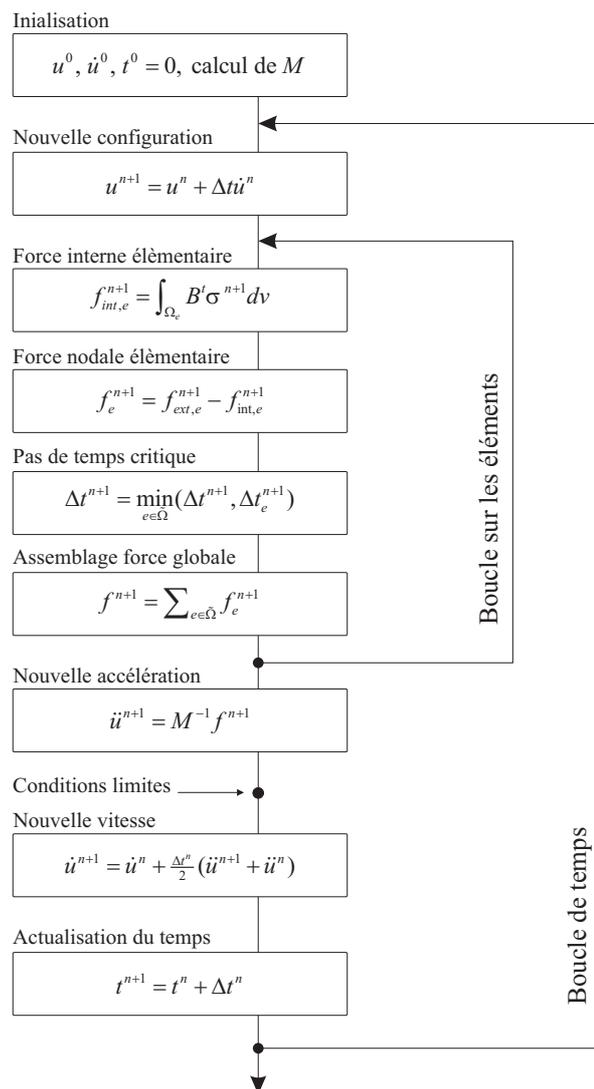


Figure 1.4: Algorithme de dynamique explicite.

Calcul du nouvel équilibre Une fois les forces internes calculées, celles-ci peuvent être soustraites aux forces externes. On obtient alors un vecteur de forces nodales élémentaires $f_e^n = f_{ext,e}^n - f_{int,e}^n$ qu'il convient d'assembler sur l'ensemble du maillage pour obtenir le second membre f^n de l'équation d'équilibre :

$$f^{n+1} = \sum_{e \in \tilde{\Omega}} f_{ext,e}^{n+1} - f_{int,e}^{n+1} = \sum_{e \in \tilde{\Omega}} f_e^{n+1} \quad (1.1.65)$$

Le calcul du nouvel équilibre à l'instant t^{n+1} est alors réalisé sans avoir à résoudre aucun système d'équation, pourvu que la matrice masse soit diagonale. Cette approximation qu'est la diagonalisation de M consiste à concentrer la masse sur les noeuds du maillage. Ceci caractérise généralement les méthodes explicites en simplifiant fortement le calcul des nouvelles composantes d'accélération :

$$\ddot{u}_i^{n+1} = \frac{f_i^{n+1}}{m_i} \quad (1.1.66)$$

où m_i est la masse concentrée au noeud i . Le calcul des vitesses à partir de (1.1.59) achève la détermination du nouvel état d'équilibre à l'instant t_{n+1} . Les conditions limites peuvent alors être prises en compte, avant le calcul du nouveau déplacement (au prochain cycle) en forçant la vitesse des noeuds contraints. Nous remarquerons qu'en cas d'utilisation de la méthode des multiplicateurs de Lagrange, ces conditions seront directement prises en compte en ajoutant un terme supplémentaire, dit de liaison, au vecteur de force global f^n .

Calcul du pas de temps Conformément à (1.1.60), le pas de temps utilisé lors du calcul correspond donc au plus petit pas de temps élémentaire repéré sur le maillage. En pratique, le calcul du pas de temps de chaque élément est réalisé en même temps que le calcul des forces internes.

En présence de non-linéarités géométriques, la déformation des éléments nécessite l'adaptation de la valeur du pas de temps au cours du calcul. Il s'agit de préserver la stabilité de l'algorithme si la taille du plus petit élément diminue, ou d'accroître ses performances en terme de temps de calcul si elle augmente. Nous considérerons dès lors le pas de temps comme étant variable et nous noterons Δt^n le pas entre les instants t^n et t^{n+1} .

1.2 Méthode de décomposition de domaines

Comme nous l'avons vu précédemment, la discrétisation d'équations aux dérivées partielles engendre des systèmes algébriques linéaires, souvent de grandes dimensions. Compte tenu de cette spécificité, la résolution de ces systèmes par les méthodes directes ou itératives classiques devient de plus en plus préoccupante, et généralement très délicate compte tenu du mauvais conditionnement des matrices. Depuis une vingtaine d'années, l'évolution des architectures des machines et l'apparition des multiprocesseurs, nous obligent à reconsidérer les algorithmes de résolution classique pour ce type de supercalculateur. C'est dans ce cadre que se sont développées les méthodes de décomposition de domaine. En effet, le découpage du problème initial en plusieurs sous-problèmes de petite taille et sur des géométries plus simples s'adapte naturellement aux architectures parallèles qui permettent d'en tirer profit.

Ces techniques apparues il y a plus d'un siècle avec Schwartz (1869) [Sch69], doivent être considérées comme des techniques de résolution à part entière et non uniquement comme des stratégies de calcul adaptées aux nouveaux moyens informatiques. Il est d'ailleurs intéressant de noter que ces méthodes peuvent être plus rapides, même sur calculateur séquentiel, que les méthodes de résolution classiques. Enfin ces techniques facilitent l'utilisation de modélisation et de méthodes de résolution différentes sur chaque sous-domaine. Elles sont particulièrement bien adaptées aux calculs multi-échelles en utilisant pour chaque sous-domaine la modélisation la plus adéquate. Nous reviendrons plus en détail sur cet aspect dans le chapitre 3.

1.2.1 Bref état de l'art sur les méthodes multi-domaines

Ce paragraphe a pour but de présenter brièvement les principales techniques de décomposition de domaines. Pour une étude plus exhaustive, on peut se rapporter à [Gra00]. Parmi les différentes méthodes présentées, nous distinguerons les méthodes avec recouvrement (Schwartz) des méthodes sans recouvrement (*Schur primale*, *Schur duale*, mixte). Comme nous le verrons par la suite, les méthodes avec recouvrement se différencient des méthodes sans recouvrement par le fait qu'elles ne conduisent pas à la résolution d'un problème d'interface entre les sous-domaines voisins.

Pour simplifier, nous nous restreignons volontairement au cadre original des équations elliptiques et plus particulièrement au cas de l'équation d'élasticité. Nous regardons ici le problème discrétisé de manière à présenter les différentes méthodes sous forme de relations matricielles. Sous l'hypothèse des petites perturbations, nous considérons donc le problème statique d'élasticité linéaire suivant :

$$\begin{cases} Ku = f & \text{sur } \Omega \\ u = \bar{u} & \text{sur } \Gamma \end{cases} \quad (1.2.1)$$

Enfin, pour illustrer nos propos, nous considérons dans cette partie une décomposition du domaine principal en deux sous-domaines. L'extension à un nombre quelconque de sous-domaines reste toutefois évidente.

1.2.1.1 Méthode de *Schwarz*

Les méthodes avec recouvrement sont encore assez peu utilisées en mécanique des structures. C'est pourquoi nous ne présenterons ici que la plus connue d'entre elles : la méthode de *Schwarz* [Sch69], qui historiquement est à l'origine de nombreuses méthodes de décomposition. La méthode *Arlequin* [BDR05, BD05] présentée dans le cadre des approches multi-échelles du dernier chapitre peut toutefois être classée dans cette catégorie de technique avec recouvrement.

Dans sa forme originale, la méthode alternée de *Schwarz* s'applique à la résolution de problèmes elliptiques (le but de Schwarz était d'étudier l'opérateur de Laplace) sur un domaine formé d'une réunion de deux sous-domaines. On considère notre domaine Ω partitionné en deux sous-domaines ouverts Ω_1 et Ω_2 tels que :

$$\Omega = \Omega_1 \cup \Omega_2 \quad (1.2.2)$$

On distingue les parties de frontières réelles $\Gamma_k \in \Gamma$ ($k = 1, 2$) des parties de frontières virtuelles γ_k incluses dans le sous-domaine Ω_l ($k = 1, 2; l \neq k$) (cf. Figure 1.5).

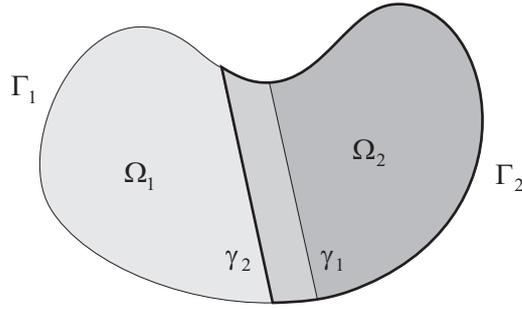


Figure 1.5: Décomposition de domaine - Méthode de *Schwarz*.

Comme pour toutes les méthodes avec recouvrement, la méthode de *Schwarz* est une méthode itérative qui consiste à résoudre le problème (1.2.1) de façon alternée sur chaque sous-domaine. Une étude de convergence de la méthode est faite dans [Lio88]. Nous noterons $u^{(i)}$ l'approximation de la solution calculée à l'itération i et $u_k^{(i)}$ la restriction de $u^{(i)}$ au sous-domaine Ω_k .

Pour chaque sous-domaine Ω_k , la condition limite sur la frontière virtuelle γ_k , est imposée par le sous-domaine voisin Ω_l . L'itéré choisi pour mettre à jour les conditions aux limites détermine les deux variantes de cette méthode, à savoir la méthode de *Schwarz* multiplicative et la méthode de *Schwarz* additive.

Dans la méthode de ***Schwarz* multiplicative** la solution $u^{(i)}$ à l'itération i est calculée en résolvant successivement le problème pour u_1 puis le problème pour u_2 :

$$\begin{cases} Ku_1^{(i)} = f_1 & \text{sur } \Omega_1 \\ u_1^{(i)} = \bar{u} & \text{sur } \Omega_1 \setminus \Gamma_1 \\ u_1^{(i)} = u_2^{(i-1)} & \text{sur } \gamma_1 \end{cases} \quad \begin{cases} Ku_2^{(i)} = f_2 & \text{sur } \Omega_2 \\ u_2^{(i)} = \bar{u} & \text{sur } \Omega_2 \setminus \Gamma_2 \\ u_2^{(i)} = u_1^{(i)} & \text{sur } \gamma_2 \end{cases} \quad (1.2.3)$$

La solution globale calculée à chaque itération est reconstruite comme suit :

$$u^{(i)}(M) = \begin{cases} u_2^{(i)}(M) & \text{si } M \in \Omega_2 \\ u_1^{(i)}(M) & \text{si } M \in \Omega \setminus \Omega_2 \end{cases} \quad (1.2.4)$$

En pratique, on définit pour chaque sous-domaine la matrice K_i correspondante à la sous-matrice de K associée au sous-domaine Ω_k :

$$K_k = R_k \cdot K \cdot R_k^t \quad (1.2.5)$$

où R_k est la matrice de restriction qui retourne le vecteur de composante à l'intérieur de Ω_k . Sa transposé R_k^t est la matrice de prolongation (ou d'interpolation). Chaque itération peut être calculée en deux temps :

$$\begin{aligned} u^{(i+\frac{1}{2})} &= u^{(i)} + B_1 \left(f - K u^{(i)} \right) \\ u^{(i+1)} &= u^{(i+\frac{1}{2})} + B_2 \left(f - K u^{(i+\frac{1}{2})} \right) \end{aligned} \quad (1.2.6)$$

Avec $B_k = R_k^t K_k^{-1} R_k$. Dans l'algorithme décrit ci-dessus, le calcul doit être fait séquentiellement sur les sous-domaines, ce qui n'est pas adapté au calcul parallèle. Cette méthode s'interprète algébriquement comme une itération de Jacobi par blocs ou chaque bloc est relatif à l'opérateur restreint sur un sous-domaine.

Dans la seconde variante, appelée méthode de **Schwarz additif**, les itérés successifs sont calculés comme suit :

$$u^{(i+1)} = u^{(i)} + \sum_{k=1}^2 B_k \left(f - K u^{(i)} \right) \quad (1.2.7)$$

En termes algébriques, cette méthode s'apparente à une méthode de Gauss-Seidel par blocs. La résolution sur chaque sous-domaine pouvant être cette fois réalisée simultanément, cette méthode est plus adaptée au calcul parallèle. Toutefois, cela se fait au dépend d'une convergence plus lente [SBG96].

Remarque 1 : La vitesse de convergence de ces méthodes est fortement influencée par la taille du recouvrement. En effet, la méthode converge d'autant plus vite que le recouvrement est grand. Néanmoins, le recouvrement implique une redondance de calculs sur les degrés de liberté associés à plusieurs sous-domaines. L'efficacité de ces méthodes résulte donc d'un bon compromis entre la vitesse de convergence et le volume de calculs redondants.

1.2.1.2 Méthode de *Schur primale*

Contrairement à la méthode de *Schwarz*, les méthodes de type *Schur* (*primale* ou *duale*) sont très populaires en mécanique des structures. Pour ces méthodes qualifiées de non recouvrantes, le domaine Ω est décomposé en deux ouverts disjoints Ω_1 et Ω_2 et d'une interface de liaison Γ_l (cf. Figure 1.6).

La version *primale* consiste à condenser le problème sur les interfaces entre les domaines, en vérifiant constamment la continuité en déplacement [Rou90], à savoir :

$$u_1(X) = u_2(X) \quad \forall X \in \Gamma_l \quad (1.2.8)$$

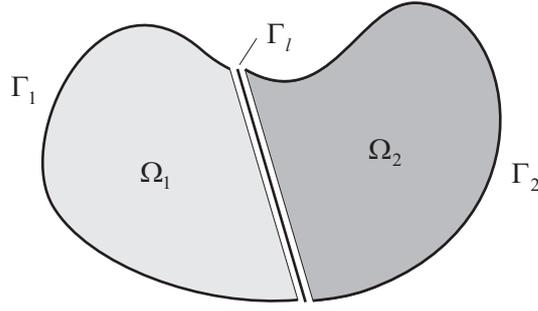


Figure 1.6: Décomposition de domaine - Méthode de *Schur primale*.

On procède pour cela à la renumérotation des degrés de liberté du système de manière à isoler les inconnues associées exclusivement à chaque sous-domaine, des inconnues associées à l'interface. Le système algébrique (1.2.1) peut alors s'écrire sous la forme :

$$\begin{bmatrix} K_{11} & 0 & K_{1l} \\ 0 & K_{22} & K_{2l} \\ K_{l1} & K_{l2} & K_{ll} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_l \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_l \end{Bmatrix} \quad (1.2.9)$$

où les sous-blocs diagonaux K_{11} , K_{22} correspondent aux matrices de rigidité associées aux noeuds situés à l'intérieur des sous-domaines Ω_1 et Ω_2 (interface exclue) et K_{ll} correspond à la matrice de rigidité associée aux noeuds d'interface Γ_l . Les sous-blocs K_{il} et K_{li} ($i = 1, 2$) (identiques à une transposition près) correspondent aux interactions entre les noeuds de Ω_i et les noeuds d'interface.

On peut alors opérer par élimination de Gauss à une condensation de (1.2.9) sur les inconnues primales u_l de l'interface. En effet, en éliminant les inconnues internes u_1, u_2 grâce aux deux premières lignes de (1.2.9), on obtient le problème condensé (ou problème d'interface) spécifique des méthodes de décomposition sans recouvrement :

$$Hu_l = B \quad (1.2.10)$$

où l'opérateur H est une matrice dense, dite complément de *Schur* :

$$H = K_{ll} - \sum_{k=1}^2 K_{lk} K_{kk}^{-1} K_{kl} \quad (1.2.11)$$

et où le second membre B est donné par :

$$B = f_l - \sum_{k=1}^2 K_{lk} K_{kk}^{-1} f_k \quad (1.2.12)$$

Le système (1.2.10) traduit l'équilibre des noeuds d'interface soumis aux efforts extérieurs f_l , aux efforts issus de l'équilibre intérieur aux sous-domaines $-K_{lk} K_{kk}^{-1} f_k$ et aux efforts de réaction de chaque sous-domaine en réponse à un déplacement d'interface $K_{lk} K_{kk}^{-1} K_{kl} u_l$.

Cette méthode est qualifiée de *primale* car le traitement du problème à l'interface s'effectue sur les inconnues primales, à savoir ici les déplacements. En procédant ainsi, on assure de manière forte la continuité cinématique entre les sous-domaines, c'est-à-dire que dans ce cas précis, la continuité des déplacements est assurée de part et d'autre de l'interface. L'inconvénient de cette continuité forte se trouve dans la minimisation de l'énergie (1.1.19), où l'on cherchera seulement à vérifier au mieux l'équilibre de la structure.

Remarque 2 : La résolution du problème d'interface (1.2.10) peut être obtenue par une méthode directe ou itérative. Dans le cas d'une résolution par méthode directe, la matrice H doit être explicitement construite, ce qui n'est pas nécessaire pour une méthode itérative de type gradient conjugué.

Remarque 3 : Dans cette approche, les noeuds d'interfaces sont communs aux deux domaines en vis-à-vis, ce qui nécessite une liaison parfaite entre maillages compatibles. En théorie, le collage de maillage incompatible reste toutefois possible en dupliquant les noeuds d'interface et en introduisant explicitement des relations de continuité cinématique entre les noeuds.

1.2.1.3 Méthode de *Schur duale*

Cette méthode a été présentée dans [FR91, FR94], elle est aussi connue sous le nom de FETI (Finite Element Tearing and Interconnecting). Elle se distingue de la méthode *primale* par la condition de continuité imposée sur les interfaces. En effet, dans la méthode de *Schur duale* on relaxe (ou dualise) la contrainte de continuité (1.2.8), en cherchant des solutions telles que :

$$\int_{\Gamma_l} \lambda (u_1 - u_2) d\Gamma = 0 \quad (1.2.13)$$

On introduit une variable supplémentaire, un multiplicateur de Lagrange λ , pour prendre en compte la contrainte à l'interface. Pour bien comprendre les fondements de la méthode de *Schur duale*, il faut remonter à l'écriture variationnelle d'un problème sous contrainte. Nous ne nous attarderons pas ici sur ce point qui sera détaillé dans la section 2.1.2.1. Comme nous le verrons, la formulation variationnelle du problème sous contrainte conduit à un problème de minimisation d'énergie dont la solution est (u_1, u_2, λ) . Le système algébrique résultant s'écrit de la manière suivante :

$$\begin{bmatrix} K_{11} & 0 & C_1^t \\ 0 & K_{22} & C_2^t \\ C_1 & C_2 & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \lambda \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ 0 \end{Bmatrix} \quad (1.2.14)$$

où l'on distingue les sous-blocs K_{11} et K_{22} correspondant à la matrice de rigidité associée aux noeuds des sous-domaines locaux Ω_1 et Ω_2 (interface comprise) et les sous-blocs C_1 et C_2 correspondant aux matrices de contraintes (ou de liaison) permettant de localiser les noeuds sur l'interface Γ_l . Le vecteur des multiplicateurs de Lagrange λ représente les forces d'interaction entre les sous-domaines.

A présent c'est l'équilibre de l'interface qui est supposé vérifié *a priori*. En effet, cette méthode permet de vérifier exactement l'équilibre de la structure, la minimisation de l'énergie revenant à vérifier au mieux la continuité cinématique (1.2.13) entre les sous-domaines.

Remarque 4 : Pour le collage de maillages compatibles, la matrice de contrainte C_k peut s'écrire sous la forme d'une matrice booléenne puisqu'elle ne fait que projeter les degrés de liberté intérieurs au sous-domaine Ω_k sur sa frontière commune avec le sous-domaine voisin. Ceci n'est plus le cas pour le collage de maillages incompatibles, où la matrice de liaison dépend des discrétisations choisies pour les trois champs u_1 , u_2 et λ . Nous reviendrons plus en détail sur la construction de cet opérateur par la suite.

On effectue à nouveau une condensation sur les inconnues d'interface, à savoir ici les multiplicateur de Lagrange, en éliminant les déplacements sur chaque sous-domaine grâce aux deux premières lignes du système (1.2.14).

$$u_1 = K_{11}^{-1} (f_1 - C_1^t \lambda) \quad (1.2.15)$$

$$u_2 = K_{22}^{-1} (f_2 - C_2^t \lambda) \quad (1.2.16)$$

On obtient ainsi le problème condensé suivant :

$$H\lambda = B \quad (1.2.17)$$

où l'opérateur d'interface H et le second membre B s'écrivent :

$$H = \sum_{k=1}^2 C_k K_{kk}^{-1} C_k^t \quad (1.2.18)$$

$$B = \sum_{k=1}^2 C_k K_{kk}^{-1} f_k \quad (1.2.19)$$

Le problème est ainsi décomposé en une succession d'équilibres individuels par sous-domaines, et l'interface devient un lieu propre faisant l'objet d'un traitement particulier.

La méthode de *Schur duale* est plus facile à mettre en oeuvre que la méthode de *Schur primale*, car il n'y a pas besoin de numéroter les variables du système de façon à séparer les noeuds internes des noeuds d'interface. De plus, cette méthode est adaptée au calcul parallèle, car les échanges de données pour le calcul d'interface sont limités aux sous-domaines partageant une arête. En ce sens, les interfaces entre sous-domaines sont gérées de manière disjointe.

1.2.1.4 Méthodes mixtes

Contrairement aux méthodes *primales* et *duales*, les méthodes mixtes traitent à égalité les efforts et les déplacements à l'interface. Elle ne privilégie a priori ni la continuité cinématique, ni l'équilibre de l'interface. Parmi ces approches on peut citer l'extension de l'algorithme de *Schwarz* aux cas de décomposition sans recouvrement [Lio90] ou deux autres méthodes basées sur une formulation par Lagrangien augmenté [GLT90, Lad85].

Ces méthodes largement utilisées pour l'analyse d'assemblage complexe de structures permettent de prendre compte les non-linéarités dues au contact (décollement, adhérence et glissement) et au matériaux (joint). Les assemblages sont alors décrits par un ensemble de sous-structure et d'interfaces modélisant les liaisons entre les composant de l'assemblage. Contrairement aux techniques de décomposition classique, les interfaces ne

sont pas uniquement des objets fictifs engendrés par le découpage, mais des entités mécaniques à part entière qui disposent de leurs propres inconnues, de leurs propres équations d'équilibre et de leur propre comportement.

La stratégie de résolution associée à ces approches mixtes est souvent basée sur les concepts de la méthode LATIN (LARGE Time INcrement method). L'idée de cette méthode itérative, est de séparer les problèmes linéaires globaux des problèmes non-linéaires localisés aux interfaces [Lad96]. Cette approche a dans un premier temps été utilisée pour l'analyse axisymétrique bidimensionnelle [BCCL96] puis a été étendue au cas tridimensionnel avec la méthode CONTRAST (CONtact TRidimensionnel dans les Assemblages STATiques) [Cha96, CCL99].

1.2.2 Méthode multi-domaines en dynamique explicite

1.2.2.1 Définition du problème couplé

Dans le cadre de nos travaux, la méthode de décomposition utilisée est une méthode de type *Schur duale*. En dynamique, se pose alors la question de la quantité cinématique à imposer pour assurer la continuité entre les sous-domaines. En effet, d'un point de vue continu en temps, imposer la continuité des déplacements entraîne automatiquement la continuité des vitesses et des accélérations. Cela n'est pas le cas d'un point de vue discret. D'après [GC01], il s'avère que les vitesses soient les plus adaptées pour assurer la stabilité de la méthode. La relation de continuité décrite dans le cas statique par la dernière équation du système (1.2.14) devient alors l'instant t^n :

$$C_1 \dot{u}_1^n + C_2 \dot{u}_2^n = 0 \quad (1.2.20)$$

D'après (1.2.20) et (1.1.30), le problème de référence discret peut ainsi s'écrire dans le cas de deux sous-domaines couplés :

$$\begin{cases} M_1 \ddot{u}_1^n + f_{1,int}^n = f_{1,ext}^n + C_1^t \lambda^n \\ M_2 \ddot{u}_2^n + f_{2,int}^n = f_{2,ext}^n + C_2^t \lambda^n \\ C_1 \dot{u}_1^n + C_2 \dot{u}_2^n = 0 \end{cases} \quad (1.2.21)$$

où rappelons-le, le terme $C_k^t \lambda$ ($k = 1, 2$), homogène à un effort, représente les interactions entre les sous-domaines. En utilisant les relations (1.1.33) et (1.1.34) dans le cas du schéma aux différences centrées ($\beta = 0$, $\gamma = 1/2$) et en supposant les pas de temps constant, par sous-domaine on peut réécrire le système (1.2.21) sous la forme matricielle globale :

$$\begin{bmatrix} \Delta t_1 M_1 & 0 & \Delta t_1 C_1^t \\ 0 & \Delta t_2 M_2 & \Delta t_2 C_2^t \\ \Delta t_1 C_1 & \Delta t_2 C_2 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u}_1^n \\ \ddot{u}_2^n \\ \lambda^n \end{Bmatrix} = \begin{Bmatrix} \Delta t_1 (f_{1,ext}^n - f_{1,int}^n(p u_1^n)) \\ \Delta t_2 (f_{2,ext}^n - f_{2,int}^n(p u_2^n)) \\ \Delta t_1 C_1^p u_1^n + \Delta t_2 C_2^p u_2^n \end{Bmatrix} \quad (1.2.22)$$

1.2.2.2 Décomposition *free-link*

Afin de résoudre un tel système, nous utilisons une technique de décomposition *free-link*. Chaque inconnue \ddot{u}_k peut être vue comme la somme de deux termes :

$$\ddot{u}_k = \ddot{u}_{k,free} + \ddot{u}_{k,link} \quad \forall k \in \{1, 2\} \quad (1.2.23)$$

où le terme libre $\ddot{u}_{k,free}$ est solution du problème obtenu par application des forces internes et du chargement extérieur, et le terme lié $\ddot{u}_{k,link}$ est solution du problème obtenu par la seule application des forces de liaisons. Le problème (1.2.22) peut ainsi être décomposé en deux sous-problèmes :

1. Un premier sous-problème sans liaison : problème *free*

$$\begin{bmatrix} M_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u}_{1,free}^n \\ \ddot{u}_{2,free}^n \\ \lambda^n \end{Bmatrix} = \begin{Bmatrix} f_{1,ext}^n - f_{1,int}^n ({}^p u_1^n) \\ f_{2,ext}^n - f_{2,int}^n ({}^p u_2^n) \\ 0 \end{Bmatrix} \quad (1.2.24)$$

Ce système consiste à résoudre les équations d'équilibre sur chaque sous-domaine, de manière indépendante, comme s'il était seul. On en déduit notamment les accélérations à l'instant t^n pour tous les degrés de liberté non concernés par l'interface.

2. Un second sous-problème avec liaison : problème *link*

$$\begin{bmatrix} \Delta t_1 M_1 & 0 & -\Delta t_1 C_1^t \\ 0 & \Delta t_2 M_2 & -\Delta t_2 C_2^t \\ -\Delta t_1 C_1 & -\Delta t_2 C_2 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u}_{1,link}^n \\ \ddot{u}_{2,link}^n \\ \lambda^n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 2 \sum_{k=1,2} C_k \ddot{u}_{k,free}^n \end{Bmatrix} \quad (1.2.25)$$

où l'on note $\ddot{u}_{k,free}^n$ le terme ${}^p \ddot{u}_k^n + \frac{\Delta t_k}{2} \ddot{u}_{k,free}^n$ relatif à chacun des deux sous-domaines. On déduit de ce second sous-problème, les corrections à apporter aux accélérations *free* précédemment calculées sur les noeuds d'interfaces. Cela nécessite le calcul des efforts d'interface, via les multiplicateurs de Lagrange. Après condensation du système (1.2.25) sur l'interface, on obtient finalement :

$$H \lambda^n = B \quad (1.2.26)$$

où l'opérateur de condensation H et le second membre B sont donnés par :

$$H = \sum_{k=1}^2 \frac{\Delta t_k}{2} C_k M_k^{-1} C_k^t \quad (1.2.27)$$

$$B = - \sum_{k=1}^2 C_k \left({}^p \ddot{u}_k^n + \frac{\Delta t_k}{2} \ddot{u}_{k,free}^n \right) = - \sum_{k=1}^2 C_k \ddot{u}_{k,free}^n \quad (1.2.28)$$

Remarque 1 : L'utilisation de matrices masses diagonales permet de découpler les accélérations sur les noeuds d'interface des accélérations internes aux sous-domaines. Le problème sans liaison est alors localisé sur les interfaces, ce qui rend le calcul des accélérations *link* et la correction des accélérations *free* très peu coûteux.

1.2.2.3 Généralisation de l'algorithme à n_{ssd} sous-domaines

De manière générale, on peut étendre la démarche de décomposition au cas de n_{ssd} sous-domaines. On résume cette démarche sous la forme d'un algorithme en quatre étapes :

1. Calcul du problème *free* sur chaque sous-domaine

$$M\ddot{u}_{k,free}^n = f_{k,ext}^n - f_{k,int}({}^p u_k^n) \quad (1.2.29)$$

$$\dot{u}_{k,free}^n = {}^p \dot{u}_k^n + \frac{\Delta t_k}{2} \ddot{u}_{k,free}^n \quad \forall k \in \{1, \dots, n_{ssd}\} \quad (1.2.30)$$

$$u_{k,free}^n = {}^p u_k^n \quad (1.2.31)$$

2. Calcul des multiplicateurs de Lagrange avec le problème condensé

$$H\lambda^n = - \sum_{k=1}^{n_{ssd}} C_k \dot{u}_{k,free}^n \quad (1.2.32)$$

3. Calcul du problème *link* sur chaque sous-domaine

$$M\ddot{u}_{k,link}^n = C_k^t \lambda^n \quad \forall k \in \{1, \dots, n_{ssd}\} \quad (1.2.33)$$

4. Remontée aux quantités cinématiques globales

$$\ddot{u}_k^n = \ddot{u}_{k,free}^n + \ddot{u}_{k,link}^n \quad (1.2.34)$$

$$\dot{u}_k^n = \dot{u}_{k,free}^n + \frac{\Delta t_k}{2} \ddot{u}_{k,link}^n \quad \forall k \in \{1, \dots, n_{ssd}\} \quad (1.2.35)$$

$$u_k^n = u_{k,free}^n \quad (1.2.36)$$

Remarque 2 : Cette méthode de couplage dynamique s'applique de manière générale aux différents schémas de Newmark [GC01].

1.2.3 Collage des maillages incompatibles

L'intérêt de la méthode *duale* précédemment décrite est qu'elle s'applique aussi bien au collage de maillage compatible (cf. Figure 1.7a) qu'au collage de maillage incompatible (cf. Figure 1.7b). La spécificité inhérente aux maillages incompatibles se concentre dans les matrices de liaisons. En effet, dans ce cas les matrices C_k ne seront pas booléenne, mais plus complexes car outre la projection des degrés de liberté intérieurs sur l'interface, elles rendront compte d'un couplage entre ces degrés de liberté. On se propose alors de décrire la manière dont sont obtenues ces matrices C_k dans le cas général d'un collage de maillages incompatibles.

1.2.3.1 Notion d'interface et de collage

Comme nous l'avons vu, le caractère dual de la méthode introduit une variable supplémentaire que l'on se doit de discrétiser. L'interface entre deux sous-domaines, que nous noterons Γ_l est donc constituée de trois surfaces discrètes distinctes comme présentées sur la figure 1.7c :

- ⇨ Deux surfaces réelles S_1, S_2 frontières des sous-domaines en vis-à-vis.
- ⇨ Une surface virtuelle S_Λ support des multiplicateur de Lagrange.

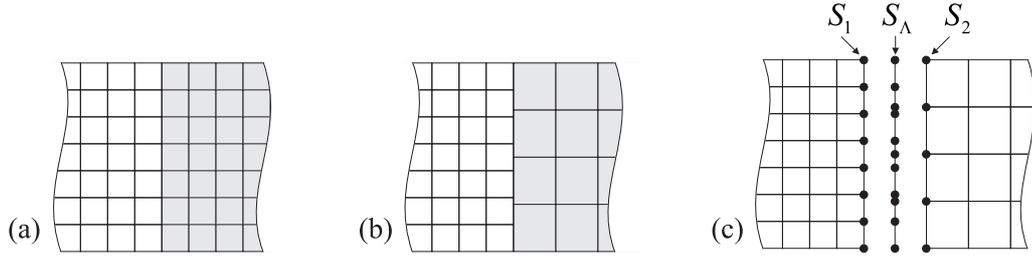


Figure 1.7: Liaison de collage permanent : (a) Maillages compatibles - (b) Maillages incompatible - (c) Définition de l'interface.

Nous ferons ici l'hypothèse que les domaines discrets et continus sont identiques de manière à ce que ces trois surfaces soient géométriquement confondues. Leur discrétisation pourra néanmoins être différente.

Intéressons nous maintenant à la continuité cinématique que l'on cherche à vérifier entre les deux sous-domaines. Au sens continu en espace, cette continuité peut s'écrire :

$$\dot{u}_1 = \dot{u}_2 \quad \text{sur} \quad \Gamma_l \quad (1.2.37)$$

Dans le cas discret et notamment dans le cas de maillage incompatible le collage ne peut toutefois plus s'exprimer de manière aussi intuitive. On introduit la grandeur duale, qui rappelons-le est homogène à un effort, pour exprimer le puissance à l'interface issu d'un défaut de continuité :

$$\mathcal{P}^{int} = \int_{\Gamma_l} \lambda \cdot (\dot{u}_1 - \dot{u}_2) d\Gamma \quad (1.2.38)$$

La continuité cinématique parfaite entre les sous-domaines, est assurée en imposant la nullité de la puissance des efforts d'interface (1.2.38).

1.2.3.2 Construction des matrices de contraintes

La construction des matrices de liaisons s'obtient en discrétisant la puissance d'interface (1.2.38). Pour les grandeurs primales, on s'appuie logiquement sur la discrétisation du bord des sous-domaines respectifs auxquels elles se rattachent. Pour les grandeurs duales on s'appuie sur une discrétisation intermédiaire dont le choix sera discuté par la suite. Ainsi, nous pouvons poser :

$$\dot{u}_1 = \sum_{i=1}^{n_1} N_{1i} \dot{u}_{1i} \quad \text{sur} \quad S_1 \quad (1.2.39)$$

$$\dot{u}_2 = \sum_{j=1}^{n_2} N_{2j} \dot{u}_{2j} \quad \text{sur} \quad S_2 \quad (1.2.40)$$

$$\lambda = \sum_{k=1}^p N_{\Lambda k} \lambda_k \quad \text{sur} \quad S_\Lambda \quad (1.2.41)$$

où n_1, n_2, p sont respectivement les nombres de degrés de liberté frontières des sous-domaines Ω_1, Ω_2 et de la surface des multiplicateurs de Lagrange, et les fonctions N_1, N_2

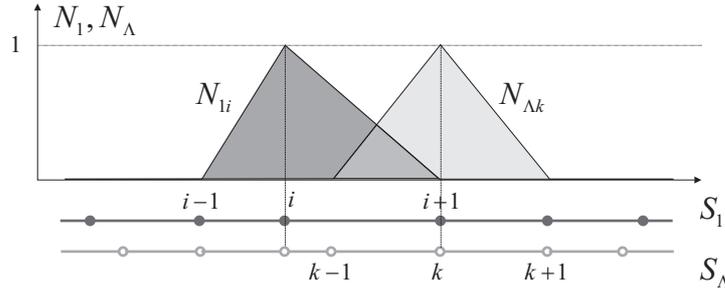


Figure 1.8: Calcul des intégrales de collage (cas d'une interface linéique).

et N_Λ sont respectivement les fonctions de formes associées. Ceci conduit à l'expression approchée de la puissance dissipée à l'interface :

$$\mathcal{P}^{int} = \int_{\Gamma_l} \sum_{k=1}^p N_{\Lambda k} \lambda_k \cdot \left(\sum_{i=1}^{n_1} N_{1i} \dot{u}_{1i} - \sum_{j=1}^{n_2} N_{2j} \dot{u}_{2j} \right) d\Gamma \quad (1.2.42)$$

$$= \lambda^t \cdot (C_1 \dot{u}_1 + C_2 \dot{u}_2) \quad (1.2.43)$$

où les matrices de liaisons C_1 et C_2 sont des matrices rectangulaires de p lignes et de n_1 (resp. n_2) colonnes dont les composantes s'expriment par :

$$C_{1ki} = \int_{\Gamma_l} N_{\Lambda k} \cdot N_{1i} d\Gamma \quad \text{et} \quad C_{2kj} = \int_{\Gamma_l} N_{\Lambda k} \cdot N_{2j} d\Gamma \quad (1.2.44)$$

Ces composantes résultent du couplage entre les multiplicateurs de Lagrange et les noeuds impliqués dans l'interface. Elles sont obtenues par produit des fonctions de formes comme l'illustre la figure 1.8. En annulant la puissance d'interface (1.2.43) on retrouve alors la relation de continuité discrète (p relations) introduite précédemment :

$$C_1 \dot{u}_1 + C_2 \dot{u}_2 = 0 \quad (1.2.45)$$

1.2.3.3 Choix de l'espace des multiplicateurs de Lagrange

Comme nous l'avons vu, la méthode de résolution consiste à condenser le problème sur les multiplicateurs de Lagrange. L'opérateur de condensation H dépend directement de la discrétisation choisie pour ces multiplicateurs, en terme de nombre de noeuds p et en terme de localisation géométrique. Ainsi, la validité de la méthode résulte d'un choix judicieux de cet espace des multiplicateurs de Lagrange. La discrétisation devra assurer une continuité cinématique entre les sous-domaines, en annulant (1.2.38) mais aussi et surtout assurer la non-singularité de l'opérateur H .

On trouve dans la littérature différentes manières de discrétiser la surface S_Λ . Dans tous les cas, l'espace de discrétisation choisi doit vérifier la condition de Ladyzenskaia-Babuska-Brezzi [BF91] afin d'assurer une solution unique du problème couplé. De manière simplifiée, cette condition implique de ne pas prendre un espace de multiplicateur trop riche si bien qu'on se limite à un nombre de relations p inférieurs au nombre total de degrés de liberté impliqués dans l'interface. A l'inverse, on conçoit aisément, que moins

l'espace est riche, moins la continuité cinématique est vérifiée [Rou90]. Si le nombre de multiplicateurs est inférieur à la discrétisation la plus grossière, on comprend qu'il y aura un manque d'informations transmises entre les sous-domaines. Nous choisirons donc :

$$\min(n_1, n_2) \leq p \leq n_1 + n_2 \quad (1.2.46)$$

Comment choisir les p noeuds pour les multiplicateur de Lagrange λ et leurs fonctions de formes associées N_Λ pour obtenir la meilleure continuité cinématique entre les sous-domaines ?

Méthode des multiplicateurs de Lagrange localisés (LLM) Parmi les méthodes d'interfaçage de maillages incompatibles plus répandues, on trouve la méthode des multiplicateurs de Lagrange localisés développée à l'origine pour les problèmes de contact-impact. Elle propose une règle originale de positionnement spatial de ces multiplicateurs de Lagrange sur une surface appelée *frame* matérialisant l'interface. Les contraintes supportées par ce *frame*, considéré comme objet indépendant soumis aux efforts des sous-domaines voisins (cf. Figure 1.9b), sont supposées constantes. Sous cette hypothèse, la méthode consiste à positionner les noeuds du *frame* de manière à ce qu'ils vérifient la règle du moment nul (cf. Figure 1.9c) [PF00]. Ceci étant, bien que cette règle s'avère pragmatique et simple d'utilisation en 2D, l'extension au cas 3D est loin d'être évidente.

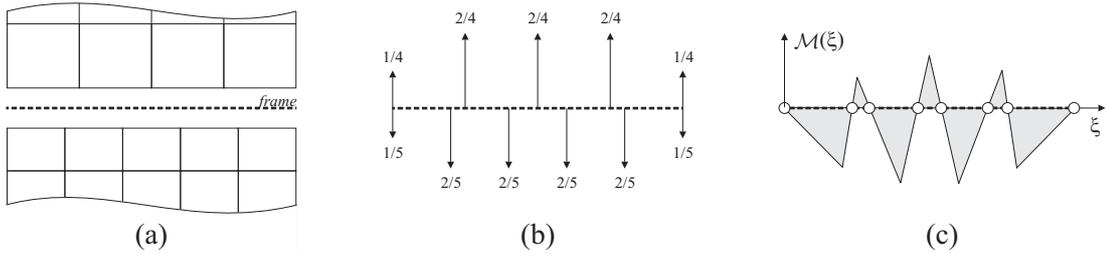


Figure 1.9: Méthode des multiplicateurs de Lagrange localisés : (a) Définition du *frame* - (b) Effort soumis au *frame* - (c) Positionnement des noeuds aux racines du moment.

Méthode de Mortar Une seconde méthode de couplages de maillages incompatibles très répandue et considérée comme très robuste est la méthode Mortar. Développée il y a une dizaine d'années dans une formulation standard [BMP94], elle est désormais développée dans le cadre d'une formulation *duale*. Elle consiste à retenir pour maillage de multiplicateur de Lagrange, la copie de l'une ou l'autre des frontières impliquées dans l'interface. On distingue alors la méthode de Mortar « grossière » et la méthode de Mortar « fine » suivant que l'espace de multiplicateur de Lagrange choisi est identique au sous-domaine maillé le plus grossièrement ou le plus finement (cf. Figure 1.10a-1.10b)).

Méthode optimale Dans le cadre de nos travaux, la méthode de discrétisation utilisée pour le collage de maillage incompatible et celle établie dans [HDVC02]. Elle propose comme support des multiplicateurs de Lagrange l'ensemble des noeuds disponibles sur l'interface en fusionnant les noeuds communs aux deux sous-domaines. Dans l'hypothèse d'interfaces planes entre des éléments finis linéaires, on montre que ce choix de discrétisation conduit à un collage parfait [Her02]. En effet, la puissance \mathcal{P}^{int} calculée au sens

continu (1.2.38) est nulle une fois recombinaés les champs de vitesses à l'aide des fonctions d'interpolations. Cette méthode est alors qualifiée d'optimale, car outre le fait d'assurer un collage parfait de l'interface, elle assure la non-singularité de l'opérateur de condensation H . (un nombre inférieur ne permet plus de parvenir au collage parfait et un nombre supérieur conduit à un problème d'interface singulier).

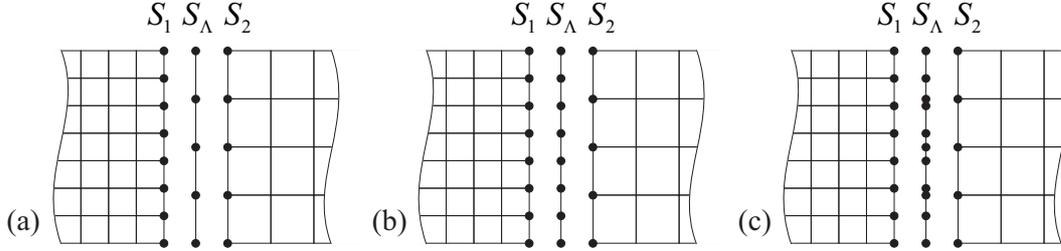


Figure 1.10: Choix de la discrétisation S_Λ : (a) Méthode Mortar grossière - (b) Mortar fine - (c) Méthode optimale.

1.2.3.4 Collage de modélisations incompatibles

Jusqu'ici le collage spatial concernait des éléments finis de même type, seule leur discrétisation était différente dans le cas incompatible. Toutefois, l'utilisation de la modélisation la plus pertinente sur chaque sous-domaine, ce qui rappelons-le, constitue un avantage majeur des méthodes de décomposition, mène à l'étude du couplage d'éléments finis de type différents (coques-poutres, coques-volumes ...). C'est pourquoi, la première partie de mes recherches consistait à étendre aux modèles incompatibles cette méthode de collage particulière initiée par B. Herry [Her02].

Nous nous sommes alors principalement intéressés au collage coques-poutres, situation fréquemment rencontrée dans le monde industriel. En effet, pour certains calculs de prédimensionnement, il est courant d'utiliser des modèles hybrides où certaines parties de la structure, tels que les corps creux, sont modélisés par des éléments poutres de caractéristiques équivalentes. Cette modélisation simplifiée permet alors d'alléger considérablement la taille des modèles.

Comme dans le cas classique, on introduit une surface S_Λ support des multiplicateurs de Lagrange, entre l'extrémité S_1 du corps creux maillé en éléments coque et l'extrémité S_2 de l'élément poutre (la surface S_2 sera ici réduite à un point ponctuel). La surface S_Λ aura assez naturellement la même géométrie que le bord coque S_1 (cf. Figure 1.11a). Sa discrétisation pourra néanmoins être différente; nous proposons par exemple de conserver un noeud sur deux de S_1 ainsi que tous les noeuds de la section placés aux « coins » d'angle supérieur à 45° .

Une fois la surface S_Λ construite, l'idée est ensuite de définir les deux collages, l'un coté coque ($S_\Lambda - S_1$), l'autre coté poutre ($S_\Lambda - S_2$). En pratique, ces collages résultent de l'écriture des matrices C_1 et C_2 (1.2.44) liant les multiplicateurs de Lagrange de S_Λ aux inconnues associés aux faces S_1 et S_2 . Afin de déterminer ces matrices, revenons sur la puissance \mathcal{P}^{int} (1.2.38) ou le travail \mathcal{W}^{int} d'interface dissipé à l'interface. On suppose que

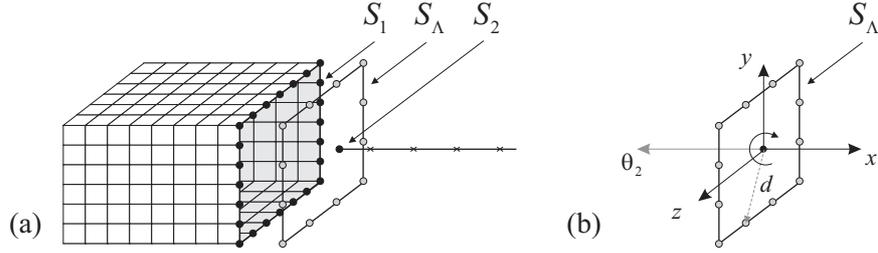


Figure 1.11: Couplage coque-poutre : (a) Discrétisation de l'interface - (b) Repère local de l'interface.

chaque noeuds d'interface dispose de 6 ddls, de manière à pouvoir distinguer les 3 ddls de translations u_x, u_y, u_z , des 3 ddls de rotations $\theta_x, \theta_y, \theta_z$. On peut désormais réécrire, le travail d'interface continue sous la forme $\mathcal{W}^{int} = \mathcal{W}_1^{int} - \mathcal{W}_2^{int}$, avec :

$$\mathcal{W}_1^{int} = \int_{\Gamma_l=S_\Lambda} \left[\lambda_{u_x}^t u_{1x} + \lambda_{u_y}^t u_{1y} + \lambda_{u_z}^t u_{1z} + \lambda_{\theta_x}^t \theta_{1x} + \lambda_{\theta_y}^t \theta_{1y} + \lambda_{\theta_z}^t \theta_{1z} \right] d\Gamma \quad (1.2.47)$$

$$\mathcal{W}_2^{int} = \int_{\Gamma_l=S_\Lambda} \left[\lambda_{u_x}^t u_{2x} + \lambda_{u_y}^t u_{2y} + \lambda_{u_z}^t u_{2z} + \lambda_{\theta_x}^t \theta_{2x} + \lambda_{\theta_y}^t \theta_{2y} + \lambda_{\theta_z}^t \theta_{2z} \right] d\Gamma \quad (1.2.48)$$

Côté poutre, on notera la difficulté d'évaluer l'intégrale (1.2.48) sur la surface S_Λ où les ddls $u_{2x}, u_{2y}, u_{2z}, \theta_{2x}, \theta_{2y}, \theta_{2z}$ ne sont à priori pas définis. L'idée est alors de transposer la cinématique de l'extrémité S_2 à la cinématique de la section S_Λ . Les relations liant les déplacements d'un point $X(x, y, z) \in S_\Lambda$ aux déplacements et rotations de S_2 dans le repère local de l'interface (cf. Figure 1.11b) sont :

$$u_{2x}|_{S_\Lambda} = u_{2x}|_{S_2} + z \theta_{2y}|_{S_2} + y \theta_{2z}|_{S_2} \quad (1.2.49)$$

$$u_{2y}|_{S_\Lambda} = u_{2y}|_{S_2} - z \theta_{2x}|_{S_2} \quad (1.2.50)$$

$$u_{2z}|_{S_\Lambda} = u_{2z}|_{S_2} + y \theta_{2x}|_{S_2} \quad (1.2.51)$$

Les matrices de couplage C_1 et C_2 peuvent alors être déterminées en discrétisant chacune des relations (1.2.47) et (1.2.48). Nous adopterons par conséquent les approximations suivantes pour les déplacements, les rotations et les multiplicateurs de Lagrange, relatifs à S_1, S_2 et S_Λ :

$$\begin{aligned} u_1(X) &= \sum_{i=1}^{n_1} N_{1i}(X) u_{1i} & u_2(X) &= u_2 + \theta_2 \wedge d(X) \\ \theta_1(X) &= \sum_{i=1}^{n_1} N_{1i}(X) \theta_{1i} & \theta_2(X) &= \theta_2 \\ \lambda(X) &= \sum_{k=1}^p N_{\Lambda k}(X) \lambda_k \end{aligned} \quad (1.2.52)$$

où n_1, p désigne les nombres de noeuds appartenant à S_1 et S_Λ . Le terme $d(X)$ désigne la distance entre un point X de la section et l'extrémité de la poutre. On remarquera que l'approximation utilisée côté poutre n'est plus de type éléments finis, mais s'apparente

à un modèle de corps rigide. En injectant les relations (1.2.52) dans (1.2.47-1.2.48), on obtient les formulations discrètes :

$$\mathcal{W}_1^{int} = \lambda_u^t C_{1u} u_1 + \lambda_\theta^t C_{1\theta} \theta_1 \quad (1.2.53)$$

$$\mathcal{W}_2^{int} = \lambda_u^t C_{2u} u_2 + \lambda_u^t C_{2u\theta} \theta_2 + \lambda_\theta^t C_{2\theta} \theta_2 \quad (1.2.54)$$

Côté coque, les matrices de couplage C_{1u} et $C_{1\theta}$ sont calculées de manière toute à fait classique, le collage $S_\Lambda - S_1$ étant le même que dans le cas du couplage coque-coque. Si l'on adopte la même approximation pour les translations et les rotations, on obtient :

$$C_{1u} = C_{1\theta} = \int_{\Gamma_l} N_1(X) N_\Lambda(X) d\Gamma \quad (1.2.55)$$

La spécificité de ce couplage, provient donc des matrices calculées coté poutre. Les opérateurs de couplage C_{2u} et $C_{2\theta}$ associées aux translations et aux rotations, sont ici réduits à des vecteurs calculés comme suit :

$$C_{2u} = C_{2\theta} = \int_{\Gamma_l} N_\Lambda(X) d\Gamma \quad (1.2.56)$$

On voit par ailleurs que (1.2.54) fait apparaître un terme supplémentaire par rapport à (1.2.53). La matrice $C_{2u\theta}$ responsable du couplage entre les ddls de translation et de rotations, pourra s'écrire :

$$C_{2u\theta} = \int_{\Gamma_l} N_\Lambda(X) \cdot d(X) d\Gamma \quad (1.2.57)$$

On remarquera alors que si les ddls de translations et de rotations peuvent être traités indépendamment pour les interfaces de collages coque-coque classiques, ceci n'est plus le cas pour notre interface de collage coque-poutre.

1.2.4 Plusieurs échelles de temps

Nous venons de voir comment s'effectuait le collage spatial aux interfaces entre les différents maillages issus de la décomposition en sous-domaine. Regardons à présent comment s'effectue le collage temporel lorsque les échelles diffèrent entre les sous-domaines.

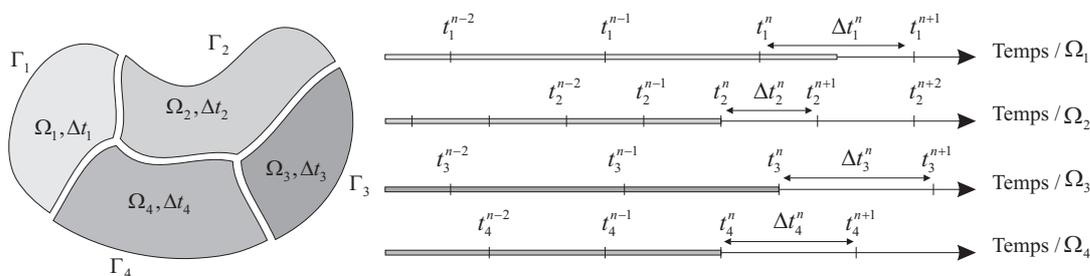


Figure 1.12: Principe du calcul à pas de temps multiples.

1.2.4.1 Motivation de la démarche

Comme nous l'avons vu dans le paragraphe 1.1.3, en dynamique explicite, la taille des éléments conditionne le pas de temps à utiliser pour assurer la stabilité du schéma. Ceci étant, le pas de temps est souvent pénalisé par une minime partie du modèle présentant une discrétisation fine. Tel est le cas des modèles numériques utilisés pour les simulations de crash, qui résulte généralement d'un assemblage de pièces aux maillages très hétérogènes. C'est essentiellement pour s'affranchir de cette contrainte majeure du schéma explicite que l'on s'intéresse aux représentations multi-domaines.

Pour être efficace, l'approche multi-domaines ne doit pas seulement diviser le problème global en sous-problème, elle doit aussi alléger la contrainte sur la stabilité de l'intégration, là où c'est possible. Alors qu'une approche mono-domaine pénalise indifféremment l'ensemble de la structure, l'approche multi-domaines permet d'adopter le pas de temps le plus économique sur chaque sous-structure (cf. Figure 1.12). Nous percevons ici l'intérêt d'une telle approche où chaque sous-domaine est intégré de manière optimale. Le découpage devra toutefois être judicieux de manière à isoler les zones du modèle les plus pénalisantes. Nous remarquerons au passage l'intérêt des liaisons aux maillages incompatibles présentées précédemment. Nous parlerons par la suite de calcul multi-pas-de-temps ou multi-échelles en temps.

Évidemment, une telle démarche n'est pas sans poser problème aux interfaces entre sous-domaines. Les temps de calcul discrets étant différents sur chaque sous-domaine, les équilibres locaux sont par conséquent vérifiés à des instants différents. Un traitement particulier est donc nécessaire pour assurer le « collage temporel » et ainsi résoudre le problème couplé.

La difficulté est donc de mettre en place une méthode capable de gérer les échanges entre sous-domaines, même si ceux-ci ne sont jamais ensemble à l'équilibre. Une difficulté supplémentaire provient de la variabilité des pas de temps au cours du calcul. La démarche présentée ici découle des travaux de thèse [Gra00, Her02]. Elle s'inspire des méthodes de sous-cyclage en temps [BYM79, BSL85] et des méthodes mixtes « implicite-explicite » [LB82].

1.2.4.2 Algorithme multi-échelles en temps

Nous considérons ici le cas de deux sous-domaines chacun dotés d'un pas de temps variable fixé par la condition CFL. Les deux échelles de temps représentées sur la figure

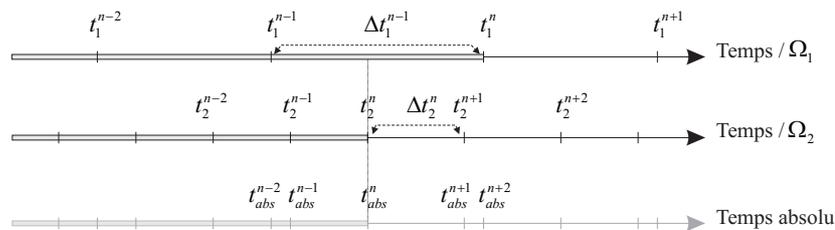


Figure 1.13: Échelles de temps locales et absolue.

1.13 sont donc supposées totalement indépendantes. L'échelle de temps du sous-domaine Ω_1 est ici choisie comme étant l'échelle grossière et celle de Ω_2 comme étant l'échelle fine.

Dans une telle démarche, on initialise le calcul en avançant simultanément les deux sous-domaines avec les pas de temps initiaux Δt_1^0 et Δt_2^0 propres à chacun d'eux. On avance ensuite le sous-domaine « en retard » jusqu'à ce qu'il devance le sous-domaine voisin. Ce dernier peut alors être avancé à son tour. Ainsi de suite, on calcule alternativement chaque sous-système jusqu'au temps de calcul final.

Nous considérerons ici le cas où le calcul sur Ω_2 et en retard sur le calcul Ω_1 (avancé jusqu'à t_1^n). Nous nous placerons donc à l'instant t_2^n où l'on cherche à calculer les quantités cinématiques globales sur Ω_2 . On suppose qu'à cet instant les problèmes *free* ont été résolus sur les deux sous-domaines :

$$M\ddot{u}_{1,free}(t_1^n) = f_{1,ext}(t_1^n) - f_{1,int}({}^p u_1) \quad (1.2.58)$$

$$M\ddot{u}_{2,free}(t_2^n) = f_{2,ext}(t_2^n) - f_{2,int}({}^p u_2) \quad (1.2.59)$$

On dispose donc de toutes les quantités *free*, calculées à l'instant t_1^n pour le sous-domaine Ω_1 et calculées à l'instant t_2^n pour le sous-domaine Ω_2 . Afin de calculer les quantités *link* et ainsi remonter aux quantités globales, revenons sur la condition de continuité (1.2.20) qui à l'instant considéré t_2^n peut s'écrire :

$$C_1 \dot{u}_1(t_2^n) + C_2 \dot{u}_2(t_2^n) = 0 \quad (1.2.60)$$

Les vitesses relatives aux sous-domaines Ω_1 n'étant pas connues à l'instant t_2^n , on procède à une interpolation linéaire entre les instants t_1^{n-1} et t_1^n :

$$\dot{u}_1(t_2^n) = (1 - \alpha_1^n) \cdot \dot{u}_1(t_1^{n-1}) + \alpha_1^n \cdot \dot{u}_1(t_1^n) \quad \text{avec} \quad \alpha_1^n = \frac{t_2^n - t_1^{n-1}}{t_1^n - t_1^{n-1}} \quad (1.2.61)$$

Or, d'après (1.2.35), chaque vitesse peut être décomposée en une quantité *free* et une quantité *link*, de telle sorte que (1.2.61) s'écrive :

$$\dot{u}_1(t_2^n) = (1 - \alpha_1^n) \cdot \dot{u}_{1,free}(t_1^{n-1}) + \alpha_1^n \cdot \dot{u}_{1,free}(t_1^n) + \quad (1.2.62)$$

$$(1 - \alpha_1^n) \cdot \frac{\Delta t_1^{n-1}}{2} \ddot{u}_{1,link}(t_1^{n-1}) + \alpha_1^n \cdot \frac{\Delta t_1^n}{2} \ddot{u}_{1,link}(t_1^n) \\ = (1 - \alpha_1^n) \cdot \dot{u}_{1,free}(t_1^{n-1}) + \alpha_1^n \cdot \dot{u}_{1,free}(t_1^n) + \quad (1.2.63)$$

$$(1 - \alpha_1^n) \cdot \frac{\Delta t_1^{n-1}}{2} M_1^{-1} C_1^t \lambda(t_1^{n-1}) + \alpha_1^n \cdot \frac{\Delta t_1^n}{2} M_1^{-1} C_1^t \lambda(t_1^n)$$

On introduit ensuite le paramètre δ_1^n , qui exprime la variabilité des pas de temps. Ce paramètre s'écrit sous la forme :

$$\delta_1^n = \frac{\Delta t_1^{n-1}}{\Delta t_1^n} \quad (1.2.64)$$

On peut alors écrire :

$$\dot{u}_1(t_2^n) = \dot{u}_{1,free}(t_2^n) + \frac{\Delta t_1^n}{2} M_1^{-1} C_1^t [(1 - \alpha_1^n) \delta_1^n \lambda(t_1^{n-1}) + \alpha_1^n \lambda(t_1^n)] \quad (1.2.65)$$

$$= \dot{u}_{1,free}(t_2^n) + \frac{\Delta t_1^n}{2} M_1^{-1} C_1^t [(1 - \alpha_1^n)(\delta_1^n - 1) \lambda(t_1^{n-1}) + \lambda(t_2^n)] \quad (1.2.66)$$

$$= \dot{u}_{1,free}(t_2^n) + (1 - \alpha_1^n)(\delta_1^n - 1) \frac{\Delta t_1^n}{2} \ddot{u}_{1,link}(t_1^{n-1}) \quad (1.2.67)$$

$$+ \frac{\Delta t_1^{n-1}}{2} M_1^{-1} C_1^t \lambda(t_2^n)$$

La vitesse $\dot{u}_{1,free}(t_2^n)$ est interpolée entre les instants t_1^{n-1} et t_1^n , où toutes les quantités *free* sont connues. En injectant cette dernière expression dans l'équation de continuité (1.2.60), on aboutit finalement au problème condensé, qui à l'instant t_{abs}^n de l'échelles absolue devient :

$$H \lambda(t_{abs}^n) = B \quad (1.2.68)$$

où l'opérateur de condensation H et le second membre B sont donnés par :

$$H = \sum_{k=1}^2 \frac{\Delta t_k^{n-1}}{2} C_k M_k^{-1} C_k^t \quad (1.2.69)$$

$$B = - \sum_{k=1}^2 C_k [\dot{u}_{k,free}(t_{abs}^n) + (1 - \alpha_k^n)(\delta_k^n - 1) \frac{\Delta t_k^{n-1}}{2} \ddot{u}_{k,link}(t_{abs}^{n-1})] \quad (1.2.70)$$

Comme auparavant, cette écriture se généralise aisément au cas d'un nombre quelconque de sous-domaine. Nous remarquerons par ailleurs la correction apportée au second membre du problème condensé par rapport au cas mono-échelles définis par (1.2.28). Cette correction est due d'une part à la variabilité du pas de temps δ_k^n et d'autre part au coefficient d'interpolation α_k^n ; elle s'annule donc en cas de calcul à pas de temps unique ($\alpha_k^n = 1$) ou lorsque le pas de temps reste constant par sous-domaine ($\delta_k^n = 1$).

On montre dans [GC01] que la stabilité globale de l'algorithme n'est pas affectée par la décomposition en sous-domaine. Cette étude de stabilité, appliquée de manière générale aux couplages de schéma Newmark de différentes natures, est reprise dans [Her02] dans le cas du schéma explicite. La démonstration, basée sur une méthode énergétique [Hug01], justifie du choix des vitesses discrètes en temps pour exprimer la continuité cinématique (1.2.20). Il est par ailleurs prouvé que l'utilisation d'échelles de temps différents conduit à la dissipation d'énergie à l'interface entre les sous-domaines, ce qui par conséquent n'affecte pas la stabilité de l'intégration toujours dictée par la condition CFL (1.1.57) à l'intérieur de chaque sous-domaine. Les exemples présentés dans [GC01] laissent à penser que cette dissipation artificielle reste faible et n'altère que très peu la solution du problème multi-domaines.

1.2.4.3 Gestion des pas de temps variables

Les pas de temps étant variables, il est peu probable de trouver des instants d'équilibres, communs au différents sous-domaines, où la continuité cinématique serait calculée de manière exacte (sans interpolation). Ce problème observé dans le cas de deux sous-domaines (cf. Figure 1.13) est d'autant plus critique pour un nombre quelconque de sous-domaines.

La suite d'interpolations linéaires à la base de cette approche multi-échelles produit une accumulation d'erreurs pouvant se révéler préjudiciable sur la qualité de la solution.

Afin de limiter et contrôler cette dérive, une gestion originale des pas de temps a été développée dans [Her02]. L'idée est de définir régulièrement des instants de calculs, appelés « points de rendez-vous » pour lesquels tous les sous-domaines sont autoritairement mis à l'équilibre. Il convient pour cela d'adapter le pas de temps de chaque sous-domaine à l'approche de ces « points de rendez-vous ». Pour ne pas affecter la stabilité des calculs, l'ajustement des pas de temps devra se faire à la baisse de manière à ce que la condition CFL soit toujours vérifiée. Si pour un sous-domaine Ω_k , l'instant final t_k^{n+1} du cycle considéré est postérieur au point de rendez-vous, le pas de temps Δt_k^n est réduit de manière à ce que le nouvel instant final coïncide avec l'instant de rendez-vous.

Néanmoins, en procédant de la sorte, le pas de temps coupé peut constituer une infime fraction du pas de temps de stabilité, notamment lorsque l'instant de calcul t_k^{n+1} est infiniment proche de l'instant de rendez-vous t^{rdv} . Ceci conduit à des variations brutales du pas de temps pouvant générer des instabilités. Pour palier à cela, il convient d'anticiper les instants de rendez-vous de manière à réguler la variation du pas de temps sur plusieurs cycles. Par exemple, en se projetant deux pas de temps en avant, trois cas de figures se présentent :

1. Si $t_k^n + 2\Delta t_k^n < t^{rdv}$ le pas de temps imposé reste le pas de temps de stabilité.
2. Si $t_k^n + 2\Delta t_k^n > t^{rdv}$ mais $t_k^n + \Delta t_k^n < t^{rdv}$ le pas de temps imposé correspond à la demi distance entre l'instant de rendez vous et l'instant courant, soit $\Delta t_k^n = (t^{rdv} - t_k^n)/2$.
3. Si $t_k^n + \Delta t_k^n > t^{rdv}$, le pas de temps imposé correspond à la distance entre l'instant de rendez vous et l'instant courant, soit $\Delta t_k^n = t^{rdv} - t_k^n$

Nous remarquerons que cette régulation à l'approche d'un instant de rendez-vous suppose un pas de temps de stabilité à l'instant t_k^{n+1} identique à celui de l'instant courant t_k^n , ce qui n'est pas forcément le cas. Toutefois, la variabilité du pas de temps étant généralement très faible, nous considérerons que cette hypothèse est vérifiée en première approximation.

1.3 Implantation dans RAD2RAD

Pour terminer ce premier chapitre, voyons de façon plus pratique comment la méthode multi-domaines précédemment décrite a été implémentée. Cette structure multi-domaines et la programmation de l'algorithme multi-pas-de-temps étant antérieure à ce travail, nous nous contenterons ici de faire une brève présentation du code de calcul existant, que l'on appelle RAD2RAD.

Ce code de calcul, qui servira de support à la plupart de nos travaux de recherche, a initialement été développé par la société MECALOG. La première version très simplifiée de ce programme [Wro99], qualifié alors de « programme principal », avait pour but de coupler différents calculs de dynamique explicite issus du logiciel RADIOSS [Rad05]. Le code est actuellement capable de gérer le collage permanent entre sous-domaines aux maillages compatibles ou incompatibles dotés de pas de temps différents. Seules les interfaces linéiques entre éléments de type coque, ou ponctuelles entre éléments de type poutre sont actuellement prises en compte. L'extension au collage coque-poutre (cf. §1.2.3.4) entrepris dans [Her02] à été complétée puis achevée au cours de cette thèse.

Nous commencerons par présenter dans un premier temps l'organisation des calculs couplés RADIOSS-RADIOSS. Nous y décrirons notamment la structure du code de couplage RAD2RAD et son mécanisme de communications avec les différents process RADIOSS. Un exemple d'application viendra alors illustrer mais surtout valider notre approche multi-domaines. Nous nous intéresserons ensuite au problème d'interface dont la résolution constitue la majeure partie du temps CPU consommé par RAD2RAD. Enfin nous clôturerons ce chapitre en évoquant le problème de parallélisation des calculs, souvent indissociable des méthodes de décomposition de domaine. Après avoir rappelé quelques notions fondamentales du calcul parallèle, nous aborderons les différents niveaux de parallélisme rencontrés dans une telle architecture de couplage.

1.3.1 Organisation du calcul couplé

Dans notre approche multi-domaines chaque sous-domaine est traité par un process RADIOSS indépendant. Chacun de ces process correspondant à un classique calcul de dynamique explicite constitue le calcul *free* défini par les relations (1.2.29) à (1.2.31). L'ensemble de ces calculs RADIOSS, sont couplés via le programme externe, RAD2RAD (pour « RADIOSS to RADIOSS »), dont la principale fonction est de gérer les échanges de données entre les processus. Ce programme doit aussi gérer le bilan des forces et des moments sur les interfaces, afin d'assurer la liaison entre les différents sous-domaines. Il se charge ainsi de récupérer les grandeurs *free* calculées par chaque process RADIOSS, de résoudre le problème d'interface (+ calcul *link*) puis de leur retourner les grandeurs cinématiques corrigées.

Il est important de remarquer que l'ensemble des communications sont centralisées par RAD2RAD (cf. Figure 1.14) et que les différents process ne communiquent pas directement entre eux. Regardons à présent comment s'opèrent les échanges de données entre les différents sous-domaines.

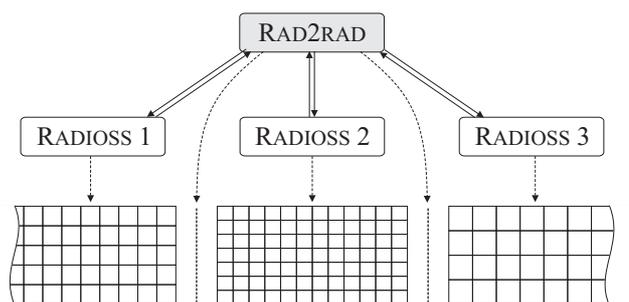


Figure 1.14: Principe du couplage RADIOSS-RADIOSS via le module RAD2RAD.

1.3.1.1 Mécanisme de communication

Les communications entre process RADIOSS, via RAD2RAD, s'opèrent au travers un mécanisme de type *pipe*. Ce mécanisme, très utilisé sous UNIX pour faire communiquer différents processus entre eux, permet d'envoyer les données directement en mémoire sans être stockées temporairement sur disque, ce qui est donc très rapide.

Les *pipes* que nous utilisons ici sont des *pipes* nommées, encore appelées « FIFOs » (first-in-first-out). Il s'agit de fichiers spéciaux, de taille nulle, qui une fois ouverts se comportent comme des tuyaux classiques. Bien que la façon de les utiliser puisse se révéler trompeuse, les tuyaux nommés transfèrent bien leurs données d'un processus à l'autre en mémoire, sans aucun stockage. L'utilisation du système de fichiers permet uniquement de pouvoir accéder au tuyau par l'intermédiaire de son nom, ce qui rend possible la communication entre une multitude de processus indépendants.

Comme les *pipes* classiques, ce protocole d'échanges ne permet toutefois qu'une communication unidirectionnelle. Chaque côté du tuyau est un descripteur de fichier ouvert soit en lecture, soit en écriture. Ceci justifie l'ouverture de deux *pipes* par process RADIOSS, l'un pour la lecture de données, l'autre pour l'écriture. A titre d'exemple, on considère un sous-domaine, appelé *PartA*, traité par un processus RADIOSS indépendant. A son lancement, RAD2RAD ouvre donc les deux *pipes* suivantes :

- ⇒ *Fifo-PartA-1* pour la communication dans le sens RADIOSS ▷ RAD2RAD (écriture par le processus RADIOSS, lecture par le processus RAD2RAD)
- ⇒ *Fifo-PartA-2* pour la communication dans le sens RAD2RAD ▷ RADIOSS (écriture par le processus RAD2RAD, lecture par le processus RADIOSS)

Nous remarquerons que la lecture d'un tuyau est destructrice, c'est-à-dire que si plusieurs processus lisent le même tuyau, toute donnée lue par l'un disparaît pour les autres. L'envoi d'une même information à plusieurs processus, nécessite donc de créer un tuyau vers chacun d'eux. D'autre part, nous remarquerons que la lecture d'un tuyau est bloquante, c'est-à-dire que si aucune donnée n'est disponible en lecture, le processus essayant de lire le tuyau sera suspendu jusqu'à ce que des données soient disponibles. L'utilisation de cette caractéristique comme effet de bord permet la synchronisation des processus (les processus lecteurs étant synchronisés sur les processeurs écrivains). Ceci est particulièrement utile pour mettre en attente les sous-domaines dont les instants d'équilibre locaux t_k^n sont en avance sur le temps courant de l'échelle absolue t_{abs}^n (cf. §1.2.4).

Remarque 1 : L'utilisation de ce protocole de communication présente toutefois un inconvénient majeur sur le plan de la parallélisation des calculs, puisque les *pipes* ainsi définis n'autorisent que des communications propres à une seule et même machine. Leurs utilisations sur une ferme de machines parallèles et distantes demeurent dès lors impossibles. L'utilisation d'autres techniques, telles que des communications par passage de message de type MPI (Message Passing Interface) devront toutefois être envisagées pour pouvoir prétendre à une future industrialisation de ce code.

1.3.1.2 Structuration des données RAD2RAD

Pour que le calcul couplé soit efficace, les variables échangées entre les différents programmes concourants doivent être réduites au strict minimum. L'algorithme de couplage proposé ici présente l'avantage de n'utiliser que l'échange d'un nombre réduit de valeurs entre sous-domaines. La plupart des variables traitées dans RAD2RAD sont des variables nodales (relatives uniquement aux noeuds de liaisons) et non élémentaires (variables internes, contraintes, déformations...), ce qui limite fortement le nombre d'échanges avec RADIOSS.

Les deux principales entités manipulées dans RAD2RAD sont les sous-domaines (PART) et les interfaces (INTER). Naturellement, chaque sous-domaine n'est représenté dans le code que par ses frontières de liaisons (LINK) avec les sous-domaines connexes. Tel est l'intérêt des méthodes de décomposition sans recouvrement qui, rappelons-le, consiste à condenser le problème sur les noeuds frontières de liaisons. Les interfaces, au sens RAD2RAD, définissent la façon dont sont liés entre eux les différents sous-domaines. Comme nous l'avons vu dans le paragraphe 1.2.3, chaque interface se compose de deux surfaces réelles (ou liens) correspondant aux frontières des sous-domaines en vis-à-vis, et d'une surface fictive support des multiplicateurs de Lagrange. Autrement dit, chaque entité INTER est composée de deux LINK et d'une discrétisation additionnelle (cf. Figure 1.15). Un fichier de données propre au couplage, définit l'interconnexion des liens (repérés par un identifiant) issus des différents sous-domaines.

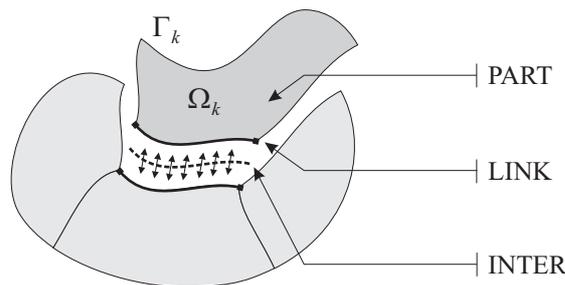


Figure 1.15: Illustration des entités de calcul PART, LINK et INTER.

Dans le code, chacune de ces entités constitue une structure de données. Ces structures sont composées d'un ensemble de variables ou éventuellement d'autres structures de données imbriquées. La figure 1.16 présente à titre d'exemple, les structures simplifiées PART, LINK et INTER correspondantes aux entités sous-domaine, liaison et interface.

INTER		PART		LINK	
Types	Variables	Types	Variables	Types	Variables
Entier	Nombre de liens	Mots	Noms	Entier	Identifiant de liaison
LINK	Liens attachés	Entier	Nombre de liens	Entier	Nombre de noeuds
NODE	Noeuds de S_Λ	LINK	Liens attachés	NODE	Noeuds de liaison
Réel	Vecteur λ	Réel	Temps local	ELEM	Élément de liaison
Réel	Vecteur B local	Réel	Pas de temps local	Réel	Matrice de liaison
Reel	Matrice H local			Entier	Qtés cinématiques
				Réel	Masse, Inertie
				Entier	Connectivités

Figure 1.16: Exemple de structures de données RAD2RAD.

1.3.1.3 Données échangées

Après avoir vu comment étaient échangées les données entre RADIOSS et RAD2RAD, regardons à présent plus précisément quelles sont les variables échangées entre les deux codes. Dans RAD2RAD nous distinguons la phase d'initialisation de la phase de calcul proprement dite. Nous noterons \triangleright les communications établies dans le sens RADIOSS-RAD2RAD et \triangleleft les communications dans le sens RAD2RAD-RADIOSS.

La phase d'initialisation correspond principalement à l'ouverture des communications *pipes* ainsi qu'à la lecture des données. Cette lecture se fait d'une part à partir du fichier de couplage (nombre de liens par sous-domaine, puis interconnexion des différents sous-domaines), et d'autre part à partir des communications établies avec chacun des process clients. Au cours de cette phase, l'échange d'informations a donc exclusivement lieu dans le sens RADIOSS-RAD2RAD. Pour chaque lien d'un sous-domaine donné, les principales variables envoyées par le process RADIOSS correspondant sont :

- ① \triangleright L'identifiant du lien.
- ② \triangleright Le nombre de noeuds constituant le lien.
- ③ \triangleright Les numéros des noeuds.
- ④ \triangleright Les coordonnées des noeuds.
- ⑤ \triangleright Les masses et inerties aux noeuds.

Après avoir renseigné ces différentes variables, il est possible de construire la surface S_Λ support des multiplicateurs de Lagrange, ainsi que les matrices de liaisons C_k et de condensation locale H_k .

Durant la phase de calcul proprement dite, les échanges s'effectuent à chaque pas de temps. Chaque process RADIOSS est chargé d'envoyer les accélérations *free* aux noeuds de liaisons ainsi que son pas de temps de stabilité (qui rappelons-le est variable) et de récupérer les accélérations corrigées définies par l'équation (1.2.34) :

- ⑥ \triangleright Les accélérations *free* aux noeuds de liaisons.
- ⑦ \triangleright Le pas de temps local au sous-domaine.
- ⑧ \triangleleft Le pas de temps local au sous-domaine (éventuellement corrigé).
- ⑨ \triangleleft Les accélérations corrigées aux noeuds de liaisons.

La communication du pas de temps est nécessaire pour la construction des opérateurs H et B du problème condensé. En multi-échelles, le pas de temps propre à chaque sous-domaine n'est pas modifié dans RAD2RAD, si ce n'est que pour atteindre un éventuel « point de rendez-vous ». La communication ⑦ est donc principalement utilisée en cas de calcul mono-échelle où seul le pas de temps minimal est retenu puis communiqué à l'ensemble des sous-domaines en présence.

1.3.2 Exemple de calcul couplé

Exposons à présent un exemple de calcul couplé afin d'illustrer et valider notre méthode multi-domaines intégré à RAD2RAD. Conformément à la remarque 1 de la section précédente, l'ensemble des tests présentés ici ont été réalisés de manière séquentielle sur machine mono-processeur. Les différents processus RADIOSS ainsi que le processus RAD2RAD s'exécute donc sur un seul et même processeur.

1.3.2.1 Corps creux en compression

On considère un tube de hauteur h , d'épaisseur e et de section S encastré sur sa frontière inférieure et soumis à une vitesse imposé $v_i = 15m/s$ sur sa frontière supérieure. La masse volumique ρ , le module de Young E ainsi que le coefficient de Poisson ν du matériau utilisé (loi élasto-plastique) sont donnés dans le tableau suivant :

Paramètres géométriques	$h = 0.412m$	Paramètres matériau	$\rho = 7.80 \times 10^3 kg/m^3$
	$e = 9.14 \times 10^{-4}m$		$E = 2.10 \times 10^{11} N/m^2$
	$S = 0.0508 \times 0.0381m$		$\nu = 0.3$

Tableau 1.1: Corps creux en compression : paramètres de calcul.

On choisit ici de décomposer le modèle en deux, la PART1 correspondant à la partie basse et la PART2 à la partie haute. L'utilisation d'éléments coque, mène à une interface linéique entre les deux sous-domaines. Nous remarquerons que la face inférieure du tube est biaisée de manière à déclencher une rotation de l'interface au cours de la déformation. Afin de tester les différentes fonctionnalités de notre approche multi-domaines, nous distinguerons quatre cas :

- ⇒ **Cas ①** : Modèle mono-domaine classique (calcul de référence).
- ⇒ **Cas ②** : Modèle multi-domaines à maillages compatibles.
- ⇒ **Cas ③** : Modèle multi-domaines à maillages incompatibles.
- ⇒ **Cas ④** : Modèle multi-domaines à modélisations différentes (coques/poutres).

Le tableau 1.2 détaille le pas de temps initial ainsi que la taille des différents modèles. L'intérêt des cas ③ et ④ porte sur modélisation plus grossière de la partie haute dont l'analyse requiert moins de précision.

La figure 1.17 illustre la déformée obtenue pour chacun des cas au temps $t = 9 \times 10^{-3}s$. Nous remarquerons que la continuité du déplacement reste parfaitement vérifiée au travers des interfaces compatibles et incompatibles. Les courbes de déplacements (cf. Figure 1.18) calculés de part et d'autre de l'interface, conforte ce maintien de continuité dans

		Cas ①	Cas ②	Cas ③	Cas ④
Pas de temps initial	PART1	$2.50 \times 10^{-7} s$			
	PART2	-	$7.70 \times 10^{-7} s$	$1.65 \times 10^{-6} s$	$3.40 \times 10^{-4} s$
Nombre d'éléments	PART1	2056 elts	1032 elts	1032 elts	1032 elts
	PART2	-	1024 elts	256 elts	1 elt

Tableau 1.2: Comparaison des tailles et pas de temps des différents modèles.

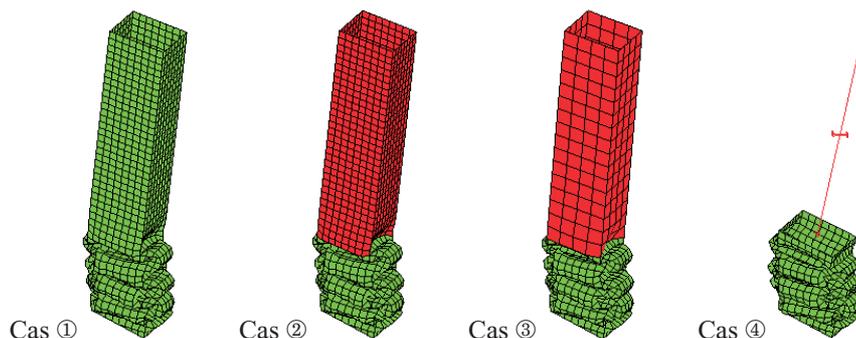


Figure 1.17: Déformées des différents modèles au temps $t = 9 \times 10^{-3} s$.

le cas incompatible ③. Par ailleurs, les courbes de la figure 1.19 correspondent aux vitesses d'un noeud d'interface. Nous noterons l'exactitude du calcul ② pour lequel la courbe est confondue avec la référence, ce qui nous conforte vis-à-vis de la fiabilité de notre approche. L'utilisation d'une modélisation haute plus grossière et de maillages incompatibles conduit par ailleurs à un résultat tout à fait acceptable. Enfin le couplage coque/poutre représenté par le cas ④ est quant à lui logiquement dégradé en fin de simulation, lorsque les lobes de flambement atteignent l'interface. La modélisation poutre, inapte à reproduire l'amorce du flambement, explique les différences observées par rapport au modèle de référence.

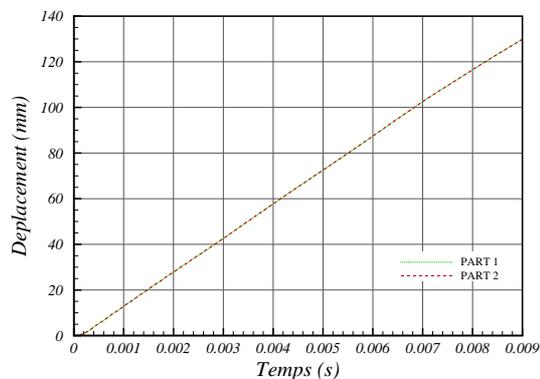


Figure 1.18: Déplacements absolus à l'interface du cas ③.

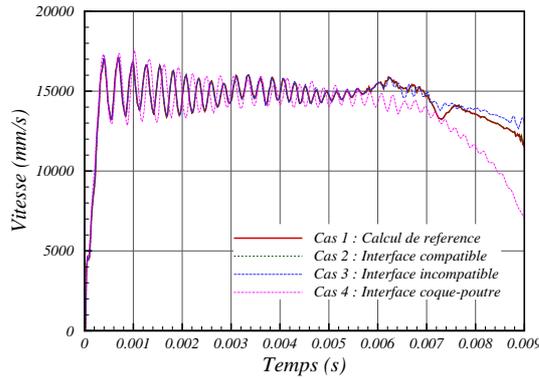


Figure 1.19: Vitesses absolues à l'interface de liaison.

La difficulté est donc de trouver le bon compromis entre les simplifications de modélisation, et le gain en temps de calcul. Cela dépend évidemment de la précision recherchée dans les différentes parties du modèle. Le tableau 1.3 résume les temps de calcul passés dans chacun des sous-domaines en ignorant le temps de calcul passé dans RAD2RAD pour le calcul d'interface. Bien qu'il s'agisse ici d'un modèle de petite taille, on remarque le gain apporté par l'approche multi-domaine ; celui-ci est bien entendu d'autant plus intéressant que la discrétisation de la partie haute (PART2) est grossière. Cette économie de temps CPU sera encore plus évidente sur des modèles industriels comme celui présenté dans le chapitre 2, où de toutes petites parties pénalisent le calcul par leur faible pas de temps.

	Cas ①	Cas ②	Cas ③	Cas ④
PART1	-	177.4 s	173.7 s	168.4 s
PART2	-	55.4 s	9.3 s	9.1 s
<i>Total</i>	318.3 s	232.8 s	186.7 s	177.5 s

Tableau 1.3: Détail des temps CPU consommés par les différents modèles.

1.3.3 Traitement du problème d'interface

1.3.3.1 Optimisation de l'algorithme multi-échelles en temps

Lors de la résolution du problème d'interface entre sous-domaines aux échelles de temps différentes, les grandeurs *link* relatives aux sous-domaines interpolés (dont le temps local ne correspond pas à l'instant de calcul sur l'échelle globale) sont sans intérêt. En effet, seules les quantités cinématiques des sous-domaines exactement calculés seront finalement envoyées aux processus RADIOSS associés. Par conséquent les seules quantités *link* à retenir sont celles évaluées sur ces sous-domaines dont l'instant d'équilibre est celui de l'échelle globale.

Ceci étant, il n'est pas rare de rencontrer au cours du calcul des interfaces ne faisant intervenir à un instant global donné, que des sous-domaines interpolés. Le calcul des multiplicateurs de Lagrange sur ces interfaces devient par conséquent inutile puisque

aucun usage n'en est fait. Pour éviter des opérations d'inversion inutiles, il semble de ce fait judicieux de traiter séparément chaque interface entre sous-domaines deux à deux. On décompose ainsi le problème d'interface globale (1.2.44) en sous-problèmes locaux de plus petites dimensions :

$$H_k \Lambda_k = B_k \quad (1.3.1)$$

où la matrice de condensation H_k et le second membre B_k sont local au sous-domaine Ω_k . Dès lors, chaque sous-système (1.3.1) peut être résolu individuellement que si le temps local à l'un des deux sous-domaines concernés correspond au temps de calcul global.

1.3.3.2 Coût de résolution du problème condensé aux interfaces

Intéressons nous plus particulièrement au problème d'interface dont le traitement conditionne la majeure partie du temps CPU utilisé par le programme RAD2RAD. On considère ici le traitement du problème dans sa globalité, avec évidemment la résolution du système linéaire (1.2.26) mais également avec la construction des opérateurs H et B (1.2.27-1.2.28) qui, nous allons le voir, peut s'avérer très coûteuse. Rappelons que la matrice H est une matrice bande dont la largeur de bande peut être aussi importante que la taille du problème et qu'en toute rigueur celle-ci doit être recalculée à chaque pas de temps.

Cette matrice étant symétrique, la résolution du système linéaire peut être réalisée par la méthode directe de Cholesky. La méthode repose sur la factorisation de la matrice H tel que :

$$H = LL^t \quad (1.3.2)$$

où L est une matrice triangulaire inférieure régulière. La résolution de (1.2.26) revient à résoudre les deux systèmes à matrice triangulaire suivant :

$$LX = B \quad \text{et} \quad L^t \lambda = X \quad (1.3.3)$$

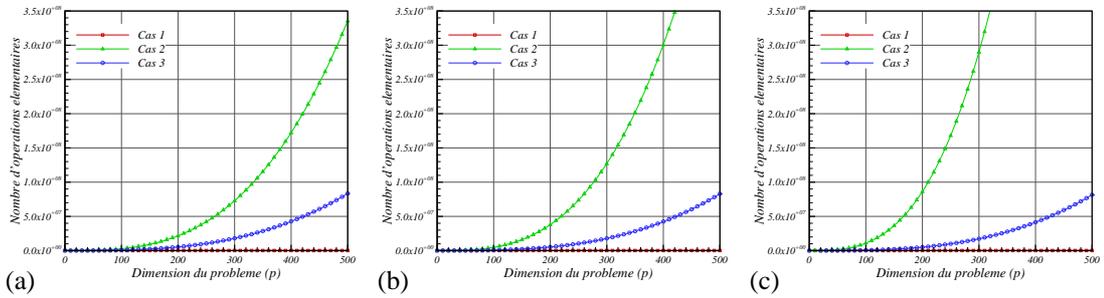
Décompte des opérations élémentaires Le tableau 1.4 détaille pour chaque étape de la résolution, le nombre d'opérations élémentaires effectuées à chaque pas de temps. Nous nous placerons ici dans le cas d'une décomposition en deux sous-domaines avec une interface unique, où p désigne le nombre de multiplicateurs de Lagrange sur la discrétisation de S_Λ , et où m et n désignent respectivement le nombre de ddls sur les frontières S_1 et S_2 . Nous noterons $q = m + n$ le nombre total de ddls compris dans l'interface. Trois cas seront considérés :

- ⇨ **Cas ①** : Il s'agit du cas idéalisé où la liaison a lieu entre maillages compatibles et où par conséquent les matrices de liaisons peuvent être considérées comme booléennes. On a dans ce cas particulier $n = m = p$.
- ⇨ **Cas ②** : Il s'agit du cas le plus général où la liaison a lieu entre maillages incompatibles. Nous nous placerons volontairement ici dans un cas extrême en considérant des matrices de liaisons pleines et en supposant que la matrice H est recalculée à chaque cycle.
- ⇨ **Cas ③** : Il s'agit d'un cas similaire au cas ② sauf que nous ignorons ici l'assemblage complet de la matrice H . Pour chaque sous-domaine, les produits matriciels $C_k M_k^{-1} C_k^t$ sont supposés constants et déterminés une fois pour toute au début du calcul. Le calcul de H revient dans ce cas à faire la somme des deux contributions multipliée par $\Delta t_k/2$.

		Cas ①	Cas ②	Cas ③
Assemblage de la matrice H	+	p	$p^2(q+1)$	—
	×	$6p$	$p^2(2+q) + pq$	—
	÷	—	—	—
Assemblage du vecteur B	+	$3p$	$p(q+1) + q$	$p(q+1) + q$
	×	$6p$	$p(q+2) + q$	$p(q+2) + q$
	÷	—	—	—
Factorisation de Cholesky	+	—	$p(p^2-1)/2$	$p(p^2-1)/2$
	×	—	$p(p^2-1)/6$	$p(p^2-1)/6$
	÷	—	$p(p-1)/2$	$p(p-1)/2$
Résolution des systèmes	+	—	$p(p-1)$	$p(p-1)$
	×	—	$p(p-1)$	$p(p-1)$
	÷	p	$2p$	$2p$
TOTAL		$17p$	$\frac{2}{3}p^3 + (2q + \frac{11}{2})p^2 + (3q + \frac{11}{6})p + 2q$	$\frac{2}{3}p^3 + \frac{5}{2}p^2 - p(\frac{11}{6} + 2q) + 2q$

Tableau 1.4: Nombre d'opérations pour une résolution directe de Cholesky.

Sur la figure 1.20 on trace le nombre d'opérations en fonction de la dimension du problème p qui, rappelons-le, est fonction du nombre total de ddls q . En effet, dans le paragraphe 1.2.3, nous avons vu que p devait être choisi tel que $p < q$ et $p > \min(n, m)$. En supposant que le ratio entre n et m varie dans des proportions raisonnables (typiquement inférieur à 5), nous tracerons les résultats obtenus pour trois valeurs représentatives du rapport q/p caractéristique du niveau d'incompatibilité de la liaison.


Figure 1.20: Évolutions du nombre d'opérations élémentaires pour différentes valeurs du rapport q/p : (a) $q = p$ - (b) $q = 2p$ - (c) $q = 5p$.

Comme attendu, le coût de résolution du problème d'interface dans le cas idéal ① est négligeable. L'utilisation de maillages incompatibles augmente considérablement le nombre d'opérations, et ce d'autant plus que l'incompatibilité est grande (rapport q/p élevé). Nous pouvons toutefois modérer les résultats obtenus dans le cas ② en précisant bien qu'il s'agit d'un cas extrême où les matrices de liaisons sont supposées pleines, ce qui n'est jamais le cas en pratique. Les opérations matricielles entre matrices creuses peuvent être fortement allégées si l'on évite les opérations élémentaires inutiles entre valeurs nulles. D'autre part, ces résultats montrent très nettement le coût additionnel qu'entraîne la construction de la matrice H . En effet, celle-ci semble être au moins aussi importante que la résolution elle-même.

Regardons alors plus précisément sous quelles conditions, la réévaluation complète de la matrice de condensation peut être évitée. En supposant un pas de temps constant et une matrice masse constante, la reconstruction de H dépend de la variabilité des matrices de liaison C_k . Au cours du calcul, les interfaces subissent généralement des transformations qui en toute rigueur modifient les termes de ces matrices.

Invariance de la matrice de condensation H : Nous supposons ici que l'interface linéique Γ_l est soumise à une déformation homogène. Cette hypothèse n'est généralement pas pénalisante puisque l'on cherchera au maximum à positionner les interfaces sur les zones peu sollicitées de la structure. Ceci étant, chaque élément se déformant dans les mêmes proportions, on peut écrire le rapport κ entre la longueur courante l^n et la longueur initiale l^0 de l'interface sous la forme :

$$\kappa = \frac{l^n}{l^0} = \frac{l_{i,i+1}^n}{l_{i,i+1}^0} \quad (1.3.4)$$

où $l_{i,i+1}$ est la distance entre les noeuds i et $i + 1$ participant à l'interface. En notant s l'abscisse curviligne, on a :

$$s_j^n = \sum_{i=0}^{j-1} l_{i,i+1}^n = \kappa \sum_{i=0}^{j-1} l_{i,i+1}^0 = \kappa s_j^0 \quad (1.3.5)$$

D'après (1.2.44), la matrice C_k étant obtenue par produit des fonctions de forme sur l'interface, chacun de ses termes devient égal avant et après transformation au facteur multiplicatif κ près. En conclusion, il sera fortement avantageux de corriger périodiquement la matrice de condensation H dès que la déformation d'interface pourra être considérée comme étant faible ou homogène. Dans le cas contraire, la reconstruction complète de cette matrice sera inévitable.

De même, on montre aisément [Her02] qu'une combinaison de grandes rotations et de grandes translations ne modifie pas les abscisses curvilignes de l'interface. Les matrices de liaisons C_k restent par conséquent inchangées au cours de tel mouvement d'ensemble de Γ_l .

1.3.3.3 Tests de solveurs

La résolution du problème d'interface peut constituer une partie non négligeable du calcul global, ce qui par conséquent peut réduire significativement l'intérêt de la décomposition de domaine. Afin de tester l'efficacité du solveur de Cholesky initialement utilisé dans RAD2RAD, nous avons comparé ses performances à celles de deux autres solveurs issues des bibliothèques publiques, l'un multifrontal, appelé MUMPS (MULTifrontal Massively Parallel Solver) et l'autre supernodal [DEG⁺99] appelé SuperLU.

Ces deux bibliothèques utilisent une méthode de résolution directe basée sur une factorisation $H = LU$ (où L est une matrice triangulaire inférieure et U une matrice triangulaire supérieure) et sont particulièrement adaptées à la résolution de systèmes linéaires creux de grandes dimensions. Bien que le choix de ces bibliothèques ait été influencé par l'existence de versions parallèles, nous ne nous sommes intéressés au cours de cette étude qu'à leurs

versions séquentielles. Pour une description plus détaillée de ces deux solveurs, nous pouvons nous rapporter aux références [ADLK03, DGL03].

A titre d'exemple nous avons testé la performance de ces trois solveurs sur le cas test de corps creux en compression de la section 1.3.2. Nous considérons ici le cas du collage aux maillages compatibles avec 32 noeuds de liaisons et 6 ddls par noeuds (soit un problème d'interface de dimension $p = 192$). Le tableau 1.5 représente les temps de calcul utilisés pour la résolution de chaque sous-domaine (RADIOSS) et pour la résolution du problème d'interface (RAD2RAD).

	Cholesky	SuperLU	MUMPS
PART1	177.4 s	178.1 s	184.9 s
PART2	55.4 s	54.0 s	53.2 s
RAD2RAD	215.8 s	243.5 s	307.2 s

Tableau 1.5: Comparaison des temps CPU pour différents solveurs d'interface.

Pour ce cas test réalisé sur machine mono-processeur, nous remarquons que l'algorithme de Cholesky développé dans RAD2RAD affiche les meilleures performances. Le décompte des opérations élémentaires nécessaires à la factorisation de H justifie ce résultat ($p^3/6$ opérations pour la décomposition LL^t contre $p^3/3$ pour la décomposition LU). Lorsque la matrice H est symétrique, comme c'est ici le cas, la factorisation de Cholesky semble donc être la plus économique. Toutefois, les bibliothèques SuperLU et MUMPS présentent d'autres avantages notamment en terme de parallélisation qu'il serait intéressant d'exploiter. Leur utilisation peut alors s'avérer intéressante pour les problèmes de plus grandes dimensions.

Outre le temps de calcul, un autre aspect de l'efficacité des solveurs dont nous nous ne sommes pas préoccupés ici, concerne l'occupation mémoire utilisée au cours de la résolution. Sur ce point, chacun des trois solveurs utilise son propre mode de stockage du système matriciel. Nous remarquerons simplement à ce sujet que, contrairement aux deux autres, la bibliothèque SuperLU ne dispose pas d'un mode de stockage spécifique aux matrices symétrique et que par conséquent, le stockage au format Harwell-Boeing (colonne compressé) qu'elle utilise n'est pas optimal dans notre cas.

1.3.4 Calculs couplés et parallélisme

1.3.4.1 Brefs rappels sur le calcul parallèle

Comme nous l'avons vu en introduction des méthodes de décomposition de domaine, celles-ci ont connu un essor considérable avec l'arrivée des machines parallèles. Le but n'est pas ici de faire une présentation détaillée du calcul parallèle [Fos95], mais plutôt de rappeler brièvement les différentes architectures utilisées ainsi que leurs modèles d'exécution associés. Nous entendons par modèle d'exécution, le mode de fonctionnement physique par opposition au modèle de programmation correspondant au mode de fonctionnement du langage utilisé.

Modèle d'exécution Généralement, une première distinction entre ces différents modèles porte sur le nombre de flots d'instructions traité par l'ensemble des processeurs :

simple (SIMD) ou multiple (MIMD) :

- ⇨ Machine SIMD : Une machine SIMD (Single Instruction Multiple Data) est une machine qui exécute à tout instant une seule instruction, mais qui agit en parallèle sur plusieurs données.
- ⇨ Machine MIMD : Le cas des machines MIMD (Multiple Instruction Multiple Data) est le plus intuitif. Ici, chaque processeur peut exécuter un programme différent sur des données différentes.

Pour ces deux définitions, il convient de préciser comment les processeurs accèdent aux données. C'est la distinction entre mémoire partagée et mémoire distribuée (cf. Figure 1.21). Dans une architecture parallèle à mémoire partagée, tous les processeurs accèdent à la mémoire de façon identique. Ces architectures sont généralement qualifiées de SMP (Shared Memory multi-Processor). Dans une machine à mémoire distribuée, chaque processeur n'accède directement qu'à sa propre mémoire locale. Pour accéder au reste de la mémoire il doit dialoguer avec les autres processeurs, généralement par envois de message. Ces machines à mémoire distribuée ayant généralement un plus grand nombre de processeur, on les qualifie de MPP (Massively Parallel multi-Processors)



Figure 1.21: Schémas d'architectures parallèles : (a) Mémoire partagée - (b) Mémoire distribuée.

Modèle de programmation Conjointement aux deux principaux modèles d'exécution SIMD et MIMD, on définit différents modèles de programmation parallèle, c'est-à-dire les différents modes de fonctionnement des langages. Il existe une multitude de modèles de programmation qui en principe, ne tiennent pas compte de l'architecture matérielle utilisée. Toutefois, la séparation entre modèle de programmation et modèle d'exécution n'est pas encore parfaitement claire et les deux principaux modèles de programmation que nous appellerons « parallélisme de données » et « parallélisme de tâche » héritent respectivement des modèles d'exécution SIMD et MIMD. De ces modèles généraux découlent de nombreux autres modèles parmi lesquels on peut citer ceux utilisés par RADIOSS.

Le premier modèle, appelé SMP (Shared Memory multi-Processor), est basé sur le concept de machine à mémoire partagée. Dans ce cas, tous les processeurs partagent un même espace mémoire. Ils peuvent ainsi lire et écrire en mémoire de manière indépendante et asynchrone, ce qui permet de s'affranchir du problème de la communication des données entre les tâches. Ce modèle est de ce fait généralement plus facile à mettre en oeuvre.

Le second modèle, dit de programmation par passage de message, correspond généralement au modèle d'exécution d'une machine MIMD à mémoire distribuée. Dans ce modèle, chaque processeur utilise sa propre mémoire locale. La communication des données et la

synchronisation se font de manière explicite (cf. Remarque 1) par passage de message au travers un réseau. Cela nécessite l'appel à des sous-programmes spécialisés constituant une bibliothèque de routines d'échanges de messages tel que MPI (Message Passing Interface). Parce qu'il n'est en général pas nécessaire d'utiliser des programmes différents pour chaque processeur, on exécute souvent le même code sur tous les noeuds d'une machine MIMD mais ceux-ci ne sont pas forcément synchronisés. On parle alors de modèle SPMD (Single Program Multiple Data).

Remarque 1 : Pour tirer bénéfice de ces architectures parallèles, il est nécessaire d'adapter la manière de programmer. On distingue alors deux approches de programmation : le parallélisme implicite et le parallélisme explicite. On parle de parallélisme explicite lorsque le programmeur a la maîtrise totale du parallélisme ; il spécifie explicitement la manière de partitionner et d'allouer les différentes tâches aux différents processeurs. A l'inverse, on parle de parallélisme implicite lorsque la parallélisation est laissée au bon vouloir du compilateur.

1.3.4.2 Parallélisation des calculs couplés

Dans une telle organisation de calcul couplé, il convient de distinguer deux niveaux de parallélisation. Le premier que l'on pourra qualifier de « bas niveau » portant sur chacun des process clients est propre au code de calcul RADIOSS. Cette parallélisation se retrouve d'ailleurs dans les calculs mono-domaine classiques où un unique calcul RADIOSS est exécuté. Nous remarquerons que le logiciel dispose de deux modèles de programmation parallèle précédemment évoqués, à savoir un modèle SMP et un modèle SPMD. Dans tout les cas la parallélisation et par conséquent, le partitionnement et la répartition des tâches est transparente vis-à-vis de l'utilisateur.

Nous nous préoccupons d'avantage ici du second niveau de parallélisation, que l'on qualifiera de « haut niveau », lié à la résolution simultanée de plusieurs sous-domaines. Le partitionnement issu de la découpe en sous-domaines est cette fois-ci spécifié de manière explicite par l'utilisateur. Se pose alors le problème de la répartition de charge, afin d'utiliser de manière optimale les ressources mises à disposition.

En effet, la répartition de charge joue un rôle important dès lors qu'il est nécessaire de mettre en place des synchronisations entre les différents processus. La tâche la plus lente détermine naturellement la performance globale du calcul, les tâches les plus rapides devant attendre les plus lentes. Tel est le cas de notre application où les sous-domaines mis en concurrence n'ont aucune raison d'avoir la même taille et par conséquent la même charge de calcul. Pour bien comprendre, plaçons-nous dans le cas d'un calcul multi-domaines à pas de temps unique. On suppose que tous les sous-domaines ont atteint leur équilibre à l'instant t_k^n . On comprend aisément que pour un même nombre de CPU par sous-domaine, l'état t_k^{n+1} sera plus rapidement atteint par les sous-domaines de petites tailles.

Pour les applications parallèles régulières dont on connaît à l'avance la charge propre à chaque sous-domaine, on peut procéder à un équilibrage statique via une distribution des calculs initiaux. Pour les problèmes de dynamique dont il est question ici, il est généralement difficile d'évaluer la charge de chaque sous-domaine. En effet, bien que la taille de

chaque sous-modèle soit connue a priori, il est impossible de connaître précisément comment évolue le pas de temps au cours du calcul et la délicate distribution des ressources initiales risque de ne pas conduire à une efficacité optimale.

Dés que les temps d'exécution pour chaque processus sont très disparates, le traitement concurrent des différents sous-domaines ne semble donc pas adapté. On est alors amené à découpler la stratégie multi-domaines de l'aspect parallélisme en affectant successivement les différentes tâches aux processeurs qui se libèrent. Concrètement, cela revient à traiter tour à tour les différents sous-domaines (et ce, à chaque pas de temps) mais en utilisant l'ensemble des processeurs disponibles. C'est cette stratégie que nous adopterons ici en traitant les sous-domaines de manière séquentielle, mais en laissant à RADIOSS le soin de travailler en parallèle sur chacun d'eux. Cette approche sera alors d'autant plus efficace si le nombre de sous-domaines est beaucoup plus important que le nombre de processeurs disponibles.

Enfin, nous noterons que pour être optimale, une parallélisation des calculs d'interface et donc du solveur RAD2RAD est inéluctable. Bien que cet aspect n'ait pas réellement été développé au cours de ces travaux, il a néanmoins été entrepris via l'introduction des bibliothèques MUMPS et SuperLU parallélisables.

Chapitre 2

Gestion des non-linéarités de contact

Sommaire

2.1	Étude du contact mono-domaine	55
2.1.1	Définition du contact	55
2.1.2	Formulation du problème de contact	58
2.1.3	Traitement numérique du contact dans les codes explicites	64
2.2	Développement d'une méthode de contact inter-domaine	71
2.2.1	Bref état de l'art sur le contact multi-domaines	71
2.2.2	Méthode de la <i>peau artificielle</i>	76
2.2.3	Extension aux liaisons permanentes	80
2.3	Collage temporel dans la méthode de couplage dynamique	83
2.3.1	Communications à pas de temps unique	83
2.3.2	Gestion des pas de temps asynchrones	84
2.3.3	Implémentation à RAD2RAD	88
2.4	Exemples numériques	92
2.4.1	Chute d'une poutre	92
2.4.2	Impact d'un corps creux	94
2.4.3	Simulation d'un choc piéton	97

AU cours de ce second chapitre, on se propose d'introduire la modélisation du phénomène du contact unilatéral et la manière de le traiter dans le cadre de notre approche multi-domaines. De part son comportement globalement non-linéaire, le contact conduit souvent à de sérieuses difficultés de calcul. Ce phénomène fréquemment rencontré en analyse des structures, particulièrement pour les problèmes d'assemblage, touche de nombreux secteurs d'applications. Il est omniprésent dans le domaine du crash automobile où un grand nombre de pièces sont susceptibles de s'impacter au cours du choc (pare-choc, radiateur, moteur...).

Avant de poursuivre notre développement, il convient de donner d'ores et déjà quelques précisions terminologiques nécessaire à la bonne compréhension de la problématique. Sur le plan de la modélisation, l'assemblage peut être considéré soit comme un ensemble de *pièces* physiquement reliées entre elles par des éléments de conception (vis, point de soudure, cordon de colle...), soit comme un ensemble de *sous-domaines* issus de la décomposition dont les interfaces purement numériques sont des liaisons parfaites dénuées de

tout sens physique. Chaque *sous-domaine* constitue donc à lui seul un modèle de calcul, généralement composé de plusieurs *pièces*.

Bien que le logiciel RADIOSS dispose d'algorithmes de contact performants, ceux-ci ne concernent que les interactions internes (entre pièces d'un même modèle) et en aucun cas les interactions entre les différents sous-domaines. Pour illustrer cette problématique revenons brièvement sur l'exemple de compression de corps creux présenté au chapitre précédent. En poursuivant la simulation de manière à ce que le flambage atteigne la partie haute du tube, on constate que le calcul multi-domaines conduit à une interpénétration des lobes de flambement à proximité de l'interface (cf. Figure 2.1). Autrement dit les deux sous-domaines calculés indépendamment ne se « voient » pas. Ceci n'est pas le cas du calcul mono-domaine, où l'interface de contact interne au modèle est gérée sans difficulté par le code RADIOSS ; on parle alors d'auto-contact.

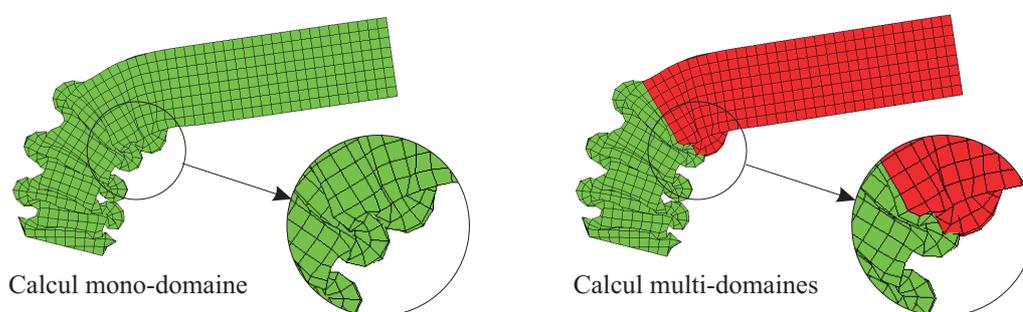


Figure 2.1: Problématique de gestion du contact pour les calculs multi-domaines.

Le but de ce chapitre est donc de proposer une méthode de gestion du contact unilatéral (avec ou sans frottement) entre sous-domaines pour les calculs de dynamique explicite. Cette approche s'inscrita évidemment dans le formalisme multi-échelles, où chaque sous-structure évolue avec son propre pas de temps de stabilité. L'objectif n'est toutefois pas de redévelopper un algorithme de contact dans le code de couplage RAD2RAD, mais plutôt d'adopter une stratégie de calcul originale utilisant les algorithmes de contact existants sous RADIOSS.

Dans une première partie nous reviendrons sur les grands principes du contact dans le cadre classique des calculs mono-domaines. Nous y aborderons les conditions de contact et détaillerons comment celles-ci sont intégrées au problème d'équilibre. Nous nous baserons pour cela sur une méthode de pénalité, qui nous le verrons, offre de nombreux avantages en dynamique explicite. Le passage de la nouvelle formulation faible à la formulation discrète nous mènera à étudier la manière dont le problème est résolu dans un code de calcul éléments finis. La seconde partie sera consacrée à notre stratégie de contact inter-domaines. Après avoir établi un bref état des lieux des méthodes existantes, nous présenterons la méthode dite de la « peau artificielle » que nous avons développée. Nous poursuivrons ensuite en détaillant la façon dont cette méthode est intégrée à notre code de couplage dynamique ; nous distinguerons alors le cas mono-échelle (pas de temps unique) du cas multi-échelles en temps (pas de temps asynchrones). Enfin, nous terminerons ce chapitre en illustrant la méthode à travers différents exemples de simulations. Une application véhicule permettra alors de conclure quant à la validité et l'efficacité de notre approche multi-domaines.

2.1 Étude du contact mono-domaine

Bien que de manière générale, le contact puisse s'appliquer à un nombre quelconque de corps, nous restreindrons par soucis de simplification notre description au cas du contact entre deux domaines notés Ω_1 et Ω_2 . L'extension au cas général sera toutefois immédiate en considérant le problème comme une série de contacts entre deux corps.

2.1.1 Définition du contact

2.1.1.1 Notion d'interface de contact

On définit l'interface de contact Γ_c comme étant l'intersection des deux frontières Γ_1 et Γ_2 correspondantes aux domaines respectives Ω_1 et Ω_2 (cf. Figure 2.2) :

$$\Gamma_c = \Gamma_1 \cap \Gamma_2 \quad (2.1.1)$$

L'interface de contact se compose donc de deux frontières physiques qui d'un point de vue continu sont théoriquement confondues et donc désignées par une seule et même surface :

$$\Gamma_c = \Gamma_{c1} = \Gamma_{c2} \quad (2.1.2)$$

Toutefois, après discrétisation, les surfaces en contact ne sont généralement plus rigoureusement les mêmes, sauf dans le rare cas où les maillages des deux corps sont coïncidents à l'interface. De ce fait, on distingue généralement ces deux surfaces théoriquement indiscernables, par les désignations *maître* et *esclave*. Cette technique fréquemment utilisée dans les algorithmes de contact consiste à écrire les conditions de contact et de frottement sur la surface d'un seul des corps (le corps *esclave*). L'interface de contact Γ_c réfère alors généralement à la surface *maître*. Par la suite nous choisirons le corps Ω_1 comme étant le *maître* et le corps Ω_2 comme étant l'*esclave* de manière à ce que :

$$\tilde{\Gamma}_c = \tilde{\Gamma}_{c1} \neq \tilde{\Gamma}_{c2} \quad (2.1.3)$$

Enfin, il convient ici de bien distinguer ce que nous appellerons interface de contact Γ_c de l'interface de liaison Γ_l . L'interface de liaison que nous avons traitée dans le chapitre précédent constitue une liaison parfaite et **permanente** entre les sous-domaines. L'interface de contact est quant à elle une liaison **potentielle** rendue active lorsque deux

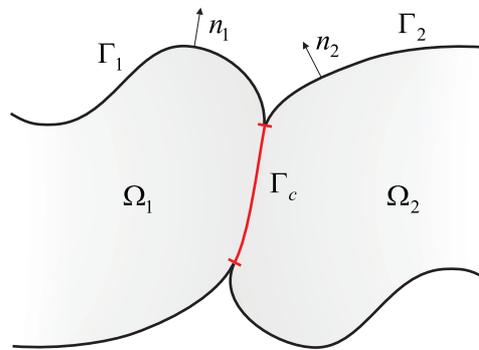


Figure 2.2: Définition de l'interface de contact.

corps entrent en contact. Nous soulignerons ici le fait que nous parlons de *corps* en contact et non de *sous-domaines* puisque que contrairement à l'interface de liaison, l'interface de contact est indépendante du calcul multi-domaines. Néanmoins, dans le cadre précis de ces travaux nous nous intéresserons au problème du contact entre domaines (cf. §2.2). Nous pourrions donc par la suite utiliser indifféremment les termes *corps* et *sous-domaines*.

2.1.1.2 Condition de contacts

Les équations régissant le comportement dynamique des corps en contact sont identiques à celles définies dans le premier chapitre (1.1.1-1.1.5). La seule spécificité propre au contact, se traduit par l'ajout de contraintes cinématiques sur l'interface Γ_c . Ce sont ces contraintes que l'on propose de présenter ici.

Condition d'impénétrabilité La principale condition, appelée condition d'impénétrabilité, est au coeur même du problème de contact. Elle exprime le fait que les corps ne puissent se chevaucher l'un l'autre :

$$\Omega_1 \cap \Omega_2 = 0 \quad (2.1.4)$$

De manière générale, il est impossible d'anticiper quels points des deux corps vont entrer en contact. C'est pourquoi, cette condition qui, pour les problèmes à grand déplacement est fortement non-linéaire, ne peut s'exprimer sous forme d'une équation algébrique ou différentielle en déplacement. Elle s'exprime alors généralement de manière simplifiée et incrémentale à chaque cycle de calcul.

Une manière simple de dériver l'équation (2.1.4) consiste à exprimer le gap (ou fonction d'interpénétration normal) g entre les deux corps. La pénétration d'un point P de Γ_2 à l'intérieur de Ω_1 est définie par sa distance minimale par rapport à chaque point de Γ_1 . Dans une approche *maître-esclave* seule la pénétration du corps *esclave* Ω_2 dans le corps *maître* Ω_1 étant mesurée, la condition d'impénétrabilité peut s'écrire :

$$g(x_2, t) = \min_{x_1 \in \Gamma_1} \|x_2(t) - x_1(t)\| \text{sign}(n_1 \cdot (x_1(t) - x_2(t))) \leq 0 \quad \forall x_2 \in \Gamma_2 \quad (2.1.5)$$

où n_1 , désigne la normale unitaire sortante à la surface Γ_1 . Nous remarquerons que la fonction gap correspond à une distance signée qu'il convient de rendre négative ou nulle pour assurer la non-interpénétration des corps. Naturellement, on peut montrer que le point de Ω_1 le plus proche de P correspond à son projeté orthogonal sur la surface Γ_1 . Si la projection est positive, le point P est à l'intérieur de Ω_1 , sinon P est à l'extérieur, il n'y a donc pas d'interpénétration.

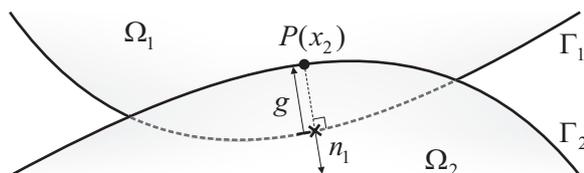


Figure 2.3: Fonction d'interpénétration de Ω_2 dans Ω_1 .

Condition d'équilibre A cette condition d'impénétrabilité se joint la condition d'équilibre à l'interface. L'interface étant supposée sans masse, l'équilibre exprime l'égalité des efforts de réactions de part et d'autre du contact :

$$\tau_1 + \tau_2 = 0 \quad \text{avec} \quad \tau_k = \sigma_k n_k \quad \forall k \in \{1, 2\} \quad (2.1.6)$$

Sur chaque corps, la force de réaction τ_k peut se décomposer en une composante normale appelée vecteur contrainte normale, τ_N et une partie tangentielle τ_T . Les vecteurs contraintes normaux s'écrivent :

$$\tau_{1N} = \tau_1 \cdot n_1 \quad \tau_{2N} = \tau_2 \cdot n_1 \quad (2.1.7)$$

Nous remarquerons que ces composantes réfèrent toutes deux à la normale du côté *maître*. La composante normale de l'équation d'équilibre est donc obtenue en prenant le produit scalaire de (2.1.6) par n_1 :

$$\tau_{1N} + \tau_{2N} = 0 \quad (2.1.8)$$

Si l'on suppose qu'il n'y a pas de frottement entre les surfaces en contact, on peut assimiler l'effort de réaction à un effort de compression pure :

$$\tau_N \equiv \tau_{1N}(x, t) = -\tau_{2N}(x, t) \leq 0 \quad (2.1.9)$$

Nous remarquerons que cette condition est non symétrique par rapport aux corps Ω_1 et Ω_2 et que le choix inversé des surfaces *maître* et *esclave* peut changer les résultats.

Loi de Signorini Les conditions d'impénétrabilité (2.1.5) et d'équilibre (2.1.9) peuvent être regroupées pour former ce que l'on appelle communément les conditions de *Signorini*. En résumé, on peut donc écrire :

$$g \leq 0 \quad (2.1.10)$$

$$\tau_N \leq 0 \quad (2.1.11)$$

$$g \cdot \tau_N = 0 \quad (2.1.12)$$

La dernière relation (2.1.12) découlant des deux autres est dite équation de complémentarité. Elle exprime le fait que les forces de contact normales ne travaillent pas. Si les deux corps sont en contact, le gap s'annule ($g = 0$) alors que si le contact cesse ($g < 0$) ce sont les efforts normaux qui s'annulent.

2.1.1.3 Loi de frottement

Jusqu'ici, nous avons uniquement considéré la composante normale du vecteur contrainte à l'interface pour définir les conditions de contact. Regardons à présent comment exprimer la composante tangentielle τ_T encore appelée scission de frottement, à partir de la loi de frottement. Parmi les nombreux modèles de frottement existants, le plus utilisé dans les codes de calcul, est sans conteste celui de *Coulomb*. Nous nous intéresserons donc à cette loi qui est notamment utilisée par le logiciel RADIOSS.

En supposant que les corps Ω_1 et Ω_2 sont en contact en x , la loi de *Coulomb* donne une relation entre la vitesse de glissement $\dot{u}_g = \dot{u}_{1T} - \dot{u}_{2T}$ et la composante tangentielle du vecteurs contrainte τ_T . Cette relation définit deux états possibles :

$$(i) \text{ si } \|\tau_T(x, t)\| < -\mu\tau_N(x, t) \quad \text{alors } \dot{u}_g = 0 \quad (2.1.13)$$

$$(ii) \text{ si } \|\tau_T(x, t)\| = -\mu\tau_N(x, t) \quad \text{alors } \exists \kappa > 0 \text{ tel que } \dot{u}_g = \kappa\tau_T(x, t) \quad (2.1.14)$$

La condition (i) est connue sous le nom de condition d'adhérence : lorsque l'effort tangentiel est inférieur à une valeur critique, aucun mouvement relatif tangentiel n'est permis. A l'inverse, la condition (ii) correspond à la condition de glissement : dans ce cas le mouvement tangentiel est possible et l'effort tangentiel de frottement agit dans un sens opposé à la vitesse tangentielle relative.

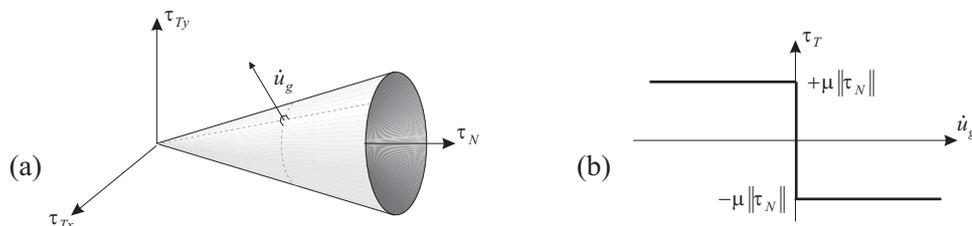


Figure 2.4: Loi de *Coulomb* : (a) Cône de frottement - (b) Loi de *Coulomb*.

Remarque 1 : Bien que le modèle de *Coulomb* s'affiche comme le modèle de frottement le plus répandu, il est souvent critiqué à cause de son manque de sens physique et des problèmes mathématiques qu'il pose. En effet, on voit aisément sur la figure 2.4b que le vecteur contrainte tangentiel est une fonction discontinue de la vitesse de glissement. Cette discontinuité pose de sérieux problèmes numériques. Pour lever ce problème on procède généralement à une régularisation, ou lissage de la loi autour de la singularité [BCF⁺94]. Nous reviendrons sur cette régularisation dans le paragraphe 2.1.2.4.

2.1.2 Formulation du problème de contact

Regardons à présent comment les lois de contacts, que nous venons de définir, se répercutent sur l'équilibre dynamique de la structure. Par soucis de simplification, nous nous concentrerons ici sur la condition d'impénétrabilité en négligeant le phénomène de frottement à l'interface. Après avoir présenté les différentes techniques permettant de prendre en compte cette contrainte adjointe au problème, nous reviendrons sur la formulation faible introduite au paragraphe 1.1.1.3, puis sur sa discrétisation. Nous nous focaliserons alors sur la méthode de pénalité qui, de par sa simplicité de mise en oeuvre, s'avère être fréquemment utilisée dans les codes de calculs éléments finis.

2.1.2.1 Prise en compte des contraintes

Au même titre que les interfaces de liaisons entre sous-domaines, les interfaces de contact peuvent être assimilées à une condition limite qu'il convient d'adjoindre à la formulation faible (1.1.19) sous forme de contrainte (au sens obligation) à respecter. Nous parlerons ici de contrainte au sens général $\mathcal{G}(u) = 0$, bien que le cas particulier du contact sera considéré pour l'écriture de la formulation faible dans le paragraphe suivant. Le but est alors de résoudre le problème de minimisation de la fonction énergie potentielle total \mathcal{W} sous contrainte :

$$\begin{cases} \delta\mathcal{W}(u, \delta u) = 0 & \forall \delta u \in \mathcal{U}_0 \\ \mathcal{G}(u) = 0 \end{cases} \quad (2.1.15)$$

où $u = \{u_i\}$ est ici un vecteur de dimension n_{ddl} désignant le champ de déplacement discret. On se propose dans le paragraphe suivant de faire un bref rappel des principales

techniques permettant de modifier le problème (2.1.15) de minimisation de la fonctionnelle sous contrainte, en un problème de stationnarité d'une fonctionnelle sans contrainte [BLM00].

Méthode des multiplicateurs de Lagrange Dans cette méthode, la contrainte pondérée par les multiplicateurs de Lagrange est ajoutée à la fonction objective, c'est-à-dire à la fonction \mathcal{W} à minimiser. La solution de (2.1.15) revient à trouver les points stationnaires de \mathcal{W}_L :

$$\mathcal{W}_L(u, \lambda) = \mathcal{W}(u) + \lambda^t \mathcal{G}(u) \quad (2.1.16)$$

où $\lambda = \{\lambda_I\}$ désigne le vecteur des multiplicateurs de Lagrange de dimension p . Cette nouvelle fonctionnelle, résultant de la dualisation de la contrainte, présente la particularité d'être convexe pour les variables primales et concave pour les variables duales. Autrement dit, le point d'équilibre est un « point selle » correspondant à un minimum par rapport à u et un maximum par rapport à λ . La dérivée par rapport à u et λ s'annule au point stationnaire :

$$\frac{\partial \mathcal{W}_L}{\partial u_i} = \frac{\partial \mathcal{W}}{\partial u_i} + \lambda_k \frac{\partial \mathcal{G}_k}{\partial u_i} = 0, \quad i = 1 \text{ à } n_{ndd} \quad (2.1.17)$$

$$\frac{\partial \mathcal{W}_L}{\partial \lambda_j} = \mathcal{G} = 0, \quad j = 1 \text{ à } p \quad (2.1.18)$$

Le système précédent est un système de dimension $n_{ddl} + n_p$. En rappelant d'après (1.1.19) que l'on a :

$$\frac{\partial \mathcal{W}}{\partial u_i} = \frac{\partial \mathcal{W}_{int}}{\partial u_i} + \frac{\partial \mathcal{W}_{ext}}{\partial u_i} + \frac{\partial \mathcal{W}_{kin}}{\partial u_i} \quad (2.1.19)$$

$$= f_{int,i} - f_{ext,i} + M_{ii} \ddot{u}_i \quad (2.1.20)$$

On peut réécrire le système (2.1.17 - 2.1.18) sous la forme matricielle suivante :

$$f_{int} - f_{ext} + M \ddot{u} + \lambda \frac{\partial \mathcal{G}}{\partial u} = 0 \quad (2.1.21)$$

$$\mathcal{G} = 0 \quad (2.1.22)$$

Comme nous pouvons le voir dans l'équation (2.1.21), la contrainte introduit une force additionnelle $\lambda \cdot \partial \mathcal{G} / \partial u$ qui est une combinaison linéaire des multiplicateurs de Lagrange.

Un des principaux inconvénients de cette méthode est que l'introduction de multiplicateur de Lagrange a pour effet d'augmenter le nombre d'inconnues du problème, parfois de manière substantielle. D'autre part le système d'équations résultant n'est pas nécessairement défini positif.

Remarque 1 : Revenons quelques instants sur la justification de l'écriture (1.2.14) dans la méthode de *Schur duale* décrite au paragraphe 1.2.1.3. On rappelle que le problème considéré est un problème statique d'élasticité linéaire et que la contrainte concerne la condition de continuité des déplacements à l'interface. En supposant pour simplifier un collage de maillage compatible cette contrainte s'écrit :

$$\mathcal{G}(u) = C_1 u_1 + C_2 u_2 = 0 \quad \text{sur} \quad \Gamma_I \quad (2.1.23)$$

où Γ_l désigne ici l'interface de liaison entre les sous-domaines. D'après (2.1.16), cela conduit à chercher les vecteurs (u_1, u_2, λ) rendant stationnaire la fonctionnelle :

$$\mathcal{W}_1(u_1) + \mathcal{W}_2(u_2) + \int_{\Gamma_l} \lambda^t (C_1 u_1 + C_2 u_2) d\Gamma \quad (2.1.24)$$

Cette solution doit alors vérifier :

$$0 = \delta \mathcal{W}_1(\delta u_1, u_1) + \delta \mathcal{W}_2(\delta u_2, u_2) + \delta \left(\int_{\Gamma_l} \lambda^t (C_1 u_1 + C_2 u_2) d\Gamma \right) \quad (2.1.25)$$

$$= \delta u_1 (K_{11} u_1 - f_1) + \delta u_2 (K_{22} u_2 - f_2) + \delta (\lambda^t (C_1 u_1 + C_2 u_2)) \quad (2.1.26)$$

$$= \delta u_1 (K_{11} u_1 - f_1 + \lambda^t C_1) + \delta u_2 (K_{22} u_2 - f_2 + \lambda^t C_2) + \delta \lambda^t (C_1 u_1 + C_2 u_2) \quad (2.1.27)$$

De part le caractère arbitraire des fonctions test δu_1 , δu_2 et $\delta \lambda$, l'équation précédente peut se décliner sous la forme du système (1.2.14), à savoir :

$$\begin{cases} K_{11} u_1 + \lambda^t C_1 = f_1 \\ K_{22} u_2 + \lambda^t C_2 = f_2 \\ C_1 u_1 + C_2 u_2 = 0 \end{cases} \quad (2.1.28)$$

Méthode de pénalité Dans la méthode de pénalité, la contrainte est imposée en ajoutant le terme $\mathcal{G}^t \mathcal{G}$ multiplié par un coefficient β , appelé paramètre de pénalité, à la fonctionnelle \mathcal{W} . Le nouveau potentiel \mathcal{W}_P à minimiser s'écrit :

$$\mathcal{W}_P(u) = \mathcal{W}(u) + \frac{\beta}{2} \mathcal{G}^t(u) \mathcal{G}(u) \quad (2.1.29)$$

Le paramètre β est choisi suffisamment grand pour s'assurer que le minimum de \mathcal{W}_P ne peut être atteint sans que la contrainte soit satisfaite. La condition de stationnarité sur \mathcal{W}_P revient à chercher u vérifiant :

$$\frac{\partial \mathcal{W}_P}{\partial u_i} = \frac{\partial \mathcal{W}}{\partial u_i} + \beta \mathcal{G}_k \frac{\partial \mathcal{G}_k}{\partial u_i} = 0, \quad i = 1 \text{ à } n_{ndd} \quad (2.1.30)$$

Que l'on peut écrire sous forme matricielle :

$$f_{int} - f_{ext} + M \ddot{u} + \beta \mathcal{G}^T \frac{\partial \mathcal{G}}{\partial u} = 0 \quad (2.1.31)$$

Nous insisterons sur le fait que β n'est pas une inconnue et que par conséquent cette technique présente l'avantage de ne pas augmenter la taille du système à résoudre. Bien que dans cette approche la contrainte ne soit jamais exactement respectée, elle est d'autant mieux vérifiée que le paramètre de pénalité est grand. Toutefois, le choix de β demeure une grande difficulté et reste spécifique à chaque problème puisque l'utilisation d'un paramètre très grand altère de manière significative le conditionnement des équations.

Méthode du Lagrangien augmenté La méthode du Lagrangien augmenté peut être vue comme une combinaison des deux méthodes précédentes et tente de palier à leurs inconvénients respectifs. Pour un paramètre de pénalité β donné, on détermine les déplacements et les multiplicateurs de Lagrange rendant stationnaire la fonctionnelle \mathcal{W}_{AL} :

$$\mathcal{W}_{AL}(u, \lambda) = \mathcal{W}(u) + \lambda^t \mathcal{G}(u) + \frac{\beta}{2} \mathcal{G}^t(u) \mathcal{G}(u) \quad (2.1.32)$$

La stationnarité de \mathcal{W}_{AL} peut être déterminée en annulant ces dérivées partielles par rapport à u et λ . Le système (2.1.15) peut alors s'écrire :

$$\frac{\partial \mathcal{W}_{AL}}{\partial u_i} = \frac{\partial \mathcal{W}}{\partial u_i} + \lambda_k \frac{\partial \mathcal{G}_k}{\partial u_i} + \beta \mathcal{G}_k \frac{\partial \mathcal{G}_k}{\partial u_i} = 0, \quad i = 1 \text{ à } n_{ndd} \quad (2.1.33)$$

$$\frac{\partial \mathcal{W}_{AL}}{\partial \lambda_k} = \mathcal{G}_k = 0, \quad k = 1 \text{ à } p \quad (2.1.34)$$

Comme pour la méthode des multiplicateurs de Lagrange ce système de $n_{ddl} + p$ équations peut se réécrire sous forme matricielle :

$$f_{int} - f_{ext} + M\ddot{u} + \lambda \frac{\partial \mathcal{G}}{\partial u} + \beta \mathcal{G} \frac{\partial \mathcal{G}}{\partial u} = 0 \quad (2.1.35)$$

$$\mathcal{G} = 0 \quad (2.1.36)$$

Nous remarquerons que si l'on pose $\lambda = 0$ on retrouve la méthode de pénalité, et si l'on pose $\beta = 0$ on retrouve la méthode des multiplicateurs de Lagrange. Grâce à l'introduction des multiplicateurs de Lagrange, le paramètre β n'est plus nécessairement aussi grand que dans la méthode de pénalité ; ceci améliore le conditionnement et rend le système matriciel défini positif.

Méthode du Lagrangien perturbé En Lagrangien perturbé, le produit d'une petite constante avec la somme des carrés des multiplicateurs de Lagrange est ajouté à la formulation classique. Il s'agit donc d'une perturbation de la méthode précédente où le paramètre de pénalisation est appliqué à λ et non plus à \mathcal{G} . Le potentiel pour le Lagrangien perturbé s'écrit donc :

$$\mathcal{W}_{PL}(u, \lambda) = \mathcal{W}(u) + \lambda^t \mathcal{G}(u) - \frac{\varepsilon}{2} \lambda^t \lambda \quad (2.1.37)$$

Comme précédemment, la solution rendant stationnaire cette fonctionnelle vérifie l'annulation de ces dérivées partielles par rapport à u et λ :

$$\frac{\partial \mathcal{W}_{PL}}{\partial u_i} = \frac{\partial \mathcal{W}}{\partial u_i} + \lambda_k \frac{\partial \mathcal{G}_k}{\partial u_i} = 0, \quad i = 1 \text{ à } n_{ndd} \quad (2.1.38)$$

$$\frac{\partial \mathcal{W}_{PL}}{\partial \lambda_k} = \mathcal{G}_k - \varepsilon \lambda_k = 0, \quad k = 1 \text{ à } p \quad (2.1.39)$$

En réécrivant ces équations sous forme matricielle, on obtient :

$$f_{int} - f_{ext} + M\ddot{u} + \lambda \frac{\partial \mathcal{G}}{\partial u} = 0 \quad (2.1.40)$$

$$\mathcal{G}(u) - \varepsilon \lambda = 0 \quad (2.1.41)$$

Nous remarquerons que la méthode du Lagrangien perturbé est essentiellement d'intérêt théorique. En éliminant λ dans les deux relations précédentes, on montre aisément que ceci conduit à une formulation identique à celle de la méthode de pénalité.

2.1.2.2 Formulation faible du problème de contact par pénalité

Revenons à présent sur la formulation faible de notre problème. Tout d'abord, l'espace des solutions $u(X, t)$ cinématiquement admissible est similaire à celui du problème classique

à un seul corps (1.1.12), sauf qu'ici les déplacements sont approximés séparément sur chacun des deux corps Ω_1 et Ω_2 . Le champ de déplacement n'est pas nécessairement continu à l'interface entre les deux corps.

$$\mathcal{U} = \{u(X, t) | u(X, t) \in C^0(\Omega_1), u(X, t) \in C^0(\Omega_2), u = \bar{u} \text{ sur } \Gamma_u\} \quad (2.1.42)$$

L'espace des fonctions test \mathcal{U}_0 est identique à \mathcal{U} sauf que les fonctions tests s'annulent sur la frontière là où est imposée la condition limite en déplacement :

$$\mathcal{U}_0 = \{\delta u(X) | \delta u(X) \in C^0(\Omega_1), \delta u(X) \in C^0(\Omega_2), \delta u = 0 \text{ sur } \Gamma_u\} \quad (2.1.43)$$

Dans la méthode de pénalité, la condition d'impénétrabilité (2.1.5) est imposée par un effort normal pénalisant le long de la surface de contact Γ_c . Conformément à (2.1.29), on peut alors réécrire la formulation faible du problème de contact sous la forme :

$$\delta \mathcal{W}_P(u, \delta u) = \delta \mathcal{W}(u, \delta u) + \delta \mathcal{G}(u, \delta u) = 0, \quad \forall \delta u \in \mathcal{U}_0 \quad (2.1.44)$$

Avec,

$$\delta \mathcal{G}(u, \delta u) = \int_{\Gamma_c} \beta g \frac{\partial g}{\partial u} \mathcal{H}(g(u)) \delta u \, d\Gamma \quad (2.1.45)$$

où β est le paramètre de pénalité. Nous remarquerons que cette formulation n'est pas une inégalité comme l'est la condition d'impénétrabilité (2.1.5). La nature discontinue du problème est introduite par la fonction Heaviside \mathcal{H} :

$$\mathcal{H}(a) = \begin{cases} 1 & \text{si } a > 0 \\ 0 & \text{si } a < 0 \end{cases} \quad (2.1.46)$$

Le caractère signé du problème est pris en compte par cette fonction dans l'intégrale sur Γ_c . A noter que l'intégrale de surface sur Γ_c comprend deux intégrales de surfaces sur Γ_{c1} et Γ_{c2} .

Contrairement à la méthode des multiplicateurs de Lagrange, la méthode de pénalité n'impose pas de manière forte la continuité des déplacements à l'interface. En effet, la formulation faible (2.1.44) n'implique pas la condition d'impénétrabilité, qui n'est alors vérifiée qu'approximativement. On montre par contre dans [BLM00] qu'elle implique que les vecteurs contraintes normaux soient compressifs et vérifient l'équilibre d'interface :

$$\tau_{N1} = -\tau_{N2} = -p = -\bar{p} \mathcal{H}(g) \quad \text{avec} \quad \bar{p} = \beta g \quad (2.1.47)$$

où p est défini comme étant la pression d'interface.

2.1.2.3 Formulation discrète

Développons à présent l'équation discrète du mouvement, en rappelant l'approximation faite sur les différents champs cinématiques et notamment sur le champ de déplacement :

$$u_i(X, t) = \sum_{I=1}^{n_{\text{noeuds}}} N_I(X) u_{iI}(t) \quad (2.1.48)$$

où u_{iI} désigne la composante du déplacement suivant la direction i au noeud I . On peut alors exprimer la fonction d'interpénétration à partir des déplacements nodaux par :

$$g = \sum_{I=1}^{nb_{noeuds}} \sum_{i=1}^3 \Phi_{iI}(X) u_{iI}(t) \quad \text{avec} \quad \Phi_{iI}(X) = \begin{cases} N_I(X) n_{1i} & \text{si } I \in \tilde{\Gamma}_1 \\ N_I(X) n_{2i} & \text{si } I \in \tilde{\Gamma}_2 \end{cases} \quad (2.1.49)$$

En injectant cette approximation dans la formulation (2.1.44), on obtient :

$$0 = \delta \mathcal{W}_P(u) = \delta u_{iI} \frac{\partial \mathcal{W}(u)}{\partial u_{iI}} + \delta u_{iI} \int_{\Gamma_c} \beta g \frac{\partial g}{\partial u_{iI}} \mathcal{H}(g(u)) d\Gamma, \quad \forall \delta u_{iI} \notin \tilde{\Gamma}_u \quad (2.1.50)$$

$$= \delta u_{iI} \frac{\partial \mathcal{W}(u)}{\partial u_{iI}} + \delta u_{iI} \int_{\Gamma_c} \Phi^t p d\Gamma, \quad \forall \delta u_{iI} \notin \tilde{\Gamma}_u \quad (2.1.51)$$

En rappelant que \mathcal{W} correspond à la somme des travaux des forces internes, des forces externes et des forces d'inertie, (2.1.51) peut se réécrire :

$$\delta u_{iI} (f_{int,iI} - f_{ext,iI} + M_{IJ} \ddot{u}_{iJ} + f_{c,iI}) = 0, \quad \forall \delta u_{iI} \notin \tilde{\Gamma}_u \quad (2.1.52)$$

où f_c représente les forces de contact dont les composantes sont données par :

$$f_{c,iI} = \int_{\Gamma_c} \Phi_{iI} p d\Gamma \quad (2.1.53)$$

Nous remarquerons que dans une approche *maître-esclave* classique, la fonction de pénétration $g(x, t)$ n'est calculée que sur la face *maître* ($\tilde{\Gamma}_c$ étant dans ce cas confondu avec $\tilde{\Gamma}_1$). Par ailleurs, l'écriture (2.1.52) étant valable pour tout δu_{iI} excepté sur le bord discret $\tilde{\Gamma}_u$, là où le déplacement est imposé, l'équation semi-discrète devient :

$$M_{IJ} \ddot{u}_{iJ} + f_{int,iI} - f_{ext,iI} + f_{c,iI} = 0 \quad \text{sur} \quad \tilde{\Omega} \setminus \tilde{\Gamma}_u \quad (2.1.54)$$

Que l'on pourra écrire sous forme matricielle :

$$[M] \ddot{u} + \{f_{int}\} - \{f_{ext}\} + \{f_c\} = 0 \quad \text{avec} \quad \{f_c\} = \int_{\Gamma_c} [\Phi]^t \{p\} d\Gamma \quad (2.1.55)$$

La contrainte est donc imposée par le vecteur d'effort pénalisant $\{f_c\}$; celui-ci introduit une source de non-linéarité supplémentaire par l'intermédiaire de la fonction Heaviside. Comme pour la formulation semi-discrète sans contact, nous abandonnerons par la suite les notations $\{\cdot\}$ et $[\cdot]$ relatives aux grandeurs vectorielles et matricielles.

2.1.2.4 Remarques sur la régularisation des lois de contact

Dans de nombreux cas, les équations modélisant un phénomène physique sont difficilement résolubles à cause de la discontinuité ou de la singularité de leur solution. Tel est le cas du contact, où les discontinuités introduites par la condition d'impénétrabilité et éventuellement par la loi de frottement compliquent généralement l'intégration des équations gouvernantes, et influencent les performances des algorithmes de résolution. La régularisation apparaît donc comme un artefact de calcul modifiant le modèle initial afin de rendre la solution plus lisse, ou plus régulière.

Concernant la condition d'impénétrabilité, on observe qu'avec une approche de type multiplicateurs de Lagrange, les vitesses deviennent discontinues à l'instant où le contact se produit. Ces discontinuités se propagent dans la structure et conduisent à une solution considérablement bruitée. En revanche, la méthode de pénalité sur laquelle nous nous sommes arrêtés s'apparente à une procédure régularisante vis-à-vis des lois de *Signorini* : elle lisse la discontinuité de vitesse et préserve la conservation de la quantité de mouvement en allongeant le temps de l'impact. Cela se fait néanmoins au dépend d'un affaiblissement de la condition d'impénétrabilité (2.1.5), en autorisant une légère interpénétration des corps en contact (cf. Figure 2.5a).

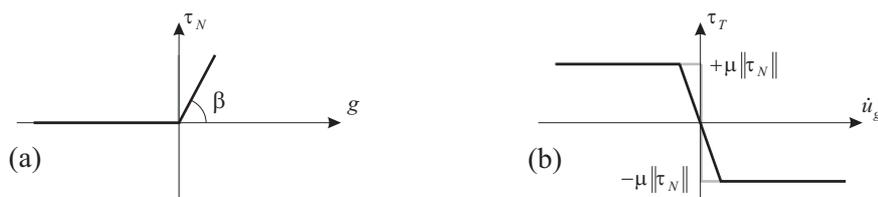


Figure 2.5: Régularisation des lois de contact : (a) Loi de *Signorini* - (b) Loi de *Coulomb*.

De la même façon pour le frottement, la loi de *Coulomb* définit la force de frottement tangentielle comme une fonction discontinue du temps (cf. Figure 2.4b). Elle peut être remplacée par un modèle plus lisse et donc plus correct sur le plan mathématique. Le simple fait d'utiliser une approximation linéaire de l'effort tangentiel au voisinage de zéro suffit à rendre la loi régulière (cf. Figure 2.5b).

Remarque 2 : Dans le cas du frottement, la régularisation peut s'interpréter physiquement en considérant par exemple le comportement élastique des aspérités sur la zone de contact. La faible interpénétration qu'autorise la méthode de pénalité peut aussi s'expliquer par la compression de ces mêmes aspérités. Toutefois, toutes les procédures régularisantes ne reposent pas sur un principe physique, et leur caractère *ad-hoc* est souvent controversé.

2.1.3 Traitement numérique du contact dans les codes explicites

Jusqu'ici nous nous étions intéressés à la formulation du contact en décrivant les différentes équations qui la régissent (de la prise en compte des conditions de contact dans la formulation faible à la formulation discrète de l'équilibre dynamique). Le but est maintenant de s'appuyer sur la formulation discrète (2.1.55) pour aborder les différents aspects de l'algorithme de contact tel qu'il se présente dans un code de calcul éléments finis. Nous nous inspirerons notamment de la méthode de pénalité développée dans RADIOSS, sur laquelle sera basée notre stratégie de contact inter-domaine.

Avant de poursuivre nous rappellerons les trois principales étapes que comprend généralement l'algorithme de contact par pénalité :

1. Reconnaissance géométrique : trouver pour chaque point, les points du corps opposé susceptibles d'entrer en contact.
2. Détection du contact : si le contact a effectivement lieu, on détermine la pénétration des surfaces en contact.

3. Calcul de l'état de contact acceptable : ajout d'une force de contact afin de réduire la pénétration.

Nous noterons qu'en dynamique cet algorithme s'inscrit dans la boucle de temps. Dans le cadre de nos travaux détaillés par la suite, les étapes de reconnaissance géométrique et de détection du contact seront laissées au soin de RADIOSS. Nous ne nous attarderons donc pas sur les méthodes de tris et de recherches associées à ces deux premières étapes [Rad05] mais nous nous concentrerons plutôt sur la manière dont le code gère concrètement la réduction de la pénétration.

2.1.3.1 Modélisation de l'interface de contact

Après la discrétisation éléments finis de la structure, les deux surfaces maillées entre lesquelles vont s'écrire les relations de contact peuvent être appariées de différentes manières. Indépendamment de la méthode utilisée (par pénalité ou par multiplicateur de Lagrange), on distingue l'appariement « noeud-noeud » de l'appariement « noeud-surface ». Pour le premier cas, apparenté à la formulation « pinball » [BN91], le contact est détecté en se basant sur un critère de distance entre les noeuds de chaque surface. Cette approche (peu recommandée en cas de maillages incompatibles) ne permet pas de traiter correctement le cas des grandes déformations. C'est pourquoi on lui préfère généralement l'appariement « noeud-surface », que l'on associe à la représentation « maître-esclave ».

Représentation « maître-esclave » Suite à la dissymétrie induite par la discrétisation éléments finis dans les relations de contact, on est amené à distinguer les deux surfaces, l'une étant qualifiée de *maître* l'autre d'*esclave* (cf. Figure 2.6a). Par la suite nous indiquerons de la variable m les grandeurs relatives à la surface *maître* discrète $\tilde{\Gamma}_m$ et de la variable e celles relatives à la surface *esclave* discrète $\tilde{\Gamma}_e$.

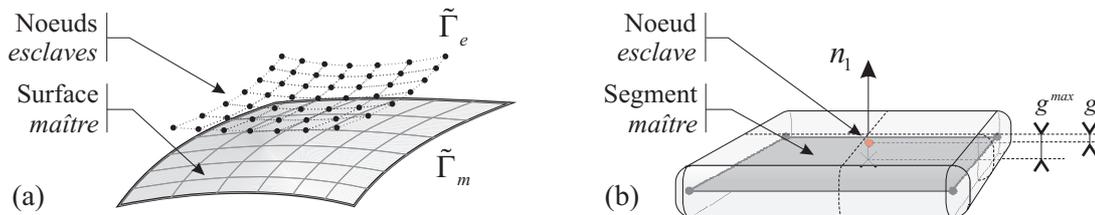


Figure 2.6: Modélisation de l'interface de contact : (a) Représentation maître-esclave - (b) Définition du gap d'interface.

Le traitement du contact consiste alors à empêcher les noeuds *esclaves* de pénétrer la surface *maître*. Nous remarquerons cependant que l'inverse reste possible, c'est-à-dire que les noeuds de la surface *maître* peuvent éventuellement pénétrer la surface *esclave*. Pour certaines géométries, ce phénomène peut devenir gênant et conduire à une interpénétration trop importante d'un relief *maître* dans la surface *esclave*. Par conséquent lorsque les surfaces présentent des tailles d'éléments très différentes, il est préférable d'affecter le maillage le plus fin au côté *esclave*.

Notion de gap d'interface Lorsque le contact s'opère entre des éléments coques maillés à la fibre noeuds, les faces discrètes ne coïncident pas exactement avec la surface

extérieure du corps effectivement impacté. L'utilisation d'un gap, noté g^{max} , autour de l'élément est donc un moyen de fournir une épaisseur physique à la surface (cf. Figure 2.6b). En pratique, il détermine la distance maximale à partir de laquelle un segment *maître* interagit avec un tiers noeuds. Si le noeud entre dans le gap, les forces de réaction deviennent effectives. La condition de non pénétrabilité discrète devient alors :

$$(x_m - x_e) \cdot n_m \leq g^{max} \quad (2.1.56)$$

Le choix de la valeur du gap reste toutefois difficile. En effet, comme nous le verrons par la suite, s'il est choisi trop petit, le pas de temps critique chute redoutablement pour éviter que le noeuds traverse la surface (sans la voir) en un seul cycle. Par contre, s'il est trop important, certains noeuds proches mais non associés à l'interface risquent d'être impactés. Lorsque l'interface fait intervenir deux maillages coques, il est naturel de choisir un gap égal à la somme des demies épaisseurs h des surfaces mises en jeu :

$$g^{max} = \frac{1}{2}(h_m + h_e) \quad (2.1.57)$$

2.1.3.2 Réduction de la pénétration

Nous nous intéresserons tout particulièrement ici au cas du contact sans frottement. L'extension au cas du contact frottant sera néanmoins immédiate en ajoutant une composante tangentielle à l'effort de contact f_c (cf. Remarque 1). On rappelle d'après (2.1.53) que pour la méthode par pénalité, l'effort normal de contact s'écrit :

$$f_c = \int_{\Gamma_c} \Phi^t p \, d\Gamma = \int_{\Gamma_c} \beta \Phi^t \Phi \mathcal{H} d\Gamma \cdot g = K_c \cdot g \quad (2.1.58)$$

où g désigne ici le vecteur des pénétrations et K_c la matrice de raideur de contact. Lorsque qu'une pénétration est détectée, un ressort fictif (sans masse) est donc introduit entre les deux entités en contact pour déplacer le noeud pénétrant en dehors de la zone de contact. Ces ressorts artificiels entre les noeuds *esclaves* et la surface *maître* sont créées tant que la pénétration est détectée et sont supprimés dès que la pénétration cesse.

Remarque 1 : Une manière simple de prendre en compte le frottement consiste à utiliser un modèle de ressort « élasto-plastique » pure. Dans ce cas, la composante tangentielle de l'effort, proportionnelle à l'incrément de déplacement transverse, est comparée à une valeur seuil fixé par la loi de *Coulomb* ; si elle est supérieure, la valeur effective de l'effort est réduite à celle du seuil critique (le contact est glissant), si elle est inférieure, elle reste inchangée (le contact est adhérent).

Raideur d'interface Comme nous l'avons déjà évoqué, une des principales difficultés de la méthode de pénalité est d'évaluer la valeur du coefficient β , qui au niveau de l'élément, se traduit par la rigidité k_c du ressort fictif. Afin de respecter au mieux la continuité cinématique, ce ressort doit être aussi rigide que possible sans toutefois provoquer des rebonds numériques (haute fréquence) après l'impact. Nous verrons qu'une raideur trop importante influe la stabilité des calculs et par conséquent fait chuter le pas de temps critique.

Il existe de nombreuses manières d'exprimer cette raideur d'interface. Elle s'exprime généralement en fonction de la rigidité des pièces en contact, et peut s'interpréter physiquement comme la rigidité des aspérités de surfaces (cf. §2.1.2.4). De manière simplifiée, elle

peut être définie par la rigidité équivalente de deux ressorts en série, l'un correspondant à la surface *maître* l'autre à la surface *esclave* :

$$k_c = \frac{\bar{k}_m \bar{k}_e}{\bar{k}_m + \bar{k}_e} \quad (2.1.59)$$

où \bar{k}_m et \bar{k}_e sont des raideurs de ressorts équivalentes aux raideurs élémentaires des surfaces *maître* et *esclave*. Pour les éléments coques elles s'expriment en fonction de son module de Young E et de son épaisseur h . Pour les éléments solides, elles s'expriment en fonction du volume V et d'air A du segment de contact :

$$\bar{k}|_{coque} = \frac{Eh}{2} \quad \bar{k}|_{volume} = \frac{EA^2}{V} \quad (2.1.60)$$

La définition (2.1.59) n'est toutefois généralement pas suffisante en dynamique dans la mesure où elle conduit souvent à une interpénétration trop importante du noeuds *esclave*. On utilise par conséquent, une rigidité de contact variable fonction de la pénétration g et dont la valeur initiale est fixée par (2.1.59). Le but est alors d'amplifier le coefficient de pénalité dès que la pénétration augmente :

$$k_c(g) = k_c^0 \frac{g^{max}}{g^{max} - g} \quad \text{avec} \quad k_c^0 = \frac{\bar{k}_m \bar{k}_e}{\bar{k}_m + \bar{k}_e} \quad (2.1.61)$$

où g^{max} est le gap d'interface défini par (2.1.57). Nous remarquerons que cette rigidité est non-linéaire et croît de manière asymptotique avec la pénétration lorsque celle-ci tend vers la valeur g^{max} . La figure 2.7a illustre l'évolution de l'effort $f_c = k_c(g)g$ au fur et à mesure que la pénétration g du noeud *esclave* augmente.

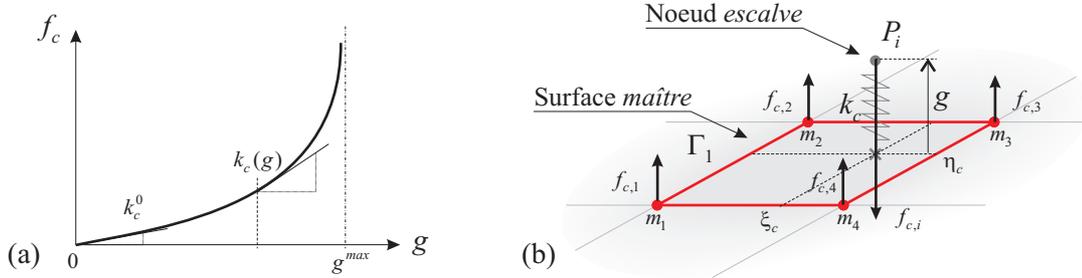


Figure 2.7: Calcul des efforts de contacts normaux : (a) Efforts de réactions aux noeuds *maître* - (b) Dépendance de la raideur d'interface à la pénétration.

Réactions aux noeuds *maître* La pénétration g n'étant calculée que sur les noeuds *eslaves*, la relation (2.1.58) ne fournit que les efforts de contact sur ce côté de l'interface ; nous noterons ici $f_{c,i}$ l'effort calculé sur le noeud *esclave* P_i (cf. Figure 2.7b). Afin de connaître la réaction aux noeuds *maîtres*, on peut exprimer l'égalité des forces au point de contact. On désigne par (ξ_c, η_c) les coordonnées paramétriques de ce point de contact dans l'élément *maître* en vis-à-vis. Pour chaque noeud j de cet élément, le vecteur force de réaction $f_{c,j}$ est alors donnée par [Rad05, LSd98] :

$$f_{c,j} = N_j(\xi_c, \eta_c) f_{c,i} \quad (2.1.62)$$

où N_j sont les fonctions de forme de l'élément *maître*. Nous remarquerons que de cette manière, aucune force externe n'est ajoutée au système et par conséquent que l'énergie totale est conservée. En effet, si on considère l'énergie de contact \mathcal{W}_c , celle-ci peut être calculée pour chaque interface de manière incrémentale entre les instants t^n et t^{n+1} :

$$\mathcal{W}_c^{n+1} = \mathcal{W}_c^n + \left[\sum_{i=1}^{n_e} \Delta f_{c,i}^n \Delta u_i^n + \sum_{j=1}^{n_m} \Delta f_{c,j}^n \Delta u_j^n \right] \quad (2.1.63)$$

où n_m, n_e désignent le nombre de noeuds *maître* et *esclave*, $\Delta u_i^n, \Delta u_j^n$ les incréments de déplacement et $\Delta f_{c,i}^n, \Delta f_{c,j}^n$ les incréments d'effort aux noeuds de $\tilde{\Gamma}_e$ et $\tilde{\Gamma}_m$. En l'absence de frottement, les énergies engendrées par les deux cotés de l'interface sont de même amplitude mais de signe opposé ; l'énergie d'interface reste donc nulle.

2.1.3.3 Contact en dynamique explicite

Les considérations précédentes concernant la géométrie de l'interface étaient valables en statique comme en dynamique. Nous attirerons ici l'attention sur quelques points particuliers aux calculs dynamiques et à l'intégration temporelle des équations de contact en abordant la méthode de pénalité du point de vue de l'algorithme explicite.

Avantage du schéma explicite L'utilisation du schéma explicite pour les applications tel que le crash automobile, est fortement liée au fait que celui-ci s'adapte particulièrement bien à la dynamique du contact. Les raisons en sont les suivantes :

- ✧ Les pas de temps sont petits à cause de la condition de stabilité : les discontinuités n'ont donc que de faibles conséquences sur la pénalisation du pas de temps. Le grand pas de temps rendu possible par les schémas implicites inconditionnellement stable n'est effectivement plus efficace pour des réponses discontinues de la structure.
- ✧ Aucune linéarisation n'étant nécessaire, on évite ainsi les éventuels effets délétères des discontinuités sur un algorithme de Newton : le problème de contact impact introduit une discontinuité dans la matrice jacobienne, empêchant ainsi la convergence de la méthode de Newton.

Outre sa facilité de mise en oeuvre, la méthode de pénalité s'intègre parfaitement à ce type de schéma. Chaque contact est traité comme un élément et est parfaitement intégré dans l'architecture du code. C'est pourquoi elle est généralement préférée à la méthode des multiplicateurs de Lagrange, où l'inversion matricielle effectuée à chaque cycle rend le calcul très coûteux dès que la taille de l'interface devient importante.

Algorithme de pénalité en approche explicite La détection du contact se faisant d'après la géométrie, et les relations de non interpénétration s'appuyant sur la direction des normales aux surfaces, il est inévitable de travailler sur la configuration réactualisée en cas de grands déplacements (nous ne parlons pas ici de grandes déformations dont le traitement est indépendant du contact). En explicite cette réactualisation est généralement faite systématiquement à chaque pas de temps et la configuration finale est connue dès le début du pas.

A chaque cycle, le calcul s'effectue donc dans un premier temps de manière indépendante sur chacun des corps, comme s'il n'y avait pas de contact. Tous les noeuds du modèle sont donc mis à jour, même ceux qui étaient en contact au pas précédent. Ce calcul « découplé » permet d'identifier les zones en contact à la fin du pas de temps. Les conditions de contact peuvent alors être imposées immédiatement après la prise en compte des conditions limites sans qu'aucune itération soit nécessaire : ceci se fait via l'effort de contact f_c ajouté au vecteur force globale (cf. Figure 1.4).

2.1.3.4 Nouvelles conditions de stabilité

Bien que la méthode de pénalité apparaisse comme une technique de régularisation des conditions de contact, elle génère elle aussi des oscillations de la réponse après l'impact ; elle engendre de ce fait une forte sensibilité au pas de temps et au schéma d'intégration utilisé. Certaines précautions supplémentaires doivent donc être prises pour assurer la stabilité des calculs.

Comme nous l'avons vu au premier chapitre, la particularité des calculs explicites réside dans l'utilisation d'un pas temps inférieur à une valeur critique Δt_c , imposée par la condition de stabilité :

$$\Delta t \leq \Delta t_c = \frac{2}{\omega_{\max}} \quad (2.1.64)$$

où ω_{\max} est la plus haute fréquence propre du système. Cette condition évite que l'onde de déformation ne traverse plus d'un élément au cours d'un pas de temps. En présence de contact, l'introduction d'une raideur additionnelle aux interfaces peut donc avoir des conséquences sur la stabilité si elle devient comparable à la raideur des éléments mis en jeu. La condition (2.1.64) se décline de trois manières :

- ⇒ le pas de temps élémentaire Δt_{el} .
- ⇒ le pas de temps nodale Δt_{nod} .
- ⇒ le pas de temps cinématique Δt_{kin} .

Nous ne reviendrons pas ici sur la définition classique du pas de temps élémentaire Δt_{el} (1.1.60) déjà détaillé au paragraphe 1.1.3.3. Regardons néanmoins les deux autres définitions utilisées en cas de simulation de contact.

Pas de temps nodal Le pas de temps nodal n'est pas spécifique au problème de contact. Il est toutefois systématiquement pris en compte dès qu'une interface de contact est introduite au modèle. Il correspond en fait à la déclinaison du pas de temps élémentaire pour les éléments de types ressorts. En effet, la vitesse de propagation du son $c = \sqrt{E/\rho}$ n'étant pas définie sur de tels éléments, à priori sans masse, on ne peut calculer leur pas de temps Δt_{el} . On peut cependant rappeler que la fréquence propre maximale du système discret composé de deux masse m_1 , m_2 et d'un ressort de raideur k s'écrit :

$$\omega_{\max} = \sqrt{k \frac{(m_1 + m_2)}{m_1 m_2}} \quad (2.1.65)$$

En approximant les masses m_1 , m_2 du ressort par la demi masse nodale, on en déduit d'après (2.1.64), la condition suivante sur le pas de temps alors qualifié de pas de temps

nodal :

$$\Delta t \leq \Delta t_{nod} = \min_{i \in \tilde{\Omega}} \sqrt{\frac{2m_i}{k_i}} \quad (2.1.66)$$

où m_i et k_i désignent les masse et raideur au noeud i . Ces rigidités nodales dénuées de véritable sens physique sont obtenues par diagonalisation de la matrice de raideur augmentée de la raideur de contact ($K + K_c$). Nous comprenons alors toute la difficulté à choisir une raideur de pénalité pertinente, c'est-à-dire pas trop faible pour vérifier au mieux la non-interpénétration des surfaces, mais pas démesurément grande pour éviter de faire chuter le pas de temps de stabilité.

Remarque 2 : Dans de nombreux codes de calcul, on procède à un contrôle du pas de temps pour éviter que celui-ci ne chute en deçà d'une certaine valeur préalablement fixée. Concernant le pas de temps nodale, ce contrôle consiste généralement à ajouter de la masse aux noeuds où la valeur du pas de temps est la plus faible. On parle alors de contrôle du pas de temps par ajout de masse nodale. Nous reviendrons sur cette artefact de calcul lorsque nous aborderons notre méthode de contact multi-domaines.

Remarque 3 : De la même manière que l'on contrôle le pas de temps nodal Δt_{nod} , on peut dans certains codes de calcul ajuster la rigidité de contact de manière à ce qu'elle ne pénalise la valeur critique Δt_{crit} repérée sur le reste du modèle (hors contact) :

$$k_i = \alpha \frac{m_i}{\Delta t_{crit}^2} \quad (2.1.67)$$

où α est un coefficient de sécurité ($\alpha < 1$). Dans ce cas, la régularisation discutée au paragraphe précédemment n'a plus de sens physique, puisqu'elle n'a plus que pour simple but d'éliminer les plus hautes fréquences propres du système.

Pas de temps d'interface Cette définition propre au logiciel RADIOSS, évite que le noeud *esclave* ne traverse complètement la surface *maître* en un pas de temps. En effet, si les vitesses mises en jeu sont trop importantes rien n'empêche les noeuds *esclaves* de passer outre le gap d'interface en un seul cycle. Cette dernière définition s'apparente donc à une condition cinématique sur chaque noeud i de la surface discrète *esclave* $\tilde{\Gamma}_e$:

$$\Delta t \leq \Delta t_{kin} = \min_{i \in \tilde{\Gamma}_e} \frac{g^{max} - g_i}{2\dot{g}_i} \quad (2.1.68)$$

où g^{max} est le gap d'interface défini par (2.1.57), g_i la pénétration et \dot{g}_i la vitesse de pénétration au noeud i . Le principal paramètre influant est donc ici le gap g^{max} dont une valeur trop faible conduirait à une chute inévitable du pas de temps cinématique. Nous remarquerons naturellement que le pas de temps de stabilité finalement retenu pour le calcul est le minimum des trois valeurs (1.1.60), (2.1.66) et (2.1.68).

2.2 Développement d'une méthode de contact inter-domaine

Nous avons vu en introduisant ce second chapitre que les codes de dynamique explicite résolvent généralement sans difficulté les problèmes de contact lorsque ceux-ci apparaissent dans un seul modèle éléments finis. La question à laquelle nous répondrons ici se pose lorsque les deux surfaces candidates au contact appartiennent à deux sous-domaines distincts alors calculés indépendamment.

Avant de détailler cette approche, commençons par établir un bref état de l'art sur les méthodes de contact multi-domaines existantes à ce jour.

2.2.1 Bref état de l'art sur le contact multi-domaines

Jusqu'ici aucune hypothèse n'a été faite quant à la décomposition en sous-domaine du problème de contact. Celui-ci était alors considéré comme agissant entre deux corps Ω_1 et Ω_2 issus du même domaine (ou modèle) de calcul. Dès lors, le souhait de joindre la modélisation du contact aux stratégies de calcul multi-domaines paraît inévitable. Toutefois, bien que le problème de contact semble particulièrement bien s'approprier aux méthodes de décomposition, il existe encore peu de travaux dans la littérature.

En effet, ce n'est que très récemment, que diverses formes de décomposition ont été introduites dans la formulation des problèmes de contact afin d'accélérer la résolution de ces problèmes fortement non-linéaire. Nous présenterons ici trois méthodes, toutes s'appliquant à des problèmes élasto-statiques.

2.2.1.1 La méthode FETI-C

La méthode FETI-C (pour FETI-Contact) est une méthode de décomposition de domaines avec multiplicateur de Lagrange dédié à la résolution itérative des problèmes de contact sans frottement. Cette méthode inspirée de la méthode FETI standard [FR91, FR94] s'adresse particulièrement aux problèmes statiques d'élasticité linéaire (1.2.1). Elle suppose de plus que les surfaces discrétisées en contact soient coïncidentes.

Avant d'introduire cette méthode, revenons brièvement sur l'approche FETI standard déjà évoqué au paragraphe 1.2.1. Cette méthode de décomposition consiste donc à partitionner le domaine d'étude et à écrire l'équilibre de la structure au niveau de chaque sous-domaines noté Ω_k . L'introduction de multiplicateurs de Lagrange permet de forcer la continuité des déplacements aux interfaces. D'après (2.1.21-2.1.22), notre problème d'élasticité peut s'écrire sous la forme :

$$K_k u_k = f_k - C_k^t \lambda, \quad \text{sur } \Omega_k \quad (2.2.1)$$

$$\sum_{k=1}^{n_{ssd}} C_k u_k = 0, \quad \text{sur } \Gamma_l \quad (2.2.2)$$

La méthode FETI-C apparaît donc comme une extension de la méthode FETI au cas des interfaces de contact entre sous-domaines. En effet, alors que la méthode FETI a pour but d'imposer la continuité des déplacements aux interfaces de liaison dites parfaites Γ_l , la méthode FETI-C a pour but de vérifier les conditions de *Signorini* (2.1.10-2.1.12) aux

interfaces de contact potentiel Γ_c . L'algorithme se doit alors de distinguer et repérer la zone de contact actif, puis calculer l'état élastique de chaque sous-domaine :

$$K_k u_k = f_k - \hat{C}_k^t \lambda, \quad \text{sur } \Omega_k \quad (2.2.3)$$

$$\sum_{k=1}^{n_{ssd}} C_k u_k \leq 0, \quad \text{sur } \Gamma_c \cup \Gamma_l \quad (2.2.4)$$

où \hat{C}_k ne diffère que des matrices de liaison classique C_k qu'en cas de zone de contact inactive. La différence majeure entre les formulations concerne la non-linéarité du problème d'interface induite par l'inégalité (2.2.4). L'idée de la méthode FETI-C est donc de construire un algorithme de résolution non-linéaire basé sur le solveur de type gradient conjugué préconditionné de la méthode standard. Une description détaillée de la procédure itérative, où conditions de contact et d'équilibre sont mises à jour simultanément, est faite dans [DF00]. Une variante de l'algorithme où les conditions de contact et d'équilibre sont traitées séparément au cours de deux niveaux d'itérations distincts est proposée dans [DFS].

Comme pour l'approche FETI classique, on peut montrer la « scalabilité » numérique de l'algorithme. Ceci constitue l'un des principaux avantages de la méthode, pour laquelle les propriétés de convergence de la procédure itérative sont asymptotiquement indépendantes de la taille du problème à résoudre et du nombre de sous-domaines utilisés.

2.2.1.2 La méthode CONTRAST

La seconde approche qu'il nous a semblé intéressant d'étudier concerne la méthode CONTRAST (CONtact TRidimensionnel dans les Assemblages STatiques). Cette méthode développée dans [Cha96, CD00] est une extension au cas 3D de la méthode proposée dans [BCCL96]. Comme l'approche FETI-C, la méthode CONTRAST s'applique aux problèmes statiques comprenant des interfaces de liaisons parfaites et des interfaces de contacts unilatéraux. Elle permet cependant de traiter l'éventuel frottement agissant sur la zone de contact.

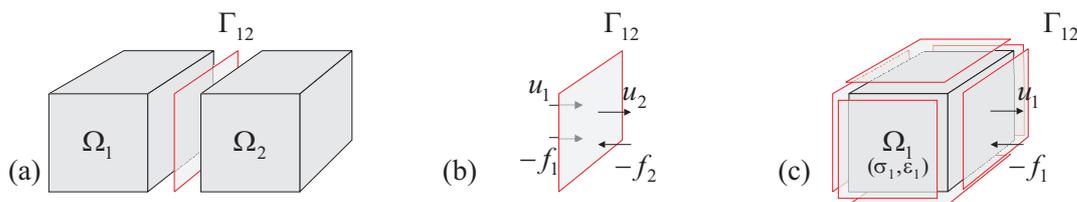


Figure 2.8: Méthode CONTRAST - (a) Décomposition en sous-structures et interfaces - (b) Problème sur l'interface - (c) Problème sur la sous-structure.

La présente étude est basée sur une décomposition particulière des assemblages en sous-structures et interfaces (cf. Figure 2.8a). Toutes deux sont considérées comme des entités mécaniques à part entière disposant de leurs propres inconnues et de leurs propres équations. L'idée est donc de traiter indépendamment les problèmes mécaniques sur chaque entité. On distingue :

- ⇨ Le problème mécanique sur chaque sous-structure (cf. Figure 2.8c) : il s'agit ici d'un problème classique, résolu par la méthode des éléments finis, dont le but est de trouver le couple (σ_k, ϵ_k) vérifiant l'équation d'équilibre (1.1.4), les liaisons cinématiques limites (1.1.1), et la loi de comportement (1.1.5) (élastique linéaire).
- ⇨ Le problème mécanique sur une interface (cf. Figure 2.8b) : pour une interface Γ_{12} , les équations dépendent du comportement de la liaison à modéliser. Ces équations mettent en jeu les champs d'effort et de déplacement locaux de part et d'autre de l'interface. Elles peuvent s'écrire sous forme d'une relation de comportement éventuellement non-linéaire :

$$R(u_1, f_1, u_2, f_2) = 0 \quad (2.2.5)$$

Cette décomposition originale confère un rôle majeur aux interfaces en leur permettant de modéliser facilement et avec précision les non-linéarités tel que le contact (avec ou sans frottement). L'intérêt est que ces non-linéarités sont traitées localement, en associant une loi de comportement (2.2.5) adaptée à chaque interface ; cette dernière peut par conséquent présenter une géométrie complexe. Cette approche permet, contrairement à d'autres techniques, de ne pas introduire de variables supplémentaires (multiplicateurs de Lagrange) dans la résolution des problèmes globaux.

La méthode de résolution proposée pour résoudre un tel problème est basée sur les principes de la méthode LATIN (LArge Time INcrement method) [Lad96]. Elle s'inscrit néanmoins dans une situation dégénérée de cette méthode puisque le problème d'évolution en temps n'est ici pas considéré. Seule la solution du problème à l'instant final est recherchée.

Le premier point de la stratégie consiste à séparer les difficultés afin d'éviter la simultanéité du caractère global et du caractère non-linéaire du problème. Les différentes équations sont alors séparées en deux groupes :

- ⇨ Les équations locales en variable d'espace et éventuellement non-linéaires.
- ⇨ Les équations linéaires et éventuellement globales en variable d'espace.

Dans l'espace des solutions $s = (\epsilon, u; \sigma, f) = \cup_k \{s_k \text{ définis sur } \Omega_k\}$ admissibles, on définit deux sous-espace Σ_Ω et Σ_Γ associés à chacun des groupes d'équations, l'un satisfaisant

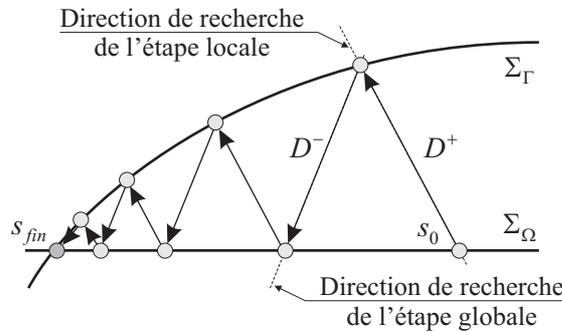


Figure 2.9: Procédure itérative de la méthode CONTRAST.

l'équilibre des sous-structures, l'autre satisfaisant la relation d'interface (pouvant être non-linéaire dans le cas d'un contact unilatéral). Le problème est donc de trouver la solution vérifiant à la fois le comportement des sous-structures ($s \in \Sigma_\Omega$) et le comportement des interfaces ($s \in \Sigma_\Gamma$).

Le second point consiste à utiliser un schéma itératif à deux étapes suivant deux directions de recherche : une direction (D+) de « montée » pour l'étape dite locale et une direction (D-) de « descente » pour l'étape dite semi-globale (ou globale par sous-structure). A partir d'un élément initial s_0 , le processus converge, par une succession d'étapes locales et globales, vers l'élément solution s_{fin} à l'intersection des deux ensembles Σ_Ω et Σ_Γ (cf. Figure 2.9).

Enfin, nous remarquerons que la méthode CONTRAST est, comme de nombreuses méthodes de décomposition, basée sur une stratégie de calcul parallèle. Ce parallélisme y est principalement employé pour introduire de la modularité dans la description du problème, mais aussi pour permettre l'utilisation de maillages incompatibles aux interfaces. Il offre alors inmanquablement une réduction du coût numérique par rapport aux approches éléments finis classiques.

2.2.1.3 Algorithme de Newton-Schur

La dernière méthode que nous présenterons ici [BAV01], diffère légèrement des deux approches précédentes. Il nous a toutefois semblé intéressant de s'y préoccuper de part sa stratégie de décomposition semblable à celle que nous adopterons par la suite. En effet, contrairement aux autres approches, les interfaces physiques de contact y sont traitées comme des interfaces internes aux sous-domaines. Autrement dit, les interfaces de contact sont localisées à l'intérieur des sous-domaines et ne constituent en aucun cas des interfaces de décomposition.

Il s'agit donc ici d'une méthode de décomposition au sens large, comme nous l'entendons au paragraphe 1.2, hormis que la présente technique est principalement dédiée aux problèmes non-linéaires et particulièrement aux problèmes multi-contacts frottant. Comme précédemment cette étude a été développée dans un cadre élasto-statique.

Dans cette approche, la modélisation du contact frottant est basée sur une formulation hybride [AC91] inspirée de la méthode du Lagrangien augmenté. On montre alors d'après (2.1.35) et (2.1.36) que l'équilibre discret des corps en contact est gouverné par le système non-linéaire suivant :

$$f_{int}(u) - f_{ext} + C^t \mathcal{F}(u, \lambda) = 0 \quad (2.2.6)$$

$$-\frac{1}{\beta}(\lambda - \mathcal{F}(u, \lambda)) = 0 \quad (2.2.7)$$

où C est l'opérateur de restriction de Ω sur la surface de contact Γ_c , β est le coefficient de pénalité, et où $\mathcal{F}(u, \lambda)$ désigne l'opérateur de contact discrétisé. En traitant simultanément les variables u et λ , avec $s = (u, \lambda)$, le système précédent peut s'écrire comme la somme d'une contribution élastique différentiable G et d'une contribution de contact frottant non-différentiable \mathcal{F} :

$$G(s) + \mathcal{F}(s) = 0 \quad (2.2.8)$$

L'idée est donc de résoudre ce système non-linéaire et non différentiable en combinant la méthode de Newton à une méthode de décomposition de domaine de type *Schur primale*. L'algorithme de Newton est alors utilisé comme méthode de résolution non-linéaire standard dans lequel le problème linéarisé est résolu, à chaque itération, par une méthode du complément de *Schur*. Comme nous l'avons vu au paragraphe 1.2.1.2, cette méthode permet de résoudre le problème linéarisé de manière itérative en le décomposant en une multitude de sous-problèmes.

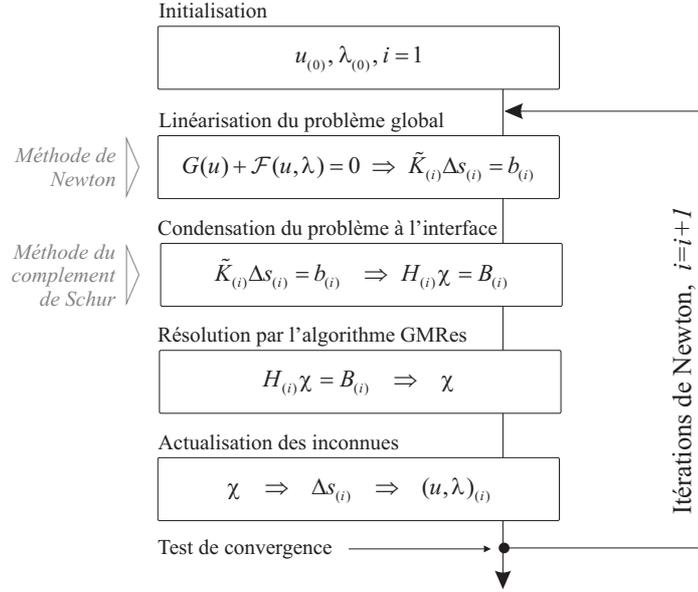


Figure 2.10: Algorithme de Newton-Schur pour les problèmes de contact frottant.

Revenons tout d'abord sur la méthode de Newton que nous présenterons de manière plus détaillée au chapitre 3. Cette méthode fondée sur la linéarisation du problème (2.2.8) revient à résoudre, à chaque itération i le problème suivant :

$$(K_{(i)} + A_{(i)})\Delta s_{(i)} = G(s_{(i)}) + \mathcal{F}(s_{(i)}) \quad (2.2.9)$$

$$\Leftrightarrow \tilde{K}_{(i)}\Delta s_{(i)} = b_{(i)}, \quad \text{avec} \quad \Delta s_{(i)} = s_{(i+1)} - s_{(i)} \quad (2.2.10)$$

où la matrice $K_{(i)}$ est la matrice de raideur élastique usuelle et $A_{(i)}$ est la matrice tangente. Nous remarquerons ici que l'introduction de frottement rend la matrice $A_{(i)}$ et par conséquent la matrice $\tilde{K}_{(i)}$ non-symétrique.

Le système linéarisé (2.2.10) peut ensuite être traité par la méthode *Schur primale*, laquelle consiste à condenser le problème sur les inconnues d'interface Γ_l . Nous insisterons ici sur le fait qu'il s'agit de l'interface de décomposition et non de contact. D'après (1.2.10) le problème condensé s'écrit donc :

$$H_{(i)}\chi = B_{(i)} \quad (2.2.11)$$

où les opérateurs H et B découlent aisément des expressions (1.2.11) et (1.2.12). Le problème d'interface (2.2.11) peut enfin être résolu par la méthode du gradient conjugué

ou la méthode GMRes (Generalized Minimum Residual) [SS86] dans le cas d'un système non-symétrique. Un préconditionneur de type Neumann-Neumann multi-niveau est utilisé pour palier au mauvais conditionnement du système et à la prise en compte des éventuels mouvements de corps rigides des sous-domaines flottant. La figure 2.10 affiche une vision globale de cet algorithme de Newton-Schur.

2.2.2 Méthode de la *peau artificielle*

Après avoir balayé les quelques techniques de contact inter-domaines existantes en régime statique, détaillons à présent la méthode mise en oeuvre dans notre approche de couplage dynamique [BCDV06]. Le principe de cette méthode que nous appellerons « méthode de la peau artificielle » repose sur la juxtaposition d'une interface de contact potentiel Γ_c avec une interface de décomposition ou de liaison Γ_l entre sous-domaines. Nous commencerons ici par présenter la stratégie de recouvrement de la peau artificielle qui est au coeur de la méthode.

2.2.2.1 Stratégie de recouvrement/duplication

L'approche que nous proposons ici a été développée dans le cadre précis des calculs couplés RAD2RAD-RADIOSS. L'objectif n'est alors pas de reprogrammer les algorithmes de contacts dans le code de couplage RAD2RAD, mais plutôt de réutiliser les algorithmes implémentés sous RADIOSS : nous nous appuyerons donc sur une utilisation très prononcée de ces algorithmes largement discutés dans la section 2.1.3. L'idée est de renvoyer les interfaces de contact « frontières », à l'intérieur des sous-domaines [BAV01]. La méthode suggère pour cela d'étendre la surface du sous-domaine à l'intérieur duquel sera géré le contact en y collant une peau, sans masse ni raideur, copie de la surface du second sous-domaine. Cette peau sans caractéristiques mécaniques sera appelée *peau artificielle*. On s'aperçoit ici de la nature dissymétrique du problème dans la mesure où les algorithmes de contact ne s'appliquent qu'à l'un des sous-domaines que l'on qualifiera alors de sous-domaine *maître*.

Regardons plus précisément comment cette stratégie se décline dans un simple contact entre deux sous-domaines : le sous-domaine qualifié de *maître* est noté Ω_1 et le sous-domaine *esclave* noté Ω_2 . Dans ce cas, le contact devant être traité à l'intérieur de Ω_1 , la méthode consiste à copier une partie des noeuds *esclaves* de Ω_2 dans Ω_1 qui verra donc sa taille augmenter. Nous appellerons noeuds *copiés* ces noeuds dupliqués dans Ω_1 et qui, dès lors, seront assimilés à des noeuds *esclaves* dans l'interface de contact interne au sous-domaine *maître* (cf. Figure 2.11).

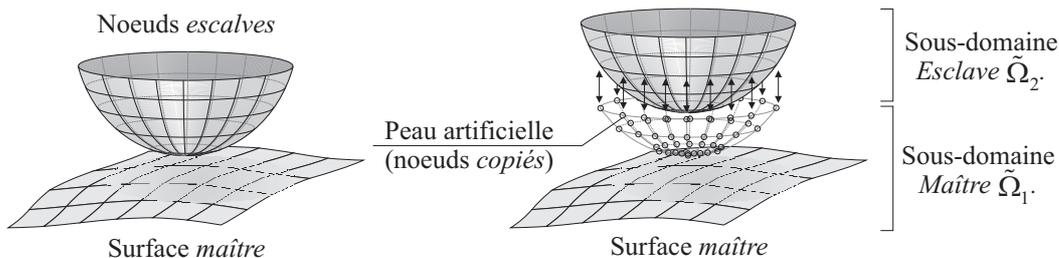


Figure 2.11: Principe du recouvrement par duplication des noeuds d'interfaces.

Les noeuds *esclaves* et les noeuds *copiés* seront par ailleurs liés rigidement par une interface de liaison. La *peau artificielle* support des noeuds *copiés*, est alors confondu avec l'interface de liaison discrétisé $\tilde{\Gamma}_l$. En rappelant que $\tilde{\Gamma}_{c2}$ désigne l'ensemble des noeuds susceptible d'être candidat au contact coté du sous-domaine *esclave*, on peut écrire :

$$\tilde{\Gamma}_l = \tilde{\Gamma}_{c2} \quad (2.2.12)$$

D'autre part, l'interface de contact discrète, appelé $\tilde{\Gamma}_c$, sera confondu d'après (2.1.3) avec la surface *maître* notée $\tilde{\Gamma}_{c1}$:

$$\tilde{\Gamma}_c = \tilde{\Gamma}_{c1} \quad (2.2.13)$$

Remarque 1 : Le principal inconvénient de cette approche vient du fait que la duplication des noeuds de Ω_2 dans Ω_1 augmente le nombre de ddls total du système. Par conséquent, les noeuds *esclaves* candidat au contact devront être correctement choisis.

2.2.2.2 Liaison rigide entre les noeuds *esclaves* et *copiés*

Intéressons nous maintenant à l'interface de liaison, Γ_l et à la méthode de décomposition adoptée pour relier les sous-domaines Ω_1 et Ω_2 sur les noeuds *esclaves*. L'avantage de la duplication décrite précédemment est qu'elle conduit à une parfaite coïncidence des noeuds de part et d'autre de l'interface Γ_l . L'utilisation d'une méthode de décomposition *duale* n'est par conséquent pas justifiée dans cette approche. Nous choisirons donc ici une approche *primale* dont le principal avantage est de ne pas ajouter d'inconnues supplémentaires.

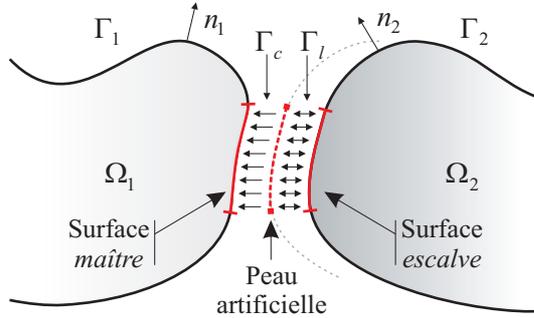


Figure 2.12: Décomposition du domaine de calcul.

On suppose les deux sous-domaines Ω_1 et Ω_2 partitions disjointes du domaine Ω tel que :

$$\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma_l \quad (2.2.14)$$

On peut ainsi renuméroter les degrés de libertés du système de manière à isoler les inconnues associées exclusivement à chacun des deux sous-domaines et à l'interface. En étendant le formalisme de la méthode de *Schur primale* évoquée au paragraphe 1.2.1.2 au cadre de la dynamique, on peut écrire :

$$\begin{bmatrix} \hat{M}_1 & 0 & 0 \\ 0 & \hat{M}_2 & 0 \\ 0 & 0 & M_l \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_l \end{Bmatrix} = \begin{Bmatrix} f_{1,ext} - f_{1,int} \\ f_{2,ext} - f_{2,int} \\ f_l \end{Bmatrix} \quad (2.2.15)$$

où \hat{M}_1, \hat{M}_2 correspondent aux matrices de masses réduites (définies sur les domaines Ω_1 et Ω_2 hors interface), et où l'indice l est relative aux degrés de liberté de l'interface $\tilde{\Gamma}_l$. La troisième équation traduit donc l'équilibre des noeuds d'interface soumis aux sollicitations des deux sous-domaines. Le vecteur f_l peut alors être vu comme la somme des contributions d'efforts f_{l1} et f_{l2} :

$$f_l = f_{l1} + f_{l2} \quad (2.2.16)$$

L'écriture (2.2.15) met en évidence l'absence de couplage direct entre les variables situées sur chacun des sous-domaines : il n'est pas nécessaire d'assembler le système complet. Dans la résolution, on utilise alors les matrices de masses locales suivantes :

$$M_1 = \begin{bmatrix} \hat{M}_1 & 0 \\ 0 & M_{l1} \end{bmatrix} \quad \text{et} \quad M_2 = \begin{bmatrix} \hat{M}_2 & 0 \\ 0 & M_{l2} \end{bmatrix} \quad (2.2.17)$$

où les termes M_{l1} et M_{l2} représentent les masses des noeuds d'interface vues par chacun des sous-domaines. Une décomposition du problème *free/link* permet d'accéder aux calculs de \ddot{u}_l :

⇨ Calcul *free*

$$\begin{bmatrix} \hat{M}_k & 0 \\ 0 & M_{lk} \end{bmatrix} \begin{Bmatrix} \ddot{u}_{k,free} \\ \ddot{u}_{lk,free} \end{Bmatrix} = \begin{Bmatrix} f_{k,ext} - f_{k,int} \\ f_{lk} \end{Bmatrix}, \quad \forall k \in \{1, 2\} \quad (2.2.18)$$

⇨ Calcul *link*

$$\begin{bmatrix} \hat{M}_k & 0 \\ 0 & M_{lk} \end{bmatrix} \begin{Bmatrix} \ddot{u}_{k,link} \\ \ddot{u}_{lk,link} \end{Bmatrix} = \begin{Bmatrix} 0 \\ f_{lj} \end{Bmatrix}, \quad \forall k \neq j \in \{1, 2\} \quad (2.2.19)$$

Le calcul *free* conduit au calcul des inconnues cinématiques internes à chaque sous-domaine ainsi qu'au calcul des quantités sans liaison à l'interface. Le calcul *link* conduit au calcul des corrections à apporter aux quantités *free* pour prendre en compte la liaison entre sous-domaines. Il ne modifie en rien les grandeurs internes aux sous-domaines. On peut néanmoins écrire pour les noeuds situés sur l'interface :

$$\ddot{u}_{l1} = \ddot{u}_{l1,free} + \ddot{u}_{l1,link} \quad (2.2.20)$$

$$\ddot{u}_{l2} = \ddot{u}_{l2,free} + \ddot{u}_{l2,link} \quad (2.2.21)$$

En exprimant la continuité des accélérations à l'interface ($\ddot{u}_{l1} = \ddot{u}_{l2}$) ainsi que l'équilibre de cette interface ($f_{l1} + f_{l2} = 0$), le problème *link* (2.2.19) peut se réécrire sous la forme :

$$\ddot{u}_{l1,link} = \frac{M_{l2}(\ddot{u}_{l2,free} - \ddot{u}_{l1,free})}{M_{l1} + M_{l2}} \quad (2.2.22)$$

$$\ddot{u}_{l2,link} = \frac{M_{l1}(\ddot{u}_{l1,free} - \ddot{u}_{l2,free})}{M_{l1} + M_{l2}} \quad (2.2.23)$$

En injectant (2.2.22) dans (2.2.20), on obtient finalement :

$$\ddot{u}_{l1} = \ddot{u}_{l2} = \frac{M_{l1}\ddot{u}_{l1,free} + M_{l2}\ddot{u}_{l2,free}}{M_{l1} + M_{l2}} \quad (2.2.24)$$

On retrouve ici la méthode *Schur primale* pour laquelle la continuité des inconnues

primales entre les deux sous-domaines est imposée de manière forte. La grandeur cinématique choisie ici pour imposer cette continuité sont les accélérations.

Remarque 2 : Dans cette stratégie dite de la *peau artificielle*, nous verrons que les quantités nodales, et notamment les masses aux noeuds *esclaves* sont transférées à leur homologue *copiées* dans le sous-domaine *maître*. Les noeuds en vis-à-vis ayant donc dans ce cas la même masse, $M_{l1} = M_{l2}$, on a d'après (2.2.24) :

$$\ddot{u}_{l1} = \ddot{u}_{l2} = \frac{1}{2}(\ddot{u}_{l1,free} + \ddot{u}_{l2,free}) \quad (2.2.25)$$

Remarque 3 : En pratique, le calcul *free* est réalisé par le code de dynamique explicite RADIOSS, lequel inclut le calcul des efforts de contact pour le sous-domaine *maître* Ω_1 . Il est dans ce cas nécessaire d'ajouter les forces de contact $f_{1,c}$ au second membre de (2.2.18), pour $k = 1$.

2.2.2.3 Démarche générale de calcul du contact inter-domaines

La démarche adoptée pour gérer le contact entre deux sous-domaines est ici décrite dans la configuration de calcul de la figure 2.12. On peut alors distinguer trois grandes phases dans le déroulement de la méthode :

② Phase de préparation des modèles

- ⇨ Décomposition du modèle initial Ω en deux sous-domaines Ω_1 et Ω_2 .
- ⇨ Duplication des noeuds *esclaves* de Ω_2 sur le sous domaine *maître* Ω_1 .

② Phase d'initialisation du calcul

- ⇨ Construction de la table de connectivité entre les noeuds *esclaves* et *copiés*.
- ⇨ Copie des conditions initiales (de type vitesse ou déplacement imposé) des noeuds *esclaves* vers les noeuds *copiés*.

② Phase de calcul (boucle de temps)

- ⇨ Calcul indépendant de l'équilibre dynamique *free* sur chaque sous-domaine. Le contact est alors géré par le sous-domaine *maître* Ω_1 .

$$M_k \ddot{u}_{k,free}^n = f_{k,ext}^n - f_{k,int}^n + f_{k,c}^n \quad \forall k \in \{1, 2\} \quad (2.2.26)$$

- ⇨ Calcul du problème *link* sur les noeuds d'interfaces et échange de données entre le programme principal et les deux calculs clients. Cette étape assure la continuité cinématique à l'interface de liaison entre les sous-domaines.

$$\ddot{u}_{lk,link}^n = \frac{1}{2}(\ddot{u}_{lj,free}^n - \ddot{u}_{lk,free}^n) \quad \forall k \neq j \in \{1, 2\} \quad (2.2.27)$$

- ⇨ Calcul des quantités cinématiques globales par sous-domaine à partir des relations d'intégration du schéma explicite.

$$\ddot{u}_k^n = \ddot{u}_{k,free}^n + \ddot{u}_{k,link}^n \quad \forall k \in \{1, 2\} \quad (2.2.28)$$

$$\dot{u}_k^n = \dot{u}_{k,free}^n + \frac{\Delta t_k}{2} \ddot{u}_{k,link}^n \quad \forall k \in \{1, 2\} \quad (2.2.29)$$

$$u_k^n = u_{k,free}^n \quad \forall k \in \{1, 2\} \quad (2.2.30)$$

En pratique, on peut omettre le problème *link* qui est ici trivial, pour se concentrer directement sur le calcul des quantités globales basé sur (2.2.25). Ce calcul ne faisant intervenir que les inconnues à l'interface de liaison, nous abandonnerons par la suite l'indice l sur ces grandeurs.

2.2.3 Extension aux liaisons permanentes

Arrêtons-nous un instant sur une utilisation « détournée » de notre stratégie de *peau artificielle*. Cette méthode développée pour faire face aux problèmes de contacts ponctuels (au sens temporaire) peut, moyennant quelques modifications, s'appliquer aux liaisons permanentes pour coller les sous-domaines entre eux. Elle peut alors se substituer sous certaines conditions à la méthode de couplage introduite au paragraphe 1.2.2 (cf. Figure 2.13).

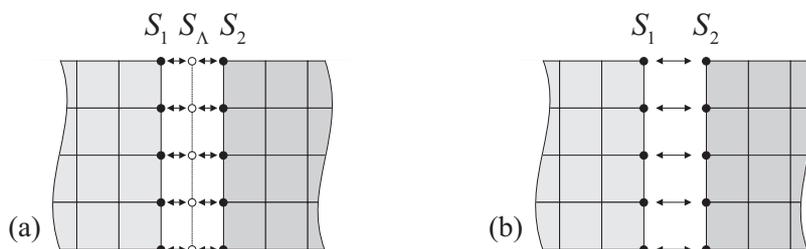


Figure 2.13: Comparaison des collages permanents : (a) Avec multiplicateurs de Lagrange - (b) Sans multiplicateurs de Lagrange.

2.2.3.1 Interface de collage rigide

En effet, la liaison rigide établie entre les noeuds *esclaves* et les noeuds *copiés* peut être utilisée pour coller des noeuds frontières en vis-à-vis. Nous avons vu que cette méthode de couplage « parfait » détaillée dans le paragraphe précédent s'apparente à une technique de décomposition *primale* (cf. §1.2.1.2). Le collage direct noeud à noeud établi par l'égalité des inconnues primales (ici les accélérations) confère une implication forte de la continuité cinématique à l'interface. L'équilibre de la structure globale restera en contre partie vérifié « au mieux ».

La peau, que nous avons précédemment qualifiée d'artificielle, est dans ce cas bien réelle puisque confondue avec le bord d'un des deux sous-domaines en vis-à-vis. La principale distinction par rapport à la liaison rigide établie dans le cadre du contact entre les noeuds *esclaves* et les noeuds *copiés* vient alors du fait qu'aucun transfert de masse n'est ici réalisé entre les noeuds coïncidents. Ces derniers n'ayant pas nécessairement la même masse, le calcul (2.2.24) des accélérations à l'interface devra tenir compte de cette spécificité.

2.2.3.2 Comparaison des méthodes *primale/duale*

L'avantage majeure d'un tel collage par rapport à celui basé sur la méthode des multiplicateurs de Lagrange est qu'il présente un problème *link* trivial (2.2.27) ; aucun système linéaire ne doit ici être résolu à l'interface. Il présente par contre le gros désavantage de

ne pas pouvoir traiter de manière simple les incompatibilités de maillage aux interfaces.

Afin de confronter cette technique de collage *primale* à son homologue *duale*, nous avons repris l'exemple de compression de tube présenté au paragraphe 1.3.2.1. La figure 2.14 montre la déformée finale du tube obtenue pour chacun des calculs suivant :

- ⇒ **Cas ①** : Calcul mono-domaine de référence.
- ⇒ **Cas ②** : Calcul multi-domaine ; liaison **avec** multiplicateurs de Lagrange.
- ⇒ **Cas ③** : Calcul multi-domaine ; liaison **sans** multiplicateurs de Lagrange.

Rappelons que dans les cas ② et ③, le modèle comprend deux sous-domaines PART1 et PART2, reliés entre eux par 32 noeuds. Nous nous placerons ici dans le cas multi-échelles en temps où chacun de ces sous-domaines évolue avec son propre pas de temps de stabilité. Dans le cas avec multiplicateur de Lagrange, le problème d'interface (de dimension $6 \times 32 = 192$ inconnues) est résolu par la méthode de Cholesky.

	Cas ①	Cas ②	Cas ③
PART1	318.3s	177.4s	175.2s
PART2	-	55.4s	58.9s
RAD2RAD	-	216.0s	9.4s
<i>Total</i>	318.3s	452.4s	243.5s

Tableau 2.1: Détail des temps CPU consommés par les différents modèles.

Sur cet exemple où le problème d'interface joue un rôle prépondérant sur le temps de calcul, le tableau 2.1 montre très nettement le gain apporté par cette technique de décomposition sans multiplicateur de Lagrange. Alors que dans le cas ② le problème d'interface pénalise le gain apporté par l'aspect multi-échelles, le cas ③ offre un gain sans équivoque. Le temps RAD2RAD correspondant ici au temps total passé dans ce code ; ceci confirme que la quasi totalité des ressources qui y sont consommées est due à la résolution du problème d'interface. Le temps utilisé dans les procédures de communications des données reste négligeable.

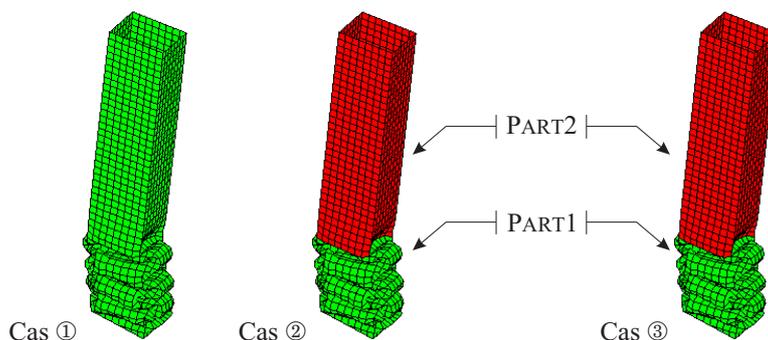


Figure 2.14: Corps creux en compression : déformées des différents modèles au temps $t = 9 \times 10^{-3}s$.

Une étude analogue sur un exemple plus complexe de choc piéton (que nous présenterons

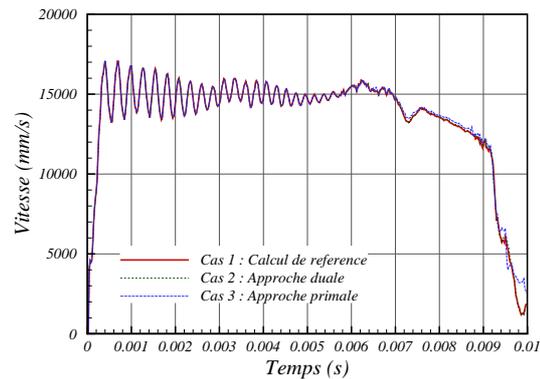


Figure 2.15: Corps creux en compression : vitesse résultante à l'interface.

au paragraphe §2.4.3) fait état d'une économie de près de 95% sur le temps de consommé par notre code de couplage ; sur cette exemple à 375 noeuds d'interface le temps passé dans RAD2RAD passe 9257s pour une liaison avec multiplicateurs de Lagrange à 560.6s pour une liaison sans multiplicateurs.

La figure 2.15 montre la résultante de vitesse d'un des noeuds d'interface entre les sous-domaines. Même si la méthode de collage *primale* affiche un résultat très proche de celui obtenu avec le calcul de référence, elle s'avère néanmoins légèrement moins fidèle que la méthode *duale*. Les faibles discordances se ressentent notamment en fin de simulation, lorsque la déformation subie par l'interface devient importante.

2.3 Collage temporel dans la méthode de couplage dynamique

2.3.1 Communications à pas de temps unique

Rappelons tout d'abord que l'on entend par mono-échelle, le calcul multi-domaines où chaque sous-domaine évolue avec le même pas de temps. Afin d'assurer la liaison entre ces sous-domaines, un certain nombre de variables doivent transiter entre eux via le programme de couplage principal RAD2RAD. Le problème à résoudre se pose donc dans les termes suivants : quelles sont les données à échanger entre sous domaines et quand doit-on les échanger ?

Il apparaît par ailleurs évident que le volume d'informations communiquant entre les sous-domaines à chaque cycle de calcul peut pénaliser le temps de calcul. Par conséquent les variables échangées seront réduites au strict minimum.

2.3.1.1 Pas de temps

Comme nous l'avons vu précédemment, en dynamique explicite, le pas de temps conditionne la stabilité du calcul. Dans le cas mono-échelle en temps, l'échange du pas de temps (généralement variable) entre les sous-domaines est donc nécessaire. Le programme principal se charge de recevoir à chaque cycle, le pas de temps de stabilité des deux sous-domaines en contact et de leur renvoyer la valeur Δt^{n*} correspondante au plus petit des deux.

$$\Delta t_1^{n*} = \Delta t_2^{n*} = \min(\Delta t_1^n, \Delta t_2^n) \quad (2.3.1)$$

Remarque 1 : Dans le cas multi-échelles que nous traiterons par la suite, le pas de temps envoyé par chaque sous-domaine lui est renvoyé sans aucun traitement. On notera alors :

$$\Delta t_1^{n*} = \Delta t_1^n \quad \text{et} \quad \Delta t_2^{n*} = \Delta t_2^n \quad (2.3.2)$$

2.3.1.2 Continuité cinématique à l'interface

Les instants d'équilibres de chaque sous-domaine étant synchronisés dans le cas mono-échelle, la continuité cinématique à l'interface peut-être calculée exactement à chaque cycle. Cette condition de liaison imposée sur les accélérations ($\ddot{u}_1 = \ddot{u}_2$) revient à corriger les termes *free* issus de chaque sous-domaine en y ajoutant la contribution *link*. En notant $\ddot{u}_{1,free}(t^n)$ l'accélération *free* des noeuds d'interface envoyée par le sous-domaine Ω_1 à

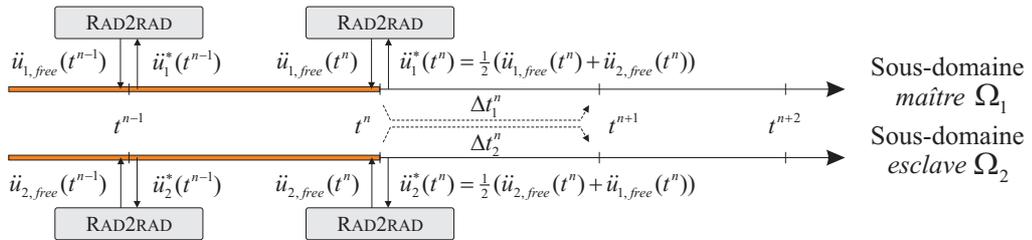


Figure 2.16: Échange des accélérations sur un calcul mono-échelle en temps.

l'instant t^n et $\ddot{u}_1^*(t^n)$ l'accélération corrigée qui lui est retournée, on peut écrire d'après (2.2.25) :

$$\ddot{u}_1^*(t^n) = \frac{1}{2}(\ddot{u}_{1,free}(t^n) + \ddot{u}_{2,free}(t^n)) \quad (2.3.3)$$

De la même façon, l'accélération des noeuds d'interface de Ω_2 est corrigée à chaque cycle par l'accélération des noeuds de Ω_1 en vis-à-vis (cf. Figure 2.16). On a donc finalement égalité des accélérations corrigées de part et d'autre de l'interface :

$$\ddot{u}_1^*(t^n) = \ddot{u}_2^*(t^n) \quad (2.3.4)$$

2.3.1.3 Masses et raideurs nodales des noeuds *copiés*

Comme nous l'avons vu précédemment (cf. §2.1.3.2), le contact entre deux corps se traduit par l'apparition d'une force de contact fonction des masses et raideurs nodales. Les noeuds *esclaves* dupliqués dans le sous-domaine Ω_1 étant de masse et de raideur nulles, il convient de leur affecter les valeurs m_2 et k_2 des noeuds réels coïncidents.

En théorie, la masse du modèle reste constante au cours de la simulation ; le transfert des masses vers les noeuds *copiés* peut donc être fait une fois pour toute avant le calcul. Toutefois, les codes de dynamique explicite font généralement appel à des techniques de traitement du pas de temps (cf. §2.1.3.4). Ces techniques consistent à augmenter artificiellement la masse de certains noeuds de façon à ce que le pas de temps nodal (2.1.66) ne passe en dessous d'un certain seuil. Il est donc nécessaire de prendre en compte ces ajouts de masse dans le cadre des calculs multi-domaines en transférant, à chaque cycle, les masses des noeuds *esclaves* vers leur homologue coté *maître* :

$$m_1^*(t^n) = m_2^*(t^n) = m_2(t^{n-1}) + \Delta m_2^{n-1} \quad (2.3.5)$$

où Δm_2^{n-1} représente l'ajout de masse sur le noeud considéré entre t^{n-1} et t^n . D'autre part, nous avons vu que dans certaines formulations, la raideur de contact k_c affectée aux noeuds esclaves dépendait de la pénétration et par conséquent évoluait au cours du calcul. Dans les calculs multi-domaines, il convient donc de prendre en compte cette évolution et corriger la raideur nodale des noeuds *copiés* :

$$k_1^*(t^n) = k_2^*(t^n) = k_2(t^n) \quad (2.3.6)$$

2.3.2 Gestion des pas de temps asynchrones

Notre méthode multi-domaines prend tout son intérêt avec l'utilisation du multi-échelles, en permettant à chaque sous-domaine d'évoluer avec son propre pas de temps de stabilité. Bien que la liste des variables échangées soit identique à celle présentée dans le cas mono-échelles quelques traitements spécifiques doivent ici être abordés.

2.3.2.1 Interpolation en temps

Comme précédemment, à chaque cycle de calcul les accélérations renvoyées à chacun des sous-domaines sont corrigées en y ajoutant les accélérations *free* du sous-domaine voisin :

$$\ddot{u}_1^*(t_1^n) = \frac{1}{2}(\ddot{u}_{1,free}(t_1^n) + \ddot{u}_{2,free}(t_1^n)) \quad (2.3.7)$$

$$\ddot{u}_2^*(t_2^n) = \frac{1}{2}(\ddot{u}_{1,free}(t_2^n) + \ddot{u}_{2,free}(t_2^n)) \quad (2.3.8)$$

Contrairement au cas mono-échelle, les sous-domaines ne sont maintenant plus à l'équilibre aux mêmes instants ($t_1^n \neq t_2^n$) (cf. Figures 2.17, 2.18). Ainsi la correction des accélérations n'est plus si évidente puisque les valeurs $\ddot{u}_{2,free}(t_1^n)$ et $\ddot{u}_{1,free}(t_2^n)$ sont a priori inconnues. Comme expliqué dans [GC01, Her02] on procède pour chaque sous-domaine à une interpolation linéaire des accélérations entre les instants t^n et t^{n-1} . Par exemple pour le sous-domaine Ω_2 , la valeur de son accélération *free* à l'instant t_1^n (avec $t_2^{n-1} < t_1^n < t_2^n$) est évaluée par interpolation linéaire entre les instants t_2^{n-1} et t_2^n :

$$\ddot{u}_{2,free}(t_1^n) = \left[\frac{t_2^n - t_1^n}{t_2^n - t_2^{n-1}} \right] \ddot{u}_{2,free}(t_2^{n-1}) + \left[\frac{t_1^n - t_2^{n-1}}{t_2^n - t_2^{n-1}} \right] \ddot{u}_{2,free}(t_2^n) \quad (2.3.9)$$

$$= \alpha \cdot \ddot{u}_{2,free}(t_2^{n-1}) + (1 - \alpha) \cdot \ddot{u}_{2,free}(t_2^n) \quad (2.3.10)$$

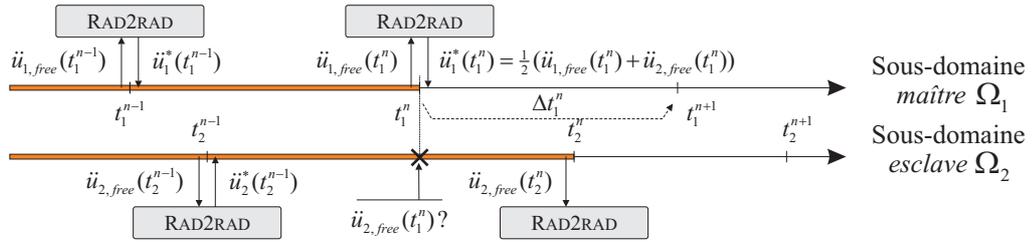


Figure 2.17: Échange des accélérations sur un calcul multi-échelles : avancée du sous-domaine Ω_1 .

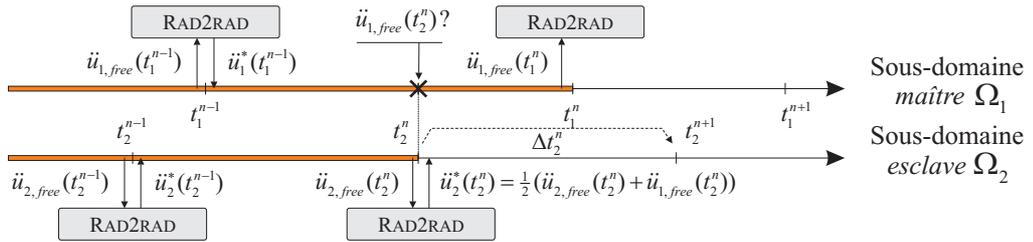


Figure 2.18: Échange des accélérations sur un calcul multi-échelles : avancée du sous-domaine Ω_2 .

Remarque 1 : Une interpolation similaire peut être réalisée sur les valeurs des masses et raideurs nodales. Nous constaterons toutefois que ces corrections de masse (2.3.5) et raideur (2.3.6) peuvent être pénalisantes dans le cas multi-échelles. En effet, supposons que Ω_2 évolue avec un pas de temps plus faible que Ω_1 et que ce pas de temps soit imposé par un de ses noeuds *esclaves*. Le noeud *copié* coté *maître*, sera d'après (2.1.66) doté du même pas de temps nodal et par conséquent pénalisera le calcul sur Ω_1 .

2.3.2.2 Problème de continuité cinématique

Si cette méthode d'interpolation fonctionne parfaitement dans le cas d'une méthode de décomposition *duale*, elle n'est plus acceptable dans cette approche *primale*. En effet, nous avons observé que l'égalisation des accélérations de part et d'autre de l'interface ne suffit pas à assurer la continuité des vitesses et/ou des déplacements. C'est ce que nous vérifions sur le petit exemple ci-dessous.

Exemple : rebond d'une masse sur un plan Cet exemple illustre ce qui se passe au niveau d'une interface de contact lorsqu'un noeud de vitesse initiale v^0 entre en contact avec un élément. La masse ponctuelle m que constitue ici le sous-domaine *esclave* peut être dupliquée dans le sous-domaine *maître* (plan). Ces deux masses sont alors intégrées indépendamment avec des pas de temps arbitraires différents (et constants).

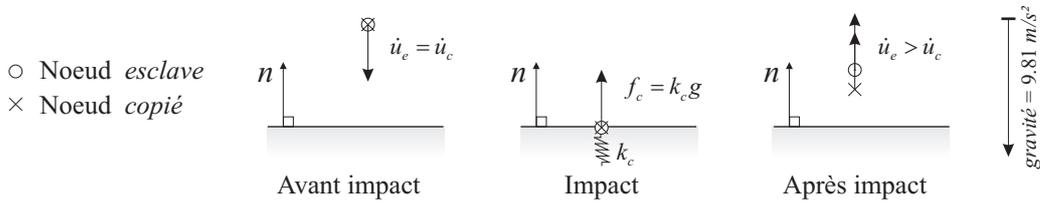


Figure 2.19: Impact d'une masse sur un plan.

Nous supposons que le plan présente une rigidité k_c assimilable à la raideur de contact. On note \dot{u}_e la vitesse du noeud *esclave* et \dot{u}_c la vitesse du noeud *copié*. Les courbes d'accélération, de vitesse et de déplacement des figures 2.20 à 2.22 ont été obtenues avec les paramètres du tableau 2.2.

m	k_c	$\dot{u}_e^0 = \dot{u}_c^0$	$\Delta t_{esclave}$	$\Delta t_{maître}$
$5.0 \times 10^2 \text{ kg}$	$2.0 \times 10^5 \text{ N/m}$	1.1 m/s	$2.3 \times 10^{-4} \text{ s}$	$4.0 \times 10^{-4} \text{ s}$

Tableau 2.2: Paramètres de calcul utilisés pour la simulation d'impact d'une masse sur un plan.

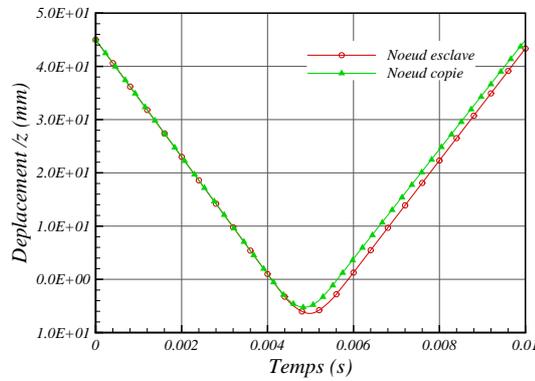


Figure 2.20: Déplacements normaux des noeuds *esclave* et *copié*.

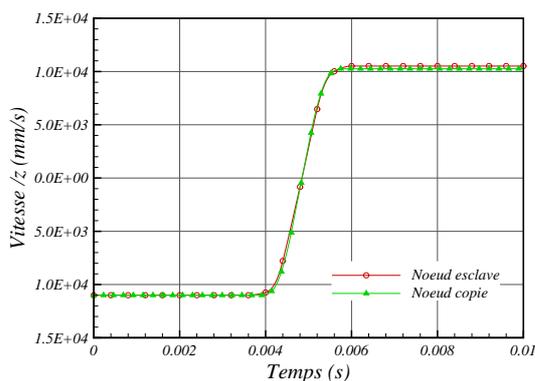


Figure 2.21: Vitesses normales des noeuds *esclave* et *copié*.

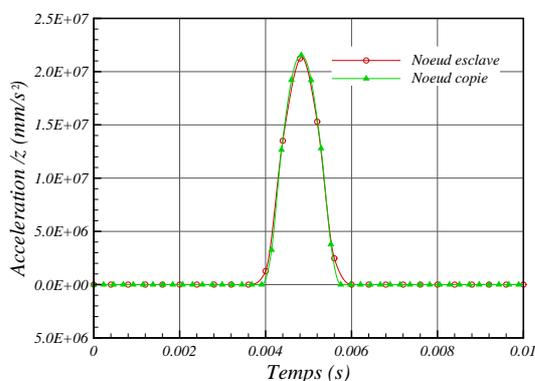


Figure 2.22: Accélérations normales des noeuds *esclave* et *copié*.

Comme le montre la figure 2.19, les noeuds *esclaves* et *copiés* se séparent après impact à $t = 0.005s$. En multi-échelles, la continuité d'accélération entre ces deux noeuds n'est plus imposée exactement mais à partir de grandeurs interpolées à chaque cycle de calcul. L'erreur commise sur l'accélération se répercute et se cumule logiquement sur les grandeurs intégrées que sont les vitesses et les déplacements (cf. Figures 2.20-2.22). Bien qu'une diminution des pas de temps permette de réduire cette erreur et d'améliorer le résultat du calcul, cela reste inacceptable dans le cas général. L'idée est donc d'imposer aux noeuds *copiés* un déplacement identique à celui des noeuds *esclaves*.

2.3.2.3 Correction des accélérations

Nous cherchons donc à imposer ici l'égalité des déplacements de chaque côté de l'interface en modifiant l'accélération des noeuds copiés de Ω_1 . Autrement dit, on cherche à corriger l'accélération $\ddot{u}_1(t^n)$ de ces noeuds de façon à ce que leur « nouveau » déplacement vérifie :

$$u_1^*(t^n) = u_2(t^n) = u_1(t^n) + \Delta u(t^n) \quad (2.3.11)$$

Plaçons nous dans la configuration de calcul décrite par les échelles de temps de la figure 2.17; le calcul Ω_2 étant en avance sur le calcul Ω_1 , il convient de lancer un nouveau

cycle *free* sur Ω_1 . Afin de déterminer la correction à apporter à $\ddot{u}_1(t^n)$, revenons sur les formules d'intégration du schéma de Newmark. D'un point de vue algorithmique, le code RADIOSS calculant les vitesses au milieu du pas de temps, nous préférons réécrire les relations (1.1.33-1.1.34) sous la forme :

$$u^{n+1} = u^n + \Delta t^n \dot{u}^{n+1/2} \quad (2.3.12)$$

$$\dot{u}^{n+1/2} = \dot{u}^{n-1/2} + \Delta t^{n-1/2} \ddot{u}^n \quad (2.3.13)$$

où la vitesse centrée $\dot{u}^{n-1/2}$ et le pas de temps « moyen » $\Delta t^{n-1/2}$ sont données par :

$$\dot{u}^{n-1/2} = (u^n - u^{n-1}) / \Delta t^{n-1} \quad (2.3.14)$$

$$\Delta t^{n-1/2} = (\Delta t^{n-1} + \Delta t^n) / 2 \quad (2.3.15)$$

En injectant (2.3.11) dans (2.3.12), on en déduit le nouveau déplacement calculé sur Ω_1 à la fin du pas :

$$u_1(t_1^{n+1}) = u_1(t_1^n) + \Delta u(t_1^n) + \Delta t_1^n \dot{u}_1(t_1^{n-1/2}) + \Delta t_1^n \Delta t_1^{n-1/2} \ddot{u}_1(t_1^n) \quad (2.3.16)$$

$$= u_1(t_1^n) + \Delta t_1^n \dot{u}_1(t_1^{n-1/2}) + \Delta t_1^n \Delta t_1^{n-1/2} \ddot{u}_1^*(t_1^n) \quad (2.3.17)$$

où \ddot{u}_1^* désigne l'accélération corrigée à renvoyer aux noeuds *copiés* de Ω_1 . On en déduit l'expression de cette nouvelle accélération :

$$\ddot{u}_1^*(t_1^n) = \ddot{u}_1(t_1^n) + \Delta \ddot{u}_1(t_1^n) \quad \text{avec} \quad \Delta \ddot{u}_1(t_1^n) = \frac{\Delta u(t_1^n)}{\Delta t_1^n \Delta t_1^{n-1/2}} \quad (2.3.18)$$

Le terme $\Delta \ddot{u}_1$ correspond donc à la correction apportée à l'accélération des noeuds *copiés* pour que ceux-ci suivent parfaitement leur vis-à-vis côté *esclave*. Nous pourrions donc finalement écrire :

$$\ddot{u}_1^*(t_1^n) = \frac{1}{2} (\ddot{u}_{1,free}(t_1^n) + \ddot{u}_{2,free}(t_1^n)) + \Delta \ddot{u}_1(t_1^n) \quad (2.3.19)$$

En réalité, la correction prend en compte les écarts de déplacements Δu au début du pas de temps et les répercute sur les accélérations. Elle peut donc être qualifiée de rétroactive puisque l'erreur mesurée à l'instant t^n ne sera compensée qu'après l'intégration de l'accélération à l'instant t^{n+1} . Cette technique permet simplement de rectifier à chaque cycle l'écart de position initiale et ainsi éviter que l'erreur en déplacement ne se propage au cours du calcul.

2.3.3 Implémentation à RAD2RAD

2.3.3.1 Algorithme d'avancé en temps généralisé à n_{ssd} sous-domaines

En notant t_{abs}^n l'instant courant de l'échelle absolue, nous pouvons désormais proposer l'algorithme multi-domaines d'avancé en temps, pour des pas de temps asynchrones et variables par sous-domaine. Cet algorithme est analogue à celui utilisé dans la méthode de décomposition *duale* introduit au paragraphe 1.2.4.2 :

- ⇒ ① **Lancement des calculs *free*** (cf. Figure 2.23a) : Une boucle sur l'ensemble des sous-domaines $k = 1, n_{ssd}$ est effectuée de manière à repérer ceux en retard par rapport au temps absolu courant. Lorsque le dernier instant équilibre t_k^n est antérieur à l'instant t_{abs}^n , le temps du sous-domaine est avancé de Δt_k^n et le calcul *free* est lancé.

$$M\ddot{u}_{k,free}^{n+1} = f_{k,ext}(t^{n+1}) - f_{k,int}(p u_k^{n+1}) \quad (2.3.20)$$

$$\dot{u}_{k,free}^{n+1} = p \dot{u}_k^{n+1} + \frac{\Delta t_k}{2} \ddot{u}_{k,free}^{n+1} \quad (2.3.21)$$

$$u_{k,free}^{n+1} = p u_k^{n+1} \quad (2.3.22)$$

- ⇒ ② **Lancement des calculs *link*** (cf. Figure 2.23b) : Une seconde boucle est enfin réalisée sur les sous-domaines afin de lancer le calcul *link*. Seul les sous-domaines pour lesquels $t_k^n = t_{abs}^n$ sont ici concernés. L'interpolation préalable à ces calculs

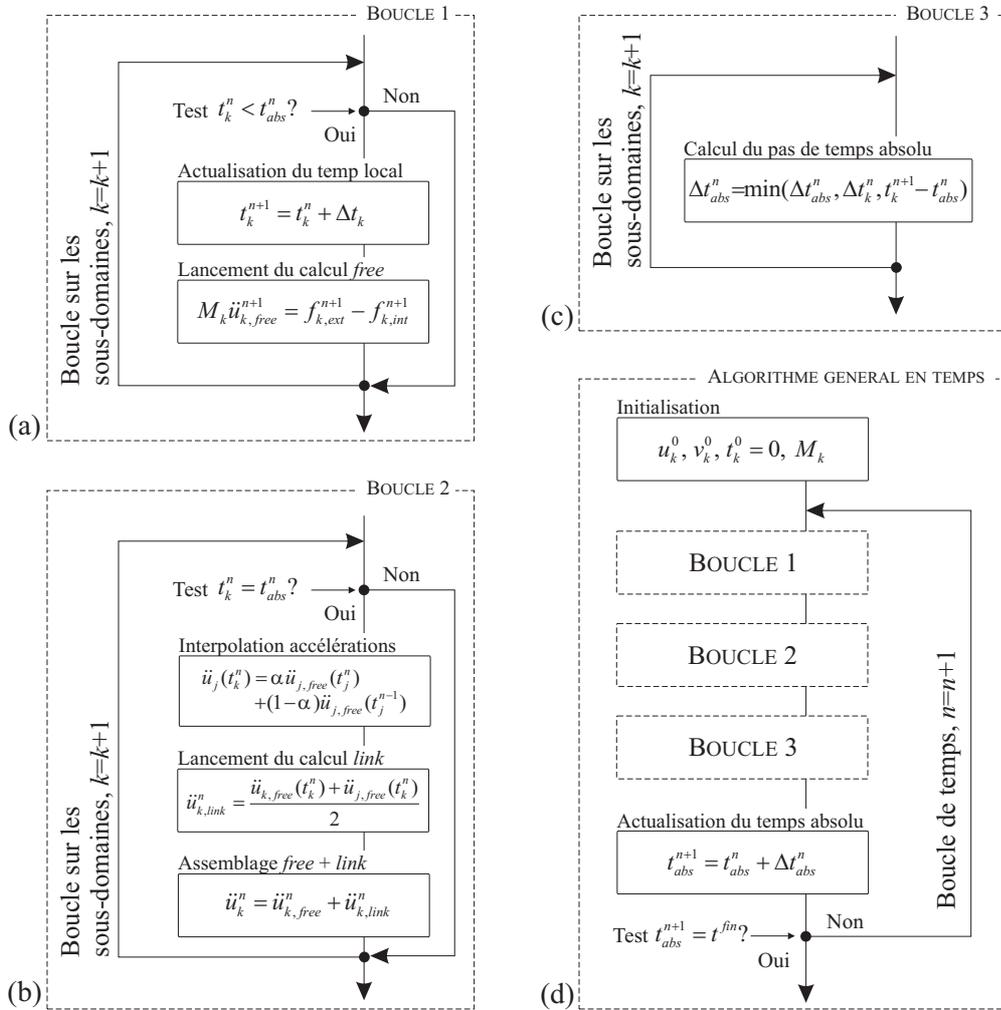


Figure 2.23: Algorithme de collage en temps : (a) Boucle de lancement des calculs *free* - (b) Boucle de lancement des calculs *link* - (c) Recherche du pas de temps absolu - (d) Algorithme général.

est réalisée à partir des accélérations *free* issues de l'étape précédente.

$$\ddot{u}_{j,free}(t_k^n) = \alpha \ddot{u}_{j,free}(t_j^n) + (1 - \alpha) \ddot{u}_{j,free}(t_j^{n+1}) \quad (2.3.23)$$

où j est l'indice du sous-domaine en contact avec le sous-domaine k ($j \neq k$). Dès lors, le problème *link* peut être résolu sur ces mêmes sous-domaines :

$$\ddot{u}_{k,link}^n = \frac{1}{2} (\ddot{u}_{j,free}(t_k^n) - \ddot{u}_{k,link}(t_k^n)) \quad (2.3.24)$$

La remontée aux quantités cinématique globales peut alors être opérée par simple sommation des quantités *free* et *link*.

$$\ddot{u}_k^n = \ddot{u}_{k,free}^n + \ddot{u}_{k,link}^n \quad (2.3.25)$$

$$\dot{u}_k^n = \dot{u}_{k,free}^n + \frac{\Delta t_k}{2} \ddot{u}_{k,link}^n \quad (2.3.26)$$

$$u_k^n = u_{k,free}^n \quad (2.3.27)$$

⇒ ③ **Actualisation du temps absolu** (cf. Figure 2.23b) : Une dernière boucle sur l'ensemble des sous-domaines est ensuite réalisée afin de déterminer de quel pas de temps Δt_{abs}^n l'échelle absolue doit être avancée. Ce pas de temps correspond au minimum entre tout les pas de temps de stabilité courants Δt_k^n et l'écart $t_k^{n+1} - t_{abs}^n$.

$$\Delta t_{abs}^n = \min_k (\Delta t_k^n, t_k^{n+1} - t_{abs}^n) \quad (2.3.28)$$

Ces trois étapes sont naturellement intégrées à la boucle de temps illustrée par la figure 2.23d.

2.3.3.2 Schéma de communications RAD2RAD-RADIOSS

Dans cette stratégie de contact inter-domaine, la principale fonction de RAD2RAD est d'assurer la liaison rigide entre les noeuds candidats au contact du sous-domaine *esclaves* et les noeuds *copiés* du sous-domaine *maître*.

Le schéma de la figure 2.24 illustre l'organisation du calcul couplé dans le cas général de pas de temps asynchrones. Il permet de résumer les différentes quantités échangées entre RAD2RAD et les calculs clients RADIOSS qui rappelons-le constituent les calculs *free*. La communication des masses et raideurs nodales ne sera pas représentée ici afin de ne pas surcharger le schéma. D'autre part, nous nous limiterons au cas de deux sous-domaines en contact ; Ω_1 désignant toujours le coté *maître* dans lequel le contact est effectivement résolu, et Ω_2 désignant le coté *esclave*.

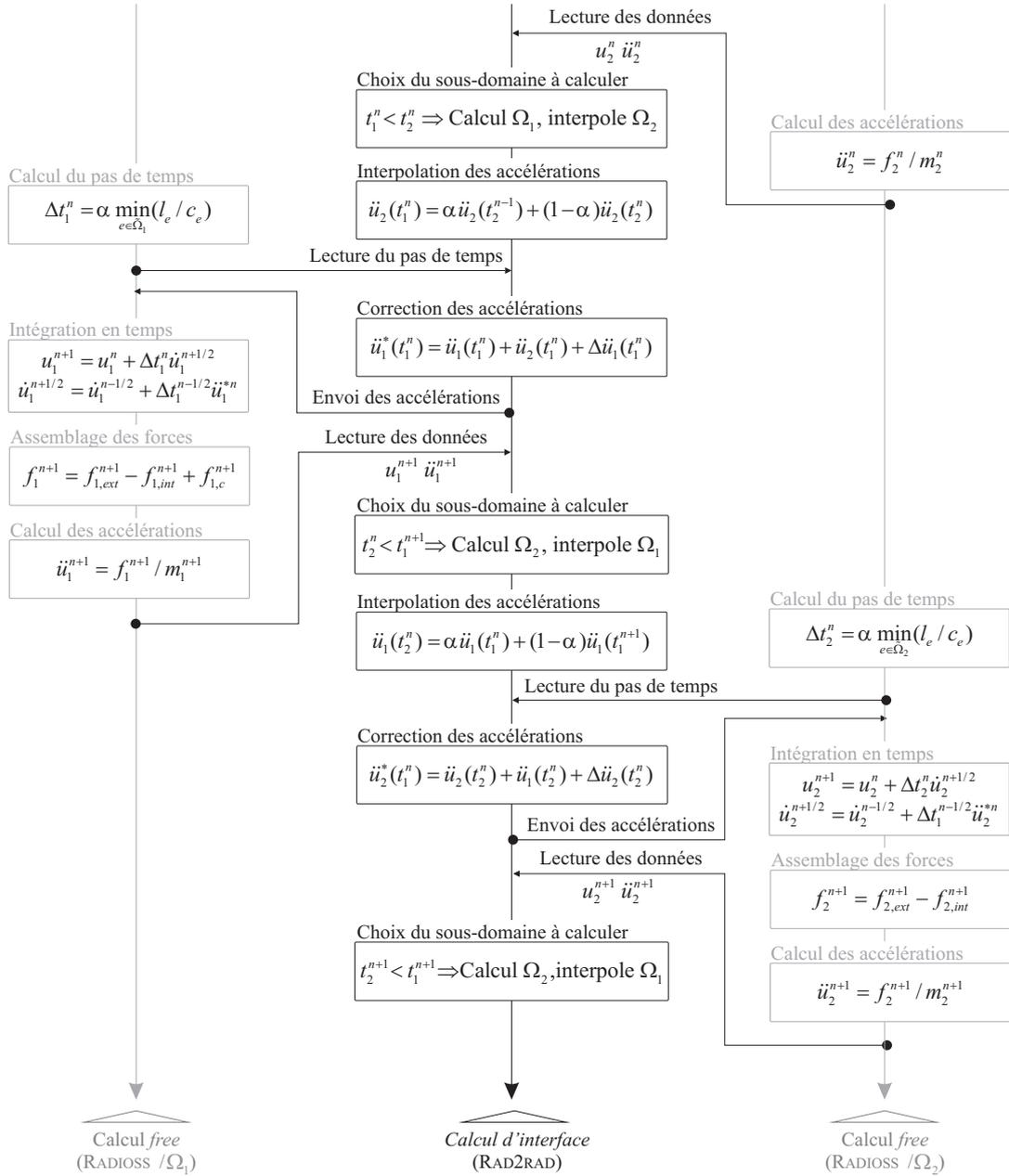


Figure 2.24: Algorithme général de calcul du contact inter-domaines.

2.4 Exemples numériques

Nous concluons ce second chapitre, en illustrant notre stratégie de calcul multi-domaines au travers quatre exemples d'applications. Le but est de valider la gestion du contact mise en oeuvre sur différents problèmes de contact inter-domaine. Les sous-structurations établies feront naturellement intervenir des interfaces de contacts potentiels, mais aussi d'éventuelles interfaces de liaisons parfaites évoquées dans le premier chapitre. Les résultats obtenus avec notre algorithme de couplage RAD2RAD-RADIOSS seront comparés aux résultats de calcul RADIOSS mono-domaine classique.

2.4.1 Chute d'une poutre

Le premier exemple a pour but de valider la méthode à travers la vérification des accélérations, vitesses et déplacements. Nous adopterons ici une approche mono-échelle où chaque sous-domaine évolue avec le même pas de temps (imposé par le domaine le plus pénalisant).

2.4.1.1 Description du cas test

On considère une POUTRE de section S , d'inertie I et de longueur l_{ptr} inclinée d'un angle $\alpha = 15^\circ$ par rapport à l'horizontale. Cette POUTRE soumise à une vitesse verticale initiale $v_0 = 10m/s$ vient impacter une PLAQUE carrée de coté l_{plq} et d'épaisseur e_{plq} encadrée sur son contour. En supposant un comportement élastique pour les deux objets, on leur attribue la même masse volumique ρ et le même module de Young E . Le tableau 2.3 résume la valeur des différents paramètres utilisés pour le calcul.

	Sous-domaine POUTRE	Sous-domaine PLAQUE
Paramètres géométriques	$l_{ptr} = 0.05 \text{ m}$ $S = 0.0063 \times 0.0063 \text{ m}$ $I = 8.5356 \times 10^{-12} \text{ m}^4$	$l_{plq} = 0.10 \text{ m}$ $e_{plq} = 9.14 \times 10^{-4} \text{ m}$
Paramètres matériaux	$\rho_{ptr} = 7.80 \times 10^3 \text{ kg/m}^3$ $E_{ptr} = 2.10 \times 10^{11} \text{ N/m}^2$	$\rho_{plq} = 7.80 \times 10^3 \text{ kg/m}^3$ $E_{plq} = 2.10 \times 10^{11} \text{ N/m}^2$

Tableau 2.3: Chute d'une poutre : paramètres de calcul.

Du point de vue modélisation, la POUTRE est composée d'un unique élément (BEAM) et la PLAQUE de 16×16 éléments coques à 4 noeuds (SHELL). Dans cet exemple assez simple, la sous-structuration est immédiate dans la mesure où POUTRE et PLAQUE constituent les deux sous-domaines à isoler. Afin de traiter le contact unilatéral (ici non frottant) entre ces deux entités, on choisit la PLAQUE comme sous-domaine *maître* et la POUTRE comme *esclave*.

Suite à la méthode mise en oeuvre les deux noeuds extrêmes de la POUTRE sont dupliqués dans le sous domaine PLAQUE (cf. Figure 2.25). En pratique, cela nécessite de définir dans ce sous-domaine PLAQUE une peau artificielle (sans masse ni raideur) qui correspond ici à la copie de la POUTRE. Dans le modèle de référence RADIOSS utilisé pour la comparaison des résultats, PLAQUE et POUTRE sont traitées dans un seul et même domaine avec une interface de contact classique.

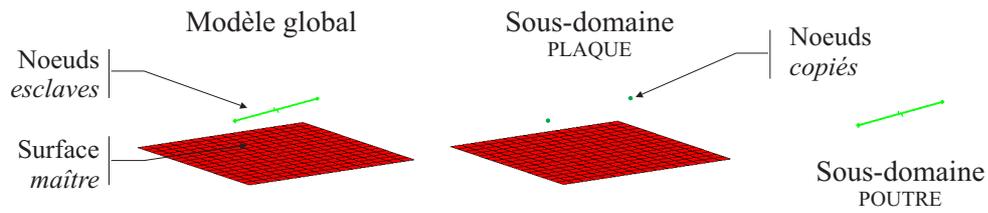


Figure 2.25: Chute d'une poutre : décomposition du domaine de calcul.

2.4.1.2 Étude cinématique

On s'intéresse ici à l'évolution des trois grandeurs cinématiques que sont les déplacements, les vitesses et les accélérations (cf. Figures 2.26-2.28). On peut tout d'abord vérifier que toutes ces grandeurs sont identiques de part et d'autre de l'interface, ce qui traduit le collage cinématique parfait des noeuds d'interface. D'autre part, on constate la parfaite concordance entre les résultats issus du calcul de référence et ceux issus du calcul multi-domaines. Cet exemple constitue une première validation de la méthode de gestion du contact entre sous-domaines dans le cas mono-échelle.

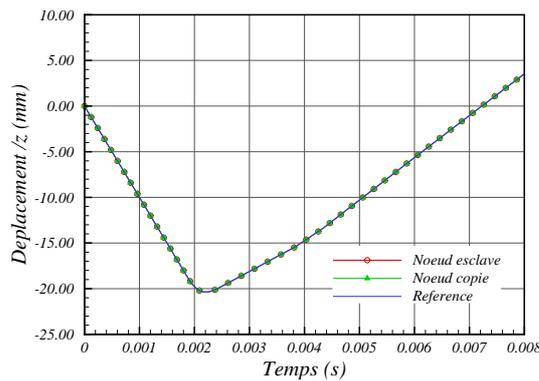


Figure 2.26: Évolution du déplacement vertical à l'extrémité de la poutre.

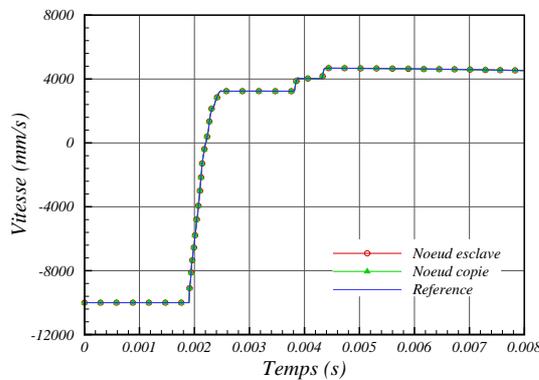


Figure 2.27: Évolution de la vitesse verticale à l'extrémité de la poutre.

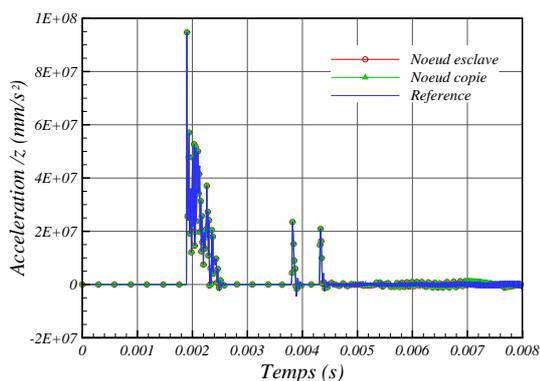


Figure 2.28: Évolution de l'accélération verticale à l'extrémité de la poutre.

Au niveau du temps de calcul l'approche mono-échelle ne présente aucun intérêt, les deux sous-domaines évoluant avec le même pas de temps. Au contraire, le temps CPU et le temps de communication passé dans le programme principal conduit globalement à un temps de calcul plus important que dans le cas mono-domaine classique.

2.4.2 Impact d'un corps creux

Le but de ce second cas test est de valider cette fois l'approche multi-échelles sur un problème légèrement plus complexe. Contrairement au cas précédent nous adopterons donc un pas de temps différent sur chacune des sous-structures issues de la décomposition.

2.4.2.1 Description du cas test

On considère ici un tube de hauteur h_t et de section S_t fermé dans sa partie inférieure venant impacter une plaque carré encastrée sur le bord. Pour cela, une vitesse $v_i = 10 \text{ m/s}$ est imposée aux noeuds supérieurs du corps creux tout au long du calcul. Comme dans le cas précédent nous adopterons un comportement élastique linéaire.

	Sous-domaine TUBE		Sous-domaine PLAQUE	
Paramètres géométriques	h_t $2.03 \times 10^{-4} \text{ m}$	S_t $0.0508 \times 0.0381 \text{ m}$	l_p 0.10 m	e_p $9.00 \times 10^{-4} \text{ m}$
Paramètres matériaux	ρ_t $7.80 \times 10^3 \text{ kg/m}^3$	E_t $2.10 \times 10^{11} \text{ N/m}^2$	ρ_p $7.80 \times 10^3 \text{ kg/m}^3$	E_p $2.10 \times 10^{11} \text{ N/m}^2$
Paramètres numériques	n_t^{el} 1088 elts	Δt_t^0 $7.89 \times 10^{-7} \text{ s}$	n_p^{el} 4096 elts	Δt_p^0 $2.71 \times 10^{-7} \text{ s}$

Tableau 2.4: Impact d'un corps creux : paramètres de calcul.

Le modèle global est toujours décomposé en deux sous-domaines, le TUBE constituant le sous-domaine *esclave* et la PLAQUE le sous-domaine *maître*. Un maillage en éléments coques à 4 noeuds (SHELL) est utilisé pour chaque partie. De part leurs tailles d'éléments différentes, on relève un facteur proche de 3 sur les pas de temps de stabilité initiaux de chaque modèle (cf. Tableau 2.4). Nous noterons enfin qu'une seule interface

(de contact unilatéral sans frottement) sera ici traité par RAD2RAD entre le fond du TUBE et la PLAQUE. Par conséquent, seuls les noeuds inférieurs du TUBE candidats au contact pourront être considérés comme *esclaves* et donc dupliqués dans le sous-domaine PLAQUE.

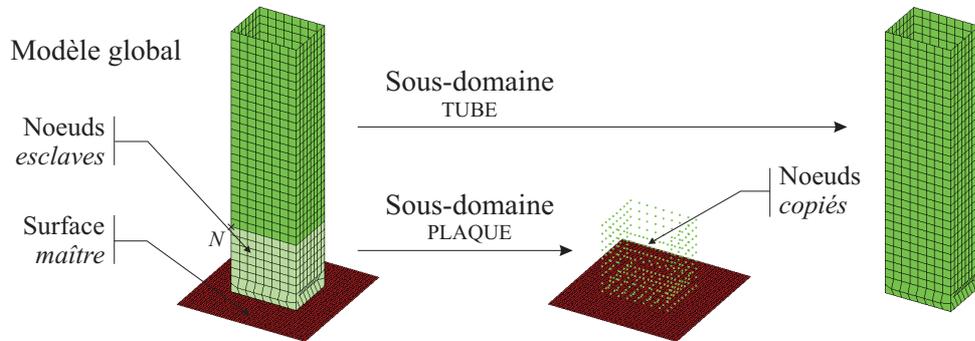


Figure 2.29: Impact d'un corps creux : décomposition du domaine de calcul.

2.4.2.2 Résultats du calcul

La Figure 2.30 illustre le flambage du TUBE. Les lobes ainsi créés nécessitent de définir un auto-contact sur ce sous-domaine. Malgré les fortes déformations subies par la structure, les noeuds *copiés* collent parfaitement au maillage du TUBE. Ceci confirme la continuité cinématique des noeuds *esclaves* dans le cas multi-échelles en temps.

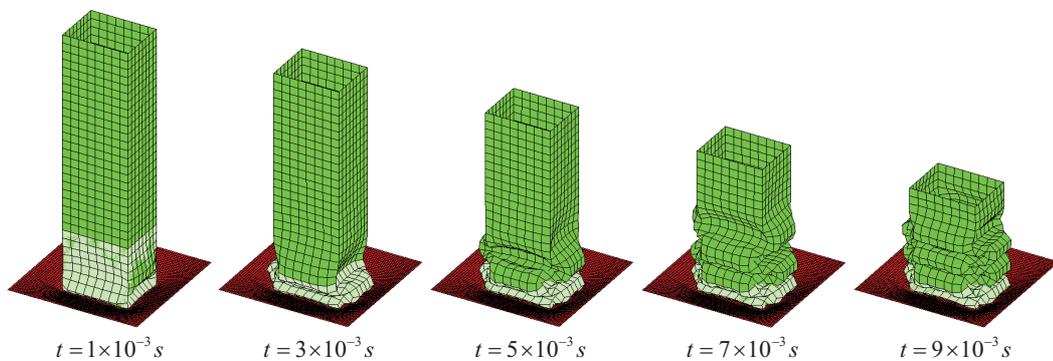


Figure 2.30: Écrasement du tube sur une plaque encastrée.

Les courbes d'effort normal (cf. Figure 2.32) et de la vitesse du noeud N (cf. Figure 2.31) montrent la très bonne qualité de la méthode proposée : en effet, on ne peut distinguer les résultats de l'analyse mono-domaine de ceux de l'analyse multi-domaines. D'autre part, le tracé des pas de temps de stabilité (cf. Figure 2.33) montre l'intérêt du calcul multi-domaines. Nous voyons parfaitement que le modèle de référence évolue avec un pas de temps correspondant au minimum des pas de temps de chaque sous-domaine, alors que dans l'analyse multi-domaines, chaque sous domaine est calculé de manière optimale. C'est la raison du gain en temps de calcul. Cependant, le modèle éléments finis étant de petite taille (5185 éléments), et le temps de calcul relativement petit (2460s), le gain

apporté par la méthode n'est pas très significatif pour cet exemple.

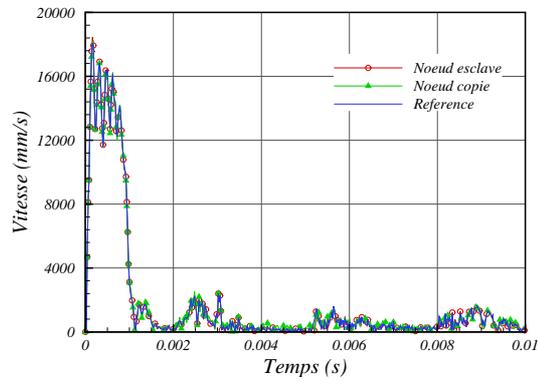


Figure 2.31: Évolution de la vitesse résultante au noeuds N .

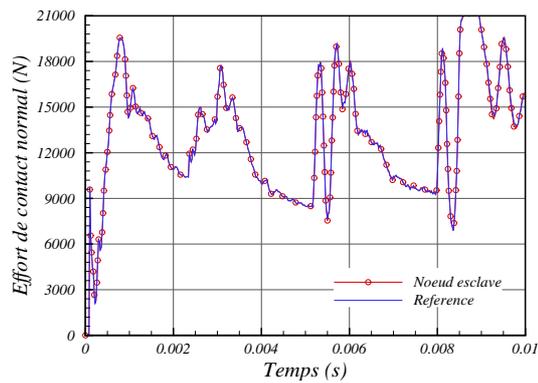


Figure 2.32: Évolution de l'effort de contact normal entre le TUBE et la PLAQUE.

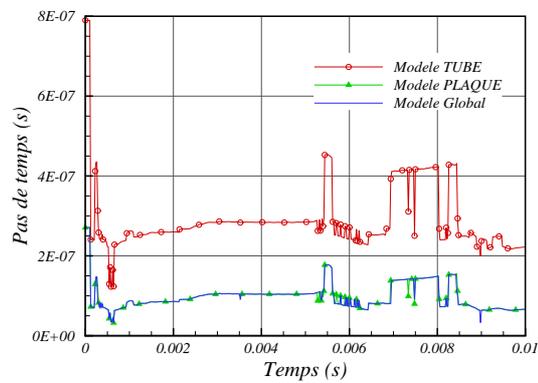


Figure 2.33: Évolution du pas de temps de stabilité de chaque modèle.

2.4.3 Simulation d'un choc piéton

Ce dernier exemple a pour but de tester et valider la méthode mise en oeuvre sur un cas industriel plus complexe. L'utilisation d'un modèle éléments finis de taille importante montrera l'intérêt des approches multi-échelles sur les temps de calcul.

2.4.3.1 Description du cas test

On s'intéresse ici à la simulation d'un choc tête à 35 km/h sur un véhicule projet PSA PEUGEOT-CITROËN®. L'intérêt de telles simulations est d'étudier le comportement de la structure ainsi que certains critères bio-mécaniques. La tête est modélisée par une sphère venant impacter le haut du capot, au niveau du bras d'essuie vitre (cf. Figure 2.34a). Dans ce modèle, le système d'essuie vitre maillé très finement dispose d'un pas de temps pénalisant fortement le reste du véhicule (cf. Tableau 2.5). Nous chercherons donc à isoler (cf. Figure 2.34b), par notre méthode de décomposition, ce sous-domaine essuie-vitre (PART1) du reste de la structure (PART2). Contrairement aux deux cas précédents, les sous-domaines ne sont plus ici liés par une seule interface mais par 8 interfaces dont 2 de contact unilatéral : l'interface « Cache-Enjoliveur » et l'interface « Bras-Grille ».

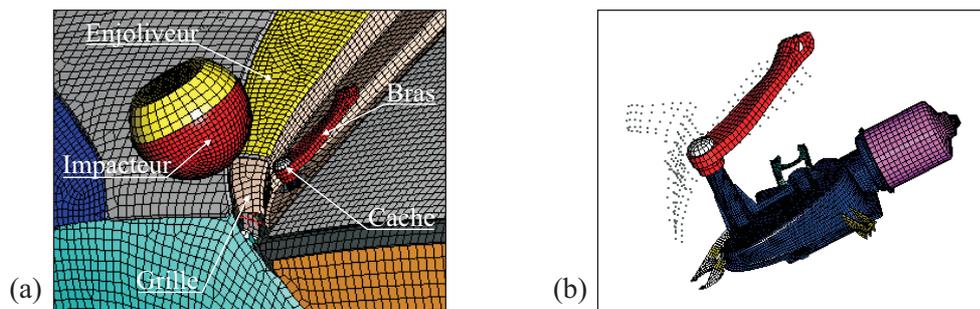


Figure 2.34: Simulation d'un choc piéton : (a) Modèle global de véhicule - (b) Sous-domaine essuie vitre maillé finement (PART1).

Le sous domaine essuie-vitre constituera ici la partie *maître* dans laquelle seront gérées les interfaces de contact. De ce fait, les noeuds de la grille et de l'enjoliveur seront partiellement reproduits dans ce sous-domaine (cf. Figure 2.34b).

	Modèle global	PART1	PART2
Nombre d'élément n^{el}	317451 elts	13803 elts	303648 elts
Pas de temps Δt^0	$0.339 \times 10^{-7} s$	$0.339 \times 10^{-7} s$	$0.834 \times 10^{-7} s$

Tableau 2.5: Comparaison des tailles et pas de temps initiaux des différents modèles.

2.4.3.2 Validation des résultats

Un des critères les plus importants d'une telle simulation concerne la décélération calculée au centre de l'impacteur tête. On peut voir sur la figure 2.35 que de ce point de vue, les résultats obtenus avec les méthodes mono-domaines et multi-domaines correspondent

parfaitement. Ces courbes d'accélération permettent de calculer le critère HIC (Head Injury Criterion) très utilisé dans le domaine de la biomécanique [SEKS99]. Ce critère est basé sur la valeur moyenne de l'accélération dans la zone de décélération la plus critique :

$$HIC = \max_{t^1, t^2} \left\{ \left[\frac{1}{t^2 - t^1} \int_{t^1}^{t^2} \ddot{u}(t) dt \right]^{2.5} (t_2 - t_1) \right\} \quad (2.4.1)$$

où t^1 et t^2 sont deux instants arbitraires du pic d'accélération. L'écart des HIC alors calculés à partir des deux méthodes est inférieur à 0.6%.

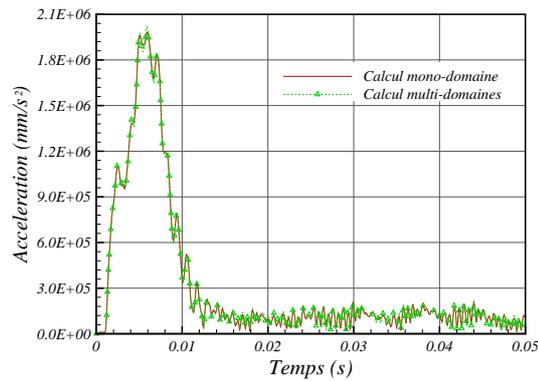


Figure 2.35: Comparaison des accélérations au centre de l'impacteur tête.

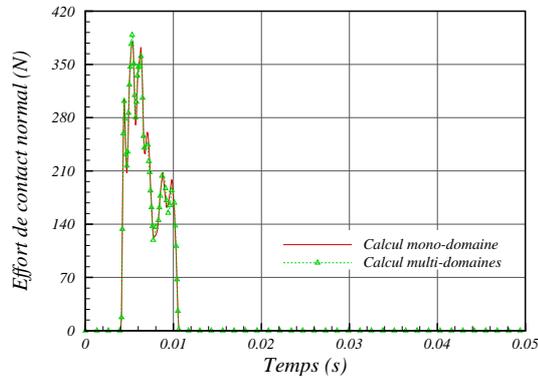


Figure 2.36: Comparaison des efforts de contact à l'interface « Bras-Grille ».

Enfin, nous pouvons nous intéresser à l'effort de contact calculé au niveau des interfaces de contact unilatéral entre sous-domaines et notamment au niveau de l'interface « Bras-Grille » dans laquelle transite le plus d'énergie. On voit sur la figure 2.36 que cette interface de contact n'est active que durant une courte période ($\approx 6 \times 10^{-3} s$). Comme précédemment, la courbe obtenue avec la méthode multi-domaines est parfaitement corrélée avec la référence mono-domaine. L'écart moyen mesuré entre les deux méthodes est alors de l'ordre de 3%.

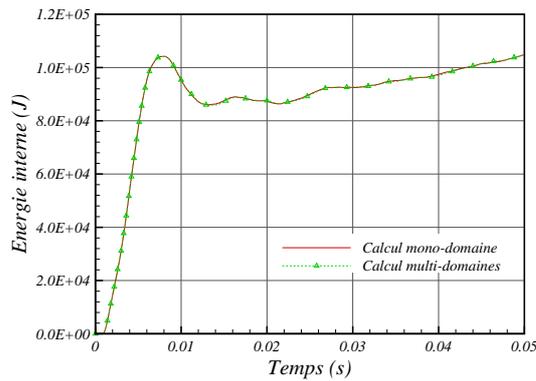


Figure 2.37: Comparaison des énergies internes.

2.4.3.3 Gain en temps CPU

Le principal objectif de cette simulation est d'apprécier le gain en temps CPU que peut apporter notre méthode multi-domaines. Les calculs ont été réalisés sur une machine mono-processeur cadencée à $3.75GHz$. Un traitement du pas de temps est effectué sur chaque modèle afin de limiter les temps de calculs. Les temps CPU utilisés pour chaque modèle sont résumés dans le tableau 2.6. Comme nous pouvions nous y attendre pour le modèle multi-domaines, la majeure partie du temps de calcul est passée sur la PART2 qui constitue la principale partie du modèle. La méthode mise en oeuvre apporte un gain substantiel en temps de calcul de 2.2 par rapport au modèle mono-domaine de référence. Ce gain prend en compte le temps passé dans le programme principal RAD2RAD (5% du temps de calcul total).

Calcul mono-domaine	Calcul multi-domaines			
	PART1	PART2	RAD2RAD	Total
391390 s	19964 s	148567 s	9257 s	177788 s

Tableau 2.6: Comparaison des temps CPU.

Chapitre 3

Calculs dynamiques à discrétisation variable

Sommaire

3.1	État de l’art des approches multi-échelles	103
3.1.1	L’approche multi-grilles	103
3.1.2	La méthode <i>Arlequin</i>	106
3.1.3	L’approche adaptative	109
3.2	Méthode de basculement de maillage en linéaire	114
3.2.1	Principe du basculement de maillage	114
3.2.2	Opérateurs de changement d’échelles	115
3.2.3	Insuffisance de la projection	119
3.2.4	Méthode de rééquilibrage dynamique	122
3.3	Extension au cas non-linéaire	125
3.3.1	Méthode de rééquilibrage itérative	125
3.3.2	Critère de changement de maillage	127
3.3.3	Implémentation de la méthode	128
3.4	Exemples numériques	133
3.4.1	Cas 2D : Poutre en flexion	133
3.4.2	Cas 3D : Flambage de prolonge	139

L_A physique, et la mécanique en particulier, sont régies par des lois et des modèles dont la validité dépend généralement de l’échelle d’observation. La description d’un système matériel sera par exemple très différente suivant que l’on se place à l’échelle moléculaire (milieu hétérogène), ou à une échelle plus macroscopique (milieu continu). Par conséquent les hypothèses doivent être correctement posées et les modélisations judicieusement choisies, afin de décrire le plus fidèlement possible le phénomène observé. Évidemment la modélisation et l’analyse numérique cherchent de surcroît à analyser conjointement les différentes échelles mises en jeu. Cela représente une complexité de calcul considérable qu’interdisent, encore souvent, les capacités des calculateurs actuels. De cette façon, depuis une quarantaine d’années, de nombreuses méthodes ont été proposées pour tenir compte de ces phénomènes aux échelles multiples dans les modélisations.

Cette tendance est déjà très perceptible en calcul des structures complexes sollicitées par des impacts localisés, où le comportement peut varier fortement entre le voisinage des zones chargées et les parties plus éloignées. La simulation numérique d'un crash frontal ne nécessitera par exemple pas la même finesse d'analyse à l'avant qu'à l'arrière du véhicule. C'est pourquoi il est souvent nécessaire, sur le plan de la modélisation et du calcul numérique d'enrichir localement le modèle global, en introduisant par exemple une altération locale (fissure excavation, raidisseur...), ou en enrichissant la discrétisation de la zone d'intérêt. Les méthodes standards permettent naturellement d'adapter la taille des éléments aux différentes parties du modèle mais cela se fait souvent au prix de difficultés de maillage et de gradients de tailles d'éléments qui généralement nuisent à la précision des calculs. De plus, en dynamique explicite le calcul reste fortement pénalisé par l'emploi du pas de temps imposé par les petits éléments utilisés à proximité de la zone d'intérêt. Le manque de flexibilité des approches classiques ne leur permet donc pas de réaliser ces tâches sans occasionner des modèles lourds et des coûts de calculs prohibitifs.

Nous entendrons ici par multi-échelles, les méthodes d'analyses accédant à différents niveaux de phénomènes physiques en jouant sur la précision des discrétisations spatiales et temporelles. Au sens large, ces méthodes englobent les méthodes de résolution par sous-domaines déjà largement évoquées dans ce mémoire. Notre méthode de décomposition présentée au chapitre 1, couplant des sous-domaines incompatibles en espace et en temps, s'apparente incontestablement à une méthode multi-échelles. Toutefois la problématique qui motive ce dernier volet de la thèse se pose en tout autre terme.

Toujours dans le souci de réduire les temps CPU des calculs dynamiques, il est en effet intéressant de pouvoir adapter la discrétisation à l'évolution temporelle. Pour de nombreuses applications, la finesse d'analyse d'une zone précise peut n'être requise que pour une durée limitée. Si l'on revient sur l'exemple des simulations numériques de crashes, le scénario de choc montre que les zones critiques du véhicule sont sollicitées à des instants différents et de manière ponctuelle. Ces problèmes élasto-plastiques en grandes déformations affichent en effet de fortes non-linéarités qui idéalement nécessiteraient de fréquents remaillages au cours de la résolution. Cette solution très coûteuse pour les géométries complexes telles que les modèles de véhicule, est par ailleurs, rendue particulièrement instable par l'utilisation d'un schéma explicite. Le but est donc ici de proposer une méthode d'adaptation maillage s'inscrivant dans notre approche de calculs multi-domaines pour les problèmes de dynamique explicite. L'idée est d'enrichir localement et temporairement la discrétisation d'un ou plusieurs sous-domaines au cours d'une phase sensible du calcul.

Nous entamerons ce dernier chapitre en dressant un bref état des lieux des méthodes multi-échelles existantes à ce jour. Nous développerons ensuite la méthode dite de « basculement de maillage » mise en oeuvre pour les problèmes dynamiques. Le cas linéaire sera tout d'abord étudié ; nous évoquerons notamment les problèmes de stabilité causés par l'intégration explicite après le changement de discrétisation et proposerons une technique d'équilibrage y remédiant. La troisième partie sera consacrée à l'extension de la méthode aux problèmes non-linéaires ; nous reviendrons sur la procédure d'équilibrage (alors itérative) mise en oeuvre pour assurer la reprise des calculs sur la nouvelle discrétisation. Un critère de raffinement indispensable à l'automatisation du basculement de maillage sera alors introduit. Nous terminerons enfin par illustrer et valider cette approche à travers deux exemples d'applications réalisés sous le code de calcul CASTEM.

3.1 État de l'art des approches multi-échelles

Afin de positionner la méthode mise en oeuvre au cours de cette thèse, nous présentons quelques développements récents de ces techniques de calcul dites multi-échelles. Le but n'est évidemment pas de faire un tour d'horizon exhaustif de la littérature sur ce sujet, mais simplement de donner quelques éclairages sur certaines approches nous apparaissant comme révélatrices des principales directions de recherche suivies.

Les différentes méthodes que nous évoquerons ici cherchent à intégrer, dans les modélisations grossières, des effets provenant des échelles inférieures. Elles sont pour la plupart à considérer dans une démarche globale, où les zones mal représentées de la modélisation grossière sont localisées puis corrigées par des techniques de raffinements ou de zooms variés.

Parmi ces techniques, les plus classiques consistent soit à augmenter le degré de l'interpolation, soit à diminuer localement la taille des mailles. Ces mécanismes à la base des puissantes méthodes de raffinements adaptatifs souffrent généralement d'un manque de flexibilité et occasionnent des calculs assez coûteux. Nous traiterons toutefois ici les cas particuliers des éléments finis hiérarchiques [Zhu86] et de raffinement automatique local [CGLC05b]. Une seconde catégorie de méthodes, concerne l'approche multi-grilles, très présente dans la littérature et par laquelle nous entamerons cet état de l'art [PH90, KP93]. Ceci étant, qu'il s'agisse d'approche adaptative ou multi-grilles la plupart de ces méthodes s'inscrivent dans une stratégie de calcul mono-modèle. Nous reviendrons alors sur la méthode *Arlequin* [Rat03, BDR05, BD05] qui rompt avec les précédentes méthodes par l'usage d'un principe de superposition d'un maillage local au maillage global.

Enfin nous remarquerons que si la plupart de ces méthodes sont bien avancées en régime statique, elles le sont moins en dynamique. Les seules approches développées en dynamique sont à notre connaissance des méthodes itératives basées sur l'intégration implicite [GC03, CGLC05c].

3.1.1 L'approche multi-grilles

Dans la littérature, différentes approches ont été inspirées des méthodes multi-grilles initiées par Brandt [Bra77] dans les années 70. Ce dernier s'intéressait alors à la résolution des systèmes linéaires résultant de la discrétisation des équations aux dérivées partielles elliptiques telle que l'équation de diffusion. Très vite ces méthodes ont été étendues à d'autres domaines et notamment en mécanique des fluides. Elles restent toutefois peu répandues en mécanique des structures. Parmi les applications plus récentes au calcul de structures, citons par exemple [PH90, KP93] qui ont permis d'adapter ces méthodes à la méthode des éléments finis pour des problèmes de structure linéaire et non-linéaire. On propose ici de décrire brièvement les grandes lignes de cette méthode dans le cas de deux grilles, l'une fine, l'autre grossière.

3.1.1.1 Procédure multi-grille classique

Les méthodes multi-grilles apparaissent comme des techniques de résolution de systèmes linéaires à part entière. Nous considérerons donc dans un premier temps le problème

d'élasticité linéaire suivant :

$$K_f u_f = f_f \quad (3.1.1)$$

où l'indice f réfère aux grandeurs associées au maillage fin. Le principe de cette approche repose sur le fait que les méthodes itératives classiques (gradient conjugué, Gauss-Seidel, Jacobi, relaxation, ...) sont plus efficaces pour réduire les résidus à faibles longueurs de variation qu'à grandes longueurs de variation. Cela se traduit par une rapide décroissance de l'erreur lors des premières itérations puis une stagnation de celle-ci, ou du moins un taux de convergence diminué. L'idée est donc de traiter séparément les hautes et basses fréquences du problème en décomposant chaque itération k de la manière suivante :

- ⇨ ① **Première relaxation** (Lissage) : Quelques itérations sont effectuées sur l'approximation initiale de la solution fine $u_f^{(k)}$ de manière à obtenir une nouvelle approximation $u_f^{(k+1/2)}$ intermédiaire. Cette relaxation peut être réalisée par une méthode itérative classique telle que la méthode du gradient conjugué.

$$u_f^{(k)} \rightarrow \begin{cases} u_f^{(k+1/2)} = u_f^{(k)} + \Delta u_f^{(k+1/2)} \\ r_f^{(k+1/2)} = f_f - K_f u_f^{(k+1/2)} \end{cases} \quad (3.1.2)$$

où $r_f^{(k+1/2)}$ est le résidu associé à l'échelle fine. L'erreur associée à $u_f^{(k)}$ étant dominée par les basses fréquences, la correction à apporter peut être calculée sur la grille grossière.

- ⇨ ② **Correction grossière** : La correction $\Delta u_g^{(k)}$ est calculée en résolvant l'équation suivante :

$$K_g \Delta u_g^{(k)} = r_g^{(k)} \quad (3.1.3)$$

où, K_g est la matrice de rigidité du maillage grossier et où le résidu $r_g^{(k)}$ est la restriction au maillage grossier du résidu $r_f^{(k)}$. Le système (3.1.3) peut alors être résolu par une méthode de résolution directe. La nouvelle approximation de la solution de (3.1.1), noté $u_f^{(k+1)}$ peut être mise à jour par :

$$u_f^{(k+1)} = u_f^{(k+1/2)} + \Delta u_f^{(k)} \quad (3.1.4)$$

où $\Delta u_f^{(k)}$ est la prolongation au maillage fin de $\Delta u_g^{(k)}$.

- ⇨ ③ **Deuxième relaxation** : Si nécessaire, une seconde étape de relaxation (3 ou 4 itérations de gradient conjugué) est introduite afin d'éliminer les hautes fréquences résultant de l'étape de prolongation.

$$u_f^{(k+1)} \rightarrow \begin{cases} u_f^{(k+1)} = u_f^{(k+1)} + \Delta u_f^{(k+1)} \\ r_f^{(k+1)} = f_f - K_f u_f^{(k+1)} \end{cases} \quad (3.1.5)$$

Comme pour la première étape, cette relaxation est calculée sur le maillage fin en utilisant une méthode itérative telle que la méthode du gradient conjugué.

Les étapes ①-③ peuvent ainsi être répétées jusqu'à ce que la solution ait convergé. Le principal intérêt de cette technique est de déterminer les composantes basses fréquences

de la solution en appliquant une méthode de résolution directe sur la grille grossière. Toutefois, si cette résolution directe s'avère trop coûteuse, on peut introduire une grille encore plus grossière de manière à dérouler le processus précédent pour la résolution de (3.1.3).

Remarque 1 : Pour les problèmes non-linéaire, la méthode multi-grilles précédente est incorporée à une procédure itérative de Newton-Raphson. Elle est alors utilisée pour résoudre le système linéaire rencontré à chaque itération de Newton.

3.1.1.2 Adaptation aux calculs multi-domaines

La méthode précédente a récemment été étendue aux calculs multi-domaines pour les problèmes dynamiques non-linéaires [GC03]. L'intérêt est de pouvoir introduire la procédure multi-grilles dans les zones fortement sollicitées de la structure, par le biais d'une méthode de décomposition. Le but de cette approche, basée sur un schéma d'intégration implicite, est d'améliorer la convergence du calcul et donc de réduire de manière significative le nombre d'itérations effectuées à chaque pas de temps. On se propose ici de présenter brièvement ces travaux en considérant deux sous-domaines dont l'un comprend deux grilles de calcul (cf. Figure 3.1).

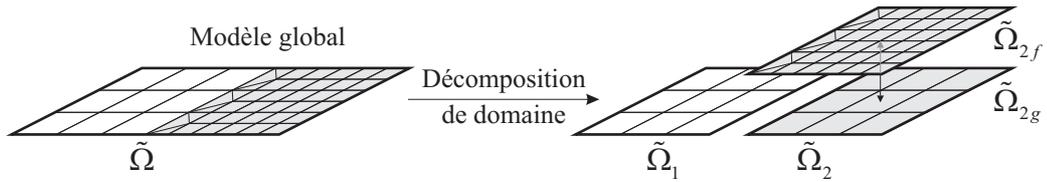


Figure 3.1: Principe de l'approche multi-grilles par sous-domaine.

L'approche repose sur une décomposition de type *Schur duale* dont la formulation variationnelle globale à l'instant t^n s'écrit :

$$\begin{aligned} \delta\mathcal{W} = & {}^t\delta u_1^n (M_1^n \ddot{u}_1^n + K_1^n u_1^n - f_{1,ext}^n - {}^tC_1^m \lambda^n) \\ & + {}^t\delta u_2^n (M_2^n \ddot{u}_2^n + K_2^n u_2^n - f_{2,ext}^n - {}^tC_2^m \lambda^n) \\ & + {}^t\delta \lambda^n (C_1^n \dot{u}_1^n + C_2^n \dot{u}_2^n) = 0, \quad \forall \delta u_1^n \in \mathcal{U}_{0,1}^n, \delta u_2^n \in \mathcal{U}_{0,2}^n, \delta \lambda^n \in \mathcal{L}^n \end{aligned} \quad (3.1.6)$$

Nous remarquerons que dans cette approche, chaque sous-domaine et chaque grille dispose de sa propre échelle de temps. Par conséquent, toutes les quantités intervenant dans (3.1.6) n'étant pas calculées à l'instant considéré, on procède à des interpolations temporelles comme celles décrites dans le paragraphe 1.2.4.

Revenons alors sur l'idée sous-jacente des méthodes multi-grilles qui est de considérer la solution du problème comme la somme de deux contributions :

$$u_f = \bar{u}_g + \tilde{u}_f \quad \text{avec} \quad \bar{u}_g = \mathcal{P}u_g \quad (3.1.7)$$

où \bar{u}_g est la projection de la solution grossière sur le maillage fin, et \mathcal{P} est l'opérateur de projection. En injectant dans (3.1.6) cette décomposition sur le second sous-domaine, on

aboutit finalement à un système de type :

$$M_1^n \ddot{u}_1^n + K_1^n u_1^n - f_{1,ext}^n = C_1^{nt} \lambda^n \quad (3.1.8)$$

$$M_{2f}^n \ddot{\tilde{u}}_{2f}^n + K_{2f}^n \tilde{u}_{2f}^n - f_{2f,ext}^n = C_{2f}^{nt} \lambda^n - M_{2f}^n \ddot{\bar{u}}_{2g}^n - K_{2f}^n \bar{u}_{2g}^n \quad (3.1.9)$$

$$M_{2g}^n \ddot{u}_{2g}^n + K_{2g}^n u_{2g}^n - f_{2g,ext}^n = C_{2g}^{nt} \lambda^n - \mathcal{P}^t M_{2f}^n \ddot{\tilde{u}}_{2f}^n - \mathcal{P}^t K_{2f}^n u_{2f}^n \quad (3.1.10)$$

$$C_1^n \dot{u}_1^n + C_{2g}^n \dot{u}_{2g}^n + C_{2f}^n \dot{u}_{2f}^n = 0 \quad (3.1.11)$$

Ce système est résolu de manière similaire au problème (1.2.22) décrit au premier chapitre. La description détaillée de l'algorithme est toutefois faite dans [GC03]. La généralisation au cas non-linéaire se fait assez naturellement en utilisant pour les étapes ① et ② de l'approche multi-grilles un algorithme de relaxation non-linéaire tel que celui du gradient-conjugué préconditionné non-linéaire. Le principal avantage de cette approche est qu'elle traite à la fois la non-linéarité et l'approche multi-grilles dans une seule et même boucle itérative. Par conséquent, les itérations peuvent être réalisées sur les intervalles de temps de l'échelle grossière et non nécessairement sur les instants intermédiaires de l'échelle fine.

3.1.2 La méthode *Arlequin*

Toujours dans cette problématique qui est de traiter des problèmes mécaniques mettant en jeu des phénomènes locaux, généralement très différents, nous nous attarderons quelque peu sur une méthode assez récente appelée méthode *Arlequin* [Rat03, BDR05, BD05]. Bien qu'il s'agisse comme précédemment d'une méthode d'analyse à plusieurs échelles, cette approche de type local-global affiche un concept toutefois très différent.

Le principal intérêt de la méthode est d'apporter une grande flexibilité en couplant des modèles pouvant être de diverses natures. Elle se base pour cela sur le principe de superposition d'un maillage local au maillage global avec le souci d'introduire la modélisation la plus adaptée. Une de ces particularités est alors de raccorder les modèles, non pas sur une frontière (comme cela se fait classiquement), mais dans le volume. Le recouvrement peut alors être total ou partiel et les interfaces entre modèle géométriquement incompatibles.

3.1.2.1 Principes de la méthode

La méthode *Arlequin* apparaît comme une méthode de décomposition particulière, où les modèles ne sont pas simplement ajoutés mais superposés et collés les uns aux autres. De ce point de vue, elle s'apparente fortement à la méthode de recouvrement de *Schwarz* décrite précédemment (cf. § 1.2.1.1). Plus précisément la méthode *Arlequin* s'appuie sur les trois idées suivantes :

- ↷ Superposition de deux modèles avec duplication des états mécaniques dans la zone de recouvrement (zone d'intérêt).
- ↷ Répartition des énergies (travaux virtuels associés aux différentes forces) entre ces deux modèles, à l'aide de fonctions de pondération.
- ↷ Collage des deux modèles sur la zone de superposition, avec un opérateur de jonction.

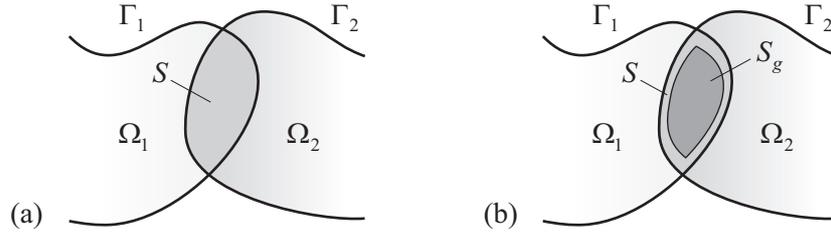


Figure 3.2: Méthode *Arlequin* : (a) Zone de superposition S - (b) Zone de collage S_g des modèles.

Superposition de deux modèles Pour former la superposition, nous considérons deux modèles définis sur deux ouverts connexes Ω_1 et Ω_2 . Ces ouverts forment une partition, avec recouvrement, du domaine d'étude Ω . La région où les ouverts se recouvrent joue un rôle essentiel (cf. Figure 3.2a). Nous la noterons S :

$$S = \Omega_1 \cap \Omega_2 \quad (3.1.12)$$

Nous supposons que sa mesure est non nulle. De cette façon, coexistent dans cette zone, deux états mécaniques indépendants, qui doivent être égaux si les deux modèles superposés sont rigoureusement identiques.

Répartition des énergies Pour ne pas compter deux fois l'énergie du système global dans la zone de recouvrement, le travail virtuel associé à chaque modèle est pondéré par des fonctions. Ces fonctions de pondération qui forment une partition de l'unité sur l'ensemble du domaine d'étude peuvent être définies comme suit :

$$\alpha_1(u_1) : \Omega_1 \rightarrow [0, 1] \quad \text{et} \quad \alpha_2(u_2) : \Omega_2 \rightarrow [0, 1] \quad (3.1.13)$$

tel que,

$$\begin{cases} \alpha_1 = 1 & \text{sur } \Omega_1 \setminus \Omega_2 \\ \alpha_2 = 1 & \text{sur } \Omega_2 \setminus \Omega_1 \\ \alpha_1 + \alpha_2 = 1 & \text{sur } \Omega_1 \cap \Omega_2 \end{cases} \quad (3.1.14)$$

Ces fonctions représentent un ingrédient important de la méthode, puisqu'elles définissent le niveau de mélange entre les deux modèles dans la zone de recouvrement. De plus, suivant leur valeur, elles précisent quelle part de confiance et de pertinence est accordée à chacun.

Collage sur la zone de recouvrement Pour raccorder les deux modèles, l'idée est d'activer des forces fictives de couplage contrôlant, sur la zone de recouvrement, l'écart entre leurs états mécaniques : déplacements, déformations, contraintes... La méthode présente alors plusieurs variantes en fonction de la formulation choisie pour le champ de forces fictives. Ce champ peut soit suivre un modèle rigide (approche Lagrangienne), soit un modèle élastique (pénalisation), soit encore une combinaison des deux (pénalisation-dualité).

Pour accorder une certaine liberté aux deux modèles sur la zone de superposition S , les conditions de raccord peuvent être seulement écrites sur une partie de cette dernière. La sous-zone correspondante, dite zone de collage, est alors notée S_g (cf. Figure 3.2b).

3.1.2.2 Formulation *Arlequin*

L'approche utilisant les multiplicateurs de Lagrange étant la plus courante, nous considérons ici le modèle de jonction rigide. Dans cette approche, le champ de force de collage est un champ de multiplicateur de Lagrange appartenant à l'espace dual des déplacements cinématiquement admissible restreint à S_g . Cela mène à un opérateur de couplage (ou opérateur de jonction) C basé sur un crochet de dualité entre l'espace de Sobolev $H^1(S_g)$ et son espace dual. En pratique ce crochet de dualité est généralement remplacé par un simple produit scalaire de l'espace $H^1(S_g)$ et l'opérateur de couplage s'écrit (3.1.17).

On considère ici un problème statique, avec une loi de comportement élastique linéaire. Sous ces hypothèses, la formulation *Arlequin* revient à déterminer les champs de déplacement u_1 et u_2 , cinématiquement admissibles, et le champ de force fictive λ , rendant stationnaire la fonctionnelle :

$$\mathcal{W}_{Arlequin} = \mathcal{W}_1(u_1) + \mathcal{W}_2(u_2) + C(\lambda, u_1 - u_2) \quad (3.1.15)$$

$$\text{Avec,} \quad \mathcal{W}_i(u_i) = \frac{1}{2} \int_{\Omega_i} \alpha_i \sigma(u_i) : \varepsilon(u_i) d\Omega - \int_{\Omega_i} \beta_i f \cdot u_i d\Omega \quad (3.1.16)$$

$$C(\lambda, u) = \frac{1}{2} \int_{S_g} \lambda \cdot u + l^2 \varepsilon(\lambda) : \varepsilon(u) d\Omega \quad (3.1.17)$$

où f est le chargement externe, $\varepsilon(u)$ et $\sigma(u)$ les tenseurs de déformation et de contrainte associés au champ de déplacement u , α et β les deux fonctions de pondérations respectivement associées aux forces internes et aux forces externes et l un paramètre strictement positif homogène à une longueur. L'influence de la répartition des énergies entre les états mécaniques, dans la zone de superposition, a été analysée. Il a entre autre été montré que la solution *Arlequin* était bornée et qu'elle tendait vers la solution u_i du problème mono-modèle sur Ω_i lorsque α_i et β_i tendaient vers 1.

3.1.2.3 Formulation discrète

La formulation discrète dérive de la formulation continue par l'application de la méthode des éléments finis. Au même titre que les deux domaines Ω_1 et Ω_2 , la zone de collage dispose elle aussi de sa propre discrétisation \tilde{S}_g . En notant ϕ_{1i} , ϕ_{2j} et ϕ_{gk} , les fonctions de bases sur les éléments de $\tilde{\Omega}_1$, $\tilde{\Omega}_2$ et \tilde{S}_g , le problème discret devient :

$$\begin{bmatrix} \mathcal{K}_1 & 0 & \mathcal{C}_1^t \\ 0 & \mathcal{K}_2 & -\mathcal{C}_2^t \\ \mathcal{C}_1^t & -\mathcal{C}_2^t & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ 0 \end{Bmatrix} \quad (3.1.18)$$

où \mathcal{K}_k , \mathcal{F}_k et \mathcal{C}_k désigne les variantes pondérées des matrices de raideurs, des vecteurs chargements et des matrices de couplages relative au domaine Ω_k . Ils sont définis par :

$$[\mathcal{K}_k]_{ij} = \int_{\Omega_k} \alpha_k \sigma(\varphi_{kj}) : \sigma(\varphi_{ki}) d\Omega \quad (3.1.19)$$

$$[\mathcal{F}_k]_{ij} = \int_{\Omega_k} \beta_k f \cdot \varphi_{ki} d\Omega \quad (3.1.20)$$

$$[\mathcal{C}_k]_{ij} = \int_{S_g} \varphi_{gi} \cdot \varphi_{kj} + l^2 \varepsilon(\varphi_{gi}) : \varepsilon(\varphi_{kj}) d\Omega \quad (3.1.21)$$

Nous remarquerons ici les similitudes du système linéaire (3.1.18) avec la formulation de *Schur duale* classique (1.2.14). La particularité réside néanmoins ici dans le recouvrement des domaines et la répartition des énergies introduites par les coefficients de pondération. Cette particularité confère à la méthode une difficulté supplémentaire dans la mesure où les discrétisations sont généralement incompatibles sur la zone de collage. Les intégrales (3.1.19-3.1.21) sont de ce fait hétérogènes au niveau des éléments et les formules d'intégration numérique généralement utilisées dans la méthode éléments finis classique ne sont plus valables. Une technique d'intégration plus appropriée est alors présentée dans [BDR05].

Coté validation, les résultats numériques semblent encourageants. L'une des applications phare pour laquelle cette méthode s'est avérée très efficace est la propagation flexible de défauts, tels que des fissures, sur des modèles numériques macroscopiques existants. De nombreux développements ont aussi portés sur les problèmes de contact-impact entre structures hétérogènes en introduisant de nouvelles échelles d'espaces et de temps. Enfin un aspect important de la méthode *Arlequin* est qu'en régime dynamique, elle permet aux ondes de transiter entre les différents modèles, sans piéger l'énergie au niveau des échelles fines, par réflexions parasites.

3.1.3 L'approche adaptative

La dernière approche que nous aborderons dans ce bref état de l'art des méthodes multi-échelles concerne les méthodes adaptatives. L'idée sous-jacente de cette approche très répandue est de contrôler l'erreur numérique au cours du calcul et d'enrichir la discrétisation dans les zones où elle est la plus importante. Parmi ces techniques de raffinement adaptatives, les plus répandues sont incontestablement les méthodes dites *h*-méthode et *p*-méthode :

Les *h*-méthodes Ces techniques sont sans doute les plus célèbres dans la mesure où elles sont utilisées dans de nombreux codes de calcul éléments finis. Elles utilisent généralement des éléments très simples avec des fonctions d'approximation linéaire.

Les *p*-méthodes Ces techniques consistent à augmenter l'ordre des polynômes d'approximation utilisé sur chaque élément mais en gardant cette fois le même maillage tout au long du calcul. Si tous les éléments ont le même degré, on parle de *p*-méthode uniforme ; dans le cas contraire, on parle de *p*-méthode adaptative.

A noter que la combinaison de ces deux approches forme ce qu'on appelle les *hp*-méthodes. Nous présenterons ici deux méthodes adaptatives particulières ; la première, inspirée des *p*-méthodes concerne les éléments finis hiérarchiques. La seconde, inspirée des *h*-méthodes, traite d'un raffinement local automatique original.

3.1.3.1 Éléments finis hiérarchiques

Les éléments finis hiérarchiques ont pour la première fois été proposés par Zhu [Zhu86] il y a une vingtaine d'années. Il s'agit d'une amélioration des *p*-méthodes dans le sens où ils enrichissent les fonctions d'approximation des éléments. Prenons par exemple l'approximation du champ déplacement u sur un élément isolé :

$$u(\xi) = \sum_{i=1}^p N_i(\xi) d_i \tag{3.1.22}$$

Cette approximation est dite hiérarchique si l'ajout d'un noeud (passage de p à $p + 1$) n'affecte pas les fonctions de forme N_i ($i = 1$ à p). Pour illustrer ce concept, plaçons-nous dans le cas d'un élément hiérarchique unidimensionnel ($p = 2$). Ces fonctions de forme s'écrivent alors :

$$N_1 = \frac{1}{2}(1 - \xi), \quad N_2 = \frac{1}{2}(1 + \xi), \quad N_i = \phi_{i-1}(\xi) \quad (3.1.23)$$

Avec,
$$\phi_{i-1}(\xi) = \sqrt{\frac{2i-1}{2}} \int_{-1}^{\xi} \mathcal{P}_{i-1}(s) ds \quad (3.1.24)$$

où \mathcal{P}_{i-1} désigne les célèbres polynômes de Legendre. Les fonctions N_1 et N_2 , identiques à celles de l'élément classique, sont dites fonctions de formes nodales ou encore modes externes. Par ailleurs, les fonctions N_i ($2 < i < p + 1$), pour lesquelles on montre la propriété de nullité aux extrémités, sont dites fonctions de formes internes ou modes bulles. Les modes externes permettent de se connecter avec d'autres éléments pour assurer la continuité des déplacements, tandis que les modes internes permettent d'enrichir le champ des déplacements à l'intérieur de l'élément.

Le terme « hiérarchique » est utilisé car l'ensemble des fonctions de forme de degré p inclut ceux des degrés inférieurs $p - 1, p - 2, \dots, 1$ comme sous-ensembles. Le champ des déplacements discrétisés à l'intérieur de l'élément devient alors :

$$u(\xi) = \sum_{i=1}^{p+1} N_i(\xi) d_i \quad (3.1.25)$$

où d_1 et d_2 sont des déplacements nodaux. Par contre, les fonctions internes n'étant associées à aucun des noeuds, les coefficients d_3, \dots, d_{p+1} ne représentent pas des déplacements réels mais plutôt des amplitudes de modes internes ou de déplacements relatifs (cf. Figure 3.3). Leur nature est de ce fait différente de celle des éléments définis en formulation totale où d_3, \dots, d_{p+1} sont réellement des déplacements aux noeuds internes.

Vu la structure hiérarchique des fonctions de forme, la matrice de raideur élémentaire et le vecteur des forces élémentaires ont aussi une structure hiérarchique. Ceci confère à la méthode un avantage majeur puisqu'elle permet une réutilisation à l'ordre $p + 1$ de ces matrices élémentaires issues de l'ordre p . Enfin, bien que le calcul soit accéléré, l'enrichissement est souvent limité du fait de l'apparition des problèmes numériques pour des

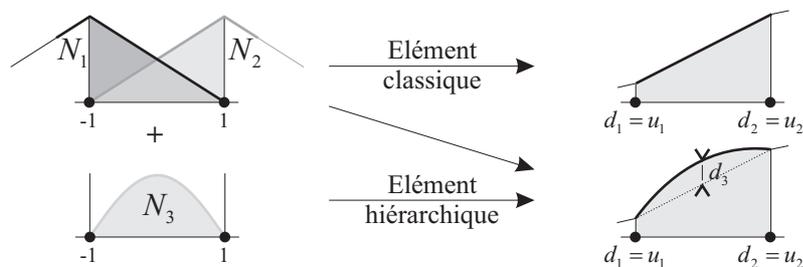


Figure 3.3: Exemple d'élément finis hiérarchique unidimensionnel.

ordres trop élevés. Pour palier à cela, certains travaux proposent l'utilisation de fonctions d'interpolation trigonométriques simples comme base aux éléments finis hiérarchiques [BN97].

3.1.3.2 Méthode de raffinement local automatique - STAR Méthode

On se propose pour terminer de présenter une nouvelle approche adaptative, qui certes s'apparente à une h -méthode mais dont l'originalité mérite une attention particulière [CGLC05b]. Alors que les techniques traditionnelles recourent généralement à un remaillage complet peu performant, la méthode proposée utilise un raffinement local par subdivision d'éléments. Toutefois, la grande particularité de cette méthode réside dans la façon de garantir la précision de la solution. En effet, l'idée est ici de fournir le résultat du calcul éléments finis avec une précision préalablement définie par l'utilisateur. L'utilisation d'un estimateur d'erreur à posteriori est donc couplé à l'algorithme de raffinement. Regardons plus précisément ces deux composants majeurs de la méthode.

Algorithme de raffinement La méthode présentée aborde les problèmes d'élasticité linéaire (2 ou 3 dimensions) en régime statique. On cherche alors à résoudre le système classique :

$$K_{final}u_{final} = f_{final} \tag{3.1.26}$$

Dans cette approche, le premier niveau de discrétisation (grossier), noté $\tilde{\Omega}_1$, est défini par l'utilisateur sur l'ensemble du domaine Ω . La matrice de rigidité K_1 , ainsi que le chargement extérieur f_1 sur le contour discret $\tilde{\Gamma}_1$ étant connu, la solution u_1 peut être calculée sur le premier niveau de discrétisation. Le second niveau, noté $\tilde{\Omega}_2$, peut alors être construit en subdivisant chaque élément de $\tilde{\Omega}_1$ en 4 (cas 2D). On dispose alors des quantités K_2 et f_2 nécessaire au calcul de u_2 . On remarquera que l'on a équivalence des domaines (au sens continu) $\Omega = \Omega_1 = \Omega_2$.

La différence entre les deux solutions est ensuite comparée à la précision préalablement imposée en utilisant les indicateurs d'erreur décrits dans le paragraphe suivant. Dès que cet indicateur dépasse la précision sur un élément de niveau 1, alors cet élément est subdivisé en 16. On forme ainsi la discrétisation de niveau 3 sur le domaine, éventuellement disjoint, Ω_3 .

Ainsi de suite, l'algorithme de raffinement se généralise : en comparant les solutions u_{k-1} et u_k , la grille $k + 1$ sur le domaine Ω_{k+1} peut être définie, et les quantités K_{k+1} , f_{k+1}

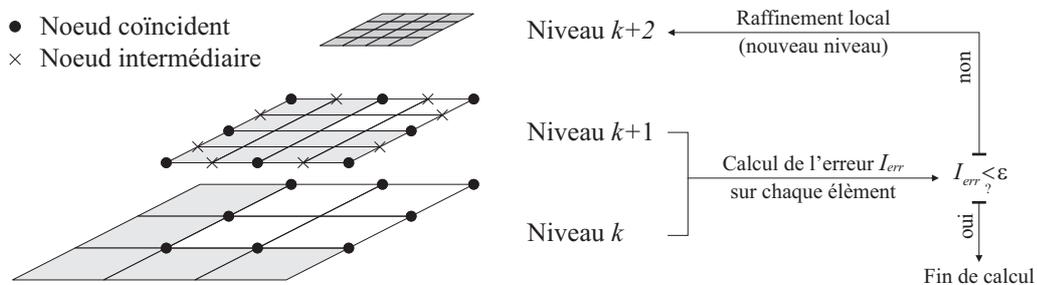


Figure 3.4: Méthode de raffinement automatique local.

peuvent être construites (cf. Figure 3.4). Ce processus est répété jusqu'au niveau n sur lequel la norme est partout inférieure à la précision cherchée. Nous remarquerons alors que les différents domaines de calculs créés sont tel que $\Omega_n \subset \dots \subset \Omega_k \subset \dots \subset \Omega_1$. La solution au problème (3.1.26) s'écrit finalement :

$$u_{final} = u_n + u_{n-1}^* + \dots + u_k^* + \dots + u_2^* \quad (3.1.27)$$

où la solution u_k^* est la partie de u_k sur le domaine $\Omega_k \setminus \Omega_{k+1}$

Indicateur d'erreur Regardons maintenant le second aspect de la méthode concernant les indicateurs d'erreur. Ceux-ci sont obtenus au niveau k en comparant les solutions des niveaux k et $k+1$. Ils peuvent porter sur les différentes grandeurs intervenant lors d'un calcul éléments finis standard, à savoir : le déplacement, la contrainte, la densité d'énergie de déformation et l'énergie de déformation :

- ⇨ **Pour le déplacement** : Celui-ci étant défini aux noeuds, l'indicateur d'erreur est obtenu simplement en soustrayant la solution aux noeuds coïncidents entre les niveaux k et $k+1$. En notant \bar{u}_{k+1} la restriction de u_{k+1} au noeud de Ω_k , cet indicateur peut s'écrire :

$$I_u = \frac{\|\bar{u}_{k+1} - u_k\|}{\max \|u_1\|} \quad (3.1.28)$$

- ⇨ **Pour la contrainte** : Pour ces valeurs définies aux points de Gauss, les valeurs au niveau $k+1$ sont interpolées au point de Gauss de l'élément parent. En notant f_{VM} l'opérateur de Von Mises, l'indicateur en contrainte s'écrit :

$$I_c = \frac{f_{VM}(\bar{\sigma}_{k+1} - \sigma_k)}{f_{VM}(\sigma_1)} \quad (3.1.29)$$

- ⇨ **Pour la densité d'énergie** : Comme pour la contrainte, la densité d'énergie est définie au point de Gauss. On note $e = tr(\sigma\varepsilon)$ la densité d'énergie, et \bar{e}_{k+1} sa valeur projetée du niveau $k+1$ au niveau k :

$$I_d = \sqrt{\frac{\bar{e}_{k+1} - e_k}{\max(e_1)}} \quad (3.1.30)$$

- ⇨ **Pour l'énergie** : Celle-ci étant définie au centre de gravité de l'élément, l'indicateur d'erreur est obtenu de manière analogue en comparant la valeur au centre de gravité des quatre éléments $k+1$ coïncidents avec le celui de l'élément parent :

$$I_e = \sqrt{\frac{\int_{\Omega_k} (\bar{e}_{k+1} - e_k) d\Omega}{\int_{\Omega_1} e_1 d\Omega}} \quad (3.1.31)$$

On montre dans [CGLC05b] que pour une précision donnée, la solution finalement obtenue est indépendante du maillage initial et du critère d'erreur utilisé. Par ailleurs les performances qu'affiche la méthode sont remarquables si l'on compare les temps CPU à ceux obtenus avec un seul maillage homogène mais suffisamment fin pour assurer la même précision.

Remarque 1 : Le principal avantage de cette approche vient du fait qu'à chaque raffinement une partie de la solution est conservée, ce qui par conséquent est plus efficace que les méthodes de remaillage classique. Seules les deux premières itérations (de niveau 1 et 2), sont effectuées sur le domaine complet afin d'initialiser le processus; celles-ci restent néanmoins très peu coûteuses aux vues de la grosseur des grilles qui leurs sont associées.

Remarque 2 : On trouve dans [CGLC05c, CGLC05a] une extension de cette approche au régime dynamique linéaire. La méthode STAR (Space-Time Automatic Refinement) est basée sur un concept d'élément « espace-temps » dont le raffinement automatique est analogue au précédent. La stabilité de la méthode est vérifiée en démontrant qu'aucune énergie n'est produite ou dissipée à l'interface entre deux niveaux successifs.

3.2 Méthode de basculement de maillage en linéaire

On se propose ici de décrire la méthode de basculement de maillage destinée aux calculs de dynamique rapide et plus particulièrement aux calculs de crash. Bien que ces calculs soient pour la plupart fortement non-linéaires, nous nous intéresserons dans cette partie au cas linéaire. L'équation d'équilibre discrétisée que l'on cherche à résoudre à travers notre nouvelle approche s'écrira donc dans un premier temps :

$$M\ddot{u}^n + Ku^n = f_{ext}^n \quad (3.2.1)$$

où K est la matrice de rigidité. Comme nous allons le voir, la méthode mise en oeuvre repose sur l'utilisation successive de maillages, plus ou moins fins, rendus actifs ou inactifs suivant la précision requise pour l'analyse. Nous supposons ici que les différents maillages utilisés pour un même domaine sont préalablement définis ; c'est pourquoi nous parlerons de « basculement » ou « changement » de maillages plutôt que de remaillage.

3.2.1 Principe du basculement de maillage

3.2.1.1 Cadre de l'étude - Limites de l'approche proposée

Comme nous l'avons vu en introduction, cette méthode de raffinement est dédiée à notre approche de calcul multi-domaines présentée au chapitre 1 (cf. §1.2.2). L'idée est donc de pouvoir agir localement et temporairement en modifiant le maillage d'un ou plusieurs sous-domaines au cours de la simulation (cf. Figure 3.5).

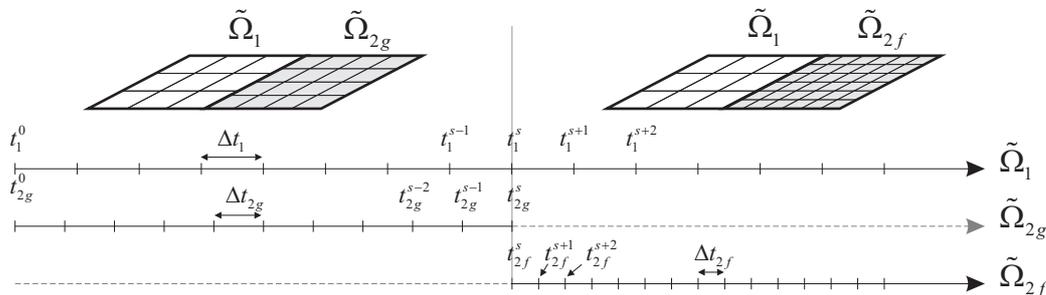


Figure 3.5: Changement de maillage en calcul multi-domaines.

Toutefois, dans le cadre de ce travail, nous nous limiterons au raffinement propre à un sous-domaine sans se préoccuper des interactions avec les sous-domaines voisins. De plus, nous nous intéresserons à un seul raffinement de maillage entre deux niveaux de discrétisation. On considère pour cela un domaine Ω sur lequel sont définies deux discrétisations, l'une grossière notée $\tilde{\Omega}_g$, l'autre fine notée $\tilde{\Omega}_f$. On suppose donc que le calcul est amorcé sur $\tilde{\Omega}_g$, puis poursuivi sur $\tilde{\Omega}_f$, à partir de l'instant t^s que nous appellerons « instant de bascule ». Cet instant, déterminé par un critère que nous détaillerons par la suite (cf. §3.3.2), sera dans un premier temps choisi de manière totalement arbitraire.

D'autre part, la méthode proposée a l'avantage de n'imposer aucune contrainte particulière sur les maillages ; ceux-ci ne sont pas nécessairement hiérarchiques comme cela est souvent le cas dans les méthodes adaptatives. Seul le ratio de « raffinement » l_g/l_f , l

étant la longueur caractéristique moyenne de l'élément, doit rester raisonnable (typiquement inférieure à 10).

Nous remarquerons enfin que l'extension de la méthode mise en oeuvre à une multitude de niveaux est immédiate (succession de bascules entre deux niveaux). De même, cette méthode s'appliquera indifféremment au processus inverse qu'est le déraffinement. Ce dernier, bien que plus rare dans les simulations, peut être envisagé lorsque les déformations subies par une partie critique de la structure n'évoluent quasiment plus.

3.2.1.2 Problématique

Naturellement, le raffinement spatial s'accompagne en explicite d'un raffinement temporel (cf. Figure 3.5). On observe de ce fait une variation brutale du pas de temps généralement source d'instabilité de l'algorithme. Mais outre ceci, la véritable difficulté réside dans la façon d'obtenir un état d'équilibre stable lors de la reprise du calcul sur le maillage fin.

Le problème à résoudre se pose donc en ces termes : partant d'un état mécanique X_g^n définis sur le maillage $\tilde{\Omega}_g$ à l'instant t^s , comment obtenir un état mécanique X_f^{s+1} stable sur $\tilde{\Omega}_f$ à l'instant t^{s+1} ?

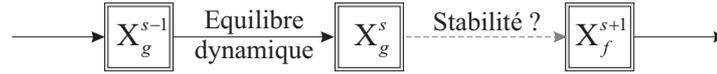


Figure 3.6: Schéma d'évolution entre les instants t^s et t^{s+1} .

Lors du basculement, les différents champs mécaniques associés à l'ancien maillage doivent être transférés au nouveau maillage pour la reprise et la continuation des calculs. On utilise pour cela les opérateurs de changement d'échelles identiques à ceux utilisés dans les approches multi-grilles. On distingue généralement l'opérateur de prolongation \mathcal{P} assurant le passage du maillage grossier $\tilde{\Omega}_g$ au maillage fin $\tilde{\Omega}_f$, et l'opérateur de restriction \mathcal{R} assurant le passage inverse. Pour les grandeurs vectorielles a , nous adopterons les notations suivantes :

$$\bar{a}_f = \mathcal{P}a_g \quad \text{et} \quad \bar{a}_g = \mathcal{R}a_f \quad (3.2.2)$$

où \bar{a} désigne le vecteur projeté (nous noterons ici que l'on a $\mathcal{P}^t = \mathcal{R}$). De même, pour les grandeurs matricielles A , leurs projetés seront notés :

$$\bar{A}_f = \mathcal{P}A_g\mathcal{P}^t \quad \text{et} \quad \bar{A}_g = \mathcal{R}A_f\mathcal{R}^t \quad (3.2.3)$$

Dans le paragraphe suivant, nous détaillerons plus précisément la manière de construire ces opérateurs indispensables (mais nous le verrons, insuffisants) à notre stratégie de discrétisation variable.

3.2.2 Opérateurs de changement d'échelles

3.2.2.1 Techniques de transfert de champs

Il existe dans la littérature différentes techniques de transfert de champs, plus ou moins complexes, et qui généralement dépendent de la formulation utilisée. En formulation Lagrangienne, on peut citer à titre d'exemple les techniques d'approximation diffuse basées

sur l'utilisation de coefficients de pondération des points compris dans une sphère de rayon r centrée au noeud du nouveau maillage. On trouve aussi les techniques du point le plus proche (Closest Point Technique), lesquelles cherchent parmi les noeuds de l'ancien maillage celui le plus proche du nouveau noeud et lui affecte ses propriétés. D'autres approches plus complexes utilisent des techniques d'optimisation sous contrainte, afin que les champs construits vérifient certaines conditions ; parmi ces approches on peut citer de récents travaux où l'optimisation porte sur la minimisation de l'erreur élément fini entre l'ancien et le nouveau champ [EHBVM05, VBS02].

Pour notre problème, nous nous appuyerons sur une simple technique d'interpolation que l'on peut qualifier de transformation isoparamétrique inverse. Celle-ci consiste à déterminer les coordonnées isoparamétriques d'un nouveau noeud dans l'ancien maillage et de les utiliser avec les fonctions de forme de ce dernier pour interpoler les champs. Nous nous placerons ici dans le cas où les éléments sont d'ordre 1. L'opérateur d'interpolation linéaire, relativement simple à mettre en oeuvre, s'avère souvent très efficace.

On se propose ici de décrire ces opérateurs d'interpolations tel que ceux-ci ont été construits dans RAD2RAD. Nous nous focaliserons pour cela au transfert de champs définis sur des maillages surfaciques 3D.

3.2.2.2 Transformation isoparamétrique inverse

Dans le cas de surfaces gauches tridimensionnelles, les maillages utilisés pouvant être quelconques (non hiérarchiques), les noeuds de $\tilde{\Omega}_f$ n'appartiennent pas nécessairement aux surfaces de $\tilde{\Omega}_g$ (et inversement). C'est pourquoi une étape de projection est souvent indispensable. Cette étape pouvant être coûteuse il convient généralement de faire une recherche préalable et rapide des éléments susceptibles de contenir le projeté. Cette recherche consiste à repérer pour chaque noeud du nouveau maillage, le noeud de l'ancien maillage le plus proche ; les éléments contenant ce noeud sont alors examinés un à un.

Cas des éléments triangulaires Intéressons-nous tout d'abord au cas des éléments triangulaires. Le but est de rechercher pour chacun de ces éléments issus de la première sélection, les coordonnées paramétriques $\xi(P_H) = (\xi_1(P_H), \xi_2(P_H), \xi_3(P_H))$ du point P_H , projection orthogonale du point P_M selon la normale sortante n . (cf. Figure 3.7a). Nous noterons par la suite c_{ij} le vecteur d'origine P_i pointant vers le point P_j . Le point P_H

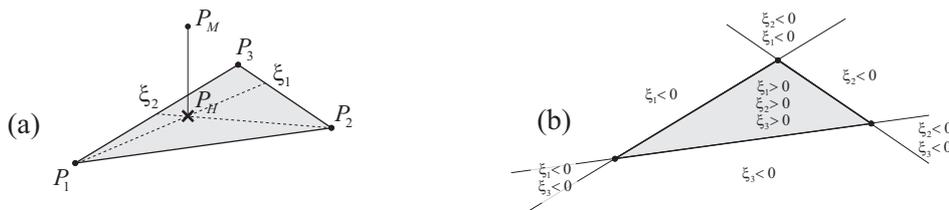


Figure 3.7: Construction des opérateurs de projection : (a) Calcul des coordonnées isoparamétriques du projeté P_H - (b) Zones de projection possibles du point P_M .

est alors défini par :

$$c_{3H} = \xi_1(P_H)c_{32} + \xi_2(P_H)c_{31} \quad (3.2.4)$$

$$c_{HM} \wedge n = 0 \quad (3.2.5)$$

Ici, les coordonnées isoparamétriques sont confondues avec les coordonnées barycentriques de l'élément. On peut donc écrire :

$$\xi_1(P_H) = \frac{\|c_{2H} \wedge c_{23}\|}{\|c_{21} \wedge c_{23}\|} \quad \text{et} \quad \xi_2(P_H) = \frac{\|c_{1H} \wedge c_{13}\|}{\|c_{12} \wedge c_{13}\|} \quad (3.2.6)$$

Si l'on pose $\xi_3 = 1 - \xi_1 - \xi_2$, les valeurs des trois coordonnées paramétriques permettent de trouver dans quelle zone de l'espace se trouve le projeté, et donc de savoir s'il appartient ou non au triangle considéré. (cf. Figure 3.7b). Les points P_1, P_2, P_3, P_H étant coplanaires, les deux produits vectoriels de (3.2.6) sont colinéaires et normaux au plan de l'élément ; la première coordonnée ξ_1 peut donc se réécrire sous la forme :

$$\xi_1(P_H) = \frac{(c_{2H} \wedge c_{23}) \cdot (c_{21} \wedge c_{23})}{(c_{21} \wedge c_{23}) \cdot (c_{21} \wedge c_{23})} \quad (3.2.7)$$

En rappelant que le produit mixte de trois vecteurs est une forme tri-linéaire alternée ($(a \wedge b) \cdot c = (c \wedge a) \cdot b$), on peut réécrire le numérateur de (3.2.7) de la manière suivante :

$$(c_{2H} \wedge c_{23}) \cdot (c_{21} \wedge c_{23}) = [(c_{21} \wedge c_{23}) \wedge c_{2H}] \cdot c_{23} \quad (3.2.8)$$

D'autre part, en appliquant la formule du double produit vectoriel $((a \wedge b) \wedge c = (a \cdot c)b - (c \cdot b)a)$ à (3.2.8), on obtient :

$$(c_{2H} \wedge c_{23}) \cdot (c_{21} \wedge c_{23}) = [(c_{21} \cdot c_{2H})c_{23} - (c_{2H} \cdot c_{23})c_{21}] \cdot c_{23} \quad (3.2.9)$$

$$= (c_{21} \cdot c_{2H})(c_{23} \cdot c_{23}) - (c_{2H} \cdot c_{23})(c_{21} \cdot c_{23}) \quad (3.2.10)$$

$$= (c_{21} \cdot c_{2M})(c_{23} \cdot c_{23}) - (c_{2M} \cdot c_{23})(c_{21} \cdot c_{23}) \quad (3.2.11)$$

En procédant de la même manière pour le dénominateur de (3.2.7), on remarque qu'il est finalement possible de calculer directement $\xi_1(P_H)$ à partir des coordonnées du point P_M et des sommets P_1, P_2, P_3 de l'élément dans le repère global.

$$\xi_1(P_H) = \frac{(c_{21} \cdot c_{2M})(c_{23} \cdot c_{23}) - (c_{2M} \cdot c_{23})(c_{21} \cdot c_{23})}{c_{12}^2 c_{13}^2 - (c_{12} \cdot c_{13})^2} \quad (3.2.12)$$

La détermination explicite des coordonnées de P_H dans le repère global est de ce fait inutile. Une conséquence de cela est qu'au cours du calcul, les étapes de projection et de calcul des coordonnées isoparamétriques restent indissociables.

Cas des éléments quadrangulaires Pour les quadrangles, la détermination des coordonnées paramétriques dans l'élément courant nécessiterait le retour à l'élément de référence, et donc la résolution d'un problème non linéaire, qui est coûteux. Nous avons ici choisi une solution approchée, qui consiste à découper virtuellement le quadrangle en deux triangles, selon les deux manières possibles (cf. Figure 3.8). On peut alors calculer la distance du point P_M à chacun des quatre triangles ainsi définis et choisir le triangle affichant la plus petite distance projetée. La relation d'interpolation est alors écrite comme précédemment entre le point P_M et les trois sommets du triangle choisi. Dans le cas où

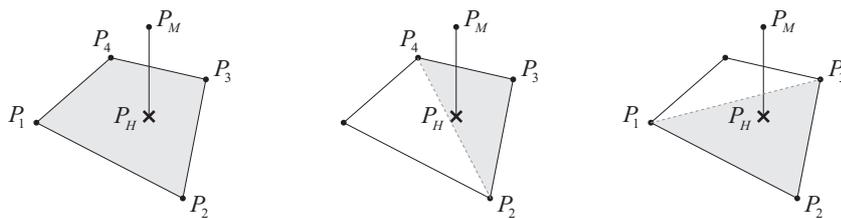


Figure 3.8: Découpage d'un quadrangle linéaire.

le quadrangle reste plan, la projection sur le triangle est équivalente à la projection sur le quadrangle ; dans le cas plus général où le quadrangle est gauche, cette opération est un moyen de prendre en compte, d'une certaine façon, la courbure.

3.2.2.3 Calcul de l'opérateur de projection

Parmi les variables d'états transférées d'une discrétisation à l'autre, on distingue celles qui sont stockées aux noeuds (comme par exemple les déplacements, vitesses...) de celles stockées aux points d'intégrations (comme par exemple les contraintes, les variables d'écrouissage...). Les opérateurs de projections \mathcal{P} et \mathcal{R} définis par (3.2.2) permettent de traiter indifféremment ces différents champs. Afin d'illustrer leurs constructions nous nous placerons ici dans le cas d'un transfère de champ nodal, tel que par exemple le champ de déplacement :

$$u_f = \mathcal{P}u_g \quad \text{et} \quad u_g = \mathcal{R}u_f \quad (3.2.13)$$

Si l'on se concentre sur l'opérateur de \mathcal{P} , la relation précédente peut se réécrire sous la forme matricielle :

$$\begin{Bmatrix} u_{f,1} \\ \vdots \\ u_{f,i} \\ \vdots \\ u_{f,n} \end{Bmatrix} = \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1j} & \cdots & \mathcal{P}_{1n} \\ \vdots & \ddots & & & \vdots \\ \mathcal{P}_{i1} & & \mathcal{P}_{ij} & & \mathcal{P}_{in} \\ \vdots & & & \ddots & \vdots \\ \mathcal{P}_{m1} & \cdots & \mathcal{P}_{mj} & \cdots & \mathcal{P}_{mn} \end{bmatrix} \begin{Bmatrix} u_{g,1} \\ \vdots \\ u_{g,j} \\ \vdots \\ u_{g,m} \end{Bmatrix} \quad (3.2.14)$$

où n et m sont respectivement les nombres de ddls sur le maillage fin $\tilde{\Omega}_f$ et grossier $\tilde{\Omega}_g$. Concrètement, chaque composante \mathcal{P}_{ij} représente la contribution du noeud j de $\tilde{\Omega}_g$ dans l'évaluation du champ faite au noeud i de $\tilde{\Omega}_f$:

$$u_{f,i} = \sum_{j=1}^m \mathcal{P}_{ij} u_{g,j} \quad (3.2.15)$$

Ces composantes \mathcal{P}_{ij} peuvent donc être directement calculées à partir des fonctions de formes de l'élément :

$$\mathcal{P}_{ij} = N_j(x_i) \quad (3.2.16)$$

où N_j est la fonction de forme associée au noeud j et x_i la coordonnée du noeud i . En se plaçant au niveau élémentaire on peut réécrire (3.2.16) en fonction des coordonnées isoparamétriques du point P_i dans l'élément de projection. Si l'on remarque alors que

pour les éléments triangles linéaires, les fonctions de formes sont confondues avec les coordonnées barycentriques (soit ici $N_j = \xi_j$), on obtient :

$$\mathcal{P}_{ij} = N_j(P_i) = \xi_j(P_i) \quad (3.2.17)$$

3.2.3 Insuffisance de la projection

Lors du basculement, la reprise et la continuation des calculs sont facilitées si les champs associés au nouveau maillage (après transfert) vérifient approximativement toutes les équations du problème (équation d'équilibre, critère de plasticité, conditions aux limites...). Nous allons voir dans cette partie que le transfert de champs par simple interpolation ne garantit généralement pas cette condition ; ceci sera d'autant plus vrai pour les problèmes fortement non-linéaires. De plus, ces méthodes classiques de raccord par interpolation sont rendues particulièrement instables par l'utilisation du schéma explicite. La difficulté majeure à laquelle nous sommes confrontés est donc de marier stabilité et précision de calcul. Nous nous focaliserons pour cela sur l'étape de transition et sur la manière d'équilibrer la solution sur la nouvelle discrétisation.

3.2.3.1 Étude de stabilité de l'algorithme à discrétisation variable

On se propose ici d'étudier la stabilité du schéma de Newmark en se plaçant au moment du changement de discrétisation [Rét05], lors de la reprise du calcul sur le maillage fin. Nous nous baserons pour cela sur l'approche énergétique proposée dans [Hug01]. Cette approche énergétique est une alternative à l'approche modale utilisée au paragraphe 1.1.3 pour l'étude de stabilité des calculs dynamiques standards, sans évolution de maillage. On définit les notations suivantes pour la moyenne $\langle \cdot \rangle$ et la différence $[\cdot]$ d'une quantité a entre les instants t^s et t^{s+1} .

$$\langle a \rangle = \frac{1}{2}(a^{s+1} + a^s) \quad (3.2.18)$$

$$[a] = (a^{s+1} - a^s) \quad (3.2.19)$$

Avec la propriété remarquable,

$$\langle a \rangle^t [a] = \frac{1}{2} [a^t a] \quad (3.2.20)$$

Dans le cas particulier du changement de discrétisation, on suppose que les variables définies sur le maillage fin à l'instant t^s sont estimées par prolongation des valeurs calculées sur le maillage grossier.

$$\langle a_f \rangle = \frac{1}{2}(a_f^{s+1} + \bar{a}_f^s) \quad (3.2.21)$$

$$[a_f] = (a_f^{s+1} - \bar{a}_f^s) \quad (3.2.22)$$

En utilisant ces notations, les relations de Newmark définissant l'actualisation des champs cinématiques peuvent se réécrire de la façon suivante :

$$[u_f] = \Delta t \langle \dot{u}_f \rangle + \frac{\Delta t^2}{2}(2\beta - \gamma) [\ddot{u}_f] \quad (3.2.23)$$

$$[\dot{u}_f] = \Delta t \langle \ddot{u}_f \rangle + \Delta t(\gamma - \frac{1}{2}) [\ddot{u}_f] \quad (3.2.24)$$

D'après [HL78b, HL78a] la stabilité du schéma n'est pas influencée par les efforts extérieurs (il doit être stable quel que soit le problème posé). Nous nous intéresserons donc

aux problèmes d'équilibres sans second membre, aux instants t^s et t^{s+1} :

$$M_f \ddot{u}_f^{s+1} + K_f u_f^{s+1} = 0 \quad (3.2.25)$$

$$M_g \ddot{u}_g^s + K_g u_g^s = 0 \quad (3.2.26)$$

L'étude de stabilité par la méthode énergétique consiste à pré multiplier les deux équations d'équilibres par \dot{u}_f et à soustraire les deux quantités obtenues. Il convient pour cela de projeter l'équation d'équilibre (3.2.26) sur la discrétisation fine, soit :

$$\bar{M}_f \bar{\ddot{u}}_f^{s+1} + \bar{K}_f \bar{u}_f^{s+1} = 0 \quad (3.2.27)$$

où d'après (3.2.3), \bar{M}_f et \bar{K}_f sont les matrices de masses et raideurs grossières projetées sur le maillage fin. La condition de stabilité peut alors s'écrire :

$$[\dot{u}_f] (M_f \ddot{u}_f^{s+1} - \bar{M}_f \bar{\ddot{u}}_f^{s+1} + K_f u_f^{s+1} - \bar{K}_f \bar{u}_f^{s+1}) = 0 \quad (3.2.28)$$

En remarquant, par ailleurs que l'on a ,

$$[M_f \ddot{u}_f] = M_f [\ddot{u}_f] + [M_f] \bar{\ddot{u}}_f^n \quad (3.2.29)$$

$$[K_f u_f] = K_f [u_f] + [K_f] \bar{u}_f^n \quad (3.2.30)$$

et en utilisant les relations (3.2.23-3.2.24) on obtient finalement une relation du type :

$$B = \frac{1}{2} [\ddot{u}_f^t A_f \ddot{u}_f + \dot{u}_f^t K_f \dot{u}_f] \quad (3.2.31)$$

$$= -(\gamma - \frac{1}{2}) [\ddot{u}_f]^t A_f [\ddot{u}_f] - \frac{1}{\Delta t} [\dot{u}_f]^t ([M_f] \bar{\ddot{u}}_f^n + [K_f] \bar{u}_f^n) \quad (3.2.32)$$

Avec,
$$A_f = M_f + \frac{\Delta t^2}{2} (2\beta - \gamma) K_f \quad (3.2.33)$$

Théoriquement, l'algorithme est stable si le terme B de (3.2.31) est négatif ou nul (dissipation d'énergie). On reconnaît alors dans le membre de droite le terme en $(\gamma - 1/2)$ dont dépend la stabilité du calcul sans évolution de maillage. Dans ce cas, l'étude de stabilité, revient à vérifier que la matrice A_f soit définie positive. L'analyse de ses valeurs propres conduit aux conditions classiques du schéma de Newmark, à savoir :

$$\begin{cases} \frac{1}{2} \leq \gamma \leq 2\beta & \text{Schéma inconditionnellement stable.} \\ \frac{1}{2} \leq \gamma \text{ et } 2\beta \leq \gamma & \text{Schéma stable si } \Delta t \leq \Delta t_c. \end{cases} \quad (3.2.34)$$

Si l'on se place dans le cas particulier du schéma aux différences centrées explicites ($\gamma = 1/2$, $\beta = 0$), on vérifie directement que le terme $(\gamma - 1/2) [\ddot{u}]^t A_f [\ddot{u}]$ est nul. Cependant, la présence d'un terme supplémentaire $\frac{1}{\Delta t} [\dot{u}_f]^t ([M_f] \bar{\ddot{u}}_f^n + [K_f] \bar{u}_f^n)$, dont on ne connaît a priori pas le signe, remet en cause les conditions de stabilités habituelles. Les instabilités éventuelles proviennent des évolutions du maillage et des opérations de projections intervenant dans les termes $[M] = M_f - \bar{M}_f$ et $[K] = K_f - \bar{K}_f$ généralement non nuls.

3.2.3.2 Interprétation du déséquilibre

Les champs cinématiques $\bar{u}_f^s, \bar{u}_f^s, \bar{u}_f^s$, estimés à partir du calcul grossier, ne vérifient pas nécessairement l'équilibre sur le maillage fin. En effet, à l'instant t^s , on a :

$$M_g \ddot{u}_g^s + K_g u_g^s = f_{ext,g}^s \quad (3.2.35)$$

Si on projette cet équilibre sur le maillage fin, on obtient :

$$0 = \mathcal{P}(M_g \ddot{u}_g^s + K_g u_g^s - f_{ext,g}^s) \quad (3.2.36)$$

$$= \mathcal{P}(M_g \ddot{u}_g^s) + \mathcal{P}(K_g u_g^s) - \mathcal{P} f_{ext,g}^s \quad (3.2.37)$$

$$= (\mathcal{P} M_g \mathcal{P}^t) \bar{u}_f^s + (\mathcal{P} K_g \mathcal{P}^t) \bar{u}_f^s - \bar{f}_{ext,f}^s \quad (3.2.38)$$

$$= \bar{M}_f \bar{u}_f^s + \bar{K}_f \bar{u}_f^s - \bar{f}_{ext,f}^s \quad (3.2.39)$$

où \bar{M}_f et \bar{K}_f sont des matrices de masse et de rigidité interpolées à partir de leurs homologues réellement calculés sur le maillage grossier. Elles sont donc généralement différentes des matrices M_f et K_f construites classiquement par assemblage des matrices élémentaires. Cette différence est à l'origine du problème de stabilité décrit précédemment ; on remarquera que si $\bar{M}_f = M_f$ et $\bar{K}_f = K_f$ alors le terme de (3.2.31) source d'instabilité s'annule. On a par conséquent :

$$M_f \ddot{u}_f^s + K_f \bar{u}_f^s - f_{ext,f}^s \neq 0 \quad (3.2.40)$$

De plus, durant la phase de projection, les noeuds du nouveau maillage viennent se « plaquer » sur l'ancien maillage (cf. Figure 3.9). Ceci entraîne une sur-contrainte dans la structure qui logiquement voit son énergie interne augmenter.

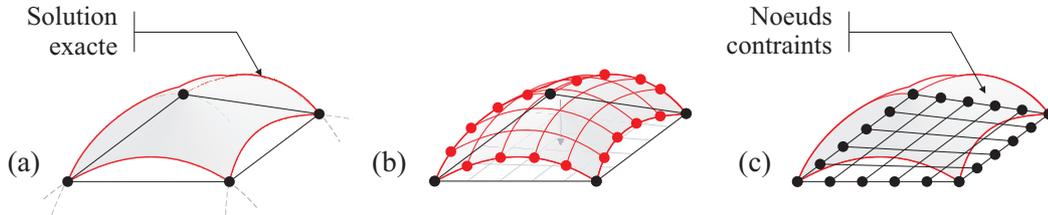


Figure 3.9: Précontrainte des noeuds intermédiaires : (a) Solution grossière - (b) Solution fine - (c) Solution grossière projetée sur le maillage fin.

Exemple : flexion de poutre 1D Pour illustrer cela, prenons l'exemple simple d'une poutre encastree/libre de longueur l soumise à un chargement statique f (cf. Figure 3.10). La formulation forte du problème s'écrit :

$$EI \frac{\partial^4 u(x)}{\partial x^4} = f(x) \quad (3.2.41)$$

où E est le module de Young et I le moment d'inertie de la poutre. On choisit ici de discrétiser le système par des éléments 1D à 2 ddls par noeud (u, θ) , en utilisant les fonctions de formes cubiques qui assurent la continuité des déplacements et des rotations aux noeuds :

$$N_1(\xi) = 1 - 3\xi^2 + 2\xi^3 \quad N_2(\xi) = \xi - 2\xi^2 + \xi^3 \quad (3.2.42)$$

$$N_3(\xi) = 3\xi^2 - 2\xi^3 \quad N_4(\xi) = \xi^2 + \xi^3 \quad (3.2.43)$$

On définit alors deux maillages, l'un grossier $\tilde{\Omega}_g$ comprenant un unique élément et l'autre fin $\tilde{\Omega}_f$ comprenant deux éléments. La résolution analytique du problème conduit pour chacune des discrétisations aux solutions suivantes :

$$u_g = \{u_1, \theta_1, u_2, \theta_2\}_g = \left\{ 0, 0, \frac{fl^3}{3EI}, \frac{fl^2}{2EI} \right\} \quad (3.2.44)$$

$$u_f = \{u_1, \theta_1, u_2, \theta_2, u_3, \theta_3\}_f = \left\{ 0, 0, \frac{5fl^3}{48EI}, \frac{3fl^2}{8EI}, \frac{fl^3}{3EI}, \frac{fl^2}{2EI} \right\} \quad (3.2.45)$$

On peut ensuite procéder à la projection de u_g sur le maillage fin. En utilisant un opérateur de prolongation \mathcal{P} linéaire, on obtient de manière trivial :

$$\bar{u}_f = \left\{ 0, 0, \frac{fl^3}{6EI}, \frac{fl^2}{4EI}, \frac{fl^3}{3EI}, \frac{fl^2}{2EI} \right\} \neq u_f \quad (3.2.46)$$

Si l'on pousse l'analyse jusqu'au calcul des énergies de déformations discrètes, on obtient pour le calcul fin et pour le calcul grossier projeté sur le maillage fin, les valeurs suivantes :

$$\mathcal{W}_{int,f} = \{u_f\}^t [K_f] \{u_f\} = \frac{1}{3} \cdot \frac{f^2 l^3}{EI} \quad (3.2.47)$$

$$\bar{\mathcal{W}}_{int,f} = \{\bar{u}_f\}^t [K_f] \{\bar{u}_f\} = \frac{4}{3} \cdot \frac{f^2 l^3}{EI} \quad (3.2.48)$$

Dans ce cas précis, on observe que l'énergie interne du système augmente d'un facteur 4 lorsque l'on contraint le noeud central de $\tilde{\Omega}_f$ à suivre la déformation de $\tilde{\Omega}_g$. Ceci montre bien que le transfert de champs par simple interpolation linéaire conduit à un état de déséquilibre et se traduit par une énergie interne anormalement élevée.

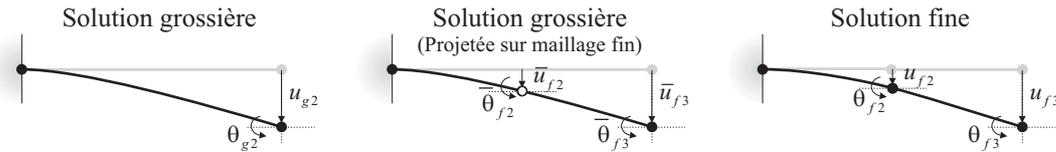


Figure 3.10: Exemple 1D : Poutre encastrée/libre en flexion.

3.2.4 Méthode de rééquilibrage dynamique

Pour palier aux problèmes cités précédemment, une phase de rééquilibrage est indispensable. On introduit ainsi sur le schéma de la figure 3.11 un état intermédiaire noté X_f^s , défini sur $\tilde{\Omega}_f$ à l'instant t^s . Les deux discrétisations, grossière et fine, cohabitent donc à cet instant. De cette manière, les matrices projetées sur la discrétisation fine sont remplacées par les matrices réellement calculées M_f et K_f et l'annulation des termes $[M]$ et $[K]$ assure la stabilité du schéma entre t^s et t^{s+1} . Intéressons-nous alors au passage de la configuration X_g^s à la configuration X_f^s .

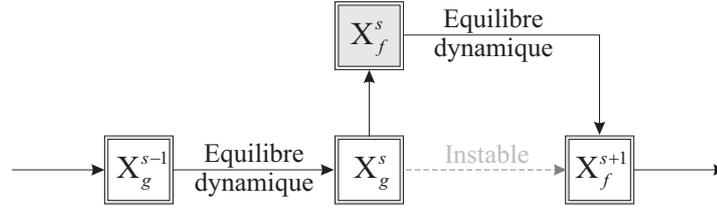


Figure 3.11: Schéma d'évolution entre les instants t^s et t^{s+1} .

3.2.4.1 Recouvrement d'échelles

La méthode de rééquilibrage mise en oeuvre repose sur une période de recouvrement, où les échelles de temps relatives à l'ancienne et à la nouvelle discrétisation se chevauchent, (cf. Figure 3.12). L'instant de bascule étant à priori imprévisible, un retour en arrière est ici indispensable.

L'idée est de consacrer cette période de recouvrement à un cycle « d'équilibrage » sur la nouvelle discrétisation, en s'inspirant du schéma d'intégration implicite nettement plus stable. En effet, l'implicite étant inconditionnellement stable, la longueur du pas de temps d'équilibrage Δt_{eq} correspondant à la période de recouvrement pourra être choisi de manière tout à fait arbitraire. Le but de ce cycle particulier est d'obtenir un état d'équilibre stable sur la nouvelle discrétisation à l'instant t^s et ainsi pouvoir assurer la continuation des calculs (cette fois explicite) sur celle-ci.

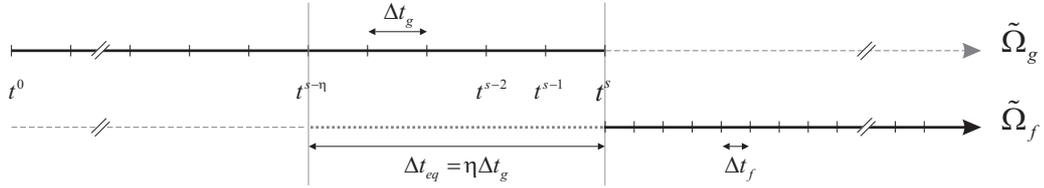


Figure 3.12: Recouvrement des échelles de temps avant raffinement.

Un paramètre η que nous appellerons « paramètre de recouvrement » est alors introduit ; il correspond au nombre de pas de temps grossiers, antérieurs à t^s que recouvre Δt_{eq} :

$$\Delta t_{eq} = t^s - t^{s-\eta} = \sum_{i=1}^{\eta} \Delta t_g^{s-i} \quad (3.2.49)$$

L'instant antérieur à partir duquel s'opère le pas de temps « implicite » sera alors noté $t^{s-\eta}$. Nous supposons par la suite que le pas de temps est constant de manière à avoir :

$$\Delta t_{eq} = \eta \Delta t_g \quad (3.2.50)$$

3.2.4.2 Cycle d'équilibrage « implicite »

Revenons sur l'équation d'équilibre discrète que l'on souhaite vérifier sur la nouvelle discrétisation (ici la discrétisation fine) à l'instant t^s :

$$M_f \ddot{u}_f^s + K_f u_f^s = f_{ext,f}^s \quad (3.2.51)$$

En utilisant les notations définies au paragraphe précédent, on peut réécrire le schéma général de Newmark, appliqué aux champs cinématiques du maillage fin, entre les instants $t^{s-\eta}$ et t^s :

$$u_f^s = {}^p u_f^s + \beta \Delta t_{eq}^2 \ddot{u}_f^s \quad (3.2.52)$$

$$\dot{u}_f^s = {}^p \dot{u}_f^s + \gamma \Delta t_{eq} \ddot{u}_f^s \quad (3.2.53)$$

Avec,
$${}^p u_f^s = u_f^{s-\eta} + \Delta t_{eq} \dot{u}_f^{s-\eta} + \frac{\Delta t_{eq}^2}{2} (1 - 2\beta) \ddot{u}_f^{s-\eta} \quad (3.2.54)$$

$${}^p \dot{u}_f^s = \dot{u}_f^{s-\eta} + (1 - \gamma) \Delta t_{eq} \ddot{u}_f^{s-\eta} \quad (3.2.55)$$

Évidemment la principale difficulté est d'estimer les prédicteurs en déplacement ${}^p u_f^s$ et en vitesse ${}^p \dot{u}_f^s$, puisque ces derniers font intervenir des quantités, calculées à l'instant $t^{s-\eta}$, à priori inconnues. En première approximation, nous avons choisi d'évaluer ces quantités en utilisant les valeurs issues du calcul grossier à cet instant antérieur. Autrement dit, on fait l'hypothèse que l'on a :

$$u_f^{s-\eta} = \mathcal{P} u_g^{s-\eta} \quad (3.2.56)$$

$$\dot{u}_f^{s-\eta} = \mathcal{P} \dot{u}_g^{s-\eta} \quad (3.2.57)$$

$$\ddot{u}_f^{s-\eta} = \mathcal{P} \ddot{u}_g^{s-\eta} \quad (3.2.58)$$

En injectant (3.2.52) dans l'équation d'équilibre (3.2.51), on obtient dans le cas linéaire un système dont la seule inconnue est le vecteur déplacement u_f^s :

$$\frac{M_f}{\beta \Delta t_{eq}^2} (u_f^s - {}^p u_f^s) + K_f u_f^s = f_{ext,f}^s \quad (3.2.59)$$

Une fois résolu, le vecteur u_f^s est intégré aux relations de Newmark, afin de calculer les vitesses \dot{u}_f^s et les accélérations \ddot{u}_f^s .

Nous verrons par la suite que le pas de temps d'équilibre Δt_{eq} doit être suffisamment grand pour assurer une bonne stabilité de la solution. Une valeur trop faible du paramètre de recouvrement (typiquement $\eta = 1$) peut conduire à une perturbation de la solution après la reprise de l'algorithme explicite. Une manière de gommer les perturbations résiduelles consiste à introduire un amortissement numérique, par l'intermédiaire du coefficient γ , sur les premiers cycles explicites succédant la bascule. Ceci constitue toutefois un artifice de calcul sur lequel nous ne nous attarderons pas ici.

Remarque 1 : En pratique, cette méthode revient à opérer une double bascule explicite/implicite puis implicite/explicite. Elle s'apparente d'ailleurs aux techniques de calculs combinés explicite/implicite développés dans [NSPB02, NSP04a, NSP04b]. Bien que la séquence implicite est réduite à un seul pas de temps dans la démarche présentée, il est tout à fait possible d'en envisager plusieurs. Quelques études de sensibilité ont été réalisées à ce sujet sur des géométries simples. Il s'est alors avéré que le paramètre influant était le paramètre de recouvrement η plutôt que le nombre de cycles d'équilibrage.

3.3 Extension au cas non-linéaire

Dans les applications telles que le sont les simulations numériques de crash, les sources de non-linéarités sont généralement très variées. On distingue alors les non-linéarités géométriques (grands déplacements, grandes déformations), les non-linéarités d'interfaces (conditions limites variables, chocs, contacts, frottements...) et les non-linéarités matérielles (élasto-plasticité, visco-plasticité, endommagement...). Dans cette section, nous nous intéresserons au cas de ces non-linéarités matérielles en considérant, par exemple, une loi de comportement de type élasto-plastique. L'équilibre discret s'écrit alors de manière générale :

$$M\ddot{u}^n + f_{int}^n(u^n) = f_{ext}^n \quad (3.3.1)$$

Bien que la méthode de basculement de maillage proposée précédemment reste sensiblement la même, son extension aux problèmes non-linéaires demande quelques considérations particulières. La phase de rééquilibrage sera notamment inscrite dans une procédure itérative de Newton-Raphson afin de prendre en compte la dépendance à l'histoire des non-linéarités matérielles.

3.3.1 Méthode de rééquilibrage itérative

La méthode de Newton-Raphson apparaît comme la plus célèbre des méthodes de résolution d'équations algébriques non-linéaires. Avant de développer cette méthode, il convient de reformuler l'équilibre discret sous la forme :

$$r(u_f^s) = M_f \ddot{u}_f^s + f_{int,f}^s(u_f^s) - f_{ext,f}^s = 0 \quad (3.3.2)$$

où $r(u_f^{s+1})$ est appelé résidu. Ce dernier peut être interprété comme une mesure du déséquilibre des forces. En pratique, (3.3.2) est annulé de manière itérative ; nous noterons $u_f^{s(k)}$ le déplacement calculé sur le maillage fin à l'itération k , et $\Delta u_f^{s(k)}$ l'incrément de déplacement entre deux itérés successifs :

$$\Delta u_f^{s(k)} = u_f^{s(k+1)} - u_f^{s(k)} \quad (3.3.3)$$

3.3.1.1 Résolution du pas de temps d'équilibre

Comme pour le cas linéaire, en utilisant les relations (3.2.52-3.2.53) du schéma de Newmark, on peut écrire l'équilibre (3.3.2) sous la forme :

$$r(u_f^s) = \frac{M_f}{\beta \Delta t_{eq}^2} (u_f^s - p u_f^s) + f_{int,f}^s(u_f^s) - f_{ext,f}^s = 0 \quad (3.3.4)$$

La procédure d'équilibrage s'apparente toujours à un calcul implicite sur un pas de temps Δt_{eq} entre les instants t^s et $t^{s-\eta}$.

Initialisation du pas de temps Ce pas de temps est initialisé comme précédemment à l'instant $t^{s-\eta}$ en projetant les champs cinématiques provenant de l'ancien maillage :

$$u_f^{s-\eta} = \bar{u}_f^{s-\eta} = \mathcal{P} u_g^{s-\eta} \quad (3.3.5)$$

$$\dot{u}_f^{s-\eta} = \bar{\dot{u}}_f^{s-\eta} = \mathcal{P} \dot{u}_g^{s-\eta} \quad (3.3.6)$$

$$\ddot{u}_f^{s-\eta} = \bar{\ddot{u}}_f^{s-\eta} = \mathcal{P} \ddot{u}_g^{s-\eta} \quad (3.3.7)$$

Nous rappellerons que l'état mécanique à cet instant n'est pas équilibré (cf. §3.2.3). Dans le cas non-linéaire, les prédicteurs du schéma calculés à partir de (3.3.5-3.3.7) seront supposés constants tout au long de l'équilibrage. A chaque itération, le calcul du nouveau déplacement $u_{f(k)}^s$ permet alors d'accéder aux nouvelles vitesses et accélérations grâce aux relations :

$$\ddot{u}_{f(k)}^s = \frac{1}{\beta \Delta t_{eq}^2} (u_{f(k)}^s - {}^p u_f^s) \quad (3.3.8)$$

$$\dot{u}_{f(k)}^s = {}^p u_f^s + \gamma \Delta t_{eq} \ddot{u}_{f(k)}^s \quad (3.3.9)$$

Procédure itérative de Newton-Raphson Rappelons brièvement comment l'algorithme de Newton s'applique aux problèmes dynamiques. Cet algorithme repose sur un développement de Taylor du résidu r autour de la valeur courante du déplacement :

$$r(u_{f(k+1)}^s) = r(u_{f(k)}^s) + \frac{\partial r}{\partial u} (u_{f(k)}^s) \Delta u_{f(k)}^s + \mathcal{O}(\Delta u_f^2) \quad (3.3.10)$$

La matrice $\partial r / \partial u$, appelée Jacobienne, pourra alors s'écrire :

$$J = \frac{\partial r}{\partial u} = \frac{1}{\beta \Delta t_{eq}^2} M_f + \frac{\partial f_{int,f}^s}{\partial u} = \frac{1}{\beta \Delta t^2} M_f + K_f^{tan} \quad (3.3.11)$$

où K_f^{tan} désigne la matrice de raideur tangente. En substituant J dans (3.3.10) et en négligeant les termes en Δu_f d'ordre supérieur à un, on obtient le modèle linéarisé :

$$J \cdot \Delta u_{f(k)}^s = -r(u_{f(k)}^s) \quad (3.3.12)$$

La solution du problème non-linéaire est donc obtenue en résolvant une suite de systèmes linéaires et le déplacement solution est construit en ajoutant les incréments de déplacement $\Delta u_{f(k)}^s$ calculés à chaque itération. La procédure itérative continue jusqu'à ce que le critère de convergence soit vérifié, c'est-à-dire jusqu'à ce que la solution obtenue atteigne la précision voulue. Nous utilisons ici un critère de convergence portant sur la norme, au sens ℓ_2 , du résidu r :

$$\|r\|_{\ell_2} = \sqrt{\sum_{i=1, n_{ddl}} r_i^2} \leq \epsilon \max(\|f_{ext}\|_{\ell_2}, \|f_{int}\|_{\ell_2}, \|M\ddot{u}\|_{\ell_2}) \quad (3.3.13)$$

La tolérance ϵ détermine la précision avec laquelle le déplacement u_f^s sera finalement calculé.

Convergence de l'algorithme : Sous certaines conditions la méthode de Newton-Raphson offre une convergence quadratique vers la solution. Elle nécessite d'une part un résidu r continument dérivable, et d'autre part une matrice Jacobienne J inversible telle que J^{-1} soit uniformément bornée au voisinage de la solution. Toutefois, ces conditions ne sont généralement pas satisfaites dans les problèmes d'ingénieur. Par exemple, dans le cas d'un matériau élasto-plastique, lorsque les points d'intégration de l'élément passent du comportement élastique au comportement plastique, la dérivée de r est discontinue. De plus, dans de tels problèmes d'équilibre, la convergence de la méthode de Newton échoue souvent à proximité des états instables. Dans ce cas, la matrice Jacobienne n'est plus régulière et la preuve de convergence quadratique ne s'applique plus.

Cela peut alors conduire à une convergence lente ou même à une non-convergence de la méthode. Le mauvais conditionnement que cela entraîne n'assure généralement pas la validité de la solution. Heureusement, ces difficultés disparaissent habituellement en dynamique puisque les effets d'inertie liés à la matrice consistante M_f rendent la Jacobienne plus positive (3.3.11).

3.3.1.2 Initialisation de la procédure

Transfert de champs : L'entame du processus itératif nécessite l'utilisation d'une valeur initiale du déplacement u_f^s . La prolongation de la solution grossière calculée au même instant apparaît donc comme une bonne estimation de cette solution :

$$u_{f(0)}^s = \bar{u}_f^s = \mathcal{P}u_g^s \quad (3.3.14)$$

Cependant, cette initialisation est insuffisante en non-linéaire où un certain nombre d'autres variables internes caractérisant l'histoire de la transformation doivent être prises en compte (contrainte, déformation plastique, énergie interne, paramètre d'écrouissage...). Pour une loi de comportement de type elasto-plastique, nous relèverons essentiellement l'initialisation du champ de contrainte σ , et du champ de déformation plastique ε_{pl} , calculé à l'instant t^s . Celles-ci seront également issues du calcul réalisé sur le maillage grossier :

$$\sigma_{f(0)}^s = \bar{\sigma}_f^s = \mathcal{P}\sigma_g^s \mathcal{P}^t \quad (3.3.15)$$

$$\varepsilon_{pl,f(0)}^s = \bar{\varepsilon}_{pl,f}^s = \mathcal{P}\varepsilon_{pl,g}^s \mathcal{P}^t \quad (3.3.16)$$

Le transfert d'autres grandeurs, non indispensables au calcul, mais intéressantes au post-traitement, pourra être fait à ce stade.

Calcul de la matrice Jacobienne : Nous adopterons ici une méthode de Newton modifiée, où la matrice Jacobienne, alors notée \tilde{K} , est calculée uniquement au début de la procédure. La matrice de raideur tangente K_f^{tan} sera dans ce cas choisie comme étant confondue avec la matrice de rigidité construite sur le maillage fin K_f :

$$J = \tilde{K} = \frac{1}{\beta\Delta t^2}M_f + K_f \quad (3.3.17)$$

Nous noterons ici le surplus de calcul, par rapport aux cycles explicites traditionnels où la matrice de rigidité K_f n'est habituellement pas assemblée. Evidemment l'unique évaluation de K_f au début d'équilibrage limite fortement le surplus d'opérations mais cela au dépend de la convergence de l'algorithme. Malgré cela, nous remarquerons que la procédure converge rapidement et que quelques itérations suffisent généralement pour satisfaire le critère adopté. Ceci ne rend par conséquent pas la méthode prohibitive et inefficace suite au surcoût de calcul qu'engendre le rééquilibrage.

3.3.2 Critère de changement de maillage

Jusqu'ici l'instant t^s auquel avait lieu le changement de maillage était supposé choisi de manière arbitraire. Toutefois, cela demande au calculateur de connaître au préalable le comportement de la structure, ce qui n'est généralement pas le cas. Le choix d'un critère de raffinement (ou de déraffinement) automatique est donc primordial. Les solutions sont

nombreuses et dépendent alors des aspects que l'on souhaite privilégier dans la simulation. Comme nous l'avons vu au paragraphe 3.1.3, de nombreuses méthodes adaptatives reposent sur un estimateur d'erreur à posteriori. Ce critère basé sur des considérations essentiellement mathématiques compare généralement les solutions obtenues sur les deux discrétisations hiérarchiques et nécessite de ce fait un calcul préalable sur la discrétisation fine. Une telle approche est difficilement concevable dans notre stratégie où l'économie du temps de calcul joue un rôle capital. C'est pourquoi, l'utilisation d'un simple critère basé sur la physique du phénomène nous a paru ici plus pertinent.

3.3.2.1 Critère de raffinement

Le critère physique en question repose sur la déformation plastique des éléments. La décision de basculer du maillage grossier au maillage fin est prise lorsque la déformation plastique maximale repérée sur $\tilde{\Omega}_g$ atteint un certain seuil ε_{pl}^{max} (que l'on appellera seuil de raffinement) préalablement fixé.

$$\max_{e \in \tilde{\Omega}_l} \left[\max_{i,j} (\varepsilon_{pl}^e)_{ij} \right] \leq \varepsilon_{pl}^{max} \quad (3.3.18)$$

où $(\varepsilon_{pl}^e)_{ij}$ désigne les composantes du tenseur de déformation plastique sur l'élément e . Ce tenseur étant calculé aux points d'intégrations, le maximum portera naturellement sur l'ensemble de ces points. Outre sa simplicité de mise en oeuvre dans l'algorithme de dynamique explicite, le choix de ce critère se justifie par le fait que l'erreur commise au moment du raffinement est principalement due à la non-linéarité, c'est-à-dire à la plasticité de la structure. Comme nous le verrons par la suite à travers plusieurs exemples d'applications, la valeur du seuil ε_{pl}^{max} sera très influente vis-à-vis de la précision de la solution.

3.3.2.2 Critère de déraffinement

Bien que cela soit plus rarement réalisé, on peut envisager à l'inverse, la transition d'une discrétisation fine à une discrétisation plus grossière dès que l'intérêt porté à la structure s'amointrit. Ceci peut être le cas lorsque les déformations subies par la zone sensible structure n'évoluent quasiment plus. On peut dans ce cas utiliser un indicateur basé non plus sur la déformation, mais sur la vitesse de déformation maximale mesurée au sein du domaine de calcul :

$$\max_{e \in \tilde{\Omega}} \left[\max_{i,j} \left(\frac{(\dot{\varepsilon}^e)_{ij}}{\dot{\varepsilon}_0} \right) \right] \leq \dot{\varepsilon}^{max} \quad (3.3.19)$$

où $(\dot{\varepsilon}^e)_{ij}$ désigne les composantes du tenseur taux déformation, et $\dot{\varepsilon}_0$ est une valeur de référence arbitraire. Le terme $\dot{\varepsilon}^{max}$ est alors appelé seuil de déraffinement.

3.3.3 Implémentation de la méthode

À l'heure actuelle, la méthode a été développée sous le code de calcul CASTEM [LF98]. Son intégration à RAD2RAD dans le cadre des calculs multi-domaines de crash, nécessite nous le verrons quelques développements spécifiques coté RADIOSS. Bien que ces développements soient en cours, ils ne permettent pas encore d'assurer la phase d'équilibrage indispensable à l'évolution de maillage. Nous présenterons dans un premier temps l'algorithme général tel que celui-ci a été mis en oeuvre sous CASTEM, puis nous aborderons

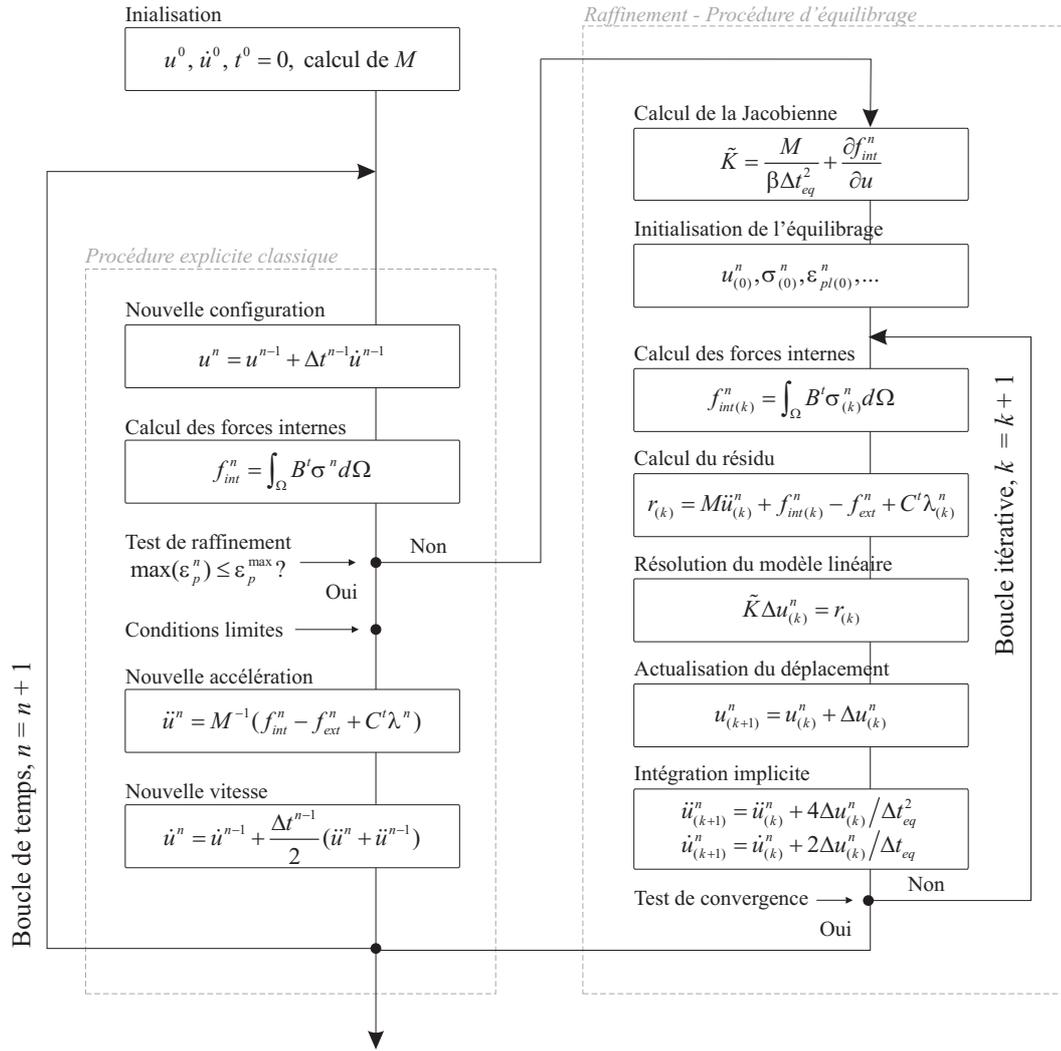


Figure 3.13: Algorithme général du calcul à discrétisation variable.

ensuite la manière d'intégrer ce même algorithme dans le cadre du couplage RAD2RAD-RADIOSS.

3.3.3.1 Algorithme général en grands déplacements et grandes rotations

La figure 3.13 montre comment la procédure d'équilibrage, de type implicite se substitue aux pas de temps explicites traditionnels dès que la décision de raffinement est prise. L'algorithme a été développé dans le cadre des grandes déformations, en utilisant une formulation Lagrangienne réactualisée. Nous nous focaliserons sur l'implémentation de la boucle d'équilibrage, en détaillant un cycle du processus itératif.

Calcul des forces internes : La première étape concerne le calcul des forces internes, analogue à celui réalisé en explicite. Partant de l'incrément de déplacement $\Delta u_{f(k-1)}^s$ calculé à l'itération précédente, on en déduit un incrément de déformation $\Delta E_{f(k-1)}^s$, et par conséquent le tenseur de déformation de Green-Lagrange défini sur la configuration

$X_{f(k-1)}^s$. On utilise ensuite la loi de comportement pour obtenir l'incrément de contrainte $\Delta\sigma_{f(k-1)}^s$. La contrainte de Piola-Kirshhoff est ainsi connue à l'itération k mais sur la configuration $X_{f(k-1)}^s$:

$$\Pi_{f(k)}^s = \sigma_{f(k-1)}^s + \Delta\sigma_{f(k-1)}^s \quad (3.3.20)$$

Le passage à la contrainte de Cauchy sur la nouvelle configuration est réalisé à partir de transformation (1.1.10) :

$$\sigma_{f(k)}^s = \frac{1}{\det(F)} F \Pi_{f(k)}^s F^t \quad (3.3.21)$$

où F est le gradient de déformation défini par (1.1.8). Les forces internes peuvent alors être calculées, puis assemblées de manière tout à fait classique sur la configuration courante. L'approche Lagrangienne réactualisée nécessite pour cela, une mise à jours préalable des coordonnées nodales en utilisant l'incrément de déplacement $\Delta u_{f(k-1)}^s$.

$$f_{int,f(k)}^s = \int_{\Omega} B^t \sigma_{f(k)}^s d\Omega \quad (3.3.22)$$

Résolution du problème linéarisé : Une fois déterminées, les forces internes peuvent être additionnées aux forces d'inerties et aux forces externes pour calculer le résidu. De plus, les conditions limites étant ici traitées par la méthode des multiplicateurs de Lagrange, il convient de les prendre en compte en y ajoutant un terme $C^t \lambda_{(k)}^s$:

$$r_{(k)} = M_f \ddot{u}_{f(k)}^s + f_{int,f(k)}^s - f_{ext,f}^s + C^t \lambda_{(k)}^s \quad (3.3.23)$$

L'incrément de déplacement courant peut ensuite être calculé en résolvant le système linéaire (3.3.24). Par défaut, ce système est résolu sous CASTEM en utilisant une méthode directe de Crout.

$$\tilde{K} \Delta u_{f(k)}^s = -r_{(k)} \quad (3.3.24)$$

Mise à jour des champs cinématiques : La dernière étape de la boucle consiste à réactualiser les trois champs cinématiques $u_f^s, \dot{u}_f^s, \ddot{u}_f^s$. En utilisant les relations du schéma d'intégration, on obtient :

$$u_{f(k+1)}^s = u_{f(k)}^s + \Delta u_{f(k)}^s \quad (3.3.25)$$

$$\ddot{u}_{f(k)}^s = \ddot{u}_{f(k-1)}^s + \Delta \ddot{u}_{f(k-1)}^s, \quad \text{avec} \quad \Delta \ddot{u}_{f(k-1)}^s = \Delta u_{f(k)}^s / \beta \Delta t_{eq}^2 \quad (3.3.26)$$

$$\dot{u}_{f(k)}^s = \dot{u}_{f(k-1)}^s + \Delta \dot{u}_{f(k-1)}^s, \quad \text{avec} \quad \Delta \dot{u}_{f(k-1)}^s = \gamma \Delta u_{f(k-1)}^s / \beta \Delta t_{eq} \quad (3.3.27)$$

Le schéma adopté ici étant de type implicite non amorti, nous utiliserons les paramètres $\beta = 1/4$ et $\gamma = 1/2$.

3.3.3.2 Adaptation aux calculs multi-domaines RAD2RAD-RADIOSS

Bien que la méthode ait été implémentée sous CASTEM dans un cadre mono-domaine, l'algorithme proposé sur la figure 3.15 s'inscrit dans le cadre général de la résolution multi-domaines, tel que celui-ci doit apparaître dans RAD2RAD. Nous nous placerons ici dans le cas d'un calcul à pas de temps asynchrones portant sur deux sous-domaines Ω_1 et Ω_2 . On supposera alors que le raffinement a lieu sur le domaine Ω_2 , comme cela est

illustré sur la figure 3.5.

La philosophie de calcul est alors la suivante : le logiciel RADIOSS ne disposant pas de mailleur automatique, les différents maillages (pour un même sous-domaine) seront préalablement construits avant le calcul. Pour un sous-domaine donné, les différents niveaux de discrétisation devront par conséquent être définis dans un seul et même modèle RADIOSS. Outre le fait de gérer les problèmes d'interfaces entre sous-domaines, le couplage avec RAD2RAD permettra d'assurer les fonctions suivantes relatives au changement de discrétisation :

- ⇒ Gérer l'instant de bascule suivant le critère de raffinement (ou de déraffinement).
- ⇒ Gérer l'activation et la désactivation des maillages par envoi de messages aux sous-domaines.

Nous remarquerons que la phase d'équilibrage, préalable à l'activation de chaque sous-domaine au cours du calcul devra être réalisée par RADIOSS. En effet, RAD2RAD n'ayant accès qu'aux grandeurs nodales, et non aux points d'intégrations de l'élément, l'initialisation de la contrainte sur le nouveau maillage, via les valeurs interpolées de l'ancien maillage, ne peut être établie dans ce code. Une fois de plus nous profiterons donc des fonctionnalités présentes dans RADIOSS, en réutilisant notamment certains de ses récents développements dédiés aux calculs implicites.

Traitement particulier des interfaces de liaison Le changement de discrétisation demande une attention particulière au niveau des interfaces entre sous-domaines. Tout d'abord, si l'on s'intéresse au raffinement du sous-domaine j , celui-ci s'accompagne assez logiquement d'un enrichissement de l'espace des multiplicateurs de Lagrange sur les surfaces S_Λ (cf. Figure 3.14) et de la matrice de liaison C_j associée. D'autre part, le rééquilibrage d'un sous-domaine, nécessite la prise en compte des forces de liaisons des sous-domaines connexes. Il convient pour cela d'adjointre la contribution $C_j^t \lambda$ au résidu (3.3.2), où le vecteur des multiplicateurs de Lagrange λ est issu du problème condensé :

$$H\lambda^s = B^s \quad \text{avec} \quad B^s = \sum_{i=1}^{n_{ssd}} C_i \left(p\ddot{u}_i^s + \frac{\Delta t_i}{2} \ddot{u}_{i,free}^s \right) \quad (3.3.28)$$

Nous supposons qu'au cours de l'équilibrage, l'interface ne subisse pas trop de déformation, de manière à pouvoir considérer la matrice H constante. Néanmoins, le second membre B^s , faisant apparaître les accélérations *free* des noeuds frontières, le problème condensé (3.3.28) devra s'inscrire au processus itératif, afin de prendre en compte l'évolution de ces accélérations au cours du rééquilibrage.

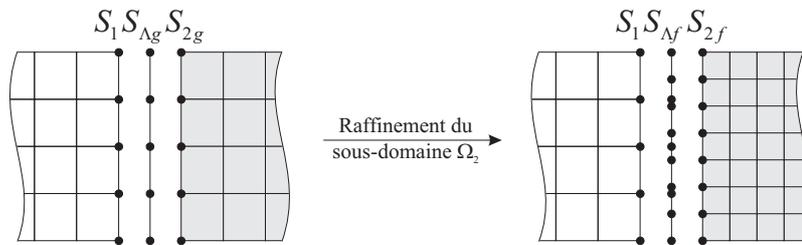


Figure 3.14: Enrichissement de l'espace des multiplicateurs de Lagrange.

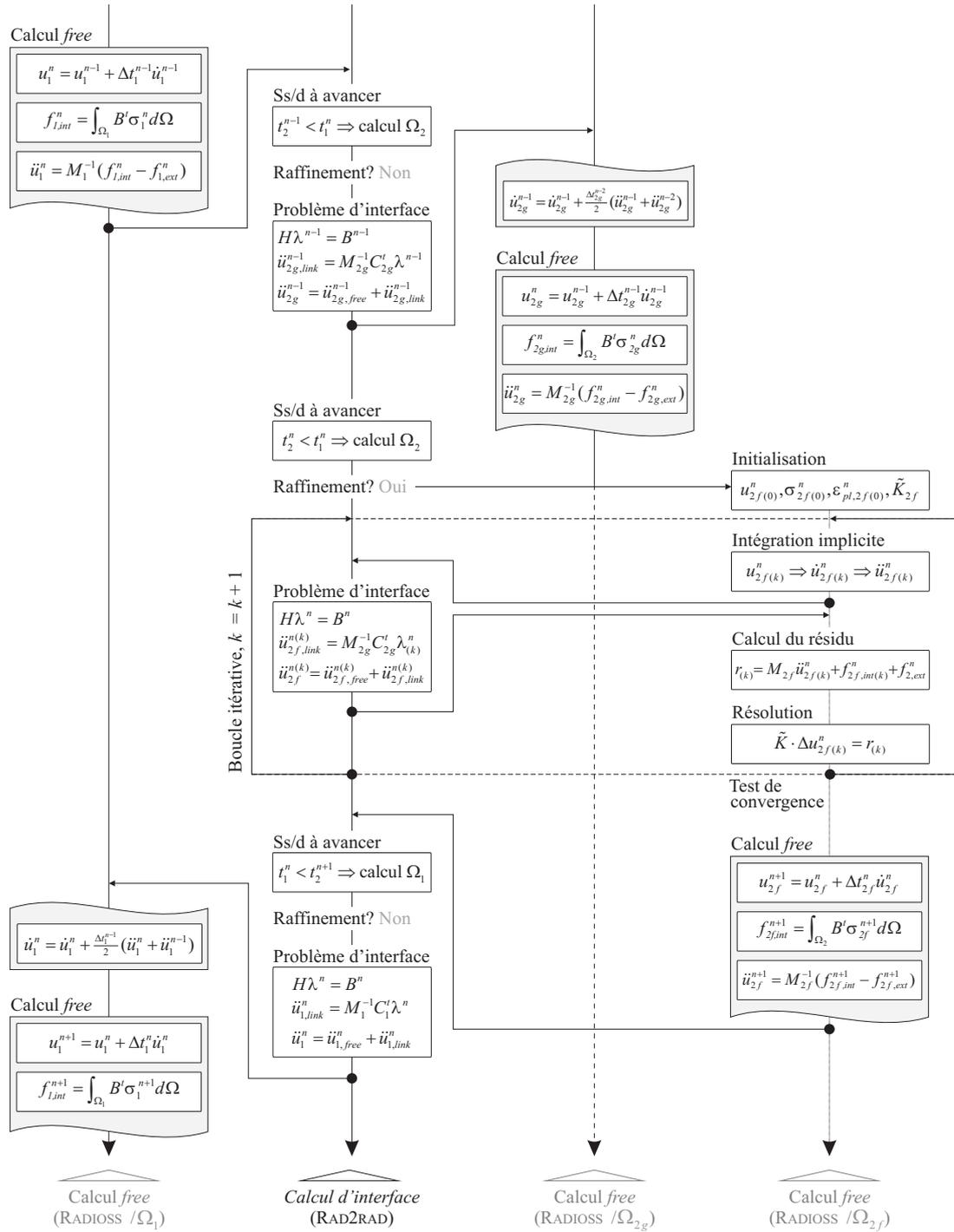


Figure 3.15: Schéma simplifié des communications RAD2RAD-RADIOSS lors d'un calcul multi-domaines à discrétisation variable.

3.4 Exemples numériques

Nous terminerons ce chapitre en illustrant et en validant la méthode proposée à travers deux exemples d'applications. Plusieurs études de sensibilité viendront se greffer à ces exemples afin d'analyser l'influence de certains paramètres vis-à-vis de l'efficacité de la méthode. L'ensemble des calculs feront ici état d'un changement de maillage unique (passage d'un maillage grossier à un maillage fin). Les résultats présentés dans cette partie ont été réalisés sur machine séquentielle avec le code éléments finis CASTEM dans lequel a été développé l'algorithme décrit précédemment.

3.4.1 Cas 2D : Poutre en flexion

Intéressons nous tout d'abord au problème bidimensionnel, en déformations planes, d'une poutre encastree/libre de longueur l et d'épaisseur h (cf. Figure 3.16a). Cette poutre est soumise à un chargement extérieur f_{ext} dont l'évolution en fonction du temps est représentée sur la figure 3.16b.

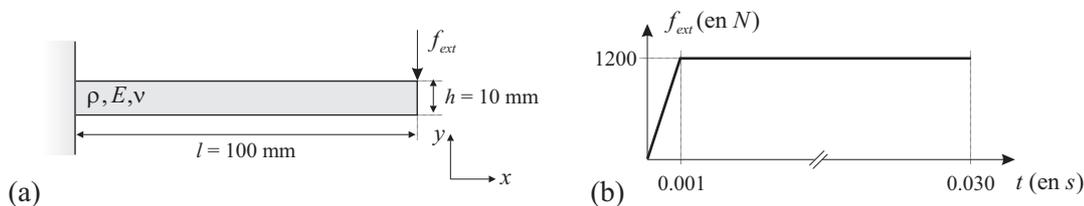


Figure 3.16: Cas test de poutre en flexion : (a) Caractéristique géométrique de la poutre - (b) Chargement appliqué à l'extrémité de la poutre.

Nous supposons ici un milieu homogène et isotrope sujet à de grandes déformations, et adopterons un matériau élasto-plastique de module de Young E , de masse volumique ρ et de coefficient de poisson ν . Le comportement plastique sera imposé par une loi de Johnson-Cook modifiée :

$$\sigma_{eq} = (A + B\varepsilon_{eq}^p) \cdot \left(1 + C \ln \left(\frac{\dot{\varepsilon}_{eq}}{\dot{\varepsilon}_{eq}^0}\right)\right) \quad (3.4.1)$$

où la contrainte équivalente d'écoulement σ_{eq} , la limite élastique A , le facteur d'écrouissage B et le coefficient d'écrouissage p sont donnés dans le tableau 3.1. Le terme $\dot{\varepsilon}_{eq}^0$ désigne la vitesse de déformation plastique de référence et C la sensibilité à la vitesse de déformation plastique. Cette sensibilité sera ici négligée.

Caractéristiques matériaux			
$\rho = 7.80 \times 10^3 \text{ kg/m}^3$	$\nu = 0.3$	$A = 4.00 \times 10^8 \text{ N/m}^2$	$p = 0.557$
$E = 2.10 \times 10^{11} \text{ N/m}^2$	-	$B = 5.48 \times 10^8 \text{ N/m}^2$	$C = 0.0$

Tableau 3.1: Cas test de poutre en flexion : caractéristiques matériaux.

Le domaine est discrétisé en éléments quadrangles à 4 noeuds (QUA4). L'objectif est de basculer au cours du calcul, du maillage grossier (4×40 éléments) vers le maillage fin (8×80 éléments) (cf. Figure 3.17). Ce raffinement s'accompagne d'une diminution du

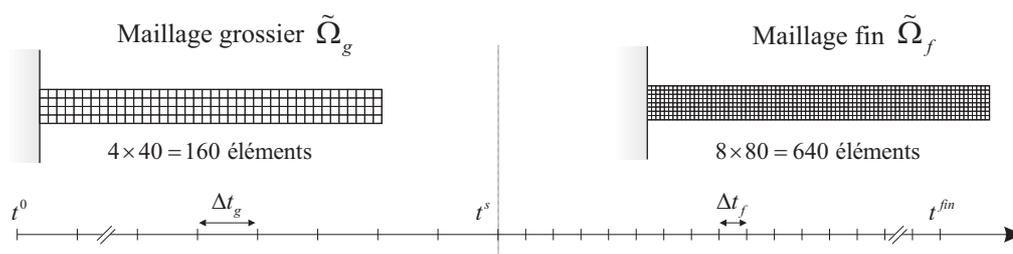


Figure 3.17: Changement de discrétisation de la poutre.

pas de temps de stabilité qui passe alors de $\Delta t_g = 4.82 \times 10^{-6} s$ à $\Delta t_f = 2.41 \times 10^{-6} s$. Nous imposerons à ce pas de temps un coefficient de sécurité suffisamment grand pour qu'il puisse être supposé constant durant chacune des deux phases de calcul.

3.4.1.1 Validation de la démarche d'équilibrage

Le but est ici de montrer l'intérêt de la phase de rééquilibrage sur le résultat. On compare pour cela les solutions obtenues sur les quatre calculs suivants :

- **Cas ①** : Calcul à maillage unique sur le maillage fin (calcul de référence).
- **Cas ②** : Calcul à maillage unique sur le maillage grossier.
- **Cas ③** : Calcul avec changement de maillage sans rééquilibrage (transfert seul).
- **Cas ④** : Calcul avec changement de maillage avec rééquilibrage.

Le critère de raffinement, égal à la déformation plastique maximale sera ici fixé à $\varepsilon_{pl}^{max} = 1\%$, lequel conduit à un basculement de maillage au temps $t^s = 0.0051s$. Le paramètre de recouvrement sera fixé à $\eta = 10$. On représente sur la figure 3.18 les vitesses transverses calculées à l'extrémité de la poutre pour chacune des quatre configurations. Nous noterons tout d'abord l'intérêt du rééquilibrage entre les cas ③ et ④ indispensable à la stabilisation des calculs après le changement de maillage. En effet, en absence d'équilibrage on observe une courbe de vitesses fortement perturbée mais surtout une énergie de déformation démesurée (cf. Figure 3.19) après la reprise des calculs sur la discrétisation fine.

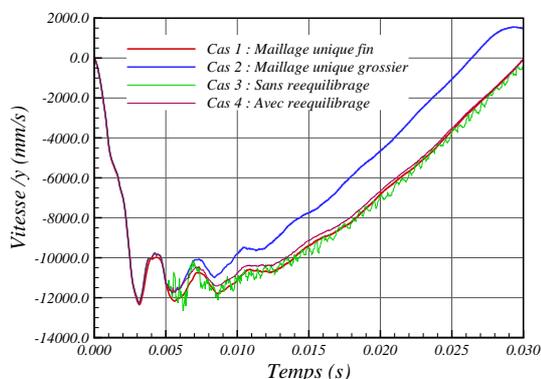


Figure 3.18: Évolution de la vitesse transverse à l'extrémité de la poutre.

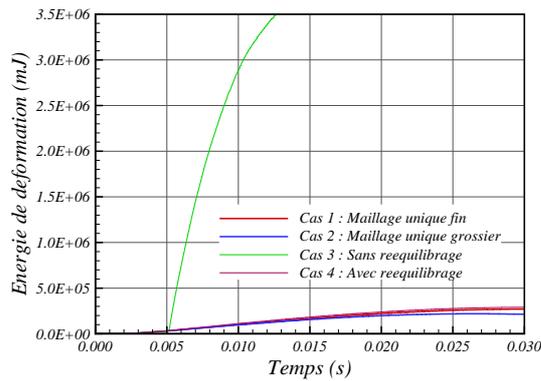


Figure 3.19: Évolution de l'énergie de déformation.

D'autre part, nous remarquerons que la solution obtenue avec notre méthode est assez logiquement bornée par les solutions de référence fine et grossière. Elle converge même assez nettement vers la solution fine, et ce d'autant plus que le changement de maillage aura lieu précocement dans la simulation. D'ailleurs, on constate d'après la figure 3.20 que la distribution de contrainte équivalente de Von Mises représentée sur la déformée finale (facteur d'amplification = 2) fait également état d'une bonne corrélation entre la solution à discrétisation variable et la solution de référence obtenue sur maillage fin.

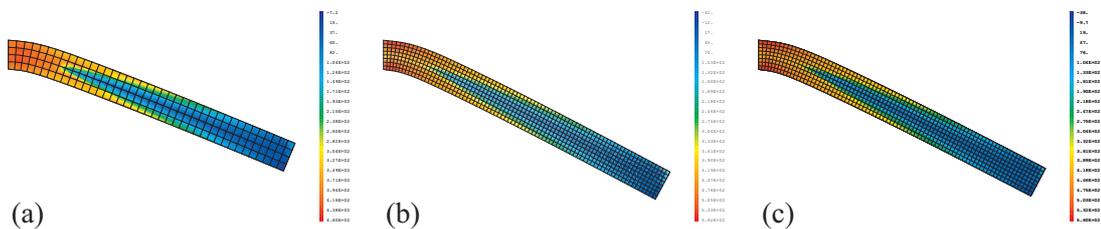


Figure 3.20: Comparaison des distributions de contraintes équivalentes : (a) Calcul à maillage unique grossier - (b) Calcul à maillage unique fin - (c) Calcul à discrétisation variable.

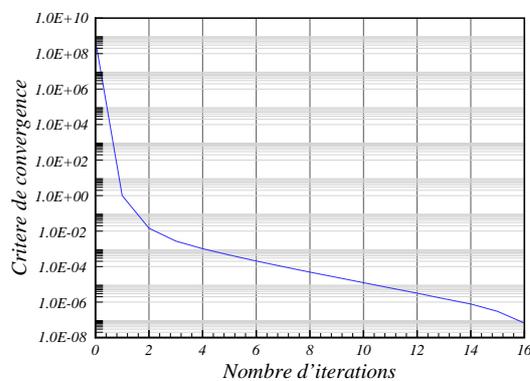


Figure 3.21: Convergence de la procédure itérative de Newton-Raphson.

Nous remarquerons enfin d'après la figure 3.21 que le nombre d'itération de Newton nécessaire au rééquilibrage après basculement de maillage reste très faible. Par exemple, pour un seuil fixé à 10^{-7} sur les résidus (3.3.13), une quinzaine d'itérations est suffisante pour atteindre la convergence. Ceci conforte l'idée que l'efficacité de la méthode n'est pas remise en cause par le surcoût de calcul engendré par le rééquilibrage.

3.4.1.2 Influence du recouvrement

Comme nous l'avons décrit au paragraphe 3.2.4, la stratégie mise en oeuvre fait intervenir un paramètre de recouvrement η caractérisant la longueur du pas de temps « implicite » utilisé pour le rééquilibrage. Nous profitons de cet exemple académique pour étudier la sensibilité de la solution à ce paramètre quelque peu influant.

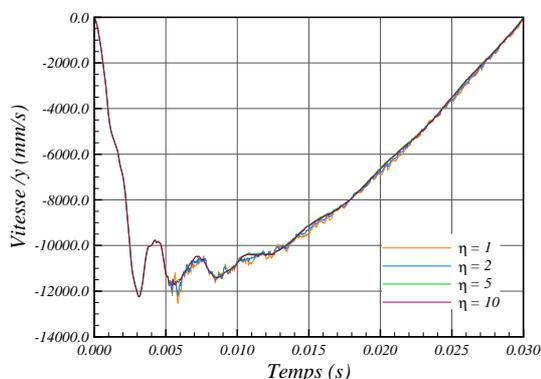


Figure 3.22: Évolution de la vitesse transverse pour différentes valeurs de η .

La figure 3.22 montre les vitesses transverses obtenues pour quatre valeurs différents de η . On constate ici que pour $\eta = 1$, le pas de temps d'équilibre Δt_{eq} alors égal au pas de temps de stabilité du maillage fin Δt_f est visiblement insuffisant pour assurer le bon rééquilibrage de la structure. La vitesse reste dans ce cas fortement bruitée après le changement de discrétisation.

Cette perturbation peut être interprétée par l'approximation (3.3.5) qui consiste à projeter les champs cinématiques issus du calcul grossier au début du pas de temps d'équilibre. Le déséquilibre qu'engendre cette approximation semble alors trop proche de l'instant de bascule pour être « totalement » résorbé avant la reprise du schéma explicite (à l'extrême, la valeur $\eta = 0$ conduirait à un simple transfert de champs sans rééquilibrage comme présenté au cas ③ du paragraphe précédent). Évidemment l'ajout d'un faible amortissement numérique (par exemple, $\gamma = 0.55$) sur quelques pas de temps explicites succédant au raffinement permet de lisser la solution et de s'affranchir d'un tel bruit. Toutefois nous préférons recourir ici à un paramètre de recouvrement plus élevé. On constatera alors que si la solution semble être très sensible entre $\eta = 1$ et $\eta = 5$, elle ne l'est quasiment plus pour les valeurs supérieures. Cependant, au plus le recouvrement sera important, au plus le nombre d'itération d'équilibre nécessaire pour atteindre la convergence le sera également. La valeur $\eta = 10$ semble ici conduire à une taille pertinente du pas de temps d'équilibre.

3.4.1.3 Influence du critère de raffinement

Intuitivement, on comprend que le gain apporté par la méthode sera d'autant plus important que le raffinement aura lieu tardivement et que l'erreur commise restera acceptable. On se propose pour cela d'analyser les résultats de calculs obtenus pour différentes valeurs du critère de raffinement, c'est-à-dire pour différents instants du basculement de maillage. Le paramètre de recouvrement restera ici fixé à $\eta = 10$.

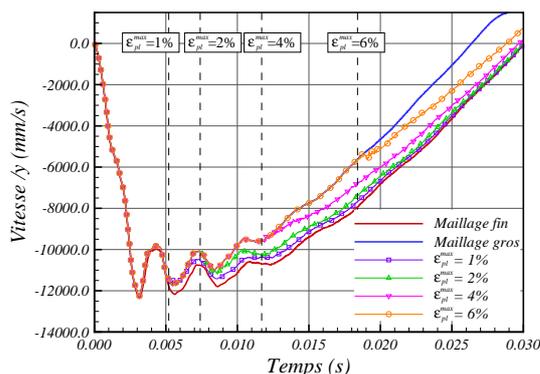


Figure 3.23: Évolution de la vitesse transverse pour différentes valeurs de ε_{pl}^{max} .

La figure 3.23 illustre toujours la vitesse transverse calculée à l'extrémité de la poutre, mais cette fois pour différentes valeurs de ε_{pl}^{max} . Les courbes ainsi obtenues sont toujours bornées par les solutions de calcul à maillage unique, fin et grossier, qui constituent en quelque sorte les cas extrêmes où ε_{pl}^{max} serait nul ou égal à 1. Certes, cette étude montre la dégradation inéluctable des résultats au fur et à mesure que le raffinement apparaît tardivement, mais la solution « raffinée » semble à chaque fois rejoindre la solution fine de référence. Ceci constitue indéniablement un résultat encourageant de la démarche.

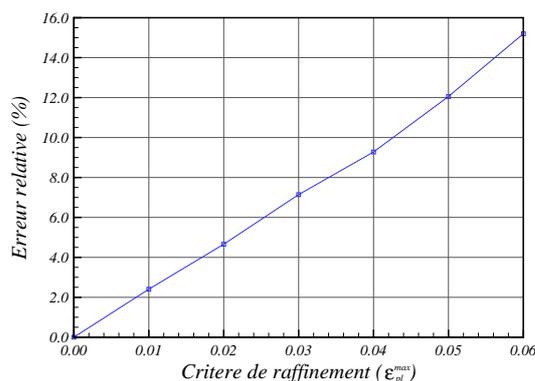


Figure 3.24: Évolution de l'erreur relative avec le critère de raffinement ε_{pl}^{max} .

D'autre part, on peut chercher à estimer l'erreur commise par le calcul à discrétisation variable. On choisit pour cela de mesurer l'écart de flèche maximale u_y^{max} par rapport à

la solution de référence \hat{u}_y^{max} calculée sur le maillage fin.

$$err = \frac{|\hat{u}_y^{max} - u_y^{max}|_{x=l}}{|\hat{u}_y^{max}|_{x=l}} \quad (3.4.2)$$

Les résultats sont présentés sur la figure 3.24 ; on observe une augmentation quasiment linéaire de cette erreur relative avec le critère de raffinement. Bien entendu, cette étude est à mettre en parallèle avec le gain en temps de calcul apporté pour les différentes valeurs de ε_{pl}^{max} (cf. Tableau 3.2). Ainsi, pour obtenir un gain en temps de calcul de près de 50%, nous devons accepter une erreur relative de près de 15%.

De toute évidence le gain en temps CPU se paie par la perte de précision. La difficulté est donc de trouver le bon compromis pour allier ces deux aspects du calcul. Le choix du bon critère de raffinement s'avère alors primordial. Celui-ci restera néanmoins très spécifique au cas étudié ; suivant l'application considérée, le même taux de déformation plastique peut être atteint à des instants très disparates et plus ou moins avancés de la simulation.

Calcul à discrétisation unique		Calcul à discrétisation variables			
Fin	grossier	$\varepsilon_{pl}^{max} = 1\%$	$\varepsilon_{pl}^{max} = 2\%$	$\varepsilon_{pl}^{max} = 4\%$	$\varepsilon_{pl}^{max} = 6\%$
983 s	197 s	858 s	800 s	670 s	505 s

Tableau 3.2: Temps de calcul pour différentes valeurs du critère de raffinement.

3.4.1.4 Influence du maillage

Un autre paramètre influant que nous avons cherché à analyser concerne la finesse des maillages utilisés. Pour cette étude, nous avons choisi de toujours garder le même ratio de raffinement (égal à 2), puis d'observer l'influence du maillage initial sur l'efficacité de la méthode. Trois cas seront étudiés ici :

- ↪ **Cas ①** : Basculement d'un maillage 2×20 à un maillage 4×40 éléments.
- ↪ **Cas ②** : Basculement d'un maillage 4×40 à un maillage 8×80 éléments.
- ↪ **Cas ③** : Basculement d'un maillage 6×60 à un maillage 12×120 éléments.

Pour chacune de ces configurations, nous avons tracé sur la figure 3.25, le gain en temps CPU (en %), en fonction de l'erreur calculée sur la flèche maximale (en %). On observe alors des tendances très nettes pour chacune des trois courbes. En supposant qu'une erreur de 10% soit tolérée, le gain en temps de calcul sera près de 80% dans le cas ③ mais à peine 25% dans le cas ①. Ceci se comprend aisément dans la mesure où au plus on raffine au plus on converge vers la solution « exacte » du problème. Par conséquent, l'erreur finale sera en toute logique d'autant plus faible que le maillage initial sera fin.

On montre ainsi que pour un même ratio de raffinement, la méthode est d'autant plus efficace que les maillages mis en jeu sont fins. A titre d'exemple, nous pourrions enfin vérifier le fonctionnement de cette approche pour un ratio de raffinement plus élevé ; basculement d'un maillage 4×40 à un maillage 12×120 . Avec un critère fixé à $\varepsilon_{pl}^{max} = 5\%$,

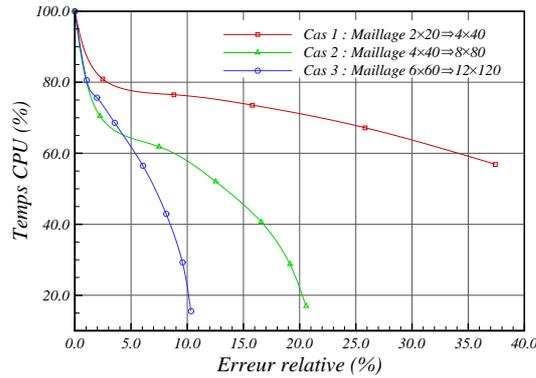


Figure 3.25: Courbes précision/temps de calcul pour différents maillages.

le calcul mène à une erreur sur la flèche maximale de 14.1% contre 16.1% dans le cas du raffinement précédent avec un ratio égal à 2 (pour un même maillage initial). Nous remarquerons alors que dans ce cas précis, un tel ratio de raffinement n'est pas forcément avantageux puisqu'il mène à un gain en temps CPU de 46% contre un gain de 80% pour le ratio de 2.

3.4.2 Cas 3D : Flambage de prolonge

Appliquons maintenant notre méthode à un cas plus proche des problématiques industrielles en matière de crash. On considère pour cela un essai sous-système sur une prolonge de berceau d'un véhicule de la gamme PSA PEUGEOT-CITROËN[®]. Afin de reproduire des conditions de chargements de dynamiques rapides analogues à celles du choc, nous choisissons d'encasturer cette prolonge dans sa partie avant, puis d'appliquer une vitesse initiale de 15m/s (soit 54km/h) aux noeuds de la face arrière. Une masse ajoutée de 500kg sera répartie sur ces mêmes noeuds de manière à simuler l'inertie de l'arrière du véhicule au moment de l'impact.

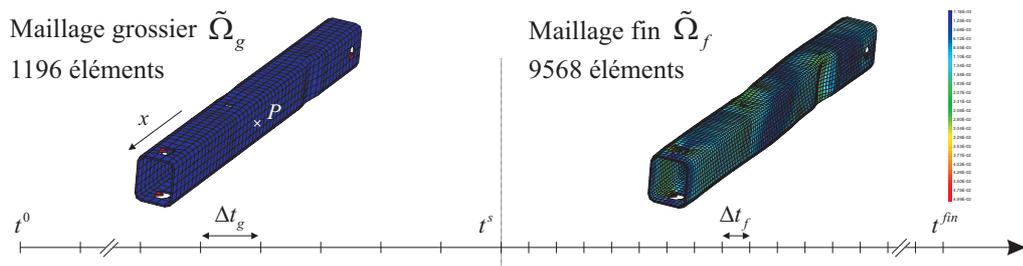


Figure 3.26: Changement de discrétisation de la prolonge (Déformation plastique cumulée sur géométrie déformée).

Nous supposons toujours la structure comme étant soumise à de grandes déformations. Comme précédemment, nous nous placerons dans le cas d'un matériau homogène et isotrope dont le comportement élasto-plastique est régi par la loi de Johnson-Cook (3.4.1). Les différentes caractéristiques matériaux sont données dans le tableau 3.3.

Caractéristiques matériaux			
$\rho = 7.82 \times 10^3 \text{ kg/m}^3$	$\nu = 0.3$	$A = 4.00 \times 10^8 \text{ N/m}^2$	$p = 0.800$
$E = 2.10 \times 10^{11} \text{ N/m}^2$	-	$B = 5.00 \times 10^8 \text{ N/m}^2$	$C = 0.0$

Tableau 3.3: Cas test de flambage de prolonge : caractéristiques matériaux.

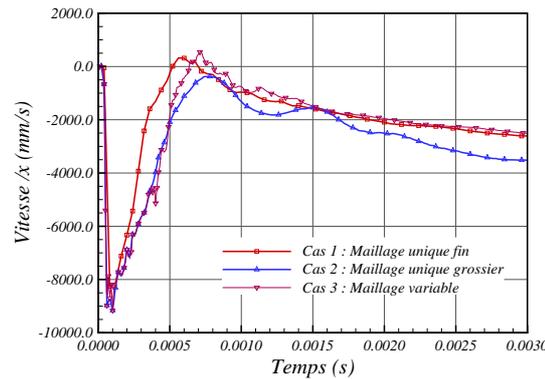
Concernant la discrétisation, nous utiliserons ici des éléments volumiques à 8 noeuds (CUB8). Au cours du raffinement chaque élément sera donc découpé en 8 ; nous passerons ainsi d'un maillage à 1196 éléments à un maillage à 9568 éléments. Bien que le nombre d'éléments soit ici augmenté d'un facteur 8, le ratio de raffinement basé sur la longueur caractéristique de l'élément sera toujours égal à 2. Les pas de temps explicites seront alors fixés à $\Delta t_g = 1.8 \times 10^{-7} \text{ s}$ sur le maillage grossier et à $\Delta t_f = 0.9 \times 10^{-7} \text{ s}$ sur le maillage fin.

3.4.2.1 Validation du calcul avec raffinement

Nous comparons toujours les résultats de calculs obtenus avec notre méthode de discrétisation variable à ceux obtenus sur maillages uniques. Les trois configurations étudiées seront repérées de la manière suivante :

- ⇨ **Cas ①** : Calcul à maillage unique sur le maillage fin (calcul de référence).
- ⇨ **Cas ②** : Calcul à maillage unique sur le maillage grossier.
- ⇨ **Cas ③** : Calcul avec basculement de maillage et rééquilibrage.

La géométrie étant globalement plus lourde et les calculs plus coûteux, nous n'avons pas reproduit l'ensemble des études de sensibilité relatives au paramètre de recouvrement η faites dans le cas de la poutre. Nous choisirons ici ce paramètre comme étant égal à $\eta = 20$. Le critère de raffinement sera quant à lui fixé à $\varepsilon_{pl}^{max} = 5\%$.


Figure 3.27: Évolution de la vitesse axiale au point P .

On représente sur la figure 3.27, la vitesse axiale d'un noeud de post-traitement P situé sur la peau externe de la prolonge, au niveau du premier lobe de flambement (cf. Figure 3.26). On observe tout d'abord que les courbes de vitesses obtenues dans les cas ① et ② présentent un écart déjà fortement marqué à l'instant de bascule $t \cong 0.35 \text{ ms}$. Par

conséquent, dans le cas du calcul avec raffinement, la reprise du calcul sur le maillage fin est initialisée avec une erreur conséquente qu'il est impossible de corriger par la suite. Néanmoins, il est encourageant de constater qu'après la transition de maillage, la vitesse suit une évolution très proche, de celle obtenue avec le calcul fin de référence. Cette tendance se retrouve d'ailleurs très nettement sur les courbes d'énergies de déformations, lesquelles constituent un indicateur plus macroscopique de la solution (cf. Figure 3.28) ; le passage à un maillage plus fin rendant la structure plus souple, celle-ci accumule en toute logique moins d'énergie.

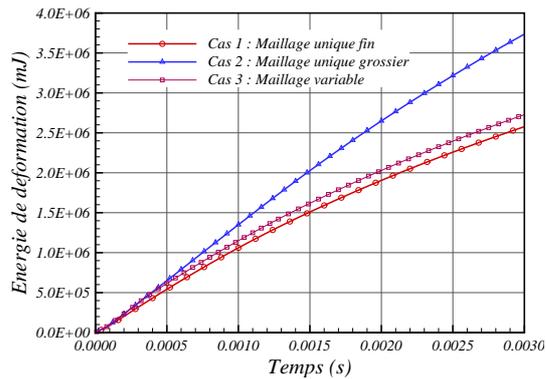


Figure 3.28: Évolution de l'énergie de déformation.

Enfin, nous pouvons étudier la distribution de contrainte dans la pièce, en traçant la contrainte équivalente de Von Mises sur la géométrie déformée (cf. Figure 3.29). Nous constaterons ici encore la bonne concordance des résultats du calcul à double maillages avec ceux du calcul à maillage fin unique.

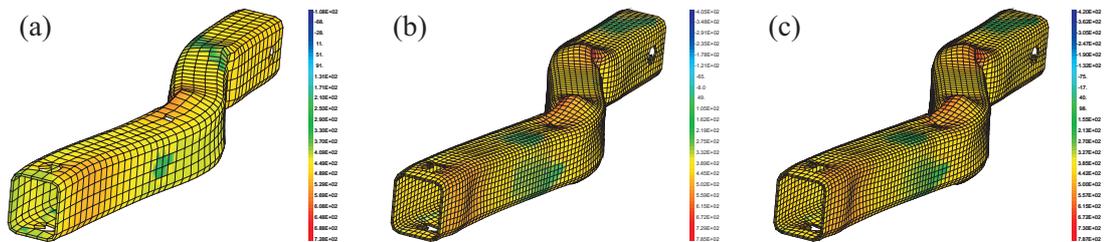


Figure 3.29: Comparaison des distributions de contraintes équivalentes : (a) Calcul à maillage unique grossier - (b) Calcul à maillage unique fin - (c) Calcul à discrétisation variable.

3.4.2.2 Confrontation précision - gain de temps CPU

Afin de mesurer le gain apporté par la méthode, nous avons choisi de réaliser plusieurs calculs en faisant varier la valeur du critère de raffinement. La figure 3.30 illustre l'évolution du déplacement axial, mesuré au point de post-traitement P (cf. Figure 3.26), pour quatre valeurs de ε_{pl}^{max} comprises entre 2% et 10%. Ces déplacements sont comparés à ceux obtenus par les calculs à discrétisation unique fin et grossier. Nous remarquerons une fois de plus, l'important écart entre ces deux calculs de référence qui, rappelons-le, joue inéluctablement en défaveur de notre approche.

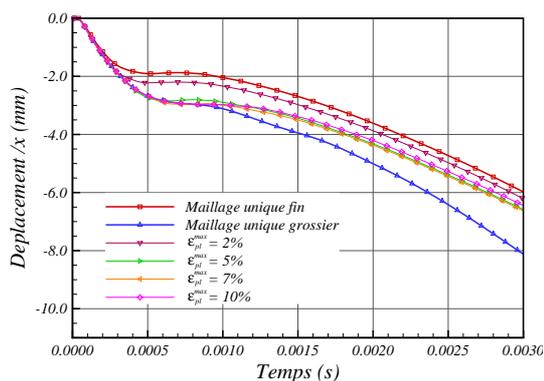


Figure 3.30: Évolution du déplacement axial au point P pour différentes valeurs du critère de raffinement ε_{pl}^{max} .

D'autre part, la figure 3.31 représente l'évolution du temps CPU en fonction du critère de raffinement utilisé. Il s'agit ici d'un gain relatif par rapport au temps de référence obtenu par le calcul fin à maillage unique. Naturellement, au plus le critère sur la déformation plastique sera élevé, au plus le temps d'activation du nouveau maillage sera tardif et au plus le gain sera important.

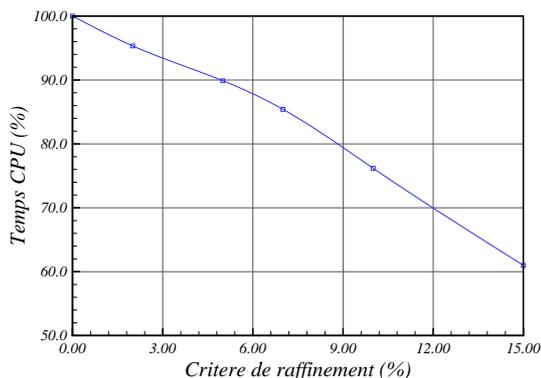


Figure 3.31: Évolution du temps CPU avec le critère de raffinement.

Comme dans le cas de la poutre, nous pouvons évaluer l'erreur commise par rapport au calcul de référence sur maillage fin. Celle-ci sera estimée cette fois à partir du déplacement final (suivant x), calculé au point P . La confrontation de cette erreur au temps CPU ne permet cependant pas ici de dégager une tendance générale, dans la mesure où la solution obtenue avec le critère $\varepsilon_{pl}^{max}=10\%$ ne conduit pas nécessairement à l'erreur la plus importante. A titre d'exemple, on pourra néanmoins noter qu'une erreur acceptable de 7% sur la solution (obtenue pour $\varepsilon_{pl}^{max}=10\%$) ne mène qu'à un gain d'à peine 25% sur le temps de calcul. Le prix à payer pour obtenir une bonne précision de calcul est ici nettement plus marqué que dans le cas test de la poutre.

Pour conclure cet exemple d'application, nous rappellerons que le gain apporté par notre démarche reste dépendant d'un grand nombre de paramètres (sensibilité au maillage

avant le raffinement, mais aussi dépendance au scénario de la déformation, dépendance à la durée totale simulée...etc). Même si ce gain semble assez limité dans certaines configurations, il faut garder à l'esprit que cette approche s'inscrit dans un cadre plus général de calcul multi-domaines. Par conséquent, le bénéfice sera accru par l'utilisation de la discrétisation en espace et en temps la plus adéquate à chaque sous-structure.

Conclusions et perspectives

Cette thèse prolonge les récents développements réalisés en dynamique explicite sur la décomposition de domaines dédiés aux calculs de crash automobile. L'originalité de cette décomposition qui découle du caractère multi-échelles du choc, repose sur la volonté de modéliser chaque partie de la structure de la manière la plus pertinente possible. La méthode proposée dans [Her02] est basée sur une technique de *Schur duale* ; elle assure le collage « permanent » des sous-domaines par l'introduction d'un espace des multiplieurs de Lagrange aux interfaces et gère l'utilisation de pas de temps propre à chacun d'eux. Les présents travaux de recherches étendent alors les résultats de B. Herry dans deux directions principales : la gestion des non-linéarités de contact entre sous-domaines et l'introduction d'une approche multi-échelles dite à discrétisation variable.

Dans cette stratégie de calcul multi-domaines, l'extension pour la prise en compte des liaisons « ponctuelles » de type contact unilatéral, s'avère indispensable à la simulation numérique du crash où le phénomène d'impact est omniprésent. La méthode proposée qui consiste à repousser la problématique de contact à l'intérieur de chaque sous-structure repose cette fois sur un formalisme de décomposition *primale*, où l'on cherche à vérifier de manière exacte la continuité cinématique aux interfaces. La nature multi-pas-de-temps du problème constitue un point essentiel de la démarche afin de disposer du pas de temps le plus économique sur chacune des parties en contact. Plusieurs exemples, des plus élémentaires aux plus représentatifs de la réalité industrielle, ont permis de valider et montrer l'efficacité de la méthode. Évidemment, le gain en temps de calcul dépend fortement de la discrétisation des sous-domaines en contact et s'avère d'autant plus important que la différence de pas de temps entre les deux sous-domaines est grande. De plus, si la simulation numérique du choc piéton présentée dans ce mémoire fait état d'un gain d'environ 55% pour une simple décomposition en deux sous-domaines, l'économie devrait assez logiquement être accrue par un découpage plus fin du modèle.

La seconde partie de ce travail concernant le développement d'une approche multi-échelles en dynamique non-linéaire, consistait à adapter la discrétisation du modèle au cours du calcul. La difficulté majeure qui était de marier stabilité et précision après la reprise des calculs sur la nouvelle discrétisation a été levée par une méthode de rééquilibrage de type « implicite » au moment du basculement de maillage. Bien que cette stratégie soit destinée à raffiner localement les sous-domaines issus de la décomposition, elle n'a été implémentée puis testée que dans un cadre mono-domaine. Les exemples académiques sur lesquels elle a été illustrée ont révélé que le gain en temps de calcul restait étroitement lié à l'erreur que l'on s'accorde sur la solution. Les résultats demeurent néanmoins très encourageants pour mettre en oeuvre les développements propres aux calculs couplés dans RAD2RAD-RADIOSS. L'extension de la démarche au calcul multi-domaines devrait

justifier de son intérêt sur des cas tests industriels de tailles plus importantes.

Outre ces deux grandes problématiques, divers travaux complémentaires mais ne constituant pas de véritables activités de recherches, ont été réalisés sur le code de couplage RAD2RAD, pour en améliorer ses performances. La mise en oeuvre de plusieurs cas tests ont notamment permis d'isoler certains dysfonctionnements et de rendre le code plus robuste. Par ailleurs, certains développements ont permis d'étendre ces fonctionnalités. Le couplage d'éléments coques et d'éléments poutres utilisé pour les calculs de prédimensionnement à par exemple été implémenté puis validé à travers plusieurs exemples. De même, l'optimisation du solveur et des constructions matricielles a permis une diminution d'un facteur 5 des temps de calcul consommés par RAD2RAD lors du problème d'interface. Cependant, même si à l'heure actuelle, le code est opérationnel et a été industrialisé sur les serveurs de test PSA PEUGEOT-CITROËN[®] certains travaux sont encore à réaliser pour parvenir à un outil de calcul massivement utilisé sur les plateaux projets.

Parmi les perspectives à court terme, on peut évoquer la parallélisation de RAD2RAD et notamment de la résolution du problème condensé à l'interface, dont le coût pourra s'avérer non-négligeable sur les problèmes de grande dimension. La parallélisation des solveurs MUMPS et SuperLU actuellement testés dans leurs versions séquentielles devrait inéluctablement conduire à un gain substantiel du temps de calcul. D'autres précieux développements pourront être apportés pour améliorer les fonctionnalités de ce code de couplage. A titre d'exemple, on peut citer la prise en compte du collage surfacique entre éléments solides. Bien que cette fonctionnalité ne soit pas indispensable, elle peut rapidement devenir d'une importante utilité lorsque la décomposition nécessite un découpage de pièces volumiques.

De plus, une réflexion portant sur la stratégie de découpe à adopter par l'ingénieur devrait contribuer à accroître les performances de calcul. Savoir où placer une interface de couplage et quelle modélisation adopter sur tel ou tel sous-domaine est sans doute l'un des éléments clés quant à l'efficacité de la méthode. Si pour des applications simples la sous-structuration est souvent évidente, elle le devient beaucoup moins dès que les modèles de calcul se complexifient. Différentes stratégies peuvent être envisagées suivant que l'on adoptera une découpe très macroscopique du véhicule (quelques sous-domaines seulement) ou au contraire une découpe très fine de chaque organe fonctionnel. Même si cette dernière paraît être la plus efficace, elle engendre une augmentation du nombre de noeuds d'interface et par conséquent conduit à un coûteux problème d'interface pouvant compromettre l'efficacité du calcul. Ceci étant, bien que l'avantage du calcul multi-domaines ne soit plus à démontrer, son emploi pourra néanmoins être discutable si le gain en temps de calcul devient inférieur à 20%. Le manque de savoir faire dans ce domaine ne permettant pas d'anticiper le profit en matière de temps CPU, la mise à disposition d'un indicateur de gain par comparaison au calcul mono-domaine paraît essentielle. On peut dès lors imaginer une estimation basée sur le nombre de cycles économisés par sous-domaine et sur la taille du sous-domaine en question.

D'un point de vu plus pragmatique, même si celle-ci accélère le temps de restitution des calculs, force est de constater que l'approche multi-domaines complique généralement la tâche de l'ingénieur. La mise en données encore peu conviviale s'avère de toute évidence plus complexe et source d'erreur de modélisation. Le développement d'un préprocesseur

(voir d'un postprocesseur) adapté sera donc indispensable pour assister l'utilisateur dans sa démarche de sous-structuration et ainsi rendre l'outil industriel pleinement utilisable.

Les perspectives à plus long terme sont pour la plupart relatives à la méthode de basculement de maillage développé dans ce mémoire. Comme nous l'avons déjà largement discuté, le gain en temps CPU reste fortement lié au critère de raffinement (ou de déraffinement) ainsi qu'à la précision de calcul attendue. Le choix du critère reste de surcroît un élément capital et mérite sans doute une analyse plus approfondie que le simple seuil de déformation plastique utilisé en premier lieu. Autrement dit, même si le principal objectif qui était d'assurer un changement de maillage stable est atteint, de nombreux travaux restent à effectuer sur la détermination de l'instant de bascule offrant le meilleur compromis précision/temps de calcul. Une approche basée sur un estimateur d'erreur pourrait par exemple être envisagée pour optimiser le temps CPU en fonction de l'erreur que le calculateur est prêt à accepter sur la solution.

Par ailleurs, bien que la méthode repose actuellement sur une utilisation successive de maillages préalablement définis, l'emploi d'un remaillage automatique par sous-domaines pourrait être envisagé sans compromettre le bien-fondé de cette approche. Enfin, pour conclure ces nombreuses perspectives qui découlent de ce travail, nous évoquerons le basculement automatique de schéma explicite-implicite pour lequel notre démarche constitue un premier avancement. L'intérêt d'un tel développement est considérable dans la simulation du crash où l'emploi de l'implicite peut s'avérer indispensable. A titre d'exemple, citons les phases de mise en gravité du véhicule avant choc et de retour élastique après choc qui ne peuvent être simulées en explicite sans occasionner des temps de calcul prohibitifs. Le caractère inconditionnellement stable de l'implicite permettrait de simuler ces phases pseudo-statiques et très peu non-linéaires à moindre coût en utilisant un pas de temps nettement plus grand que celui utilisé durant la phase de choc à proprement parlé.

Bibliographie

- [ABLTV01] P. ALART, M. BARBOTEU, P. LE TALLEC et M. VIDRASCU : Additive schwarz method for non-symmetric problems : application to frictional multicontact problems. *In* N. DEBIT, M. GARBEY, R. HOPPE, J. PÉRIAUX, D. KEYES et Y. KUZNETSOV, éditeurs : *Thirteenth International Conference on Domain Decomposition Methods*, pages 3–13, Lyon, France, October 2001. CIMNE.
- [AC91] P. ALART et A. CURNIER : A mixed formulation for frictional contact problems prone to newton like solution methods. *Computer Methods in Applied Mechanics and Engineering*, 92(3):353–375, 1991.
- [ADLK03] P.R. ARMESTOY, I.S. DUFF, J.Y. L’EXCELLENT et J. KOSTER : Multifrontal massively parallel solver (MUMPS version 4.3) users’ guide. Rapport technique, ENSEEIHT-IRIT, July 2003.
- [BAV01] M. BARBOTEU, P. ALART et M. VIDRASCU : A domain decomposition strategy for nonclassical frictional multi-contact problems. *Computer Methods in Applied Mechanics and Engineering*, 190:4785–4803, 2001.
- [BCCL96] C. BLANZÉ, L. CHAMPANEY, J.Y. COGNARD et P. LADEVÈZE : A modular approach to structure assembly computations. *Engineering Computations*, 13:15–32, 1996.
- [BCDV06] B. BOUREL, A. COMBESURE et L. DI VALENTIN : Handling contact in multi-domain simulation of automobile crashes. *Finite Elements in Analysis and Design*, 42:766–779, 2006.
- [BCF⁺94] M. BELLET, J.L. CHENOT, L. FOURMENT, E. MASSONI et P. MONTMITONNET : Lois de frottement et mesure du frottement. *In Séminaire de plasticité : Éléments finis et mise en forme des métaux*, volume 1, Sophia Antipolis, Septembre 1994. Ecole Nationale Supérieure des Mines de Paris, CEMEF.
- [BD05] H. BEN DHIA : Approches locales-globales, méthode arlequin. *In Septième colloque national en Calcul des structures*, volume 2, pages 527–532, Giens, Mai 2005. CSMA, AFM, UTC, CNRS, Hermes.
- [BDR05] H. BEN DHIA et G. RATEAU : The arlequin method as a flexible engineering design tool. *International Journal Numerical Methods of Engineering*, 62: 1442–1462, 2005.
- [Bel83] T. BELYTSCHKO : An overview of semidiscretization and time integration procedures. *Computational Methods for Transient Analysis*, pages 1–65, 1983.

- [BF91] F. BREZZI et M. FORTIN : *Mixed and hybrid finite element methods*. Springer-Verlag, New York, 1991.
- [BLM00] T. BELYTSCHKO, W.K. LIU et B. MORAN : *Nonlinear Finite Elements for Continua and Structures*. Wiley, England, 2000.
- [BMP94] C. BERNARDI, Y. MADAY et A.T. PATERA : A new nonconforming approach to domain decomposition : the mortar element method. In H. BRÉZIS et J.L. LIONS, éditeurs : *Nonlinear partial differential equations and their applications, Collège de France Seminar, Volume XI*, numéro 299 de Pitman Research Notes in Mathematics, pages 13–51, Paris, 1994. Pitman Advanced Publishing Program.
- [BN91] T. BELYTSCHKO et M.O. NEAL : Contact-impact by the pinball algorithm with penalty and lagrangian methods. *International Journal Numerical Methods of Engineering*, 31:547–572, 1991.
- [BN97] O. BESLIN et J. NICOLAS : A hierarchical functions set for predicting very high order plate bending modes with any boundary conditions. *Journal of Sound and Vibration*, 202, 1997.
- [Bra77] A. BRANDT : Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.
- [BSL85] T. BELYTSCHKO, P. SMOLINSKI et W.K. LIU : Stability of multi-time step partitioned integrators for first-order finite elements systems. *Computer Methods in Applied Mechanics and Engineering*, 49:281–297, 1985.
- [BYM79] T. BELYTSCHKO, H.J. YEN et R. MULLEN : Mixed methods for integration. *Computer Methods in Applied Mechanics and Engineering*, 17/18:259–275, 1979.
- [CCL99] L. CHAMPANEY, J.Y. COGNARD et P. LADÈVEZE : Modular analysis of assemblages of three-dimensional structures with unilateral contact conditions. *Computers and Structures*, 73:249–266, 1999.
- [CD00] L. CHAMPANEY et D. DUREISSEIX : Calcul de structures et parallélisme : un bilan et quelques développements récents. mécanique et industries. *Mécanique et Industries*, 1(1):43–46, 2000.
- [CGLC05a] P. CAVIN, A. GRAVOUIL, T. LUBRECHT et A. COMBESURE : Automatic energy conserving space-time refinement for linear dynamic structural problems. *International Journal Numerical Methods of Engineering*, 64:304–321, 2005.
- [CGLC05b] P. CAVIN, A. GRAVOUIL, T. LUBRECHT et A. COMBESURE : Efficient fem calculation with predefined precision through automatic grid refinement. *Finite Elements in Analysis and Design*, 41:1043–1055, 2005.
- [CGLC05c] P. CAVIN, A. GRAVOUIL, T. LUBRECHT et A. COMBESURE : Etude de la stabilité d’une méthode à raffinement espace-temps automatique (star-method). In *Septième colloque national en Calcul des structures*, volume 1, pages 21–32, Giens, Mai 2005. CSMA, AFM, UTC, CNRS, Hermes.
- [Cha96] L. CHAMPANEY : *Une nouvelle approche modulaire pour l’analyse d’assemblage de structures tridimensionnelles*. Thèse de doctorat, Laboratoire de Mécanique et Technologie, ENS de Cachan, Université Paris 6, 1996.

-
- [Dan98] W.J.T. DANIEL : A study of the stability of subcycling algorithms in structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 156:1–13, 1998.
- [DEG⁺99] J.W. DEMMEL, S.C. EISENSTAT, J.R. GILBERT, X.S. LI et J.W.H LIU : A supernodal approach to sparse partial pivoting. *SIAM J. Matrix analysis and Applications*, 20(3):720–755, 1999.
- [DF00] D. DUREISSEIX et C. FARHAT : A numerically scalable domain decomposition method for the solution of frictionless contact problems. *International Journal Numerical Methods of Engineering*, 00:1–26, 2000.
- [DFS] Z. DOSTAL, A. FRIEDLANDER et S.A. SANTOS : Solution of coercive and semicoercive contact problems by feti domain decomposition.
- [DGL03] J.W. DEMMEL, J.R. GILBERT et X.S. LI : Super lu user’s guide. Rapport technique, EECS Department, University of California, Berkeley, October 2003.
- [EHBVM05] A. EL HRAIECH, H. BOROUCHEKI, P. VILLON et L. MOREAU : Transfert consistant de champs. In *17ème Congrès Français de Mécanique*, Troyes, 2005.
- [Fau03] V. FAUCHER : *Méthodes de réduction en dynamique explicite multi-échelles pour l’analyse des structures complexes sous impact*. Thèse de doctorat, Laboratoire de Mécanique et Technologie, ENS de Cachan, Université Paris 6, 2003.
- [FC03] V. FAUCHER et A. COMBESCURE : A time an space mortar method for coupling linear modal subdomains and non-linear subdomains in explicit structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 192:509–533, 2003.
- [FCM95] C. FARHAT, P. CHEN et J. MANDEL : A scalable lagrange multiplier baser domain decomposition method for time-dependent problems. *International Journal Numerical Methods of Engineering*, 38:3831–3853, 1995.
- [Fos95] I. FOSTER : *Designing and Building Parallel Programs*. Addison-Wesley, 1995. <http://www-unix.mcs.anl.gov/dbpp/>.
- [FR91] C. FARHAT et F.X. ROUX : A method of finite elements tearing and interconnecting and its parallel solution algorithm. *International Journal Numerical Methods of Engineering*, 32:1205–1227, 1991.
- [FR94] C. FARHAT et F.X. ROUX : Implicit parallel processing in structural mechanics. In Oden J.T, éditeur : *Computational Mechanics Advances*, volume 2, pages 1–124. North Holland, 1994.
- [GC01] A. GRAVOUIL et A. COMBESCURE : Multi-time-step explici-implicit method for non-linear structural dynamics. *International Journal Numerical Methods of Engineering*, 50:199–225, 2001.
- [GC03] A. GRAVOUIL et A. COMBESCURE : Multi-time-step and two-scale domain decomposition method for non-linear structural dynamics. *International Journal Numerical Methods of Engineering*, 58:1545–1569, 2003.
- [GLT90] R. GLOWINSKI et P. LE TALLEC : Augmented lagrangian interpretation of the nonoverlapping schwarz alternating method. In T. F. CHAN, R. GLO-
-

- WINSKI, J. PÉRIAUX et O. B. WIDLUND, éditeurs : *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–231, SIAM, Philadelphia, 1990.
- [Gra00] A. GRAVOUIL : *Méthode multi-échelles en temps et en espace avec décomposition de domaines pour la dynamique non-linéaire des structures*. Thèse de doctorat, Laboratoire de Mécanique et Technologie, ENS de Cachan, Université Paris 6, 2000.
- [HDVC02] B. HERRY, L. DI VALENTIN et A. COMBESCURE : An approach to the connection between subdomains with non-matching meshes for transient mechanical analysis. *International Journal Numerical Methods of Engineering*, 55:973–1003, 2002.
- [Her02] B. HERRY : *Développement d'une approche multiéchelles parallèle pour la simulation de crash automobile*. Thèse de doctorat, Laboratoire de Mécanique et Technologie, ENS de Cachan, Université Paris 6, 2002.
- [Hil00] P. HILD : Numerical implementation of two nonconforming finite element methods for unilateral contact. *Computer Methods in Applied Mechanics and Engineering*, 184:99–123, 2000.
- [HL78a] T.J.R. HUGHES et W.K. LIU : Implicit-explicit finite elements in nonlinear transient analysis : Implementation and numerical example. *Journal of Applied Mechanics*, 45:375–378, 1978.
- [HL78b] T.J.R. HUGHES et W.K. LIU : Implicit-explicit finite elements in nonlinear transient analysis : Stability theory. *Journal of Applied Mechanics*, 45:371–374, 1978.
- [HPT79] T.J.R. HUGHES, K.S. PISTER et R.L. TAYLOR : Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17/18:159–182, 1979.
- [Hug83] T.J.R. HUGHES : Analysis of transient algorithms with particular reference to stability behavior. *Computational Methods for Transient Analysis*, pages 67–155, 1983.
- [Hug01] T.J.R. HUGHES : *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 2001.
- [Iro70] B.M. IRONS : *Applications of a Theorem on Eigenvalues to Finite Element Problems*. University of Wales, Department of Civil Engineering, Swansea, U.K., 1970. CR/132/70.
- [KP93] S. KACOU et I.D. PARSONS : A parallel multigrid method for history-dependent elastoplasticity computations. *Computer Methods in Applied Mechanics and Engineering*, 108:1–21, 1993.
- [Lad85] P. LADEVÈZE : Algorithmes adaptés aux calculs vectoriels et parallèles pour des méthodes de décomposition de domaine. In PLURALIS, éditeur : *Troisième Colloque Tendances actuelles en calcul des structures*, pages 893–907, 1985.
- [Lad96] J. LADEVÈZE : *Mécanique non linéaire des structures : Nouvelle approche et méthodes de calcul non incrémentales*. Hermes, Paris, 1996.

-
- [LB82] W.K. LIU et T. BELYTSCHKO : Mixed-time implicit-explicit finite elements for transient analysis. *Computers and Structures*, 15:445–450, 1982.
- [LF98] E. LE FICHOUX : *Présentation et utilisation de CASTEM 2000*, Janvier 1998. Rapport ENSTA / LME.
- [Lio88] P.L. LIONS : On the schwarz alternating method. i. In R. GLOWINSKI, G.H. GOLUB, G.A. MEURANT et J. PÉRIAUX, éditeurs : *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1988.
- [Lio90] P.L. LIONS : On the schwarz alternating method. iii : A variant for non-overlapping subdomains. In Chan T. F., Glowinski R., Périaux J. et Widlund O. B., éditeurs : *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 202–223, SIAM, Philadelphia, 1990.
- [LSd98] Ls-dyna theoretical manual. Livermore, California, May 1998.
- [New59] N.M. NEWMARK : A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, pages 67–94, 1959. ASCE.
- [NSP04a] L. NOELS, L. STAINIER et J.P. PONTHOT : Combined implicit-explicit algorithms for crashworthiness analysis. *International Journal of Impact Engineering*, 30:1161–1177, 2004.
- [NSP04b] L. NOELS, L. STAINIER et J.P. PONTHOT : Combined implicit/explicit time-integration algorithms for the numerical simulation of sheet metal forming. *Journal of Computational and Applied Mathematics*, 168:331–339, 2004.
- [NSPB02] L. NOELS, L. STAINIER, J.P. PONTHOT et J. BONINI : Combined implicit-explicit algorithms for non-linear structural dynamics. *Revue européenne des éléments finis*, 11:565–591, 2002.
- [PF00] K.C. PARK et C.A. FELIPPA : A variational principle for the formulation of partitioned structural systems. *International Journal for Numerical Methods in Engineering*, 47:395–418, 2000.
- [PH90] I.D. PARSONS et J.F. HALL : The multigrid method in solid mechanics : part i - algorithm description and behaviour. *International Journal for Numerical Methods in Engineering*, 29:719–737, 1990.
- [Rad05] Radioss theory manual. Sophia Antipolis, Septembre 2005.
- [Rat03] G. RATEAU : *Méthode Arlequin pour les problèmes mécaniques multi-échelles*. Thèse de doctorat, Laboratoire de mécanique des sols, structures et matériaux, Ecole Centrale Paris, 2003.
- [Rou90] F.X. ROUX : Méthodes de résolution par sous-domaines en statique. *La recherche Aérospatiale*, 1:37–48, 1990.
- [Rét05] J. RÉTHORÉ : *Méthode éléments finis étendus en espace et en temps : application à la propagation dynamique des fissures*. Thèse de doctorat, Laboratoire de Mécanique du Contact et des Solides, INSA Lyon, 2005.
- [SBG96] B. SMITH, P. BJORSTAD et W. GROPP : *Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
- [Sch69] H.A. SCHWARZ : Über einige abbildungsaufgaben. *Gesammelte mathematische Abhandlungen*, 11:65–83, 1869.
-

- [SEKS99] E. SUN, R. EPPINGER, S. KUPPA et R. SAUL : Development of improved injury criteria for the assessment of advanced automotive restraint systems - ii. NHTSA, Novembre 1999.
- [SS86] Y. SAAD et M. H. SCHULTZ : Gmres : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [VBS02] P. VILLON, H. BOROUCAKI et K. SAANOUNI : Transfert de champs plastiquement admissibles. *C. R. Mecanique*, 330:313–318, 2002.
- [Wro99] M. WRONSKI : Couplage radiois-radiois : Rapport intermédiaire de développement. Eurosim Mecalog Services, Novembre 1999.
- [Zhu86] D.C. ZHU : Development of hierarchical finite element methods at biao. *In Proceedings of the International Conference on Computational Mechanics*, pages 123–128, Tokyo, 1986.
- [ZTJ05] O.C. ZIENKIEWICZ, R.L. TAYLOR et Zhu J.Z. : *The Finite Element Method, 6th Edition*. Elsevier Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford OX28DP, 2005.

FOLIO ADMINISTRATIF

THÈSE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

NOM : BOUREL

DATE de SOUTENANCE : 19 décembre 2006

Prénoms : Benjamin, Louis, Dominique

TITRE : Calcul multi-domaines et approches multi-échelles pour la simulation numérique de crashes automobiles

NATURE : Doctorat

Numéro d'ordre : 2006-ISAL-XXX

École doctorale : MEGA

Spécialité : Mécanique - Génie Mécanique - Génie Civil

Cote B.I.U. - Lyon : T 50/210/19

/ et bis

CLASSE :

RÉSUMÉ :

En dynamique explicite, l'analyse des structures est souvent très coûteuse en temps de calcul à cause du grand nombre de pas de temps nécessaire à la stabilité des modèles. C'est pourquoi, il est crucial de trouver des algorithmes plus performants pour diminuer les temps de calcul. Tel est l'intérêt des méthodes de décomposition de domaine qui consistent à diviser le modèle éléments finis en plusieurs sous-domaines ayant chacun leur propre schéma d'intégration. L'avantage d'une telle approche est d'adopter la discrétisation la plus pertinente et par conséquent le pas de temps le plus économique sur chaque sous-domaine. Cela permet de raffiner la zone sensible d'une structure sans pénaliser le reste du modèle par une discrétisation temporelle trop fine.

Le premier volet de cette thèse concerne la mise au point d'une méthode de gestion du contact unilatéral entre les sous-domaines issus de la décomposition. Ce développement a une grande importance dans les simulations de crash, où l'on ne peut exclure à priori ce type de phénomène. La méthode dite de « la peau artificielle » introduite ici, repose sur la superposition d'une interface de contact (par pénalité) et d'une interface de liaison; elle renvoie ainsi la gestion du contact inter-domaine à un traitement interne au sous-domaine. L'interface de liaison que nous qualifions de « rigide » classe la méthode parmi les techniques de décomposition « primale ». Le caractère multi-échelles lié au traitement des pas de temps multiples et asynchrones sur les différents sous-domaines constituera un point crucial de la méthode.

Le second volet de la thèse traite d'une méthode de discrétisation variable intégrée à notre approche multi-domaines. L'intérêt d'une telle méthode est d'enrichir localement et temporairement la discrétisation du modèle d'étude au cours d'une phase sensible du calcul. La stratégie de « basculement de maillage » étant rendue particulièrement instable par l'utilisation d'un schéma explicite, l'accent sera mis sur le transfert de champs et sur le rééquilibrage de la solution au moment du changement de discrétisation. Le changement d'échelle sera ici automatisé en adoptant un critère physique basé sur la déformation plastique maximale. Pour chacun de ces deux volets, plusieurs exemples d'application viendront illustrer puis valider les méthodes mises en oeuvre. Leurs efficacités seront alors mesurées par comparaison des résultats à ceux issus de l'approche mono-domaine classique.

MOTS-CLÉS : Éléments finis, Dynamique explicite, Décomposition de domaine, Calcul multi-échelles.

Laboratoire(s) de recherche : Laboratoire de Mécanique des Contacts et des Solides

UMR CNRS 5514 - INSA de Lyon

20, avenue Albert Einstein

69621 Villeurbanne Cedex FRANCE

Directeur de thèse : Monsieur le Professeur Alain COMBESURE

Président du jury : JEZEQUEL Louis

Composition du jury : DRAZETIC Pascal
WINKELMULLER Gérard
DI VALENTIN Laurent

PESEUX Bernard
COMBESURE Alain