

**Institut National Polytechnique de Toulouse**

**Laboratoire Génie de Production**

Ecole Nationale d'Ingénieurs de Tarbes

---

**Mémoire d'Habilitation à Diriger des Recherches**

Spécialité Génie Mécanique

Présenté par

**Olivier Pantalé**

le 13 Juillet 2005

---

**Plateforme de prototypage virtuel pour la  
simulation numérique en Grandes Transformations  
Thermomécaniques Rapides**

---

Devant le Jury composé de :

Pr Jean-Michel Bergheau	LTDS - ENI de St Etienne	Rapporteur
Pr Gérard Coffignal	LPMS - ENSAM Paris	Rapporteur
Pr Eric Ragneau	GMA - INSA Rennes	Rapporteur
Pr Serge Caperaa	LGP - ENI Tarbes	
Pr Moussa Karama	LGP - ENI Tarbes	
Dr Ir Jean-Philippe Ponthot	LTAS - MC&T - Université de Liège	



Plateforme de prototypage virtuel pour la simulation  
numérique en Grandes Transformations  
Thermomécaniques Rapides

Olivier Pantalé

13 juillet 2005



# Table des matières

<b>Nota benè</b>	<b>ix</b>
<b>Notations</b>	<b>xi</b>
<b>Introduction et positionnement</b>	<b>1</b>
<b>Informations générales</b>	<b>3</b>
<b>Parcours de recherche</b>	<b>11</b>
<b>1 Modélisation numérique de la coupe des métaux</b>	<b>13</b>
1.1 Introduction et positionnement . . . . .	13
1.2 Mise en oeuvre de modèles numériques de simulation de la coupe . . . . .	14
1.2.1 Approche et problèmes mis en jeu lors de la coupe . . . . .	14
1.2.2 Modèles numériques de coupe . . . . .	15
1.2.3 La modélisation numérique de la coupe au sein du LGP . . . . .	16
1.3 Synthèse des difficultés liées à la modélisation numérique de la coupe . . . . .	22
1.3.1 Choix du formalisme adapté . . . . .	23
1.3.2 Modélisation comportementale . . . . .	23
1.3.3 Calcul parallèle et décomposition en sous-domaines . . . . .	24
1.4 Conclusions et axes de recherche à explorer . . . . .	24
<b>2 Lois de comportement en Grandes Transformations</b>	<b>27</b>
2.1 Formulation des lois de comportement en grandes transformations . . . . .	27
2.1.1 Formulation hypo-élastique en grandes transformations . . . . .	28
2.1.2 Intégration temporelle des équations constitutives . . . . .	31
2.1.3 Lois d'écoulement . . . . .	32
2.2 Le laboratoire d'identification . . . . .	34
2.2.1 Moyens expérimentaux du laboratoire . . . . .	34
2.2.2 Les divers essais expérimentaux développés . . . . .	36
2.3 Identification des lois de comportement . . . . .	40
2.3.1 Procédure d'identification . . . . .	40
2.3.2 Exemple d'identification . . . . .	42

<b>3 Développement du code de calcul explicite DynELA</b>	<b>45</b>
3.1 Pourquoi développer un nouveau code de calcul . . . . .	45
3.2 Intégration numérique explicite . . . . .	46
3.3 Implémentation numérique Orientée-Objets du code de calcul . . . . .	48
3.3.1 Approche Orientée-Objets et avantages pour un code éléments finis . . . . .	48
3.3.2 Description de l'architecture du code <b>DynELA</b> . . . . .	50
3.3.3 Spécificités du code de calcul <b>DynELA</b> . . . . .	52
3.4 Validation numérique du code de calcul . . . . .	54
3.4.1 Striction d'un barreau cylindrique . . . . .	54
3.4.2 Test d'impact de Taylor . . . . .	57
3.4.3 Impact d'un longeron axi-symétrique . . . . .	59
3.5 Parallélisation du code de calcul <b>DynELA</b> . . . . .	61
3.5.1 Standards de programmation parallèle . . . . .	61
3.5.2 Présentation du standard OpenMP . . . . .	62
3.5.3 Mise en oeuvre pour le code <b>DynELA</b> . . . . .	63
3.6 Utilisations du code de calcul <b>DynELA</b> . . . . .	68
<b>4 Plateforme Orientée-Objets de développement</b>	<b>69</b>
4.1 Plateforme de développement Orientée-Objets . . . . .	69
4.1.1 Décomposition spatiale en sous-domaines . . . . .	69
4.1.2 Décomposition temporelle . . . . .	71
4.1.3 Multi-physique . . . . .	73
4.1.4 Développement de l'approche eXtended Finite Element Method . . . . .	74
4.2 Développements d'applications métier . . . . .	75
4.2.1 Application MulPhyDo . . . . .	76
4.2.2 Application DynaCrack . . . . .	78
4.2.3 Application DynaCoupe . . . . .	78
<b>Conclusions, perspectives et projet de recherche</b>	<b>81</b>
<b>Bibliographie</b>	<b>85</b>

# Table des figures

1.1	Description des paramètres de coupe orthogonale . . . . .	15
1.2	Localisation des bandes de cisaillement dans un processus de coupe . . . . .	15
1.3	Conditions aux limites du modèle éléments finis stationnaire . . . . .	17
1.4	Sensibilité de la solution obtenue en fonction du coefficient de frottement . . . . .	18
1.5	Influence de l'usure en cratère de la face de coupe . . . . .	18
1.6	Champs de températures et de contraintes en coupe 3D orthogonale . . . . .	19
1.7	Champ de contraintes en coupe 3D oblique . . . . .	20
1.8	Champ de contraintes en coupe 2D orthogonale . . . . .	21
1.9	Description de l'opération de fraisage . . . . .	21
1.10	Champ de contraintes en fraisage . . . . .	22
2.1	Surface de plasticité . . . . .	29
2.2	Essais expérimentaux en fonction des déformations et taux de déformation . . . . .	34
2.3	Lanceur à gaz du laboratoire . . . . .	35
2.4	Fenêtre principale de l'application imageAnalyser . . . . .	35
2.5	Comparaison entre mesure optique et mesure 3D . . . . .	36
2.6	Essai de Taylor symétrique pour un acier 42CrMo4 . . . . .	37
2.7	Profils de la cible et du projectile dans l'essai d'impact de Taylor symétrique . . . . .	37
2.8	Essai de Taylor direct pour un projectile en aluminium . . . . .	37
2.9	Essai de cisaillement dynamique . . . . .	38
2.10	Essai de traction dynamique . . . . .	39
2.11	Essai d'extrusion conique . . . . .	39
2.12	Essai de rupture dynamique . . . . .	40
2.13	Interface graphique du module <b>Identif</b> . . . . .	41
2.14	Lois de plasticité de Johnson-Cook pour le 42CrMo4 à $\dot{\varepsilon}^p = \dot{\varepsilon}_0^p$ . . . . .	42
3.1	Diagramme UML simplifié de la classe Element . . . . .	50
3.2	Diagramme UML simplifié de la librairie femLibrary . . . . .	51
3.3	Post-processeur graphique de <b>DynELA</b> . . . . .	54
3.4	Géométrie initiale de l'éprouvette pour l'essai de traction . . . . .	55

3.5	Loi d'écoulement pour l'essai de striction . . . . .	55
3.6	Essai de striction d'un barreau cylindrique sur <b>DynELA</b> . . . . .	56
3.7	Variation du diamètre de la zone de striction en fonction de l'élongation pour l'essai de striction . . . . .	56
3.8	Déformations plastiques équivalentes pour le test d'impact de Taylor . . . . .	58
3.9	Données géométriques de l'impact d'un longeron cylindrique . . . . .	59
3.10	Impact d'un longeron cylindrique . . . . .	60
3.11	Comparaison des efforts de réaction pour l'impact d'un longeron cylindrique avec les résultats de J. P. Ponthot [37] . . . . .	60
3.12	Architecture à mémoire partagée (SMP) . . . . .	61
3.13	Compaq Proliant 8000 . . . . .	62
3.14	Parallélisme des tâches sur architecture SMP . . . . .	62
3.15	Description de la classe Jobs . . . . .	65
3.16	Variation du Speedup en fonction du nombre de processeurs . . . . .	67
4.1	Principe général de la méthode XFEM . . . . .	74
4.2	Direction de propagation pour la fissure . . . . .	76
4.3	Développement d'applications métier . . . . .	76
4.4	Assemblage de la puce . . . . .	77
4.5	Conditions aux limites pour la puce . . . . .	77
4.6	Résultats numériques obtenus avec le logiciel MulPhyDo . . . . .	77
4.7	Résultats numériques obtenus avec le logiciel DynaCrack . . . . .	78



# Liste des tableaux

1.1	Paramètres matériau pour la pièce usinée et l'outil . . . . .	17
1.2	Paramètres de la loi d'endommagement du 42CrMo4 . . . . .	21
2.1	Paramètres identifiés de la loi de plasticité de Johnson-Cook pour le 42CrMo4 . . . . .	42
3.1	Paramètres matériau pour l'essai de striction . . . . .	55
3.2	Paramètres matériau pour l'essai de Taylor . . . . .	57
3.3	Comparaison des résultats de <b>DynELA</b> et références pour le test d'impact de Taylor . . . . .	57
3.4	Paramètres matériau pour l'impact d'un longeron . . . . .	59
3.5	Speedup du calcul de $\vec{F}^{int}$ pour les diverses méthodes . . . . .	67
4.1	Nature des quantités selon le type de physique . . . . .	73



# Remerciements

A l'heure de la finalisation de ce mémoire d'Habilitation à Diriger des Recherches, je remercie vivement M. le Professeur *Serge Caperaa*, responsable du groupe de Grandes Transformations Thermomécaniques Rapides (G2TR) au sein de l'équipe Conception Mécanique Assistée de l'Ordinateur (CMAO) qui a su guider et orienter mon travail de recherche au cours de ces 10 dernières années. C'est à ses cotés que j'ai pu apprendre à animer un groupe de recherche.

Je remercie vivement MM. les Professeurs *Jean-Michel Bergheau* du LTDS de l'ENI de Saint-Etienne, *Gérard Coffignal* du LMSM de l'ENSAM de Paris et *Eric Ragneau* du GMA de l'INSA de Rennes d'avoir accepté la tâche d'examiner et de rédiger un rapport sur ce travail alors que leurs emplois du temps sont relativement chargés et qu'ils sont largement sollicités par ailleurs.

Je remercie également *Jean-Philippe Ponthot*, chargé de cours à l'Université de Liège, pour d'une part, avoir contribué au travers des échanges constructifs que nous avons eu depuis 1999 à l'avancement de ces travaux de recherche, et d'autre part, avoir accepté de venir participer à ce Jury de soutenance. Je souhaite que nos collaborations se poursuivent dans le futur et qu'elles soient bénéfiques pour tous.

Je remercie également M. le Professeur *Moussa Karama*, responsable de l'équipe CMAO, qui a accepté de faire partie de ce jury.

J'exprime mes remerciements à MM. les Professeurs *Germain Lacoste* et *Daniel Noyes*, respectivement directeurs de l'Ecole Nationale d'Ingénieurs de Tarbes et du Laboratoire Génie de Production de m'avoir permis d'exercer mon travail d'enseignant-chercheur au sein de leurs structures respectives.

Je me dois également de remercier les étudiants que j'ai eu l'honneur d'encadrer en thèse au cours de mes travaux de recherche : *Jean-Louis Bacaria*, *Chakib Sattouf*, *Laurent Menanteau* et *Ionel Nistor*. Je leur associe également les nombreux étudiants en stage de DEA que j'ai côtoyé et dont j'ai encadré les travaux au cours de ces années. C'est à leur contact que j'ai appris beaucoup sur la manière d'encadrer des projets de recherche.

Un travail de recherche est avant tout un travail d'équipe, je me dois donc d'associer l'ensemble de mes collègues du groupe de travail G2TR et de l'équipe CMAO du laboratoire avec qui j'ai pu avoir des échanges, des discussions et des collaborations très fructueuses sur le plan technique et humain. Je pense plus particulièrement à *Roger Rakotomalala*, qui a dans une première période dirigé mes travaux de thèse puis été un des membres du groupe de travail G2TR qui a vu le jour au milieu des années 90, ainsi qu'à mes collègues de l'équipe CMAO : *Olivier Dalverny* et *Sébastien Mistou*. J'associe également d'une manière plus large tous les membres du Laboratoire Génie de Production, ainsi que les personnels de laboratoires extérieurs avec qui j'ai eu des contacts scientifiquement enrichissants.

Je remercie également *Philippe Stempflé* qui a contribué à ce travail en effectuant la relecture de ce manuscrit, et j'espère être un jour en mesure de lui rendre ce service.

Je pense que ce travail n'aurait jamais pu être entrepris, et surtout mené au point actuel, sans l'aide indirecte mais non moins importante de la communauté scientifique et informatique liée aux logiciels GPL (GNU Public Licence). C'est dans cet esprit que ce travail est effectué, dans l'idée de proposer une avancée dans le domaine de la modélisation numérique en grandes transformations et de faire profiter en retour cette communauté scientifique des travaux réalisés dans le cadre de cette recherche.

Pour terminer, et sur un plan personnel, je tiens à remercier vivement pour sa patience et sa compréhension mon épouse *Christine* de m'avoir supporté pendant mes périodes de doute, et bien entendu, *Adeline* d'accepter que son papa ne soit pas toujours présent pour s'occuper d'elle.

# Nota bene

Dans tout le texte de ce mémoire, les références sont indiquées en utilisant le formalisme suivant :

- Les publications et productions scientifiques personnelles sont notées : **[1,2,3,...]**
- Les références à des travaux externes sont notées : [1,2,3,...]
- Les références à des thèses encadrées sont notées : [I,II,III,...]
- Les références à des mémoires de DEA encadrés sont notées : [i,ii,iii,...]

Certains termes, spécialement dans le cadre du vocabulaire lié à l'informatique étant délicats à traduire en français j'ai choisi de les insérer dans le texte dans leur version originale en italiques, après avoir pris soin, si besoin, de les définir lors de leur première utilisation.



# Notations

D'un point de vue général, il est habituel de constater que dans le domaine de la mécanique, comme dans d'autres domaines sûrement, une des principales difficultés vient de la non homogénéité des notations entre les différents auteurs. Il est alors aisé de rendre complètement incompréhensible la moindre théorie lorsque l'on décide de changer de notation. La notion de notation universelle n'étant pas encore d'actualité (même si certaines conventions peuvent être assimilées à des concepts universels), on présente alors ci-dessous le jeu de notation utilisé tout au long de ce document.

D'une manière générale, les diverses quantités scalaires, vectorielles, matricielles, etc... sont notées comme suit :

- $a$  Scalaire
- $\vec{a}$  Vecteur
- $\mathbf{A}$  Tenseur d'ordre 2 ou matrice
- $\mathbb{A}$  Tenseur d'ordre 4

Le détail des diverses notations utilisées dans ce document est donné ci-dessous :

- $\vec{b}$  Vecteur des forces externes volumiques
- $\mathbf{B}$  Matrice des dérivées des fonctions d'interpolation :  $\mathbf{B} = \partial \mathbf{N} / \partial \vec{x}$
- $\mathbb{C}$  Tenseur de comportement élastique de Hooke
- $\mathbf{D}$  Partie symétrique du tenseur des vitesses de déformation :  $\mathbf{D} = \frac{1}{2} (\mathbf{L} + \mathbf{L}^T)$
- $E$  Module de Young du matériau
- $\mathbf{E}$  Tenseur des déformations de Green-Lagrange
- $\Delta \mathbf{e}$  Tenseur incrément de déformation logarithmique
- $\mathbf{I}$  Tenseur de Kronecker, tenseur unitaire du deuxième ordre
- $\mathbb{I}$  Tenseur unitaire du quatrième ordre
- $f$  Fonction de charge
- $\mathbf{F}$  Tenseur gradient de déformation  $\mathbf{F} = \partial \vec{x} / \partial \vec{X}$
- $\vec{F}^{int}$  Vecteur des efforts internes
- $\vec{F}^{ext}$  Vecteur des efforts externes
- $G$  Module de cisaillement du matériau
- $\mathbf{J}$  Jacobien de la transformation domaine physique/parent
- $K$  Module de compressibilité du matériau
- $\mathbf{K}$  Matrice de rigidité
- $\mathbf{L}$  Tenseur des vitesses de déformation  $\mathbf{L} = \dot{\mathbf{F}} \mathbf{F}^{-1}$
- $\mathbf{M}$  Matrice de masse
- $\mathbf{n}$  Direction de l'écoulement plastique
- $\mathbf{N}$  Matrice des fonctions d'interpolation
- $\mathbf{q}$  Variable d'hérédité dans la formulation élastoplastique
- $\mathbf{R}$  Tenseur de rotation

<b>S</b>	Déviateur du tenseur des contraintes de Cauchy
$T$	Température du matériau
$T_0$	Température de référence du matériau
$T_{melt}$	Température de fusion du matériau
<b>U</b>	Tenseur de déformation pure droit
<b>V</b>	Tenseur de déformation pure gauche
<b>W</b>	Partie anti-symétrique du tenseur des vitesses de déformation : $\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T)$
$\vec{x}$	Coordonnées spatiales d'un point matériel
$\vec{X}$	Coordonnées de référence d'un point matériel
$\alpha$	Backstress
$\overline{\varepsilon^p}$	Déformation plastique équivalente
$\phi$	$\phi = \mathbf{S} - \alpha$
$\vec{\phi}$	Mouvement des points matériels
$\Gamma$	Incrément d'écoulement plastique $\Gamma = \int_{t_n}^{t_{n+1}} \Lambda dt$
$\Gamma_x$	Frontière du domaine de référence
$\Lambda$	Scalaire représentant la norme de l'écoulement plastique
$\nu$	Coefficient de Poisson du matériau
$\Omega_x$	Configuration ou domaine actuel
$\Omega_X$	Configuration ou domaine de référence
$\rho$	Masse volumique du matériau
$\sigma^v$	Limite apparente d'élasticité du matériau
$\sigma_0^v$	Limite d'élasticité initiale du matériau
$\sigma_\infty^v$	Limite asymptotique de plasticité du matériau
<b><math>\sigma</math></b>	Tenseur des contraintes de Cauchy
$\bar{\sigma}$	Contrainte équivalente de von Mises



# Introduction et positionnement

Ce document décrit les phases importantes de mon parcours d'enseignant-chercheur, depuis mes travaux de thèse (1992-1996) jusqu'à ce jour (2005). Ce parcours s'est déroulé dans l'environnement de l'**Ecole Nationale d'Ingénieurs de Tarbes** (ENIT) pour ce qui concerne les activités d'enseignement, et au sein du groupe de **Grandes Transformations Thermomécaniques Rapides** (G2TR) de l'équipe **Conception Assistée par Ordinateur** (CMAO) du **Laboratoire Génie de Production** (LGP) de l'ENIT pour ce qui concerne les activités de recherche. Le terme de "**Grandes Transformations Thermomécaniques Rapides**" couvre les domaines pour lesquels on est confronté à de grandes variations des grandeurs thermomécaniques telles que : les déplacements, les déformations, les vitesses de déformation, les températures, etc... Ces situations sont généralement rencontrées lors d'opérations de fabrication mécanique (mise en forme, usinage, etc...) ou lors d'impacts et de "crash" subis par des structures. Ce sont des phénomènes ayant un caractère fortement non-linéaire et pouvant être très rapides dans le cas des chocs et des crash de structures.

D'un point de vue chronologique, l'ensemble de ces travaux puise son origine dans le développement de modèles numériques de la coupe des métaux réalisés au sein du LGP durant la période 1991-2001 au cours de trois thèses successives. Il ressort principalement de cet ensemble de travaux que le développement de modèles numériques de coupe efficaces nécessite une meilleure approche dans deux axes de développement essentiels. Le premier concerne la prise en compte du comportement du matériau, au travers de la formulation mathématique des lois de comportement volumiques et surfaciques, et l'identification des paramètres de ces lois au travers d'essais expérimentaux adaptés aux types de sollicitations rencontrées lors du processus de coupe. Le second concerne la nécessité de disposer d'un code de calcul entièrement ouvert et maîtrisé afin de pouvoir implémenter des algorithmes numériques spécialement adaptés aux conditions sévères rencontrées lors de la modélisation du processus de coupe.

Par suite, l'objectif des travaux que j'ai entrepris, en collaboration avec les membres du groupe de travail sous la coordination du Professeur Serge Caperaa (responsable du groupe G2TR), est alors de proposer un ensemble d'outils numériques et expérimentaux, intégrés au sein d'une plateforme de prototypage virtuel, en vue de permettre à l'ingénieur spécialiste en Conception Mécanique Assistée par Ordinateur de modéliser numériquement le comportement de ces structures. Ces travaux portent sur les deux volets suivants :

- Sur le plan expérimental, l'objectif principal consiste à élaborer une plateforme d'identification des lois de comportement volumiques et surfaciques pour des matériaux soumis à de fortes sollicitations thermomécaniques rapides. Ces travaux sont à l'origine de la mise en place du laboratoire d'identification, principalement équipé d'un lanceur à gaz de calibre 20mm, puis du développement des activités associées depuis 2001.
- Sur le plan numérique, les travaux s'articulent autour du développement en langage C++ d'un code de calcul totalement ouvert (le logiciel **DynELA**) permettant d'intégrer efficacement des algorithmes et des méthodes de calcul novateurs dans le domaine de la modélisation numérique en grandes transformations. Une extension a ensuite pu être développée dans des

domaines autres que celui de la modélisation numérique en grandes transformations et a permis de proposer des applications métier spécifiques dans le cadre de la simulation numérique de convertisseurs de puissance (le logiciel MulPhyDo [III]) et de la modélisation de la propagation de fissures basée sur la méthode *eXtended Finite Element Method* (XFEM) (le logiciel DynaCrack [IV]).

Le champ d'applications couvert par la plateforme de prototypage virtuel développée au cours de ces travaux n'est pas simplement réductible au seul cadre de la modélisation de la coupe, mais couvre d'une manière plus générale toute application de modélisation numérique des structures en grandes transformations. D'un point de vue rédactionnel, la présentation de ce mémoire s'articule en quatre parties :

- Le premier chapitre est consacré à la description de l'ensemble des recherches menées au sein du laboratoire LGP concernant la modélisation numérique de la coupe des métaux entre les années 1991 et 2001. Ce chapitre présente la problématique liée à la modélisation numérique de la coupe, les développements réalisés au cours de trois thèses successives en utilisant les moyens classiques de modélisation numérique puis les principaux points bloquants qu'il convient de lever afin de mettre au point des modèles de coupe réalistes et prédictifs.
- Le chapitre 2 est consacré à la formulation mécanique et à l'identification des lois de comportement volumiques en grandes transformations. Dans ce chapitre, on présente successivement la formulation mathématique des lois de comportement, les essais expérimentaux développés au sein du laboratoire concernant l'identification du comportement des structures soumises à des sollicitations dynamiques et la procédure d'identification associée.
- Le chapitre 3 est consacré à une présentation des principaux points concernant le développement en langage C++ du code de calcul en grandes transformations **DynELA** développé au laboratoire depuis 1996. Les principales caractéristiques du code sont présentées à travers la formulation cinématique et mécanique en grandes transformations, l'approche orientée objets, l'implémentation numérique, la validation du code au moyen d'un ensemble de cas test dynamiques et la présentation d'une stratégie de parallélisation sur machine à mémoire partagée en utilisant le formalisme OpenMP.
- Enfin, le chapitre 4 est consacré aux apports novateurs et aux réalisations effectuées à partir de la plateforme de développement Orientée-Objets développée au cours de ce travail, notamment au travers de la présentation :
  - du développement d'un module de prototypage virtuel multi-domaine, multi-physique et multi-pas de temps pour le dimensionnement des convertisseurs de puissance [III],
  - du développement et de l'implémentation numérique de la méthode XFEM [IV].

---

# Informations générales

---



## Curriculum Vitae

### Etat civil

Olivier Pantalé  
né le 09 mai 1969 à Cahors (46)  
Marié, 1 enfant



Groupe G2TR - Equipe CMAO  
Laboratoire Génie de Production  
Ecole Nationale d'Ingénieurs de Tarbes  
47 Avenue d'Azereix - 65016 Tarbes Cedex



(+33) 5 62 44 29 33



(+33) 5 62 44 27 08



Olivier.Pantale@enit.fr

### Grades et titres

1996	Doctorat de l'Université de Bordeaux I <i>Modélisation et simulation tridimensionnelles de la coupe des métaux</i>
1992	D.E.A de Mécanique (Université de Bordeaux I)
1992	Diplôme d'Ingénieur de Production (E.N.I de Tarbes)

### Fonctions et expérience professionnelle

depuis 1998	Maître de Conférences - 60 <sup>ème</sup> section - E.N.I de Tarbes
1997-98	Maître de Conférences stagiaire à l'E.N.I de Tarbes
1996-97	Enseignant contractuel sur poste vacant à l'E.N.I de Tarbes
1995-96	A.T.E.R à l'E.N.I de Tarbes
1992-95	Vacataire pour l'enseignement à l'E.N.I de Tarbes

### Liste des travaux de recherche

Travaux scientifiques			Encadrement Doctoral	
Articles	Communications avec actes	Communications sans actes	DEA	Thèses de Doctorat
9	18	5	10	4 <sup>(1)</sup>

(1) dont une en cours

## Travaux de recherche

### Encadrement de thèses de Doctorat

- **Jean Louis Bacaria** : “Un modèle comportemental et transitoire pour la coupe des métaux” *INP de Toulouse* (le 13 novembre 2001).  
*Situation actuelle du diplômé* : Ingénieur chez Latécoère  
taux d’encadrement : 30%
- **Chakib Sattouf** : “Caractérisation en dynamique rapide du comportement de matériaux utilisés en aéronautique” *INP de Toulouse* (le 30 juin 2003).  
*Situation actuelle du diplômé* : Enseignant à l’Université d’Aleppo (Syrie)  
taux d’encadrement : 40%
- **Laurent Menanteau** : “Développement d’un module de prototypage virtuel multi-physique, multi-domaine et multitemps. Application aux convertisseurs de puissance.” *INP de Toulouse* (le 25 octobre 2004).  
*Situation actuelle du diplômé* : ATER à l’ENI de Tarbes  
taux d’encadrement : 50%
- **Ionel Nistor** : “Identification expérimentale et simulation numérique de l’endommagement en dynamique rapide : Application aux structures aéronautiques ” (commencée en octobre 2002).  
taux d’encadrement : 30%

### Encadrement de DEA

- B. Peressini. “Description Euléro-Lagrangienne-Arbitraire : développement d’un code de calcul élastique” (DEA GM, Bordeaux, 1997).
- B. Bonnac. “Modélisation et implémentation d’une loi de comportement du béton” (DEA GM, Bordeaux, 1998).
- C. Dalos. “Implémentation numérique d’un modèle d’endommagement dans un code de dynamique rapide” (DEA GM, Bordeaux, 1999).
- P. Laffont. “Implémentation de lois de comportement élastoplastiques en dynamique rapide dans un code de calcul explicite” (DEA GM, Bordeaux, 1999).
- C. Depierre. “Comportement d’une tige bi-articulée soumise à un chargement conservatif” (DEA GM, Bordeaux, 2000).
- P. Louvet. “Développement et mise au point d’un essai d’impact pour la caractérisation du comportement des matériaux soumis à des sollicitations à grande vitesse” (DEA GM, Toulouse, 2001).
- I. Nistor. “Identification du comportement dynamique à partir des résultats d’essais d’impact de Taylor direct et symétrique” (DEA GM, Toulouse, 2002).
- R. Maran. “Développement d’un essai de traction dynamique pour la caractérisation du comportement des matériaux lors d’impacts” (DEA GM, Toulouse, 2003).
- E. Gorce. “Développement d’une méthodologie d’identification des paramètres constitutifs d’un matériau soumis à de grandes sollicitations thermomécaniques” (DEA GM, Toulouse, 2004).
- B. Aubert. “Modélisation de la rupture dynamique des matériaux fragiles par la méthode des éléments finis étendue (XFEM)” (M2R GM, Toulouse, 2005).

## Activités collectives et administratives liées à la recherche

- Membre de l'association DYMAT (Association Européenne pour la promotion des études du comportement dynamique des matériaux et applications).
- Membre détaché au sein du laboratoire P.E.A.R.L. (Power Electronics Associated Research Laboratory) de la société ALSTOM à Séméac (65).
- Responsable du laboratoire d'identification du comportement dynamique à grande vitesse des matériaux du LGP.
- Responsable de la gestion des moyens de calcul numériques de l'équipe CMAO du LGP.

## Rayonnement scientifique

- Participation au comité de lecture de la revue Européenne des Eléments Finis.
- Correspondant de l'ENIT pour le réseau OPTIMUS (OPTimisation et Modélisation de l'Usinage à Grande Vitesse et/ou à Sec).
- Membre du comité d'organisation de la journée scientifique ENIT 1992 "Création d'outils d'aide à la définition du produit" du 21 mai 1992.
- Membre du comité d'organisation de la journée scientifique ENIT 1997

## Production scientifique

### Publications dans des revues à comité de lecture

- O. Pantalé. Parallelization of an Object-Oriented FEM Dynamics Code : Influence of the strategies on the Speedup. *Advances in Engineering Software*. 36 (6) :361-373, 2005.
- O. Pantalé, S. Caperaa and R. Rakotomalala. Development of an object oriented finite element program : Application to metal forming and impact simulations. *Journal of Computational and Applied Mathematics*. 168 (1-2) :341-351, 2004.
- O. Pantalé, J. L. Bacaria and O. Dalverny, R. Rakotomalala and S. Caperaa. 2D and 3D numerical models of metal cutting with damage effects. *Computers Methods in Applied Mechanics and Engineering* 193(39-41) :4383-4399, 2004.
- I. Nistor, O. Pantalé, S. Caperaa and C. Sattouf. A new dynamic test for the identification of high speed friction law using a gaz-gun device. *Journal de Physique IV*, 110 :519-524, 2003.
- O. Pantalé. An Object-Oriented Programming of an Explicit Dynamics Code : Application to Impact Simulation. *Advances in Engineering Software*. 33 (5) :275-284,8 2002.
- J. L. Bacaria and O. Dalverny and O. Pantalé and R. Rakotomalala. Transient Numerical models of metal cutting using the Johnson-Cook's Rupture Criterion. *International Journal of Forming Processes*, 5(1) :53-70, 2002.
- O. Dalverny, S. Caperaa, O. Pantalé and C. Sattouf. Identification de lois constitutives et de lois de frottement adaptées aux grandes vitesses de sollicitation. *Journal de Physique IV*, 12 :275-281, 2002.
- O. Pantalé, R. Rakotomalala, and M. Touratier. An ALE three-dimensional model of orthogonal and oblique metal cutting processes. *International Journal of Forming Processes*, 1(3) :371-388, 09 1998.
- P. Joyot, R. Rakotomalala, O. Pantalé, M. Touratier, and N. Hakem. A numerical simulation of steady state metal cutting. *Proceedings in Inst. Journal for Mechanical Engineers*, 212 (part C) :331-341, 1998.

**Congrès avec actes édités**

- O. Pantalé and S. Caperaa. Strategies for a parallel 3D FEM code : Application to impact and crash problems in structural mechanics. In *Coupled Problems 2005*, Santorini Island, may 25-28, 2005.
- L. Menanteau, O. Pantalé and S. Caperaa. A coupled electro-thermo-mechanical FEM code for large scale problems including multi-domain and multiple time-step aspects. In *Coupled Problems 2005*, Santorini Island, may 25-28, 2005.
- H. Abichou, O. Pantalé, I. Nistor, O. Dalverny and S. Caperaa. Identification of metallic material behaviors under high-velocity impact : A new tensile test. In *15th Technical Meeting DYMAT*, Metz, june 1-2, 2004.
- O. Pantalé, I. Nistor and S. Caperaa. Identification et modélisation du comportement des matériaux métalliques sous sollicitations dynamiques. In Military Technical Academy, editor, *30th Internationally attended scientific conference of the military technical academy*, ISBN 973-640-012-3, Bucharest, 2003.
- I. Nistor, O. Pantalé, S. Caperaa and C. Sattouf. Identification of a dynamic visco-plastic flow law using a combined Levenberg-Marquardt and Monte-Carlo algorithm. *VII International Conference of Computational Plasticity*, Barcelona, 2003.
- L. Menanteau, S. Caperaa and O. Pantalé. A multigrid method for the thermomechanical behaviour simulation applicated to power electronics converter. *VII International Conference of Computational Plasticity*, Barcelona, 2003.
- O. Pantalé and S. Caperaa and R. Rakotomalala. Development of an object-oriented finite element program : application to metal-forming and impact simulations. *ACOMEN 2002 Second international conference on advanced computational methods in engineering*. ISBN 2-930322-39-X, Liège University (Belgium), may 28-31, 2002.
- J.L. Bacaria, O. Dalverny, O. Pantalé, R. Rakotomalala and S. Caperaa. Transient numerical model of discontinuous chip formation based on a damage effect. *Mini-symposium Machining in the Esaform 2001 congress, Liège - Belgium*. april 23-25, 2001.
- C. Sattouf, O. Pantalé and S. Caperaa. A methodology for the identification of constitutive and contact laws of metallic materials under High Strain Rates. In Euromat, *Advances in Mechanical Behaviour, Plasticity and Damage (Tours)*, november, 2000.
- J. L. Bacaria, O. Dalverny, O. Pantalé, R. Rakotomalala, and S. Caperaa. 2D and 3D numerical models of metal cutting with damage effects. In ECCOMAS, editor, *European Congress on Computational Methods in Applied and Engineering (Barcelone)*, september, 2000.
- O. Pantalé, R. Rakotomalala, and S. Caperaa. Développement d'un code de calcul explicite ALE : Application à la coupe des métaux. In *14ème congrès français de mécanique - Toulouse*, pages 951.1-6, septembre, 1999.
- O. Pantalé, R. Rakotomalala, and M. Touratier. Simulation of chip formation in machining. In Neat Press, editor, *Proceedings of plasticity'99 (Cancun)*, pages 397-400, january, 1999.
- O. Pantalé, R. Rakotomalala, M. Touratier and S. Caperaa. A Numerical Model of Metal Cutting : Effect of the Friction and the Tool Wear. *First French and German Conference on HIGH SPEED MACHINING - (Metz)*, pages 420-423, june, 1997.
- O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. A three-dimensional numerical model of orthogonal and oblique metal cutting processes. In ASME, editor, *PD-Vol. 75; Engineering Systems Design and Analysis*, volume 3, pages 199-206, july 1-4, 1996.
- O. Pantalé, R. Rakotomalala, M. Touratier, and S. Caperaa. Un modèle numérique d'usinage. Analyse de l'effet du frottement et de l'usure. In *Journée PRIMECA*, 1996.
- O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. Modélisation tridimensionnelle



de la coupe orthogonale. In Hermès, editor, *2ème Colloque en Calcul de Structures - Giens - ISBN 2-86601-471-5*, volume 2, pages 597-602, 16-19 mai, 1995.

- O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. Modélisation tridimensionnelle de la coupe orthogonale. In *Journée PRIMECA Procédés d'Usinage - CER ENSAM - 71250 Centre de Cluny*, 3-5 avril, 1995.
- P. Joyot, R. Rakotomalala, O. Pantalé, M. Touratier, and N. Hakem. Un modèle prédictif de coupe orthogonale basé sur une formulation Arbitraire Lagrangienne Eulérienne. In *Journée Procédés d'usinage - PRIMECA*, 19 mai, 1994.

### Congrès sans actes édités

- O. Pantalé and S. Caperaa. Développement d'un code de calcul explicite en grandes transformations : Application à la coupe des métaux. *Séminaire Optimus - ENSAM de Cluny*, 6-7 octobre, 2004.
- O. Pantalé, I. Nistor, O. Dalverny, E. Gorce and S. Caperaa. Caractérisation du comportement dynamique des matériaux à partir d'essais d'impact. *Séminaire Optimus - ENSAM de Bordeaux*, 13-14, mai 2004.
- O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. An ale tridimensionnal numerical simulation of orthogonal cutting process. *First International Radioss Users Conference - SOPHIA ANTIPOLIS*, june 19, 1995.
- O. Pantalé, P. Joyot, R. Rakotomalala, M. Touratier, and N. Hakem. Présentation d'un modèle tridimensionnel A.L.E de coupe orthogonale. *Usinage des Matériaux à Caractéristiques Mécaniques élevées E.T.C.A*, 21 juin, 1994.

### Conférences devant un public non spécialisé

- O. Pantalé and P. Charbonnaud. Avancées en modélisation, optimisation, commande et supervision des processus de coupe. *Conférence scientifique à PYRENEXPO 97*, 8 octobre, 1997.

### Mémoire et thèse

- O. Pantalé. Modélisation et simulation tridimensionnelles de la coupe de métaux. *Thèse de doctorat - Université de Bordeaux I*, juillet, 1996.
- O. Pantalé. Modélisation et Implantation Numérique de l'Endommagement de Fatigue dans le Code de Calcul de Structures ABAQUS. *Mémoire de D.E.A - Université de Bordeaux I*, juin, 1992.

## Activités d'enseignement

### Service d'enseignement

N'ayant pas eu de financement pour la réalisation de mes travaux de thèse, j'ai commencé par assurer une charge d'enseignement en deuxième cycle à l'Ecole Nationale d'Ingénieurs de Tarbes dès le début de ma thèse en 1992. Durant cette période, j'ai pu saisir les opportunités d'enseigner des disciplines variées, aussi bien dans des enseignements de mécanique que de physique ou d'informatique.

Dans le cadre de ces enseignements, une grande partie de mes activités ont concerné des mises à niveau destinées à des étudiants recrutés au niveau de la 3<sup>ème</sup> et 4<sup>ème</sup> année à

l'ENIT (élèves issus de formations BTS, IUT, Maîtrise, Math Sup...). Un des intérêts de ce type d'enseignement réside dans le fait que la diversité de recrutement de ces étudiants oblige à une adaptation du niveau d'enseignement à la demande. Depuis ma nomination sur un poste de maître de conférences, mes enseignements se sont recentrés sur la filière classique de formation des ingénieurs en deuxième cycle et la formation en M2R de mécanique, tout en prenant part à des enseignements dans des filières particulières de formation (Formation continue CMAO, Ecole de la Performance, etc...).

Pour l'année universitaire 2004/2005, on pourra citer les enseignements suivants pour lesquels tous les volumes horaires sont convertis en heures équivalent TD/groupe :

### Cours

M2R	Grandes Transformations Thermomécaniques Rapides	12
5 <sup>eme</sup> Année	Eléments finis de structures (poutres et plaques)	21
5 <sup>eme</sup> Année	Module optionnel CMAO	9
5 <sup>eme</sup> Année	Système d'exploitation UNIX	6
4 <sup>eme</sup> Année	Machines mécaniques et turbomachines	24
4 <sup>eme</sup> Année	Thermodynamique	24
4 <sup>eme</sup> Année	Mécanique des milieux continus	48
F. continue	Résistance des Matériaux	12

### Travaux Dirigés

4 <sup>eme</sup> Année	Machines mécaniques et turbomachines	8
------------------------	--------------------------------------	---

### Travaux Pratiques

M2R	Grandes Transformations Thermomécaniques Rapides	4
5 <sup>eme</sup> Année	Eléments finis de structures (poutres et plaques)	4
4 <sup>eme</sup> Année	Elements finis linéaires	6

### Activités collectives et administratives liées à l'enseignement

- Membre titulaire du Conseil d'Administration de l'ENIT (2005-2009).
- Membre titulaire du Conseil Pédagogique de l'ENIT (2001-2005).
- Responsable de la coordination des enseignements de mécanique sur les 5 années pour l'ENIT.
- Responsable de la région Benelux pour les relations internationales.
- Membre de la commission informatique de l'ENIT.
- Membre de la commission Bibliothèque de l'ENIT.

### Documents pédagogiques

- Cours de Grandes Transformations Thermomécaniques Rapides (M2R) - 107 pages
- Cours de mécanique des structures par éléments finis (5<sup>eme</sup> année ENIT) - 70 pages
- Cours de thermodynamique (4<sup>eme</sup> année ENIT) - 30 pages
- Cours de mécanique des milieux continus et élasticité (4<sup>eme</sup> année ENIT) - 88 pages
- Cours de machines mécaniques et turbomachines (4<sup>eme</sup> année ENIT) - 106 pages
- Programmation et développement dans un environnement orienté-objets (Formation Laboratoire) - 100 pages

---

# **Parcours de recherche**

---



# 1

# Modélisation numérique de la coupe des métaux

---

*The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.*

John von Neumann (1903-1957)

Ce premier chapitre est destiné à mettre en place une problématique liée à la modélisation numérique en Grandes Transformations. Sur l'ensemble des problèmes pour lesquels on est confronté à ce type d'approche, j'ai choisi le thème de la modélisation numérique de la coupe des métaux car il se situe dans le prolongement de mes travaux de thèse effectués au Laboratoire Génie de Production (LGP) entre 1992 et 1996. Ce chapitre présente donc une synthèse des problèmes liés à la modélisation de la coupe des métaux tels qu'ils ont été identifiés au cours des études réalisées sur ce thème.

## 1.1 Introduction et positionnement

A l'heure actuelle, l'obtention de pièces industrielles passe généralement par des procédés de mise à forme et d'usinage par enlèvement de matière. L'activité concernant l'usinage est en pleine évolution actuellement et les besoins exprimés en termes de progrès nécessitent une approche scientifique. En dépit de quelques avancées récentes, l'évolution des procédés d'usinage dans la dernière décennie est principalement le résultat de méthodes heuristiques plutôt qu'une conséquence d'un développement scientifique réel. En Europe, divers programmes ont permis le développement de la technologie d'Usinage à Grande Vitesse (UGV) et la réduction du recours à l'utilisation de lubrifiants de coupe motivée par les contraintes liées à l'environnement. Ces divers programmes de recherche ont également permis au cours des dernières années de développer de nouvelles compétences dans les domaines de :

- la compréhension des phénomènes d'usinage,
- le développement de nouveaux revêtements pour les outils de coupe,
- la compréhension des phénomènes physico-chimiques conduisant à la dégradation des outils,
- la modélisation numérique du procédé d'usinage,
- la caractérisation des matériaux,
- la métrologie de la zone de coupe (efforts, température...).

Les difficultés principales dans l'analyse des procédés d'usinage sont dues aux conditions extrêmes de sollicitation de la matière, en particulier l'écoulement du copeau sous déformation et vitesse de déformation très élevées, au frottement important entre l'outil et la matière, aux variations brusques et importantes de température, ou aux vibrations du système usinant... En plus de ces conditions extrêmes, le procédé est multi-échelle et très difficile à caractériser en temps réel.

A l'échelle matériau/outil (échelle mésoscopique), domaine où est formé le copeau et où est générée la surface fonctionnelle, il est primordial de pouvoir comprendre et maîtriser les phénomènes physico-chimiques, thermiques et mécaniques qui sont à l'origine du mécanisme de coupe. L'une des voies d'investigation consiste alors à construire des modèles numériques ou analytiques de simulation de la coupe. Lors de la construction de ces modèles, la méconnaissance des caractéristiques des matériaux usinés (lois de comportement déterminées dans des conditions de sollicitation différentes de celles de la coupe), et les difficultés de caractérisation expérimentale de la zone de coupe (températures, déformations, vitesses de déformation) font que l'approche numérique est loin d'être optimale, et que la vérification expérimentale est complexe à mettre en oeuvre. Ceci constitue un verrou technologique et scientifique qu'il est indispensable de lever dans l'avenir.

Dans le cadre des travaux de recherche effectués au sein du LGP, seule la voie de la modélisation numérique de la coupe des métaux est exploitée. Celle-ci correspond à une approche mésoscopique des phénomènes liés à la coupe. Dans cette même période, des approches analytiques ont été développées, mais, j'ai délibérément choisi de recentrer mes travaux de recherche sur la seule modélisation numérique.

## 1.2 Mise en oeuvre de modèles numériques de simulation de la coupe

### 1.2.1 Approche et problèmes mis en jeu lors de la coupe

Afin de limiter les problèmes géométriques, la modélisation numérique du processus de coupe est généralement envisagée dans le cas particulier de la coupe orthogonale pour laquelle l'arête de coupe est simultanément perpendiculaire à la vitesse de coupe  $V_c$  et à la vitesse d'avance  $V_a$  comme indiqué sur la figure 1.1(a). C'est typiquement le cas du chariotage d'un tube creux ou de l'usinage d'une ailette en tournage. Ce cas correspond donc à une valeur de l'angle de direction d'arête  $K_r = 90^\circ$  comme indiqué sur la figure 1.1(b). Dans cette configuration  $R_b$  est la valeur du rayon de bec reliant la face de coupe à la face de dépouille,  $w$  est la largeur de coupe,  $a$  est la profondeur d'avance,  $y$  est la longueur de contact outil-copeau et  $e$  est l'épaisseur du copeau formé.

Le principal avantage de la coupe orthogonale est de simplifier la cinématique de la coupe afin de se ramener à une modélisation bidimensionnelle. D'un point de vue phénoménologique, et indépendamment du cas particulier de la coupe orthogonale, l'analyse de la formation du copeau fait apparaître principalement 2 zones principales de sollicitation du matériau (voir figure 1.2) :

- la zone de cisaillement primaire dans laquelle se situe le changement de la direction d'écoulement du copeau,
- la zone de cisaillement secondaire induite par le frottement du copeau sur la face de coupe de l'outil, qui est le siège de contraintes de cisaillement intenses entraînant une forte élévation de température.

Toutes ces zones sont le siège de déformations et de taux de déformation élevées ( $1 < \epsilon < 6$  et  $10^3 s^{-1} < \dot{\epsilon} < 10^5 s^{-1}$ ). D'un point de vue thermique les analyses [1] montrent que les vitesses

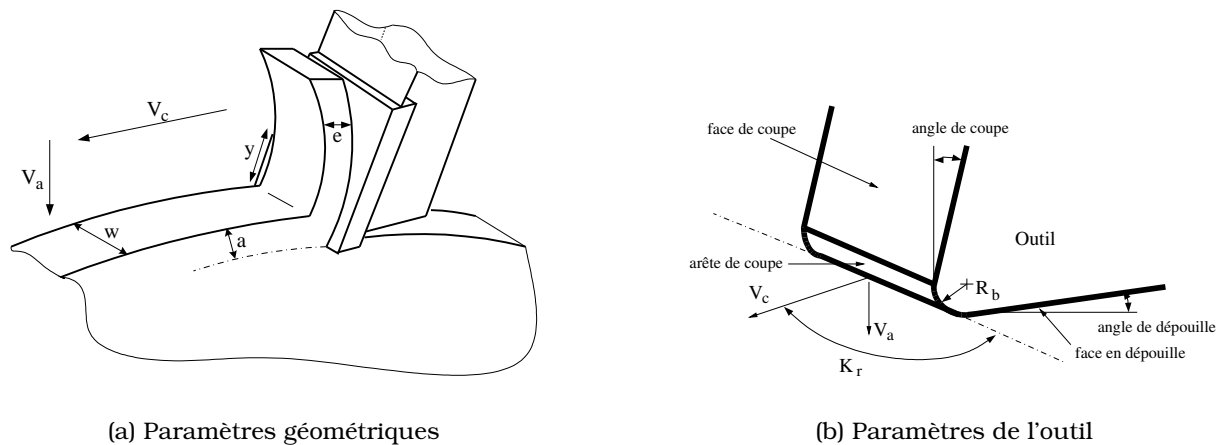


FIG. 1.1 – Description des paramètres de coupe orthogonale

de variation de température sont de l'ordre de  $10^6 \text{ }^\circ\text{K/s}$  lors de la traversée de la bande de cisaillement primaire.

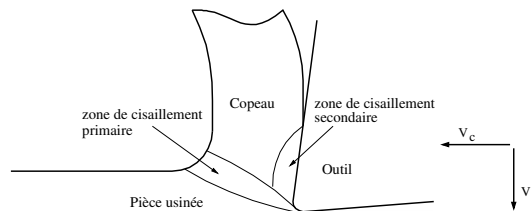


FIG. 1.2 – Localisation des bandes de cisaillement dans un processus de coupe

### 1.2.2 Modèles numériques de coupe

Les premières simulations numériques de la coupe des métaux ont commencées à être mises au point dans les années soixante-dix pour le cas de la coupe continue orthogonale en tournage. En 1974, Shirakashi et al. [2] présentent un modèle numérique de coupe orthogonale élastoplastique dans lequel la géométrie et les vitesses sont supposées connues. Dans les années 80, les modèles de coupe commencent à se développer et deux voies distinctes de modélisation prennent place (les modèles Eulériens et les modèles Lagrangiens) :

- les modèles numériques Eulériens se basent sur une géométrie de copeau initiale et traitent le cas de la coupe continue en considérant le régime stationnaire établi lorsque les principales grandeurs physiques sont stabilisées. Cette approche nécessite la connaissance de la géométrie finale du copeau en tant que donnée du problème modélisé. Ce type d'approche a été utilisé par Iwata et al. [3] sur un acier C45 afin de prédire les valeurs de déformation et de vitesses de déformation dans les bandes de cisaillement primaire et secondaire. Suivant la même approche, Carroll et al. [4] proposent un modèle thermomécanique de coupe pour un aluminium 2024 et montrent que la diminution de l'angle de coupe augmente la valeur de l'effort de coupe.
- les modèles Lagrangiens pour leur part simulent la formation du copeau en régime transitoire jusqu'à obtention du régime stationnaire. La principale difficulté rencontrée dans les modèles Lagrangiens est liée à la nécessité de séparer physiquement le matériau usiné au niveau de la pointe de l'outil. La voie généralement adoptée consiste alors à définir un

ensemble de noeuds couplés que l'on sépare l'un de l'autre au moyen d'un critère géométrique [5] basé sur une notion de distance, un critère mécanique [6] basé sur une déformation plastique équivalente ou une combinaison de ces deux critères [7]. Cette approche restreint les modèles numériques à des situations pour lesquelles la trajectoire de la pointe de l'outil est simple, ce qui exclue les modèles numériques de fraisage réalistes.

A partir des années 1990, les modèles numériques de coupe ont notablement évolué et ont commencé à devenir réalistes. Ces modèles numériques donnent des informations concernant les déformations et les contraintes, les zones de cisaillement et le champ de température dans le cas de comportements thermomécaniques couplés. Lin et al. [8] en 1993 présentent une comparaison entre résultats numériques et expérimentaux concernant les efforts de coupe. Marusich et al. [9] en 1995 et Obikawa et al. [7] en 1997 présentent les premières applications de coupe orthogonale discontinue transitoire. D'un point de vue commercial et industriel, sous la demande principale de grandes compagnies américaines, des outils spécifiques de simulation numérique de la coupe des métaux (Third Wave Systems [9]), ou du comportement dynamique de l'usinage (Cutpro [10]) ont été développés. Les évaluations effectuées sur ces logiciels montrent que si pour une approche industrielle, ils semblent attractifs, d'un point de vue scientifique, ils souffrent de lacunes au niveau de la physique du processus de coupe et de la possibilité de prendre en compte des lois de comportement complexes.

### 1.2.3 La modélisation numérique de la coupe au sein du LGP

Au début des années 1990, le laboratoire Génie de Production de l'ENIT a initié un programme de recherche dans le domaine de la modélisation numérique de la coupe des métaux. Ces travaux ont été développés aux cours de trois thèses successives portant sur la modélisation de la coupe des métaux entre 1991 et 2001, ainsi :

- P. Joyot [11] en 1994 présente une première approche de la modélisation numérique de coupe orthogonale continue,
- O. Pantalé [1] en 1996 présente un modèle numérique tridimensionnel de coupe orthogonale et de coupe oblique continue,
- J. L. Bacaria [II] en 2001 présente un modèle numérique de simulation du tournage en coupe transitoire discontinue 2D et un modèle numérique de simulation du fraisage 3D en contournage.

Le point commun de l'ensemble des travaux réalisés au laboratoire est liée à l'utilisation d'un formalisme Euléro-Lagrangien Arbitraire (ALE) pour la description des mouvements [12]. Cette approche qui permet de fédérer les avantages des deux formulations classiques de manière unifiée a depuis été reprise par d'autres auteurs parmi lesquels : Olovsson et al. [13] en 1999 et Movahhedy et al. [14] en 2000.

Dans l'ensemble des travaux réalisés, on peut globalement considérer deux types d'approches distinctes bâties autour de la formulation ALE conduisant à deux familles de modélisations de la coupe des métaux. La première est une approche Eulérienne permettant la mise en oeuvre de modèles de coupe continus stationnaires alors que la seconde est une approche Lagrangienne dédiée à la mise en oeuvre de modèles de coupe transitoires discontinus.

#### 1.2.3.1 Modèles de coupe stationnaires ALE

Les modèles présentés dans cette section sont relatifs à la simulation numérique de la coupe stationnaire des métaux homogènes isotropes à copeau continu. Le formalisme adopté est ALE à tendance Eulérienne. Le calcul est effectué à partir d'une géométrie initiale arbitraire qui évolue de façon continue pour atteindre la géométrie stationnaire finale. Les simulations numériques pour le cas de ces modèles stationnaires ont été réalisées à l'aide d'une version expérimentale ALE du code de calcul Radioss développée durant la période 1991-95 dans



le cadre d'un partenariat entre la société Renault et le LGP. Historiquement, cette période d'étude est couverte par les travaux de P. Joyot [11] et O. Pantalé [1].

Dans tous les modèles développés au laboratoire, la loi de comportement adoptée est une loi élastoplastique de type Johnson-Cook [15] dont la formulation détaillée est donnée à la page 33 :

$$\sigma = (A + B\bar{\epsilon}^n) \left( 1 + C \ln \frac{\dot{\bar{\epsilon}}}{\dot{\bar{\epsilon}}_0} \right) \left[ 1 - \left( \frac{T - T_0}{T_{melt} - T_0} \right)^m \right] \quad (1.1)$$

Le matériau usiné est un acier 42CrMo4 dont les paramètres sont reportés dans le tableau 1.1, l'outil de coupe est un carbure de tungstène non revêtu SECO TPGN-160302 P10. Le contact entre la pièce et l'outil est modélisé par une loi de frottement de type Coulomb pour lequel la valeur du coefficient de frottement  $\mu = 0,32$  a été identifiée expérimentalement au sein du laboratoire [16].

TAB. 1.1 – Paramètres matériau pour la pièce usinée et l'outil

$E$	206 GPa	$A$	595 MPa	$E$	470 GPa
$\nu$	0.3	$B$	580 MPa	$\nu$	0.224
$\rho$	7850 kg/m <sup>3</sup>	$C$	0.023	$T_0$	300°K
$C_p$	358 J/kg°K	$m$	13	$T_{melt}$	1793°K
$\lambda$	38 W/m°K	$n$	0.133	$\dot{\bar{\epsilon}}_0$	1000
Paramètres de la pièce usinée				Paramètres de l'outil	

Les conditions aux limites générales des modèles stationnaires sont données sur la figure 1.3. La pièce est modélisée selon un formalisme ALE alors que l'outil est modélisé avec un formalisme Lagrangien (il est mécaniquement indéformable et seule la partie thermique est considérée). La notion de surface libre fait appel à des noeuds qui sont simultanément Lagrangiens dans la direction perpendiculaire à la surface et Eulériens dans le plan tangent à la surface [1]. Cette approche permet la ré-actualisation des surfaces libres du modèle. Toutes les surfaces du modèle sont adiabatiques à l'exception de la surface de contact entre la pièce usinée et l'outil pour laquelle le flux de chaleur généré par frottement est réparti pour moitié dans les deux parties en contact dans notre modèle.

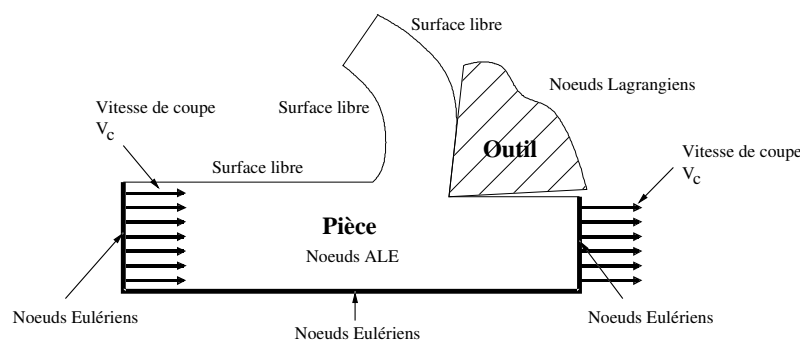


FIG. 1.3 – Conditions aux limites du modèle éléments finis stationnaire

Les premiers modèles ALE développés au cours de cette étude sont des modèles de coupe 2D orthogonale. Ils couvrent la période de 1991 à 1994. Les principaux résultats issus de ces modèles concernent l'évaluation des efforts de coupe mis en jeu, la répartition des champs de températures. Des résultats originaux ont été proposés [17] sur la base d'une étude de la sensibilité de la solution obtenue en fonction de la valeur du coefficient de frottement. On a ainsi montré que le coefficient de frottement a une influence directe sur la longueur de

contact entre l'outil et la pièce, la géométrie du copeau et la répartition des températures comme illustré sur la figure 1.4 pour laquelle seule la valeur du coefficient de frottement varie entre  $\mu = 0.1$  et  $\mu = 0.45$ .

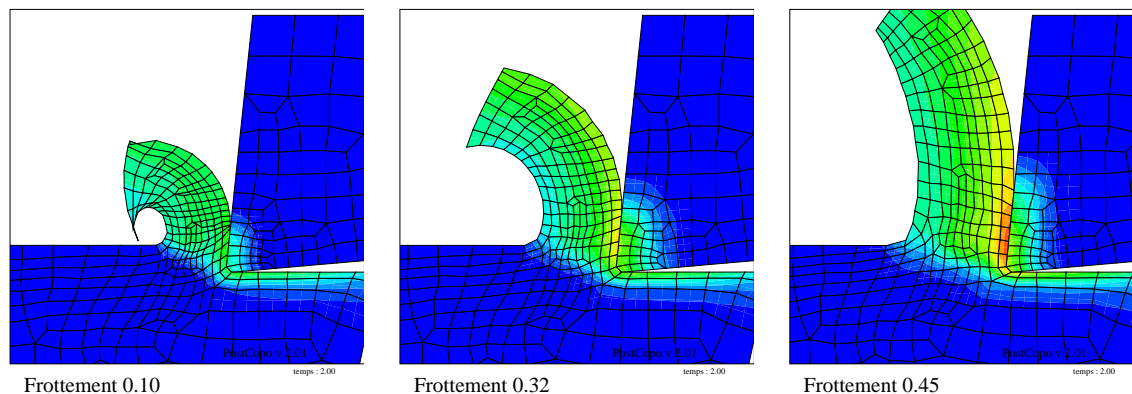


FIG. 1.4 – Sensibilité de la solution obtenue en fonction du coefficient de frottement

L'usure de la plaquette de coupe, et principalement l'usure en cratère de la face de coupe a également une influence sur la géométrie du copeau et la répartition des contraintes équivalentes de von Mises comme illustré sur la figure 1.5. Cette influence se retrouve également sur la valeur des efforts de coupe, on peut ainsi montrer [17, 18] que, une usure en cratère de la face de coupe de  $0.35\mu\text{m}$  de profondeur, accroît l'effort de coupe de 15%.

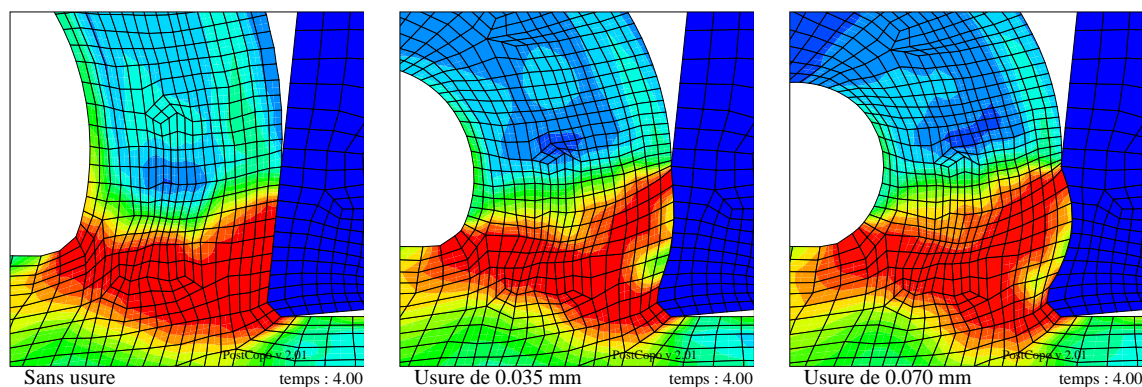


FIG. 1.5 – Influence de l'usure en cratère de la face de coupe

A la suite des développements de modèles de coupe orthogonale 2D plans, une approche tridimensionnelle de la coupe orthogonale a été entreprise [19, 20, 21, 22]. Celle-ci permet dans un premier temps de faire abstraction des limitations liées à l'utilisation d'un formalisme 2D en contraintes planes ou en déformations planes. Pour des raisons de symétrie du problème, uniquement la moitié de la largeur de la pièce et de l'outil sont modélisés. Dans le modèle proposé, la largeur de l'outil est supérieure à la largeur de la pièce pour respecter les conditions de coupe orthogonale. La figure 1.6 présente le champ de températures et le champ de contraintes après stabilisation des valeurs. Les valeurs numériques obtenues sont sensiblement les mêmes que pour la modélisation 2D, mais l'apport de la troisième dimension permet de compléter les résultats numériques disponibles. Les résultats supplémentaires macroscopiques concernent par exemple l'évaluation d'un gonflement latéral de 14% de la largeur du copeau par rapport à la largeur de la pièce usinée. Un autre résultat intéressant concerne la corrélation entre la cartographie de la température sur la face de coupe et l'usure en cratère mesurée expérimentalement [1] de manière similaire aux travaux de Tay [23]. On est ainsi en mesure de déterminer que dans les conditions de coupe utilisées pour cette modélisation,

l'usure de la face de coupe est liée à la distribution des températures sur la face de coupe.

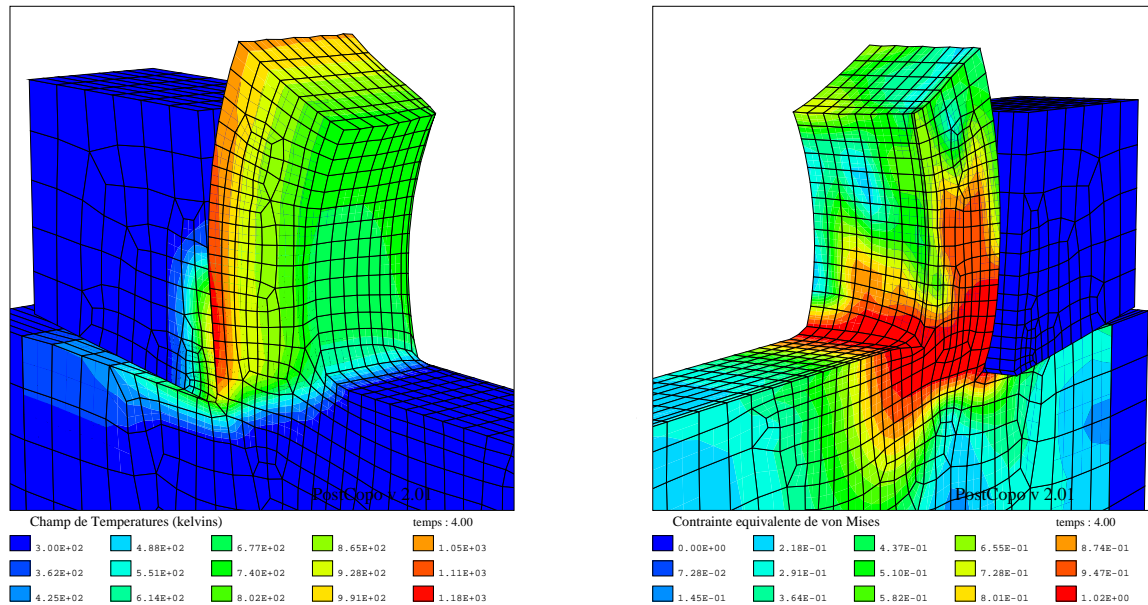


FIG. 1.6 – Champs de températures et de contraintes en coupe 3D orthogonale

Le dernier apport des modèles développés au cours de cette période concerne la modélisation tridimensionnelle de la coupe oblique. La coupe oblique correspond à une variation de l'angle de direction d'arête  $\kappa_r \neq 90^\circ$ . La figure 1.7 présente la cartographie des contraintes de von Mises pour une valeur de direction d'arête  $\kappa_r = 75^\circ$ . L'introduction d'un angle de direction d'arête  $\kappa_r \neq 90^\circ$  a une incidence directe sur les résultats numériques traduite par une composante  $F_z$  dans les efforts de coupe [24, 25, 26]. Cette composante augmente rapidement avec la diminution de  $\kappa_r$  pour atteindre 28% de l'effort total pour un angle  $\kappa_r = 75^\circ$ . Dans le même temps, la direction d'écoulement du copeau varie d'un angle  $\delta = 48^\circ$  par rapport au plan formé par les 2 vitesses  $\vec{V}_a$  et  $\vec{V}_c$ . Le facteur de proportionnalité identifié à partir des modélisations numériques permet de définir une loi de variation linéaire de la direction d'écoulement pour une valeur de  $\kappa_r$  comprise entre  $75^\circ$  et  $90^\circ$  sous la forme :

$$\delta = 3(90 - \kappa_r) \quad (1.2)$$

### 1.2.3.2 Modèles de coupe transitoires ALE

Dans une deuxième période, à la suite des développements de modèles numériques stationnaires, les nouvelles orientations ont porté sur le développement de modèles numériques transitoires au cours de la thèse de J. L. Bacaria [1]. Pour cette approche, le choix du logiciel s'est porté sur le code de calcul Abaqus Explicit v. 5.8, et le formalisme adopté est cette fois ci ALE à tendance Lagrangienne. Comme présenté précédemment, divers modèles numériques transitoires Lagrangiens ont été développés au cours des dernières années. Une avancée intéressante concerne dans ce cadre d'étude le développement de modèles numériques permettant de simuler la formation de copeaux fragmentés comme proposé par Obikawa et al. [7] dans lequel les auteurs ont recours à un double critère de séparation du copeau : un premier purement géométrique destiné à séparer le copeau en pointe d'outil, et un second basé sur une valeur de déformation plastique équivalente maximale destiné à permettre la formation d'un copeau fragmenté. Les résultats présentés montrent l'influence de la fragmentation du copeau sur les efforts de coupe.

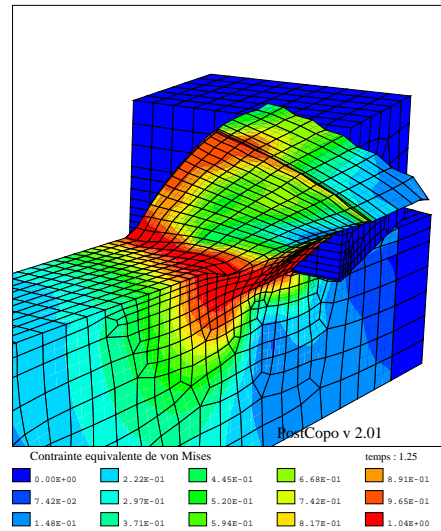


FIG. 1.7 – Champ de contraintes en coupe 3D oblique

Une autre approche très intéressante est proposée par Marusich et al. [9]. C'est une approche comportementale de la coupe des métaux. Les auteurs utilisent un critère basé sur la propagation de fissures dépendant de l'état de contraintes et de déformations au sein de la pièce usinée. Cette nouvelle approche permet de développer des modèles Lagrangiens transitoires sans nécessiter le recours à des "pré-découpages" constitués de noeuds doubles que l'on sépare au cours de l'avancement de l'outil.

L'originalité de ce travail par rapport aux études réalisées par d'autres auteurs réside dans l'utilisation d'un critère d'endommagement associé à la loi d'écoulement de Johnson-Cook [27] prenant en compte la déformation, le taux de déformation et la température. Cette loi est déterminée à partir d'essais de traction et de torsion. L'endommagement est déterminé pour chaque élément à partir de la relation suivante :

$$D = \sum \frac{\Delta \bar{\varepsilon}^p}{\bar{\varepsilon}^{p,f}} \quad (1.3)$$

dans laquelle  $\Delta \bar{\varepsilon}^p$  est l'incrément de déformation plastique équivalente,  $\bar{\varepsilon}^{p,f}$  est la déformation équivalente à rupture du matériau. La rupture du matériau est effective lorsque  $D = 1.0$  et les éléments concernés sont alors supprimés. L'expression générale de la déformation équivalente à rupture associée à la loi d'écoulement de Johnson-Cook est donnée par la relation :

$$\bar{\varepsilon}^{p,f} = (D_1 + D_2 \exp(D_3 \sigma^*)) \left( 1 + D_4 \ln \frac{\dot{\bar{\varepsilon}}^p}{\dot{\bar{\varepsilon}}_0} \right) \left[ 1 - D_5 \left( \frac{T - T_0}{T_{melt} - T_0} \right) \right] \quad (1.4)$$

dans laquelle  $\sigma^* = \frac{\sigma_m}{\bar{\sigma}}$  représente le taux de triaxialité issu du rapport de la contrainte moyenne  $\sigma_m$  sur la contrainte équivalente de von Mises  $\bar{\sigma}$ . Les variables  $D_1, \dots, D_5$  définissent les paramètres matériaux de la loi d'endommagement qu'il convient d'identifier. Les autres paramètres ont la même signification que pour la loi d'écoulement de Johnson-Cook présentée précédemment. L'expression dans le premier jeu de parenthèses de l'équation (1.4) est similaire à la forme présentée par Hancock et al. [28] et traduit essentiellement la décroissance de la déformation équivalente à rupture lorsque la pression hydrostatique  $\sigma_m$  augmente. Il est également important de noter que les simulations numériques effectuées au cours de ce travail ont montré que les paramètres  $D_4$  et  $D_5$  n'ont pas d'influence significative dans le contexte de l'usinage

[29, 30]. Les paramètres matériau  $D_1$ ,  $D_2$  et  $D_3$  ont été identifiés expérimentalement au laboratoire par J.L. Bacaria et O. Dalverny pour le 42CrMo4 en utilisant la procédure proposée par Johnson et al. [27], les paramètres  $D_4$  et  $D_5$  sont ceux de l'acier 4340 proposé par Johnson et al. [27]. Les valeurs correspondantes sont reportées dans la table 1.2.

TAB. 1.2 – Paramètres de la loi d'endommagement du 42CrMo4

$D_1$	1.5	$D_2$	3.44	$D_3$	-2.12
$D_4$	0.002	$D_5$	0.61		
Paramètres pour le 42CrMo4					

Les premiers modèles développés dans cette étude sont des modèles de coupe 2D orthogonale transitoire. Le but de ces modèles est d'établir la faisabilité du procédé et de comparer les résultats issus de cette nouvelle approche avec les résultats obtenus lors des précédentes études. Le premier exemple présenté ici correspond à la simulation numérique d'un procédé de tournage en coupe orthogonale. La figure 1.8 montre la distribution des contraintes de von Mises. Les résultats numériques obtenus sont en accord avec les précédentes simulations et les résultats de la littérature, du point de vue géométrique et mesure des efforts de coupe.

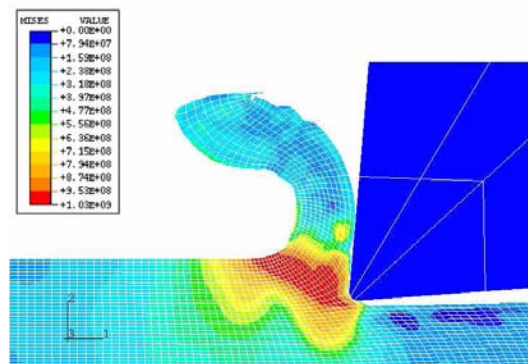


FIG. 1.8 – Champ de contraintes en coupe 2D orthogonale

Suite à l'utilisation d'un critère global d'endommagement il est alors possible de modéliser des trajectoires d'outil relativement complexes et donc d'aborder la simulation numérique du fraisage dans la configuration décrite sur la figure 1.9.

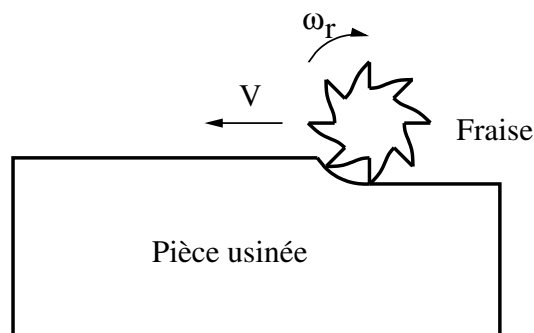


FIG. 1.9 – Description de l'opération de fraisage

Dans cette modélisation, la fraise, considérée comme une surface rigide indéformable, est animée d'une vitesse de rotation  $\omega_r = 120tr/min$  et d'une vitesse de translation  $V = 50m/s$ . Elle a un diamètre  $d = 3mm$ , un angle d'hélice  $\theta = 30^\circ$  et possède 8 dents. Dans cette simulation,

la phase préliminaire transitoire est relative au travail des deux premières dents. A partir de l'engagement de la troisième dent dans la matière, le processus devient cyclique. La figure 1.10 montre la répartition des contraintes équivalentes de von Mises au cours du processus de coupe. Lorsque la dent de la fraise pénètre la pièce, la bande de cisaillement primaire est clairement visible comme indiqué sur la figure 1.10(a). Cette situation correspond à un cas de coupe oblique comme présenté précédemment. La situation à l'instant où la dent sort de la pièce est représentée sur la figure 1.10(b). Pour les détails de ces simulations transitoires, on renvoie le lecteur aux résultats publiés par Bacaria et al. [31] et Pantalé et al. [32].

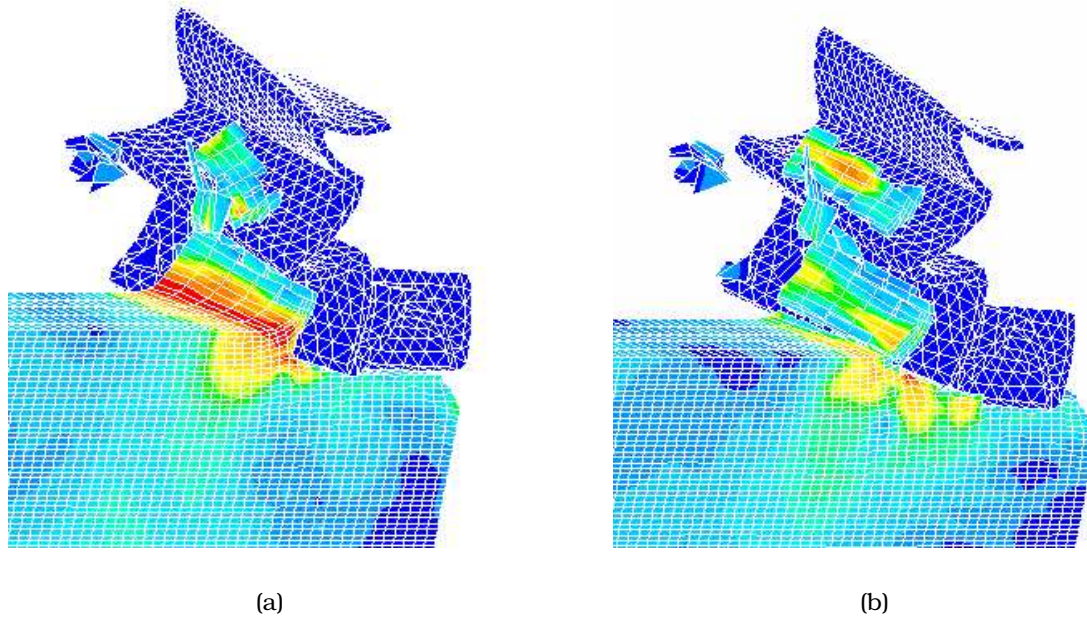


FIG. 1.10 – Champ de contraintes en fraisage

### 1.3 Synthèse des difficultés liées à la modélisation numérique de la coupe

A l'issue de l'ensemble des travaux réalisés au sein du laboratoire sur la simulation numérique de la coupe des métaux, on est en mesure de dégager un certain nombre de points clés qu'il convient de prendre en compte afin de développer des modèles numériques efficaces et fiables. Comme précisé précédemment, les études de modélisation de la coupe ont été menées sur une période de 10 ans environ. Au cours de cette période d'étude, 2 logiciels majeurs ont été utilisés : le code de calcul de crash Radioss et le code commercial non-linéaire Abaqus/Explicit. Force est de constater au terme de ces études qu'aucun des deux codes de calcul n'est réellement adapté pour la modélisation de la coupe des métaux. De plus, les spécificités liées au domaine de l'usinage montrent que des développements particuliers doivent être effectués dans ces codes de calcul et qu'ils se situent bien au delà des possibilités offertes à l'utilisateur en standard. Des logiciels spécifiques existent également comme par exemple le logiciel AdventEdge [9] et sont en fait de vastes "boîtes noires" sur lesquelles il est impossible d'apporter des modifications du code source. Pour résumer, le constat actuel met en évidence le manque d'un code de calcul ouvert dédié à la simulation de la coupe. Ce constat, généralement effectué par la communauté scientifique qui travaille dans le domaine de la modélisation de la coupe est à l'origine notamment de la création du réseau de recherche Optimus (OPTimisation et Modélisation de l'Usinage à Grande Vitesse et/ou à Sec) en France.

Si l'on fait globalement la synthèse des difficultés liées au développement de modèles numériques de coupe, on est en mesure de dégager un certain nombre de caractéristiques qui doivent être intégrées afin de développer un code de calcul efficace dans le domaine de la coupe des métaux :

- choix d'un formalisme ALE,
- modèle comportemental et loi de rupture efficaces et fiables dans le domaine d'utilisation,
- algorithmes de contact et auto-contact efficaces,
- calcul parallèle et décomposition en sous-domaines,
- couplage multi-physique pour les interactions outil-copeau.

Ces divers points développés ci-après constituent une base de réflexion pour le développement d'un code de calcul dédié à la modélisation de la coupe des métaux.

### 1.3.1 Choix du formalisme adapté

Concernant le choix du formalisme à adopter pour la réalisation de modèles numériques de coupe, comme indiqué précédemment, deux principales voies sont disponibles : l'approche Eulérienne et l'approche Lagrangienne. Les travaux réalisés montrent que la tendance actuelle est à la réalisation de modèles de coupe transitoires discontinus, car ils correspondent à la réalité des processus physiques de coupe. Dans cette optique, la voie de la modélisation Lagrangienne doit être adoptée.

Le principal problème rencontré lors du développement de modèles Lagrangiens de coupe concerne la forte distorsion du maillage au cours de l'avancée de l'outil dans la matière. On peut globalement rappeler que l'une des approches utilisées pour résoudre ce problème réside dans l'utilisation d'algorithmes de remaillage. L'inconvénient principal de cette approche est lié aux performances des maillages, notamment en 3D, capables de réaliser ces remaillages automatiques. Le temps requis pour effectuer ces remaillages peut alors devenir problématique lorsque ceux-ci doivent être réalisés avec une fréquence trop élevée. Une autre approche permettant de réduire la distorsion du maillage au cours du calcul réside comme présenté précédemment dans l'utilisation d'un formalisme Euléro-Lagrangien Arbitraire qui semble être une solution efficace aux problèmes de distorsion des éléments.

L'une et l'autre de ces deux approches sont actuellement utilisées pour la modélisation des processus de coupe, mais il semble judicieux de penser que ni l'une ni l'autre ne sont totalement satisfaisantes à l'heure actuelle. Une alternative possible consiste à fédérer les avantages des deux approches en utilisant simultanément la formulation ALE et le remaillage lorsque la distorsion des éléments devient trop importante. Ceci permet de réduire notablement le nombre des phases de remaillage de la structure au cours du calcul.

### 1.3.2 Modélisation comportementale

La formulation mathématique et l'identification des coefficients des lois de comportement utilisées dans le cadre de la modélisation numérique de la coupe constitue un problème clé auquel il convient d'attacher une importance toute particulière. En effet les travaux développés sont basés sur l'utilisation de la loi d'écoulement plastique de Johnson-Cook [15], mais, on est en mesure de se demander si cette formulation est réellement adaptée à la coupe. De plus, les moyens de caractérisation mis en oeuvre permettent l'identification des coefficients de cette loi dans une plage de déformations restreinte par rapport à celle généralement rencontrée dans les processus de coupe d'où une extrapolation qui peut être source de discussions. En plus de l'identification de la loi d'écoulement plastique du matériau, il convient, compte tenu de l'approche développée, d'identifier les paramètres de la loi d'endommagement/rupture au sein du matériau. L'introduction de cette loi de rupture nécessite alors l'identification de 5

coefficients supplémentaires dans le cas de la loi de Johnson-Cook comme présenté au paragraphe 1.2.3.2.

Le code de calcul utilisé pour la modélisation de la coupe doit donc être en mesure d'utiliser les deux lois comportementales. L'approche généralement retenue alors consiste à utiliser une méthode de type 'érosion d'éléments' qui consiste à supprimer de la boucle de calcul tous les éléments dont le seuil d'endommagement est supérieur à une valeur fixée par l'utilisateur. Le premier inconvénient de cette approche réside alors dans le fait que l'on perd de la masse au cours du calcul. Le deuxième inconvénient, plus délicat à gérer est lié à la présence de vides au sein du matériau liés à l'érosion des éléments au cours du calcul. Ces vides créent alors un ensemble de surfaces libres dont il convient de gérer le contact en temps réel. Ce problème n'est par exemple actuellement pas pris en compte sur Abaqus.

Du point de vue de la loi de contact, des efforts doivent être entrepris. Actuellement, la majorité des modèles numériques développés sont basés sur le modèle de contact de Coulomb. Dans cette approche, le coefficient de frottement est une quantité macroscopique constante au cours du calcul. L'introduction de nouvelles lois de contact permettant une dépendance avec certains paramètres cinématiques et physiques doit être envisagée afin de permettre le développement de modèles numériques plus réalistes.

### 1.3.3 Calcul parallèle et décomposition en sous-domaines

Une des conséquences des points précédemment évoqués est alors l'augmentation de la taille et de la complexité des calculs à mettre en oeuvre pour développer des modèles de coupe relativement réalistes, l'implication directe étant une augmentation du temps de calcul nécessaire à la résolution du problème. On peut ainsi remarquer que l'augmentation des performances des moyens de calcul (puissance des ordinateurs) est généralement en deçà des besoins en matière de puissance nécessaire au développement de modèles numériques réalistes. L'une des alternatives possibles consiste alors à utiliser des moyens de calcul faisant intervenir des architectures parallèles (machines multiprocesseurs ou *clusters*<sup>(1)</sup>).

Dans cette même optique, les méthodes de décomposition en sous-domaines peuvent être également mises en oeuvre afin de réduire notablement les temps de calcul. De manière similaire, il semble également intéressant de mettre en oeuvre des méthodes de calcul multi-pas de temps permettant d'utiliser plusieurs pas de temps de résolution selon que l'on est proche ou loin de la zone critique en pointe d'outil.

## 1.4 Conclusions et axes de recherche à explorer

En conclusion de cette partie introductive présentant le bilan des travaux réalisés dans le domaine de la modélisation numérique de la coupe des métaux, il ressort deux points essentiels qui forment la base des recherches et développements que j'ai en partie encadré et en partie effectué personnellement durant la période allant de 1996 à ce jour.

La première voie de recherche concerne les activités liées à l'identification des paramètres des lois de comportement et des lois de contact à utiliser dans les modélisations numériques de coupe. Il ressort en effet des travaux réalisés que l'identification des paramètres de ces lois doit être effectuée en mettant en oeuvre un ensemble de moyens expérimentaux permettant non seulement de récréer au plus près les conditions réelles de sollicitation des matériaux au cours du processus de coupe, mais également de fournir par leur simplicité de mise en oeuvre

<sup>(1)</sup>Un *cluster* est un ensemble logique de serveurs qui garantissent une haute disponibilité des ressources et une répartition des charges de traitement.



et leur universalité des moyens d'identifier le comportement des matériaux pour d'autres types de sollicitations que celles rencontrées lors de l'usinage. Cette approche est plus large que celle utilisée par certains auteurs qui identifient directement les lois de comportement à partir d'essais d'usinage instrumentés. Dans cette perspective, en accord avec les objectifs du groupe G2TR animé par Serge Caperaa, j'ai entrepris le développement d'une plateforme d'identification expérimentale du comportement mécanique de matériaux sous sollicitations dynamiques. Sa présentation et les applications réalisées font l'objet du prochain chapitre de ce mémoire.

La deuxième voie de recherche concerne les activités personnelles que j'ai engagé dans le cadre du développement d'un code de calcul en grandes transformations (le code de calcul **DynELA**) au sein de l'équipe CMAO du laboratoire LGP. Le but de la mise en oeuvre de ce code est de disposer au sein du laboratoire d'un code entièrement maîtrisé et ouvert servant de base à des développements d'algorithmes et de méthodes de résolution novateurs. Les fondements du développement de ce code de calcul seront exposés dans le chapitre 3, et quelques développements novateurs concernant la mise en oeuvre du calcul multi-domaine, les méthodes multi-pas de temps et l'implémentation de la méthode XFEM seront présentés dans le chapitre 4. Du point de vue modélisation numérique de l'usinage, ce code de calcul a été retenu afin de servir de base de développement d'un outil de modélisation de la coupe des métaux par le comité de pilotage du réseau OPTIMUS (OPTimisation et Modélisation de l'Usinage à Grande Vitesse et/ou à Sec) lors du second séminaire à l'ENSAM de Cluny en octobre 2004 [33].

*Ces travaux ont fait l'objet de 5 publications internationales et de 14 participations à des congrès.*



# 2

# Lois de comportement en Grandes Transformations

---

*Les théories ont causé plus d'expériences que les expériences n'ont causé de théories. On voit par là de quelle utilité est au progrès des arts ce qui est purement rationnel dans chaque science.*

Joseph Joubert (1754-1824)

La prise en compte du comportement mécanique du matériau au travers de la formulation des lois de comportement et de l'identification des paramètres de ces lois est un élément fondamental du processus de simulation. Une fois que le choix de la formulation mathématique de la loi de comportement est effectué, bien souvent les paramètres de cette loi ne sont pas disponibles ou sont incomplets lorsque l'on souhaite passer à l'étape de simulation numérique. Un effort tout particulier doit donc porter sur la mise en place d'essais expérimentaux permettant de recréer des conditions de sollicitation proches de celles rencontrées lors du processus physique et d'identifier à partir de ces essais les paramètres numériques de la loi de comportement.

Dans ce chapitre, on présente une formulation des lois de comportement hypo-élastiques en grandes transformations au travers de la description du formalisme utilisé, de l'intégration temporelle de ces lois et de la présentation de quelques formes de lois d'écoulement classiquement utilisées. Dans une deuxième partie, on présente les moyens et les essais expérimentaux développés au sein du laboratoire afin d'identifier les paramètres des lois de comportement. La troisième partie, enfin, est consacrée à la présentation de la procédure d'identification complétée par un exemple d'application illustratif.

## 2.1 Formulation des lois de comportement en grandes transformations

On peut globalement distinguer deux approches de développement des modèles de comportement qui se distinguent l'une de l'autre par l'échelle physique utilisée lors de leur élaboration. La première approche concerne les modèles basés sur l'étude de la microstructure du matériau qui considère que les métaux et les alliages sont constitués d'assemblages de cristaux dans lesquels les atomes sont liés entre eux selon des motifs élémentaires répétitifs dans les 3 directions de l'espace. Ces mailles métalliques correspondent à l'un des trois systèmes :

- cubique centré (CC) correspondant par exemple à la phase  $\alpha$  (ferrite) du fer et au chrome,
- cubique à faces centrées (CFC) correspondant à la phase  $\gamma$  (austénite) du fer et à l'aluminium,

– hexagonal compact (HC) correspondant au magnésium et au zinc.

Dans cette approche, un métal est considéré comme un polycristal, c'est à dire un agrégat de grains d'orientations cristallographiques différentes aux comportements individuels parfaitement caractérisés. Cette approche, qui conduit à l'établissement de lois de comportement physiques, est particulièrement riche mais extrêmement lourde à mettre en oeuvre et son utilisation dans les modèles numériques de simulation de structure est très limitée.

Dans la deuxième approche, l'approche comportementale, on cherche à caractériser globalement le comportement du matériau en faisant abstraction de la structure microscopique : c'est une approche mésoscopique. On détermine alors un ensemble de relations liant des causes et des effets en fonction de chaque matériau étudié ; ce sont des lois de comportement expérimentales. On peut également s'appuyer sur une analyse à l'échelle microscopique afin de développer des modèles de comportement à l'échelle mésoscopique que l'on qualifiera alors de semi-physiques.

Dans la suite de ce travail, et compte tenu de la nature des applications, seules les lois de comportement décrites à l'échelle mésoscopique seront développées. De manière restreinte, on ne considèrera que le cas des lois de comportement hypo-élastiques et leur formulation en grandes transformations. On présente ci-après la méthodologie générale de traitement des lois de comportement hypo-élastiques en grandes transformations, et on renvoie le lecteur aux ouvrages de Simo et al. [34] et Belytschko et al. [35] ainsi qu'au manuel théorique de **DynELA** [36] pour les compléments concernant le développement de ces méthodes.

### 2.1.1 Formulation hypo-élastique en grandes transformations

La réponse d'un matériau soumis à des sollicitations mécaniques est caractérisée par son équation constitutive qui permet de déterminer les contraintes en fonction de l'histoire de la déformation. Dans la version actuelle du code de calcul **DynELA**, les lois de conservation et la loi de comportement sont formulées sur la base d'une description Lagrangienne du mouvement. Pour cette description,  $\vec{X}$  représente les coordonnées de référence d'un point matériel dans la configuration de référence  $\Omega_X \subset \mathbb{R}^3$  à l'instant  $t = 0$ , et  $\vec{x}$  représente les coordonnées spatiales du même point de référence dans la configuration  $\Omega_x \subset \mathbb{R}^3$  à l'instant  $t$ . Le mouvement du solide matériel est alors défini par la loi de mouvement  $\vec{x} = \vec{\phi}(\vec{X}, t)$  et on introduit alors le gradient de déformation  $\mathbf{F} = \partial \vec{x} / \partial \vec{X}$  par rapport à la configuration de référence  $\Omega_X$ . La décomposition polaire permet alors d'écrire  $\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}$ , avec  $\mathbf{U}$  et  $\mathbf{V}$  qui sont respectivement les tenseurs de déformation pure droit et gauche et  $\mathbf{R}$  qui est le tenseur de rotation. La formulation des lois de comportement est classiquement basée sur une décomposition additive<sup>(1)</sup> de la partie symétrique  $\mathbf{D}$  du tenseur des vitesses de déformation  $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$  en une partie élastique  $\mathbf{D}^e$  et une partie plastique  $\mathbf{D}^p$  :

$$\mathbf{D} = \mathbf{D}^e + \mathbf{D}^p \quad (2.1)$$

Dans la mesure où le code de calcul **DynELA** est dédié aux grandes transformations, nous devons assurer l'objectivité des termes qui entrent dans l'écriture des équations de comportement. La relation contrainte/déformation hypo-élastique mettant en relation le tenseur des contraintes  $\boldsymbol{\sigma}$  et la partie élastique  $\mathbf{D}^e$  du tenseur des vitesses de déformation s'écrit alors sous la forme suivante :

$$\overset{\nabla}{\boldsymbol{\sigma}} = \mathbb{C} : \mathbf{D}^e = \mathbb{C} : (\mathbf{D} - \mathbf{D}^p) \quad (2.2)$$

<sup>(1)</sup>Rappelons que d'un point de vue formel cette décomposition additive n'est pas licite. Néanmoins, elle est adoptée par la plupart des numériciens sans la discuter [37], justifiée notamment par le fait que  $\mathbf{D}^e$  est petit.

dans laquelle  $\overset{\nabla}{\boldsymbol{\sigma}}$  est une dérivée objective du tenseur des contraintes de Cauchy et  $\mathbb{C}$  est le tenseur de Hooke donné par :

$$\mathbb{C} = K\mathbf{I} \otimes \mathbf{I} - 2G\left(\mathbb{I} - \frac{1}{3}\mathbf{I} \otimes \mathbf{I}\right) \quad (2.3)$$

où  $K$  est le module de compressibilité du matériau,  $G$  est le module de cisaillement du matériau,  $\mathbf{I}$  est le symbole de Kronecker et  $\mathbb{I}$  est le tenseur unité du 4<sup>ème</sup> ordre. Le taux d'écoulement plastique  $\mathbf{D}^p$  s'écrit sous la forme suivante :

$$\mathbf{D}^p = \Lambda \mathbf{n}(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.4)$$

dans laquelle  $\Lambda$  est un scalaire représentant la norme de l'écoulement plastique et  $\mathbf{n}(\boldsymbol{\sigma}, \mathbf{q})$  est la direction de l'écoulement plastique unitaire ( $\mathbf{n} : \mathbf{n} = 1$ ) dépendant du tenseur des contraintes de Cauchy  $\boldsymbol{\sigma}$  et d'un ensemble de paramètres d'hérédité collectivement dénotés  $\mathbf{q}$ . Ces paramètres d'hérédité du matériau représentent par exemple la déformation plastique cumulée  $\bar{\varepsilon}^p$  du matériau, une variable d'endommagement ou le *back-stress*  $\boldsymbol{\alpha}$  dans les modèles comportementaux cinématiques. On a alors recours à des relations décrivant l'évolution de ces variables d'hérédité  $\mathbf{q}$  qui dans la plupart des modèles de plasticité se mettent sous la forme incrémentale suivante :

$$\dot{\mathbf{q}} = \Lambda \mathbf{r}(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.5)$$

On admet alors communément l'existence d'une fonction  $f(\boldsymbol{\sigma}, \mathbf{q})$  appelée fonction de charge qui divise l'espace des contraintes en trois zones distinctes : la zone de comportement élastique (pour laquelle  $f(\boldsymbol{\sigma}, \mathbf{q}) < 0$ ), la zone de plasticité ( $f(\boldsymbol{\sigma}, \mathbf{q}) = 0$ ) et une zone interdite ( $f(\boldsymbol{\sigma}, \mathbf{q}) > 0$ ) comme illustré sur la figure 2.1.

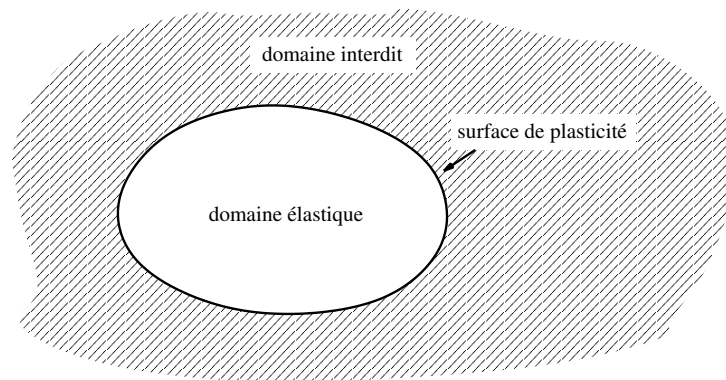


FIG. 2.1 – Surface de plasticité

Les états de contrainte admissibles sont donc ceux qui vérifient la relation  $f(\boldsymbol{\sigma}, \mathbf{q}) \leq 0$ . A partir de cette notion, la norme de l'écoulement plastique  $\Lambda$  et les lois d'évolution des paramètres d'hérédité sont obtenues en utilisant la condition de consistance  $\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0$  écrite ci-dessous :

$$\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \overset{\nabla}{\boldsymbol{\sigma}} + \frac{\partial f}{\partial \mathbf{q}} : \dot{\mathbf{q}} = 0 \quad (2.6)$$

Lors d'une déformation élastique, le point représentatif dans l'espace des contraintes se déplace à l'intérieur de la surface de charge et on a  $f < 0$  et  $\Lambda = 0$ , et lors d'une déformation plastique il reste sur la surface de charge et on a  $f = 0$  et  $\Lambda > 0$ . Les conditions de charge et de décharge peuvent se mettre sous la forme de Khun-Tucker d'un problème d'optimisation :

$$\Lambda \geq 0, f(\boldsymbol{\sigma}, \mathbf{q}) \leq 0, \Lambda f(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad (2.7)$$

L'équation de consistance (2.6) fait intervenir la normale à la surface de charge  $\frac{\partial f}{\partial \boldsymbol{\sigma}}$ . Quand la direction de l'écoulement plastique  $\mathbf{n}$  est normale à la surface de charge, l'écoulement plastique est qualifié du terme de plasticité associative. Dans ce cas, et compte tenu que  $\mathbf{n}$  est unitaire, on peut écrire :

$$\mathbf{n} = \frac{\frac{\partial f}{\partial \boldsymbol{\sigma}}}{\left\| \frac{\partial f}{\partial \boldsymbol{\sigma}} \right\|} \quad (2.8)$$

Un cas particulier généralement utilisé consiste à définir la surface de plasticité à partir de la notion de contrainte équivalente de von Mises. Cette contrainte équivalente étant basée sur le second invariant, on emploie couramment le terme générique de plasticité de type  $J_2$ . Ce modèle est particulièrement adapté au cas des métaux. Le concept de base est que l'écoulement plastique est indépendant de la pression comme l'a montré expérimentalement Bridgman [38]. Ainsi, la surface de charge et la direction de l'écoulement plastique sont construits à partir de la partie déviatorique  $\mathbf{S} = \boldsymbol{\sigma} - \frac{1}{3}\text{tr}[\boldsymbol{\sigma}]\mathbf{I}$  du tenseur des contraintes et on écrit alors :

$$f = \bar{\sigma} - \sigma^v = \sqrt{\frac{3}{2}\boldsymbol{\phi}:\boldsymbol{\phi}} - \sigma^v, \quad (2.9)$$

dans laquelle  $\sigma^v$  est la limite apparente d'élasticité du matériau et  $\boldsymbol{\phi} = \mathbf{S} - \boldsymbol{\alpha}$  avec  $\boldsymbol{\alpha}$  variable tensorielle d'hérédité appelée *back-stress* qui est nulle si l'écrouissage est isotrope. A partir de (2.9), on peut à nouveau exprimer la normale à la surface de charge donnée en (2.8) sous la forme suivante :

$$\mathbf{n} = \frac{\boldsymbol{\phi}}{\sqrt{\boldsymbol{\phi}:\boldsymbol{\phi}}} \quad (2.10)$$

Lors d'une déformation plastique, les lois d'évolution des paramètres d'écrouissage  $\sigma^v$  et  $\boldsymbol{\alpha}$  pour un comportement élastoplastique à écrouissage mixte sont données par :

$$\dot{\sigma}^v = \sqrt{\frac{2}{3}}h^i\Lambda; \quad \dot{\boldsymbol{\alpha}} = \frac{2}{3}h^k\Lambda\mathbf{n} \quad (2.11)$$

avec  $h^i$  et  $h^k$  coefficients d'écrouissage isotrope et cinématique respectivement correspondant à la pente de la courbe d'écrouissage unidimensionnelle exprimant la contrainte équivalente de von Mises  $\bar{\sigma}$  en fonction de la déformation plastique équivalente  $\bar{\varepsilon}^p$ . Ainsi, après décomposition du tenseur des contraintes de Cauchy  $\boldsymbol{\sigma}$  en une partie déviatorique  $\mathbf{S}$  et un terme hydrostatique  $p$ , la relation contrainte/déformation hypo-élastique peut alors être mise sous la forme suivante :

$$\overset{\nabla}{\mathbf{S}} = \mathbb{C} : \mathbf{D}^e; \quad \dot{p} = K\text{tr}[\mathbf{D}] \quad (2.12)$$

Une des principales difficultés survenant lors de la phase d'intégration des lois de comportement est liée à la nécessité d'utiliser une dérivée objective du tenseur des contraintes de Cauchy dans la mesure où le code de calcul **DynELA** est dédié aux grandes transformations. Un moyen largement utilisé pour cela consiste à écrire les équations constitutives dans un repère co-rotationnel défini à partir d'un tenseur rotation  $\mathbf{w}$  avec  $\dot{\mathbf{w}} = \boldsymbol{\omega}\mathbf{w}$  et  $\mathbf{w}(t=0) = \mathbf{I}$ . En définissant une correspondance entre une quantité  $()$  et son image dans le repère co-rotationnel  $()^c$ , on obtient alors pour les quantités  $\rho$ ,  $\boldsymbol{\sigma}$  et  $\mathbb{C}$  qui sont respectivement un scalaire, un tenseur du second ordre et un tenseur du quatrième ordre les lois de transformation suivantes :

$$\rho^c = \rho; \quad \boldsymbol{\sigma}^c = \mathbf{w}^T \boldsymbol{\sigma} \mathbf{w}; \quad \mathbb{C}^c = \mathbf{w}^T [\mathbf{w}^T \mathbb{C} \mathbf{w}] \mathbf{w} \quad (2.13)$$

Pour les détails concernant ce changement de référentiel, le lecteur pourra se référer par exemple aux travaux de J.P. Ponthot [37]. Le choix d'un repère co-rotationnel tel que  $\boldsymbol{\omega} = \mathbf{W}$  avec  $\mathbf{W}$  partie anti-symétrique du tenseur gradient des vitesses de déformation  $\mathbf{L}$  conduit à

un comportement similaire à celui obtenu en utilisant une dérivée objective de Jaumann. Toutes les équations ci-dessus peuvent alors être réécrites dans le repère co-rotationnel en substituant à tous les termes  $()$  leur pendant co-rotationnel  $()^c$  et en remplaçant la dérivée objective par une simple dérivée temporelle. Compte tenu de son caractère isotrope, le tenseur de Hooke  $\mathbb{C}$  est tel que  $\mathbb{C}^c = \mathbb{C}$ .

**Remarque :** Pour toute la suite, et afin de ne pas surcharger les expressions mathématiques, on conviendra d'exprimer toutes les équations de comportement dans le repère co-rotationnel et d'omettre l'exposant  $()^c$  représentatif d'une quantité co-rotationnelle dans les expressions excepté dans les relations de passage.

### 2.1.2 Intégration temporelle des équations constitutives

Le but de l'intégration temporelle est d'actualiser dans la configuration actuelle  $\Omega_x$  à  $t = t_{n+1}$  les variables d'état (contraintes  $\boldsymbol{\sigma}$  et paramètres d'hérédité  $\mathbf{q}$ ), connues dans la configuration de référence  $\Omega_X$  à  $t = t_n$ . On posera pour la suite  $\Delta t = t_{n+1} - t_n$ . Pour réaliser cette intégration temporelle, compte tenu de (2.1) on adopte alors un schéma d'intégration de type prédicteur élastique + correcteur plastique (schéma du retour radial) défini par [37, 34, 35]. Le prédicteur élastique de la partie pression  $p_{tr}$  et le prédicteur élastique de la partie déviatorique  $\mathbf{S}_{tr}$  du tenseur des contraintes  $\boldsymbol{\sigma}$  est calculé conformément à la loi de Hooke par les relations suivantes :

$$p_{tr} = p_n + K \text{tr}[\Delta \mathbf{e}], \quad \mathbf{S}_{tr} = \mathbf{S}_n + 2G \text{dev}[\Delta \mathbf{e}] \quad (2.14)$$

dans lesquelles  $\Delta \mathbf{e} = \ln[\mathbf{U}] = \ln[\mathbf{F}^T \mathbf{F}]/2$  est le tenseur naturel ou logarithmique de déformation co-rotationnelle entre les instant  $t_n$  et  $t_{n+1}$ . Compte tenu que la configuration à  $t = t_n$  correspond à la configuration de référence (soit donc  $\vec{X} = \vec{x}(t_n)$ ), on peut poser  $\mathbf{w} = \mathbf{R}$  qui est issu de la décomposition polaire de  $\mathbf{F}$ . La prise en compte de l'objectivité qui fait intervenir l'écriture des lois de comportement dans le repère co-rotationnel consiste alors à transporter les variables d'état du repère co-rotationnel vers le repère physique à la suite de la phase d'intégration temporelle, c'est la Rotation Finale Instantanée (RFI). On obtient l'écriture suivante pour le prédicteur du déviateur des contraintes de Cauchy  $\mathbf{S}_{tr}$  et du *back-stress*  $\boldsymbol{\alpha}_n$  :

$$\mathbf{S}_{tr} \leftarrow \mathbf{R} \cdot \mathbf{S}_{tr}^c \cdot \mathbf{R}^T, \quad \boldsymbol{\alpha}_n \leftarrow \mathbf{R} \cdot \boldsymbol{\alpha}_n^c \cdot \mathbf{R}^T \quad (2.15)$$

A la suite de la détermination du prédicteur élastique du déviateur des contraintes  $\mathbf{S}_{tr}$ , on doit évaluer si celui-ci est physiquement admissible conformément à la surface de plasticité (2.9). On doit donc vérifier la relation :

$$f(\mathbf{S}_{tr}, \mathbf{q}_n) = 0 \quad (2.16)$$

Si  $f \leq 0$  alors, la contrainte prédite est physiquement admissible et l'incrément est entièrement élastique, on pose alors  $\Gamma = \int_{t_n}^{t_{n+1}} \Lambda dt = 0$ . Sinon, les conditions de Kuhn-Tucker sont violées par le prédicteur élastique et la consistance doit être restaurée. On utilise alors l'algorithme de retour radial défini dans la boite 2.1 afin de déterminer la valeur du paramètre  $\Gamma$ .

A la suite de cette opération, la part élastique et la part plastique de la déformation peuvent être séparées l'une de l'autre, et on peut alors déterminer le déviateur des contraintes à l'instant  $t_{n+1}$  en utilisant la relation :

$$\mathbf{S}_{n+1} = \mathbf{S}_n - 2G\Gamma \mathbf{n} \quad (2.17)$$

qui permet alors de calculer la valeur des contraintes à la fin de l'incrément :

$$\boldsymbol{\sigma}_{n+1} = \mathbf{S}_{n+1} + p\mathbf{I} \quad (2.18)$$

L'évolution des variables d'hérédité  $\bar{\varepsilon}^p$  et  $\boldsymbol{\alpha}$  est obtenue par les relations :

$$\bar{\varepsilon}^p_{n+1} = \bar{\varepsilon}^p_n + \sqrt{\frac{3}{2}}\Gamma, \quad \boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_{tr} + \frac{2}{3}h^k\Gamma \mathbf{n} \quad (2.19)$$

Algorithme 2.1 – Algorithme de retour-radial pour un écrouissage combiné isotrope/cinématique

1. Calcul des coefficients d'écrouissage  $h_n^{(i,c)}(\bar{\epsilon}^p_n)$  et de la limite apparente d'élasticité  $\sigma_n^v(\bar{\epsilon}^p_n)$
2. Calcul de la valeur du paramètre scalaire  $\Gamma^{(1)}$  défini par :

$$\Gamma^{(1)} = \frac{\sqrt{\Phi_{tr} : \Phi_{tr}} - \sqrt{\frac{2}{3}}\sigma_n^v}{2G \left(1 + \frac{h_n^i + h_n^c}{3G}\right)}$$

3. Boucle sur la condition de consistance pour  $k = 1$

- (a) Calcul de  $\sigma_{n+1}^v(\bar{\epsilon}^p_n + \sqrt{\frac{2}{3}}\Gamma^{(k)})$ ,  $h_{n+1}^{(i,c)}(\bar{\epsilon}^p_n + \sqrt{\frac{2}{3}}\Gamma^{(k)})$  et  $\alpha_{n+1} = \alpha_n + \frac{2}{3}h_{n+1}^c\Gamma^{(k)}\mathbf{n}$
- (b) Calcul de  $g = 2G\sqrt{\frac{3}{2}}\Gamma^{(k)} - \bar{\sigma} + \sigma_{n+1}^v$  et  $dg = 2G\sqrt{\frac{3}{2}} + \sqrt{\frac{2}{3}}(h_{n+1}^c + h_{n+1}^i)$
- (c) Si  $\frac{g}{\sigma_{n+1}^v} < tolerance$  aller en 4
- (d) Calcul de  $\Gamma^{(k+1)} = \Gamma^{(k)} - g/dg$
- (e)  $k \leftarrow k + 1$  et retour en 3a

4. La convergence est atteinte et  $\Gamma = \Gamma^{(k)}$

### 2.1.3 Lois d'écoulement

Comme présenté dans le paragraphe précédent, lors de l'intégration des lois de comportement élastoplastiques, le comportement du matériau est décrit par la loi d'écrouissage définissant la limite apparente d'élasticité  $\sigma^v$  comme donnée du problème. Les formes de loi d'écoulement sont très variées dans la littérature et sont bien souvent spécifiques à une classe de matériaux ou une classe d'applications. Nous n'en présenterons qu'une petite sélection dans ce paragraphe, spécifiques au domaine d'application qui nous concerne.

- La forme la plus simple de loi d'écoulement élastoplastique non linéaire est la loi de Ramberg-Osgood qui se met sous la forme suivante :

$$\sigma^v = \sigma_0^v + K_Y (\bar{\epsilon}^p)^{\frac{1}{M_Y}} \quad (2.20)$$

dans laquelle  $\sigma_0^v$  est la limite d'élasticité initiale,  $K_Y$  est le coefficient de résistance plastique et  $M_Y$  est l'exposant d'écrouissage. Cette loi ne propose qu'une dépendance avec la déformation plastique équivalente  $\bar{\epsilon}^p$ .

- D'autres formes de loi ne dépendant que de la déformation plastique équivalente  $\bar{\epsilon}^p$  existent, ainsi, à partir des essais réalisés sur un acier spécial utilisé pour la construction de tuyauteries dans l'industrie nucléaire (acier A-533 Grade B classe 1) par Norris et al. [39], Simo et al. [34] proposent une loi d'écoulement non-linéaire donnée par la relation suivante :

$$\sigma^v = \sigma_0^v + (\sigma_\infty^v - \sigma_0^v)(1 - \exp(-\delta\bar{\epsilon}^p)) + \xi\bar{\epsilon}^p \quad (2.21)$$

dans laquelle  $\sigma_0^v$  est la limite d'élasticité initiale,  $\sigma_\infty^v$  est la limite asymptotique d'élasticité et  $\delta$  et  $\xi$  sont des constantes d'atténuation.

- Afin d'enrichir les dépendances de la loi d'écoulement, de nombreux auteurs proposent des lois dépendant non seulement de la déformation plastique équivalente  $\bar{\epsilon}^p$ , mais également de la vitesse de déformation plastique  $\dot{\bar{\epsilon}}^p$  et de la température  $T$ . Le modèle de Zerilli-Armstrong



[40] résulte d'une approche semi-physique basée sur la théorie des dislocations et propose deux formes distinctes de lois d'écoulement selon la nature cubique centrée (CC) ou cubique faces centrées (CFC) de la structure cristallographique du matériau étudié :

$$\begin{aligned}\sigma^v &= (C_0 + C_2 \bar{\varepsilon}^n) \exp\left(-C_3 T + C_4 T \ln \frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0}\right) & \text{CFC} \\ \sigma^v &= C_0 + C_1 \exp\left(-C_3 T + C_4 T \ln \frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0}\right) + C_5 \bar{\varepsilon}^n & \text{CC}\end{aligned}\quad (2.22)$$

- Historiquement, la loi d'écoulement la plus répandue dans le domaine des applications dynamiques est la loi de Johnson-Cook [15] dont la formulation dans sa forme originale est donnée ci-dessous :

$$\sigma^v = (A + B \bar{\varepsilon}^n) \left(1 + C \ln \frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0}\right) \left[1 - \left(\frac{T - T_0}{T_{melt} - T_0}\right)^m\right] \quad (2.23)$$

Dans cette expression,  $T_{melt}$  est la température de fusion,  $T_0$  est la température initiale,  $\dot{\varepsilon}_0$  est la vitesse de déformation de référence et  $A$ ,  $B$ ,  $C$ ,  $n$  et  $m$  sont les paramètres liés au matériau qu'il convient d'identifier.  $A$  est la limite élastique du matériau,  $B$  et  $n$  décrivent la courbe d'écrouissage,  $C$  est la sensibilité à la vitesse de déformation et  $m$  la sensibilité à la température. C'est une adaptation de la loi d'écrouissage de Ramberg-Osgood à laquelle on ajoute un terme de sensibilité à la vitesse de déformation caractérisé par la constante  $C$  et un terme de sensibilité à la température caractérisé par la constante  $m$ . Cette forme originale est implémentée dans plusieurs codes de calcul commerciaux comme Abaqus [41].

- Par la suite, un certain nombre de travaux complémentaires ont conduit à compléter et à améliorer la forme originale de la loi d'écoulement de Johnson-Cook. On doit ainsi à Holmquist et al. [42] la forme dite modifiée qui utilise une expression puissance pour le terme de sensibilité à la vitesse de déformation :

$$\sigma^v = (A + B \bar{\varepsilon}^n) \left(\frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0}\right)^C \left[1 - \left(\frac{T - T_0}{T_{melt} - T_0}\right)^m\right] \quad (2.24)$$

- Plus récemment, Rule et al. [43] ont proposé une forme révisée dans laquelle l'influence de  $\frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0}$  est accentuée pour des valeurs de vitesse de déformation élevées. L'expression proposée pour la contrainte d'écoulement se met sous la forme :

$$\sigma^v = (A + B \bar{\varepsilon}^n) \left(1 + C \ln \frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0} + C_1 \left(C_2 - \ln \frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_0}\right)^{-1} - \frac{1}{C_2}\right) \left[1 - \left(\frac{T - T_0}{T_{melt} - T_0}\right)^m\right] \quad (2.25)$$

L'utilisation de ces lois d'écoulement dans un code de calcul nécessite une étape d'identification des divers paramètres qui entrent dans la formulation de la loi retenue. Il est donc nécessaire d'identifier un nombre de paramètres compris entre 3 pour la loi d'écoulement de Ramberg-Osgood (2.20) et 7 pour la forme révisée de la loi d'écoulement de Johnson-Cook (2.25) à partir d'essais expérimentaux permettant de créer et de contrôler des niveaux de sollicitation proches du domaine d'utilisation. C'est dans cette optique que le laboratoire d'identification du comportement des matériaux soumis à de fortes sollicitations thermomécaniques dont j'ai pris la responsabilité a été créé. Sa présentation, fait l'objet de la prochaine section.

## 2.2 Le laboratoire d'identification

Afin de caractériser le comportement dynamique des matériaux on fait appel à divers dispositifs expérimentaux selon la nature (gamme de déformations et de vitesses de déformations) de la sollicitation souhaitée comme illustré sur la figure 2.2. Il ressort de l'analyse de cette figure

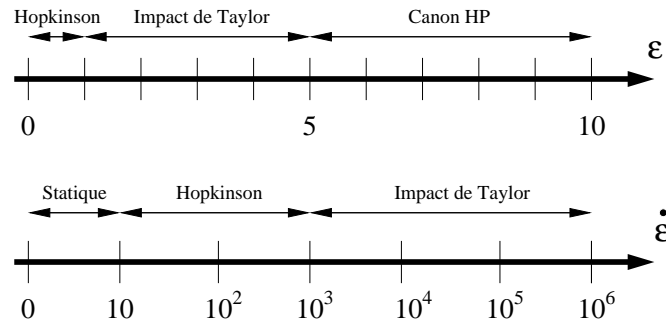


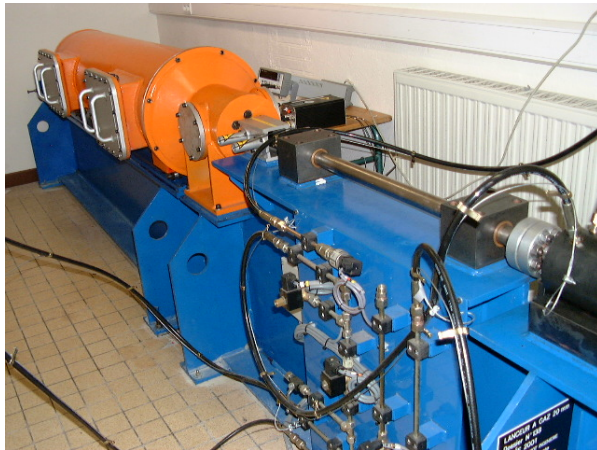
FIG. 2.2 – Essais expérimentaux en fonction des déformations et taux de déformation

que, comme présenté au paragraphe 1.2.1 page 14, les essais d'impact de Taylor couvrent une grande part de la gamme de déformations et de vitesses de déformations typiques à l'usinage. Ils se situent entre les essais utilisant des barres de Hopkinson (pour lesquels la gamme des déformations est relativement restreinte) et les essais utilisant des moyens pyrotechniques (donc impossibles à mettre en oeuvre, pour des raisons de sécurité, dans un contexte de laboratoire rattaché à un établissement public d'enseignement et de recherche).

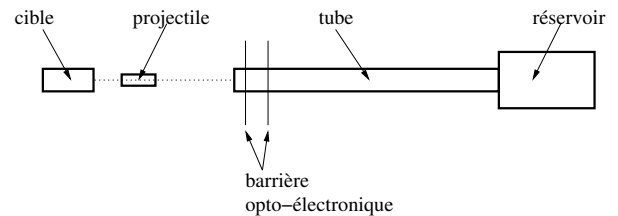
### 2.2.1 Moyens expérimentaux du laboratoire

Après concertation avec l'ensemble des membres du groupe de recherche G2TR, le choix de l'équipement du laboratoire d'identification, au cours du premier semestre de l'année 2001, s'est porté sur un lanceur à gaz de calibre  $\phi = 20\text{mm}$  (voir figure 2.3) capable de lancer des projectiles constitués de divers matériaux (acier, aluminium, composites...) dans une gamme de vitesses variant de  $30\text{m/s}$  à  $350\text{m/s}$  pour une charge totale de 30 grammes. Le dispositif mis en oeuvre comprend une chambre de tir, un dispositif de mesure de vitesse par barrière opto-électronique et un système de récupération des échantillons. L'ensemble est dans une enceinte étanche et les tirs sont effectuées sous vide partiel à quelques milli-bar. Dans tous les essais réalisés, on utilise un sabot en polycarbonate afin de guider le projectile à l'intérieur du tube du lanceur. Lors des tirs, ce sabot est fragilisé de manière à absorber un minimum d'énergie lors de sa rupture et d'interférer au minimum avec les conditions de l'impact. Néanmoins, la présence de ce sabot est délicate à prendre en compte et son influence n'est pas nulle. Une étude est actuellement en cours afin de développer un dispositif permettant de séparer le projectile du sabot avant l'impact.

Pour des raisons principalement de coût, les mesures effectuées au cours de ces essais sont des analyses post-mortem des échantillons. Dans un premier temps, les mesure dimensionnelles ont été effectuées à l'aide d'une machine de mesure tri-dimensionnelle. L'inconvénient principal de ce type de mesure réside dans le temps mis pour effectuer ces mesures et dans les problèmes liés à la disponibilité de ce matériel de mesure. Afin de simplifier et de faciliter la tâche d'analyse dimensionnelle, on utilise actuellement un procédé macro-photographique basé sur un appareil photo numérique Nikon D1X muni d'un objectif macro 60mm 1:2.8 D (rapport macro 1:1). Les performances de cet appareil photo permettent d'obtenir une image numérique de  $4028 \times 2648$  pixels pour laquelle la résolution spatiale est de  $5,9 \cdot 10^{-3}\text{mm}$  pour un rapport macro de 1:1.



(a) Vue d'ensemble



(b) Schéma de principe

FIG. 2.3 – Lanceur à gaz du laboratoire

L'image acquise est alors traitée par le logiciel **imageAnalyser** [44] (voir figure 2.4), dont j'ai pris en charge le développement, qui permet notamment de construire un repère 2D sur l'image (origine + axes) afin d'associer à chaque pixel de l'image des coordonnées réelles en millimètres. Les principaux avantages de cette approche sont la rapidité d'obtention des résultats (moins de 5 minutes), et l'autonomie puisque la mesure est effectuée directement dans le laboratoire d'impact à proximité du lanceur. La figure 2.5 montre les écarts de mesure obtenus sur un cylindre de Taylor entre le procédé optique et une mesure sur machine tri-dimensionnelle. Ces résultats permettent de valider le procédé de mesure avec un écart absolu inférieur à  $1/100^{\text{ème}}$  de millimètre.

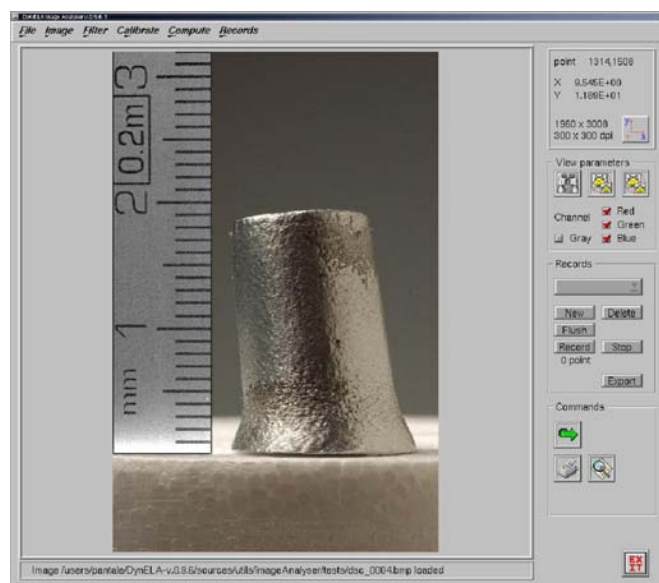


FIG. 2.4 – Fenêtre principale de l'application imageAnalyser

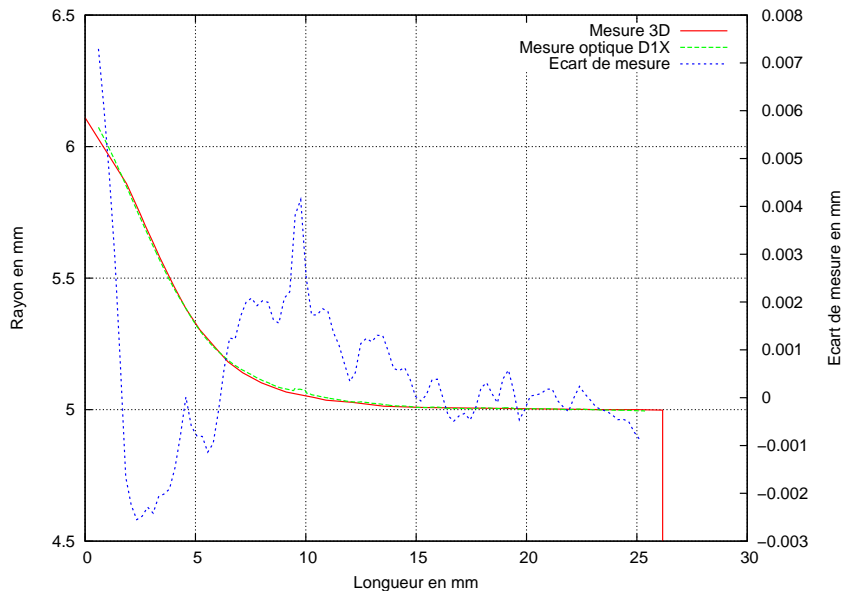


FIG. 2.5 – Comparaison entre mesure optique et mesure 3D

## 2.2.2 Les divers essais expérimentaux développés

Plusieurs essais dynamiques ont été développés selon la nature de la sollicitation souhaitée, principalement en traction, compression et cisaillement. Certains sont classiques (essai de Taylor et essai de cisaillement), d'autres sont totalement novateurs (essai de traction dynamique et essai de frottement).

### 2.2.2.1 Essai d'impact de Taylor

Le premier essai développé avec le dispositif expérimental est l'essai d'impact de Taylor [45, 46]. Cet essai, qui permet de caractériser le comportement principalement en compression des matériaux aux forts taux de déformation ( $10^4 - 10^5 s^{-1}$ ), consiste à projeter un échantillon cylindrique (ici, de diamètre  $\phi = 10mm$  et longueur  $l = 28mm$ ) sur une cible rigide dans le cas de l'essai de Taylor direct ou sur un échantillon identique dans le cas d'un essai de Taylor symétrique [47]. La difficulté principale de réalisation d'un essai de Taylor symétrique réside dans l'alignement de l'axe de la cible avec l'axe du projectile lors du tir. La figure 2.6 montre les échantillons et le sabot après impact pour un acier 42CrMo4 dans une configuration d'essai symétrique pour une vitesse d'impact  $V_c = 328m/s$ .

La figure 2.7 montre les profils de la cible et du projectile après déformations mesurés à l'aide du logiciel **imageAnalyser**. Les deux profils sont identiques, ce qui traduit les bonnes conditions de réalisation de l'essai. Ces profils sont alors utilisés comme réponses expérimentales dans le processus d'identification de la loi de comportement qui sera présenté au paragraphe 2.3.

La gamme des déformations pouvant être obtenues avec des essais d'impact de Taylor est relativement large comme illustré sur la figure 2.8 présentant l'état déformé d'un échantillon d'aluminium, de dimensions identiques à celui en acier, après un essai d'impact direct sur une cible en acier pour vitesse d'impact  $V_c = 278m/s$ . Les très forts taux de déformation obtenus lors de cet essai conduisent ici à la présence de zones endommagées et fissurées sur le projectile.



(a) Echantillon cible déformé

(b) Sabot et échantillon projectile déformé

FIG. 2.6 – Essai de Taylor symétrique pour un acier 42CrMo4

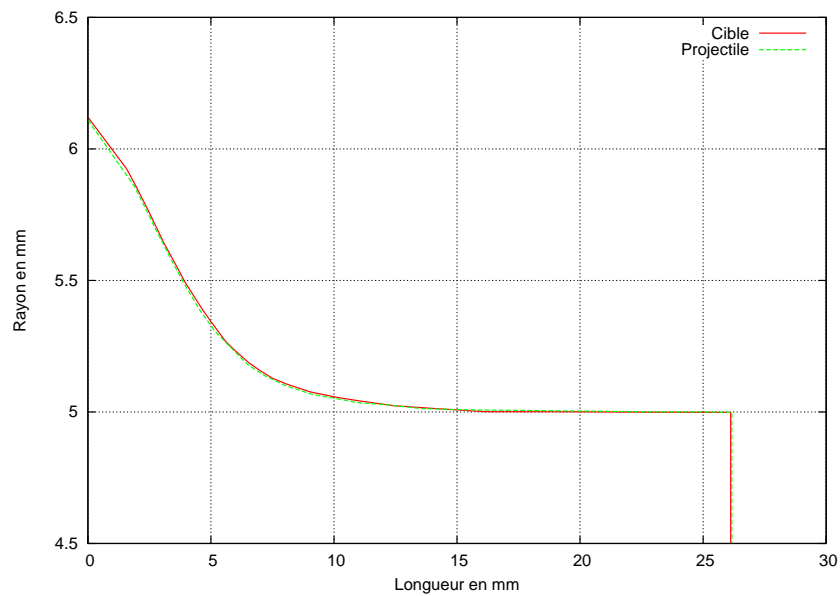


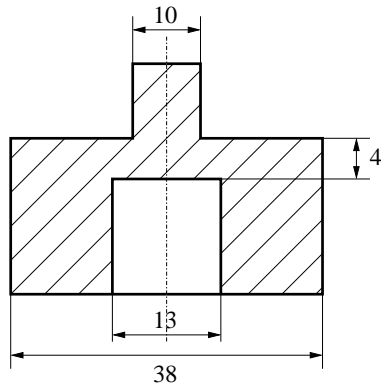
FIG. 2.7 – Profils de la cible et du projectile dans l'essai d'impact de Taylor symétrique



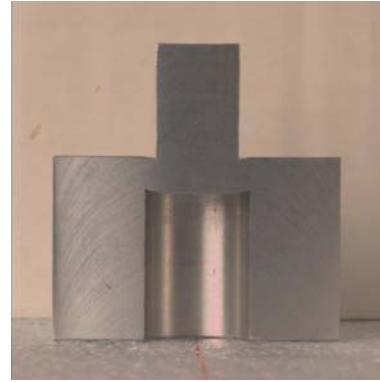
FIG. 2.8 – Essai de Taylor direct pour un projectile en aluminium

### 2.2.2.2 Essai de cisaillement dynamique

Le second type de sollicitation envisagé concerne la caractérisation du comportement en cisaillement dynamique d'un matériau. Le principe de cet essai est similaire à celui de l'essai de Taylor, mais la forme de l'échantillon (voir figure 2.9(a)) est telle que l'essentiel de la déformation plastique obtenue résulte du cisaillement dans une zone confinée de la cible. Cette forme, généralement connue sous la dénomination de "éprouvette chapeau" est issue des travaux de Meyers et al. [48]. La figure 2.9(b) montre un échantillon en aluminium 2017 déformé après un impact par un projectile constitué d'un cylindre de Taylor à une vitesse  $V_c = 111 \text{ m/s}$ .



(a) Schéma de principe



(b) Echantillon déformé

FIG. 2.9 – Essai de cisaillement dynamique

### 2.2.2.3 Essai de traction dynamique

Un essai original a été développé au sein du laboratoire afin de caractériser le comportement d'un matériau soumis à une sollicitation de traction dynamique [49, 50]. D'autres systèmes de sollicitation dynamique en traction existent dans la littérature, comme le dispositif proposé par Sturges et al. [51]. La cible, un tube creux (voir figure 2.10), est soumise à une sollicitation de traction sur sa partie utile (la zone centrale) suite à l'impact du projectile sur le fond intérieur. Après l'impact, la longueur de la zone utile, la longueur totale et le diamètre de la cible sont pris en compte afin d'identifier les paramètres de la loi de comportement. Concernant les caractéristiques dimensionnelles de la cible, l'épaisseur de la zone utile soumise à la traction est  $e = 1 \text{ mm}$ , le diamètre intérieur est  $\phi = 22 \text{ mm}$  permettant au projectile (de diamètre  $\phi = 20 \text{ mm}$ ) de venir percuter le fond de la cible en évitant des contacts latéraux parasites. Les simulations numériques réalisées au cours de cette étude montrent que la plus grande partie de l'énergie plastique dissipée (environ 80%) se situe dans la zone utile sollicitée en traction de l'éprouvette. La figure 2.10(c) montre une vue d'un échantillon totalement rompu après impact. Les dimensions et la forme optimale de la cible ont été déterminées au cours des travaux de DEA de R. Maran [vii] et E. Gorce [viii].

### 2.2.2.4 Essai d'extrusion conique

Un deuxième essai original développé au LGP concerne l'étude du frottement dynamique à grande vitesse pour l'identification de la loi de contact associée à un couple de matériaux A et B. L'essai d'extrusion conique [52, 53] développé consiste à projeter un lopin d'un matériau A dans une cible en matériau B percée d'un alésage en forme de tronc de cône prolongé par une forme cylindrique (figure 2.11). La partie en tronc de cône en tête de l'échantillon

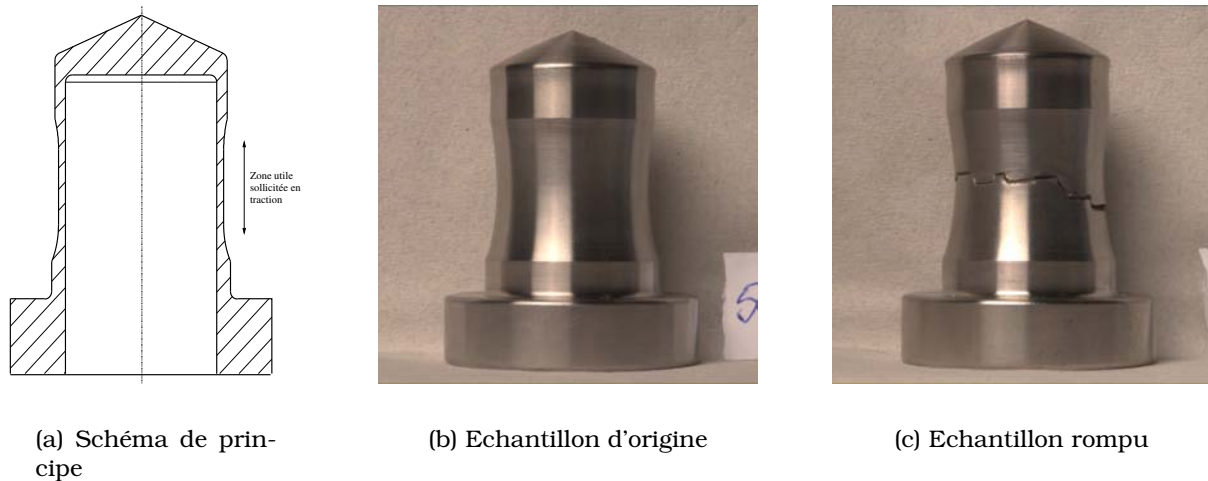


FIG. 2.10 – Essai de traction dynamique

facilite l'auto-adaptation du projectile dans la cible. La conception du système a pour objectif l'obtention d'une énergie de dissipation par frottement la plus grande possible (au moins 60% de l'énergie totale dissipée). L'ensemble est dimensionné afin que le projectile soit stoppé par frottement dans la cible au cours de l'essai. Après l'essai on mesure la déformée du lopin, en particulier la longueur d'extrusion obtenue. L'essai doit être précédé de l'identification de la loi constitutive visco-élasto-plastique des deux matériaux de manière à limiter les paramètres à identifier à ceux de la loi de frottement uniquement.



FIG. 2.11 – Essai d'extrusion conique

### 2.2.2.5 Essai de rupture dynamique

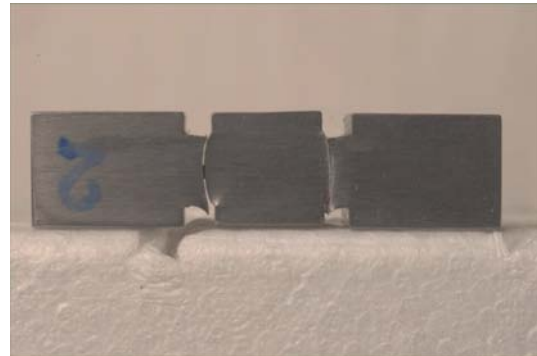
La gamme des essais de caractérisation de la loi de comportement est complétée par la caractérisation de la loi d'endommagement du matériau basée sur un essai de rupture par cisaillement dynamique (voir figure 2.12). Cet essai consiste à impacter un échantillon entaillé de forme parallélépipédique avec un projectile de type cylindre de Taylor. Cet essai est le seul de la gamme faisant apparaître une non axi-symétrie de la cible par rapport à l'axe du canon. Deux configurations peuvent être mises en oeuvre :

- un essai à déformation contrôlée, dans lequel on contrôle le déplacement de la face arrière de l'échantillon lors de l'impact comme indiqué sur la figure 2.12(a),
- et un essai à déformation libre dans laquelle on peut observer la rupture complète de l'échantillon comme indiqué sur la figure 2.12(b).

Le but de cet essai est d'étudier les conditions d'initiation de la rupture au sein du matériau ainsi que de caractériser la propagation de la fissure au sein du matériau. Cet essai est le dernier mis en oeuvre au laboratoire et est actuellement en cours de mise au point dans le cadre du travail de thèse de I. Nistor [IV].



(a) Essai à déformation contrôlée



(b) Essai sans contrôle de la déformation

FIG. 2.12 – Essai de rupture dynamique

## 2.3 Identification des lois de comportement

### 2.3.1 Procédure d'identification

Le principe de base de l'identification des lois de comportement consiste à réaliser une série de simulations numériques des essais expérimentaux mis en oeuvre. On utilise alors une procédure d'identification dans laquelle on cherche à minimiser l'écart entre les résultats numériques et expérimentaux sur un certain nombre de réponses expérimentales représentatives de la configuration de l'essai. Cette procédure d'identification est implémentée dans le module **Identif** [54], développé au cours du stage de DEA de I. Nistor [vi]. Ce module d'identification est inséré dans le logiciel **Pilote**, développé par S. Caperaa, utilisé pour "piloter" le code de calcul, c'est à dire qu'il permet de générer des fichiers paramétrés, de lancer l'application et d'extraire un ensemble de réponses numériques de la base de données du code de calcul utilisé. Ce logiciel est actuellement interfacé avec les code de calcul Abaqus et **DynELA**, mais peut bien entendu être adapté à tout autre code de calcul pour peu que celui-ci permette l'utilisation de fichiers de données paramétrés et l'extraction de résultats numériques par des routines externes.

Le domaine des applications pour lequel on peut employer **Identif** est relativement vaste. Le fonctionnement de cet outil est indépendant de la nature des charges, du type d'éléments finis utilisé, ou de la nature du schéma d'intégration. Compte tenu de la diversité des applications possibles caractérisées par des lois de comportement différentes, des paramètres de nature différentes pour la même loi, de la nature très variée des réponses numériques et expérimentales, la procédure d'identification implémentée dans ce logiciel est scindée en deux étapes :

- une recherche préliminaire par une méthode aléatoire stochastique de type Monte-Carlo ;
- un affinement de recherche, à partir des résultats fournis par la recherche préliminaire, utilisant l'algorithme de Levenberg-Marquardt.

Pour ce logiciel d'identification l'idée d'utiliser un algorithme de type Monte-Carlo est liée à la nécessité de l'exploration de l'espace topologique des paramètres dans le but de simuler



numériquement, avec les jeux de paramètres ainsi obtenus notre modèle et d'utiliser par la suite les résultats obtenus pour définir les meilleurs paramètres assurant une très bonne modélisation du comportement du matériau.

L'introduction des données de départ (les valeurs initiales pour les paramètres à identifier, les valeurs expérimentales pour la déformée de l'échantillon analysé et les paramètres du fonctionnement pour le logiciel) apparaît sur l'interface graphique du module **Identif** présenté sur la figure 2.13.

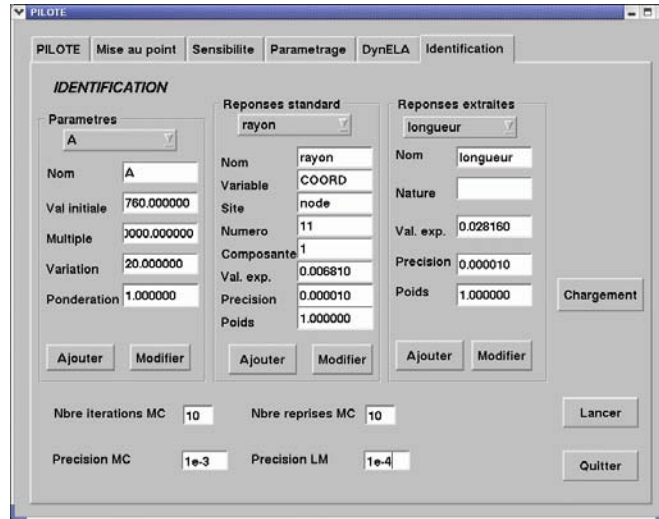


FIG. 2.13 – Interface graphique du module **Identif**

L'analyse du modèle numérique pour les jeux de paramètres fournis par les tirages au sort est possible grâce au paramétrage du modèle et au lancement du solveur numérique par le logiciel **Pilote** pour chaque jeu de paramètres.

Pour le calcul de l'écart expérience-calcul on utilise une relation basée sur une norme euclidienne :

$$f = \sqrt{\sum_{j=1}^m w_r[j] \left[ \frac{r_{EF}[j] - r_{exp}[j]}{r_{exp}[j]} \right]^2} \quad (2.26)$$

où  $m$  est le nombre de réponses,  $r_{EF}[j]$  sont les valeurs des réponses obtenues suite à la simulation,  $r_{exp}[j]$  sont les valeurs expérimentales des réponses et  $w_r[j]$  sont les poids attribués à chaque réponse respectivement.

L'exécution de l'algorithme de Monte-Carlo s'arrête lorsque l'une des conditions imposées comme critère de satisfaction (valeur de l'écart minimal ou constat de divergence de l'algorithme) est remplie et la procédure d'identification utilise alors l'algorithme de Levenberg-Marquardt. Le but de celui-ci est d'affiner la recherche afin d'obtenir les meilleurs paramètres de la loi de comportement, en utilisant le dernier jeu de paramètres trouvé par l'algorithme de Monte-Carlo comme données de départ. Cet algorithme, dérivé de la méthode de Gauss-Newton, est basé sur la résolution itérative du système d'équations suivant :

$$\left( \mathbf{J}^T \mathbf{J} + \lambda_{LM} \mathbf{I} \right) \left( \vec{q}^{(k+1)} - \vec{q}^{(k)} \right) = \mathbf{J}^T \left( r_{EF}(\vec{q}^{(k)}) - r_{exp} \right) \quad (2.27)$$

ou  $\mathbf{J}$  est la matrice Jacobienne des réponses,  $\vec{q}$  est le vecteur des paramètres à identifier et  $\lambda_{LM}$  est le paramètre de convergence d'algorithme. L'identification s'achève lorsque la valeur de l'écart obtenu est inférieure au seuil de précision défini par l'utilisateur.

### 2.3.2 Exemple d'identification

A titre d'exemple, on présente l'identification des paramètres de la loi de comportement de Johnson-Cook identifiée à partir de deux essais (impact de Taylor et traction dynamique) pour un acier 42CrMo4. Les essais étant effectués à température ambiante, l'identification ne concerne que les 4 premiers paramètres de la loi de Johnson-Cook ( $A$ ,  $B$ ,  $C$  et  $n$ ). La première identification présentée ici repose sur un essai de compression de Taylor effectué à la vitesse  $V = 328,1m/s$  pour lequel la longueur finale est  $l_f = 26,16mm$  et le diamètre de la face déformée est  $d_f = 12,22mm$ . La seconde concerne un essai de Traction dynamique à la vitesse  $V = 80m/s$ . Les coefficients de la loi de comportement de Johnson-Cook issus de cette identification sont reportés dans le tableau 2.1. Les détails de cette procédure d'identification peuvent être consultés dans la thèse de C. Sattouf [III] ainsi que Pantalé et al. [47,50] et Abichou et al. [49].

TAB. 2.1 – Paramètres identifiés de la loi de plasticité de Johnson-Cook pour le 42CrMo4

Test	A (Mpa)	B (Mpa)	n	C
Taylor	806	614	0.168	0.0089
Traction	817	699	0.157	0.0088

La figure 2.14 montre le tracé de la limite apparente d'élasticité  $\sigma^v$ , issue de l'identification par l'essai de traction et l'essai de Taylor, en fonction de la déformation plastique équivalente  $\bar{\epsilon}^p$  pour une vitesse de déformation égale à la vitesse de déformation de référence  $\dot{\epsilon}_0^p$ .

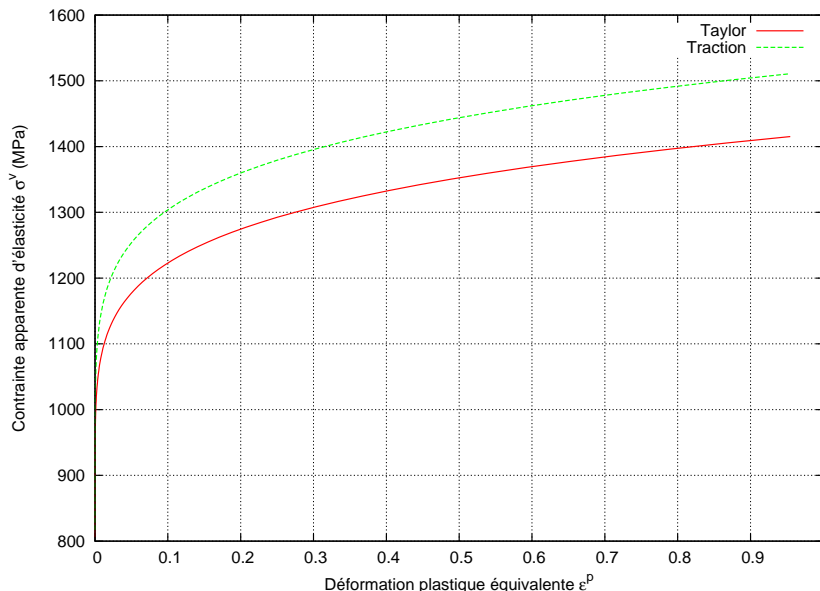


FIG. 2.14 – Lois de plasticité de Johnson-Cook pour le 42CrMo4 à  $\dot{\epsilon}^p = \dot{\epsilon}_0^p$

La figure 2.14 met en évidence la disparité des résultats obtenus en utilisant des tests d'identification différents (ici un test de Taylor et un test de traction dynamique). A ce niveau d'analyse, on peut tirer quelques conclusions préliminaires :

- La première concerne l'importance du choix d'un test d'identification reproduisant au plus près les sollicitations mécaniques rencontrées au sein de la structure en service. Cette disparité peut avoir plusieurs origines parmi lesquelles, la notion d'écroutissement isotrope et de contrainte équivalente de von Mises figurent en bonne place. En effet, et d'autant plus que l'on est ici en présence d'un comportement dynamique, la notion d'isotropie du comportement, traduite au travers de l'utilisation du critère de contrainte équivalente de von Mises semble devoir être remise en cause en Grandes Transformations.
- La seconde concerne la possibilité de mettre en oeuvre une procédure basée sur l'utilisation simultanée de plusieurs expériences (et donc de plusieurs simulations numériques) au sein de la même identification. Cette procédure a été mise en oeuvre dans le logiciel d'identification qui est maintenant en mesure de prendre en compte plusieurs essais simultanés. L'un des problèmes à résoudre alors concerne la construction de la fonction définissant l'écart expérience-calcul et le choix de la pondération à affecter à chacune des expériences.
- Une troisième voie plus exploratoire consiste à combiner les résultats issus de plusieurs campagnes d'identification dans la formulation de la loi de comportement au sein de la routine de calcul des efforts internes du code de calcul. Le logiciel utilise alors pour chaque point d'intégration la loi d'écoulement identifiée dans les conditions les plus proches de la sollicitation rencontrée au cours du calcul. Cette approche est à l'heure actuelle au stade de projet d'étude et devrait être explorée dans un futur proche dans le cadre d'un travail proposé à un étudiant en projet de M2R.

*Ces travaux ont fait l'objet de 2 publications internationales et de 5 participations à des congrès.*



# 3

## Développement du code de calcul explicite DynELA

---

*Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning.*

Rich Cook

Comme introduit en conclusion du chapitre 1, au cours de mes travaux de thèse, les réflexions menées avec R. Rakotomalala (directeur de thèse) et S. Caperaa m'ont conduit à proposer un nouveau thème de recherche axé sur le développement numérique d'un code de calcul en grandes transformations : le code **DynELA**. Les principales caractéristiques de ce code sont : un schéma d'intégration explicite et une formulation en grandes déformations, une approche Orientée-Objets pour l'implémentation numérique en C++ et une architecture de code totalement ouverte basée sur des développements propres et une utilisation de bibliothèques open-source.

Dans ce chapitre, on présente dans une première partie quelques précisions concernant l'intégration numérique explicite des lois de conservation. La deuxième partie est consacrée à une présentation de l'approche Orientée-Objets utilisée avec les conséquences sur l'architecture du code. La troisième partie est consacrée à l'implémentation et la validation numérique des performances du code au moyen de cas test. Enfin, la dernière partie est consacrée à la présentation des développements réalisés afin de paralléliser le code de calcul **DynELA**.

Dans la mesure où un grand nombre de codes de calcul existent à l'heure actuelle, aussi bien de nature commerciale que libres de droits, on est en mesure de se poser la question de la pertinence de cette entreprise relativement longue et coûteuse. Certains éléments de réponse à cette question sont reportés dans la section ci-après.

### 3.1 Pourquoi développer un nouveau code de calcul

La première source de motivation que l'on peut évoquer pour justifier ce type d'approche réside dans la possibilité de détenir au sein du laboratoire d'un code de calcul totalement ouvert et maîtrisé, qui, s'il ne dispose pas des possibilités d'un code commercial tel que Abaqus, offre néanmoins des performances relativement efficaces dans son domaine d'application. Dans le cas qui nous concerne ici, le but n'est pas de dupliquer un code commercial, mais au contraire, de recentrer les efforts de développement sur la modélisation en grandes transformations. Du côté des logiciels libres, à l'heure actuelle, très peu d'applications existent dans

le domaine des grandes transformations et encore moins dans les applications dynamiques liées à la modélisation numérique des impacts.

D'un point de vue pratique, l'utilisation d'un code de calcul développé en interne permet de maîtriser au mieux le comportement numérique de ce code. Bien souvent dans le cas de l'utilisation de codes commerciaux, l'utilisateur n'a aucune connaissance de la théorie réelle sur laquelle sont basés les algorithmes qu'il utilise. Bien souvent également, la documentation théorique et la réalité ne coïncident pas. De plus, et notamment dans le domaine des grandes transformations, les codes commerciaux utilisent des "artifices" permettant de stabiliser au mieux les algorithmes numériques afin que le code "converge" dans la majorité des cas. Ces artifices, utiles pour les utilisateurs classiques, sont généralement problématiques pour des applications dans le domaine de la recherche. Ainsi, comment comparer efficacement deux solutions numériques lorsque l'une des deux au moins est altérée par un de ces artifices? De plus, la mise à disposition d'une plateforme de développement permet de tester de manière efficace de nouveaux algorithmes et de nouvelles idées puisque l'on a accès à toutes les parties du code et non à certaines routines utilisateur généralement "ouvertes" par l'éditeur du logiciel.

La maîtrise totale du logiciel permet de plus de contrôler efficacement les évolutions. En effet, bien souvent, lors de l'utilisation de logiciels commerciaux, les nouvelles fonctionnalités ajoutées lors d'un changement de version s'accompagnent généralement de modifications dans le formalisme, les formats de fichiers de sortie, etc... Si ces modifications sont généralement non perceptibles pour les utilisateurs classiques, elles peuvent se révéler rédhibitoires dans le cas de l'intégration de ce code dans des procédures externes (identification, liaison entre plusieurs logiciels, etc ...) car on doit bien souvent modifier les interfaces avec le code. Dans d'autres cas, les modules externes comme l'implémentation de lois de comportement via des routines UMAT sur Abaqus, doivent être également révisés et modifiés lors des changements de version.

A la liste de ces différents points, il convient d'ajouter que le développement d'un code de calcul permet un enrichissement intellectuel personnel très important. Les développements numériques permettent d'approfondir les connaissances dans le domaine de la mécanique en grandes transformations puisque l'on est obligé de maîtriser la théorie pour pouvoir l'implémenter numériquement. C'est une phase de consolidation des connaissances directement par l'expérience, certes plus délicate, mais bien plus approfondie.

Bien entendu, cette liste n'est nullement exhaustive, et les personnes qui de près ou de loin ont entrepris cette approche peuvent présenter des arguments supplémentaires selon leurs sensibilités.

### 3.2 Intégration numérique explicite

L'équation majeure à discrétiser est l'équation de conservation de la quantité de mouvement qui est exprimée en termes de coordonnées spatiales (Eulériennes) et en fonction des contraintes de Cauchy (contraintes physiques). La discrétisation spatiale, basée sur le Principe des Puissances Virtuelles servant de base à une formulation éléments finis des équations d'équilibre, conduit à l'écriture suivante pour l'équation d'équilibre [35] :

$$\mathbf{M} \ddot{\vec{x}} + \vec{F}^{int}(\vec{x}, \dot{\vec{x}}) - \vec{F}^{ext}(\vec{x}, \dot{\vec{x}}) = 0 \quad (3.1)$$

dans laquelle  $\dot{\vec{x}}$  est le vecteur des vitesses nodales,  $\ddot{\vec{x}}$  le vecteur des accélérations nodales,  $\mathbf{M}$  est la matrice de masse,  $\vec{F}^{ext}$  est le vecteur des efforts externes appliqués à la structure et  $\vec{F}^{int}$  le vecteur des forces internes. Cette équation est complétée par les conditions initiales

définies ci-dessous à l'instant  $t_0$  :

$$\vec{x}_0 = \vec{x}(t_0); \quad \dot{\vec{x}}_0 = \dot{\vec{x}}(t_0) \quad (3.2)$$

En utilisant les mêmes fonctions de forme  $\mathbf{N}$  pour les fonctions d'interpolation et les fonctions test on obtient l'expression suivante pour les matrices élémentaires entrant dans l'écriture de l'équation d'équilibre (3.1) :

$$\begin{aligned} \mathbf{M} &= \int_{\Omega_x} \rho \mathbf{N}^T \mathbf{N} d\Omega_x \\ \vec{F}^{int} &= \int_{\Omega_x} \mathbf{B}^T \boldsymbol{\sigma} d\Omega_x \\ \vec{F}^{ext} &= \int_{\Omega_x} \rho \mathbf{N}^T \vec{b} d\Omega_x + \int_{\Gamma_x} \mathbf{N}^T \vec{t} d\Gamma_x \end{aligned} \quad (3.3)$$

dans laquelle  $\mathbf{B} = \partial \mathbf{N} / \partial \vec{x}$  est la matrice des dérivées des fonctions d'interpolation,  $\Gamma_x$  est la surface du domaine  $\Omega_x$  sur laquelle les efforts externes sont appliqués,  $\rho$  est la masse volumique,  $\boldsymbol{\sigma}$  est le tenseur des contraintes de Cauchy,  $\vec{b}$  est le vecteur des efforts volumiques et  $\vec{t}$  est le vecteur des efforts surfaciques.

L'intégration explicite dans le code de calcul **DynELA** fait appel au schéma d'intégration de Chung-Hulbert [55] dont la principale caractéristique réside dans la présence d'une dissipation numérique. L'intégration temporelle est effectuée à partir des relations suivantes :

$$\ddot{\vec{x}}_{n+1} = \frac{\mathbf{M}^{-1} (\vec{F}^{ext}_n - \vec{F}^{int}_n) - \alpha_M \ddot{\vec{x}}_n}{1 - \alpha_M} \quad (3.4)$$

$$\dot{\vec{x}}_{n+1} = \dot{\vec{x}}_n + \Delta t_{n+1} \left[ (1 - \gamma) \ddot{\vec{x}}_n + \gamma \ddot{\vec{x}}_{n+1} \right] \quad (3.5)$$

$$\vec{x}_{n+1} = \vec{x}_n + \Delta t_{n+1} \dot{\vec{x}}_n + \Delta t_{n+1}^2 \left[ \left( \frac{1}{2} - \beta \right) \ddot{\vec{x}}_n + \beta \ddot{\vec{x}}_{n+1} \right] \quad (3.6)$$

Les valeurs particulières des paramètres de pondération qui conduisent à une dissipation numérique optimale sont données en fonction du rayon spectral  $\rho_b$  à la fréquence de bifurcation. Un rayon spectral égal à 1 conduit à un algorithme conservatif alors qu'un rayon spectral inférieur à 1 conduit à un algorithme dissipatif. Dans tous les cas, on doit respecter la condition :  $\rho_b \in [0.0 : 1.0]$ . Les paramètres  $\alpha_M$ ,  $\beta$  et  $\gamma$  sont définis à partir de la valeur de  $\rho_b$  par les relations suivantes :

$$\alpha_M = \frac{2\rho_b - 1}{1 + \rho_b}; \quad \beta = \frac{5 - 3\rho_b}{(1 + \rho_b)^2 (2 - \rho_b)}; \quad \gamma = \frac{3}{2} - \alpha_M \quad (3.7)$$

L'utilisation d'un schéma d'intégration numérique explicite nécessite le calcul de la valeur du time-step critique de la structure puisqu'il est conditionnellement stable. La taille du pas de temps est limitée par cette valeur critique qui doit être évaluée en fonction du modèle numérique. L'incrément de temps  $\Delta t$  est alors défini par la relation suivante :

$$\Delta t = \gamma_s \Delta t_{crit} = \gamma_s \frac{\Omega_s}{\omega_{max}} \quad (3.8)$$

relation dans laquelle  $\Delta t_{crit}$  est une expression dépendant de la fréquence propre maximale  $\omega_{max}$  de la structure et  $\gamma_s$  est un coefficient de sécurité destiné à prendre en compte le fait que la fréquence propre maximale est évaluée à partir d'une formulation linéaire. La valeur de la fréquence propre maximale  $\omega_{max}$  de la structure est évaluée à partir de la méthode *power iteration* décrite par D. J. Benson [56]. Le paramètre  $\Omega_s$  est défini à partir de la valeur de  $\rho_b$  par la relation suivante :

$$\Omega_s = \sqrt{\frac{12(1 + \rho_b)^3 (2 - \rho_b)}{10 + 15\rho_b - \rho_b^2 + \rho_b^3 - \rho_b^4}} \quad (3.9)$$

L'algorithme général d'intégration explicite est donné dans la boîte 3.2. Le détail des différentes étapes de cet algorithme est donné dans le manuel de **DynELA** [36].

---

 Algorithme 3.2 – Algorithme général d'intégration explicite

1. Calcul des matrices internes,  $\mathbf{N}$ ,  $\mathbf{B}$ ,  $\mathbf{J}$ ,  $\det[\mathbf{J}]$
  2. Calcul de la matrice de masse globale  $\mathbf{M}$
  3. Calcul des efforts internes  $\overrightarrow{F^{int}}$
  4. Calcul du time-step de la structure
  5. Initialisation du step de calcul
  6. Prédiction des quantités
  7. Initialisation de l'itération
  8. Résolution explicite
  9. Fin du step
  10. Test de fin de calcul, si oui aller en 13
  11. Calcul des matrices internes,  $\mathbf{N}$ ,  $\mathbf{B}$ ,  $\mathbf{J}$ ,  $\det[\mathbf{J}]$
  12. Retour en 4
  13. Sauvegarde des résultats
- 

### 3.3 Implémentation numérique Orientée-Objets du code de calcul

Traditionnellement, les codes numériques sont programmés en utilisant des langages de programmation procéduraux tels que le Fortran ou le C. Dans l'esprit des langages procéduraux, les ajouts de nouvelles fonctionnalités à des programmes déjà très longs demandent de la part des programmeurs un travail dont la complexité augmente exponentiellement avec la taille initiale du code. Cette complexité conduit bien souvent à la spécialisation de ces programmes conduisant à de nombreux exécutables différant les uns des autres par quelques algorithmes.

Actuellement, les méthodes de conception et de Programmation Orientée-Objets (POO) sont les mieux adaptées aux développements de grande envergure. L'abstraction que permet la POO permet au développeur de mieux organiser l'architecture du programme et d'anticiper les développements futurs. Des exemples spécifiques au développement de codes de calcul ont été décrits par Miller [57] et Mackie [58]. D'un point de vue langage de programmation, si la préférence générale actuelle est à l'utilisation du C++, d'autres langages peuvent être utilisés pour le développement d'applications numériques. Ainsi, Nikishkov [59] présente une analyse détaillée des performances d'un code de calcul implicite utilisant une formulation élastique programmé en langage Java<sup>(1)</sup>.

Dans cette section, les principaux avantages issus de l'utilisation d'une approche Orientée-Objets sont présentés dans un premier temps. Dans une deuxième partie, on décrit l'architecture générale adoptée pour le développement du code de calcul **DynELA**.

#### 3.3.1 Approche Orientée-Objets et avantages pour un code éléments finis

Ces dernières années, l'utilisation de la Programmation Orientée-Objets a pris une place prépondérante, conduisant à des codes hautement modularisés par la définition de classes qui peuvent être vues comme l'association de données et de méthodes effectuant des opérations sur ces données. Cette approche est très attractive du fait de la ré-utilisabilité de code et de

<sup>(1)</sup>Il ressort de cette étude que les performances du code en Java sont équivalentes à celles du même code en C concernant le calcul des matrices de rigidité et l'assemblage du système global.



la modularité qu'elle procure. Dans une approche POO, l'association d'un ensemble de données (attributs) et des méthodes qui utilisent ces données conduit à la définition d'une classe. Les instances d'une classe sont des objets. Les classes peuvent représenter des notions très diverses comme par exemple des matrices, des vecteurs, des éléments ou des algorithmes d'intégration numérique. En résumant, on peut dire que la POO permet l'implémentation de concepts mathématiques comme par exemple les équations différentielles dans la librairie de développement Diffpack [60]. L'efficacité numérique dans l'approche POO résulte de l'association de routines numériques traditionnelles encapsulées dans un formalisme de haut niveau dont les principales caractéristiques sont l'encapsulation, l'héritage, l'abstraction, la surdéfinition des membres et opérateurs et les classes génériques [61]. Toutes ces notions classiques pour les programmeurs sont synthétisées ci-après avec leurs implications directes dans le cadre du développement d'un code de calcul par éléments-finis.

- L'encapsulation est un mécanisme qui permet de restreindre l'accès aux membres d'une classe donnée aux seules méthodes de cette même classe. On évite les interférences extérieures et les organisations globales de données qui rendent l'évolutivité des programmes très difficile comme on le voit classiquement en Fortran avec l'utilisation abusive des *COMMON*.
- L'héritage est un mécanisme qui permet d'exploiter les caractéristiques communes à plusieurs objets. Ainsi, dans le cas d'une formulation par éléments finis, les divers éléments tels que les éléments de solide 2D, 3D, poutres ou plaques diffèrent entre autres, les uns des autres, par leurs fonctions d'interpolation, mais peuvent partager des méthodes communes comme le calcul des forces internes ou la gestion des points d'intégration. La figure 3.1 présente une version simplifiée du diagramme UML<sup>(2)</sup> [62] définissant la classe **Element** dans **DynELA**. Dans cet exemple, le calcul des forces internes via la méthode `computeInternalForces()` est implémenté dans la classe **Element** car il est commun à tous les types d'éléments alors que l'évaluation des fonctions d'interpolation, via la méthode `getShapeFunction()`, qui dépend du type exact d'élément est implémentée dans les classes dérivées.
- L'abstraction permet de définir des objets abstraits comportant des méthodes virtuelles. Les classes abstraites permettent l'écriture de codes génériques et une extension simple d'un code existant. Un exemple d'application est la classe **Element** définie sur la figure 3.1. Dans ce cas, on ne crée jamais d'instance de la classe **Element**, mais uniquement de l'une des classes dérivées selon le type d'élément physique désiré.
- La surcharge des membres et opérateurs autorise une écriture simple des équations mathématiques comme par exemple le produit de matrices en autorisant une syntaxe générique de la forme  $A = B * C$  dans laquelle  $A$ ,  $B$  et  $C$  sont trois matrices de tailles compatibles entre elles. Les deux opérateurs surchargés  $*$  et  $=$  utilisent des algorithmes efficaces dans le cadre du calcul matriciel incluant, si besoin, des vérifications de compatibilité des opérands. Le même type d'opération est possible lorsque les paramètres sont des instances de différentes classes comme par exemple dans le cas du produit d'une matrice et d'un vecteur.
- Les classes génériques permettent par exemple de créer des listes d'objets de tout type (noeuds, éléments, points d'intégration...). Les classes génériques sont un des piliers des technologies permettant de créer des codes abstraits et des codes hautement performants en C++ [61]. Dans le cadre du développement de **DynELA**, la classe générique la plus importante est la classe **List** destinée à gérer tous les objets en mémoire sous forme de listes dynamiques. Elle sert notamment à gérer la liste des éléments d'une structure, la liste des

<sup>(2)</sup>UML : *Unified Modeling Language* est le concept de modélisation objet apparu en 1994 et normalisé par l'OMG début 1997. Elle simplifie le processus complexe de conception d'un système informatique en définissant 9 points de vue (ou diagrammes) de modélisation.

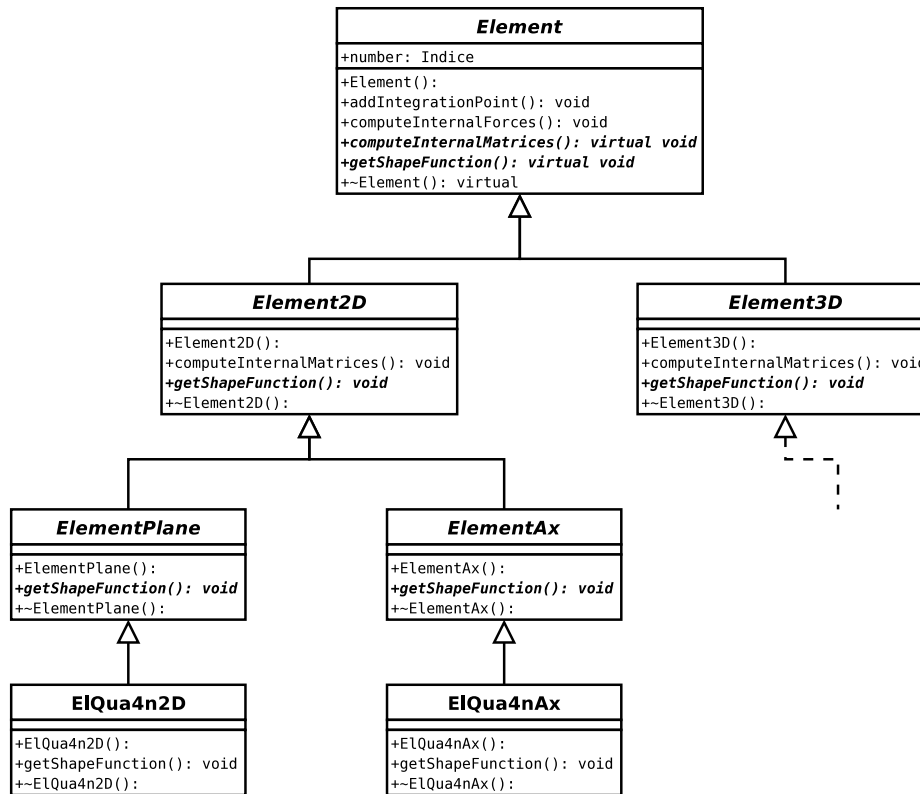


FIG. 3.1 – Diagramme UML simplifié de la classe Element

noeuds d'un élément, etc...

L'utilisation de la POO, et ici plus précisément du C++ a été critiquée par le passé à cause de son efficacité numérique qui était jugée moins bonne que celle des codes procéduraux tels que ceux écrits en Fortran. Des études récentes [61, 63] ont montré que l'on pouvait obtenir des performances comparables à celles des codes procéduraux à condition d'implémenter les parties calcul dans des bibliothèques mathématiques encapsulées en C++. Ceci afin de tirer partie des optimisations disponibles au sein des compilateurs C et Fortran classiques. Ainsi, pour les classes matricielles développées dans cette application, on utilise des routines de calcul de bas niveau en C et Fortran issues des bibliothèques Lapack [64] et Blas [65] encapsulées dans du code C++.

### 3.3.2 Description de l'architecture du code DynELA

#### 3.3.2.1 Architecture de l'application

Dans un processus de développement d'un code de calcul par éléments finis, la phase préliminaire consiste à définir un ensemble de bibliothèques de base pour la gestion des données (sous forme de listes, piles, ...) ainsi que les diverses bibliothèques mathématiques adaptées aux cas de modélisation rencontrés. Le choix retenu dans ce travail à été de développer entièrement l'ensemble des bibliothèques de base et d'encapsuler une partie de la bibliothèque mathématique Lapack [64] afin de développer de nouvelles classes matricielles et tensorielles car aucune bibliothèque mathématique classique ne propose par exemple la notion de tenseur d'ordre 4. Le code de calcul **DynELA** est composé d'un ensemble de bibliothèques et de fichiers exécutables distincts qui ont chacun des tâches particulières. La liste simplifiée de ces bibliothèques est donnée ci-dessous :

- **basicTools** : bibliothèque de base qui contient les classes de base de l'application **DynELA**.

- **linearAlgebra** : librairie de calcul algébrique. Elle définit notamment les notions de vecteurs, matrices, tenseurs et fonctions mathématiques.
- **interpretor** : librairie qui définit le langage de commande interprété de **DynELA**. C'est un des points forts du logiciel comme on le verra par la suite.
- **femLibrary** : librairie de calcul éléments-finis (c'est le coeur du solveur éléments finis).

A partir de ces diverses bibliothèques, on crée un ensemble de programmes exécutables qui correspondent aux divers modules du code de calcul : le solveur éléments finis, le post-processeur graphique, le programme d'analyse de courbes, le générateur de langage, etc...

### 3.3.2.2 Description de la librairie éléments finis

D'une manière simplifiée, le modèle éléments finis représenté par la classe **Structure** est principalement composé de l'association des classes **Node**, **Element**, **Material** et **Interface** comme indiqué sur la figure 3.2.

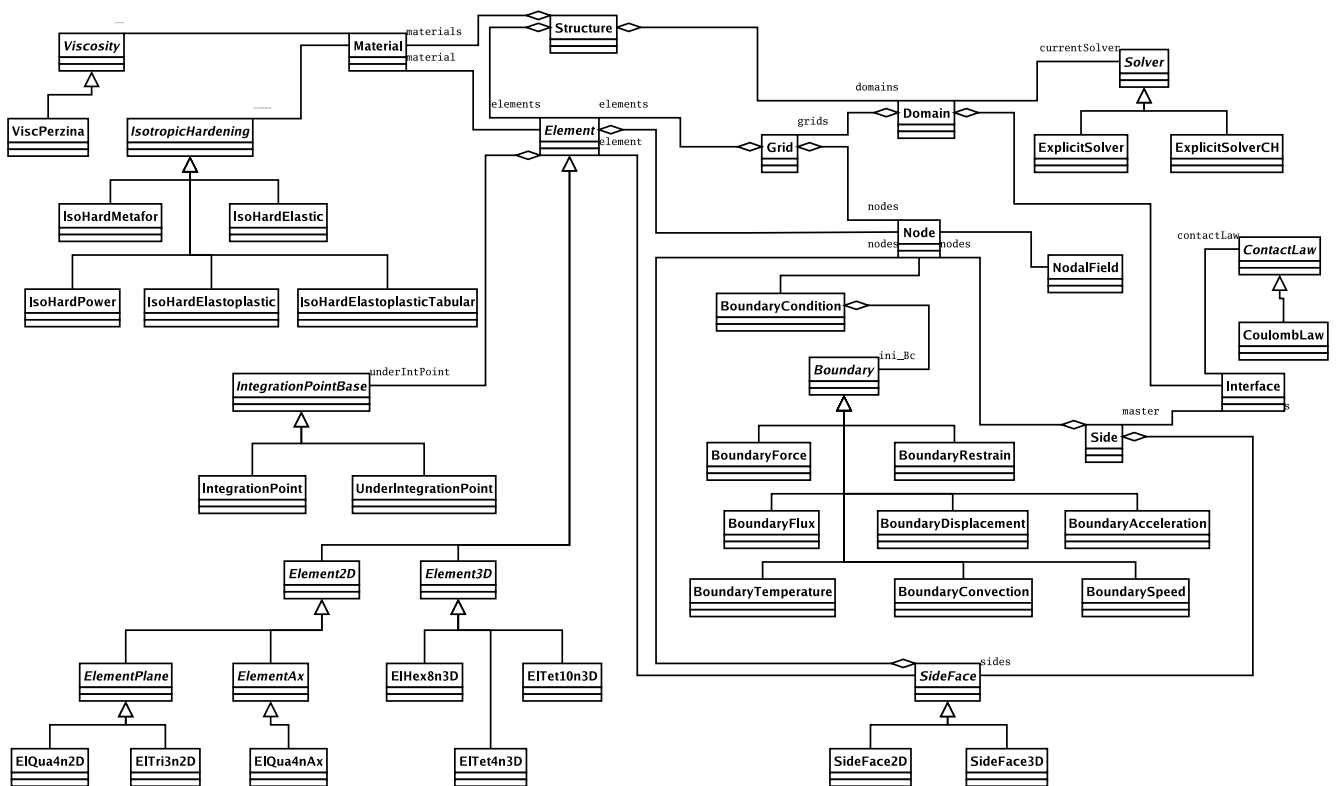


FIG. 3.2 – Diagramme UML simplifié de la librairie femLibrary

- La classe **Node** contient les données nodales comme le numéro du noeud ou les coordonnées spatiales. Deux instances de la classe **NodalField** sont rattachées à chaque noeud de la structure, la première contient les quantités nodales à l'instant  $t$ , la seconde celles à l'instant  $t + \Delta t$ . A la fin d'un incrément de temps, on échange les références de ces instances afin de transférer les données d'un incrément à l'autre.
- Les conditions aux limites sont portées par la classe **BoundaryCondition** qui modifie le comportement d'un noeud en fonction de la nature de ces conditions. Ces conditions limites sont attachées dynamiquement aux noeuds et peuvent donc varier au cours d'un calcul. Ceci est important par exemple pour les noeuds impliqués dans un contact.
- La classe **Element** est une classe purement virtuelle contenant la définition de chaque élément de la structure. Plusieurs classes dérivent de cette classe de base selon la nature réelle de l'élément désiré.

- La classe **Interface** contient la définition des interfaces de contact, la loi de contact est portée par la classe **ContactLaw** et la définition du contact par la classe **Side**.
- La classe **Material** est utilisée pour la définition des matériaux et lois d'écoulement.
- La classe **Solver** sert de classe de base pour les différents types d'algorithmes d'intégration numérique.
- Plusieurs autres classes utilitaires ont également été développées pour gérer par exemple les *time-history*, les groupes de noeuds et éléments, ou les fichiers de données et de résultats.

On se référera au manuel utilisateur de **DynELA** [36] pour de plus amples détails concernant l'architecture du code de calcul.

### 3.3.3 Spécificités du code de calcul DynELA

En termes de taille de développement, même si cette notion n'a rien de très scientifique puisque l'on peut artificiellement augmenter la taille d'un code (en dupliquant des morceaux de source inutiles par exemple), on peut globalement estimer le nombre de lignes de code C++ des différents modules comme suit :

- Solveur éléments finis : 80.000 lignes de code C++
- Interpréteur de commande : 10.000 lignes de code C++, Lex et Yacc<sup>(3)</sup>
- Post-processeur graphique : 20.000 lignes de code C++
- Utilitaires : 45.000 lignes de code C++

La formulation retenue actuellement est Lagrangienne, mais les dispositions ont été prises afin de prendre en compte une formulation Arbitraire Lagrangienne Eulérienne (ALE) : notion de vitesse de grille, algorithmes de vitesse de grille multiples, etc... L'intégration numérique est explicite et actuellement, deux types de schémas d'intégration sont utilisables : une intégration explicite classique et le schéma d'intégration de Chung-Hulbert décrit au paragraphe 3.2. Le code de calcul dans sa version actuelle est uniquement orienté vers le calcul de solides. Les éléments finis développés sont des éléments de solide, les déclinaisons sont les suivantes :

- Éléments triangulaires 2D et axi-symétriques à 3 et 6 noeuds.
- Éléments quadrilatères 2D et axi-symétriques à 4 noeuds.
- Éléments tétraédriques à 4 et 10 noeuds.
- Éléments hexahédriques à 8 et 20 noeuds.

#### 3.3.3.1 Langage de commande

L'un des points majeurs dans le développement d'un code de calcul réside dans la manière dont l'utilisateur spécifie les données d'une modélisation. Plusieurs alternatives sont possibles. Certains logiciels privilégient l'utilisation d'une interface graphique permettant de construire pas à pas le modèle numérique, d'autres utilisent un fichier de commandes que l'utilisateur édite de manière externe. Au cours des phases préliminaires du développement de **DynELA**, la réponse apportée à cette question a été de pouvoir utiliser les deux approches simultanément. Dans l'approche retenue, l'utilisation de l'interface graphique permet de créer de manière conviviale les groupes de noeuds et éléments, alors que le langage de commande permet d'effectuer des opérations plus détaillées.

La tendance actuelle est au développement de langages de commande pour le pilotage des codes de calcul afin de remplacer les fichiers de commande hérités de l'époque des cartes perforées que l'on trouve encore dans beaucoup de codes de calcul. C'est par exemple l'approche proposée actuellement par le code de calcul Abaqus pour lequel un nouvel interpréteur de

<sup>(3)</sup>*Yet Another Compiler-Compiler* (Encore un autre compilateur de compilateur). Lex et Yacc sont des intégrateurs de programmes UNIX et accessoirement sous MS-DOS. Ils permettent de créer des modules de reconnaissance de motifs et des analyseurs syntaxiques.

commande basé sur le langage Python<sup>(4)</sup> [66, 67] est développé. Concernant le code de calcul **DynELA**, après plusieurs essais (notamment en utilisant Python), le choix s'est porté sur le développement d'un *parser* lexical et grammatical spécifique développé en utilisant les outils standards Lex et Yacc [68]. Le fichier d'entrée est unique, il comporte la définition géométrique de la structure (définition des domaines, des noeuds et éléments...), les conditions aux limites, les paramètres du calcul (temps de simulation, nombre d'incrémentes, fichiers de sauvegarde...). D'une manière générale, ce fichier de commande possède une syntaxe proche du langage C++ et permet la manipulation de données orientées-objet, l'écriture de tests et de boucles. Le formalisme du langage retenu permet de répondre de la manière la plus efficace aux besoins énoncés ci-dessous :

- Formalisme Orienté-Objet du langage de commande incluant les notions de classes, d'héritage, d'opérateurs multiples afin de profiter de l'approche Orientée-Objets du code au niveau de la syntaxe.
- Intégration de nouvelles classes telles que les sous-domaines, les solveurs multiples...
- Evolutivité conviviale du langage à travers une interface graphique et un script de définition ASCII simple.
- Possibilité d'accéder à toutes les classes de **DynELA** de manière interactive au travers du langage de commande, à la manière du langage Python.
- Possibilité de développer des modules à partir du langage de commande, puis de compiler ce fichier source afin d'étendre les bibliothèques de **DynELA** après une phase de test.
- Mise à jour automatique de la documentation utilisateur en fonction des évolutions du langage en lien direct avec le code source.

Afin de faciliter le développement et l'extension du langage de commande, notamment par des personnes implémentant de nouvelles fonctionnalités dans le code et ne possédant pas de connaissances en Lex et Yacc, un outil graphique de spécification a été spécialement développé. Son but est de générer automatiquement les fichiers de définition lexicaux et grammaticaux du langage de commande ainsi que la mise à jour automatique de la documentation utilisateur.

### 3.3.3.2 Post-processeur graphique

L'exploitation des résultats numériques est effectué au moyen d'un post-processeur graphique spécialement développé pour le code de calcul **DynELA**. Une vue de la fenêtre graphique de ce post-processeur est donnée sur la figure 3.3. La partie graphique 3D utilise le formalisme OpenGL<sup>(5)</sup>, et la construction de l'interface est réalisée en utilisant la librairie graphique Qt<sup>(6)</sup>.

Historiquement, certaines fonctionnalités de ce post-processeur graphique ont été développées durant mes travaux de thèse [1] puisque le code de calcul utilisé à l'époque (Radioss) ne disposait pas de module de pré et post-processeur graphique. Certaines parties de l'ancien post-processeur ont donc été ré-utilisées lors du développement de cette application.

### 3.3.3.3 Utilitaires et environnement

Un ensemble d'utilitaires a été développé afin de faciliter au maximum l'utilisation du code de calcul. L'idée force est de permettre un maximum de confort lors de l'utilisation du code de calcul tout en permettant, aux utilisateurs désireux d'utiliser efficacement le code, un accès direct au travers du langage de commande. Parmi les divers utilitaires :

<sup>(4)</sup>Python est un langage interprété de haut niveau Orienté-Objets. Il s'interface aisément avec des composantes C et bénéficie d'une imposante collection de bibliothèques et de modules d'extension.

<sup>(5)</sup>*Open Graphic Library* est une bibliothèque de fonctions graphiques permettant de réaliser des rendus volumiques en 3D. (<http://www.opengl.org>)

<sup>(6)</sup>Qt édité par la société Trolltech est une bibliothèque de programmation C++ qui permet de créer des interfaces graphiques. (<http://www.trolltech.com>)

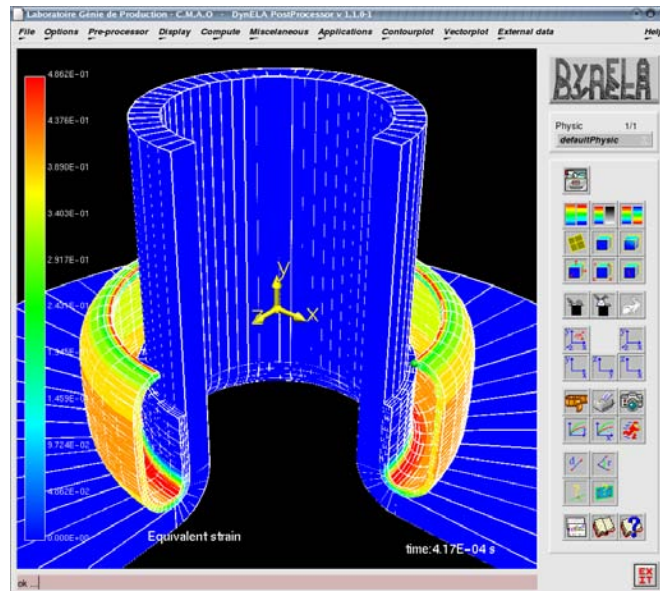


FIG. 3.3 – Post-processeur graphique de **DynELA**

- Interface graphique de pilotage permettant de contrôler visuellement l'exécution et la progression d'un calcul.
- Extraction de résultats au moyen de scripts basés sur le langage de commande de **DynELA**. Cette facilité permet de créer des boucles de calcul et d'intégrer le code de calcul dans une procédure d'optimisation afin de faire de l'identification paramétrique comme indiqué au paragraphe 2.3.
- Logiciel de traçage et de manipulation de courbes permettant d'exploiter notamment les résultats numériques temporels.
- Interfaces de communication et d'échange de données avec les logiciels Abaqus (en importation/exportation) et Ideas (en importation). Ces utilitaires permettent de réaliser les maillages sur Ideas ou Abaqus CAE puisque **DynELA** est actuellement dépourvu de module de maillage volumique évolué. Des travaux sont à l'étude actuellement pour développer un pré-processeur volumique et un mailleur en utilisant la librairie OpenCascade<sup>(7)</sup>.

### 3.4 Validation numérique du code de calcul

Afin de valider les développements réalisés, un ensemble de cas tests a été retenu. Parmi ceux-ci, on présente dans cette section trois exemples de validation liés à la simulation numérique en grandes déformations. L'ensemble des cas tests de validation réalisés est répertorié dans le manuel de **DynELA** [36].

#### 3.4.1 Striction d'un barreau cylindrique

Ce premier cas test est destiné à valider l'implémentation des algorithmes d'intégration numérique de la loi de comportement. Le test de striction d'un barreau cylindrique a été traité par de nombreux auteurs [34, 39, 69, 70]. Le problème retenu pour ce cas test a été initialement proposé avec des résultats expérimentaux par Norris et al. [39]. Par la suite, Simo [34] a identifié les paramètres de la loi de comportement. Cet exemple axi-symétrique est celui de l'éprouvette de traction représentée sur la figure 3.4 où on remarque que la géométrie de la

<sup>(7)</sup>Ensemble de bibliothèques graphiques permettant de développer de manière efficace une application de CAO (<http://www.opencascade.org/>).

barre est telle que le diamètre au milieu de l'éprouvette est légèrement plus petit que celui des extrémités. Ceci permet de localiser la déformation maximale dans la zone centrale de l'éprouvette. Le chargement de la structure est réalisé en imposant un déplacement horizontal relatif en traction de  $14\text{ mm}$  aux extrémités de l'éprouvette tandis que le déplacement vertical de ces mêmes points est nul.

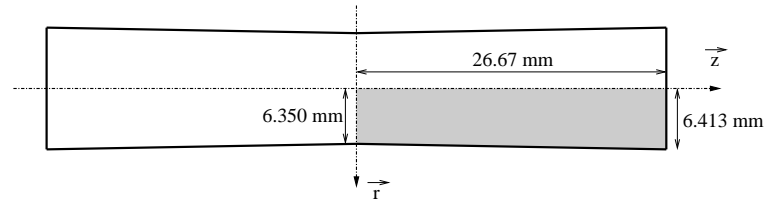


FIG. 3.4 – Géométrie initiale de l'éprouvette pour l'essai de traction

La loi d'écoulement proposée par Simo [34] (voir équation (2.21) page 32 pour la forme mathématique de la loi) correspond à un acier spécial utilisé pour la construction de tuyauteries dans l'industrie nucléaire (acier A-533 Grade B classe 1) :

$$\sigma^v = \sigma_0^v + (\sigma_\infty^v - \sigma_0^v)(1 - \exp(-\delta \bar{\epsilon}^p)) + \xi \bar{\epsilon}^p \quad (3.10)$$

C'est une loi élastoplastique non-linéaire dont les paramètres matériau identifiés par Simo sont répertoriés dans le tableau 3.1 et le tracé sur la figure 3.5.

TAB. 3.1 – Paramètres matériau pour l'essai de striction

$E$	206.9 GPa	$\sigma_\infty^v$	715 MPa
$\nu$	0.29	$\delta$	16.93
$\sigma_0^v$	450 MPa	$\xi$	129.24 MPa

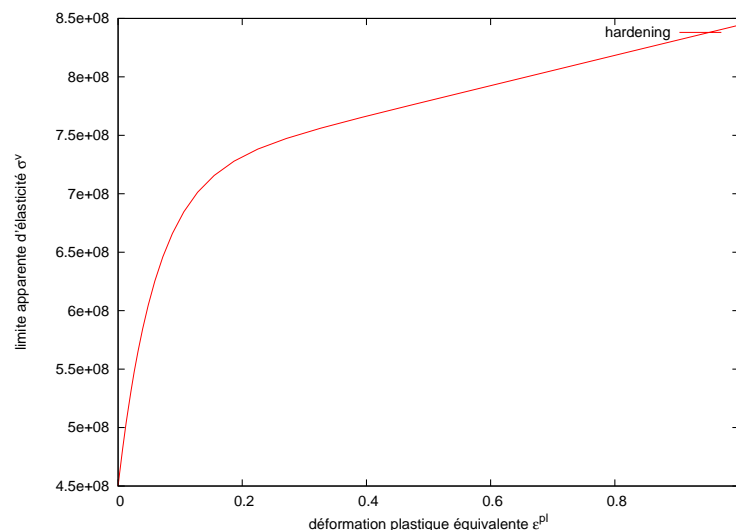
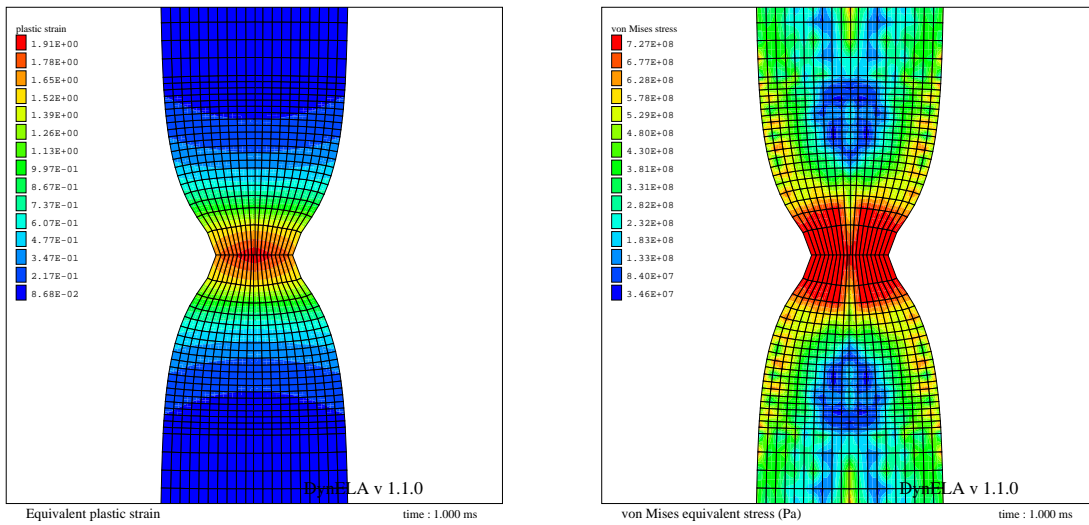


FIG. 3.5 – Loi d'écoulement pour l'essai de striction

La partie gauche de la figure 3.6 présente à la fin de la mise en charge la valeur de la déformation plastique équivalente dans la structure, et la partie droite de cette même figure les contraintes équivalentes de von Mises. La figure 3.7 présente la variation du diamètre de la zone de striction  $\frac{r}{r_0}$  en fonction de l'élongation  $\frac{\Delta L}{L_0}$  et la comparaison avec les résultats expérimentaux proposés par Norris et al. [39] et les résultats numériques proposés par Simo [34].

Les solutions obtenues par le code de calcul **DynELA** permettent de valider l'implémentation de la loi de comportement.



(a) déformation plastique équivalente

(b) contraintes de von Mises

FIG. 3.6 – Essai de striction d'un barreau cylindrique sur **DynELA**

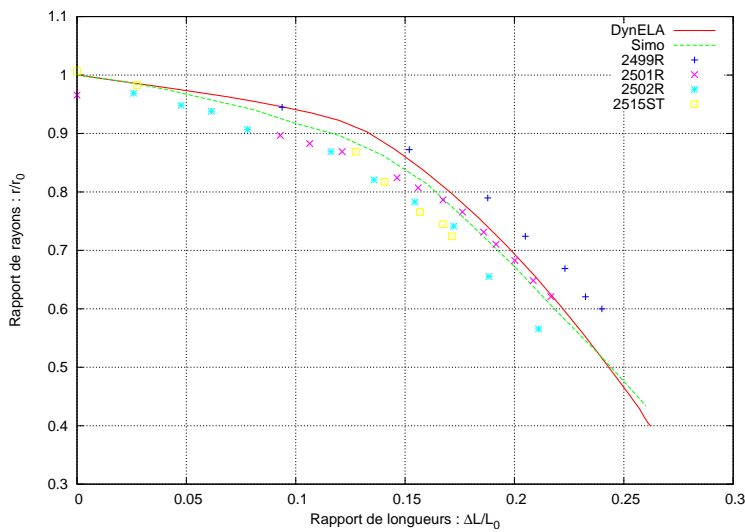


FIG. 3.7 – Variation du diamètre de la zone de striction en fonction de l'élongation pour l'essai de striction

tel-00009913, version 1 - 8 Aug 2005



### 3.4.2 Test d'impact de Taylor

Le second cas test proposé ici est relatif au comportement dynamique. Le cas test d'impact de Taylor [45, 46] est par excellence le cas test classique permettant de valider le bon comportement d'un code de calcul par éléments finis dynamique en grandes transformations. Cet exemple a été largement abordé dans la littérature, et nombre de résultats numériques [37, 71] et expérimentaux sont disponibles. Le principe de ce test est de projeter un cylindre métallique sur une surface rigide indéformable. La loi constitutive est une loi élastoplastique à écrouissage linéaire isotrope. Le contact entre le cylindre et la cible est sans frottement. Les dimensions initiales et les paramètres matériau correspondant à un cuivre OHFC sont reportés dans le tableau 3.2.

TAB. 3.2 – Paramètres matériau pour l'essai de Taylor

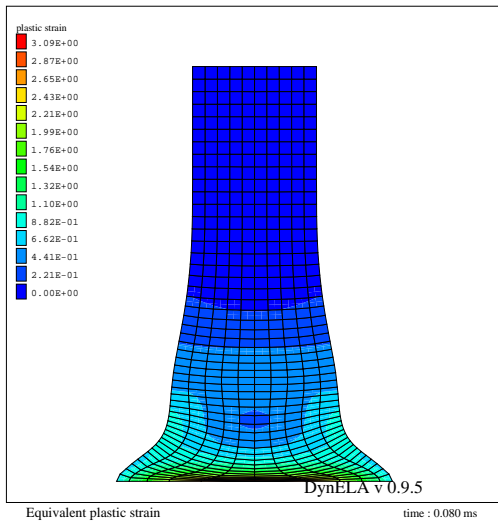
rayon	3.2 mm	$E$	117 GPa	$\sigma_0^v$	400 MPa
longueur	32.4 mm	$\nu$	0.35	$h$	100 MPa
		$\rho$	8930 kg/m <sup>3</sup>		

La vitesse initiale est  $V_c = 227 \text{ m/s}$ . Le maillage utilisé dans les travaux référencés comporte généralement 250 éléments ( $5 \times 50$ ) axi-symétriques. Ce choix est conservé pour la modélisation 2D axi-symétrique. Afin d'illustrer les possibilités de modélisation tri-dimensionnelles du code, un modèle 3D est également réalisé avec des éléments hexahédriques à 8 noeuds. La période d'observation du phénomène est de  $t = 80 \mu\text{s}$ . A cet instant, la quasi-totalité de l'énergie cinétique initiale a été dissipée sous forme de travail plastique.

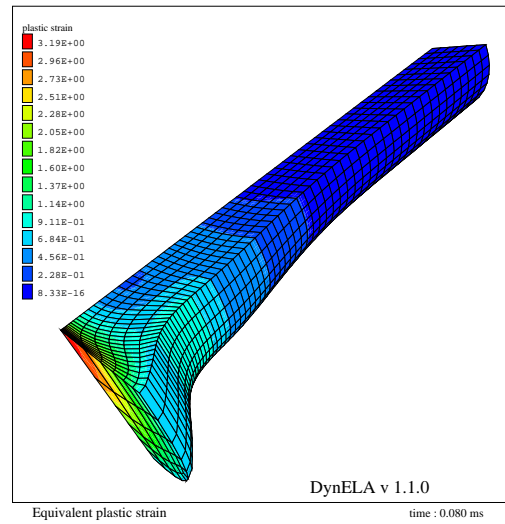
Afin de valider le comportement du code de calcul, on compare dans le tableau 3.3 les résultats obtenus à l'aide du code de calcul de **DynELA** concernant la longueur et le rayon final ainsi que la valeur de la déformation plastique équivalente maximale  $\bar{\epsilon}_{max}^p$  avec les résultats issus de la littérature. La figure 3.8 montre la cartographie des déformations plastiques équivalentes  $\bar{\epsilon}^p$  à la fin du calcul pour la simulation 2D axi-symétrique (figure 3.8(a)) et la simulation 3D (figure 3.8(b)). Sur la partie gauche de cette figure, la totalité de la pièce est représentée (la moitié seulement est calculée). La disparité des résultats numériques concernant la valeur de la déformation plastique équivalente  $\bar{\epsilon}^p$  résulte de la méthode utilisée par le post-processeur pour extrapoler les résultats entre les points d'intégration (sur lesquels sont calculées la déformation plastique équivalente) et les noeuds sur lesquels sont extrapolées les valeurs.

TAB. 3.3 – Comparaison des résultats de **DynELA** et références pour le test d'impact de Taylor

Code	Rayon maximum	Hauteur finale	$\bar{\epsilon}_{max}^p$
<b>DynELA</b> 2D axi	7.13	21.43	3.09
<b>DynELA</b> 3D	7.12	21.43	3.19
Abaqus Explicit	7.24	21.45	3.13
Liu [71]	7.15	21.42	
DYNA2D	7.13	21.47	3.05
MARC	7.02	21.66	3.24
METAFOR [37]	7.13	21.43	3.04
NIKE2D	7.07	21.47	2.47
GARCIA-G	7.11	21.47	3.09



(a) Calcul 2D axi-symétrique



(b) Calcul 3D

FIG. 3.8 – Déformations plastiques équivalentes pour le test d'impact de Taylor

### 3.4.3 Impact d'un longeron axi-symétrique

Le troisième cas test proposé est relatif à la prise en compte du contact dans un cas test dynamique. Le problème de l'impact d'un longeron cylindrique a été proposé par J. P. Ponthot [37]. Il s'agit de simuler l'écrasement d'un longeron cylindrique en régime dynamique. La fonction structurale de cet élément est de dissiper le maximum d'énergie (sous forme de travail plastique) lors d'un choc. On considère donc un tube cylindrique venant s'écraser dans une surface axi-symétrique rigide comme indiqué sur la figure 3.9. Les paramètres matériau correspondant à une loi d'écoulement élastoplastique linéaire sont donnés dans la table 3.4.

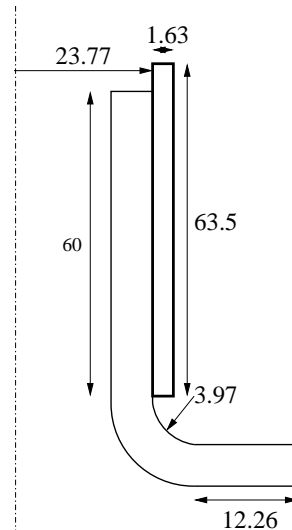
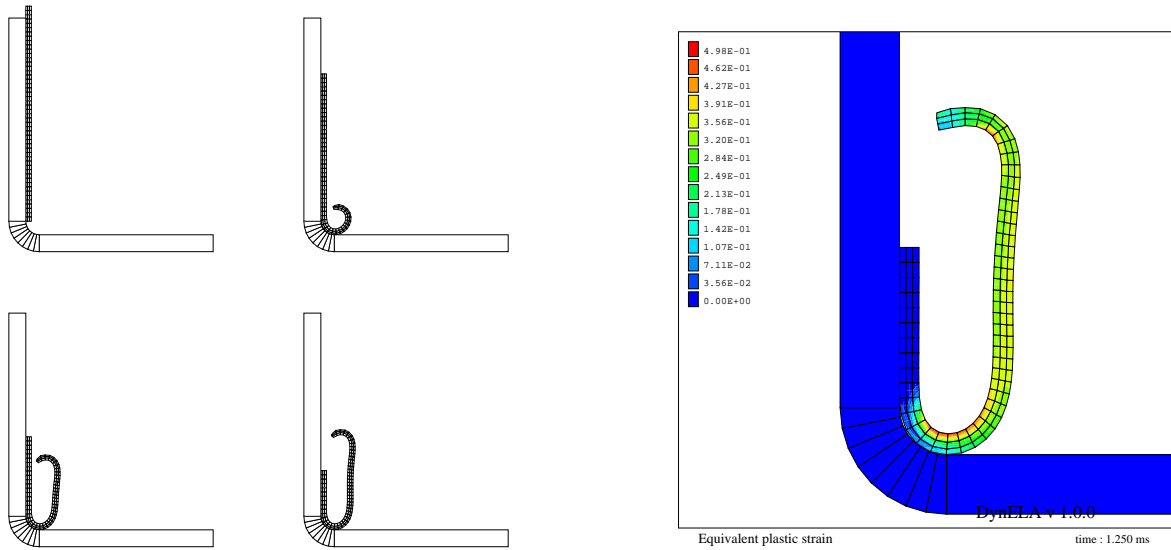


FIG. 3.9 – Données géométriques de l'impact d'un longeron cylindrique

TAB. 3.4 – Paramètres matériau pour l'impact d'un longeron

$E$	67.0GPa	$\sigma_0^y$	150MPa
$\nu$	0.33	$h$	44.7MPa
$\rho$	2700kg/m <sup>3</sup>		

Le maillage de la partie déformable est constitué de 150 éléments quadrilatères axi-symétriques à 4 noeuds ( $50 \times 3$ ). Les conditions aux limites et le chargement sont tels que la surface rigide est maillée grossièrement et tous les noeuds sont encastés. Le bord supérieur du longeron est soumis à un déplacement vertical à une vitesse constante de  $V_c = 40\text{ m/s}$ . Un contact sans frottement est créé entre le longeron et la structure rigide. La période d'observation est de  $t = 1,25 \cdot 10^{-3}\text{ s}$  soit un déplacement total de la partie supérieure du longeron de  $50\text{ mm}$ . La partie gauche de la figure 3.10 montre l'évolution de la géométrie à différents instants de la simulation ( $t = 0\text{ ms}$ ,  $t = 0.5\text{ ms}$ ,  $t = 1.0\text{ ms}$ ,  $t = 1.25\text{ ms}$ ). La partie droite de la figure 3.10 montre la répartition des déformations plastiques équivalentes dans la structure à la fin du calcul qui est en accord avec les résultats proposés par J. P. Ponthot [37] en utilisant le code de calcul Metafor développé au LTAS de l'université de Liège. De même, la comparaison des efforts de réaction en fonction du déplacement de la partie haute du longeron présentée sur la figure 3.11 montre une bonne corrélation des résultats numériques. Une vue 3D reconstituée par le post-processeur est donnée sur la figure 3.3 page 54.



(a) Evolution de la géométrie au cours du calcul

(b) Répartition des déformations plastiques équivalentes

FIG. 3.10 – Impact d'un longeron cylindrique

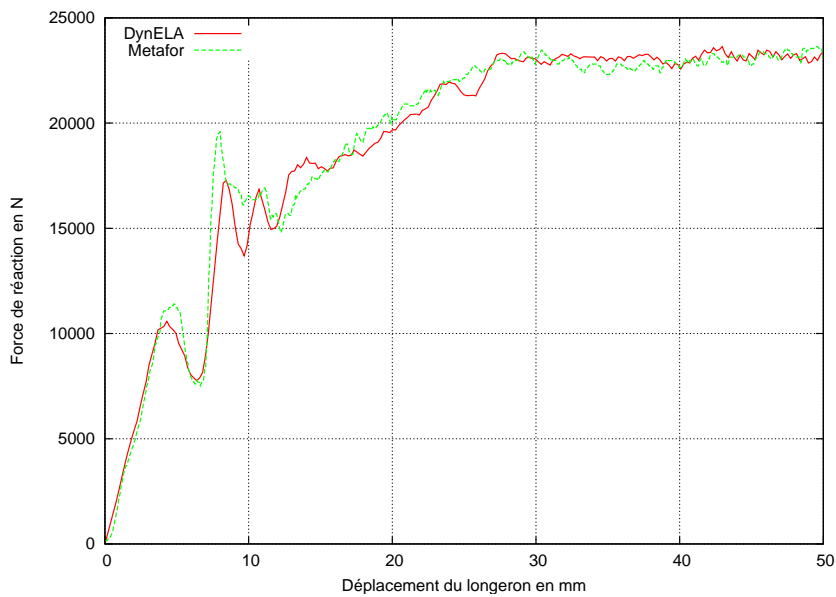


FIG. 3.11 – Comparaison des efforts de réaction pour l'impact d'un longeron cylindrique avec les résultats de J. P. Ponthot [37]

tel-00009913, version 1 - 8 Aug 2005

### 3.5 Parallélisation du code de calcul DynELA

L'augmentation de la taille des problèmes simulés et de la complexité des algorithmes utilisés dans les codes numériques actuels entraîne des temps de calcul qui peuvent se révéler rédhibitoires. D'une manière générale, les besoins en matière de puissance de calcul dépassent les capacités de résolution des processeurs, entraînant le recours à des moyens de calcul de plus en plus performants. Depuis de nombreuses années, la solution généralement proposée consiste à utiliser des architectures de calcul parallèle.

#### 3.5.1 Standards de programmation parallèle

Au cours de ces dernières années, les super-calculateurs ont été remplacés par des plateformes de calcul parallèle à base de microprocesseurs basées sur des architectures à mémoire partagée (*Shared Memory Processing* ou SMP) ou à mémoire distribuée (*Distributed Memory Processing* ou DMP). Dans une architecture SMP, l'ensemble des processeurs ont accès à la même zone mémoire centrale comme indiqué sur la figure 3.12 alors que dans une architecture DMP, chaque processeur dispose de sa propre zone mémoire. De plus, chaque processeur dispose de sa propre mémoire cache pour laquelle les accès sont très rapides. Cette présence de mémoire cache est un facteur très important pour la performance des codes de calcul.

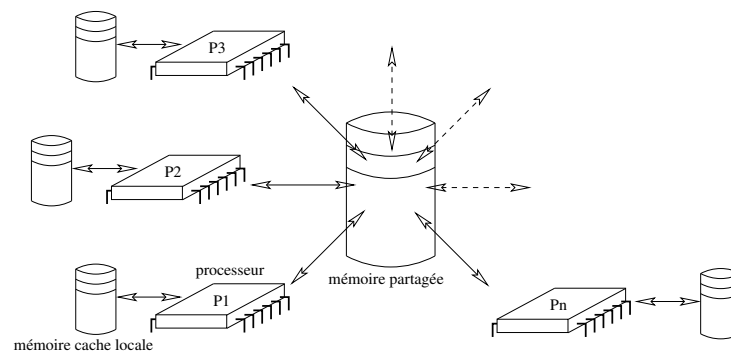


FIG. 3.12 – Architecture à mémoire partagée (SMP)

Les techniques de parallélisation mises en oeuvre dans les codes de calcul par éléments finis dépendent de l'architecture matérielle disponible. La première catégorie concerne les machines à mémoire distribuée (DMP) pour lesquelles le standard MPI (*Message Passing Interface*) est maintenant bien établi. C'est une bibliothèque de communication portable permettant d'écrire de manière relativement performante et simple les échanges de données entre les différents processeurs. Plusieurs travaux de développement de codes de calcul parallèles utilisant le standard MPI ont été publiés, on peut citer par exemple les travaux de Anderheggen et al. [72] et Brown et al. [73] qui présentent le développement de codes de calcul parallèles de crash.

La parallélisation sur les machines SMP fait appel à des directives de compilation spéciales permettant de contrôler l'exécution d'un programme de manière distribuée sur les différents processeurs et de synchroniser ces exécutions entre elles. Les opérations de contrôle de l'exécution des tâches sur les processeurs étant dépendantes de l'architecture physique de la machine, chaque constructeur propose généralement son propre jeu d'instructions ce qui rend difficile le portage des applications. Le standard OpenMP [74] est maintenant bien établi pour réaliser cette tâche. L'étendue de ses directives de programmation est plus restrictive que celle disponible en utilisant Pthread [75], mais, la simplicité de mise en oeuvre de OpenMP rend les développements plus rapides, et sa portabilité est meilleure.

Concernant les développements réalisés dans le cas de la parallélisation du code de calcul **DynELA**, le choix de l'architecture physique de calcul s'est porté sur une machine SMP de

type Compaq Proliant 8000 fonctionnant sous environnement Linux Redhat 8.0. Cette machine dont une illustration est donnée sur la figure 3.13 est équipée de 8 processeurs Intel Xeon PIII 550/2Mb autour d'une mémoire centrale partagée de 5Go. La compilation du code



FIG. 3.13 – Compaq Proliant 8000

source utilise le compilateur Intel C++ 7.1 sans paramètres d'optimisation de manière à pouvoir comparer les différentes techniques de parallélisation sans influence du compilateur. Le standard OpenMP a été retenu pour la parallélisation du code. Ses principales caractéristiques font l'objet de la section ci-après.

### 3.5.2 Présentation du standard OpenMP

Le type de parallélisme utilisé dans le standard OpenMP est parfois appelé *fork-join* car il consiste à lancer l'exécution de plusieurs tâches (ou **thread**<sup>(8)</sup>) simultanées sur plusieurs processeurs dans les régions parallélisées du code (action de ramification créant un nouveau thread à partir d'un autre ou **fork**) et à les joindre dans un seul *thread* (action de regroupement ou **join**) dans les régions non parallèles comme illustré sur la figure 3.14. Sur cette

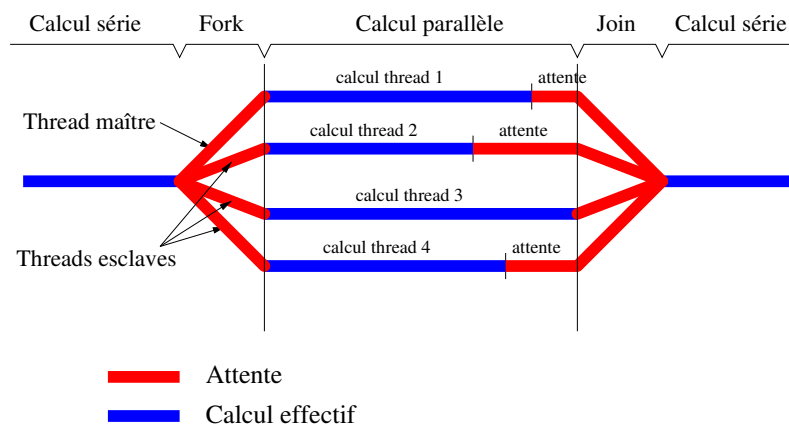


FIG. 3.14 – Parallélisme des tâches sur architecture SMP

même figure, les zones de calcul effectif et les zones d'attente sont reportées. Plus le temps nécessaire pour réaliser les opérations de **fork** et de **join** est élevé, plus l'algorithme est pénalisant, et ce d'autant plus que cette opération doit être répétée souvent. La parallélisation d'un code existant sous OpenMP peut être effectuée automatiquement par le compilateur, ou manuellement en incluant des directives de compilation dans le code source. Les deux méthodes ont été mises en oeuvre, mais, comme rapporté par Turner et al. [76], la parallélisation

<sup>(8)</sup>Un *thread* est une instance d'un programme exécutée par un utilisateur ou un processus.

automatique conduit à des performances désastreuses dans le cas des codes de calcul. La parallélisation manuelle se fait au moyen de directives `#pragma` en C++, comme illustré dans l'exemple ci-dessous :

```
void buildSystem(List <Element> elements) {
#pragma omp parallel for // defines a parallel loop
  for (int i=0;i<elements.size();i++) {
    elements(i).computeMatrices();
  }
}
```

Dans cet exemple la directive `#pragma omp parallel for` indique au compilateur que la prochaine boucle `for` doit être parallélisée et le travail correspondant doit être réparti sur plusieurs processeurs. La répartition est effectuée sur  $n$  processeurs, chacun se voyant attribuer  $\frac{elements.size()}{n}$  calculs à effectuer. Cette répartition est effectuée à priori, et on crée alors  $n$  *thread*. Par défaut, tous les *thread* effectuent les mêmes types d'opérations. Afin que la zone d'exécution parallèle fonctionne correctement, les diverses itérations effectuées doivent être totalement indépendantes les unes des autres, et la méthode `computeMatrices` de la classe `Element` doit être **thread-safe**, c'est à dire qu'elle doit pouvoir être appelée depuis les différents *thread* du programme sans que ceux-ci interfèrent. En utilisant des routines **thread-safe**, le risque que l'un des *thread* interfère et modifie les données d'un autre *thread* est évité en contrôlant et en synchronisant les accès simultanés aux zones de mémoire partagée. Le code parallèle est exécuté par tous les *thread* mis en jeu, y compris le *thread* maître. L'utilisateur dispose de plus de certaines directives (`shared`, `private`,...) utilisées pour définir la nature du partage des variables entre les différents *thread*. La définition par défaut est de type `shared`, une variable de type `private` possède une copie distincte par *thread*. De manière similaire aux boucles `for` parallèles, l'utilisateur peut définir des blocs de calcul parallèle en utilisant la directive `#pragma omp parallel`.

La synchronisation entre les différents *thread* utilise des directives spécifiques au formalisme OpenMP de type `barrier` ou `critical`. Une directive de type `barrier` implique pour un *thread* donné qu'il attende que tous les autres *thread* aient atteint le même point du programme. Une directive de type `critical` est utilisée pour restreindre l'accès à une partie donnée de code à un seul *thread* simultanément. C'est une notion fondamentale à utiliser lorsque les divers *thread* doivent modifier des variables partagées, mais c'est également une source de perte de performances en calcul parallèle comme on le verra plus loin.

### 3.5.3 Mise en oeuvre pour le code DynELA

La méthode la plus couramment adoptée pour mettre en oeuvre la parallélisation d'un code de calcul est le recours à la décomposition en sous-domaines (*Domain Decomposition Method* ou DDM) comme dans les travaux présentés par Anderheggen et al. [72] et Brown et al. [73]. Dans cette approche, la structure est décomposée en sous-domaines dont le traitement est affecté à un processeur. Cette approche fera l'objet de la section 4.1.1. La solution finale est assemblée en collectant les informations traitées par l'ensemble des processeurs en utilisant MPI. Une autre approche consiste à paralléliser les traitements numériques effectués au plus bas niveaux comme par exemple dans le calcul des produits de matrices en faisant appel à des bibliothèques de calcul algébrique parallèles. C'est une méthode particulièrement simple à mettre en oeuvre dans les codes de calcul implicites pour lesquels, on parallélise ainsi les phases d'inversion de matrice très coûteuses en temps de calcul.

L'approche adoptée dans le cadre de la parallélisation du code de calcul **DynELA** est différente puisque l'on s'est focalisé sur les techniques à mettre en oeuvre pour paralléliser localement les routines les plus consommatrices de temps CPU à l'intérieur de l'algorithme de résolution explicite. Après analyse des temps CPU requis pour les différentes étapes de l'algorithme général explicite donné dans la boîte 3.2 page 48, seules certaines parties ont été parallélisées en utilisant le standard OpenMP.

Afin de pouvoir comparer les performances relatives des diverses techniques de parallélisation, nous avons retenu l'impact de Taylor présenté au paragraphe 3.4.2 page 57 comme cas test numérique. Afin d'augmenter le temps de calcul de la structure et de montrer l'influence de la mémoire cache sur les performances, on utilise un premier maillage de 1000 éléments ( $10 \times 100$ ) et un second maillage de 6250 éléments ( $25 \times 250$ ).

### 3.5.3.1 Evaluation des temps de calcul

Dans le cas de l'utilisation d'un code de calcul explicite, la mesure des temps CPU est relativement délicate à mettre en oeuvre. En effet, on cherche à évaluer non seulement le temps global de résolution, mais surtout le détail des temps des opérations élémentaires de l'algorithme explicite. Pour cet objectif, une classe spécifique a été développée en utilisant l'instruction assembleur des processeurs Pentium RDTSC (*Real Time Stamp Counter*) qui renvoie le nombre de tops d'horloge écoulés depuis l'allumage de l'ordinateur. La limitation principale de cette méthode réside dans des erreurs de mesure liées au caractère multi-tâche du système d'exploitation Unix. En effet, en mesurant le temps écoulé entre deux points d'un algorithme, on n'est pas en mesure de certifier que ceci correspond réellement au temps de calcul nécessaire à l'avancement du programme. Sur un système multi-tâches, la phase de calcul a pu être interrompue plusieurs fois afin que le système réalise d'autres tâches. Néanmoins, en répétant plusieurs fois les opérations de calcul et en limitant au maximum les perturbations (machine hors réseau, désactivation d'un maximum de *daemons*<sup>(9)</sup>...) cette influence peut être minimisée.

Le but de la mesure des temps CPU est de pouvoir quantifier l'efficacité de la parallélisation du code de calcul au travers de l'évaluation du Speedup ( $s_p$ ). Il se définit comme étant le rapport du temps CPU ( $t_s$ ) nécessaire à l'exécution du programme avec un seul processeur, sur le temps CPU ( $t_m$ ) obtenu avec la version parallèle à  $n$  processeurs de l'application :

$$s_p = \frac{t_s}{t_m} \quad (3.11)$$

A partir de cette notion, on peut également définir l'efficacité ( $e_f$ ) de la parallélisation comme le rapport du Speedup sur le nombre de processeurs :

$$e_f = \frac{s_p}{n} \quad (3.12)$$

### 3.5.3.2 Calcul des forces internes

Comme présenté au chapitre 3, on utilise dans le code de calcul **DynELA** une approche prédicteur / correcteur pour l'intégration de la loi de comportement. Dans les applications dynamiques, le temps CPU nécessaire pour intégrer les lois de comportement varie d'un élément à l'autre et au cours du temps car l'écoulement plastique, très consommateur en temps de calcul, intervient dans des zones localisées. Ainsi, comme présenté au paragraphe 2.1.2 page 31, si le prédicteur élastique est physiquement admissible, l'algorithme de retour radial présenté dans la boîte 2.1 page 32 n'est pas exécuté. Mais cette détermination ne peut être faite a priori et nécessite l'évaluation en temps réel de l'équation (2.9) page 30.

Par conséquent, la prédiction du temps CPU nécessaire pour le calcul du vecteur des efforts internes  $\vec{F}^{int}$  est impossible à réaliser ici. Les diverses tâches de la zone de calcul parallèle demandent des temps de calcul différents conduisant à des pertes de temps comme illustré sur la figure 3.14 où le *thread 2* est le plus lent et le *thread 3* est le plus rapide. Pour réduire

<sup>(9)</sup>Un *daemon* est un programme installé sur un système UNIX pour accomplir automatiquement une tâche spécifique ou pour contrôler un périphérique.



ces temps d'attente, il est nécessaire d'implémenter un système d'équilibrage dynamique de la charge de chaque processeur. C'est le rôle affecté à la classe **Jobs** définie sur la figure 3.15. La classe **Job** contient la liste des éléments traités par chaque *thread*. La principale

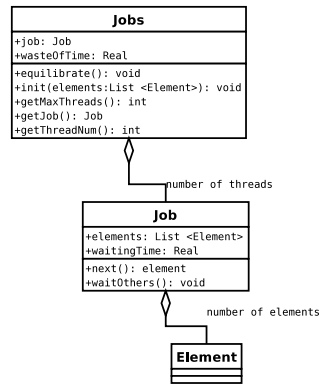


FIG. 3.15 – Description de la classe Jobs

difficulté de mise en place de cet algorithme d'équilibrage par rapport aux travaux réalisés par Rus et al. [77] est liée à l'utilisation d'un solveur explicite. Ceci implique que la procédure d'équilibrage doit être très rapide car elle est intégrée à l'intérieur de la boucle explicite de résolution.

Le calcul du vecteur des efforts internes  $\vec{F}^{int}$  est la partie qui présente le plus de besoins en calcul numériques. On présente ci-après plusieurs méthodes mises en oeuvre pour paralléliser cette opération avec leur influence sur l'efficacité. On présente ci-dessous le fragment de code C++ servant à calculer le vecteur des efforts internes  $\vec{F}^{int}$  :

```

Vector Fint;
for (int elm = 0; elm < elements.size (); elm++)
{
    Vector FintElm;
    FintElm = elements(elm).computeInternalForces ();
    Fint.gatherFrom (FintElm, elements(elm));
}
  
```

Dans le fragment de code C++ ci-dessus, la méthode **computeInternalForce** est appliquée pour chaque élément du maillage. Elle renvoie la valeur du vecteur des efforts internes local pour l'élément. La méthode **gatherFrom** assemble alors les vecteurs locaux dans le vecteur global des efforts internes de la structure. On présente ci-après 4 techniques différentes permettant de paralléliser cette partie de code avec leur influence sur l'efficacité du code pour le cas test de Taylor utilisant un maillage comprenant 1000 éléments.

1. Pour cette première méthode, on choisit d'utiliser la directive OpenMP **parallel for** dans la boucle principale et on utilise une variable partagée pour le vecteur **Fint**. On utilise donc une directive **critical** en amont de la méthode **gatherFrom** puisque le vecteur **Fint** est une variable partagée. Le code source modifié est donné dans la boîte 3.3.
2. Pour cette méthode, on utilise maintenant une directive de région parallèle (**parallel region**). Dans cette région parallèle, tous les *thread* ont une liste d'éléments à traiter. Le vecteur **Fint** est déclaré comme variable privée. Toutes les opérations peuvent être effectuées sans recours à la directive **critical**. A la fin du processus, les processeurs sont sollicités en parallèle afin d'assembler les copies locales propres à chaque *thread* du vecteur **Fint** dans le vecteur global.
3. Cette méthode est similaire à la précédente, mais ici, chaque *thread* à un nombre pré-défini d'éléments à traiter. Ceci évite d'utiliser une liste d'éléments à traiter partagée

### Algorithme 3.3 – Parallélisation du calcul de $\vec{F}^{int}$ par la méthode 1

```

Vector Fint; // internal force Vector

// parallel loop base on OpenMP pragma directive
#pragma omp parallel for
for (int elm = 0; elm < elements.size (); elm++)
{
    Vector FintElm; // local internal force Vector

    // compute local internal force vector
    FintElm = elements(elm).computeInternalForces ();

    // gather operation on global internal force vector
#pragma omp critical
    Fint.gatherFrom (FintElm, elements(elm));
} // end of parallel for loop

```

entre les divers *thread* (comme dans la méthode 2). La classe **Jobs** est utilisée pour cette opération.

4. La quatrième méthode est similaire à la précédente, mais ici, on introduit en plus l'équilibrage dynamique de charge présenté précédemment. Le code source correspondant est donné dans la boîte 3.4. On remarque une forte modification du source modifié par rapport à la version d'origine.

### Algorithme 3.4 – Parallélisation du calcul de $\vec{F}^{int}$ par la méthode 4

```

jobs.init(elements); // list of jobs to do
int threads = jobs.getMaxThreads(); // number of threads
Vector Fint = 0.0; // internal force Vector
Vector FintLocal[threads]; // local internal force vectors

// parallel computation of local internal force vectors
#pragma omp parallel
{
    Element* element;
    Job* job = jobs.getJob(); // get the job for the thread
    int thread = jobs.getThreadNum(); // get the thread Id

    // loop while exists elements to treat
    while (element = job->next())
    {
        Vector FintElm; // element force vector

        // compute local internal force vector
        FintElm = elements(elm).computeInternalForces ();

        // gather operation on local internal force vector
        FintLocal[thread].gatherFrom (FintElm, element);
    }
    job->waitOthers(); // compute waiting time for the thread
} // end of parallel region

// parallel gather operation
#pragma omp parallel for
for (int row = 0; row < Fint.rows(); row++)
{
    // assemble local vectors into global internal force vector
    for (thread = 0; thread < threads; thread++)
        Fint(row) += FintLocal[thread](row);
} // end of parallel for loop

// equilibrate the sub-domains
jobs.equilibrate();

```

Le tableau 3.5 regroupe les résultats des tests numériques effectués et montre que la valeur de l'efficacité peut parfois être supérieure à 100% comme indiqué par les travaux de J. Gustafson [78]. Ce résultat provient du fait que, suite à la répartition des diverses tâches sur les processeurs, le problème local au niveau de chaque processeur est réduit et peut plus facile-

TAB. 3.5 – Speedup du calcul de  $\vec{F}^{int}$  pour les diverses méthodes

méthode	1 CPU	4 CPU			8 CPU		
	temps	temps	Speedup	efficacité	temps	Speedup	efficacité
1	167.30	57.95	2.88	72.2%	72.25	2.31	28.9%
2	163.97	45.98	3.56	89.1%	25.39	6.45	80.7%
3	164.52	42.18	3.90	97.5%	20.86	7.88	98.5%
4	164.25	38.55	4.26	106.5%	19.66	8.35	104.4%

ment être intégralement stocké dans la mémoire cache du processeur d'où un gain de vitesse notable. Afin de confirmer cette remarque, si l'on effectue la même simulation en utilisant le maillage à 6250 éléments au lieu du maillage à 1000 éléments, on obtient une efficacité de 90% pour 8 processeurs au lieu de 104.4%.

La figure 3.16 montre la variation du Speedup en fonction du nombre de processeurs utilisés pour le calcul. On vérifie bien graphiquement sur cette figure que la méthode 1 conduit à

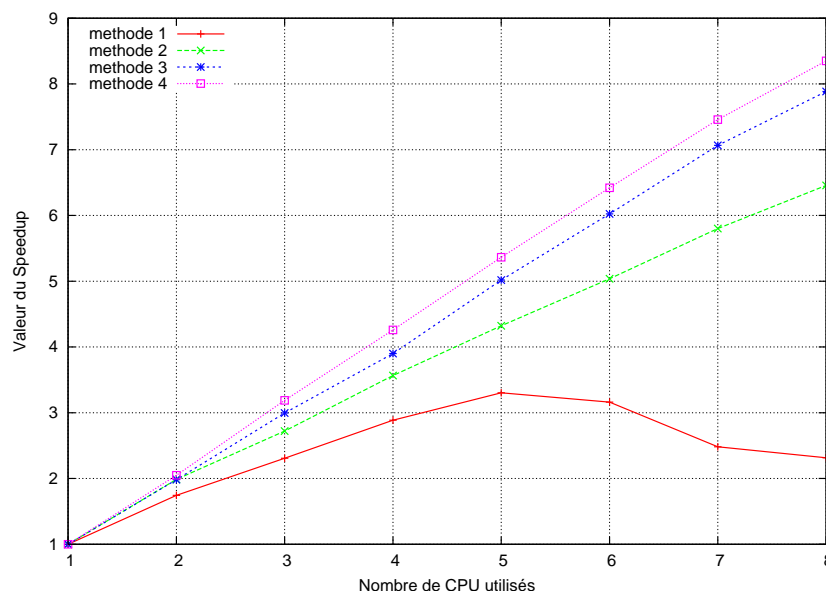


FIG. 3.16 – Variation du Speedup en fonction du nombre de processeurs

des résultats désastreux, spécialement au delà de 5 processeurs alors que les performances des méthodes 3 et 4 sont très satisfaisantes. En fait, pour la méthode 1, la présence de la directive **critical** au niveau de l'opération **gatherFrom** conduit à une file d'attente car un seul processeur à la fois peut réaliser cette opération très consommatrice de temps CPU. Dans la seconde méthode, il subsiste une directive **critical** destinée à synchroniser l'accès à la liste partagée des éléments à traiter; celle-ci pénalise la performance. Les méthodes 3 et 4 sont les plus optimisées. Bien que nécessitant des instructions supplémentaires, la méthode 4 qui inclue l'équilibrage dynamique est globalement la plus performante.

Une application plus détaillée concernant la simulation numérique d'un essai de traction est présentée dans [79,80] avec une analyse de la procédure d'équilibrage dynamique de la charge. Il est nécessaire de préciser que dans cette application de parallélisation, on ne peut pas mettre en oeuvre plus de processeurs que le nombre disponible sur notre machine. Les performances obtenues en terme d'efficacité de la parallélisation ne sont donc valables que pour un nombre de processeurs limité à 8. Il n'est pas garanti que l'utilisation du même code sur une machine ayant plus de processeurs conduise aux mêmes résultats. Il est cependant

nécessaire de rappeler que les machines à architecture SMP ont généralement un nombre de processeurs limité à 32. Les principes proposés ici ne sont donc pas transposables directement dans le cas de structures de calcul massivement parallèles telles que les COW (*Cluster of Workstations*). Dans le cas d'architectures parallèles plus larges, un autre type d'approche généralement adopté consiste à utiliser une combinaison des standards OpenMP et MPI simultanément. Ce type d'approche est envisagé dans les perspectives de développement du code de calcul **DynELA** sur la base d'un *cluster* composé de plusieurs machines multiprocesseurs ou non reliées entre elles par un réseau à haut débit.

### 3.6 Utilisations du code de calcul DynELA

Les possibilités offertes par la mise à disposition du code de calcul **DynELA** sont multiples. Parmi celles-ci, on peut citer les principaux points suivants :

- Le premier domaine d'application du code de calcul **DynELA** réside dans le développement et la mise au point de nouveaux algorithmes de calcul dans le cadre des méthodes éléments finis en grandes transformations principalement. La plateforme développée est actuellement opérationnelle, elle peut donc servir de base pour de nouveaux développements numériques donnant lieu à une application métier. Une présentation des deux principales applications métier réalisées à l'heure actuelle en utilisant le code de calcul **DynELA** comme plateforme de développement fait l'objet du chapitre 4 de ce document.
- L'une des applications directes du code de calcul réalisé réside dans l'identification des lois de comportement puisqu'il peut facilement être utilisé à partir du logiciel **Pilote**. Les avantages résident alors dans la possibilité d'implémenter de nouvelles lois de comportement puisque l'on dispose des sources du code de calcul et d'identifier les paramètres de ces lois, et ceci quelles que soient leurs formulations. De plus la gamme d'utilisation est adaptée aux moyens expérimentaux du laboratoire.
- Un autre avantage réside dans le fait que le laboratoire LGP est propriétaire du code de calcul. On est ainsi en mesure de le distribuer sans contrainte à des utilisateurs externes qui peuvent l'utiliser localement dans leurs laboratoires.
- D'un point de vue pédagogique, actuellement, le code de calcul **DynELA** est utilisé dans le cadre du cours de Grandes Transformations du Master Recherche (M2R) à l'ENIT par les étudiants en séances de Travaux Pratiques. L'objectif de ces séances est de permettre aux étudiants d'appréhender les difficultés liées à la modélisation numérique en grandes transformations au travers d'un code de calcul ne disposant pas des "artifices" numériques lui permettant de converger vers une solution comme dans le cas de l'emploi du code de calcul Abaqus. On peut également agir sur les diverses parties du code de calcul, comme les paramètres de l'algorithme d'intégration numérique, afin d'illustrer les problèmes rencontrés dans la modélisation en grandes transformations.

*Ces travaux ont fait l'objet de 3 publications internationales et 4 participations à des congrès.*

# 4

# Plateforme Orientée-Objets de développement

---

*The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but rather, 'hmm.... that's funny...'*

Isaac Asimov (1920-1992)

Les développements numériques réalisés dans le cadre de l'implémentation du code de calcul explicite en dynamique rapide **DynELA** ont servi de base de travail pour le développement de nouvelles applications. L'idée force ici est d'utiliser les bibliothèques développées pour le code de calcul **DynELA** afin de constituer une plateforme de développement pour de nouvelles applications numériques. On peut ainsi utiliser les librairies de base et les librairies mathématiques, la structure éléments finis, l'interpréteur de commande et le post-processeur graphique. A partir des classes de base de la librairie éléments finis, on dérive de nouvelles classes plus spécialisées dans lesquelles on implémente les nouveaux algorithmes. On profite ainsi de l'approche Orientée-Objets via le mécanisme d'héritage. Lorsque les nouveaux développements sont stabilisés, s'ils sont jugés utilisables dans le cadre des applications liées à l'impact et au crash, les développements sont alors intégrés dans le code source de base. Dans ce chapitre, on présente des travaux de recherche faisant appel à la plateforme de développement Orientée-Objets de **DynELA**.

## 4.1 Plateforme de développement Orientée-Objets

Dans cette section, on présente une sélection de nouveaux algorithmes et de nouvelles approches développés au cours des travaux de thèse de L. Menanteau [III] et I. Nistor [IV] dans le cadre de la modélisation numérique par éléments finis. La particularité de ces travaux est liée à leur implémentation numérique en utilisant la plateforme de développement de **DynELA**.

### 4.1.1 Décomposition spatiale en sous-domaines

Dans le cadre de la collaboration de recherche avec le laboratoire PEARL<sup>(1)</sup>, de nouvelles problématiques ont été mises en place au niveau du groupe de recherche. La problématique posée par le laboratoire PEARL concerne la simulation numérique de convertisseurs de puissance pour des chaînes de traction ferroviaires. La principale caractéristique de ces modèles numériques est liée à la taille de ces modèles qui nécessitent des ressources de calcul importantes.

---

<sup>(1)</sup>Power Electronics Associated Research Laboratory : Situé au sein de la société Alstom Transport sur le site de Séméac (65).

L'une des solutions proposée lors de cette collaboration concerne l'implémentation numérique d'une méthode de décomposition en sous-domaines. Ces travaux de collaboration ont été réalisés dans le cadre de la thèse de L. Menanteau [III]. L'approche retenue repose sur une décomposition de type duale [81] pour laquelle le domaine discrétisé est partitionné en  $s$  sous-domaines séparés par une interface. A partir d'une formulation variationnelle classique, on exprime la formulation éléments finis relative à chaque sous-domaine  $j$ , la continuité des quantités primales sur l'interface et le vecteur général des interactions interfaciales qui conduisent au système suivant :

$$\begin{bmatrix} \mathbf{K}^{(1)} & \dots & 0 & -\mathbf{B}^{(1)T} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathbf{K}^{(s)} & -\mathbf{B}^{(s)T} \\ -\mathbf{B}^{(1)} & \dots & -\mathbf{B}^{(s)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}^{(1)} \\ \vdots \\ \mathbf{q}^{(s)} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{g}^{(1)} \\ \vdots \\ \mathbf{g}^{(s)} \\ 0 \end{bmatrix} \quad (4.1)$$

dans lequel  $\mathbf{K}^{(j)}$  est la matrice représentant le comportement physique du sous-domaine  $j$ ,  $\mathbf{q}^{(j)}$  est le vecteur des inconnues nodales primales,  $\mathbf{g}^{(j)}$  est le vecteur des actions extérieures appliquées sur le sous-domaine  $j$ ,  $\mathbf{B}^{(j)}$  est la matrice qui sélectionne les degrés de liberté du sous-domaine  $j$  qui sont relatifs à l'interface et  $\lambda$  est le vecteur général des interactions interfaciales.

La difficulté principale rencontrée dans ce type d'approche concerne la présence de sous-domaines flottants pour lesquels la matrice  $\mathbf{K}^{(j)}$  étant singulière et non inversible, on doit alors recourir à la notion d'inverse généralisée notée  $\mathbf{K}^+$ . En introduisant la notation  $\mathbf{R}^{(j)}$  pour désigner le noyau de  $\mathbf{K}^{(j)}$ ,  $\gamma^{(j)}$  pour les amplitudes inconnues de  $\mathbf{R}^{(j)}$  et en utilisant  $\mathbf{K}^+$  pour désigner indifféremment une inverse généralisée ou une inverse de  $\mathbf{K}$ , on est en mesure de formuler le problème interfacial sous la forme suivante pour un problème à  $s$  sous-domaines dont  $f \leq s$  sont flottants [III] :

$$\begin{bmatrix} \mathbf{F}_I & \mathbf{G}_I \\ \mathbf{G}_I^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\lambda \\ \mathbf{g}_\gamma \end{bmatrix} \quad (4.2)$$

avec les notations :

$$\begin{cases} \mathbf{F}_I = \sum_{j=1}^s \mathbf{B}^{(j)} \mathbf{K}^{(j)+} \mathbf{B}^{(j)T} \\ \mathbf{G}_I = \begin{bmatrix} \mathbf{B}^{(1)} \mathbf{R}^{(1)} & \dots & \mathbf{B}^{(f)} \mathbf{R}^{(f)} \end{bmatrix} \\ \gamma = \begin{bmatrix} \gamma^{(1)T} & \dots & \gamma^{(f)T} \end{bmatrix}^T \\ \mathbf{g}_\lambda = -\sum_{j=1}^s \mathbf{B}^{(j)} \mathbf{K}^{(j)+} \mathbf{g}^{(j)} \\ \mathbf{g}_\gamma = \begin{bmatrix} \mathbf{R}^{(1)T} \mathbf{g}^{(1)} & \dots & \mathbf{R}^{(f)T} \mathbf{g}^{(f)} \end{bmatrix}^T \end{cases} \quad (4.3)$$

Dans la version actuelle, liée aux travaux réalisés pendant la période de thèse de L. Menanteau [III], l'étude est restreinte au cas où tous les noeuds faisant partie de l'interface ne sont liés au plus qu'à deux sous-domaines simultanément afin de simplifier l'écriture des matrices de liaison interfaciales  $\mathbf{B}^{(i)}$  qui sont alors booléennes signées. Une extension au cas général est actuellement à l'étude. D'un point de vue numérique, la résolution du problème relatif à chaque sous-domaine peut être effectuée de manière directe ou itérative comme présenté dans Menanteau et al. [82].

D'un point de vue pratique, l'implémentation de la décomposition en sous-domaines a nécessité une modification de la structure des données de la plateforme de développement afin de pouvoir disposer de plusieurs domaines (liste de domaines) au sein d'une structure. Cette modification peut permettre également de spécifier plusieurs solveurs de nature différente sur diverses zones de la structure (solveurs explicites+implicites). Les classes des solveurs ont donc été rattachées à la classe définissant le domaine et non à la classe structure comme précédemment fait. Cette nouvelle approche a été également adoptée dans le code de calcul

**DynELA** comme illustré sur la figure 3.2 page 51. Il est important de préciser que la totalité des modifications, liées au changement de la structure de données, a été effectuée en moins d'une journée du fait de l'approche Orientée-Objets.

Dans le cadre du développement du code de calcul **DynELA**, les apports principaux de la décomposition en sous-domaines concernent les deux points suivants :

- utilisation sur environnement de calcul parallèle de type *cluster* avec la librairie de communication MPI,
- utilisation de plusieurs solveurs simultanés comme par exemple une résolution implicite sur certaines zones et explicite sur d'autres.

#### 4.1.2 Décomposition temporelle

Le deuxième axe de travail exploré au cours de la collaboration LGP / PEARL concerne l'intégration temporelle et le calcul multi-pas de temps dont l'idée générale réside dans l'utilisation de plusieurs pas de temps pour l'intégration numérique sur une même structure, afin de réduire les temps de simulation. On est alors en mesure d'utiliser des pas de temps petits dans les zones fortement non-linéaires et plus grands dans les autres zones. La décomposition temporelle présentée ici est rendue possible grâce à la décomposition en sous-domaines présentée précédemment.

Selon la nature du système à résoudre ; système du premier ordre en thermique ou système du second ordre en mécanique, les schémas d'intégration en temps diffèrent. Afin de rester cohérent (approche en mécanique) et succinct dans cette présentation, j'ai choisi de ne proposer ici que quelques points significatifs relatifs aux systèmes du second ordre. L'ensemble des études concernant le premier et second ordre peut être consulté dans la thèse de L. Menanteau [III].

Concernant les problèmes du second ordre, le schéma d'intégration temporel de type Newmark exprime le vecteur vitesse  $\dot{\vec{x}}^{(j)}$  et le vecteur de position  $\vec{x}^{(j)}$  entre les instants  $n$  et  $n+1$  pour chaque sous-domaine  $j$  par :

$$\dot{\vec{x}}_{n+1}^{(j)} = \dot{\vec{x}}_n^{(j)} + (1 - \gamma^{(j)}) \Delta t^{(j)} \ddot{\vec{x}}_n^{(j)} + \gamma^{(j)} \Delta t^{(j)} \ddot{\vec{x}}_{n+1}^{(j)} \quad (4.4)$$

$$\vec{x}_{n+1}^{(j)} = \vec{x}_n^{(j)} + \Delta t^{(j)} \dot{\vec{x}}_n^{(j)} + \left(\frac{1}{2} - \beta^{(j)}\right) \Delta t^{(j)^2} \ddot{\vec{x}}_n^{(j)} + \beta^{(j)} \Delta t^{(j)^2} \ddot{\vec{x}}_{n+1}^{(j)} \quad (4.5)$$

où  $\gamma^{(j)}$  et  $\beta^{(j)}$  sont les paramètres du schéma d'intégration relatifs à chaque sous-domaine  $j$ . On applique alors classiquement un schéma de type prédicteur/correcteur pour lequel, en notant  $(\ )_p$  un prédicteur pour la quantité  $(\ )$ , on a :

$$\begin{cases} \dot{\vec{x}}_p^{(j)} = \dot{\vec{x}}_n^{(j)} + (1 - \gamma^{(j)}) \Delta t^{(j)} \ddot{\vec{x}}_n^{(j)} \\ \vec{x}_p^{(j)} = \vec{x}_n^{(j)} + \Delta t^{(j)} \dot{\vec{x}}_n^{(j)} + \left(\frac{1}{2} - \beta^{(j)}\right) \Delta t^{(j)^2} \ddot{\vec{x}}_n^{(j)} \\ \dot{\vec{x}}_{n+1}^{(j)} = \dot{\vec{x}}_p^{(j)} + \gamma^{(j)} \Delta t^{(j)} \ddot{\vec{x}}_{n+1}^{(j)} \\ \vec{x}_{n+1}^{(j)} = \vec{x}_p^{(j)} + \beta^{(j)} \Delta t^{(j)^2} \ddot{\vec{x}}_{n+1}^{(j)} \end{cases} \quad (4.6)$$

Compte tenu de la décomposition en sous-domaines, l'équation d'équilibre (3.1) présentée page 46, se met sous la forme suivante pour chaque sous-domaine  $j$  :

$$\mathbf{M}_n^{(j)} \ddot{\vec{x}}_n^{(j)} + \mathbf{K}_n^{(j)} \vec{x}_n^{(j)} - \overrightarrow{F}_n^{ext(j)} = \overrightarrow{F}_n^{ifc(j)} \quad (4.7)$$

dans laquelle  $\overrightarrow{F_n^{ifc(j)}}$  représente le vecteur des efforts interfaciaux sur le sous-domaine  $j$  et  $\mathbf{K}_n^{(j)}$  est la matrice de rigidité du domaine  $j$ . Cette équation s'accompagne alors de celle traduisant la continuité des quantités primales sur l'interface :

$$\sum_{j=1}^s \mathbf{B}^{(j)} \dot{\overrightarrow{x}}_n^{(j)} = 0 \quad (4.8)$$

Le choix concernant la quantité continue à l'interface peut être fait sur les déplacements  $\overrightarrow{x}_n^{(j)}$ , les vitesses  $\dot{\overrightarrow{x}}_n^{(j)}$  ou les accélérations  $\ddot{\overrightarrow{x}}_n^{(j)}$ . Les travaux réalisés dans le cadre de la thèse ont montré que le choix d'une continuité en vitesse n'altérerait pas la précision et la stabilité quelles que soient les valeurs utilisées pour les paramètres  $\gamma^{(j)}$  et  $\beta^{(j)}$  du schéma d'intégration. Compte tenu alors du schéma d'intégration utilisé et de la partition du problème en une partie libre (solution des efforts extérieurs et intérieurs à chaque sous-domaine  $j$ ) et une partie liée (solution des efforts interfaciaux appliqués au sous-domaine  $j$ ), on peut établir alors l'algorithme de résolution transitoire d'un problème du second ordre comme indiqué dans la boîte 4.5.

#### Algorithme 4.5 – Résolution transitoire pour un problème du second-ordre

1. Construction de la matrice de rigidité  $\mathbf{K}$  et calcul de la matrice  $\widetilde{\mathbf{M}}$  :

$$\widetilde{\mathbf{M}}^{(j)} = \mathbf{M}^{(j)} + \beta^{(j)} \Delta t^{(j)2} \mathbf{K}^{(j)}$$

2. Résolution du problème libre sur chaque sous-domaine  $j$  :

$$\ddot{\overrightarrow{x}}_{libre}^{(j)} = \mathbf{M}^{(j)-1} \left( \overrightarrow{F}^{ext(j)} - \mathbf{K}^{(j)} \overrightarrow{x}_p^{(j)} \right)$$

3. Calcul des quantités cinématiques libres des sous-domaines :

$$\begin{cases} \dot{\overrightarrow{x}}_{libre}^{(j)} = \dot{\overrightarrow{x}}_p^{(j)} + \gamma^{(j)} \Delta t^{(j)} \ddot{\overrightarrow{x}}_{libre}^{(j)} \\ \overrightarrow{x}_{libre}^{(j)} = \overrightarrow{x}_p^{(j)} + \beta^{(j)} \Delta t^{(j)2} \ddot{\overrightarrow{x}}_{libre}^{(j)} \end{cases}$$

4. Résolution du problème interfacial :

$$\lambda = - \left( \sum_{j=1}^s \gamma^{(j)} \Delta t^{(j)} \mathbf{B}^{(j)} \widetilde{\mathbf{M}}^{(j)} \mathbf{B}^{(j)T} \right)^{-1} \left( \sum_{j=1}^s \mathbf{B}^{(j)} \dot{\overrightarrow{x}}_{libre}^{(j)} \right)$$

5. Résolution du problème libre sur chaque domaine :

$$\ddot{\overrightarrow{x}}_{lie}^{(j)} = \mathbf{M}^{(j)-1} \mathbf{B}^{(j)T} \lambda$$

6. Correction des valeurs finales :

$$\begin{cases} \ddot{\overrightarrow{x}}_n^{(j)} = \ddot{\overrightarrow{x}}_{libre}^{(j)} + \ddot{\overrightarrow{x}}_{lie}^{(j)} \\ \dot{\overrightarrow{x}}_n^{(j)} = \dot{\overrightarrow{x}}_{libre}^{(j)} + \gamma^{(j)} \Delta t^{(j)} \ddot{\overrightarrow{x}}_{lie}^{(j)} \\ \overrightarrow{x}_n^{(j)} = \overrightarrow{x}_{libre}^{(j)} + \beta^{(j)} \Delta t^{(j)2} \ddot{\overrightarrow{x}}_{lie}^{(j)} \end{cases}$$

Dans le cas où les divers sous-domaines n'ont pas le même pas de temps  $\Delta t^{(j)}$ , on introduit



alors pour tous les sous-domaines ayant un pas de temps supérieur au plus petit pas de temps de la structure  $\Delta t_{min}$  une interpolation de la vitesse et on ré-écrit alors la continuité des quantités primales sur l'interface par la relation :

$$\sum_{j=1}^s \mathbf{B}^{(j)} \dot{\hat{x}}_n^{(j)} = 0 \quad (4.9)$$

avec  $\dot{\hat{x}}_n^{(j)}$  calculée par interpolation de la vitesse  $\dot{x}_n^{(j)}$  à l'instant  $t$ . Comme on utilise une interpolation des quantités, on résout les problèmes sur les sous-domaines en les considérant dans l'ordre des pas de temps décroissants.

La même démarche peut être menée pour l'intégration temporelle des systèmes du premier ordre, et les détails de l'implémentation numérique sont disponibles dans la thèse de L. Menanteau [III] et Menanteau et al. [83].

### 4.1.3 Multi-physique

Un dernier aspect des travaux liés à la collaboration avec le laboratoire PEARL concerne le développement d'un module de calcul multi-physique capable de simuler le comportement des convertisseurs de puissance pour lesquels, 3 physiques entrent en jeu : les physiques électrique, thermique et mécanique. D'un point de vue implémentation, on introduit une nouvelle classe (**Physic**) permettant de définir la physique, entre la classe **Structure** et la classe **Domain**. Le solveur attaché à la structure est chargé de piloter les divers solveurs associés à chaque physique et d'effectuer les échanges d'informations entre les physiques. Chaque physique peut être résolue seule, ou de manière couplée avec une ou plusieurs autres physiques. La nature des quantités utilisées selon le type de physique est décrite dans le tableau 4.1.

TAB. 4.1 – Nature des quantités selon le type de physique

physique	$\mathbf{K}^{(j)}$	$\mathbf{q}^{(j)}$	$\mathbf{g}^{(j)}$ et $\lambda$
électrique	conductivité électrique	potentiels	courants
thermique	conductivité thermique	températures	flux de chaleur
mécanique	rigidité mécanique	déplacements	efforts

Trois types de résolutions couplées ont été implémentées :

- La résolution électro-thermique pour laquelle les résultats issus du calcul électrique (potentiels) permettent de calculer les dissipations par effet Joule dans les conducteurs passifs, et par commutation dans les conducteurs actifs. L'analyse thermique est complétée par la prise en compte des conditions aux limites thermiques et fournit alors le champ de températures de la structure.
- L'analyse thermo-mécanique suit le même principe. L'analyse thermique permet de calculer les températures en tout point de la pièce à partir des conditions aux limites et des chargements. L'analyse mécanique permet alors de calculer le champ des contraintes sous l'action des chargements mécaniques et des efforts issus notamment des dilatations différentielles dues aux gradients de températures.
- L'analyse électro-thermo-mécanique dans laquelle le logiciel enchaîne les étapes électro-thermique et thermo-mécanique précédentes.

Actuellement, et pour des raisons de simplicité, dans la formulation multi-physique, toutes les physiques sont basées sur un maillage élément finis identique. Cette simplification permet d'éviter le recours à l'interpolation spatiale des données entre les différentes physiques. Dans une version ultérieure, les diverses physiques pourront porter des maillages différents et seules les géométries des modèles devront être identiques.

#### 4.1.4 Développement de l'approche eXtended Finite Element Method

Dans le cadre d'une collaboration entre le LGP et l'ATM<sup>(2)</sup>, une nouvelle problématique concernant la modélisation numérique de la fissuration en dynamique rapide a été développée. Comme introduit au paragraphe 2.2.2.1, lorsque la sollicitation dynamique est trop importante sur une structure, un mécanisme d'endommagement conduisant à la fissuration prend place. La prise en compte de ces phénomènes de fissuration a pris de plus en plus d'importance dans les simulations numériques ces dernières années. Plusieurs approches sont actuellement proposées afin de modéliser numériquement le comportement de ces fissures dans un matériau, et les approches classiquement retenues consistent à remailler la pièce au fur et à mesure de l'avancement de la fissure. Récemment, de nouvelles méthodes basées sur la discontinuité des champs à l'intérieur des éléments ont été proposées. L'une d'elles, connue sous la dénomination de *eXtended Finite Element Method* (XFEM) est celle qui a été retenue pour le travail de thèse de I. Nistor [IV]. Elle est basée sur les travaux de Moes et al. [84] et Dolbow et al. [85]. La représentation d'une fissure dans le cadre de l'approche XFEM est basée sur le principe de l'enrichissement du champ de déplacement par une fonction de type Heaviside. Ainsi, dans le cas d'une analyse bidimensionnelle, les noeuds enrichis possèdent alors 2 degrés de liberté classiques et 2 degrés de liberté enrichis pour prendre en compte la discontinuité du champ de déplacements. Dans cette approche, la fissure se propage à l'intérieur des éléments comme indiqué sur la figure 4.1, contrairement aux méthodes classiques utilisées dans les codes du commerce pour lesquels la fissure se propage entre les éléments.

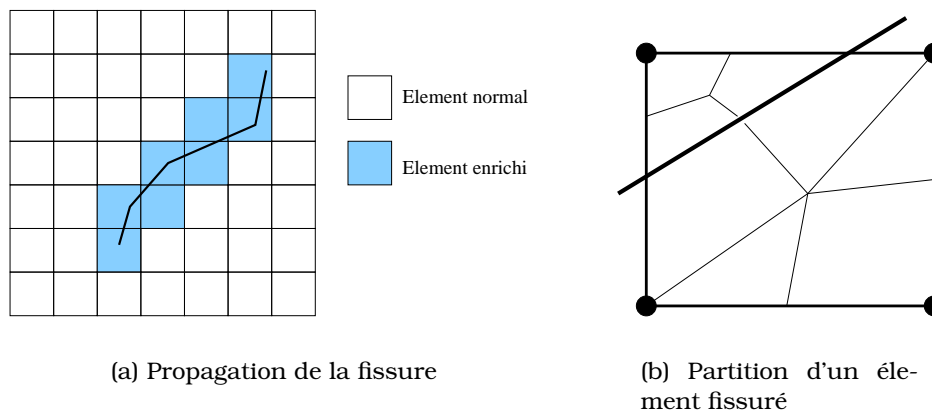


FIG. 4.1 – Principe général de la méthode XFEM

Si l'on prend en considération une approche bidimensionnelle pour laquelle la structure est discrétisée avec des éléments de type quadrilatère à 4 noeuds, le champ de déplacement  $\vec{u}^h(\vec{x})$  en un point donné est alors donné par la relation :

$$\vec{u}^h(\vec{x}) = \sum_{i=1}^4 \mathbf{N}_i(\vec{x}) \vec{u}_i + \sum_{i=1}^4 \mathbf{N}_i(\vec{x}) H(\vec{x}) \tilde{\vec{u}}_i \quad (4.10)$$

<sup>(2)</sup>Académie Technique Militaire de Bucarest

dans laquelle  $\vec{\hat{u}}_i$  est le vecteur des déplacements enrichis,  $H(\vec{x})$  est la fonction de Heaviside permettant la localisation de la fissure et  $\mathbf{N}_i$  sont les fonctions d'interpolation au noeud  $i$  des champs de déplacement réels  $\vec{u}$  et enrichis  $\vec{\hat{u}}$ . Du fait de l'introduction des enrichissements dans le champ de déplacement, les divers termes constitutifs de l'équation d'équilibre (3.1) donnée page 46 diffèrent de l'écriture proposée en (3.3) :

$$\begin{bmatrix} \mathbf{M}_{uu} & \mathbf{M}_{u\hat{u}} \\ \mathbf{M}_{\hat{u}u} & \mathbf{M}_{\hat{u}\hat{u}} \end{bmatrix} \begin{Bmatrix} \ddot{\vec{u}} \\ \ddot{\vec{\hat{u}}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{u\hat{u}} \\ \mathbf{K}_{\hat{u}u} & \mathbf{K}_{\hat{u}\hat{u}} \end{bmatrix} \begin{Bmatrix} \vec{u} \\ \vec{\hat{u}} \end{Bmatrix} - \begin{Bmatrix} \overrightarrow{F^{ext}} \\ \overrightarrow{F^{ext}} \end{Bmatrix} = 0 \quad (4.11)$$

avec la matrice de masse définie par :

$$\begin{aligned} \mathbf{M}_{uu} &= \mathbf{M}_{\hat{u}\hat{u}} = \int_{\Omega_x} \rho \mathbf{N}^T \mathbf{N} d\Omega_x \\ \mathbf{M}_{u\hat{u}} &= \mathbf{M}_{\hat{u}u} = \int_{\Omega_x} \rho \mathbf{N}^T H \mathbf{N} d\Omega_x \end{aligned} \quad (4.12)$$

la matrice de rigidité définie par :

$$\begin{aligned} \mathbf{K}_{uu} &= \mathbf{K}_{\hat{u}\hat{u}} = \int_{\Omega_x} \mathbf{B}^T \boldsymbol{\sigma} d\Omega_x \\ \mathbf{K}_{u\hat{u}} &= \mathbf{K}_{\hat{u}u} = \int_{\Omega_x} \mathbf{B}^T H \boldsymbol{\sigma} d\Omega_x \end{aligned} \quad (4.13)$$

et le vecteur des efforts externes défini par :

$$\begin{aligned} \overrightarrow{F^{ext}} &= \int_{\Omega_x} \rho \mathbf{N}^T \vec{b} d\Omega_x + \int_{\Gamma_x} \mathbf{N}^T \vec{t} d\Gamma_x \\ \overrightarrow{F^{ext}} &= \int_{\Omega_x} \rho \mathbf{N}^T H \vec{b} d\Omega_x + \int_{\Gamma_x} \mathbf{N}^T H \vec{t} d\Gamma_x \end{aligned} \quad (4.14)$$

L'équation (4.11) est intégrée à l'aide du schéma explicite de Chung-Hulbert présenté au paragraphe 3.2 page 46. Selon le nombre de noeuds enrichis utilisés, les tailles des vecteurs portant les champs de déplacement réels  $\vec{u}$  et enrichis  $\vec{\hat{u}}$  diffèrent. Cette particularité a nécessité le développement d'une classe destinée à gérer des tables d'allocation dynamique d'adresses pour l'assemblage des matrices et des vecteurs.

Les développements numériques proposés dans le travail de thèse de I. Nistor [IV] sont complétés par l'introduction d'un critère de propagation dynamique de la fissure. Le but de ce critère est de déterminer quand, et de quelle manière (direction et vitesse) la fissure se propage dans la structure. Plusieurs approches sont possibles, et les propositions de modèles sont nombreuses dans la littérature. Dans l'approche développée par I. Nistor, la détermination de la direction de propagation est basée sur la recherche de la contrainte circonférentielle maximale  $\sigma_{\theta\theta}$  pour les points situés à une distance fixée de la pointe de la fissure comme illustré sur la figure 4.2. La détermination de la longueur d'avancement de la fissure est déterminée par un modèle combinant la notion de contrainte maximale utilisée par Moes et al. [84] avec la notion de modèle cohésif utilisée par Camacho et al. [86].

L'implémentation de la méthode XFEM est effectuée en dérivant de nouvelles classes permettant de définir l'enrichissement des noeuds (classe `XNode`) et les éléments finis enrichis (classe `XElement`). Il est également nécessaire de développer de nouvelles classes permettant notamment l'intégration des champs discontinus sur les éléments partitionnés et la gestion des enrichissements dans la phase d'assemblage des matrices.

## 4.2 Développements d'applications métier

Dans cette section, on présente la réalisation d'applications métier à partir des bibliothèques numériques afin de créer des logiciels spécialisés dans des domaines divers. On présente ici trois applications majeures. Deux d'entre elles font partie des travaux liés aux thèses de L. Menanteau [III] et I. Nistor [IV]. La troisième est plus exploratoire et concerne le développement d'une application dédiée à la modélisation numérique de la coupe des métaux. Ce principe de développement est schématisé sur la figure 4.3.

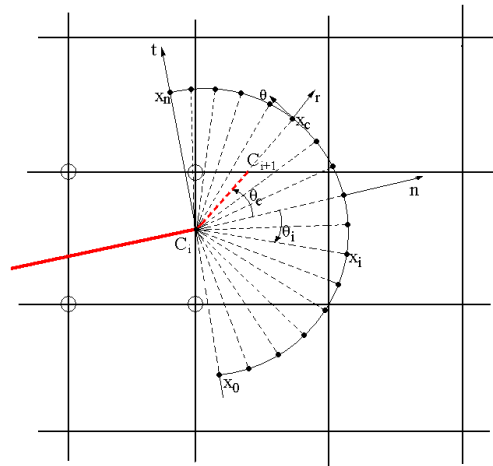


FIG. 4.2 – Direction de propagation pour la fissure

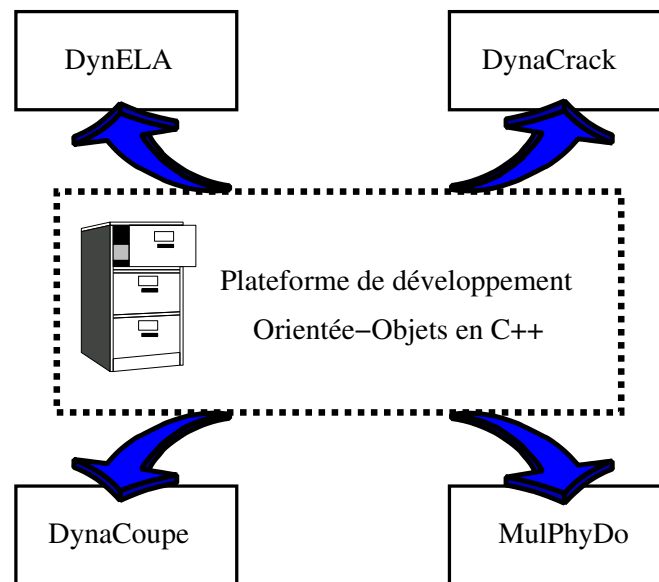


FIG. 4.3 – Développement d'applications métier

### 4.2.1 Application MulPhyDo

L'application MulPhyDo (pour Multi-Physique et Multi-Domaine) a été développée dans le cadre des travaux de thèse de L. Menanteau [III] pour le compte du Laboratoire PEARL. C'est une application métier pour le dimensionnement des convertisseurs de puissance dans les chaînes de transport ferroviaires.

Afin d'illustrer les possibilités du code de calcul MulPhyDo, on présente ici un exemple d'application concernant le calcul électro-thermique d'un sandwich utilisé en électronique de puissance composé d'une puce en silicium brasée sur un premier DBC (*Direct Bonded Copper*) sur une de ses faces, et reliée à un deuxième sur l'autre face par 12 *bumps* (cylindres de connexion brasés de part et d'autre) comme indiqué sur la figure 4.4. La circulation du courant et les conditions aux limites thermiques sont données sur la figure 4.5.

A l'instant  $t = 0$  l'ensemble de la structure est soumise à un potentiel de 0V. A l'instant  $t$ , la surface inférieure de la puce est portée à un potentiel de 100V alors que l'un des cotés de la métallisation supérieure est maintenu à 0V. La différence de potentiel engendre alors un gradient de potentiel et le courant s'écoule dans la structure comme illustré sur la figure 4.6(a).

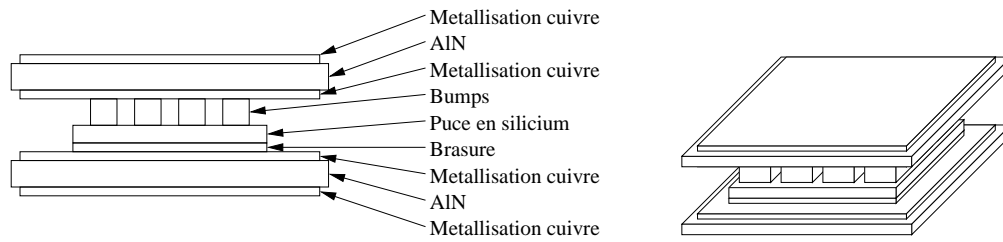


FIG. 4.4 – Assemblage de la puce

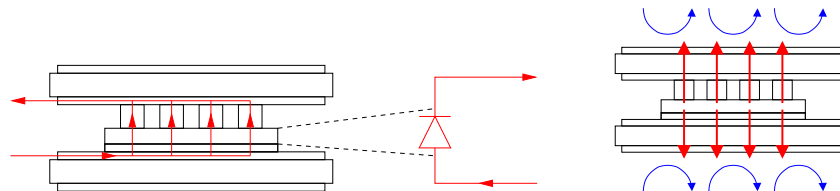
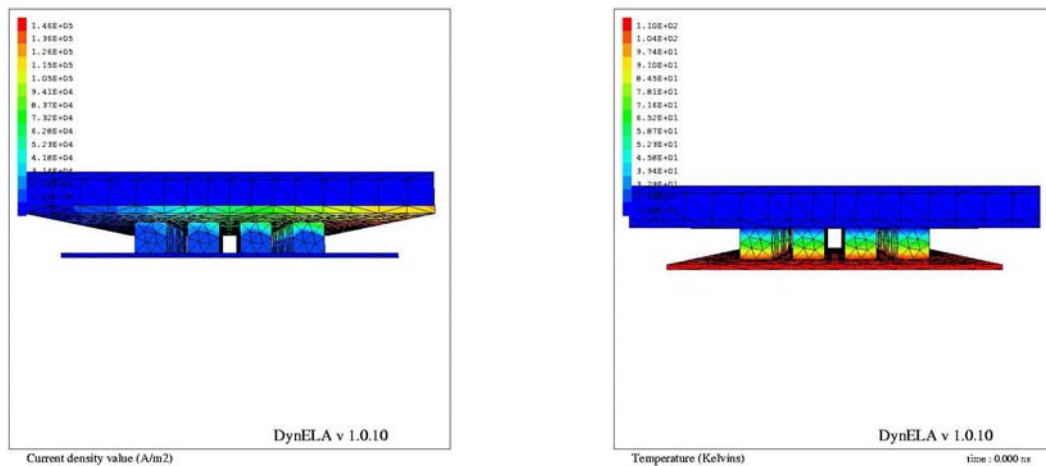


FIG. 4.5 – Conditions aux limites pour la puce

A partir des résultats issus du calcul électrique, le logiciel enchaîne alors automatiquement la phase de calcul thermique. A l'instant initial, la température de la structure est uniforme et sa valeur est de  $20^{\circ}\text{C}$ . La face supérieure de la métallisation supérieure est maintenue à  $20^{\circ}\text{C}$ . Le calcul prend ici en compte les dissipations par effet Joule dans les conducteurs. Le gradient de température obtenu est donné sur la figure 4.6(b).



(a) Densité de courant dans la structure

(b) Températures dans la structure

FIG. 4.6 – Résultats numériques obtenus avec le logiciel MulPhyDo

Actuellement, la version 1.0 est opérationnelle, la version 2.0 est en cours de développement. Les apports de cette nouvelle version concernent les points suivants :

- introduction de la dépendance des propriétés matérielles en fonction de la température,
- prise en compte des couplages entre plus de deux domaines pour un noeud de l'interface,
- possibilité d'utiliser des maillages différents pour des physiques différentes.

### 4.2.2 Application DynaCrack

Des travaux sont en cours concernant l'implémentation numérique de la méthode XFEM sur la plateforme de développement dans le cadre des travaux de thèse de I. Nistor [IV]. L'objectif est de proposer une application spécifique destinée à modéliser le comportement dynamique d'une fissure dans un matériau à l'aide de la méthode XFEM. Afin d'illustrer les possibilités du code DynaCrack développé au cours de ce travail de thèse, on considère l'exemple de validation proposé par Duarte et al. [87] présenté sur la figure 4.7(a). Le chargement est imposé sur la structure par une fonction de type Heaviside avec une amplitude de  $63750N$ . Le matériau est élastique linéaire avec  $E = 200GPa$  et  $\nu = 0.3$ . Sur la figure 4.7(b) montrant la répartition des contraintes de von Mises pour  $t = 1.2ms$  on peut observer le champ de contraintes en pointe de fissure.

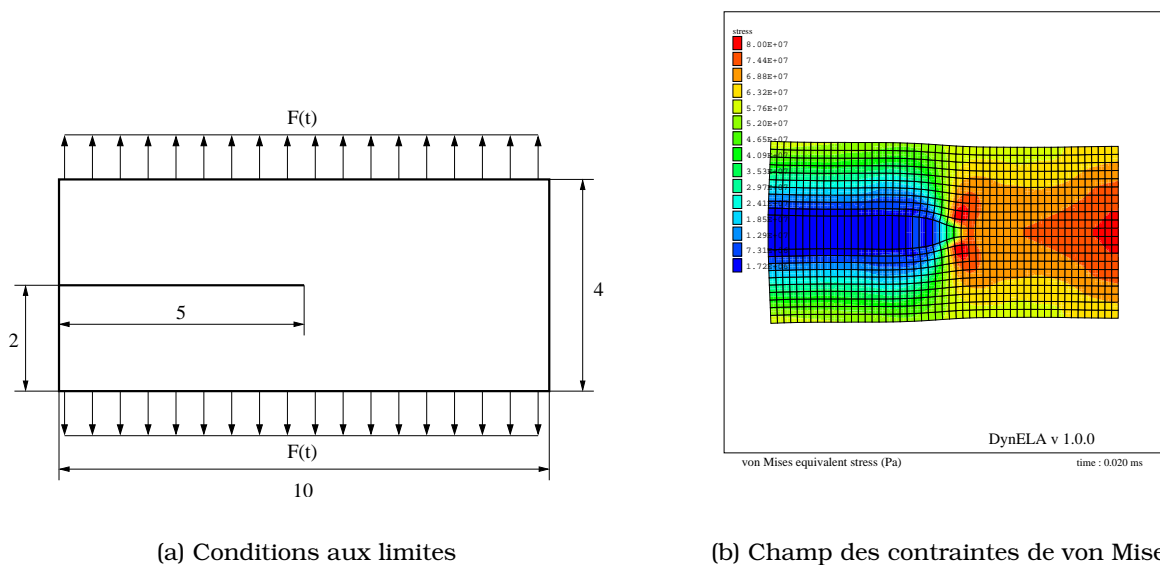


FIG. 4.7 – Résultats numériques obtenus avec le logiciel DynaCrack

L'utilisation de cette approche dans le cas de la modélisation numérique du comportement de structures soumises à des impact et des crash nécessite l'introduction d'un modèle d'endommagement dans la loi de comportement. Lorsque ce critère d'endommagement est violé, on est alors en mesure d'initier la phase de fissuration de la structure. Les perspectives de développements numériques concernent un certain nombre de points qu'il convient de maîtriser afin de rendre le code pleinement opérationnel :

- l'introduction d'une loi d'endommagement permettant l'initiation d'une fissure dans un milieu initialement vierge,
- la prise en compte d'une loi de comportement non-linéaire hypo-élastique,
- la gestion de plusieurs fissures dans le matériau et notamment les problèmes liés à la présence de plusieurs fissures dans le même élément.

### 4.2.3 Application DynaCoupe

L'introduction d'algorithmes de gestion du contact efficaces et la prise en compte de l'endommagement dans le code de calcul **DynELA** doivent permettre de jeter les bases de développement d'une version capable de modéliser la simulation numérique de la coupe des métaux

telle qu'elle a été définie dans le chapitre 1. Dans cette perspective, la plateforme de développement actuelle a été retenue comme base de développement dans le cadre de la mise en place du réseau OPTIMUS par proposition d'un sujet de thèse : "Elaboration d'un outil de simulation numérique intégrant les propositions des membres du réseau OPTIMUS". Dans le cadre du développement de cette application spécifique dédiée à la simulation numérique de la coupe des métaux, un certain nombre de développements sont en cours de réalisation. Les principaux développements ci-dessous sont donc prévus :

- Concernant le comportement volumique, des travaux sont en cours de réalisation concernant la prise en compte de l'endommagement dans le code de calcul **DynELA** de manière similaire à ce qui est implémenté dans le code de calcul Abaqus. Ces travaux peuvent être étendus via la méthode XFEM afin de prendre en compte des phénomènes de fissuration du matériau usiné selon les conditions de coupe utilisées.
- Le point faible, actuellement, du code de calcul **DynELA** concerne la prise en compte du contact puisque seul un algorithme de contact de type solide déformable/indéformable est pris en compte. S'il semble suffisant dans la mesure où l'outil de coupe est généralement assimilé à un solide indéformable, des travaux sont néanmoins en cours de réalisation pour l'implémentation de la gestion du contact entre deux solides déformables.

*Ces travaux ont fait l'objet de 2 publications internationales (soumises) et 2 participations à des congrès.*





# Conclusions, perspectives et projet de recherche

Ce mémoire d'Habilitation à Diriger les Recherches décrit mes activités effectuées au cours de la période de 1992 à 2005 autour du thème central des Grandes Transformations Thermomécaniques Rapides. Il permet de dresser un bilan de l'état actuel d'avancement de ces travaux, à partir duquel, les perspectives incluses dans mon projet de recherche fournissent un ensemble de pistes pour le développement de la plateforme de prototypage virtuel en Grandes Transformations Thermomécaniques Rapides.

## Bilan des activités de recherche

Le premier point de ce bilan concerne la création du groupe de recherche en Grandes Transformations Thermomécaniques Rapides qui a été mis en place à la fin des années 1990 afin de porter une nouvelle voie de recherche pour le laboratoire. Ce groupe de recherche, composé de 4 personnels permanents du laboratoire (S. Caperaa, R. Rakotomalala, O. Dalverny et O. Pantalé), s'intéresse à la modélisation numérique et à l'identification expérimentale des lois de comportement pour des structures soumises à de fortes sollicitations thermomécaniques rapides. Ces activités sont orientées d'une manière générale sur :

- la modélisation numérique des processus de mise en forme par déformation plastique et d'usinage par enlèvement de matière,
- l'identification des lois de comportement pour des matériaux soumis à de fortes sollicitations thermomécaniques rapides,
- le développement d'applications de calcul et d'utilitaires dans le cadre de la simulation numérique par éléments finis.

L'objectif principal des travaux menés par ce groupe de recherche est d'apporter une contribution dans le domaine de la simulation numérique des phénomènes thermo-mécaniques liés à la dynamique rapide. Les études bibliographiques et les travaux réalisés ont montré la place importante que tient la formulation des lois de comportement dans la modélisation numérique de ces phénomènes. D'une manière générale, l'identification et la prise en compte numérique des lois de comportement non-linéaires anélastiques constitue donc une des principales pré-occupations de ce groupe de recherche.

Le deuxième point concerne les activités liées à la modélisation numérique de la coupe des métaux. Comme présenté dans le chapitre 1, au cours de la période 1991-2001, plusieurs modèles numériques de la coupe des métaux ont été développés au sein du Laboratoire. Ces divers travaux ont permis de proposer des modèles novateurs de simulation numérique de la coupe orthogonale et oblique 3D permanents et transitoires, ainsi que l'application au fraisage en coupe discontinue transitoire. L'ensemble des modèles numériques réalisés utilise une approche Euléro-Lagrangienne Arbitraire pour la description des mouvements. Le développement des modèles numériques transitoires utilise un critère d'endommagement global

basé sur l'approche proposée par Johnson et Cook permettant l'élaboration de modèles comportementaux prédictifs de la coupe. L'ensemble de ces travaux contribue à l'avancement des recherches dans le domaine de la coupe des métaux et plus particulièrement l'étude de la formation du copeau lors d'une opération d'usinage. Les travaux réalisés ont montré les limites liées à l'utilisation de codes de calcul commerciaux tels que Radioss et Abaqus/Explicit. C'est notamment à partir de ce constat que les activités liées au développement d'un nouveau code de calcul en dynamique rapide ont été initiées.

Concernant les activités expérimentales, comme indiqué dans le chapitre 2, le premier point notable concerne la création du laboratoire d'identification expérimentale du comportement des matériaux soumis à de fortes sollicitations thermomécaniques rapides. Cette notion de création englobe à la fois l'acquisition des moyens matériels et le développement de méthodes expérimentales et de moyens logiciels adaptés. Dans la suite de cette mise en place, plusieurs essais ont été mis au point au cours de la période 2001-2005 : essai de Taylor, essai de traction, essai de cisaillement, essai de frottement... Une procédure et un logiciel d'identification ont été développés, notamment au cours des travaux de DEA de I. Nistor et de la thèse de C. Sattouf. On est désormais en mesure d'identifier le comportement dynamique d'un matériau à partir d'un ou plusieurs types d'essais expérimentaux.

Les activités liées au développement numérique du code de calcul **DynELA** décrit au chapitre 3 et les travaux réalisés à partir de la plateforme de développement Orientée-Objets décrits au chapitre 4 constituent la plus grande partie de l'effort de recherche effectué personnellement au cours de ces dernières années. Les apports liés à ces travaux sont multiples :

- Le laboratoire dispose désormais d'un code de calcul ouvert dédié à la simulation numérique des problèmes en grandes transformations. Ce code sert notamment de support pour le développement de nouveaux algorithmes de calcul (intégration, XFEM, DDM, travaux de parallélisation...).
- Au delà de l'utilisation dans le cadre des travaux de recherche en dynamique rapide, ce code de calcul est également utilisé au niveau du laboratoire d'impact afin de simuler le comportement des essais dans la procédure d'identification. Il est également utilisé par des étudiants dans le cadre de leur formation pédagogique en séances de Travaux Pratiques.
- Au travers de ces développements, une expertise a été mise en place concernant la simulation numérique en grandes transformations, les formulations hypo-élastiques et leur prise en compte dans les codes de calcul. Une partie de ces connaissances est transmise au étudiants en M2R dans le cadre du cours "Grandes Transformations Thermomécaniques Rapides".

*Globalement, l'ensemble de ces travaux de recherche a donné lieu à la parution de 10 publications internationales, la participation à 22 congrès et la soutenance de 5 thèses de doctorat.*

## Perspectives et projet de recherche

Concernant les perspectives et le projet de recherche, l'idée centrale autour de laquelle s'articulent les travaux à réaliser est le développement d'une plateforme de prototypage virtuel pour la simulation numérique en Grandes Transformations. Sous cette dénomination, on peut regrouper les divers besoins liés à la simulation numérique des structures, faisant intervenir les approches expérimentale et numérique qui sont dans la continuité des travaux réalisés à l'heure actuelle au sein du laboratoire.

Toute simulation numérique est effectuée sur une structure physique, elle nécessite donc une phase d'identification du comportement des matériaux constitutifs de cette structure lorsque elle est soumise à de fortes sollicitations thermomécaniques rapides. Les efforts réalisés dans le cadre de la mise en place du laboratoire d'identification doivent donc être poursuivis. Plusieurs voies exploratoires sont proposées dans ce projet de recherche.

- Le premier point concerne le développement de nouveaux essais expérimentaux et des procédures d'identification associées à ces essais. Comme indiqué au paragraphe 2.2.2.5, l'essai de rupture dynamique permettant d'étudier les conditions d'initiation et de propagation de fissures au sein du matériau est actuellement en cours de développement au laboratoire.
- Actuellement, les travaux d'identification réalisés concernent principalement deux matériaux métalliques : l'acier 42CrMo4 et l'aluminium 2017. Dans cette continuité, et dans le cadre de la mise en place du réseau OPTIMUS, le groupe de recherche propose deux sujets de thèse en cotutelle avec le LMT de l'ENS Cachan et le LPMI de l'ENSAM concernant l'identification du comportement des aciers 42CrMo4 et du 316L. D'une manière similaire dans le cadre d'une thèse en cotutelle avec l'ESST de Tunis, des travaux d'identification doivent être réalisés afin de simuler numériquement une opération de mise forme de tôles par emboutissage.
- Concernant le dispositif expérimental, des travaux d'amélioration sont à réaliser afin de se prémunir de l'influence du sabot lors du choc. Deux alternatives sont possibles : la prise en compte du comportement du sabot dans le modèle numérique ou la séparation physique du sabot et du projectile avant l'impact. La deuxième voie, si elle est plus délicate à mettre en oeuvre d'un point de vue physique reste bien entendue préférable.

Concernant les activités liées à la modélisation numérique de la coupe des métaux, l'implication du groupe de recherche a été suspendue momentanément à la fin du travail de thèse de J. L. Bacaria, en 2001. Le bilan dressé à cette époque ayant montré que le développement de modèles numériques plus évolués était limité par l'utilisation du code de calcul Abaqus/Explicit, il a été décidé de suspendre ces activités de recherche afin de recentrer les efforts sur l'identification des lois de comportement et le développement du code de calcul **DynELA**. A l'heure actuelle, et compte tenu des évolutions du code de calcul Abaqus/Explicit, sauf opportunité spéciale, cette activité de modélisation de la coupe des métaux n'est pour l'instant pas encore d'actualité. Au vu de l'orientation actuelle des travaux du groupe de recherche, la modélisation numérique de la coupe des métaux nécessitant des moyens intellectuels relativement importants (approche numérique + validation expérimentale + ...), dans l'idée d'une reprise de cette activité au sein du laboratoire, les efforts fournis seraient concentrés principalement sur l'aspect modélisation numérique. La validation expérimentale serait préférentiellement réalisée par un tiers, comme dans le cadre du sujet de thèse sur la mise en forme de tôles proposé en cotutelle avec l'ESST.

Dans le cadre des activités liées au développement du code de calcul **DynELA**, beaucoup d'axes de travail sont ouverts, tant du point de vue modélisation numérique, implémentation informatique, utilisation du code, création de partenariats, etc... A l'issue des travaux actuellement réalisés, comme indiqué au chapitres 3 et 4, le laboratoire dispose d'une plateforme de développement Orientée-Objets entièrement ouverte et maîtrisée qui peut servir de base de travail pour de nombreux développements.

- Le premier axe de travail concerne le développement du code de calcul **DynELA**, au travers de l'amélioration des performances et des capacités du code. Comme indiqué dans le chapitre 3, à l'heure actuelle, la version 1.1 du code de calcul est disponible et les améliorations principales concernent le développement de lois de comportement visco-plastiques avec endommagement et la prise en compte du contact entre deux solides déformables. D'un point

de vue numérique, d'autres travaux peuvent être entrepris comme le développement d'éléments de structures (poutres et coques), le développement de solveurs implicites ou la prise en compte de conditions aux limites thermiques.

- Un autre axe de travail relativement important concerne la performance du code en matière de rapidité d'exécution. Comme indiqué au chapitre 3, actuellement, une version parallèle basée sur l'utilisation de la librairie OpenMP dédiée à une machine de type SMP a été développée. L'idée majeure issue de ces travaux, permettant de développer une application plus performante, consiste à utiliser une approche hybride MPI/OpenMP dans le but de créer une version parallèle fonctionnant sur un réseau de machines distribuées de type *cluster*. Cette nouvelle approche nécessite des modifications dans la structure du code, l'implémentation d'une méthode de décomposition en sous-domaines, et bien sûr, l'acquisition d'une architecture physique de calcul dédiée.
- Le troisième axe de travail dans le cadre des applications numériques concerne les logiciels construits à partir de la plateforme de développement Orientée-Objets de **DynELA** et présentés dans le chapitre 4. Ce sont les applications DynaCoupe, DynaCrack et MulPhyDo. La première fait l'objet actuellement d'une proposition de thèse dans le cadre du réseau OPTIMUS, la seconde fait partie intégrante des travaux de thèse réalisés actuellement par I. Nistor, alors que la troisième fait l'objet de développements complémentaires afin d'améliorer les performances et les capacités de calcul dans le domaine de la simulation des convertisseurs de puissance.
- Un point très important auquel il convient de porter une attention toute particulière concerne la diffusion du code de calcul **DynELA** et la création de partenariats avec des Universités et des entreprises désireuses de participer à l'effort de développement. Le code de calcul dans sa version actuelle est utilisé au sein du laboratoire par des personnels et des étudiants en M2R dans le cadre de leur formation. Il est donc important de faire connaître l'existence de ce code de calcul auprès de partenaires et de mettre en place une politique de coopération scientifique dans le domaine des Grandes Transformations. Dans cette optique, un site web est actuellement en cours de développement afin d'offrir un lieu d'échange pour les utilisateurs et les développeurs du code. C'est en diffusant le plus largement possible, dans un contexte de collaboration, un programme que l'on est en mesure de le faire vivre et donc évoluer.

# Bibliographie

## Thèses encadrées

- [I] J. L. Bacaria. *Un modèle comportemental et transitoire pour la coupe des métaux*. INP de Toulouse, 13 novembre 2001.
- [II] C. Sattouf. *Caractérisation en dynamique rapide du comportement de matériaux utilisés en aéronautique*. INP de Toulouse, 30 juin 2003.
- [III] L. Menanteau. *Développement d'un module de prototypage virtuel multi-physique, multi-domaine et multitemps. Application aux convertisseurs de puissance*. INP de Toulouse, 25 octobre 2004.
- [IV] I. Nistor. *Identification expérimentale et simulation numérique de l'endommagement en dynamique rapide : Application aux structures aéronautiques*. INP de Toulouse, 2005.

## Mémoires de DEA encadrés

- [i] B. Peressini. *Description Euléro-Lagrangienne-Arbitraire : développement d'un code de calcul élastique*. DEA de l'université de Bordeaux I, 1997.
- [ii] B. Bonnac. *Modélisation et implémentation d'une loi de comportement du béton*. DEA de l'université de Bordeaux I, 1998.
- [iii] C. Dalos. *Implémentation numérique d'un modèle d'endommagement dans un code de dynamique rapide*. DEA de l'université de Bordeaux I, 1999.
- [iv] P. Laffont. *Implémentation de lois de comportement élastoplastiques en dynamique rapide dans un code de calcul explicite*. DEA de l'université de Bordeaux I, 1999.
- [v] P. Louvet. *Développement et mise au point d'un essai d'impact pour la caractérisation du comportement des matériaux soumis à des sollicitations à grande vitesse*. DEA de l'INP de Toulouse, 2001.
- [vi] I. Nistor. *Identification du comportement dynamique à partir des résultats d'essai d'impact de Taylor direct et symétrique*. DEA de l'INP de Toulouse, 2002.
- [vii] R. Maran. *Développement d'un essai de traction dynamique pour la caractérisation du comportement des matériaux lors d'impacts*. DEA de l'INP de Toulouse, 2003
- [viii] E. Gorce. *Développement d'une méthodologie d'identification des paramètres constitutifs d'un matériau soumis à de grandes sollicitations thermomécaniques*. DEA de l'INP de Toulouse, 2004
- [ix] B. Aubert. *Modélisation de la rupture dynamique des matériaux fragiles par la méthode des éléments finis étendue (XFEM)*. M2R de l'INP de Toulouse, 2005

## Bibliographie

- [1] O. Pantalé. *Modélisation et Simulation Tridimensionnelles de la Coupe des Métaux*. PhD thesis, Université de Bordeaux I, 1996.
- [2] T. Shirakashi and E. Usui. Simulation analysis of orthogonal metal cutting mechanism. In *Proceedings of the International Conference on Production Engineering*, pages 535–540, 1974.
- [3] K. Iwata, K. Osakada, and Y. Terasaka. Process modeling of orthogonal cutting by the rigid-plastic finite element method. *Transactions of the ASME*, 106 :132–138, 1984.
- [4] J. T. Carroll and J. S. Strenkowski. Finite element models of orthogonal cutting with application to single point diamond turning. *International Journal of Mechanical Sciences*, 30(12) :899–920, 1988.
- [5] A. J. M. Shih, S. Chandrasekar, and H. T. Y. Yang. Finite element simulation of metal cutting process with strain rate and temperature effects. *Fundamental Issues of Machining, ASME PED 43*, pages 11–24, 1990.
- [6] G. S. Sekhon and J-L. Chenot. Numerical simulation of continuous chip formation during non steady orthogonal cutting. *Engineering computations*, 10 :31–48, 1993.
- [7] T. Obikawa, H. Sasahara, T. Shirakashi, and E. Usui. Application of computational machining method to discontinuous chip formation. *Journal of Manufacturing Science and Engineering*, 119 :667–674, 1997.
- [8] Z. C. Lin and W. C. Pan. A thermoelastic-plastic large deformation model for orthogonal cutting with tool flank wear. i : Computational procedures. *International Journal of Mechanical Sciences*, 35(10) :829–840, 1993.
- [9] T. D. Marusich and M. Ortiz. Modelling and simulation of high-speed machining. *International Journal for Numerical Methods in Engineering*, 38 :3675–3694, 1995.
- [10] Y. Altintas. Modeling approaches and software for predicting the performance of milling operations at mal-ubc. In *Machining science and technology*, volume 4, pages 445–478, 2000.
- [11] P. Joyot. *Modélisation Numérique et Expérimentale de l'enlèvement de Matière. Application à la Coupe Orthogonale*. PhD thesis, Université Bordeaux I, 1994.
- [12] J. Donea, A. Huerta, J. P. Ponthot, and A. Rodriguez-Ferran. Arbitrary lagrangian-eulerian methods. In *Encyclopedia of Computational Mechanics - Fundamentals*, volume 1, page Chap 14, 2004.
- [13] L. Olovsson, L. Nilsson, and K. Simonsson. An ale formulation for the solution of two-dimensional metal cutting problems. *Computers and Structures*, 72 :497–507, 1999.
- [14] M. Movahhedy, M. S. Gadala, and Y. Altintas. Simulation of the orthogonal metal cutting process using an arbitrary lagrangian-eulerian finite element method. *Journal of Materials Processing Technology*, 103 :267–275, 2000.
- [15] R. Johnson and W. K. Cook. A constitutive model and data for metals subjected to large strains; high strain rates and high temperatures. *7th International Symposium on Ballistics; The Hague*, pages 541–547, 1983.
- [16] P. Joyot, R. Rakotomalala, O. Pantalé, M. Touratier, and N. Hakem. Un modèle prédictif de coupe orthogonale basé sur une formulation arbitraire lagrangienne eulérienne. In *Journée Procédés d'usinage - PRIMECA*, 19 mai 1994.
- [17] O. Pantalé, R. Rakotomalala, M. Touratier, and S. Caperaa. Un modèle numérique d'usinage : Analyse de l'effet du frottement et de l'usure. In *Journée PRIMECA*, 1996.
- [18] P. Joyot, R. Rakotomalala, O. Pantalé, M. Touratier, and N. Hakem. A numerical simulation of steady state metal cutting. *Journal for Mechanical Engineers*, 212(part C) :331–341, 1998.

- [19] O. Pantalé, P. Joyot, R. Rakotomalala, M. Touratier, and N. Hakem. Présentation d'un modèle tridimensionnel a. l. e de coupe orthogonale. In *Usinage des Matériaux à Caractéristiques Mécaniques élevées E. T. C. A*, 21 juin 1994.
- [20] O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. Modélisation tridimensionnelle de la coupe orthogonale. In Hermès, editor, *2ème Colloque en Calcul de Structures - Giens*, volume 2 of ISBN 2-86601-471-5, pages 597–602, 16-19 mai 1995.
- [21] O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. An ale tridimensionnal numerical simulation of orthogonal cutting process. In *First International Radioss Users Conference - SOPHIA ANTIPOLIS*, 19 juin 1995.
- [22] O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. Modélisation tridimensionnelle de la coupe orthogonale. In *Journée PRIMECA Procédés d'Usinage - CER ENSAM - 71250 Centre de Cluny*, 3-5 avril 1995.
- [23] A. A. O. Tay. A review of methods of calculating machining temperature. *Journal of Materials Processing Technology*, 36 :225–257, 1993.
- [24] O. Pantalé, R. Rakotomalala, M. Touratier, and N. Hakem. A three-dimensional numerical model of orthogonal and oblique metal cutting processes. In ASME, editor, *PD-Vol. 75 ; Engineering Systems Design and Analysis*, volume 3, pages 199–206, 1-4 juillet 1996.
- [25] O. Pantalé, R. Rakotomalala, and M. Touratier. An ale three-dimensional model of orthogonal and oblique metal cutting processes. *International Journal of Forming Processes*, 1(3) :371–388, 09 1998.
- [26] O. Pantalé, R. Rakotomalala, and M. Touratier. Simulation of chip formation in machining. In Neat Press, editor, *Proceedings of plasticity'99 (Cancun)*, pages 397–400, january 1999.
- [27] G. R. Johnson and W. H. Cook. Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures. *Engineering Fracture Mechanics*, 21(1) :31–48, 1985.
- [28] J. W. Hancock and A. C. Mackenzie. On the mechanisms of ductile failure in high-strength steels subjected to multi-axial stress-states. *Journal of Mechanical Physics of Solids*, 24 :147–169, 1976.
- [29] J. L. Bacaria, O. Dalverny, O. Pantalé, R. Rakotomalala, and S. Caperaa. 2d and 3d numerical models of metal cutting with damage effects. In *ECCOMAS 2000*, 2000.
- [30] J. L. Bacaria, O. Dalverny, O. Pantalé, and R. Rakotomalala. Transient numerical models of metal cutting using the johnson-cook's rupture criterion. *International Journal of Forming Processes*, 5(1) :53–70, 2002.
- [31] J. L. Bacaria, O. Dalverny, O. Pantalé, R. Rakotomalala, and S. Caperaa. Transient numerical models of discontinuous chip formation based on a damaged effect. In A.M. Habrakem, editor, *4th Int. ESAFORM Conf. on Material Forming*, volume 2, pages 611–614, Liège Belgium, 2001.
- [32] O. Pantalé, J. L. Bacaria, O. Dalverny, R. Rakotomalala, and S. Caperaa. 2d and 3d numerical models of metal cutting with damage effects. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41) :4383–4399, 2004.
- [33] O. Pantalé and S. Caperaa. Développement d'un code de calcul explicite en grandes transformations : Application à la coupe des métaux. In *Séminaire Optimus*, 2004.
- [34] J. C. Simo and T. J. R. Hughes. *Computational inelasticity*. Springer, 1998.
- [35] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Element for Continua and Structures*. Wiley, 2000.
- [36] O. Pantalé. *Manuel du code de calcul DynELA v. 1.0*. Laboratoire LGP ENI Tarbes, Av d'Azereix 65016 Tarbes, France, 2004.

- [37] J. P. Ponthot. *Traitement unifié de la mécanique des milieux continus en grandes transformations par la méthode des éléments finis*. PhD thesis, Faculté des Sciences Appliquées - Université de Liege, 1997.
- [38] P. Bridgman. *The physic of high pressure*. Bell and Sons, London, 1949.
- [39] D. M. Norris, J. R. B. Morran, J. K. Scudder, and D. F. Quinones. A computer simulation of the tension test. *Journal of Mechanical Physics of Solids*, 26 :1–19, 1978.
- [40] F. J. Zerilli and R. W. Armstrong. Dislocation mechanics based constitutive relations for material dynamics calculations. *Journal of Applied Physics*, 61(5) :1816–1825, March 1987.
- [41] Abaqus. *Reference Manual*. Hibbitt, Karlsson and Sorensen Inc., 100 Medway Street, Providence, RI, 02906-4402, USA, 1989.
- [42] T. J. Holmquist and G. R. Johnson. Determination of constants and comparison of results for various constitutive models. *Journal de Physique*, 1 :853–860, October 1991.
- [43] W. K. Rule and S. E. Jones. A revised form for the johnson-cook strength model. *International Journal of Impact Engineering*, 21(8) :609–624, 1998.
- [44] O. Pantalé. *Manuel utilisateur du logiciel imageAnalyser v. 1.0*. Laboratoire LGP ENI Tarbes, 47 av d'Azereix - 65016 Tarbes, 2002.
- [45] R. Dormeival, J. P. Ansart, J. Sorel, F. Motta, and P. Calleja. Le 'taylor-test' ou 'essai d'impact sur cible rigide'. In *Cinquième Colloque Mécanique et Métallurgie de Tarbes*, 1984.
- [46] J. E. Field, S. M. Walley, N. K. Bourne, and J. M. Huntley. Experimental methods at high rates of strain. *Journal de Physique IV, Colloque C8*, 4 :3–22, septembre 1994.
- [47] O. Pantalé, I. Nistor, and S. Caperaa. Identification et modélisation du comportement des matériaux métalliques sous sollicitations dynamiques. In Military Technical Academy, editor, *30th Internationally attended scientific conference of the military technical academy*, ISBN 973-640-012-3, Bucharest, 2003.
- [48] M. A. Meyers, V. F. Nesterenko, J. C. LaSalvia, and Q. Xue. Shear localization in dynamic deformation of materials : microstructural evolution and self-organization. *Materials Science and Engineering A*, 317(1-2) :204–225, 2001.
- [49] H. Abichou, O. Pantalé, I. Nistor, O. Dalverny, and S. Caperaa. Identification of metallic material behaviors under high-velocity impact : A new tensile test. In *15th Technical Meeting DYMAT*, Metz, 1-2 June 2004.
- [50] O. Pantalé, I. Nistor, O. Dalverny, E. Gorce, and S. Caperaa. Caractérisation du comportement dynamique des matériaux à partir d'essais d'impact. In *Séminaire Optimus*, Bordeaux, 05 2004.
- [51] J.L. Sturges and B.N. Cole. The flying wedge : A method for high-strain-rate tensile testing. part 1. reasons for its development and general description. *International Journal of Impact Engineering*, 25 :251–264, 2001.
- [52] O. Dalverny, S. Caperaa, O. Pantalé, and C. Sattouf. Identification de lois constitutives et de lois de frottement adaptées aux grandes vitesses de sollicitation. In *Journal de Physique IV*, volume 12, pages 275–281, 2002.
- [53] I. Nistor, O. Pantalé, S. Caperaa, and C. Sattouf. A new dynamic test for the identification of high speed friction law using a gas-gun device. In *Journal de Physique IV*, volume 110, pages 519–524, 2003.
- [54] I. Nistor, O. Pantalé, S. Caperaa, and C. Sattouf. Identification of a dynamic viscoplastic flow law using a combined levenberg-marquardt and monte-carlo algorithm. In CIMNE, editor, *VII International Conference on Computational Plasticity, COMPLAS 2003*, Barcelona, 4 2003.



- [55] G. M. Hulbert and J. Chung. Explicit time integration for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics and Engineering*, 137 :175–188, 1996.
- [56] D. J. Benson. An efficient accurate simple ale method for nonlinear finite element programs. *International Journal for Numerical Methods in Engineering*, 72 :305–350, 1989.
- [57] G. R. Miller. An object oriented approach to structural analysis and design. *Computers and Structures*, 40(1) :75–82, 1991.
- [58] R. I. Mackie. Object oriented programming of the finite element method. *International Journal for Numerical Methods in Engineering*, 35 :425–436, 1992.
- [59] G. P. Nikishkov. Performance of a finite element code written in java. *Advances in Computational Engineering and Sciences*, 1 :264–269, 2000.
- [60] M. Daehlen and A. Tveito. *Numerical methods and software tools in industrial mathematics*. Birkhauser, 1997.
- [61] S. Haney and J. Crotinger. How templates enables high-performance scientific computing in c++. *Computing in Science and Engineering*, 1(4) :66–72, jul/aug 1999.
- [62] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. The Addison-Wesley Object Technology Series, 1999.
- [63] R. Pozo. Template numerical toolkit for linear algebra : High performance programming with c++ and the standard pemplate library. Internet, 0.
- [64] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Philadelphia : Society for industrial and applied mathematics, 3rd edition, 1999.
- [65] C. Lawson, R. Hanson, D. Kincaid, and F. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Trans. math. Software*, 5 :308–329, 1979.
- [66] G. Swinnen. *Apprendre à programmer avec Python*. Number ISBN : 2-84177-294-2. O'REILLY, 2003.
- [67] A. Pletzer and J. C. Mollis. Ellipt2d : A flexible finite element code written in python. Technical Report PPPL-3552, U.S. department of energy, PPPL, Princeton University, Princeton, New Jersey, 2001.
- [68] T. Mason, J. Levine, and D. Brown. *lex & yacc*. Number 1-56592-000-7. O'Reilly, 2nd edition, October 1992.
- [69] A. Huerta and F. Casadei. New ale applications in non-linear fast-transient solid dynamics. *Engineering Computations : Int J for Computer-Aided Engineering*, 11 :317–345, 1994.
- [70] J. P. Ponthot. Unified stress update algorithms for the numerical simulation of large deformation elasto-plastic and visco-plastic processes. *International Journal of Plasticity*, 18 :91–126, 2002.
- [71] W. K. Liu, H. Chang, J. S. Chen, and T. Belytschko. Arbitrary lagrangian-eulerian petrov-galerkin finite elements for nonlinear continua. *Computer Methods in Applied Mechanics and Engineering*, 68 :259–310, 1988.
- [72] E. Anderheggen and J. F. Renau-Munoz. A parallel explicit solver for simulating impact penetration of a three-dimensional body into a solid substrate. *Advances in Engineering Software*, 31 :901–911, 2000.
- [73] K. Brown, S. Attaway, S. Plimpton, and B. Hendrickson. Parallel strategies for crash and impact simulations. *Computer Methods in Applied Mechanics and Engineering*, 184 :375–390, 2000.
- [74] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon. *Parallel Programming in OpenMP*. Academic press, 2001.

- [75] R. Jia and B. Sundén. Parallelization of a multi-blocked cfd code via three strategies for fluid flow and heat transfer analysis. *Computers & Fluids*, 33 :57–80, 2004.
- [76] E. L. Turner and H. Hu. A parallel cfd rotor code using openmp. *Advances in Engineering Software*, 32 :665–671, 2001.
- [77] P. Rus, B. Stok, and N. Mole. Parallel computing with load balancing on heterogeneous distributed systems. *Advances in Engineering Software*, 34 :185–201, 2004.
- [78] J. Gustafson. Fixed time, tiered memory, and superlinear speedup. In *Proceedings of the Fifth Distributed Memory Computing Conference (DMCC5)*, October 1990.
- [79] O. Pantalé. Parallelization of an object-oriented fem dynamics code : Influence of the strategies on the speedup. *Advances in Engineering Software*, 36(6) :361–373, 2005.
- [80] O. Pantalé and S. Caperaa. Strategies for a parallel 3d fem code : Application to impact and crash problems in structural mechanics. In *Coupled Problems 2005*, Santorini Island, 25-28 may 2005.
- [81] C. Farhat. A method of finite element tearing end interconnecting and its parallel solution algorithm. *International Journal of Numerical Methods in Engineering*, 32 :1205–1227, 1991.
- [82] L. Menanteau, S. Caperaa, and O. Pantalé. A multigrid method for the thermomechanical behaviour simulation applicated to power electronics converter. In CIMNE, editor, *VII International Conference on Computational Plasticity, COMPLAS 2003*, Barcelona, 4 2003.
- [83] L. Menanteau, O. Pantalé, and S. Caperaa. A coupled electro-thermo-mechanical fem code for large scale problems including multi-domain and multiple time-step aspects. In *Coupled Problems 2005*, Santorini Island, 25-28 may 2005.
- [84] N. Moes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46 :131–150, 1999.
- [85] J. Dolbow, N. Moes, and T. Belytschko. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design*, 36 :235–260, 2000.
- [86] G. T. Camacho and M. Ortiz. Computational modeling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33 :2899–2938, 1996.
- [87] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydlo. A generalized finite element method for the simulation of three-dimensionnal dynamic crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 30 :541–564, 2001.