Application de la différenciation automatique pour l'identification, l'optimisation et l'étude de sensibilité dans quelques problèmes mécaniques

Imen Elkhaldi Elkhaldi-Mkaouar

▶ **To cite this version:**

Imen Elkhaldi Elkhaldi-Mkaouar. Application de la différenciation automatique pour l'identification, l'optimisation et l'étude de sensibilité dans quelques problèmes mécaniques. Autre. Université de Lorraine, 2012. Français. NNT : 2012LORR0041 . tel-01749210

HAL Id: tel-01749210
https://hal.univ-lorraine.fr/tel-01749210

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

THÈSE

Application de la Différentiation Automatique pour l'identification, l'optimisation et l'étude de sensibilité dans quelques problèmes mécaniques

présentée et soutenue publiquement le 12/03/2012 à 14h dans la Salle Ferrari

pour l'obtention du

Doctorat de l'Université de Lorraine

(Dicipline: Génie Mécanique et Énergétique)

par

Imen ELKHALDI-MKAOUAR

Composition du jury

Rapporteurs :	Laurent TABOUROT	Polytech'Annecy Chambéry
	Loïc DARIDON	LMGC, Montpellier
Examinateurs :	El Mostafa DAYA	LEM3, Metz
	Gérard Ferron	LEM3, Metz
	Paolo VANNUCCI	Université de Versailles-Saint Quentin en Yvelines, Versailles
	Laurent DURRENBERGER	ArcelorMittal, Maizières-les-Metz
Directrice :	Isabelle CHARPENTIER	LEM3, Metz

Laboratoire d'Etudes des Microstructures et de Mécanique des Matériaux – UMR CNRS 7239
Ecole Doctorale EMMA (Énergitique Mécanique MAtériaux)

Mis en page avec la classe thloria.

Remerciements

Les mots ne me suffisent pas pour exprimer ma gratitude et ma reconnaissance envers toi Isabelle. Sans tes encouragements, ton dévouement, ta protection, ta compréhension, ta disponibilité et ton sens de responsabilité je ne serai pas aujourd'hui devant mon poste entrain d'écrire mes remerciements de fin de thèse. Un grand merci du fond du cœur.

Je tiens également à adresser mes sincères remerciements à Messieurs les professeurs Gérard FERRON et El Mostafa DAYA, pour leurs contributions dans ce travail de thèse. Ma sincère reconnaissance à vous Messieurs.

Mes sincères remerciements à Messieurs les professeurs Laurent TABOUROT et Loïc DARIDON pour avoir rapporter sur mes travaux de thèse.

J'adresse aussi mes vifs remerciements aux professeurs Paolo VANNUCCI, Gérard FERRON et El Mostafa DAYA ainsi qu'au Docteur Laurent DURRENBERGER d'avoir été examinateurs de cette thèse.

Un grand merci à Anne et Karima, les secrétaires du LEM3, pour leur disponibilité et leur dévouement.

Beaucoup de reconnaissance pour les collègues et les professeurs du Laboratoire pour leur disponibilité. Un clin d'oeil pour Mme ARUFFAT et Mme MARTINY pour leur soutien.

Merci chère Rim pour ton aide précieuse et ton amour pour ma petite Ines. Merci mes chères copines Imen et Mandana pour vos encouragements et votre présence aujourd'hui pour me soutenir. Merci Wafa pour tes encouragements et ton salé :). Merci Souheil pour tes encouragements et tes conseils.

Enfin, un grand merci à ma famille et à celle de mon mari pour leur soutien.

Cher mari, je ne trouve pas les mots pour te remercier pour tout ce que tu a fais pour m'aider à finir ma thèse. Merci Foued, mon Foued.

A ma maman,
à mon Foued,
à Isabelle,
à ma petite Ines,

"Tant qu'il y a la vie, il y a de l'espoir"

Table des matières

Liste des tableaux **13**

Introduction générale **1**

1 Différentiation Automatique et optimisation **5**

1.1 Introduction 6

1.2 Différentiation Automatique en mode linéaire tangent 8

1.2.1 Différentiation par transformation de source 9

1.2.2 Différentiation par surcharge d'opérateur 15

1.2.3 Validation du code différentié 17

1.3 Introduction à l'optimisation 18

1.3.1 Définitions 19

1.3.2 Méthodes de gradient 21

1.3.3 Méthodes de Quasi-Newton 24

1.3.4 Ordre des méthodes. 25

1.3.5 Recherche globale 26

1.3.6 Logiciels et bibliothèques d'optimisation 26

1.4 Conclusions 27

2 Identification de l'écrouissage des tôles métalliques **29**

2.1 Introduction 31

2.2 Matériaux élastoplastiques 33

2.2.1 Modélisation de l'écrouissage 35

2.2.2 Écrouissage isotrope 35

2.2.3 Écrouissage cinématique 35

2.2.4 Écrouissage mixte (isotrope et cinématique 36
2.2.5 Modélisation des surfaces de charge 37
2.2.6 Lois d'écrouissage 41
2.2.7 Essais de caractérisation de l'écrouissage 41
2.3 Identification de la loi d'écrouissage 48
2.3.1 Choix de la méthode d'identification 48
2.3.2 Résultats expérimentaux 49
2.3.3 Outils de simulation numérique 53
2.3.4 Mise en œuvre de l'identification 56
2.3.5 Validation de la différentiation 57
2.3.6 Etude de sensibilité des paramètres 58
2.3.7 Expériences Jumelles pour tester l'optimiseur 59
2.4 Résultats 60
2.5 Discussions 62
2.5.1 Courbes force-déplacement 62
2.5.2 Champs de déformation 66
2.6 Conclusions 69

3 Identification de la loi de comportement d'une poutre sandwich 71
3.1 Introduction 73
3.2 Matériaux viscoélastiques 75
3.2.1 Loi de comportement d'un matériau viscoélastique 76
3.2.2 Modélisation rhéologique de la loi viscoélastique 86
3.3 Méthodes de résolution du problème des vibrations linéaires 88
3.3.1 Méthodes approchées 89
3.3.2 Méthodes directes 92
3.3.3 Méthode Asymptotique Numérique 93
3.4 Problématique et constats 96
3.4.1 Essais expérimentaux 96
3.4.2 Simulation Numérique 98
3.4.3 Choix d'une loi viscoélastique pour l'identification 99
3.5 Identification de la loi viscoélastique 100
3.5.1 Modélisation inverse 100
3.5.2 Calcul du gradient 101
3.5.3 Différentiation Automatique 102

3.5.4 Pilote d'optimisation 105
3.6 Résultats de l'identification 106
3.7 Conclusion 108

Conclusion **111**

Résumé **123**

Table des figures

1.1 Quelques approches de différentiation. 6
1.2 Architecture de TAPENADE [Hascoët *et al.*, 2005a] 11
1.3 Interface du serveur Tapenade 13
1.4 Interface du serveur Tapenade : Code différentié *calcul_sigma_d* 14
1.5 Interface de TAPENADE en installation locale 14
1.6 Interface de sortie de Tapenade en installation locale 15

2.1 Etapes de l'identification par recalage (d'après Hendricks. [1991]) 33
2.2 Les différentes possibilités de chargement 34
2.3 Écrouissage isotrope (représentation dans le plan déviatoire) 35
2.4 Écrouissage cinématique (représentation dans le plan déviatoire) 36
2.5 Écrouissage isotrope et cinématique (représentation dans le plan déviatoire) 37
2.6 Surfaces de plasticité : Critère de Tresca et de Von Mises 39
2.7 Surfaces de plasticité dans le plan déviatorique 40
2.8 Essai de traction uni-axiale 42
2.9 Éprouvettes proposées par Wagoner [1981] pour des essais de traction large 44
2.10 Géométrie dans un essai de cisaillement simple 45
2.11 Géométrie d'un essai de gonflement hydraulique 45
2.12 Machine de traction compression ZWICK Z100 49
2.13 Les éprouvettes utilisées 50
2.14 Courbe de traction uni-axiale 50
2.15 Courbe de traction uni-axiale 51
2.16 Éprouvettes Acier multiphasé 52
2.17 Éprouvettes Acier doux 52
2.18 Éléments de maillage 54
2.19 Code différentié par TAPENADE 57

2.20 Étude de sensibilité 59
2.21 Expérience Jumelle 60
2.22 Courbes Force-Déplacement Acier multiphasé 63
2.23 Courbes Force-Déplacement Acier doux 64
2.24 Acier multiphasé 67
2.25 Acier doux 68
2.26 Acier doux 69

3.1 Déphasage contrainte-déformation 82
3.2 Hystérésis contrainte-déformation en fonction de η 83
3.3 Variation du module et facteur de perte en fonction de la température . . . 84
3.4 Courbes maitresse en fréquence réduite 85
3.5 Modèle d'amortissement à deux ou trois paramètres 87
3.6 Modèles généralisés : (a) Kelvin-Voigt généralisé ; (b) Maxwell généralisé . 88
3.7 Fonctionnement de l'appareil de flexion Oberst 91
3.8 Courbe de résonance au voisinage d'une fréquence de résonance 91
3.9 Ligne pilote de fabrication de tôles sandwich (Landier [1993]) 97
3.10 Amortissement de l'éprouvette sandwich 98
3.11 Chargement du code dans Tapenade 103
3.12 Différentiation du code par Tapenade 104
3.13 Variation du facteur d'amortissement en fonction de la fréquence 108
3.14 Identification des paramètres ER et EI à 20°C, 50°C, 60°C et 70°C 108

Liste des tableaux

1.1 Test de Taylor sur l'exemple Algorithme 1 dans la direction $(1, 0, 0, 0, 0, 0)^T$ 18

2.1 Lois d'écrouissage usuelles. 41
2.2 Propriétés mécaniques 51
2.3 Test de Taylor effectué dans les directions K et N. 58
2.4 Jeu de paramètres de l'expérience jumelle. 60
2.5 Domaines d'allongement choisis pour l'identification. 61
2.6 Paramètres identifiés. 61
2.7 Allongement d'une éprouvette de traction 65

3.1 Modèles rhéologiques linéaires simples 87
3.2 Modèles rhéologiques linéaires d'ordre élevé 88
3.3 Propriétés des matériaux et géométrie de la poutre. 97
3.4 Fréquences naturelles et facteurs de perte de la poutre sandwich. 99
3.5 Solveur du problème aux valeurs propres complexes 100
3.6 Test de Taylor effectué dans les directions η et E_R. 105
3.7 Processus d'optimisation pour un mode donné. 106
3.8 Fréquences et facteurs de perte ($w_2 = 10^{-6}$) 107

Introduction générale

Avec l'évolution rapide de la puissance de calcul des ordinateurs, de nouvelles pratiques ont émergé dans le domaine du calcul structurel numérique. La méthode des éléments finis est le premier choix pour ce type de calculs et permet de déterminer la réponse physique d'une structure à une sollicitation. Cette technique permet non seulement de déterminer les états de contrainte à l'intérieur d'une structure mécanique, mais aussi de modéliser des procédés de fabrication complets. De nos jours, la réduction importante des temps de calcul permet d'aborder des problèmes dits inverses. En répétant les calculs par éléments finis tout en modifiant les paramètres du matériau ou la géométrie de la structure, on arrive à identifier une solution optimale pour le problème considéré. La procédure, couplant optimisation et calcul par éléments finis est d'une importance primordiale pour l'industrie manufacturière car ce développement virtuel (réalisé à l'aide d'outils informatiques) permet de réduire le temps et les coûts de développement de nouveaux produits. La terminologie de " problème inverse " est utilisée par opposition à celle de " problème direct" par laquelle on comprend, par exemple, à la résolution d'une équation différentielle à partir de paramètres connus pour calculer la réponse du système. Dans le cas d'un problème inverse, la réponse du système est supposée connue. On cherche alors à déterminer les paramètres physiques ou géométriques qui, utilisés dans le problème direct, permettront de retrouver la réponse du système prescrite. Le problème inverse fait ainsi intervenir une fonction objectif, à construire en fonction de l'application, mesurant un écart entre la réponse connue et les réponses obtenues à partir de jeux de paramètres différents en résolvant le problème direct. On peut distinguer deux grandes classes de techniques pour résoudre un problème inverse. Les méthodes stochastiques [N.Bartoli et Moral, 2001] ou évolutives ont une place majeure en optimisation non différentiable et sont un recours pour les problèmes qui présentent des minima locaux. Les méthodes de gradients interviennent lorsque la fonction à optimiser est différentiable. Elles utilisent les informations données par les dérivées partielles. Dans le cas de fonctions différentiables dont on ne peut garantir

la convexité, des algorithmes hybrides ou mixtes sont parfois utilisés pour combiner les avantages des algorithmes stochastiques et des algorithmes de gradients. Le choix d'une méthode dépend de la nature du problème inverse (différentiable, non différentiable, ...) et surtout du temps de calcul nécessaire à l'évaluation de la réponse du système. Une hiérarchie générale est impossible et l'efficacité d'une méthode donnée varie selon les problèmes traités. Les techniques de type gradient ont été fréquemment considérées dans des applications pour lesquelles le temps nécessaire à l'évaluation du problème direct est important. Elles consistent à identifier le minimum de la fonction objectif comme un point où le gradient de cette fonction s'annule. Cette approche ne garantit pas l'identification du minimum global, mais elle a l'avantage de converger rapidement vers un minimum. Ce minimum sera global si l'itéré initial est assez proche de la solution recherchée (i.e. dans son bassin d'attraction), ce qui est assez souvent réalisé dans le cas de problèmes d'ingénierie.

Dans le premier chapitre, la technique de Différentiation Automatique (DA) [Griewank et Walther, 2008]en mode direct est exposée. La DA est une technique qui, à partir d'un code informatique évaluant une fonction, produit un code qui évalue les valeurs exactes des dérivées partielles de cette fonction. Le mode direct est le mode le plus simple à appliquer. Il est adapté au calcul des dérivées directionnelles d'une fonction quelconque représentée par un programme.On distingue deux types d'outils de DA : les logiciels qui génèrent des sources de codes dérivés à partir de la source du code de la fonction à différentier, et les bibliothèques qui proposent des classes (au sens informatique) décrivant des variables " dérivées " et leurs opérateurs. Dans le cadre de cette thèse, l'outil de DA par transformation de source TAPENADE [Hascoët *et al.*, 2005a] est utilisé pour différentier nos codes. Il est présenté en utilisant un exemple simple avant d'être appliqué à la résolution d'un problème d'identification de l'écrouissage des tôles métalliques (Chap. 2) et d'un problème d'identification de la loi de comportement d'une poutre sandwich (Chap. 3). Ensuite, on introduit l'optimisation par la méthode de gradient.

Le code choisi pour appliquer la technique de DA est un code éléments finis FLagSHyP [Bonnet et Wood, 2008] écrit en FORTRAN 77 . Ce code à but académique illustre les théories de la mécanique des milieux continus non linéaire décrites dans ce livre. Dans un premier temps, des pré-requis sont présentés sur la modélisation de la loi élasto-plastique ainsi que les essais de caractérisation. Par la suite, on expose le cadre général de notre application : loi de comportement élasto-plastique à 6 paramètres implémentée dans le code ainsi que les essais de traction uni-axiales réalisés par ArcelorMittal. Pour effectuer l'identification de ces paramètres, une fonction coût a été aussi implémenté dans FLagSHyP . La méthode du gradient est utilisée pour l'identification des 6 paramètres

de la loi de comportement. Le code est différentié automatiquement par rapport à ces paramètres avec l'outil de différentiation automatique TAPENADE afin de calculer le gradient de la fonction coût. Enfin un module d'optimisation LBFGS [Byrd *et al.*, 1995] est interfacé avec le code pour réaliser l'identification inverse. On présente les différents résultat obtenus pour deux aciers un isotrope et un anisotrope.

Dans le troisième chapitre, on propose une solution à une problématique mécanique d'identification. Dans [Daya et Potier-Ferry, 2001] on traite un problème de vibrations libres d'une tôle sandwich acier/polymère/acier. Le problème non-linéaire aux valeurs propres complexes résultant est résolu par la Méthode Asymptotique Numérique (MAN) [Cochelin *et al.*, 2007] implémentée dans un code éléments finis écrit en FORTRAN . Le comportement viscoélastique de la tôle est modélisé par la loi de Maxwell généralisé. Les paramètres de cette loi de comportement sont obtenus en utilisant les courbes maîtresses du polymère seul comme décrit par Landier [1993]. Contraint, le matériau viscoélastique peut ne pas avoir le même comportement. Ce qui peut expliquer les différences observées entre les simulations de Daya et Potier-Ferry [2001] et les essais réalisés par Landier [1993]. Ces différences ont motivé l'identification de la loi VE par une méthode inverse en utilisant la méthode de gradient. Dans un premier temps, on rappelle des propriétés mécaniques des matériaux viscoélastiques ainsi que leur modélisation numérique. Dans un second temps, on présente les objectifs d'identification ainsi que la solution numérique du problème non-linéaire aux valeurs propres complexes. Ensuite, on décrit la différentiation automatique du code par rapport aux paramètres de la loi VE. Quelques améliorations apportées au code différentié sont aussi présentées. Après quoi on décrit les étapes d'optimisation. Finalement on présente les résultats obtenues.

Dans la conclusion générale on présente l'ensemble des résultats importants mis en avant durant ce travail de thèse. On y évoque aussi diverses perspectives afin d'améliorer les résultats obtenues.

CHAPITRE 1

Différentiation Automatique et optimisation

Sommaire

1.1 Introduction **6**

1.2 Différentiation Automatique en mode linéaire tangent **8**

1.2.1 Différentiation par transformation de source 9

1.2.2 Différentiation par surcharge d'opérateur 15

1.2.3 Validation du code différentié 17

1.3 Introduction à l'optimisation **18**

1.3.1 Définitions 19

1.3.2 Méthodes de gradient 21

1.3.3 Méthodes de Quasi-Newton 24

1.3.4 Ordre des méthodes. 25

1.3.5 Recherche globale 26

1.3.6 Logiciels et bibliothèques d'optimisation 26

1.4 Conclusions **27**

1.1 Introduction

Disposer d'un modèle reproduisant un phénomène mécanique permet d'obtenir des informations qualitatives et quantitatives sur celui-ci. Cependant, certaines informations sont inaccessibles sans l'usage des dérivées. En supposant ce modèle différentiable, les dérivées de premier ordre renseignent sur la sensibilité du modèle à des variations de ses paramètres d'entrée et offrent des gradients pour la résolution de problèmes inverses. Les dérivées de second ordre s'emploient, par exemple, pour le calcul du Jacobien des contraintes ou pour le calcul de la matrice Hessienne dans la méthode de Newton. Enfin les dérivées d'ordre élevé servent par exemple à l'approximation d'une fonction par les séries de Taylor tronquées [Cochelin *et al.*, 2007]. Différentes approches de différentiation sont résumées par la figure 1.1.

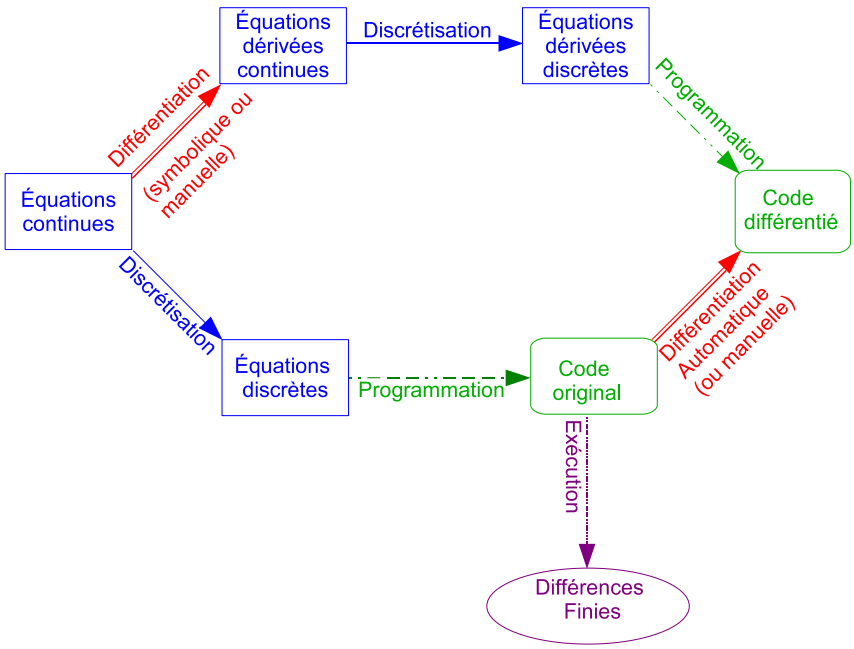

FIG. 1.1 – Quelques approches de différentiation.

Dans cette figure, les rectangles bleus et les cadres arrondis verts distinguent les systèmes d'équations, des codes. Ils sont liés par des flèches dont la couleur indique le traitement effectué : trait plein bleu pour la discrétisation, trait double rouge pour la différentiation, trait pointillé vert pour la programmation et trait violet pour l'exécution.

La méthode des *différences finies* peut être appliquée pour approcher une dérivée à l'ordre 1. Pour un calcul de gradient, l'approximation est effectuée indépendamment pour

chacune des variables. Soit :

$$f(x,y) : \mathbb{R}^2 \to \mathbb{R}$$
$$(x,y) \longmapsto f(x,y),$$

la dérivée f_x de f par rapport à x est approchée par :

$$\frac{f(x+h,y) - f(x,y)}{h} = f_x(x,y) + o(h).$$

Cette méthode approchée est coûteuse lorsque le nombre de variables est élevé.

La différentiation symbolique est proposée par des outils comme Maple ou Mathématica. Cette méthode suppose que la fonction à dériver admette une représentation symbolique. La dérivée est aussi une représentation symbolique.

Un programme de simulation numérique s'écrit à l'aide d'opérations élémentaires (+, -, *, ...), de fonctions intrinsèques (cos, sin...), et d'instructions de contrôle (boucles, tests, ...). Pour calculer la valeur de la dérivée du programme en un point donné, il suffit d'être capable d'évaluer la dérivée de chaque opération élémentaire et de chaque fonction du langage. La dérivation de ces instructions peut s'effectuer manuellement. Dans le cas d'instructions complexes, la *dérivation manuelle* devient fastidieuse car, bien que les règles de différentiation soient connues de tous, leur usage répété constitue une source d'erreurs. Erreurs d'indices dans les boucles, vecteurs, tableaux et autres erreurs de signe..., sont souvent le prix à payer avant de parvenir à un code différentié correct.

La technique de *Différentiation Automatique* (DA) peut être utilisée pour différentier des codes entiers. Les principes de base de la DA ont été établis il y a une cinquantaine d'années [Wengert, 1964]. Depuis, cette approche n'a cessé de se développer. De nombreux outils sont désormais proposés et l'usage de cette approche est de plus en plus fréquent dans de nombreux domaines scientifiques. Le livre de Griewank et Walther [2008] fait référence dans la communauté scientifique qui se consacre à la DA. Cette technique permet de calculer des dérivées du programme de manière exacte, c'est à dire aux erreurs d'arrondis près. Il existe trois modes de DA : le mode direct (mode linéaire tangent ou tangent linéaire vectoriel), le mode inverse (appelé aussi mode adjoint) et la différentiation de haut degré. Dans ce manuscrit, nous ne nous intéressons qu'au premier. Deux classes d'outils existent. Des logiciels de différentiation par transformation de source, c'est à dire générant le code dérivé à partir du code source de l'utilisateur, et des outils de différentiation par surcharge d'opérateur redéfinissant variables et opérateurs dans des bibliothèques dédiées à la DA. Des outils de DA ont été développés pour différents langages de programmation

(C/C++, FORTRAN 77/95, Matlab, Python,...). Le site http://www.autodiff.org/ propose la liste des logiciels disponibles et une vaste bibliographie sur la DA.

La première partie de ce chapitre expose la technique de DA en mode direct, puis l'outil TAPENADE utilisé pour différentier nos codes FORTRAN. Dans la seconde partie, on introduit la méthode de gradient pour la résolution de problèmes d'optimisation.

1.2 Différentiation Automatique en mode linéaire tangent

Définitions et principes de la DA sont illustrés par le calcul de la contrainte σ :

$$\sigma(\varepsilon) = K(\varepsilon_0 + \varepsilon)^N + S(1 - A * \mathrm{Exp}(-B\varepsilon)), \tag{1.1}$$

dont le gradient évalué dans la direction $\delta = (\delta K, \delta\varepsilon_0, \delta N, \delta S, \delta A, \delta B)^T$ est :

$$\nabla\sigma(\varepsilon) \cdot \delta = \begin{pmatrix} (\varepsilon_0 + \varepsilon)^N \\ NK(\varepsilon_0 + \varepsilon)^{N-1} \\ K(\varepsilon_0 + \varepsilon)^N Log(\varepsilon_0 + \varepsilon) \\ 1 - AExp(-B\varepsilon) \\ -SExp(-B\varepsilon) \\ S\varepsilon AExp(-B\varepsilon) \end{pmatrix} \cdot \begin{pmatrix} \delta K \\ \delta\varepsilon_0 \\ \delta N \\ \delta S \\ \delta A \\ \delta B \end{pmatrix}. \tag{1.2}$$

La routine FORTRAN *calcul_sigma*, l'Algorithme 1 permet le calcul de la contrainte σ en fonction de la déformation et des propriétés du matériaux $(K, \varepsilon_0, N, S, A, B)$, réunis dans un vecteur nommé *props*.

```
      subroutine calcul_sigma(props,epsilon,sigma)

      double precision epsilon,sigma,sigma1,sigma2,props(6)
c     props(1)=K, props(2)=ε0, props(3)=N, props(4)=S, props(5)=A, props(6)=B
      sigma1=props(1)*(props(2)+epsilon)**props(3) ! instruction 1
      sigma2=props(4)*(1-props(5)*exp(-props(6)*epsilon)) ! instruction 2
      sigma=sigma1+sigma2   ! instruction 3
      end
```

Algorithme 1: Calcul de σ

La différentiation est effectuée par rapport à la variable d'entrée *props*. Cette variable est dite *variable active indépendante* (ou variable indépendante). La variable de sortie *sigma* est une *variable active dépendante* (ou variable dépendante) car, dépendante de *props*, elle en devient active pour la différentiation. La différentiation automatique en

mode direct est très naturelle : le code est différentié par l'outil, instruction par instruction et ligne par ligne, en utilisant les règles de différentiation classiques.

1.2.1 Différentiation par transformation de source

Un outil de différentiation par transformation de source (ADIFOR [Bischof *et al.*, 1994] ou TAPENADE [Hascoët *et al.*, 2005a] par exemple)[1] prend en entrée un code source et un ensemble de variables actives indépendantes pour générer le code source différentié. Ces outils permettent de différentier l'intégralité d'un programme : instructions d'affectation, instructions de contrôle, boucles, expressions conditionnelles et appels de routines. Dans les logiciels, les étapes de différentiation décrites ci-dessous sont cachées à l'utilisateur.

a) Principe des logiciels

Une instruction d'affectation soumise à un logiciel de différentiation par transformation de source, est différentiée en trois étapes. Pour illustrer ces idées revenons à l'exemple décrit dans l'Algorithme 1. Les codes différentiés sont produits par le logiciel TAPENADE décrit dans le paragraphe b).

```
sigma1=props(1)*(props(2)+epsilon)**props(3) ! instruction 1
```

Algorithme 2: Instruction 1

Décomposition et analyse : Le logiciel analyse et décompose l'instruction 1 en instructions élémentaires par introduction de variables intermédiaires ($pwx1$ et $pwr1$). On obtient :

```
pwx1 = props(2) + epsilon
pwr1 = pwx1**props(3)
sigma1 = props(1)*pwr1
```

Algorithme 3: Décomposition de l'instruction 1

[1] voir la liste des logiciels sur http://www.autodiff.org/?module=Tools

Différentiation : Une variable "dérivée" est associée à chaque variable active (dépendante ou indépendante) du code à différentier. Son nom est celui de la variable originale auquel on ajoute un suffixe *d*. La différentiation de l'instruction 1 par rapport aux variables actives indépendantes *props* donne :

```
pwx1d = propsd(2)
IF (pwx1 .GT. 0.0) THEN
  pwr1d = pwx1**props(3)*(LOG(pwx1)*propsd(3)+props(3)*pwx1d/pwx1)
ELSE IF (pwx1 .EQ. 0.0) THEN
  IF (props(3) .EQ. 1.0) THEN
    pwr1d = pwx1d
  ELSE
    pwr1d = 0.0
  END IF
ELSE IF (props(3) .EQ. INT(props(3))) THEN
  pwr1d = props(3)*pwx1**(props(3)-1)*pwx1d
ELSE
  pwr1d = 0.0
END IF
sigma1d = propsd(1)*pwr1 + props(1)*pwr1d
```

Algorithme 4: Différentiation de l'instruction 1

Ces instructions sont complexes car l'outil de différentiation prend en compte le fait que N peut être à valeur entière ou réelle.

```
C
      SUBROUTINE CALCUL_SIGMA_D(props, propsd, epsilon, sigma, sigmad)
      IMPLICIT NONE
C
      DOUBLE PRECISION epsilon, props(6), propsd(6), sigma, sigmad
      DOUBLE PRECISION pwr1, pwr1d, pwx1, pwx1d, sigma1, sigma1d, sigma2
     +                 , sigma2d
      INTRINSIC EXP

      pwx1d = propsd(2)
      pwx1 = props(2) + epsilon
      IF (pwx1 .GT. 0.0) THEN
        pwr1d = pwx1**props(3)*(LOG(pwx1)*propsd(3)+props(3)*pwx1d/pwx1)
      ELSE IF (pwx1 .EQ. 0.0) THEN
        IF (props(3) .EQ. 1.0) THEN
          pwr1d = pwx1d
        ELSE
          pwr1d = 0.0
        END IF
      ELSE IF (props(3) .EQ. INT(props(3))) THEN
        pwr1d = props(3)*pwx1**(props(3)-1)*pwx1d
      ELSE
        pwr1d = 0.0
      END IF
      pwr1 = pwx1**props(3) ! Trajectoire
      sigma1d = propsd(1)*pwr1 + props(1)*pwr1d
      sigma1 = props(1)*pwr1 ! Trajectoire
C
C
C
      sigma2d = propsd(4)*(1-props(5)*EXP(-(props(6)*epsilon))) + props(
     +  4)*(props(5)*epsilon*propsd(6)*EXP(-(props(6)*epsilon))-propsd(5
     +  )*EXP(-(props(6)*epsilon)))
      sigma2 = props(4)*(1-props(5)*EXP(-(props(6)*epsilon))) ! Trajectoire
C
      sigmad = sigma1d + sigma2d
      sigma = sigma1 + sigma2 ! Trajectoire
      END
```

Algorithme 5: Code différentié *Calcul_sigma_d*

Affichage : Dans cette étape, certaines variables intermédiaires sont éliminées pour améliorer la lisibilité aux codes dérivés.

Quand on traite plusieurs instructions, la trajectoire d'évaluation des dérivées est reproduite dans le code dérivé en intercalant les instructions originales et les instructions dérivées comme signalé dans l'Algorithme 5. Les instructions conditionnelles et les boucles itératives sont reproduites dans le code.

b) TAPENADE : un logiciel libre de DA

Présentation Parmi les outils de DA existants, on a choisi TAPENADE [Hascoët *et al.*, 2005a] pour différentier nos codes FORTRAN. Depuis la fin des années 90, l'équipe Tropics de l'INRIA Sophia Antipolis développe le logiciel de différentiation automatique TAPENADE. Cet outil propose à la fois la dérivation en mode linéaire tangent, en mode adjoint et en mode vectoriel tangent.
TAPENADE [Hascoët *et al.*, 2005a], accepte des codes en FORTRAN 77, 95 et C. La figure 1.2 présente l'architecture de TAPENADE. Lorsqu'un code source est soumis à TAPENADE, ce logiciel est capable de produire un code source qui calcule les dérivées du code original (ou d'une partie du code) en dérivant le code par rapport à un ensemble de variables d'entrée et un ensemble de variables de sortie.

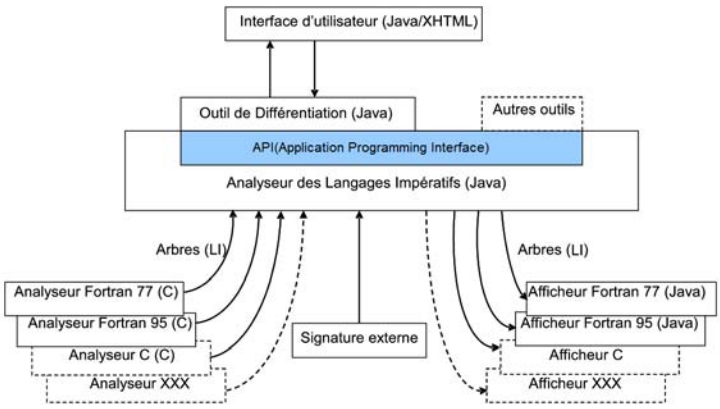

FIG. 1.2 – Architecture de TAPENADE [Hascoët *et al.*, 2005a]

Les analyseurs d'entrée et afficheurs de sortie (un pour chaque langage) sont clairement séparés du noyau qui ne connaît qu'un langage abstrait appelé Langage Impératif. Idéalement, le langage impératif contient toutes les constructions sémantiques qui existent

dans les langages de programmation FORTRAN et C.
Le noyau de TAPENADE est un Analyseur de Langage Impératif (ALI) [Hascoët *et al.*, 2005a] qui génère un graphe d'appel du programme. L'outil de DA est construit au dessus de ce noyau et interfacé à l'aide d'une API (Application Programming Interface) qui met en œuvre les analyses spécifiques de la DA. Le noyau est alors capable d'envoyer des programmes en Langage Impératif différentiés à l'afficheur du langage considéré. Plusieurs techniques ont été implémentées dans TAPENADE pour améliorer ses performances :

- l'analyse de l'utilité [Hascoët *et al.*, 2005b] permet de ne pas différentier les variables dépendantes des variables actives indépendantes n'ayant aucune influence sur les variables dépendantes intéressant l'utilisateur ;
- l'analyse de sauvegarde en mode adjoint qui permet de minimiser les sauvegardes ;
- l'analyse du flot des données, localement à chaque bloc d'instructions, pour regrouper les instructions dérivées et améliorer les performances du mode tangent vectoriel ;
- l'analyse In-Out utilisée pour réduire la taille mémoire nécessaire à l'exécution répétée de certaines sous-parties de programme.

Distribution TAPENADE est téléchargeable sur `ftp://ftp-sop.inria.fr/tropics/tapenade/` pour une installation locale et libre d'utilisation pour les recherches académiques. Il est aussi accessible en ligne via un serveur web à l'adresse
`http://tapenade.inria.fr:8080/tapenade/index.jsp`

Utilisation Pour différentier un code par TAPENADE (Fig. 1.3), l'utilisateur renseigne différents champs :

1. choix du langage de programmation

2. choix des programmes à différentier,

3. indication du nom de la routine principale,

4. indication du nom des variables actives indépendantes(variables d'entrées),

5. indication du nom des variables actives dépendantes(variables de sorti, optionnel),

6. choix du mode de différentiation (linéaire tangent, linéaire tangent vectoriel ou adjoint).

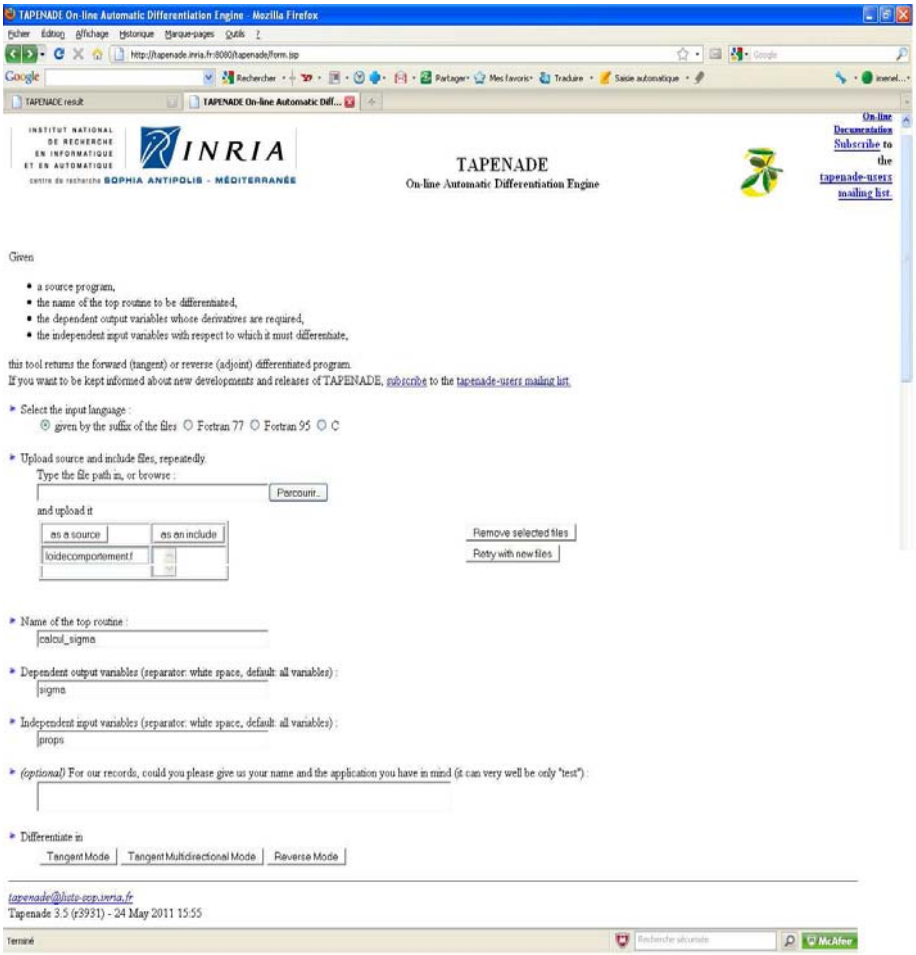

FIG. 1.3 – Interface du serveur Tapenade

Après avoir précisé ces informations à l'aide de la fenêtre de navigation, on lance la différentiation. L'outil de DA analyse alors le code avec l'analyseur syntaxique FORTRAN pour extraire l'architecture du code et la dépendance des routines par rapport aux variables actives indépendantes. TAPENADE traite les routines nécessitant une dérivation. TAPENADE ajoute la lettre "d" aux variables différentiées dans le mode tangent et "$_d$" aux noms des routines. Une fois le code différentié, une nouvelle fenêtre apparaît. L'utilisateur peut télécharger le code source généré à l'aide du bouton "Download differentiated file"(Fig. 1.4).

Lorsque le code à différentier est grand (plus de 15000 lignes), on ne peut plus utiliser l'outil en ligne. On doit alors installer TAPENADE localement. Des versions de TAPENADE existent pour les systèmes d'exploitation : Windows, Linux et Mac. La figure (Fig. 1.5) présente l'interface locale de TAPENADE pour Windows. Les résultats de

différentiation s'affichent sur une page web (Fig. 1.6) consultable dans un dossier créé au cours de l'exécution. Les routines différentiées sont générées dans le même répertoire.

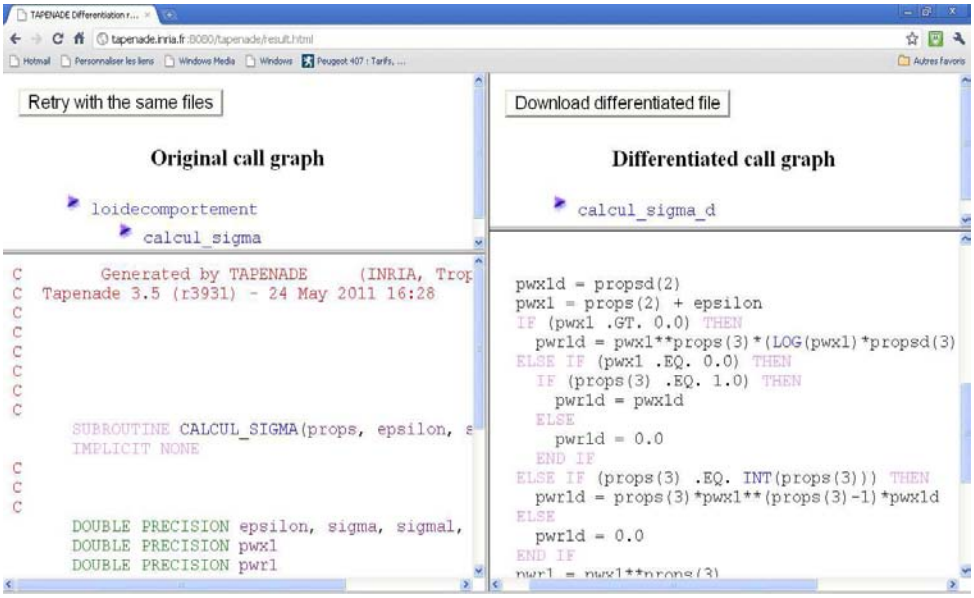

FIG. 1.4 – Interface du serveur Tapenade : Code différentié *calcul_sigma_d*

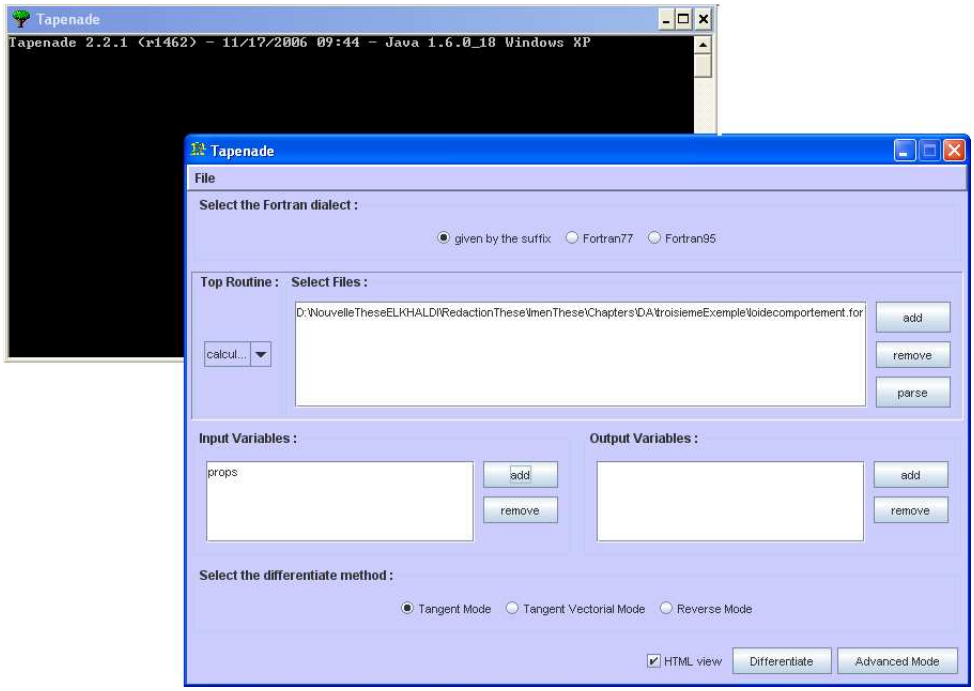

FIG. 1.5 – Interface de TAPENADE en installation locale

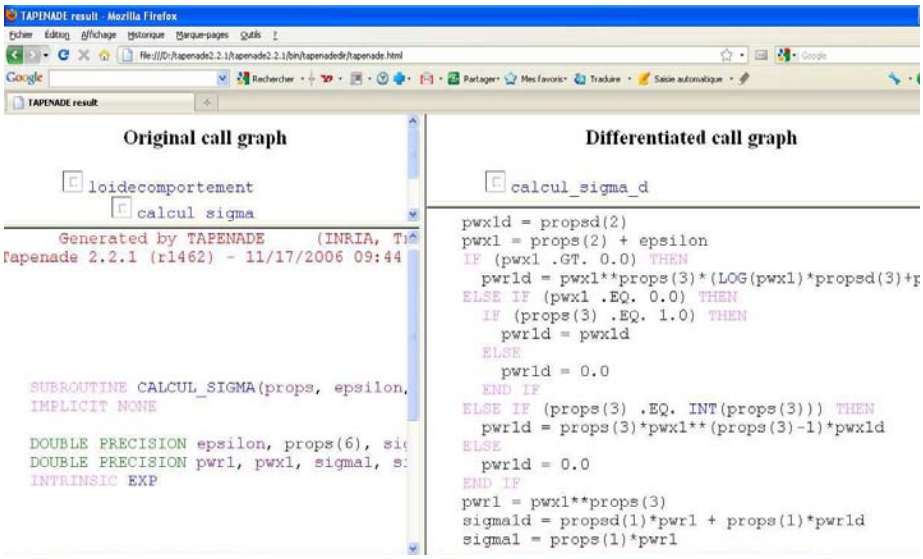

FIG. 1.6 – Interface de sortie de Tapenade en installation locale

1.2.2 Différentiation par surcharge d'opérateur

a) Principe

D'un point de vue informatique, la surcharge d'opérateur (Operator Overloading) est un cas particulier de polymorphisme, par lequel des opérateurs peuvent avoir plusieurs implémentations. Le compilateur choisit, en fonction du type des opérandes, quelles opérations effectuer. La technique de surcharge d'opérateur est proposée par des langages de programmation orienté objet comme FORTRAN 90, C++ et Matlab. Parmi les logiciels de DA par surcharge d'opérateur, on trouve ADOL-C [Griewank *et al.*, 1996], FADBAD [Bendtsen et Stauning, 1996], FAD [Aubert *et al.*, 2001], Rapsodia [Charpentier et Utke, 2009], Diamant [Charpentier, 2008]. La surcharge d'opérateurs en particulier utilisés pour la différentiation d'ordre élevé.
En différentiation automatique, la surcharge d'opérateur permet d'obtenir un code capable de calculer des dérivées sans avoir changer la structure du code d'origine. Il suffit de définir un nouveau type et de redéfinir les opérations. En pratique, la construction d'un type et la redéfinition des opérateurs sont faites une fois pour toutes. Il suffit donc de coder une bibliothèque, puis de changer le type des variables actives dans le code. La propagation est gérée par l'utilisateur. La bibliothèque et le code modifié sont compilés ensemble pour produire un exécutable capable de calculer des dérivées.

b) Exemple

Le type *dreal* dédié à la différentiation au premier ordre est présenté ci-dessous.

```
TYPE dreal
REAL : : d0, d1
END TYPE dreal
```

Algorithme 6: Le type *dreal*

Une variable x de type *dreal* a ses champs $x\%d0$ et $x\%d1$ respectivement consacrés à la valeur de la variable (qui sert de trajectoire) et à la valeur de sa dérivée. L'opérateur surchargé "$*$" pour deux variables de type *dreal* est présenté dans l'Algorithme 6 :

```
Function prod_dreal_dreal(a,b) : Result (r)
    TYPE (dreal), INTENT(IN) : : a, b
    TYPE dreal : :r
    r%d0 = a%d0 * b%d0
    r%d1 = a%d1 * b%d0+b%d1 * a%d0
End
```

Algorithme 7: Surcharge de l'opérateur $*$ pour deux variables de type *dreal*

Si le premier opérande est une variable de type *dreal* et le second de second de type *real* alors l'opérateur " $*$ " est surchargé comme suit :

```
Function prod_dreal_real(a, b) : Result (r)
    TYPE (dreal), INTENT(IN) : : a
    TYPE (real), INTENT(IN) : :b
    TYPE (dreal) : : r
    r%d0 = a%d0 * b%d0
    r%d1 = a%d1 * b%d0
End
```

Algorithme 8: Surcharge de l'opérateur $*$ pour une variable a de type *dreal*, et une variable réelle b

L'operation "*" pour le couple (real,dreal) doit également être implémentée. En pratique, la surcharge concerne tous les opérateurs arithmétiques et fonctions élémentaires pour chaque paire de types de variable.

1.2.3 Validation du code différentié

Après avoir différentié un code, il est usuel de le valider par un test de Taylor dans lequel les dérivées partielles produites par le code dérivé sont comparées avec celles approchées à l'ordre 1 par une méthode de différence finies. Le test consiste à :

1. choisir une direction de perturbation *propsd* du vecteur *props*. On choisira *propsd* dans la base canonique pour effectuer le test par rapport à l'une ou l'autre variable.

2. Calculer le rapport $r(\alpha)$ en faisant tendre α vers 0 :

$$r(\alpha) = \frac{\mid calcul_sigma(props + \alpha propsd) - calcul_sigma(props) \mid}{\alpha \mid \nabla(calcul_sigma_d(props, propsd \mid}$$

En théorie, le code généré est correct si pour chaque direction de perturbation *propsd* de la variable *props*, le rapport $r(\alpha)$ "tend linéairement vers 1" lorsque α tends vers 0.

Le tableau (Tab. 1.1) montre le comportement du test de Taylor effectué sur l'exemple Algorithme 1 dans la direction $propsd = (1, 0, 0, 0, 0, 0)^T$: il y a convergence linéaire vers 1 jusqu'au pas optimal de différence finies de l'ordre de 10^{-7}. Pour un α inférieur à ce pas, les erreurs d'arrondi dominent l'erreur de troncature, et $r(\alpha)$ s'éloigne de 1. Le code a été correctement dérivé par rapport à la variable considérée(K dans cet exemple) lorsqu'un tel comportement est observé.

α	$r(\alpha)$
10^{-1}	0.975609756097573
10^{-2}	0.997506234413947
10^{-3}	0.999750062485797
10^{-4}	0.999975000629168
10^{-5}	0.999997500154374
10^{-6}	0.999999751153761
10^{-7}	0.999999988001339
10^{-8}	0.999999845892792
10^{-9}	0.999998898502478
10^{-10}	1.00001784630876
10^{-11}	0.999970476793047
10^{-12}	0.999496781635874
10^{-13}	0.994759830064140
10^{-14}	0.947390314346800

TAB. 1.1 – Test de Taylor sur l'exemple Algorithme 1 dans la direction $(1, 0, 0, 0, 0, 0)^T$

1.3 Introduction à l'optimisation

Les premières tentatives de minimisation de fonctions sont attribuées à Cauchy en 1847. La méthode utilisée, dite de la plus grande pente, a été depuis largement améliorée. Depuis, les domaines d'application de l'optimisation ne cessent de se diversifier.
Les méthodes d'optimisation sont d'une importance majeure dans les applications mécaniques. Pour la fabrication d'une nouvelle pièce, par exemple, on doit tenir compte dans le choix du matériau utilisé, des contraintes mécaniques et géométriques auxquelles elle sera soumise. Une fois que ce problème de conception est résolu, on a besoin de déterminer les procédés qui permettent d'obtenir la pièce à moindre coût. Résoudre les problèmes de conception et de procédés est une tâche complexe.
D'un point de vue mathématique, le problème inverse est un problème d'optimisation de fonctionnelles par rapport à un ensemble de paramètres de contrôle. Dans ce domaine vaste, on trouve un très grand nombre de méthodes.

Deux grandes familles de méthodes d'optimisation existent. Les méthodes de type gradient exigent le calcul du gradient de la fonction à minimiser. Les méthodes stochastique ou d'exploration directe ne nécessitent pas ce calcul. Parmi les méthodes de gradient figurent notamment la méthode de la plus grande pente, la méthode de Gauss-Newton [Gavrus *et al.*, 1996], la méthode du gradient conjugué, la méthode de Quasi-Newton [Byrd *et al.*, 1995], [Gilbert et Lemaréchal, 1995]. Une méthode de gradient construit ses

itérés en diminuant la valeur de la fonction objectif dans la direction du plus grand gradient. Elle va nécessairement atteindre un minimum (s'il en existe un), mais celui-ci n'est que local lorsque la fonction à minimiser n'est pas convexe. Le minimum atteint est déterminé par l'itéré initial. À chaque minimum local est associée une zone de l'espace des contrôles (appelé bassin d'attraction), l'ensemble des itérés initiaux choisis dans cette zone mènent la séquence d'itérés vers ce minimum. Le point initial de la méthode numérique de recherche est alors d'une très grande importance. Les méthodes d'exploration directe cherchent à optimiser les paramètres par l'évaluation successives des valeurs de la fonction objectif. Parmi ces méthodes figurent des méthodes heuristiques, la méthode de Monte-Carlo, les méthodes du simplexe, les méthodes génétiques...

Dans le cadre de cette thèse on s'intéresse à la méthode du gradient.

1.3.1 Définitions

a) Problème de minimisation générique.

Soit $\mathcal{J}$ une fonction continue et différentiable de $\mathcal{X}$ en $\mathbb{R}$. Le problème de minimisation s'écrit :

$$\textit{Chercher } x^* \in \mathcal{X} \textit{ tel que } x^* \textit{ vérifie } : \quad \mathcal{J}(x^*) = \min_{x \in \mathcal{X}} \mathcal{J}(x). \tag{1.3}$$

Dans cette formulation :

1. $\mathcal{J}$ est le critère à minimiser ;

2. x est la variable de contrôle ;

3. x^* est la solution du problème de minimisation ;

4. $\mathcal{X}$ est l'espace des contrôles, appelé espace des contraintes en optimisation. Les contraintes h_i (à valeur réelle) imposées à la solution x^* peuvent être :

$$\begin{cases} h_i(x) \leq 0, & i = 1, .., m, \\ h_i(x) = 0, & i = m+1, .., m+p. \end{cases} \tag{1.4}$$

Dans un problème sans contraintes, les paramètres m et p sont nuls.

b) Problème de maximisation générique.

Soit $\mathcal{J}$ une fonction continue et différentiable de $\mathcal{X}$ en $\mathbb{R}$. Le problème de maximisation de $\mathcal{J}$ s'écrit comme le problème de minimisation de $-\mathcal{J}$:

$$\textit{Chercher } x^* \in \mathcal{X} \textit{ tel que } x^* \textit{ vérifie } : \quad -\mathcal{J}(x^*) = \min_{x \in \mathcal{X}} -\mathcal{J}(x). \tag{1.5}$$

c) Minimum global.

Si x^* est un minimum global de $\mathcal{J}$ dans l'espace $\mathcal{X}$, alors,

$$\mathcal{J}(x^*) \leq \mathcal{J}(x) \quad \forall x \in \mathcal{X}. \tag{1.6}$$

Lorsque ce minimum global est unique, il vérifie :

$$\mathcal{J}(x^*) < \mathcal{J}(x), \quad \forall x \in \mathcal{X}. \tag{1.7}$$

d) Minimum local.

Un minimum local $\widehat{x}$ de $\mathcal{J}$ se définit en fonction de la zone $\mathcal{V}(\widehat{x})$ dans laquelle $\widehat{x}$ est le minimum (global) de $\mathcal{J}$. Comme $\widehat{x}$ n'est que le minimum local de $\mathcal{J}$ dans $\mathcal{V}(\widehat{x})$, il existe dans l'espace $\mathcal{X} \setminus \mathcal{V}(\widehat{x})$ au moins un minimum (local ou global) $\widehat{\widehat{x}}$ tel que :

$$\mathcal{J}(\widehat{\widehat{x}}) < \mathcal{J}(\widehat{x}). \tag{1.8}$$

e) Unicité.

Il est souvent difficile de prouver l'unicité d'un minimum global.

f) Convergence globale.

En optimisation, un algorithme converge globalement lorsque la séquence des itérés x^k qu'il génère, atteint un point $\widehat{x}$ satisfaisant les conditions d'optimalité quel que soit l'itéré initial x^0, c'est à dire

$$\nabla \mathcal{J}(x^k) \to 0 \text{ lorsque } k \to +\infty, \qquad \forall x^0, \tag{1.9}$$

où $\nabla \mathcal{J}$ est le gradient de la fonction $\mathcal{J}$. Le point $\widehat{x}$ peut être le minimum global x^*.

g) Vitesse de convergence.

La qualité d'un algorithme d'optimisation s'apprécie à l'aide du ratio :

$$\rho^{k+1} = \frac{x^{k+1} - \widehat{x}}{x^k - \widehat{x}}, \tag{1.10}$$

équation où $\widehat{x}$ est le point atteint lorsqu'est vérifiée (1.9). La convergence est dite :

1. linéaire lorsque $\rho^k < 1$ quel que soit k ;

2. super-linéaire lorsque $\lim \rho^k = 0$. Pour affiner la description, le terme $x^{k+1} - \widehat{x}$ peut être comparé à des puissances de $x^k - \widehat{x}$. La convergence est quadratique lorsque $x^{k+1} - \widehat{x}$ est du même ordre que $(x^k - \widehat{x})^2$.

Un algorithme ayant une convergence quadratique est théoriquement meilleur qu'un algorithme ayant une convergence linéaire.

1.3.2 Méthodes de gradient

Dans le cursus universitaire, ces méthodes sont abordées pour la première fois au travers de la résolution de systèmes linéaires :

$$Ax = b, \tag{1.11}$$

où la matrice réelle définie positive A est de dimension $n \times n$, le second membre b et l'inconnue x sont des vecteurs réels de dimension n. Ce système linéaire peut être résolu par une méthode dite directe (LU par exemple) qui décompose la matrice A en matrices triangulaires, puis résout des systèmes linéaires impliquant ces matrices. Ces méthodes sont efficaces lorsque n n'est pas très grand. L'alternative classique est d'utiliser une méthode itérative (Jacobi, Gauss-Seidel, gradient conjugué). La méthode du gradient est très différente des deux autres dans sa conception, car elle propose de résoudre le système (1.11) en résolvant problème de minimisation :

$$\textit{Chercher } x \textit{ tel que} \quad \min_y \mathcal{J}(y) = \min_y \frac{1}{2}(y, Ay) - y^T.b = \frac{1}{2}(x, Ax) - x^T.b. \tag{1.12}$$

Comme $\mathcal{J}$ est différentiable en y, le point x vérifie :

$$\nabla \mathcal{J}(x) = \nabla \left[\frac{1}{2}(x, Ax) - x^T.b \right] = Ax - b = 0. \tag{1.13}$$

Les méthodes décrites dans ce paragraphe permettent de résoudre ce problème, mais aussi des problèmes d'optimisation impliquant des fonctions objectif $\mathcal{J}o\mathcal{M}$ beaucoup plus complexes. Elles ont en commun un processus itératif calculant le nouvel itéré par une formule du type :

$$x^{k+1} = x^k - \rho^k d^k, \tag{1.14}$$

où ρ^k et $d^k \simeq -\nabla\mathcal{J}(x^k)$ sont respectivement le pas et la direction de descente.

1. Initialisation. Choix d'un point x^0.

2. Itérés. L'itéré x^k sert à calculer l'itéré x^{k+1} comme suit :

a) Calcul $\mathcal{J}(x^k)$ et $\nabla\mathcal{J}(x^k)$;

b) Test du critère d'arrêt :

i. Si le critère d'arrêt est satisfait :
Arrêt de l'optimiseur, la solution est x^k.

ii. Sinon :
Calcul du pas ρ^k,
Calcul de la direction d^k,
Calcul du nouvel itéré $x^{k+1} = x^k - \rho^k d^k$,
Retour en (2) ;

.

Algorithme générique

L'algorithme de gradient à pas optimal est l'un des plus simples de la famille mais il est peu efficace. Trois points demeurent ouverts dans cet algorithme.

Choix de ρ^k -

- **Méthode 1 : Algorithme de gradient à pas constant.** Le paramètre ρ^k est constant. Cet algorithme n'est pas performant car si la descente dans la direction d^k est insuffisante, il est possible qu'un prochain calcul de gradient implique à nouveau cette direction.
- **Méthode 2 : Algorithme de gradient à pas optimal.** Le paramètre ρ^k détermine le pas permettant de diminuer au maximum la fonction $\mathcal{J}$ dans la direction

$\nabla\mathcal{J}(x^k)$. Il est identifié par la résolution du problème de minimisation :

$$\rho^k = \min_{\rho\geq 0} \mathcal{J}(x^k - \rho\nabla\mathcal{J}(x^k)). \tag{1.15}$$

Cette seconde optimisation, appelée recherche linéaire, est peu chère car le gradient $\nabla\mathcal{J}(x^k)$ est connu.

Critère d'arrêt - Évalué pour chaque itéré, il porte par exemple sur
1. le gradient :

$$||\mathcal{J}(x^k)|| < \varepsilon_g; \tag{1.16}$$

2. la différence entre deux calculs successifs de $\mathcal{J}$:

$$|\mathcal{J}(x^k) - \mathcal{J}(x^{k-1})| < \varepsilon_h; \tag{1.17}$$

3. ou la différence entre deux itérés successifs :

$$||x^k - x^{k-1}|| < \varepsilon_d. \tag{1.18}$$

Les paramètres ε_d, ε_g et ε_h sont des paramètre réels positifs choisis par l'utilisateur.

Direction de descente - Sa construction dépend de la méthode. L'expression la plus simple est :

$$d^k = -\nabla\mathcal{J}(x^k). \tag{1.19}$$

Dans les méthodes plus complexes, elle est calculée à l'aide du hessien (méthode de Newton) ou d'une approximation de celui-ci (méthode de quasi-Newton).

– **Méthode 3 : Algorithme de Newton.** La méthode de Newton nécessite le gradient et le Hessien de la fonction $\mathcal{J}$. De convergence quadratique, elle est la plus rapide en nombre d'itérations. Par contre, ses performances effectives, en temps de calcul, dépendent du coût de calcul des dérivées secondes. Lorsque celui-ci est important, il est préférable de recourir à la méthode de quasi-Newton, qui recourt à une approximation du Hessien de construction moins honéreuse.
Cette méthode n'est pas la plus adéquate pour la résolution des problèmes impliquant des systèmes d'équations complexes (pas de dérivées secondes explicites pour $\mathcal{J}o\mathcal{M}$) ou dépendants de nombreuses entrées.

– **Méthode 4 : Algorithme de Quasi-Newton.** La méthode de quasi-Newton, utilisée lorsque le Hessien est difficile ou coûteux à calculer, construit une approximation B^k de ce Hessien à l'aide des gradients successifs.

1.3.3 Méthodes de Quasi-Newton

1. **Initialisation.** Choix d'un point x^0, choix d'une matrice B^0 ($B^0 = Id$ en général).
2. **Itérés.** L'itéré x^k et l'approximation B^k servent à calculer l'itéré x^{k+1} et l'approximation B^{k+1} comme suit :
   a) Calcul de $\mathcal{J}(x^k)$ et $\nabla\mathcal{J}(x^k)$;
   b) Test du critère d'arrêt :
      i. Si le critère d'arrêt est satisfait :
      Arrêt de l'optimiseur, la solution est x^k.
      ii. Sinon :
      Calcul de la direction d^k comme solution de $B^k d^k = -\nabla\mathcal{J}(x^k)$,
      Calcul du pas ρ^k optimal par recherche linéaire,
      Calcul de B^{k+1},
      Calcul du nouvel itéré $x^{k+1} = x^k - \rho^k d^k$,
      Retour en (2).

Alogrithme de Quasi-Newton

La méthode de quasi-Newton utilise l'information apportée par le calcul de $\mathcal{J}$ et $\nabla\mathcal{J}$ entre les points x^k et x^{k+1} pour améliorer l'approximation B^{k+1} de la matrice hessienne H. La mise à jour de la méthode de Quasi-Newton est basée sur le théorème fondamental du calcul intégral :

Théorème 1 *Si* $y^k = x^{k+1} - x^k$ *et* $w^k = \nabla\mathcal{J}(x^{k+1}) - \nabla\mathcal{J}(x^k)$, *alors :*

$$\left(\int_0^1 \nabla^2\mathcal{J}(x^k + ty^k)dt\right)y^k = w^k. \tag{1.20}$$

La matrice $\left(\int_0^1 \nabla^2\mathcal{J}(x^k + ty^k)dt\right)$ peut être vue comme la moyenne de la matrice Hessienne sur le segment linéaire $[x^k, x^k+y^k] = [x^k, x^{k+1}]$. En conséquence, la matrice B^{k+1} approchant le Hessien est construite de manière à vérifier la

condition de quasi-Newton écrite ci-dessous :

$$B^{k+1}y^k = w^k. \tag{1.21}$$

Cette équation a plusieurs solutions. Par exemple, les mises à jour de Broyden s'écrivent :

$$B^{k+1} = B^k - \frac{B^k y^k (B^k y^k)^T}{(y^k)^T B^k y^k} + \frac{w^k (w^k)^T}{(w^k)^T y^k} + \phi^k [(y^k)^T B^k y^k] v^k (v^k)^T, \tag{1.22}$$

où $\phi^k \in [0, 1]$ et v^k est choisi tel que :

$$v^k = \frac{w^k}{(w^k)^T y^k} - \frac{B^k y^k}{(y^k)^T B^k y^k}. \tag{1.23}$$

Lorsque ϕ^k est nul, la formule (1.22) correspond à la mise à jour de Broyden, Fletcher, Goldfarb et Shannon, plus connue sous le nom BFGS. Des études ont montré ses intérêts théoriques et numériques. D'un autre côté, $\phi^k = 1$ correspond à la mise à jour proposée par Davidson, Fletcher et Powell appelée DFP. Le lecteur consultera le livre de Bonnans *et al.* [2002] pour plus de détails.

Remarque. Le cas particulier $B^k = Id$ correspond à une méthode de gradient à pas optimal.

1.3.4 Ordre des méthodes.

Une méthode d'optimisation est dite de :

1. premier ordre lorsqu'elle utilise la matrice Jacobienne $\nabla \mathcal{J}$ de $\mathcal{J}$. C'est le cas de la méthode de gradient à pas optimal et de la méthode du gradient conjugué ;

2. second ordre lorsqu'elle utilise la Jacobienne et le Hessien ($\nabla^2 \mathcal{J}$) ou une approximation du Hessien . C'est le cas de la méthode de Newton (Hessien) de convergence quadratique, et des méthodes de quasi-Newton (approximation du hessien) de convergence super-linéaire.

Comme on vient de le voir, ces méthodes d'optimisation ont besoin d'évaluer des gradients et des Hessiens. Ceux-ci peuvent être déterminés grâce à un outil de différentiation automatique. L'utilisation de la DA permet d'éviter de programmer manuellement les calculs des gradients de fonctions. C'est un gain important en terme de temps de

développement.

1.3.5 Recherche globale

Un algorithme de gradient peut donner de très mauvais résultats lorsque la fonction comporte de nombreux minima locaux. Trouver le minimum global est une tâche ardue et demande notamment de changer la fonction objectif en introduisant un terme de régularisation de Tikhonov par exemple.

1.3.6 Logiciels et bibliothèques d'optimisation

Une revue complète des logiciels et bibliothèques d'optimisation est proposée sur le serveur NEOS (`http://www-neos.mcs.anl.gov/neos/solvers/`) qui illustre l'optimisation numérique dans toute sa complexité. Le site présente, entre autres, les méthodes de gradient. Chaque méthode y est discutée théoriquement et numériquement. Des *solveurs en ligne* peuvent être testés sur des exemples académiques, voire parfois sur le problème de l'utilisateur.

a) Optimisation avec contraintes

Le minimum est cherché dans un espace des contrôles réel (de dimension égal à la dimension du vecteur de contrôle) et borné afin de contraindre, par exemple, la variable de contrôle à demeurer positive au cours des itérations. Des limites inférieures et supérieures différentes peuvent être imposées simultanément à chaque élément d'une variable de contrôle vectorielle. Le site NEOS offre la possibilité de tester les bibliothèques L-BFGS-B [Byrd *et al.*, 1995](détaillée ci-dessous), BLMVM [Benson et Moré, 2001] et TRON [Lin et Moré, 2006] qui travaillent, selon les cas, avec des codes écrits en C ou FORTRAN.

L-BFGS-B(Limited-BFGS-Bounded) Dans cette thèse nous utilisons cette bibliothèque écrite en FORTRAN 77 par Byrd *et al.* [1995], basée sur une méthode BFGS (BFGS : Broyden-Fletcher-Goldfarb-Shanno) avec mémoire limitée. Elle propose la résolution de problèmes d'optimisation de dimension n grande ayant éventuellement des contraintes simples sur certaines variables de contrôle. L-BFGS-B est libre d'utilisation. Elle se compose de :

1. trois fichiers appelés `driver1.f`, `driver2.f` et `driver3.f` contenant des exemples ;

2. un fichier `routines.f` qui contient les routines spécifiques à l'optimisation ;

3. un `Makefile` pour la compilation, et
4. plusieurs fichiers de documentation.

Le fichier `driver.f` (l'un des 3) sert de base lors de la conception de l'optimiseur de l'utilisateur. Dans ce fichier, largement documenté, figurent plusieurs paramètres et variables dépendant du problème inverse à résoudre :

1. `n`, la dimension du vecteur de contrôle ;
2. `x`, le vecteur de contrôle ;
3. `factr` et `pgtol`, les tolérances des critères d'arrêt de l'algorithme ;
4. `nbd(i)`, `l(i)` et `u(i)`, représentant respectivement le nombre de contraintes sur la variable `x(i)` et, le cas échéant, ses limites inférieure et supérieure.

Plus bas dans la routine `driver.f`, un commentaire signale le début de la boucle itérative. L'utilisateur intervient au niveau du calcul de `f` et de son gradient `g` en les remplaçant respectivement par son modèle et sa fonction objectif et les codes dérivés correspondants.

b) Optimisation sans contraintes

Le minimum est cherché dans un espace des contrôles réel de dimension égal à la dimension du vecteur de contrôle. Ces algorithmes sont un peu plus rapides que les précédents.

Le site NEOS offre la possibilité de tester les bibliothèques CGPLUS et NMTR qui travaillent, selon les cas, avec codes écrits en C ou FORTRAN.

MODULOPT - Écrite en FORTRAN 77 par Gilbert et Lemaréchal [1995], cette bibliothèque est basée sur une méthode BFGS avec mémoire limitée. Elle propose la résolution de problèmes d'optimisation sans contraintes de grande dimension. D'un point de vue pratique, son utilisation est très semblable à celle de la bibliothèque L-BFGS-B.

1.4 Conclusions

On a introduit dans ce chapitre les idées de base de la différentiation automatique dans l'optique de son utilisation en optimisation. Dans le cadre de cette thèse, et pour les deux applications traitées par la suite dans le chapitre 2 et le chapitre 3, on utilise l'outil TAPENADE de type "transformation de source" pour produire automatiquement les codes dérivés à fournir au pilote d'optimisation.

CHAPITRE 2

Identification de l'écrouissage des tôles métalliques à partir d'un essai de traction uni-axiale

Sommaire

2.1 Introduction 31
2.2 Matériaux élastoplastiques 33
2.2.1 Modélisation de l'écrouissage 35
2.2.2 Écrouissage isotrope 35
2.2.3 Écrouissage cinématique 35
2.2.4 Écrouissage mixte (isotrope et cinématique 36
2.2.5 Modélisation des surfaces de charge 37
2.2.6 Lois d'écrouissage 41
2.2.7 Essais de caractérisation de l'écrouissage 41
2.3 Identification de la loi d'écrouissage 48
2.3.1 Choix de la méthode d'identification 48
2.3.2 Résultats expérimentaux 49
2.3.3 Outils de simulation numérique 53
2.3.4 Mise en œuvre de l'identification 56
2.3.5 Validation de la différentiation 57
2.3.6 Etude de sensibilité des paramètres 58
2.3.7 Expériences Jumelles pour tester l'optimiseur 59
2.4 Résultats 60
2.5 Discussions 62

2.5.1 Courbes force-déplacement 62
2.5.2 Champs de déformation 66
2.6 Conclusions 69

2.1 Introduction

La caractérisation précise du comportement mécanique des matériaux au delà de leur limite élastique représente une phase préliminaire indispensable à la simulation des procédés de mise en forme. Dans la littérature, de très nombreux auteurs ont mené des travaux sur ce sujet afin de proposer des méthodes fiables et efficaces pour caractériser le comportement élastoplastique des matériaux métalliques. L'identification des paramètres pilotant ce comportement nécessite la réalisation d'essais mécaniques adaptés. Les essais pour lesquels la répartition spatiale du champ de contrainte est supposée connue indépendamment de la loi de comportement dans la " partie utile " des éprouvettes conduisent à une identification immédiate (par exemple, l'essai de traction uniaxiale). L'essai de traction présente toutefois certaines limitations liées au développement de la striction diffuse, puis de la striction localisée dans le cas d'éprouvettes de section rectangulaire issues de tôles.

La simulation numérique des procédés de mise des tôles métalliques ne cesse d'évoluer grâce au développement continu des outils numériques (tels que la méthode des éléments finis) et aux performances croissantes des calculateurs. Ces simulations nécessitent l'implémentation de modèles de lois de comportement qui doivent obéir aux lois générales de la mécanique, et qui doivent rendre compte de tous les aspects du comportement plastique : anisotropie, influence du trajet de chargement, écrouissage. On trouve deux grandes classes de modèles utilisés en calcul de structures métalliques. La première classe est celle des modèles phénoménologiques représentés par des fonctions constitutives qui relient des variables internes. Ces fonctions constitutives ont pour rôle de tenir compte de la structure interne du matériau ainsi que de l'histoire de la sollicitation. La deuxième classe consiste en des modèles micro-macro qui se basent sur des approches polycristallines permettant de tenir compte de la structure cristallographique des matériaux. L'objectif du présent travail est d'identifier un comportement en combinant des essais expérimentaux et des techniques numériques. De ce fait, on se limitera à la présentation des modèles phénoménologiques.

Pour effectuer des simulations numériques fiables des opérations de mise en forme des tôles métalliques, on a besoin d'avoir des paramètres précis qui sont classiquement identifiés à partir d'essais expérimentaux standard (traction uniaxiale), ou dédiés à l'analyse d'une sollicitation particulière (traction plane, cisaillement, traction biaxiale par gonflement hydraulique..). Ces approches consistent à réaliser ces essais en s'assurant de la validité de l'hypothèse d'homogénéité des champs de contraintes et de déformations dans l'éprouvette (traction uniaxiale, cisaillement) ou à utiliser certaines hypothèses condui-

sant à une formulation analytique qui permet de transformer les mesures expérimentales (force ou pression, déplacement, éventuellement mesures locales dans le cas du gonflement hydraulique) en valeurs de contraintes et de déformations. Ensuite, l'identification des paramètres est effectuée par un lissage du modèle choisi sur la courbe contrainte-déformation obtenue. Ces approches, que l'on peut qualifier de " méthodes directes ", souffrent de la forte limitation liée aux hypothèses qui permettent de s'affranchir de l'aspect " structure " de l'essai. Ainsi, les analyses de ce type d'essais expérimentaux peuvent mener à une identification paramétrique peu précise.

Une étape nouvelle a été franchie en proposant des essais volontairement hétérogènes, dont la réponse est plus riche que celle des essais standard. Le principe est d'effectuer un recalage d'une réponse mesurée et d'une réponse simulée par éléments finis. Cette méthode permet de prendre en considération quelques effets du matériau ou de la structure.

Au début de ces recherches, les travaux concernaient des mesures globales, telles l'effort appliqué, le déplacement d'ensemble de l'éprouvette ainsi que quelques mesures locales en utilisant des jauges de déformation. Depuis l'apparition des techniques de mesures de champs, les réponses mesurées ont été étendues aux champs de déplacements et de déformations sur une partie ou sur l'ensemble de l'éprouvette. Plusieurs techniques de mesures de champs existent, [Surrel, 2005] [Kobayashi, 1993] : corrélation de mouchetis, interférométrie de speckle, méthodes de grilles, moiré géométrique, interférométrique, etc. Le caractère non intrusif et sans contact de ces techniques offre la possibilité de mesurer des données cinématiques difficilement accessibles par d'autres moyens (éprouvette de petites dimensions, mesures dans une enceinte...). La technique la plus répandue fait appel à l'analyse d'images numériques, qui s'est développée grâce aux capteurs CCD (Charged Coupled Device) et aux techniques de traitement d'images automatisées. L'idée générale est alors de mettre en œuvre une méthode inverse d'identification combinant des résultats de simulation et des mesures expérimentales.

Dans ce cas, le problème d'identification des paramètres pilotant le modèle mathématique choisi pour décrire le comportement du matériau à partir de la réponse mécanique sur une structure donnée consiste à mettre en œuvre une procédure de recalage qui peut être schématisée par le graphe Fig. 2.1.

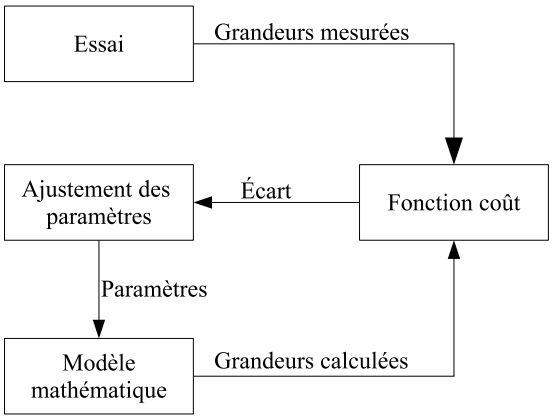

FIG. 2.1 – Etapes de l'identification par recalage (d'après Hendricks. [1991])

Nous présentons tout d'abord le cadre général des modèles phénoménologiques de l'élastoplasticité, puis les essais classiques d'identification " directe " du comportement et les essais proposés pour être exploités dans le cadre d'une identification inverse. Nous présentons ensuite le travail que nous avons réalisé, à savoir l'identification de l'écrouissage sur une courbe de traction uniaxiale complète, incluant la zone de chute de la force correspondant au développment de la striction.

2.2 Matériaux élastoplastiques

Afin de décrire le comportement des matériaux écrouissablesPour un modèle élastoplastique macroscopique, on doit :

1. Modéliser dans l'espace des contraintes σ_{ij}, pour un état donné du milieu, une surface Σ fermée, convexe, appelée surface de charge. Cette surface, d'équation $f = f(\sigma_{ij}, X_j) = 0$ où X_j sont les forces thermodynamiques associées à l'état actuel du matériau, permet de définir les conditions de chargement, élastique ou élastoplastique. Un accroissement infinitésimal de la contrainte $d\sigma_{ij}$ mène aux différentes situations représentées sur la figure 2.2.

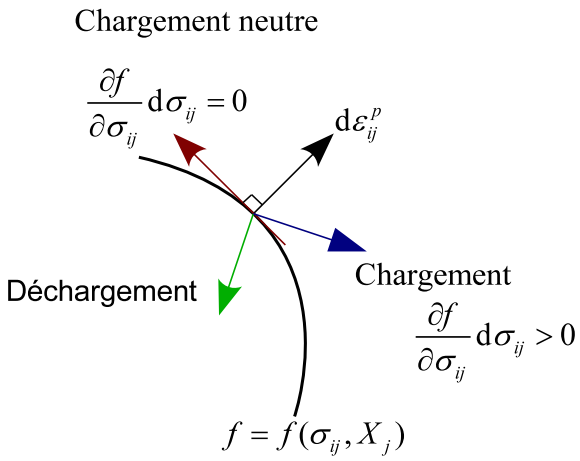

FIG. 2.2 – Les différentes possibilités de chargement

Chargement élastique : $f < 0$.
Déchargement : $f = 0$; $\dfrac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} < 0$; l'incrément est dirigé vers l'intérieur de la surface de charge.
Chargement neutre : $f = 0$; $\dfrac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} = 0$; l'incrément de chargement est tangent à la surface de charge : il n' y a pas d'écrouissage.
Chargement plastique : $f = 0$; $\dfrac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} > 0$; l'incrément de contrainte est dirigé vers l'extérieur de la surface de charge : la surface de charge évolue, traduisant le phénomène d'écrouissage.
La surface de charge pour l'état initial du matériau s'identifie à la notion de surface de limite élastique.

2. Modéliser l'écrouissage afin de décrire l'évolution de la forme, de la taille et de la position de la surface de charge au cours du chargement. Deux grandes classes d'écrouissage ont été proposées : l'écrouissage isotrope et l'écrouissage cinématique.

3. Définir la loi d'écoulement plastique reliant l'écoulement plastique à l'état de contraintes auquel est soumis le matériau. La description macroscopique de l'écoulement plastique obéit en général à la théorie de l'écoulement utilisant le principe du travail maximal de Hill [1950]. Ce principe conduit à la convexité de la surface de charge par rapport aux variables σ_{ij} et X_j. Les incréments de déformation plastique $d\varepsilon_{ij}^{p}$ et des variables internes $d\alpha_j$ sont données alors par l'expression suivante :

$$\begin{cases} d\varepsilon_{ij}^{p} &= d\lambda \dfrac{\partial f}{\partial \sigma_{ij}}, \\ d\alpha_j &= -d\lambda \dfrac{\partial f}{\partial X_j}, \end{cases} \tag{2.1}$$

où dλ est un scalaire positive appelé le multiplicateur plastique.

2.2.1 Modélisation de l'écrouissage

2.2.2 Écrouissage isotrope

[Taylor et Quinney, 1931]. Lors de la déformation plastique, ce mode d'écrouissage se traduit par une simple dilatation de la surface de charge symétriquement autour de l'origine Fig. 2.3.

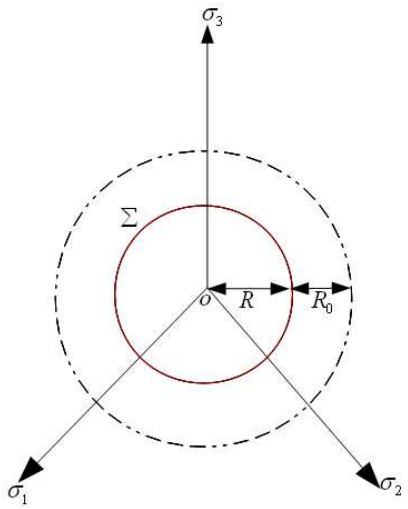

FIG. 2.3 – Écrouissage isotrope (représentation dans le plan déviatoire)

L'équation de la surface de charge s'écrit alors sous la forme :

$$f(\boldsymbol{\sigma}, R) = J(\boldsymbol{\sigma}) - R - R_0. \tag{2.2}$$

La grandeur scalaire R représente l'évolution de la surface de charge.

2.2.3 Écrouissage cinématique

[Prager, 1955]. Pour une surface de charge Σ de centre o, lorsqu'on charge le matériau, la limite élastique augmente dans la direction de la déformation (trajet OA) marquant ainsi l'écrouissage qui s'est produit. Par contre dans la direction inverse (trajet OB), la limite élastique diminue, ce qui explique en partie l'effet BAUSHINGER Fig. 2.4.

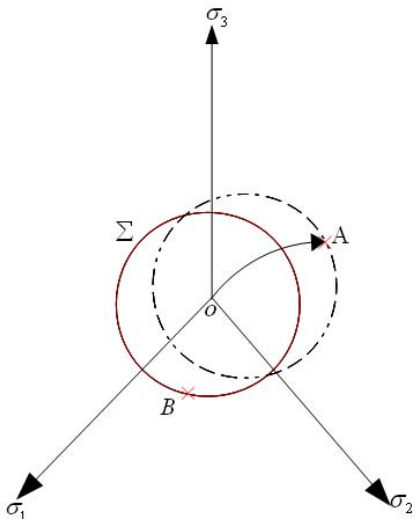

FIG. 2.4 – Écrouissage cinématique (représentation dans le plan déviatoire)

L'équation de la surface de charge prend la forme :

$$f(\boldsymbol{\sigma}, \boldsymbol{X}) = J(\boldsymbol{\sigma}, \boldsymbol{X}) - R_0. \tag{2.3}$$

La grandeur tensorielle $\boldsymbol{X}$ représente les micro-contraintes internes engendrées par l'incompatibilité des différents grains du matériau.
La fonction de charge est paramétrée par la variable $\boldsymbol{X}$, qui elle même est associée à la variable extensive α désignant la variable cachée (interne) qui caractérise l'écrouissage cinématique.

2.2.4 Écrouissage mixte (isotrope et cinématique

pour un milieu initialement isotrope, la combinaison des deux types d'écrouissage conduit à un écrouissage isotrope et cinématique (Fig. 2.5) dont la fonction de charge est définie comme suit :

$$f(\boldsymbol{\sigma}, \boldsymbol{X}, R) = J(\boldsymbol{\sigma}, \boldsymbol{X}) - R - R_0. \tag{2.4}$$

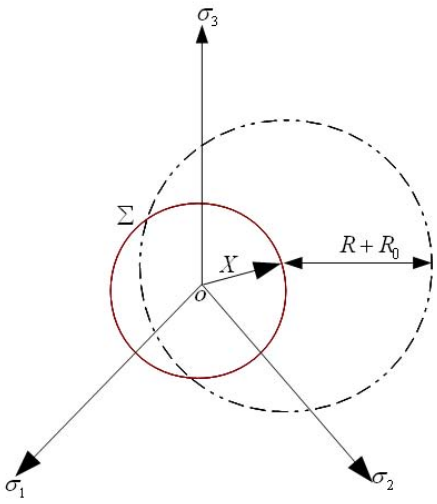

FIG. 2.5 – Écrouissage isotrope et cinématique (représentation dans le plan déviatoire)

2.2.5 Modélisation des surfaces de charge

a) Critères de plasticité isotropes

Un matériau est dit isotrope lorsqu'il présente les mêmes propriétés mécaniques dans toutes les directions. Pour ce type de matériau, on atteint le régime plastique pour des états de contraintes indépendants de la direction des contraintes principales. De ce fait, le critère de plasticité peut être exprimé sous les différentes formes équivalentes suivantes :

$$\begin{cases} f & = F(\sigma_1, \sigma_2, \sigma_3) - \bar{\sigma} = 0, \\ \text{ou } f & = F(I_1, I_2, I_3) - \bar{\sigma} = 0, \\ \text{ou } f & = F(I_1, J_2, J_3) - \bar{\sigma} = 0, \end{cases} \tag{2.5}$$

où I_1, I_2, I_3 sont les invariants du tenseur des contraintes $\boldsymbol{\sigma}$, et $J_1 = 0$, J_2, J_3, sont les invariants du déviateur des contraintes $\boldsymbol{s} = \boldsymbol{\sigma} - \sigma_m \boldsymbol{I}$, avec $\sigma_m = I_1/3$, soit :

$$J_1 = tr(\boldsymbol{\sigma}) = \sigma_{ii},\ J_2 = [tr(\boldsymbol{s})^2]^{1/2} = [s_{ij}s_{ij}]^{1/2},\ J_3 = [tr(\boldsymbol{s})^3]^{1/3} = [s_{ij}s_{jk}s_{ki}]^{1/3}. \tag{2.6}$$

La contrainte équivalente $\bar{\sigma}$ dans les équations (2.5) remplace la grandeur $(R + R_0)$ dans l'équation (2.2). Elle est identifiée comme la contrainte mesurée dans un essai de référence, souvent l'essai de traction uniaxiale.

Comme les métaux sont plastiquement incompressibles, l'écoulement plastique est insensible à une sollicitation hydrostatique : l'invariant I_1 n'intervient pas dans les critères

de plasticité qui se présentent finalement sous la forme :

$$\begin{cases} f = F(J_2, J_3) - \bar{\sigma} = 0, \\ \text{ou } F(J_2, J_3) = \bar{\sigma}. \end{cases} \tag{2.7}$$

Dans l'espace des contraintes principales $(\sigma_1, \sigma_2, \sigma_3)$, les critères de plasticité sont représentés par une surface cylindrique d'axe Δ parallèle à la trisectrice du repère $(\sigma_1, \sigma_2, \sigma_3)$. Les surfaces associées à ces critères peuvent être caractérisées par leur trace : soit sur le plan du déviateur, d'équation : $\sigma_{11} + \sigma_2 + \sigma_3 = 0$, soit sur le plan de contraintes planes d'équation : $\sigma_3 = 0$. Parmi les critères de plasticité isotrope les plus utilisés on trouve :

Le critère de Tresca (1864) : C'est le premier critère utilisé pour les matériaux métalliques. Selon ce critère, le matériau atteint la limite élastique lorsque la contrainte de cisaillement maximale atteint une valeur critique k. La surface de plasticité est déterminée par :

$$\text{Max}[\frac{1}{2}\left|\sigma_{11} - \sigma_{22}\right|; \frac{1}{2}\left|\sigma_{22} - \sigma_{33}\right|; \frac{1}{2}\left|\sigma_{11} - \sigma_{33}\right|] = k = \frac{\bar{\sigma}}{2}, \tag{2.8}$$

où k représente la limite élastique en cisaillement pur et $\bar{\sigma}$ représente la limite élastique en traction uni-axiale. Ce critère est représenté dans l'espace des contraintes principales $(\sigma_{11}, \sigma_{22}, \sigma_{33})$ par un prisme droit d'axe Δ (voir Fig. 2.6) ayant pour base un hexagone régulier dans le plan déviatoire (voir Fig. 2.7).

Le critère de Von Mises (1913) : Ce critère est le plus utilisé dans le cas d'un comportement isotrope pour les matériaux métalliques ductiles. Il revient à supposer que la fonction de charge ne dépend que de l'invariant J_2. La surface de plasticité s'exprime sous la forme :

$$f(J_2) = 0. \tag{2.9}$$

Différentes expressions de ce critère sont usuellement utilisées :

$$\begin{cases} f & = \sqrt{\frac{1}{2} s_{ij} s_{ij}} - \tau = 0, \\ \text{ou } f & = \sqrt{\frac{3}{2} s_{ij} s_{ij}} - \bar{\sigma} = 0, \\ \text{ou } \bar{\sigma}^2 & = (\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 = (\sigma_{33} - \sigma_{11})^2 + 6(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2), \end{cases} \tag{2.10}$$

où τ est la limite élastique en cisaillement pur et $\bar{\sigma}$ est la limite élastique en traction uni-axiale. La troisième expression est une représentation quadratique du critère de von Mises.

Dans un contexte d'écrouissage isotrope et cinématique, l'utilisation du critère de Von Mises conduit à :

$$f(\boldsymbol{\sigma}, \boldsymbol{X}, R)) = \sqrt{(\boldsymbol{s} - \boldsymbol{X}) : (\boldsymbol{s} - \boldsymbol{X})} - R - R_0. \tag{2.11}$$

Ce critère simple d'utilisation arrive à décrire le comportement du matériau d'une manière acceptable et raisonnable. Il est implémenté dans le code éléments finis utilisé dans le présent travail de thèse.

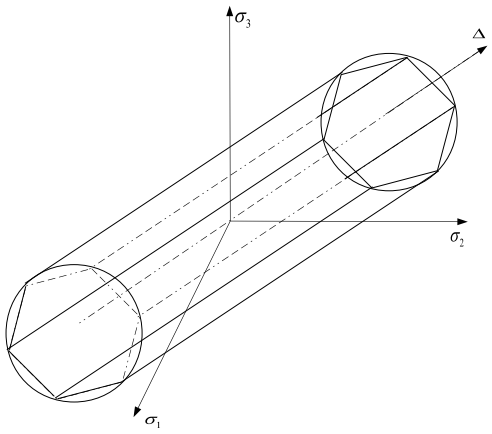

FIG. 2.6 – Représentation de la surface de plasticité correspondant aux critères de Tresca et de Von Mises, dans l'espace des contraintes principales.

Le critère de Drucker 1949 : Dans le plan déviatoire, ce critère est situé entre l'hexagone de Tresca et le cercle de Von Mises. Il est exprimé en fonction du deuxième et du troisième invariant J_2 et J_3 comme suit :

$$F(J_2, J_3) = (3J_2)^3[1 - C(\frac{J_3^2}{J_2^3})] = [1 - (\frac{4C}{27})]\sigma_0^6, \tag{2.12}$$

où C est un paramètre qui caractérise le matériau et σ_0 est la limite élastique en traction ou en compression uni-axiale. La condition de convexité est vérifiée pour $C \leq \frac{9}{4}$. Dans le plan déviatoire, le critère de Drucker est représenté par la figure 2.7.

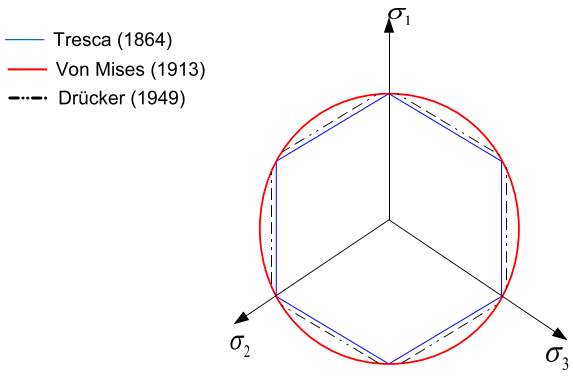

FIG. 2.7 – Représentation des surfaces de plasticité correspondants aux critères de Von Mises, Tresca et Drücker, dans le plan déviatorique (d'après Karafilis et Boyce. [1993])

Le critère Hosford (1972) : C'est un critère isotrope non quadratique. Il a été proposé pour décrire le comportement des structures cristallographiques cubiques centrées (CC) et cubiques à faces centrées (CFC) des matériaux isotropes. Formellement, ce critère se présente comme une extension du critère de Von Mises :

$$F = |s_{11} - s_{22}|^m + |s_{22} - s_{33}|^m = |s_{33} - s_{11}|^m - 2\bar{\sigma}^m = 0. \tag{2.13}$$

Le coefficient m est un paramètre matériel, dont l'estimation peut être basée sur des résultats de calculs polycristallins.

b) Critères de plasticité anisotropes

Les tôles métalliques présentent une anisotropie initiale qui résulte de leur élaboration par laminage. Il s'agit d'une anisotropie particulière, appelée orthotropie, et qui corresond à l'existence de 3 plans de symétrie orthogonaux définis par la direction de laminage (x), la direction transverse (y), et la direction de l'épaisseur (z). L'expression d'un critère de plasticité orthotrope doit donc faire référence au repère (x, y, z) ; appelé repère d'orthotropie.
De très nombreux critères de plasticité orthotropes ont été développés au cours des dernières décennies. Nous limitons cette présentation au critère quadratique de Hill [1948], qui reste le plus utilisé en contexte industriel.

Le critère quadratique de Hill (1948) : Proposé par Hill [1948], ce critère quadratique s'exprime dans le repère d'orthotropie (x, y, z) par :

$$2F(\sigma_{ij}) = F(\sigma_{yy} - \sigma_{zz})^2 + G(\sigma_{zz} - \sigma_{xx})^2 + H(\sigma_{xx} - \sigma_{yy})^2 + 2L\sigma_{yz}^2 + 2M\sigma_{zx}^2 + 2N\sigma_{xy}^2 = 1. \tag{2.14}$$

Il apparait comme une extension du critère de von Mises. Les coefficients peuvent être déterminés à partir de différents essais. L'essai le plus couramment utilisé consiste à mesurer les coefficients d'anisotropie R_0, R_{45} et R_{90}, définis comme les rapports de vitesse de déformation en largeur et en épaisseur dans des essais de traction uniaxiale à 0°, 45° et 90° de la direction de laminage, respectivement. Des essais spécifiques doivent être conçus pour accéder aux coefficients M et N.

2.2.6 Lois d'écrouissage

L'écrouissage est le plus souvent identifié dans des essais de chargement monotones. L'identification ainsi réalisée fournit alors une loi scalaire entre une contrainte équivalente ($\bar{\sigma}$) et la déformation plastique équivalente associée ($\bar{\varepsilon}^p$), ce qui correspond à un modèle d'écrouissage isotrope. Sinon, il reste à envisager des essais supplémentaires pour découpler les parts d'écrouissage isotrope et cinématique dans la réponse observée en chargement monotone.

Dans le cadre de l'écrouissage isotrope, les lois d'écrouissage $\bar{\sigma}(\bar{\varepsilon}^p)$ les plus utilisées sont résumées dans le tableau 2.1.

Loi d'écrouissage	$\bar{\sigma}(\bar{\varepsilon}^p)$	***Remarques***
Loi de Hollomon	$\bar{\sigma} = K(\bar{\varepsilon}^p)^N$ où N est l'exposant d'écrouissage et K est un paramètre de matériau.	Limite élastique nulle (la limite élastique doit être faible devant l'écrouissage).
Loi de Swift	$\bar{\sigma} = K(\varepsilon_0 + \bar{\varepsilon}^p)^N$ où $K\varepsilon_0^N$ est la limite élastique.	Bien adaptée aux aciers doux, et aciers faiblement alliés.
Loi de Ludwik	$\bar{\sigma} = \sigma_E + K(\bar{\varepsilon}^p)^N$ où σ_E est la limite élastique.	Fort écrouissage à faible déformation.
Loi de Voce	$\sigma_s(1 - \alpha\exp(\beta\bar{\varepsilon}^p))$ où σ_s est la contrainte de saturation et $0 < \alpha < 1$ et $\beta < 0$.	Bien adaptée aux alliages d'aluminium. Utilisable à haute température.

TAB. 2.1 – Lois d'écrouissage usuelles.

2.2.7 Essais de caractérisation de l'écrouissage

Pour identifier les paramètres d'une loi de comportement, il est indispensable de réaliser des essais expérimentaux, analysés par des procédures d'identification directe ou inverse. Ces essais peuvent aussi être comparés entre eux pour valider une loi de comportement plus générale que celle qui est strictement nécessaire pour décrire une seule

sollicitation ou un seul essai. La comparaison des résultats obtenus pour différents essais correspondant à différents trajets de chargement est susceptible de fournir des informations sur la surface de charge.

a) Essai de traction uni-axiale

C'est un essai mécanique simple à réaliser. Ces essais ont fait l'objet de normalisation, par exemple la norme ISO 6892 pour les essais de traction sur des matériaux métalliques à température ambiante. Dans le cas des tôles minces (épaisseur $< 3mm$), la courbe de traction rationnelle, appelée aussi courbe d'écrouissage, est la représentation graphique de la contrainte vraie (contrainte de Cauchy) appliquée à l'éprouvette en fonction de la déformation rationnelle (déformation de Hencky). Ces grandeurs sont obtenues directement, tant que la contrainte et la déformation sont supposées homogènes, c'est-à-dire jusqu'au point de force maximum. La contrainte et la déformation uni-axiales s'obtiennent à partir des résultats de l'essai, donnant la force (F) en fonction de l'allongement ($L - L0$), par les relations :

$$\sigma = \frac{F}{S} = \frac{F}{S_0}\exp^{\varepsilon}, \quad \text{et } \varepsilon = ln(\frac{L}{L_0}), \tag{2.15}$$

où F est la force appliquée, S_0 est la section initiale de l'éprouvette, ε est la déformation logarithmique et L_0 et L représentent respectivement la longueur initiale et actuelle de la zone utile de l'éprouvette en traction (voir Fig. 2.8). La déformation plastique ε_p peut être obtenue en retranchant la déformation élastique ε_e de la déformation totale.

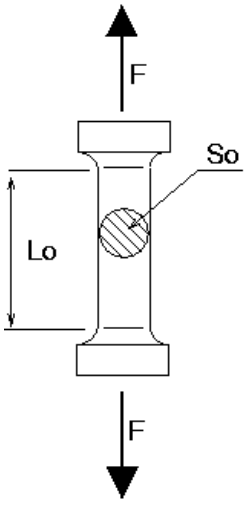

FIG. 2.8 – Essai de traction uni-axiale

La détermination de la courbe d'écrouissage $\sigma = f(\varepsilon)$, à partir des grandeurs globales

force-déplacement fournies par la machine de traction, n'est valable que si on considère l'hypothèse d'homogénéité des champs de contrainte et de déformation dans la partie utile de l'éprouvette. A partir de cet essai classique et largement connu, on détermine les caractéristiques mécaniques fondamentales telles que le module de Young, la limite élastique, la résistance à la rupture, l'allongement à la rupture, ainsi que des grandeurs telles que le coefficient d'écrouissage $\gamma = \dfrac{\mathrm{dLn}\sigma}{\mathrm{d}\varepsilon}$, ou des caractéristiques de l'anisotropie, comme le coefficient de Lankford.

L'apparition de la striction diffuse est le premier évenement limitant le domaine d'analyse directe de l'essai comme étant un essai homogène. Ce phénomène se manifeste par l'apparition d'un étranglement affectant la section droite de l'éprouvette, à partir du point de force maximum, telque :

$$\mathrm{soitdF} = \mathrm{d}(\sigma S) = 0,\ \mathrm{soit}\frac{\mathrm{d}\sigma}{\sigma} + \frac{\mathrm{dS}}{S} = 0. \tag{2.16}$$

A l'aide de la relation $\dfrac{\mathrm{d}S}{S} = -\dfrac{\mathrm{d}L}{L} = -\mathrm{d}\varepsilon$ résultant de l'incompressibilité plastique, on peut ainsi associer l'apparition de la striction diffuse à la valeur du coefficient d'écrouissage $\gamma = \frac{\mathrm{dLn}\sigma}{\mathrm{d}\varepsilon} = 1$. En particulier, pour une simple loi d'écrouissage de Hollomon, $\sigma = K\varepsilon^N$, on trouve que la striction diffuse apparaît pour une déformation $\varepsilon = N$.

Le second événement avant rupture est la striction localisée qui se manifeste sous la forme d'une bande de striction affectant l'épaisseur de l'éprouvette. Dans une analyse bi-dimensionnelle, l'apparition de cette bande est possible le long d'une ligne d'extension nulle dans le plan de la tôle, ce qui conduit à l'inclinaison de cette bande de striction. Les équations d'équilibre selon la direction du plan de la tôle normale à la bande permettent ensuite de montrer que :

$$\frac{\mathrm{d}\sigma}{\sigma} + \mathrm{d}\varepsilon_3 = 0, \tag{2.17}$$

où ε_3 est la déformation en épaisseur. En particulier, pour une loi d'écrouissage de Hollomon, la striction localisée apparaît en traction uniaxiale pour une déformation $\varepsilon = 2N$.

b) Essai de traction large

L'essai de traction sur des éprouvettes de grande largeur par rapport à leur longueur peut être envisagé pour obtenir des états proches de la traction plane dans la section médiane de l'éprouvette, comme cela a été propose par Wagoner [1981] ((Fig. 2.9)). Avec l'hypothèse de traction plane, on peut donc obtenir une estimation de la courbe contrainte-déformation en traction plane à partir de la force de traction et de la mesure locale de

la déformation axiale dans la section médiane. Une étude effectuée par Dournaux *et al.* [2009] montre toutefois que des états de contrainte et de déformation uniformes ne sont pas obtenus dans la section médiane. L'étude des champs de déformation montre de plus que le degré d'inhomogénéité des champs de déformation dépend du choix de la géométrie. Enfin, des simulations numériques montrent que l'inhomogénéité des champs dépend de la fonction de charge utilisée.
En conclusion, les résultats de cet essai pourraient plutôt être utilisés dans le cadre d'une identification inverse du comportement ([Dournaux *et al.*, 2009]).

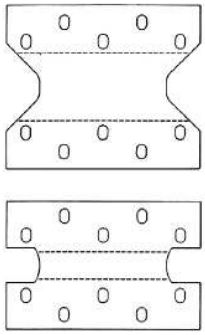

FIG. 2.9 – Éprouvettes proposées par Wagoner [1981] pour des essais de traction large

c) Essai de cisaillement simple

L'essai de cisaillement simple peut être réalisé sur une machine d'essai conventionnelle au moyen d'un dispositif imposant un déplacement de deux mors latéraux parallèles, conduisant à une déformation de glissement γ (Fig. 2.11). Par rapport à l'essai de traction, des déformations homogènes beaucoup plus importantes peuvent être atteintes. Un chargement inverse peut également être facilement imposé en inversant le sens de déplacement des mors. Enfin, on peut aussi utiliser des éprouvettes pré-déformées avec une autre sollicitation, ce qui permet d'accéder à des informations sur l'anisotropie induite (écrouissage non isotrope), comme cela est illustré sur la figure 2.11([Rauch, 1998]).

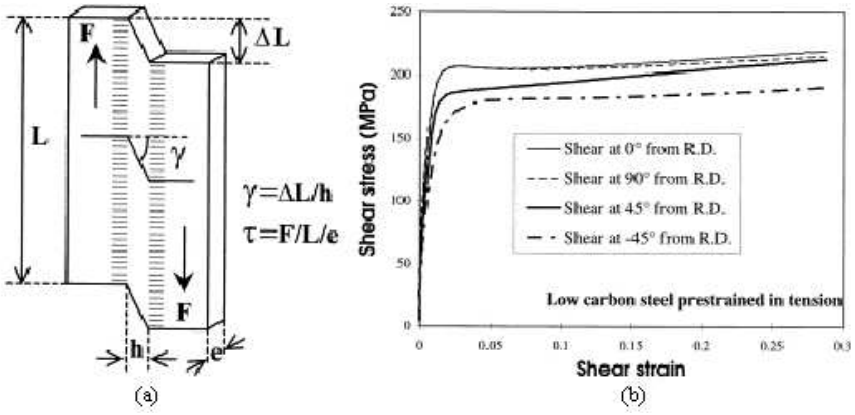

FIG. 2.10 – Géométrie dans un essai de cisaillement simple (a) et courbe contrainte-déformation obtenue après une pré-déformation de 20% en traction (b) (d'après Rauch [1998])

d) Essai de gonflement hydraulique

Suite au travail de Duncan [1965],le gonflement hydraulique d'une tôle bloquée le long d'un contour circulaire et soumise sur une face à l'action d'un fluide sous pression est largement utilisé dans de nombreux laboratoires industriels et académiques. Ce test présente l'avantage d'obtenir de fortes déformations au pôle du dôme ainsi formé. La contrainte biaxiale σ_b au pôle est estimée en mesurant la pression de fluide et en calculant l'épaisseur et la courbure par des moyens appropriés. Les déformations superficielles ε_1 et ε_2 et les courbures $1/R_1$ et $1/R_2$ le long des directions d'anisotropie (direction de laminage et direction transverse) sont déterminées comme les valeurs moyennes obtenues sur une certaine surface autour du pôle, par des moyens mécaniques ou par des moyens optiques permettant d'enregistrer les coordonnées d'une grille appliquée sur la surface de la tôle.

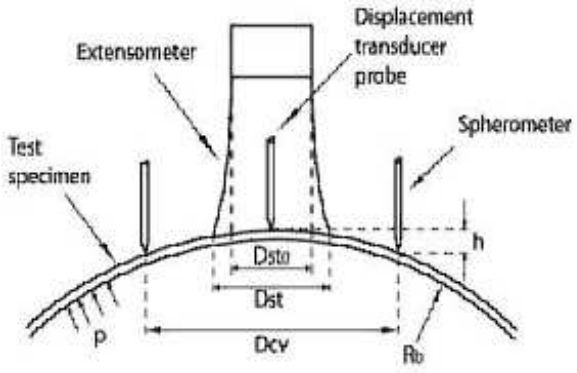

FIG. 2.11 – Géométrie d'un essai de gonflement hydraulique instrumenté pour la mesure de la déformation et de la courbure au pôle (d'après Santos *et al.* [2010])

La déformation ε_3 est estimée à partir de l'hypothèse d'incompressibilité :

$$\varepsilon_3 = -(\varepsilon_1 + \varepsilon_2), \tag{2.18}$$

et l'épaisseur t est ensuite calculée :

$$t = t_0 \exp(\varepsilon_3), \tag{2.19}$$

où $t_0$est l'épaisseur initiale.

Enfin, l'équation d'équilibre de membrane est utilisée pour déterminer les contraintes de membrane σ_1 et σ_2 :

$$\frac{\sigma_1}{R_1} + \frac{\sigma_2}{R_2} = \frac{P}{t}. \tag{2.20}$$

La forme locale de la tôle déformée autour du pôle est souvent approchée par une sphère. Dans ce cas, avec les hypothèses $R_1 \approx R_2 \equiv R$ et $\sigma_1 \approx \sigma_2 \equiv \sigma_b$, l'équation (2.20) se ramène à :

$$\sigma_b = \frac{pR}{2t}. \tag{2.21}$$

En plus de la précision expérimentale, plusieurs éléments sont susceptibles de nuire à la détermination de la loi d'écrouissage sous chargement biaxial :

- la déformation et la courbure dépendent de la position le long du profil, de sorte que les valeurs calculées des déformations et courbures au pôle dépendent de la taille de la zone utilisée pour leur détermination ;
- la détermination de l'épaisseur à partir des déformations mesurées sur la face supérieure du dôme ne prend pas en compte les gradients de déformation au travers de l'épaisseur, qui résultent de la combinaison d'efforts de traction et de flexion pour les tôles épaisses ;
- l'utilisation de l'équation d'équilibre de membrane représente une approximation pour des tôles épaisses.

Ces difficultés ont fait l'objet d'études numériques [Santos *et al.*, 2010, Lemoine *et al.*, 2011]. Dans ces études, des expériences virtuelles ont été réalisées en appliquant la procédure expérimentale de détermination des courbes d'écrouissage à des résultats de simulation numérique. La comparaison avec la loi d'écrouissage introduite dans les simulations confirme que la méthode d'analyse des résultats expérimentaux conduit à des déviations par rapport à la loi recherchée. La flexion de la tôle conduit à une sous-estimation des

contraintes au début de la déformation. Lorsque le gonflement progresse, une surestimation de la courbe d'écrouissage est ensuite observée, en partie en raison d'une sous-estimation progressive de l'épaisseur, calculée avec la valeur de déformation superficielle sur la face supérieure.

e) Méthodes inverses

Une démarche nouvelle dans le cadre de l'identification du comportement plastique a été initiée par des auteurs comme Gavrus *et al.* [1996], Mahnken et Stein [1997], Ghouati et Gelin [1998], Yoshida *et al.* [1998], Yoshida *et al.* [2003], Diot *et al.* [2003]. Ces auteurs ont proposé dans un premier temps d'étendre le domaine d'analyse des essais classiques en effectuant un recalage d'une réponse mesurée et d'une réponse simulée par éléments finis. Cette méthode permet de prendre en considération quelques effets du matériau ou de la structure.

Schnur et Zabaras [1992] sont parmi les premiers à avoir tenté de surmonter l'hypothèse d'homogénéité, en couplant un code de calcul par éléments finis avec une méthode d'optimisation afin d'identifier le comportement d'un matériau. Cette méthode inverse, qui consiste à minimiser par un procédé itératif l'écart entre la simulation et l'expérience, n'a cessé de se développer au cours de ces dernières années.

Gavrus *et al.* [1996] ont appliqué une méthode inverse pour identifier le comportement rhéologique d'un matériau thermoviscoplastique sollicité en traction et en torsion à des conditions sévères de déformation proches de celles atteintes pour des applications industrielles. Ces auteurs ont développé un algorithme qui couple la méthode des éléments finis simulant l'essai avec un module d'optimisation. La fonction objectif, formulée dans le sens des moindres carrés et présentant la différence entre les mesures simulées et expérimentales, est minimisée en utilisant la procédure de Gauss-Newton.

Mahnken et Stein [1996] ont pris en compte des mesures locales de déplacement dans la fonction coût à minimiser. Ils ont effectué un essai de traction sur une éprouvette plane en acier entaillée en son milieu. La méthode de corrélation d'image est ensuite utilisée pour mesurer les champs de déplacements au voisinage de l'entaille. L'identification des paramètres d'écrouissage et la limite élastique des deux lois de comportements élastoplastiques proposées est réalisée en minimisant l'écart entre les mesures et la simulation. Pour résoudre le problème de minimisation la méthode du gradient calculé analytiquement est utilisée.

Meuwissen *et al.* [1998] ont identifié des modèles de plasticité en utilisant des essais plans hétérogènes. La fonction coût à minimiser est l'écart quadratique portant sur l'effort appliqué et les champs de déplacement, dans les zones où les gradients sont importants,

simulés et mesurés expérimentalement. Les auteurs utilisent des coefficients de pondération dans la fonction coût afin de prendre en compte les erreurs de mesures et la dispersion des paramètres. Pour minimiser la fonction coût, une méthode de premier ordre de type Gauss-Newton est alors utilisée. Le calcul de gradient est effectué par différence finis.

Kajberga *et al.* [2004] effectuent des essais de traction sur des éprouvettes planes entaillées. Pour faciliter la localisation de la plasticité, ils mènent leurs essais jusqu'à la rupture des éprouvettes. La méthode d' interférométrie de speckle est utilisée pour mesurer les champs de déplacement. La fonction coût est choisie comme étant la différence entre les valeurs issues de la simulation et des essais expérimentaux. Finalement le problème d'optimisation est résolu par une méthode de simplexe.

L'idée générale de ces méthodes inverses combinant des résultats de simulation et des mesures expérimentales reste la même pour tous les auteurs, la différence se manifeste dans le choix des méthodes d'optimisation. On trouve principalement les méthodes de la plus grande pente, du gradient conjugué, de Quasi-Newton, de Gauss-Newton et de Levenberg-Marquart. Pour ces méthodes, il est nécessaire de calculer le gradient de la fonction objectif par rapport aux paramètres à identifier. Usuellement les techniques de différences finies, de dérivation semi-analytique ou analytique sont utilisées. Pour les méthodes d'optimisation d'ordre zéro comme la méthode de simplexe et les algorithmes génétiques on n'a pas à calculer le gradient, des évaluations successives de la fonction objectif suffisant.

2.3 Identification de la loi d'écrouissage

2.3.1 Choix de la méthode d'identification

La simulation par éléments finis de la traction simple sur un seul élément conduit à reproduire strictement la loi d'écrouissage introduite dans la simulation. Par contre, la simulation d'un essai de traction réel, avec des conditions de traction plane au niveau des mors de fixation de l'éprouvette, conduit progressivement à une localisation de la déformation, avec les deux phénomènes successifs de striction diffuse et de striction localisée prévus théoriquement comme des instabilités matérielles. Ces phénomènes apparaissent alors comme des phénomènes de localisation, engendrés par les conditions d'encastrement aux extrémités qui génèrent des champs de contraintes et de déformation de plus en plus hétérogènes et conduisent progressivement à une localisation de la déformation au centre de l'éprouvette. Nous avons donc envisagé d'identifier l'écrouissage à partir d'une méthode inverse consistant à minimiser l'écart entre les courbes force-déplacement expérimentale et calculées dans un essai de traction normalisé en exploitant les résultats expérimentaux

jusque dans le domaine final de chute rapide de la force.

2.3.2 Résultats expérimentaux

La campagne d'essais a été réalisée dans les laboratoires d'ArcelorMittal. On réalise une caractérisation en traction uni-axiale selon la norme $ISO6892-1$. Les deux matériaux testés sont un acier multiphasé à très haute résistance et un acier doux pour l'emboutissage. Les essais de traction uni-axiale sont réalisés à l'aide d'une machine de traction compression ZWICK Roell (voir (Fig. 2.12)).

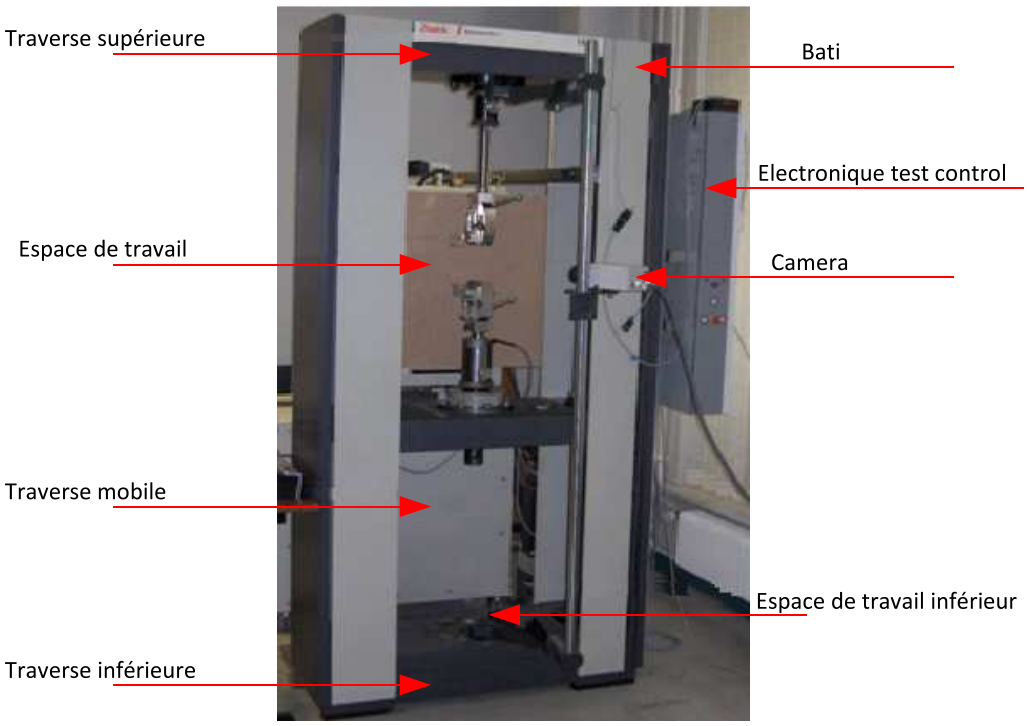

FIG. 2.12 – Machine de traction compression ZWICK Z100

La géométrie des éprouvettes, d'épaisseur moyenne de $1.5mm$, utilisées dans cet essai est représentée dans la figure 2.13.

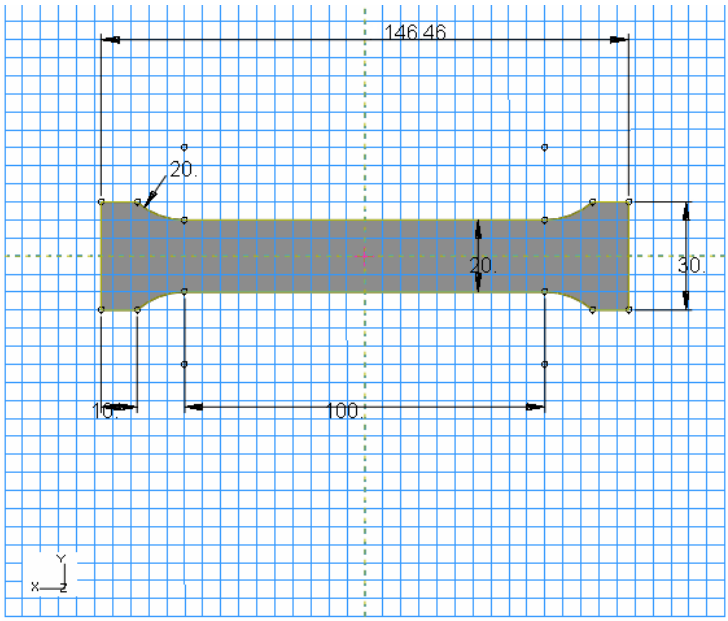

FIG. 2.13 – Les éprouvettes utilisées

Les courbes force-allongement pour l'essai de traction uni-axiale réalisé selon la direction de laminage sont données sur les figures 2.14 et 2.22, pour les aciers multiphasé et doux, respectivement.

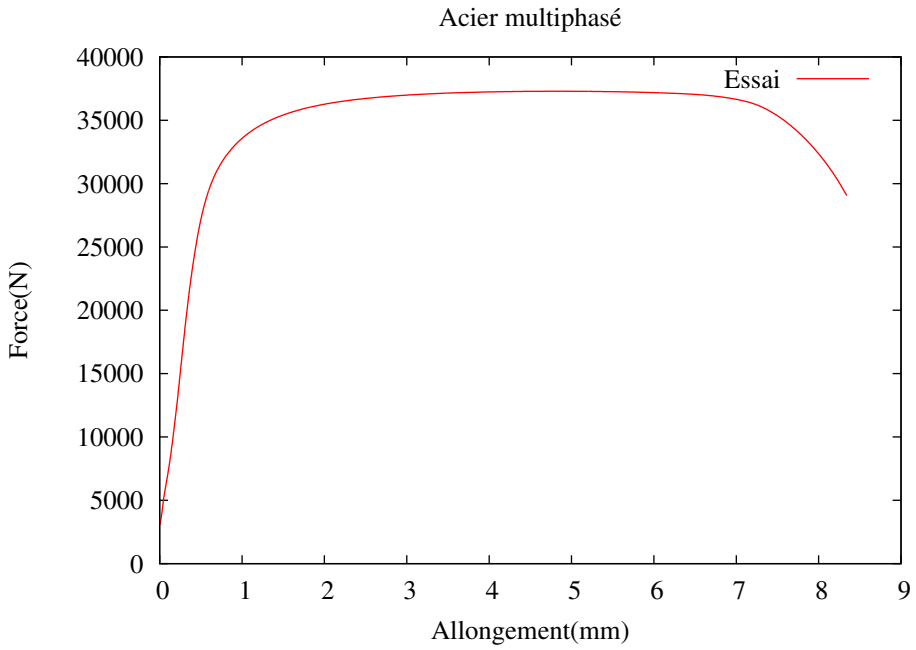

FIG. 2.14 – Courbe de traction uni-axiale

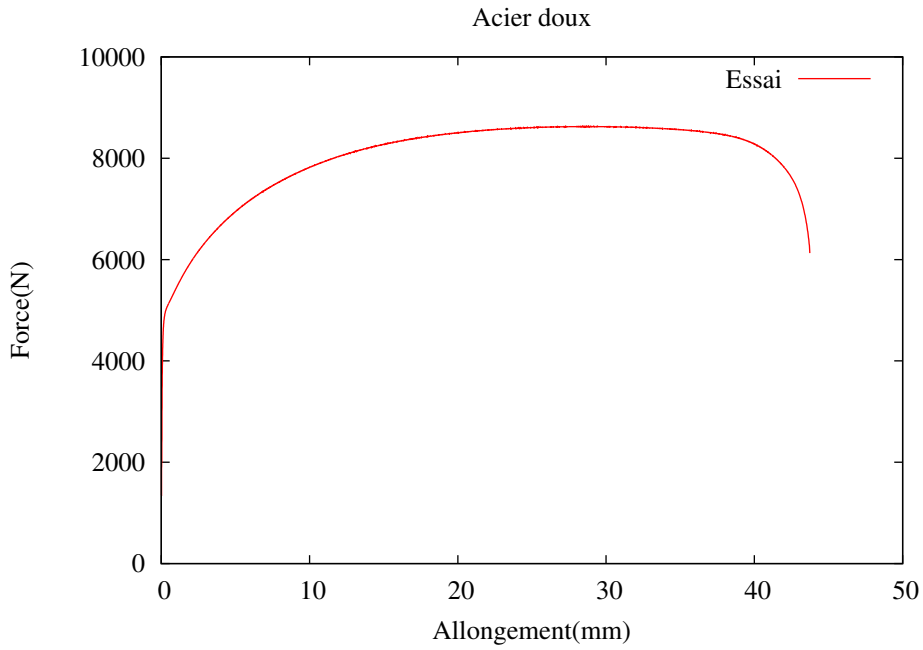

FIG. 2.15 – Courbe de traction uni-axiale

Ces courbes vont nous servir comme observations pour l'identification inverse des paramètres de la loi d'écrouissage choisie. Les propriétés mécaniques standards identifiées à partir des courbes de traction sont données dans le tableau 2.2.

Type	Re(0.2%) Mpa	Rm (MPa)	A% rép	A% rup
Acier doux	162	288	28	45
Acier multiphasé	1040	1235	4.7	8.3

TAB. 2.2 – Propriétés mécaniques

Les valeurs du coefficient de Lankford, fournies par ArcelorMittal, mesurant le rapport des déformations plastiques en largeur et en épaisseur, sont également très différentes : l'acier multiphasé est pratiquement isotrope, avec $R_0 \approx R_{45} \approx R_{90} \approx 1$, tandis que l'acier doux est fortement anisotrope, avec $R_0 \approx 2$,$R_{45} \approx 1.5$, $R_{90} \approx 2.4$

Les éprouvettes après essai montrent, pour l'acier multiphasé, une rupture dont la direction correspond à la prévision théorique de la striction localisée (Fig. 2.16). Cette rupture intervient après un amincissement localisé dont l'extension perpendiculairement à la rupture est de l'ordre de grandeur de l'épaisseur. La rupture pour l'acier doux intervient d'abord au centre de l'éprouvette, puis la déchirure se propage vers les bords (Fig. 2.17).

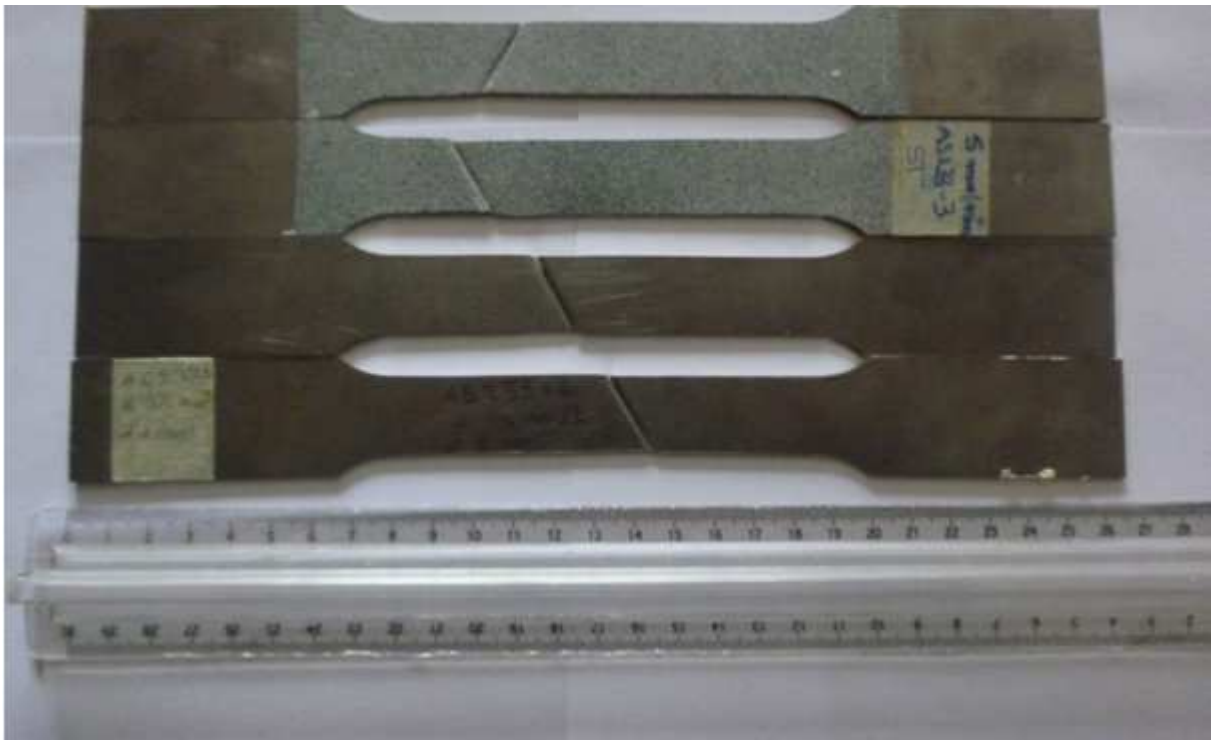

FIG. 2.16 – Éprouvettes Acier multiphasé

FIG. 2.17 – Éprouvettes Acier doux

Nous retiendrons, pour définir l'arrêt des calculs dans les simulations numériques, un critère de rupture défini par $\bar{\varepsilon} = 1,2$ pour l'acier multiphasé et $\bar{\varepsilon} = 2$ pour l'acier doux.

2.3.3 Outils de simulation numérique

a) FLagSHyP

Nous avons choisi d'utiliser FLagSHyP (Finite element Large Strain Hyperelasticity Program) pour illustrer l'utilisation de la méthode de la Différentiation Automatique sur des applications mécaniques. Le choix de ce code est motivé par le fait qu'il est en libre téléchargement ce qui permet à la communauté scientifique de pratiquer cette méthode sur un code éléments finis entier.

FLagSHyP (Finite element Large Strain Hyperelasticity Program) est un code éléments finis écrit en FORTRAN 77 développé par Bonnet et Wood [2008] pour illustrer les théories de la mécanique des milieux continus non linéaire décrites dans leur livre " Nonlinear Continuum Mechanics for Finite Element Analysis". Les codes sources de FLagSHyP écrit en FORTRAN 77 et en FORTRAN 90 (une version traduite par J. E. Akin) peuvent être téléchargés sur le site http://www.flagshyp.com.

Le programme traite les équations constitutives de plusieurs lois de comportement non linéaires. Neuf types de matériaux sont implémentés : matériau de type néo-Hookean compressible, matériau incompressible avec l'hypothèse de contrainte plane dans la direction principale ou presque incompressible, matériau hypérélastique dans la direction principale ou avec l'hypothèse de la déformation plane dans la direction principale, matériau presque incompressible dans la direction principale, matériau incompressible dans la direction principale avec l'hypothèse de contrainte plane et matériau élasto-plastique dans la direction principale. Dans le code, chaque type de matériau est désigné par un numéro noté *mat* est défini par un tableau contenant ses paramètres noté *"props"*. On lit ces données dans un fichier d'entrée *"nom_exemple.dat"* indispensable pour l'exécution du code. Dans ce fichier d'entrée, on définit aussi le maillage de la structure à solliciter, la table de connectivité et le type d'élément utilisé. Le code propose sept types d'éléments de maillage : des éléments de maillage 2D : linéaire (*truss2*), linéaire triangulaire (*tria3*), quadratique triangulaire (*tria6*), bilinéaire quadrilateral (*quad4*) et des éléments 3D : tétraédrique linéaire (*tetr4*), tri-linéaire hexaédrique (*hexa8*), quadratique hexaédrique (*tetr10*) (voir Fig. 2.18).

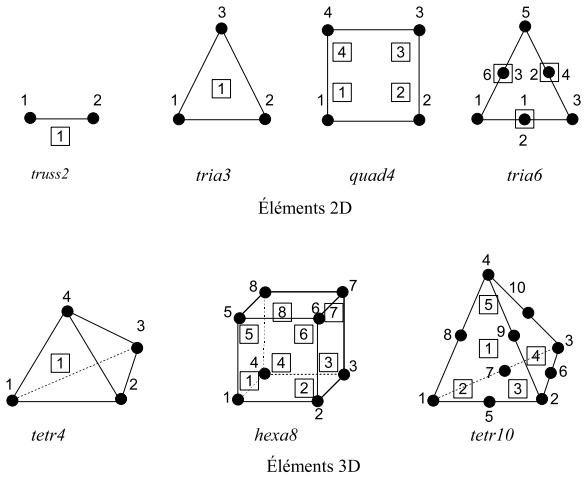

FIG. 2.18 – Éléments de maillage

Dans ce même fichier, l'utilisateur doit préciser le type de chargement : pression surfacique, force ponctuelle ou déplacement, ainsi que les conditions aux limites et les paramètres de contrôle de la solution numérique (incrément, tolérance de convergence, nombre maximum d'itération par incrément...).

Une formulation Lagrangienne est utilisée pour résoudre le problème de la déformation finie par la méthode des éléments finis. Le problème non linéaire discrétisé résultant est alors résolu par une méthode itérative de Newton-Raphson. Pour améliorer la convergence de cette dernière, la méthode de la recherche linéaire et la méthode de la longueur d'arc sont implémentées dans le code. Et pour résoudre le système d'équation linéaire à chaque itération de Newton-Raphson, un solveur standard symétrique basé sur la décomposition LDL^T [Zienkiewicz et Taylor, 2005] est utilisé.

A l'exécution de chaque problème, un fichier de sortie est créé (le nom est donner par l'utilisateur). Dans ce fichier, les résultats sont décrits pour chaque incrément de charge : les nouvelles coordonnées des nœuds, les forces nodales équivalentes et les composantes du tenseur de Cauchy aux points d'intégration de Gauss. Le programme génère également un autre fichier de sortie qui permet de continuer la mise en œuvre d'un problème s'il n'a pas convergé. Un fichier nommé *"flag.out"* est crée. Il contient le numéro d'incrément, les coordonnées et la force relative au degré de liberté, la valeur du facteur de force et la longueur d'arc actuelle si la méthode de longueur d'arc est utilisée. Ce fichier facilite ainsi le traçage des courbes force-déplacement.

Parmi toutes les sous-routines qui composent le programme, la plus importante est *"elemtk"*, une sous-routine à maîtriser. Cette dernière est chargée de calculer les forces nodales équivalentes et les composantes de la matrice de rigidité tangente. Plus de détails

sur cette sous-routine ainsi que les autres composantes du programme,figurent dans le livre [Bonnet et Wood, 2008].

Dans la littérature plusieurs auteurs ont utilisé FLagSHyP pour résoudre des problèmes de grandes déformations dans de nombreux domaines. Rocha [2008] a utilisé FLagSHyP pour réaliser un modèle de couplage électromécanique du tissu cardiaque par la méthode des éléments finis. Givoli et Patlashenko [2000] ont utilisé FLagSHyP comme base pour développer leur propre code qui traite le problème de contrôle optimal d'élasticité. Une autre application en biomécanique pour la simulation numérique du formage hyperplastique des prothèses dentaires est traitée par FLagSHyP par [Bonnet et Wood, 2002].

Les aciers doux et multiphasé sont modélisés en implémentant une loi elastoplastique à 6 paramètres dans FLagSHyP. L'élément de maillage utilisé pour les simulations est un élément 3D à 8 noeuds hexa8(voir Fig. 2.18) . L'éprouvette est maillée avec 63 éléments.

b) Code de calcul Abaqus

Une difficulté pour mener à bien notre objectif résulte de l'implémentation moyennement efficace du code FLagSHyP qui ne permet pas d'effectuer des simulations avec un nombre important d'éléments. Or, pour reproduire correctement la localisation de la déformation dans l'essai de traction, il est nécessaire de mailler l'éprouvette de façon suffisamment fine. Pour mener à bien notre étude, nous avons donc été amenés à utiliser en complément le code de calcul Abaqus qui permet des calculs avec un nombre élevé d'éléments.

Nous avons utilisé la version Abaqus/explicit, dans les mêmes conditions (dimensions d'éprouvettes, conditions aux limites). Pour ces simulations on a modélisé le quart de l'éprouvette (une symétrie en largeur et une symétrie en épaisseur). On a effectué des simulations :

- d'une part, un petit nombre d'éléments de maillage : 4x18= 72 éléments, afin de recouper les résultats avec ceux obtenus avec FLagSHyP,
- d'autre part, un nombre élevé d'éléments : 4x 18368=73472 éléments, afin d'obtenir une description fine de la localisation de la déformation et donc des courbes de traction.

L'élément de maillage utilisé est l'élément solide avec interpolation linéaire et intégration réduite C3D8R

2.3.4 Mise en œuvre de l'identification

a) Choix de la loi de comportement

Nous avons utilisé le critère de Von-Mises [1913] (équation (2.10)) avec l'hypothèse d'écrouissage isotrope. Afin d'obtenir une souplesse suffisante pour décrire l'écrouissage, nous avons choisi une loi d'écrouissage à 6 paramètres, combinant la loi de Voce et la loi de Swift, sous la forme suivante :

$$\bar{\sigma}(\bar{\varepsilon}^p) = K(\varepsilon_0 + \bar{\varepsilon}^p)^N + S(1 - A\mathrm{Exp}(-B\bar{\varepsilon}^p)), \tag{2.22}$$

où K, ε_0, N, S, A et B sont les paramètres à identifier. Cette loi est utilisée dans le contexte industriel pour améliorer les extrapolations des lois d'écrouissage en grandes déformations.

b) Différentiation de FLagSHyP

La loi à 6 paramètres (3.28) est implémentée dans FLagSHyP . Pour effectuer l'identification de ces paramètres, une fonction coût est implémentée dans le code sous la forme :

$$\mathrm{cout} = \mathrm{cout} + (observations - simulations)^2. \tag{2.23}$$

Par la suite le code est différentié automatiquement par TAPENADE [Hascoët *et al.*, 2005a] afin de calculer le gradient de la fonction coût par rapport à ces paramètres.

Afin de différentier FLagSHyP on doit fournir à TAPENADE :

- la routine dans laquelle se trouve la fonction coût : la "*top routine*", dans notre cas c'est "*simulation*",
- les variables d'entrées ou "*variables indépendantes*", par rapport aux quelles la fonction coût sera différentier, dans notre cas c'est le vecteur "*props*" contenant les paramètres de la loi de comportement choisie,
- le mode de différentiation, dans notre cas c'est le mode linéaire tangent.

Après avoir suivi les différentes étapes décrites ci-dessus, on lance Tapenade qui analyse les différentes routines constituant le code(avec un parser fortran) pour disposer de l'architecture du code ainsi que des dépendances entre les routines et les variables. Après quoi, Tapenade garde les routines dont il a besoin et commence la différentiation du code. Tapenade ajoute la suffixe "d" aux variables différentiées en mode tangent. Les routines différentiées (voir (Fig. 2.19)) sont produites en 5 secondes sur un ordinateur à processeur Intel Core2 Duo de 2GHz de fréquence et 2Go de mémoire. Il est impératif de vérifier que le gradient généré par TAPENADE est correcte.

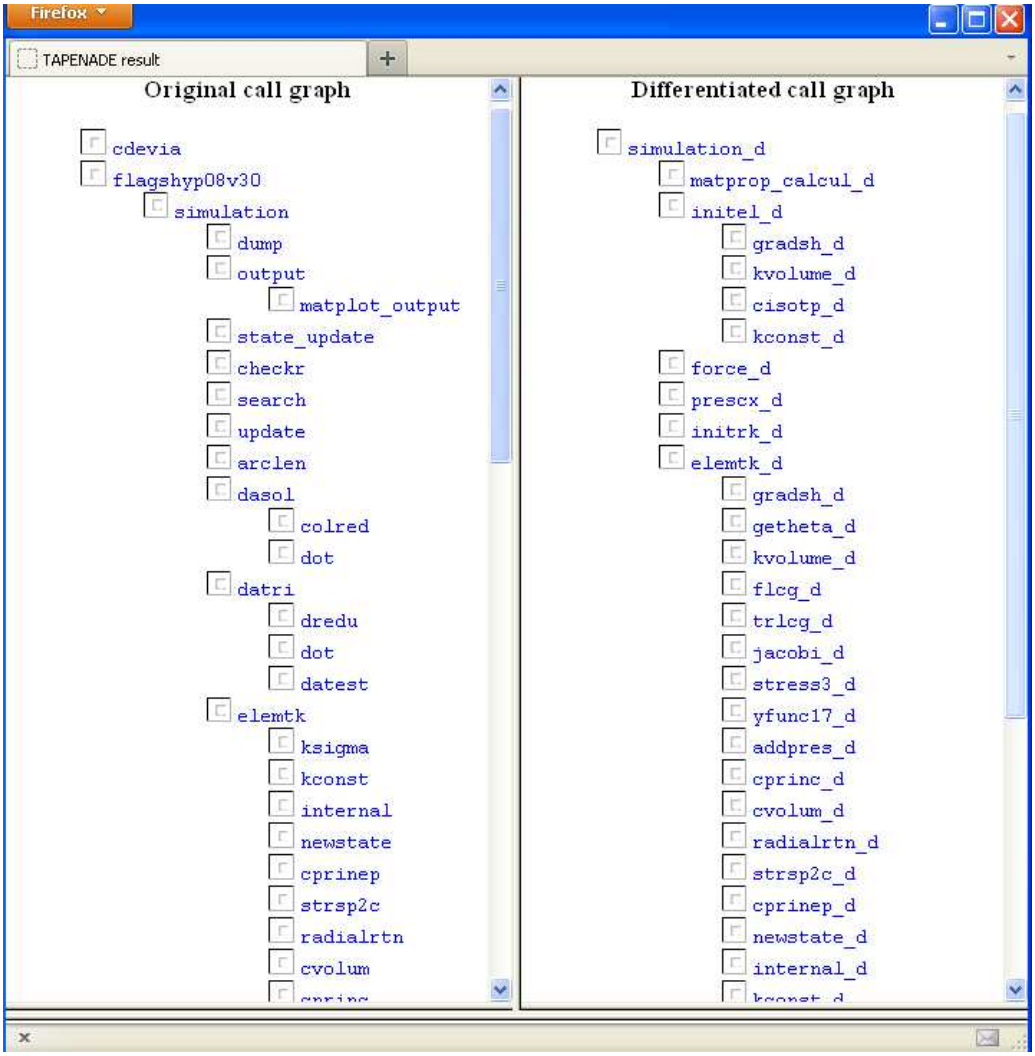

FIG. 2.19 – Code différentié par TAPENADE

2.3.5 Validation de la différentiation

On fait le test de Taylor afin de valider le calcul du gradient(voir 1.2.3) par rapport K et N. Les tests présentés dans le tableau 2.3 ont le comportement des tests de Taylor corrects (voir 1.2.3).

α	r_N	r_K
10^{2}		0.951643130059070
10^{1}		0.995156431113191
10^{0}		0.999515516916777
10^{-1}		0.999951258456755
10^{-2}	0.988310105047826	0.999993414521180
10^{-3}	0.998823876138017	0.999980133510347
10^{-4}	0.999883251485084	0.999555623720398
10^{-5}	0.999996756404297	0.993150259551969
10^{-6}	1.00015765423819	0.957328985065253
10^{-7}	1.00074926420769	0.199343952127866
10^{-8}	1.00626440059662	
10^{-9}	1.09970679443109	
10^{-10}	1.48612069388632	
10^{-11}	24.4700976907495	

TAB. 2.3 – Test de Taylor effectué dans les directions K et N.

Après avoir vérifié l'exactitude du code différentié, le module d'optimisation [Byrd *et al.*, 1995]((cf. § 1.3.6)) est interfacé avec le code pour réaliser l'identification inverse. Pour vérifier l'efficacité de notre optimiseur, on a utilisé la méthode des expériences jumelles.

2.3.6 Etude de sensibilité des paramètres

Une étude de sensibilité menée avec le code linéaire tangent permet de voir l'influence de chaque paramètre sur la loi de comportement (2.23) sans effectuer l'étude paramétrique classique (répétition des simulations directes pour des variations de chaque paramètre). Avec le code linéaire tangent, il suffit d'exécuter ce code pour un jeu de paramètres et différentes directions. Par exemple, *propsd* est choisi égal à (0.1,0,0,0,0,0) pour une perturbation de 10% de la valeur de K.

La figure 2.20, présente les sensibilités " fd " de la force en fonction du déplacement pour chaque direction de perturbation.

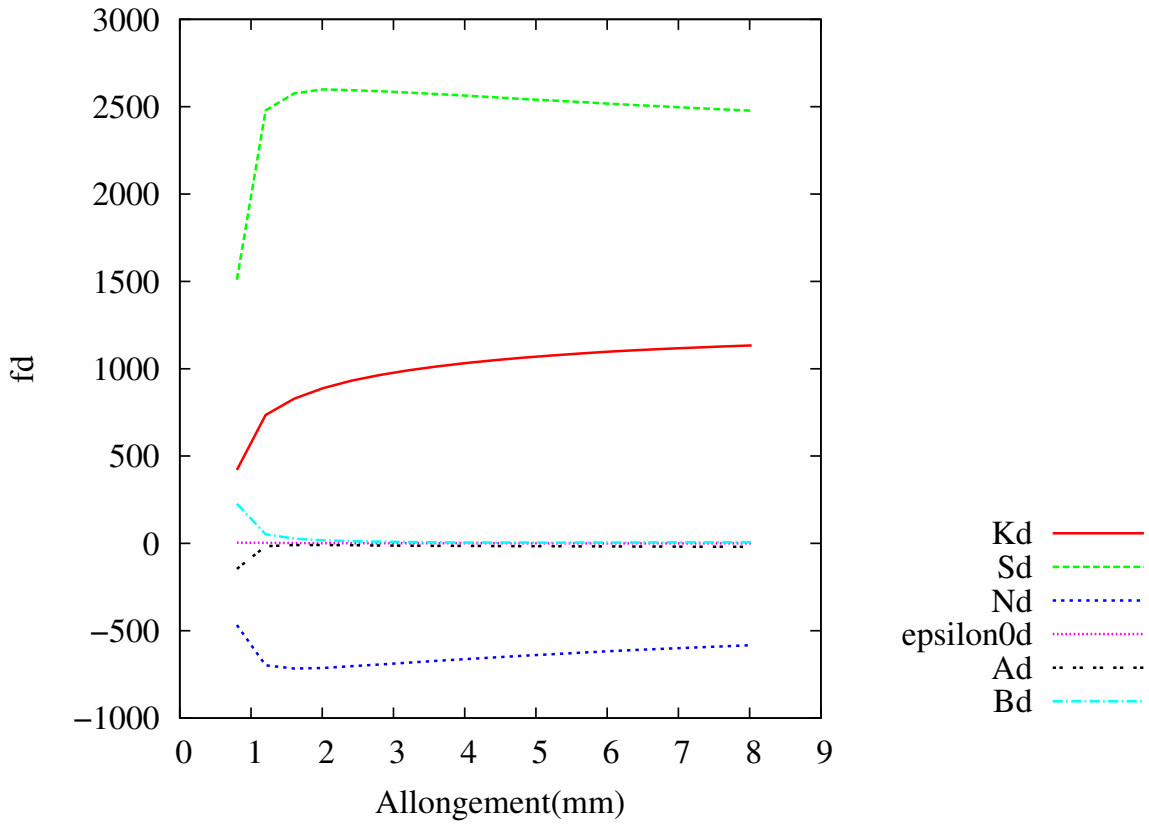

FIG. 2.20 – Étude de sensibilité par rapport à une perturbation de 10% des paramètres

On remarque que les paramètres K et S, qui définissent l'intensité de la contrainte, jouent le rôle le plus important. Les autres paramètres affectent plutôt la forme de la courbe contrainte-déformation, et jouent un rôle moindre sur l'intensité de la courbe force-déplacement. Ces résultats, attendus d'une étude analytique, sont ici quantifiés d'une manière très précise.

2.3.7 Expériences Jumelles pour tester l'optimiseur

Il s'agit d'expériences idéalisées dont les données sont parfaitement consistantes avec le modèle, ce qui est rarement le cas en pratique. Dans cette expérience, on utilise des données(observations synthétiques). On observe la présence d'une zone de striction diffuse créées par le modèle avec un jeu de paramètres donnés. Leur valeur étant parfaitement connue, il est possible de valider la méthode. Ensuite on se donne un itéré initial pour l'algorithme de minimisation de la fonction coût (2.23).

Le tableau ci-dessous résume les différents paramètres utilisés pour effectuer l'expérience jumelle. On y observe que le jeu de paramètres identifié est très proche du jeu de

paramètres utilisés pour générer les observations. La procédure d'identification que nous avons construite est correcte. Ces résultats sont confirmés dans la figure 2.21 présentant les observations synthétiques et les courbes force-allongement relatives aux paramètres de la loi initiale et de la loi identifiée : les résultats obtenus avec la loi initiale passent par les observations utilisées dans la fonction coût.

	Loi pour obs. synthétiques	Expérience Jumelle	
		Loi initiale	Loi identifiée
Paramètres d'écouissage	K=700 E=0.0001 N=0.2 S=900 A=0.5 B=1500	K=100 E=0.001 N=0.5 S=1000 A=0.57 B=1000	K=698.12 E=0.0001 N=0.196 S=897.25 A=0.498 B=1497.98

TAB. 2.4 – Jeu de paramètres de l'expérience jumelle.

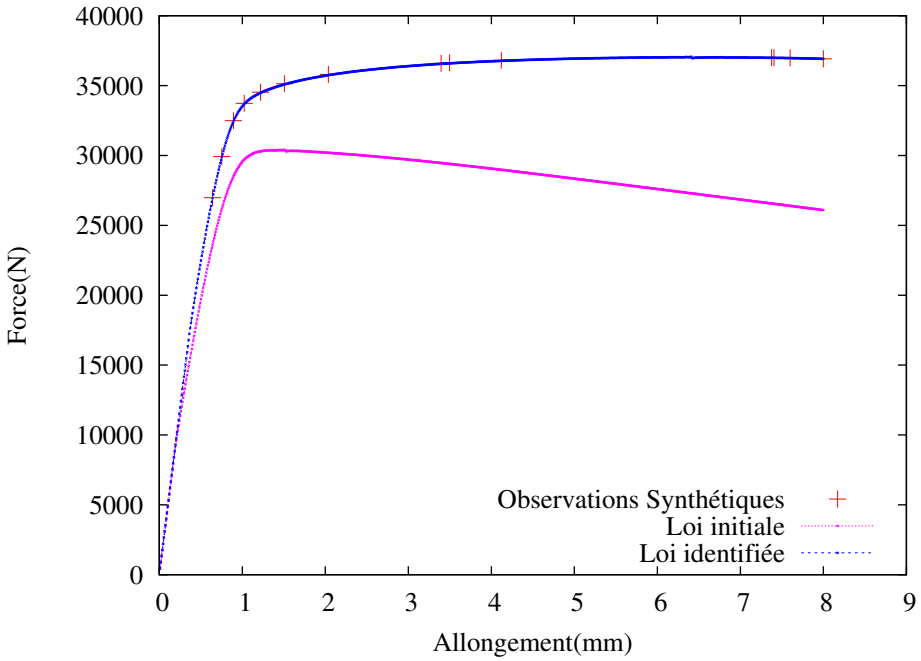

FIG. 2.21 – Expérience Jumelle

2.4 Résultats

L'incapacité de mener à bien les simulations impliquant un grand nombre d'éléments avec le code de calcul FLagSHyP, nous a amené à adopter une stratégie d'identification

qui permet seulement de montrer l'applicabilité de la méthode DA au problème d'identification que nous avons défini. En effet, un maillage très fin est nécessaire pour reproduire correctement le développement de la localisation de la déformation. Un maillage grossier donne des courbes force-déplacement qui présentent une chute de la force trop lente, car la localisation associée à la striction n'est pas correctement décrite. L'identification ne peut donc être effectuée avec le code FLagSHyP que jusqu'à une valeur de l'allongement légèrement supérieure à l'allongement à force maximum. Nous avons donc été amenés à définir la méthode suivante :

- identification des paramètres de la loi d'écrouissage avec FLagSHyP (maillage de l'éprouvette avec 63 éléments), dans un domaine d'allongement choisi pour que la courbe force-déplacement reste pratiquement insensible au nombre d'éléments finis ;
- comparaison des résultats obtenus avec les codes FLagSHyP et Abaqus, en utilisant les paramètres précédemment identifiés et un maillage comportant 63 éléments (FLagSHyP) ou 4x18 éléments (Abaqus) ; vérification de la coincidence des courbes force-déplacement obtenues avec les 2 codes ;
- simulation de l'essai avec Abaqus en utilisant un grand nombre d'éléments ; comparaison avec la courbe force-déplacement expérimentale.

Le tableau 2.5 précise pour chaque matériau le domaine d'allongement choisi pour l'identification avec FLagSHyP (63 éléments).

Matériau	Allongement réparti (valeur à F_{max}) mm	Nombre d'éléments	Domaine d'allongement mm
Acier Doux	28	63	3-38
Acier multiphasé	4.7	63	0.56-7.1

TAB. 2.5 – Domaines d'allongement choisis pour l'identification.

Les paramètres d'écrouissage obtenus selon la procédure expliquée ci-dessus (identification avec FLagSHyP, 63 éléments) sont donnés dans le tableau 2.6.

Matériau	K(Mpa)	ε_0	N	S(Mpa)	A	B
Acier multiphasé	526	1.E-7	0.049	529	0.159	54
Acier doux	200	0.005	0.27	190	0.56	4.7

TAB. 2.6 – Paramètres identifiés.

Les courbes expérimentales et calculées sont montrées sur les figures 21 et 22. On observe tout d'abord que la simulation avec FLagSHyP (63 éléments) ne permet pas de

décrire correctement le développement de la striction, et conduit de ce fait à une courbe force-déplacement qui chute beaucoup trop lentement. Par contre les résultats coïncident pratiquement avec la simulation effectuée avec Abaqus avec 4×18 éléments, ce qui valide les simulations effectuées avec les deux codes. Lorsque les simulations sont effectuées avec Abaqus et un grand nombre d'éléments, on retrouve une allure force-déplacement en bien meilleur accord avec l'expérience.

Afin de prendre en compte l'anisotropie de l'acier doux, les simulations ont de plus été effectuées avec Abaqus en utilisant l'option " potential ", qui permet d'effectuer les simulations en utilisant le critère de Hill quadratique. Les différents paramètres d'anisotropie (rapports de contraintes) calculés à partir des coefficients d'anisotropie : $R_0 = 2$, $R_{45} = 1.5$, $R_{90} = 2.4$ sont : $R_{11} = 1$, $R_{22} = 1.029$, $R_{33} = 1.279$, $R_{12} = 1.108$, R_{13} et R_{23}, qui font appel à la connaisance des contraintes de cisaillement transverse, sont pris égaux à 1.

2.5 Discussions

2.5.1 Courbes force-déplacement

Les courbes force-déplacement calculées avec Abaqus et un grand nombre d'éléments devraient reproduire les courbes expérimentales, dès lors que l'ensemble des phénomènes physiques intervenant en grandes déformations plastiques seraient pris en compte. Or, les courbes simulées montrent :

- pour l'acier multiphasé, une chute de force un peu trop lente (Fig. 2.22),
- pour l'acier doux, une chute de force beaucoup trop précoce. La prise en compte de l'anisotropie modifie peu la courbe force-déplacement simulée (Fig. 2.23).

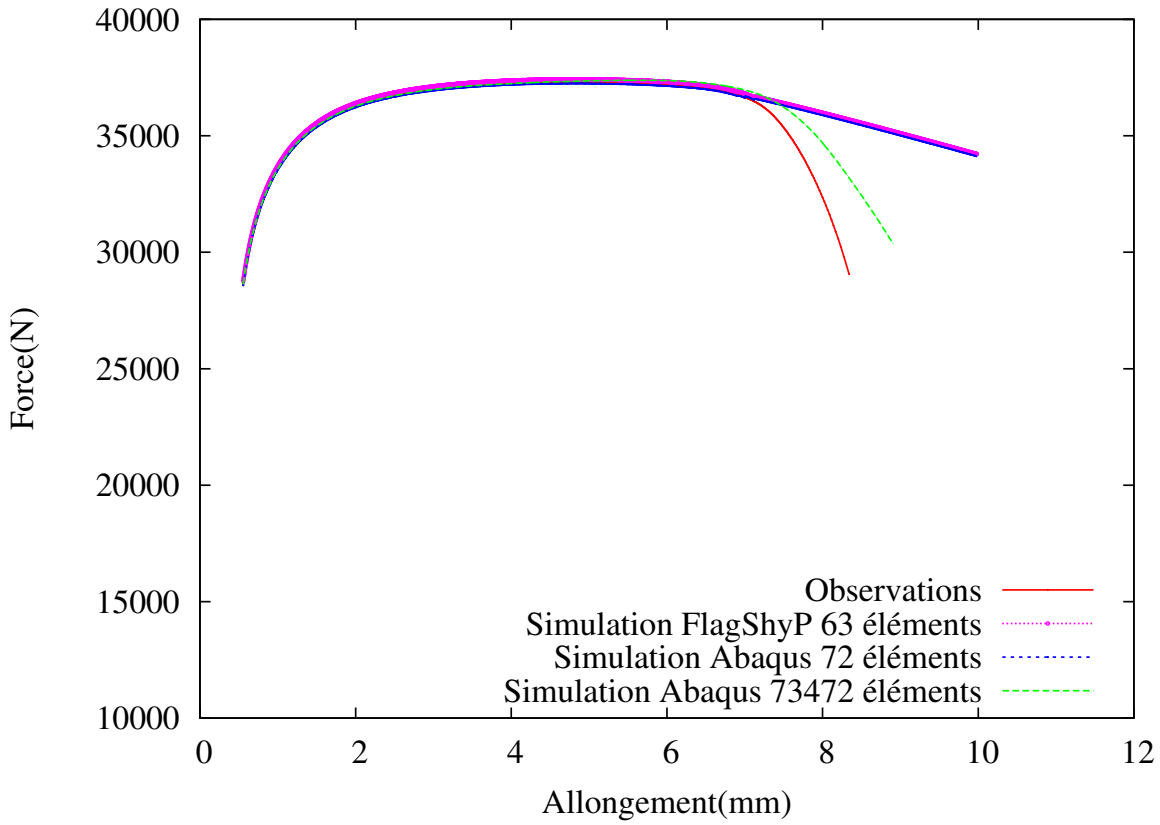

FIG. 2.22 – Courbes Force-Déplacement Acier multiphasé

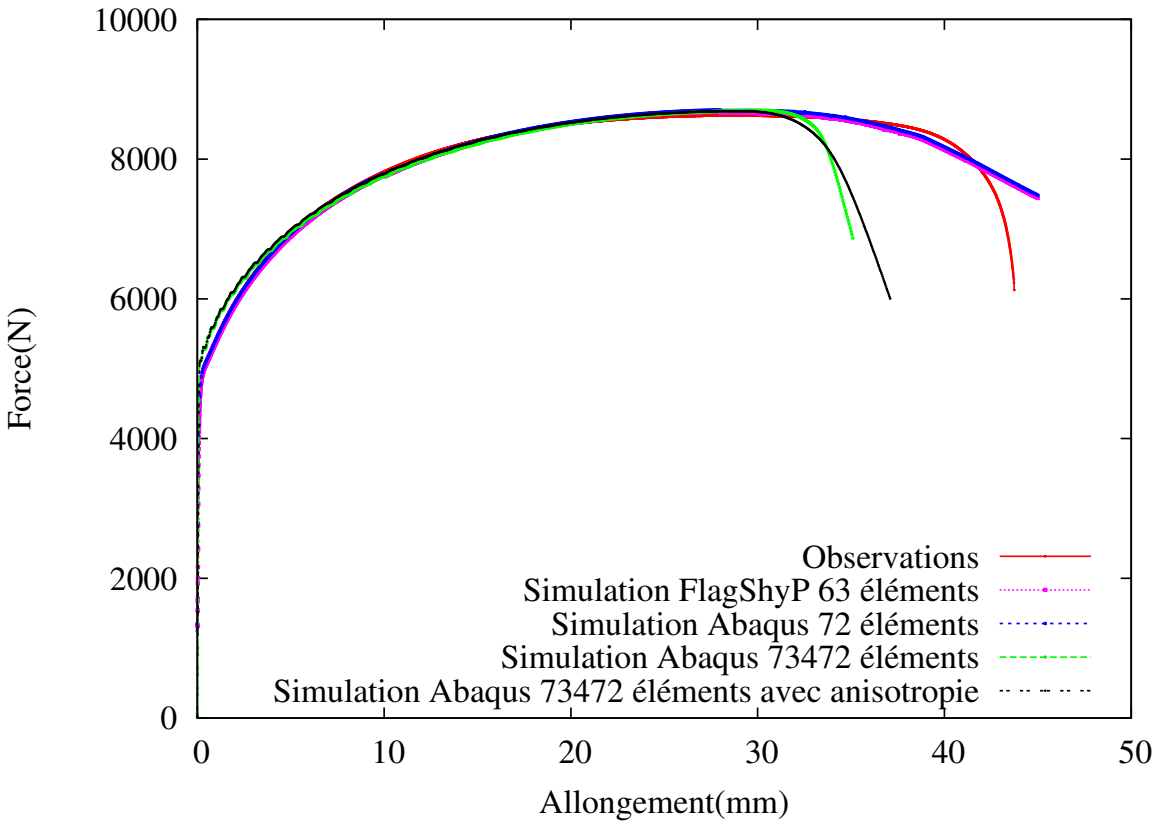

FIG. 2.23 – Courbes Force-Déplacement Acier doux

En fait, la loi d'écrouissage à 6 paramètres a été identifiée sur un domaine d'allongement de l'éprouvette (et donc de déformation plastique) limité par les possibilités d'effectuer avec FLagSHyP, des simulations avec un grand nombre d'éléments. Les extrapolations de la loi d'écrouissage peuvent diverger du comportement réel du matériau si le domaine d'identification est trop restreint. La loi d'écrouissage choisie offre en effet la possibilité :

- d'une poursuite de l'écrouissage si le terme de type " loi de Swift " prédomine ($K >> S$),
- d'une saturation de l'écrouissage si le terme de type " loi de Voce " prédomine ($S >> K$).

D'autre part, deux phénomènes physiques principaux n'ont pas été pris en compte, et peuvent jouer un rôle important dans le développement de la striction et par conséquent dans la détermination de la partie finale de la courbe de traction :

- la sensibilité à la vitesse de déformation, qui a pour effet de ralentir le développement de la striction, puisque l'augmentation de la vitesse de déformation dans la zone de striction conduit à une augmentation des contraintes [Woodford, 1969]

– l'endommagement en grandes déformations, qui a au contraire pour effet d'accélérer la localisation de la déformation puis la rupture ductile dans la zone de striction [Rice et Tracey, 1969, Lemaitre, 1986].

En particulier, l'acier doux d'emboutissage présente une sensibilité à la vitesse de déformation de l'ordre de $m = \frac{\partial \mathrm{Ln}\sigma}{\partial \mathrm{Ln}\dot{\varepsilon}} = 0.03$ ($m = 0.038$ d'après les mesures d'ArcelorMittal). Des études analytiques approchées concernant l'allongement au-delà du point de force maximum ont été menées pour montrer l'importance de cette sensibilité à la vitesse de déformation [Semiatin et Jonas, 1984] . On peut retenir une expression de la déformation ε_f (ou de l'allongement relatif ε_f) en dehors de la zone de striction quand la déformation de la zone de striction tend vers l'infini :

$$\varepsilon_f = N + mLn(I/f), \tag{2.24}$$

ou

$$e_f = (I/f)^m \exp(N) - 1, \tag{2.25}$$

où N est l'exposant d'écrouissage pour une loi de Hollomon ($\sigma = K\varepsilon^N$) et $f = \delta Ln(A_0)$ mesure la taille d'un défaut géométrique de section droite initiale le long de l'éprouvette de traction. Un accord avec l'expérience est souvent obtenu en considérant un défaut de l'ordre de $f = 0.01$.

L'acier doux pourrait être caractérisé de façon approchée par une loi d'écrouissage de Hollomon avec un exposant d'écrouissage $N = 0.26$. Ainsi, on s'attend pour ce matériau à des variations de l'allongement d'une éprouvette de traction en fonction de m qui sont définies dans le tableau 2.7 :

m	Allongement(%)
0	30
0.01	36
0.02	42

TAB. 2.7 – Allongement d'une éprouvette de traction estimé à partir de l'équation (2.5.1) pour $N = 0.26$ et différentes valeurs de m.

Considérant l'écart observé pour la chute de la force entre l'essai et la simulation avec Abaqus 73428 éléments (Fig. 2.23), on s'aperçoit de toute l'importance qu'il y aurait à prendre en compte l'influence de la sensibilité à la vitesse de déformation dans les simulations numériques. L'écart constaté est en effet parfaitement compatible avec une

sensibilité à la vitesse de déformation de l'ordre de quelques %.

Dans le cas de l'acier multiphasé, la sensibilité à la vitesse de déformation est très faible. Le désaccord entre courbes de traction expérimentale et simulée est, au contraire, compatible avec un effet d'endommagement dont le développement génère la rupture ductile.

2.5.2 Champs de déformation

Les figures 2.24 et 2.25 montrent les champs de déformation équivalente et de déformation en épaisseur à la surface des éprouvettes. Les champs sont présentés à la fin des simulations. Le test d'arrêt des simulations est défini par l'atteinte d'une déformation équivalente $\bar{\varepsilon} = 1.2$ pour l'acier multiphasé, $\bar{\varepsilon} = 2$ pour l'acier doux (simulation isotrope) et $\bar{\varepsilon} = 1.8$ pour l'acier doux (simulation anisotrope).

On observe la présence d'une zone de striction diffuse dans laquelle se sont développées deux bandes de striction localisée symétriques par rapport à l'axe de l'éprouvette. En pratique, pour l'acier multiphasé, seule une bande s'est développée pour conduire à la rupture (Fig. 2.16), en raison de gradients de propriétés métallurgiques ou de défauts géométriques (défaut d'alignement par exemple). Pour les mêmes raisons, la rupture n'a pas été observée strictement au centre de l'éprouvette. Pour l'acier doux, la rupture s'est amorcée sur l'axe de l'éprouvette (Fig. 2.17), ce qui correspond au maximum de déformation équivalente. On observe en outre, pour l'acier doux, des profils d'éprouvettes présentant logiquement une diminution de largeur plus importante dans la simulation anisotrope, en raison d'un coefficient d'anisotropie supérieur à 1. Ces profils obtenus dans la simulation anisotrope sont en meilleur accord avec l'expérience. On note en outre des valeurs relatives de déformation équivalente et de déformation en épaisseur dans la zone de striction qui correspondent à un état de contrainte biaxial.

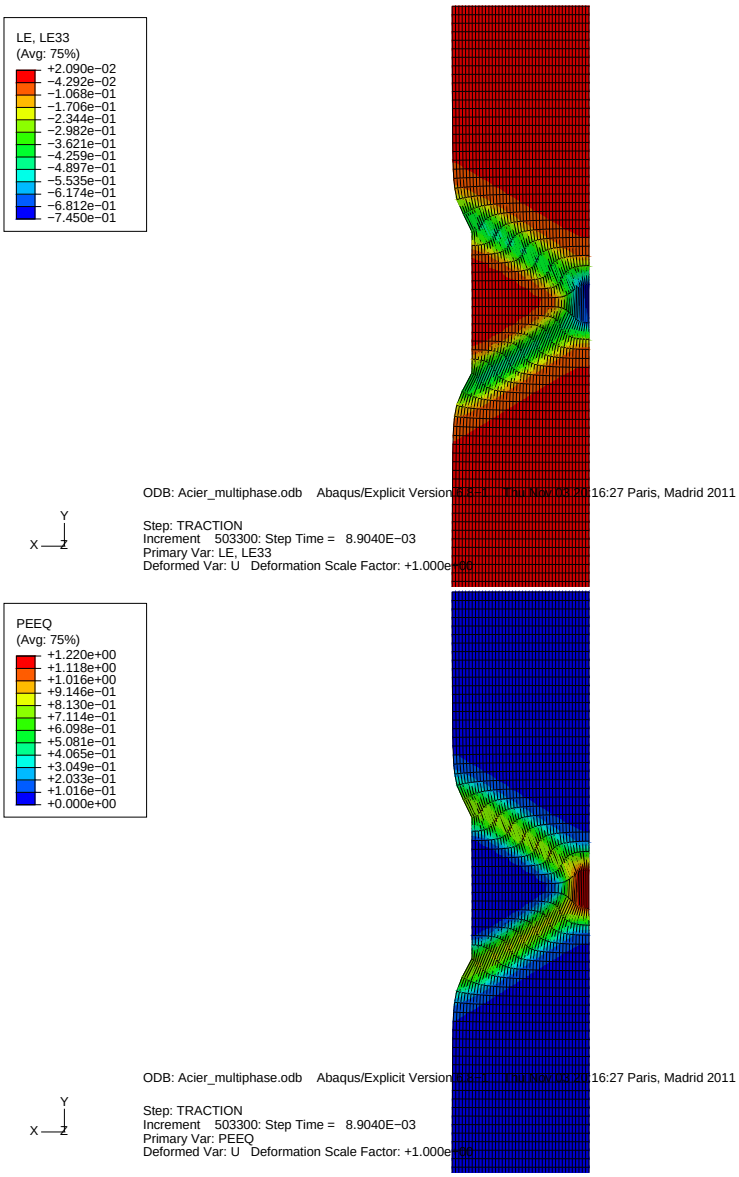

FIG. 2.24 – Acier multiphasé. Champs de déformation équivalente et champs de déformation en épaisseur à la fin de la simulation (représentation sur une demi largeur de l'éprouvette.

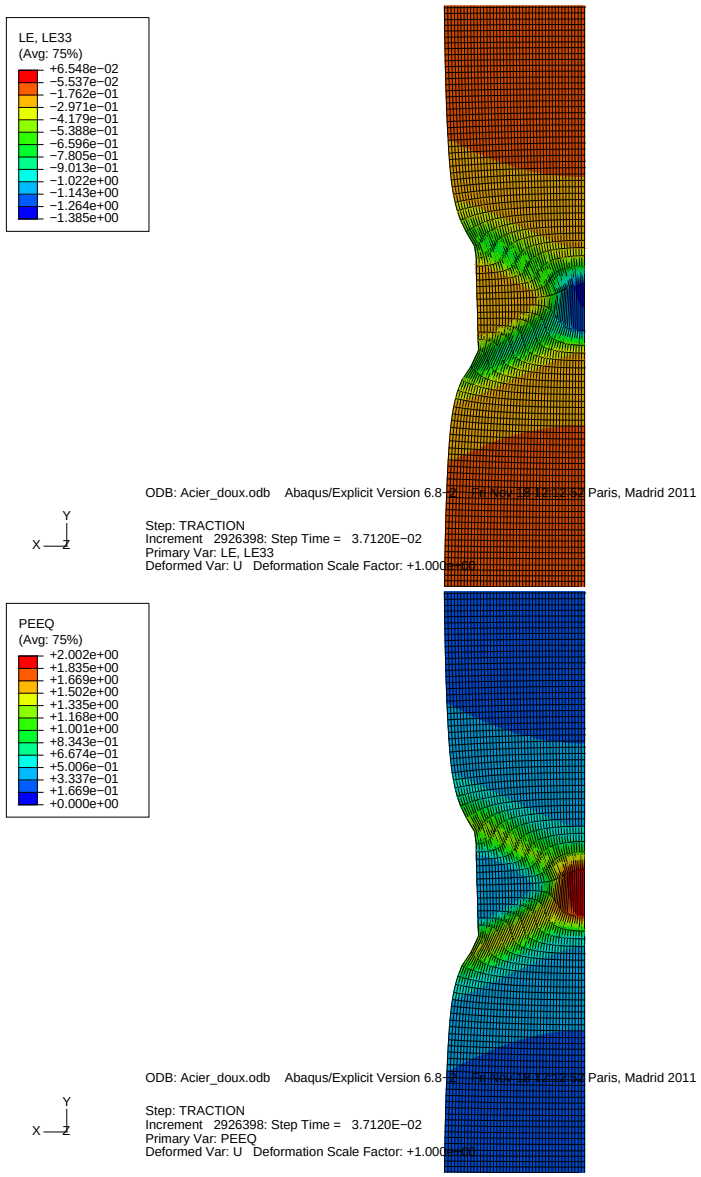

FIG. 2.25 – Acier doux. Champs de déformation équivalente et champs de déformation en épaisseur à la fin de la simulation (modélisation isotrope, représentation sur une demi largeur de l'éprouvette).

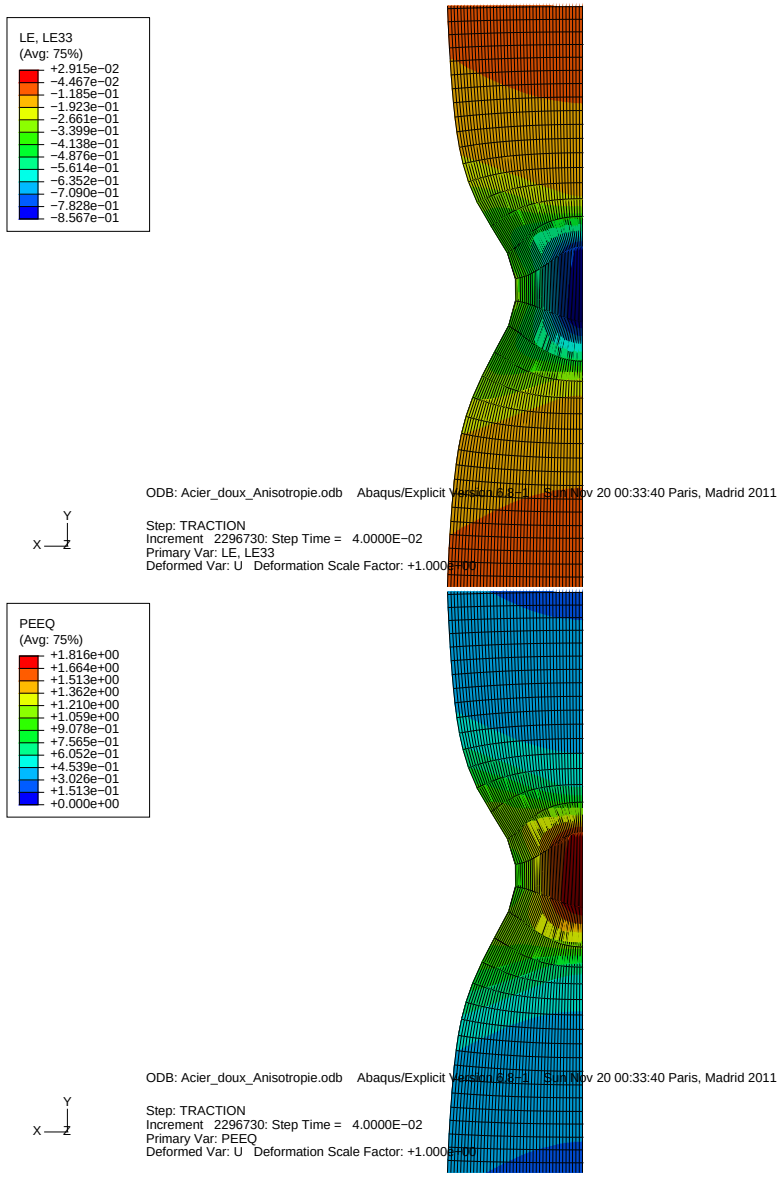

FIG. 2.26 – Acier doux. Champs de déformation équivalente et champs de déformation en épaisseur à la fin de la simulation (simulation anisotrope, représentation sur une demi largeur de l'éprouvette).

2.6 Conclusions

Le problème que nous avons choisi de traiter dans ce chapitre concerne le comportement mécanique de tôles métalliques dans un simple essai de traction uniaxiale. Le développement de la striction diffuse, puis de la striction localisée, sont les deux phénomènes qui contrôlent l'évolution de la courbe force-déplacement dans les stades avancés de la déformation de l'éprouvette. L'objectif était donc d'analyser les résultats force-

déplacement jusque dans les stades avancés de chute rapide de la force, et d'en extraire la loi d'écrouissage en grandes déformations.

La procédure choisie consiste à simuler l'essai de traction en utilisant une loi d'écrouissage à 6 paramètres, utilisée industriellement pour offrir une flexibilité " suffisante " pour des extrapolations en grandes déformations. Un code par éléments finis " ouvert ", permettant de traiter des problèmes de comportement mécanique non-linéaires, a été utilisé. Le code est différentié automatiquement par rapport aux paramètres de la loi d'écrouissage avec l'outil de différentiation automatique TAPENADE afin de calculer le gradient de la fonction coût. L'identification est réalisée sur la base des couples de valeurs (force, déplacement).

L'applicabilité de la méthode a été démontrée, mais un certain nombre de limitations sont apparues. La première limitation est liée au code par éléments finis utilisé, qui ne permet pas de mener à bien des simulations avec un grand nombre d'éléments. De ce fait, nous avons dû faire appel à un recoupement des résultats avec ceux obtenus avec le code de calcul Abaqus. Finalement, les simulations reproduisent la chute de force, associée à des champs de déformation mettant en évidence la striction diffuse puis la striction localisée. Toutefois, des écarts sont constatés par rapport à l'expérience.

En plus de l'écrouissage, le développement de la localisation de la déformation dépend de paramètres physiques tels que la sensibilité à la vitesse de déformation et l'endommagement. Ces effets devront être pris en compte dans une poursuite de l'étude. Enfin, il serait utile d'enrichir la base des données utilisées pour l'identification en déterminant expérimentalement les champs cinématiques de déplacement et de déformation grace à des techniques d'analyse d'images numériques.

CHAPITRE 3

Identification de la loi de comportement d'une poutre sandwich à partir de fréquences de résonance et de facteurs d'amortissement expérimentaux

Sommaire

3.1 Introduction 73
3.2 Matériaux viscoélastiques 75
3.2.1 Loi de comportement d'un matériau viscoélastique 76
3.2.2 Modélisation rhéologique de la loi viscoélastique 86
3.3 Méthodes de résolution du problème des vibrations linéaires 88
3.3.1 Méthodes approchées 89
3.3.2 Méthodes directes 92
3.3.3 Méthode Asymptotique Numérique 93
3.4 Problématique et constats 96
3.4.1 Essais expérimentaux 96
3.4.2 Simulation Numérique 98
3.4.3 Choix d'une loi viscoélastique pour l'identification 99
3.5 Identification de la loi viscoélastique 100
3.5.1 Modélisation inverse 100
3.5.2 Calcul du gradient 101
3.5.3 Différentiation Automatique 102
3.5.4 Pilote d'optimisation 105

3.6 Résultats de l'identification 106
3.7 Conclusion 108

3.1 Introduction

L'étude du contrôle des vibrations est devenue un élément essentiel dans la conception mécanique de système complexe dans des domaines tels que l'aéronautique, l'automobile, l'électroménager et le génie civil. Ces vibrations peuvent susciter des instabilités de la structure, entraînant des gênes importantes pour l'utilisateur (vibrations ressenties par un automobiliste à l'intérieur de l'habitacle, rayonnement acoustique élevé...), voire même le phénomène de rupture dû à la fatigue vibratoire. En pratique, un grand nombre de ces désagréments sont supprimés en ajoutant de l'amortissement dans la structure, augmentant ainsi son niveau de dissipation.

Dans les tôles sandwich, le contrôle vibroacoustique est généralement assuré par une ou plusieurs couches viscoélastiques dans le matériau. Plusieurs représentations paramétriques de son module complexe sont proposées pour décrire le comportement viscoélastique d'une telle structure : modèles rhéologiques simples (à deux ou trois paramètres) tels que le modèle de Maxwell, le modèle de Kelvin-Voigt et le modèle de Zener [Sun et Lu, 1995], ou d'ordre élevé tels que le modèle de Maxwell généralisé [Fried, 2003], le modèle de Golla-Hugures-Mac Tavish (GHM) [Golla et Hughes, 1985] et le modèle ADF (Anelastic Displacement Field) [Lesieutre et Bianchini, 1995]. En général, les courbes maitresse du matériau viscoélastique fournies sous forme d'abaque par le fabricant, sont utilisées pour identifier les paramètres de ces différents modèles. Ces courbes maitresse sont issues d'essais de flexion, de traction, de compression ou de cisaillement pour des fréquences et des températures données. Pour des températures et/ou des fréquences absentes de l'abaque, on utilise la loi WLF [Williams *et al.*, 1955]. La plupart des techniques d'estimation de l'amortissement partent d'un des modèles cités précédemment, et tentent d'en identifier les paramètres via des expériences spécifiques ou des simulations numériques. Néanmoins, aucun de ces modèles n'est capable de décrire le comportement réel des matériaux amortissants pour une large bande de fréquences [Bert, 1973], plus particulièrement quand ils sont contraints dans un sandwich. Leur modélisation pose des difficultés de part la dépendance du module de Young en fréquence et en température [Daya et Potier-Ferry, 2001].

Trois grandes classes de méthodes ont été développées pour déterminer les fréquences propres et l'amortissement des structures viscoélastiques en vibration linéaire.

- Dans la première classe, les méthodes sont basées sur le calcul de courbes de réponse pour des excitations harmoniques. En utilisant la technique de la bande passante à $-3dB$ [Soni, 1981], on lit la fréquence propre sur les courbes de réponses, puis on calcule l'amortissement. Elle est utilisable pour un amortissement faible et des

fréquences propres suffisamment distinctes. Mais elle est coûteuse en temps de calcul pour de grandes structures.

- La deuxième classe concerne la méthode d'énergie de déformation modale (Modal Strain Energy Method : MSEM) [Johnson et Kienholz, 1982]. Cette méthode approchée, consiste à définir l'amortissement comme le rapport entre l'énergie de déformation dissipée dans le matériau vicoélastique et l'énergie de déformation emmagasinée. Deux versions existent. La première utilise un mode propre réel (relatif à la structure non-amortie). Elle est utilisable pour des structures faiblement ou moyennement amorties [Daya *et al.*, 2002]. La seconde consiste à résoudre un problème de forçage harmonique basé sur la pulsation propre réelle de la structure non amortie pour obtenir une approximation du mode complexe. Cette version est peu coûteuse en temps de calcul lorsque la matrice de rigidité est indépendante de la fréquence [Soni, 1981].
- La troisième classe consiste à résoudre directement, pour chaque mode de vibration, le problème non-linéaire aux valeurs propres complexes issu de la discrétisation du problème de vibration des structures viscoélastiques. Cinq méthodes de résolution sont principalement discutées dans la littérature : la méthode "QR" [Bathe, 1982, Brezinski et Redivo-Zaglia, 2006], la méthode asymptotique [Ma et He, 1992], la méthode d'itération-réduction [Chen *et al.*, 1999] qui est une variante de la méthode de Newton Raphson, les algorithmes basés sur la Méthode Asymptotique Numérique (MAN) qui couplent la technique de perturbation et l'homotopie [Daya et Potier-Ferry, 2001, Duigou *et al.*, 2003] et les méthodes de sous-espace de Krylov et d'Arnoldi non linéaires [Betcke et Voss, 2002].

Seules les trois dernières méthodes (Itération-réduction, MAN et Krylov-Arnoldi) sont capables de résoudre un problème non-linéaire aux valeurs propres complexes en prenant en compte un comportement viscoélastique dépendant de la fréquence.

Des méthodes inverses combinant des essais expérimentaux et de la simulation numérique ont été proposées pour identifier les paramètres de la loi de comportement viscoélastique. L'idée est de partir d'une loi de comportement analytique décrivant le comportement viscoélastique de la structure, puis de définir "une fonction objectif" à minimiser mesurant l'écart entre simulation et données expérimentales. Le problème de minimisation est alors résolu par un algorithme d'optimisation. Castello *et al.* [2008] et Matter *et al.* [2009] utilisent l'algorithme de Levenberg-Marquardt. Araújo *et al.* [2002] utilisent un algorithme de Gauss-Newton combiné avec l'algorithme FAIPA (Feasible Arc Interior Point Algorithm) [Herskovits, 1998]. Kim et Lee [2009] utilisent aussi une méthode de gradient. Dans tous ces travaux, le gradient est évalué soit par la méthode de différences finies, soit

par un calcul analytique. Néanmoins, un gradient approximatif (une approximation au premier ordre par différences finies) peut fausser les résultats d'optimisation et conduire à une optimisation peu précise. Les objectifs de notre travail sont de prouver l'apport de la Différentiation Automatique (DA) dans le cadre de l'identification des paramètres du cœur viscoélastique (VE) d'une tôle sandwich Acier/Polymère/Acier.

Dans ce chapitre, nous rappelons dans un premier temps des propriétés mécaniques des matériaux viscoélastique ainsi que leur modélisation numérique. Dans un second temps, on introduit à la fois les objectifs de l'identification et la solution numérique du problème non-linéaire aux valeurs propres complexes pour une loi de comportement viscoélastique. Ensuite on décrit l'automatisation du calcul du gradient de la fonction objectif ainsi que les améliorations apportées au code différentié, après quoi on décrit les étapes d'optimisation. Finalement, on expose les résultats de notre identification ainsi que les conclusions et les perspectives.

3.2 Matériaux viscoélastiques

Les matériaux présentant un comportement amortissant sont définis comme matériaux viscoélastiques (VE). Vu le nombre important des domaines d'applications de ce type de matériau, leurs modélisation a fait l'objet de très nombreux travaux. Nous étudions dans ce chapitre, l'identification des paramètres d'une structure sandwich à cœur viscoélastique en vibration libre. Le paragraphe décrit les processus physiques conduisant au comportement dissipatif ainsi que les outils mathématiques utilisés pour les modéliser.

Un matériau est dit viscoélastique lorsqu'il présente un comportement mécanique particulier intermédiaire entre le comportement d'un solide idéal dit "Hookéen" et celui d'un liquide idéal dit "Newtonien". La réponse VE à une déformation est alors un compromis entre celle d'un solide élastique (contrainte proportionnelle à la déformation) et d'un fluide visqueux (contrainte proportionnelle à la vitesse de déformation). La caractéristique principale de ce type de matériau est le retard de la réponse visqueuse par rapport à la réponse élastique, observé aussi bien dans la phase de déformation que dans la phase de relaxation. Ce retard est dû à la dépendance des propriétés mécaniques d'un matériau VE à l'histoire de sollicitation. Pour mettre en évidence le comportement viscoélastique de ce type de matériau, on dispose de quatre essais fondamentaux : essai de relaxation, de fluage, de recouvrance et d'effacement [Salençon, 2009].

3.2.1 Loi de comportement d'un matériau viscoélastique

On établit la loi mathématique liant la réponse du matériau à une sollicitation en utilisant une formulation fonctionnelle. À chaque instant t, la contrainte σ_{ij} (respectivement $\varepsilon_{ij}(t)$) dépend de l'historique de la déformation $\varepsilon_{kl}(\tau)$ (respectivement $\sigma_{kl}(\tau)$) depuis $-\infty$ jusqu'à cet instant t. Il existe alors des fonctionnelles $\mathcal{F}$ et $\mathcal{G}$ liant les contraintes aux déformations comme suit :

$$\begin{cases} \sigma_{ij}(t) = \mathcal{F}\left[\varepsilon_{kl}(\tau^t_{-\infty})\right], \\ \varepsilon_{ij}(t) = \mathcal{G}\left[\sigma_{kl}(\tau^t_{-\infty})\right]. \end{cases} \tag{3.1}$$

a) Viscoélasticité linéaire

Le matériau est dit linéaire si pour deux états de déformations distincts $\varepsilon^{(1)}$ et $\varepsilon^{(2)}$, la correspondance fonctionnelle entre les histoires de contrainte et de déformation est linéaire :

$$\begin{cases} \mathcal{F}\left[\varepsilon^{(1)}_{kl}\left(\tau^t_{-\infty}\right) + \varepsilon^{(2)}_{kl}\left(\tau^t_{-\infty}\right)\right] & = \mathcal{F}\left[\varepsilon^{(1)}_{kl}\left(\tau^t_{-\infty}\right)\right] + \mathcal{F}\left[\varepsilon^{(2)}_{kl}\left(\tau^t_{-\infty}\right)\right], \\ \mathcal{F}\left[\lambda\varepsilon_{kl}\left(\tau^t_{-\infty}\right)\right] & = \lambda\mathcal{F}\left[\varepsilon_{kl}\left(\tau^t_{-\infty}\right)\right] \quad (\text{avec } \lambda \in \mathbb{R}). \end{cases} \tag{3.2}$$

b) Principe de superposition de Boltzman

Ce principe permet d'utiliser la propriété de linéarité du comportement pour exprimer la réponse à n'importe quelle histoire de sollicitation (déformation ou contrainte) à partir des fonctions de fluage $\mathcal{J}$ ou de relaxation $\mathcal{R}$. L'histoire de contrainte $\sigma_{kl}(t)$, par exemple, peut être considérée comme la superposition d'échelons infinitésimaux $\mathrm{d}\sigma_{kl}(\tau)\mathcal{H}(t-\tau)$ (où $\mathcal{H}$ est la fonction échelon unité de Heaviside) et de sauts notés $[\![\sigma_{kl}]\!]_n$ aux instants τ_n. Utilisant la propriété de linéarité, la réponse $\varepsilon(t)$ s'écrit comme la superposition des $\mathrm{d}\varepsilon_{kl}(t)$ produits par ces échelons, soit :

$$\mathrm{d}\varepsilon_{ij}(t) = \mathcal{J}_{ijkl}(t,\tau)\mathrm{d}\sigma_{kl}(\tau). \tag{3.3}$$

En tenant compte des sommes continues et discrètes, on déduit :

$$\varepsilon_{ij}(t) = \int_{-\infty}^{t} \mathcal{J}_{ijkl}(t,\tau)\mathrm{d}\sigma_{kl}(\tau) + \sum_n \mathcal{J}_{ijkl}(t,\tau_n)[\![\sigma_{kl}]\!]_n. \tag{3.4}$$

Notons $\dot{\sigma}_{kl}(\tau)$ la dérivée de $\sigma_{kl}(\tau)$ par rapport à τ au sens des distributions et $\delta\tau_n$ la distribution de Dirac en τ_n,

$$\dot{\sigma}_{kl}(\tau) = \begin{cases} \dfrac{\mathrm{d}\sigma_{kl}(\tau)}{\mathrm{d}\tau} & \text{si } \sigma_{kl}(\tau) \text{ est variable en } \tau, \\ [\![\sigma_{kl}]\!]_n \delta\tau_n & \text{en } \tau = \tau_n \end{cases} \tag{3.5}$$

l'intégrale de Boltzmann (3.4) peut alors être écrite comme suit :

$$\varepsilon_{ij}(t) = \int_{-\infty}^{t} \mathcal{J}_{ijkl}(t,\tau)\dot{\sigma}_{kl}(\tau)\mathrm{d}\tau. \tag{3.6}$$

Cette relation peut aussi s'écrire en contrainte en utilisant la fonction de relaxation :

$$\sigma_{ij}(t) = \int_{-\infty}^{t} \mathcal{R}_{ijkl}(t,\tau)\dot{\varepsilon}_{kl}(\tau)\mathrm{d}\tau. \tag{3.7}$$

En intégrant par parties l'équation (3.6), on constate que la réponse est la somme d'une élasticité et d'un effet mémoire tenant compte de l'histoire de la contrainte σ_{kl},

$$\varepsilon_{ij}(t) = \underbrace{\mathcal{J}_{ijkl}(t,t)\sigma_{kl}(t)}_{\text{Élasticité instantanée}} \underbrace{- \int_{-\infty}^{t} \frac{\partial \mathcal{J}_{ijkl}(t,\tau)}{\partial\tau}\sigma_{kl}(\tau)\mathrm{d}\tau}_{\text{Effet de mémoire}}. \tag{3.8}$$

On montre de la même façon que :

$$\sigma_{ij}(t) = \mathcal{R}_{ijkl}(t,t)\varepsilon_{kl}(t) - \int_{-\infty}^{t} \frac{\partial \mathcal{R}_{ijkl}(t,\tau)}{\partial\tau}\varepsilon_{kl}(\tau)\mathrm{d}\tau. \tag{3.9}$$

Le lecteur consultera le livre d'Appel [2006] pour plus de détails.

c) Matériau VE non vieillissant

On dit qu'un matériau est non vieillissant si ses propriétés n'évoluent pas avec le temps [Persoz, 1969]. Ceci signifie que sa réponse à un instant t pour une sollicitation à un instant τ antérieur à t, est la même que sa réponse à un instant $(t + \Delta\tau)$ pour la même sollicitation appliquée à l'instant $(\tau + \Delta\tau)$. Dans ce cas, les fonctions de fluage et de relaxation ne dépendent que de l'intervalle de temps $(t - \tau)$ séparant l'effet de la cause.

Appliquant cette définition à l'équation (3.8) et (3.9), on obtient :

$$\begin{cases} \varepsilon_{ij}(t) = \mathcal{J}_{ijkl}(0)\sigma_{kl}(t) + \int_{-\infty}^{t} \dfrac{\partial \mathcal{J}_{ijkl}(t-\tau)}{\partial \tau}\sigma_{kl}(\tau)\mathrm{d}\tau, \\ \sigma_{ij}(t) = \mathcal{R}_{ijkl}(0)\varepsilon_{kl}(t) + \int_{-\infty}^{t} \dfrac{\partial \mathcal{R}_{ijkl}(t-\tau)}{\partial \tau}\varepsilon_{kl}(\tau)\mathrm{d}\tau. \end{cases} \tag{3.10}$$

Pour $(t < \tau)$, les fonctions de fluage et de relaxation présentent les propriétés $(\mathcal{J}_{ijkl}(t-\tau) = 0)$ et $(\mathcal{R}_{ijkl}(t-\tau) = 0)$. Pour un matériau viscoélastique non vieillissant, les lois de comportement (3.6) et (3.7) s'écrivent alors :

$$\begin{cases} \varepsilon_{ij}(t) = \int_{-\infty}^{t} \mathcal{J}_{ijkl}(t-\tau)\dot{\sigma}_{kl}(\tau)\mathrm{d}\tau, \\ \sigma_{ij}(t) = \int_{-\infty}^{t} \mathcal{R}_{ijkl}(t-\tau)\dot{\varepsilon}_{kl}(\tau)\mathrm{d}\tau. \end{cases} \tag{3.11}$$

Mathématiquement, la forme des équations (3.11) rappelle la forme du produit de convolution de Riemann (noté ici $*$) défini pour deux fonctions quelconques f et g comme suit :

$$f(t) * g(t) = \int_{-\infty}^{t} f(t-\tau) * g(\tau)\mathrm{d}\tau. \tag{3.12}$$

Les equations (3.11) écrites en terme de produit de convolution de Riemann sont :

$$\begin{cases} \varepsilon_{ij}(t) &= \mathcal{J}_{ijkl} * \dot{\sigma}_{kl}(t), \\ \sigma_{ij}(t) &= \mathcal{R}_{ijkl} * \dot{\varepsilon}_{kl}(t). \end{cases} \tag{3.13}$$

Dans le domaine fréquentiel, la formulation fonctionnelle de la viscoélasticité s'introduit par l'usage des transformations de la Laplace et de Carson. Pour une fonction quelconque f, l'opérateur de Laplace $\mathcal{L}$ et l'opérateur $\mathcal{C}$ des transformations de Carson sont définis comme suit :

$$\begin{cases} \mathcal{L}\left[f(t)\right] = \int_{0}^{+\infty} f(t)e^{-st}\mathrm{d}t = f(s), \quad s \in \mathbb{C} \\ \mathcal{C}\left[f(t)\right] = s\int_{0}^{+\infty} f(t)e^{-st}\mathrm{d}t = f^{*}(s), \end{cases} \tag{3.14}$$

où $f(s)$ est la transformée de Laplace de $f(t)$ et $f^*(s)$ est la transformée de Carson de $f(t)$. On obtient :

$$\begin{cases} f^*(s) = sf(s), \\ \mathcal{C}\left[f(t)\right] = s\mathcal{L}\left[f(t)\right]. \end{cases} \tag{3.15}$$

La transformation de Laplace présente les propriétés suivantes :

$$\begin{cases} \mathcal{L}\left[f(t) * g(t)\right] = \mathcal{L}\left[f(t)\right]\mathcal{L}\left[g(t)\right] = f(s)g(s), \\ \mathcal{L}\left[\dot{f}(t)\right] = s\mathcal{L}\left[f(t)\right] = \mathcal{C}\left[f(t)\right] = f^*(s). \end{cases} \tag{3.16}$$

Dans l'espace de Laplace et de Carson, les lois de comportement (3.13) s'écrivent respectivement comme suit :

$$\begin{cases} \varepsilon_{ij}(s) = s\mathcal{J}_{ijkl}(s)\sigma_{kl}(s), \\ \sigma_{ij}(s) = s\mathcal{R}_{ijkl}(s)\varepsilon_{kl}(s), \end{cases} \tag{3.17}$$

et

$$\begin{cases} \varepsilon^*_{ij}(s) = \mathcal{J}^*_{ijkl}(s)\sigma^*_{kl}(s), \\ \sigma^*_{ij}(s) = \mathcal{R}^*_{ijkl}(s)\varepsilon^*_{kl}(s). \end{cases} \tag{3.18}$$

En combinant (3.17) et (3.13) on obtient :

$$\begin{cases} \varepsilon_{ij}(s) = \mathcal{J}^*_{ijkl}(s)\sigma_{kl}(s), \\ \sigma_{ij}(s) = \mathcal{R}^*_{ijkl}(s)\varepsilon_{kl}(s), \end{cases} \tag{3.19}$$

où $\mathcal{R}^*_{ijkl}$ est le tenseur d'ordre 4 des modules et $\mathcal{J}^*_{ijkl}$ est le tenseur d'ordre 4 des complaisances opérationnelles. Les équations (3.18) et (3.19) nous permettent de déduire :

$$\mathcal{R}^*_{ijkl}(s) = \left[\mathcal{J}^*_{ijkl}(s)\right]^{-1}. \tag{3.20}$$

On démontre ainsi, grâce à l'hypothèse de non vieillissement du matériau viscoélastique, que les calculs d'intégrales sont transformés en calculs opérationnels. On en déduit que les équations (3.19), sont analogues à la loi de Hooke généralisée reliant les contraintes aux déformations en élasticité linéaire :

$$\sigma_{ij} = \mathcal{C}_{ijkl}\varepsilon_{kl}, \tag{3.21}$$

où $\mathcal{C}_{ijkl}$ est le tenseur d'ordre 4 des modules élastiques. Ce principe de correspondance élastique-viscoélastique, permet de ramener la résolution d'un problème de viscoélasticité à la résolution d'un problème élastique équivalent.

d) Matériau VE soumis à des sollicitations harmoniques

On soumet un matériau VE linéaire isotrope non vieillissant à des chargements harmoniques de la forme :

$$\sigma_{ij}(t) = Re\left[\sigma^0_{ij} e^{i\omega t}\right] = \sigma^0_{ij}\cos(\omega t), \tag{3.22}$$

où $i^2 = -1$, ω est la pulsation, et Re désigne la partie réelle. D'après l'équation (3.13), les déformations résultantes sont de la forme :

$$\varepsilon_{ij}(t) = \sigma^0_{kl} \int_{-\infty}^{t} \mathcal{J}_{ijkl}(t-\tau) Re\left[i\omega e^{i\omega\tau}\right] \mathrm{d}\tau. \tag{3.23}$$

En effectuant le changement de variable : "$u = t - \tau$", on obtient :

$$\varepsilon_{ij}(t) = \sigma^0_{kl} Re\left[i\omega \int_0^{+\infty} \mathcal{J}_{ijkl}(u) e^{-i\omega u} \mathrm{d}u\right], \tag{3.24}$$

dont on déduit :

$$\begin{cases} \varepsilon_{ij}(t) & = \sigma^0_{kl} Re\left[\mathcal{J}^*_{ijkl}(i\omega) e^{i\omega t}\right], \\ \mathcal{J}^*_{ijkl}(i\omega) & = i\omega \displaystyle\int_0^{+\infty} \mathcal{J}_{ijkl}(u) e^{-i\omega u} \mathrm{d}u. \end{cases} \tag{3.25}$$

En appliquant la transformation de Carson, le tenseur $\mathcal{J}^*_{ijkl}(i\omega)$ peut être décomposé en parties réelle et imaginaire :

$$\mathcal{J}^*_{ijkl}(i\omega) = \mathcal{J}'_{ijkl}(\omega) + i\mathcal{J}''_{ijkl}(\omega), \tag{3.26}$$

où $\mathcal{J}'_{ijkl}(\omega)$ est la complaisance de stockage, et $\mathcal{J}''_{ijkl}(\omega)$ est la complaisance de perte. D'une façon analogue, pour une déformation harmonique de la forme

$$\varepsilon_{kl}(t) = Re\left[\varepsilon^0_{kl} e^{i\omega t}\right] = \varepsilon^0_{kl}\cos(\omega t), \tag{3.27}$$

la loi de comportement correspondant à la réponse en contrainte s'écrit :

$$\begin{cases} \sigma_{ij}(t) = \varepsilon^0_{kl} Re\left[\mathcal{R}^*_{ijkl}(i\omega t)e^{i\omega t}\right], \\ \mathcal{R}^*_{ijkl}(i\omega) = i\omega \displaystyle\int_0^{+\infty} \mathcal{R}_{ijkl}(u)e^{-i\omega u}\mathrm{d}u. \end{cases} \tag{3.28}$$

On décompose $\mathcal{R}^*_{ijkl}(i\omega)$ en parties réelle et imaginaire :

$$\mathcal{R}^*_{ijkl}(i\omega) = \mathcal{R}'_{ijkl}(\omega) + i\mathcal{R}''_{ijkl}(\omega), \tag{3.29}$$

où $\mathcal{R}'_{ijkl}(\omega)$ est le module de stockage et $\mathcal{R}''_{ijkl}(\omega)$ est le module de perte.

En utilisant la notation complexe comportant une partie réelle Re, une partie imaginaire Im, un module noté $|\bullet|$ et une phase φ, le tenseur de complaisances opérationnelles $\mathcal{J}^*_{ijkl}$ et le tenseur de modules opérationnels $\mathcal{R}^*_{ijkl}$ s'écrivent :

$$\begin{cases} \mathcal{R}^*_{ijkl}(i\omega) & = \left|\mathcal{R}^*_{ijkl}(i\omega)\right| e^{i\varphi(\omega)}, \\ \tan(\varphi(\omega)) & = \dfrac{Im\left(\mathcal{R}^*_{ijkl}(i\omega)\right)}{Re\left(\mathcal{R}^*_{ijkl}(i\omega)\right)} = \dfrac{\mathcal{R}''_{ijkl}(\omega)}{\mathcal{R}'_{ijkl}(\omega)} = \eta(\omega), \end{cases} \tag{3.30}$$

Les lois de comportements (3.27) et (3.28) s'écrivent alors :

$$\begin{cases} \varepsilon_{kl}(t) = \varepsilon^0_{kl}\cos(\omega t), \\ \sigma_{ij}(t) = \varepsilon^0_{kl}\left|\mathcal{R}^*_{ijkl}(i\omega)\right|\cos(\omega t + \varphi(\omega)). \end{cases} \tag{3.31}$$

On remarque que pour une sollicitation harmonique, la contrainte et la déformation sont déphasées d'un angle de $\varphi(\omega)$ (voir illustration (Fig. 3.1). L'amortissement $\eta(\omega)$ du matériau est défini par la tangente de cette angle Eq. (3.30).

Pour mettre en évidence la dissipation interne dans les matériaux VE linéaires non vieillissants, on applique une déformation harmonique uni-axiale définie comme suit :

$$\begin{cases} \varepsilon(t) = \varepsilon^0 Re\left[e^{i\omega t}\right] = \varepsilon^0\cos(\omega t), \\ \sigma(t) = \varepsilon^0\left|\mathcal{R}^*(i\omega)\right|\cos(\omega t + \varphi(\omega)), \end{cases} \tag{3.32}$$

où $\mathcal{R}^*(i\omega) = E^*(i\omega) = E'(\omega) + iE''(\omega)$ est le module de Young complexe dans le cas de

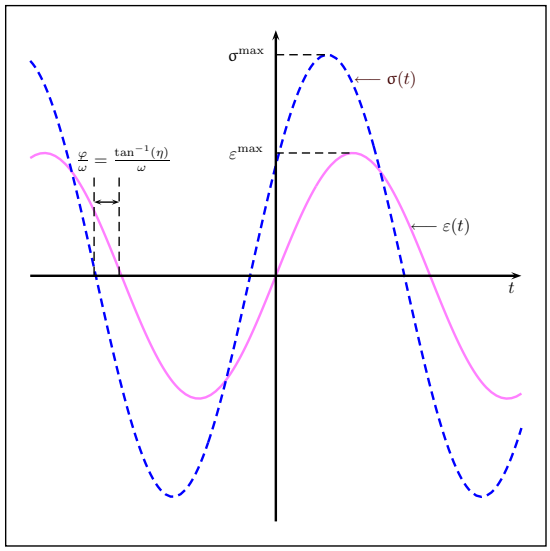

FIG. 3.1 – Déphasage contrainte-déformation

traction (compression) par exemple et $\eta(\omega) = \tan(\varphi(\omega)) = E''(\omega)/E'(\omega)$. On en déduit :

$$\begin{cases} \cos(\omega t) & = \dfrac{\varepsilon(t)}{\varepsilon^0}, \\ \sin(\omega t) & = \dfrac{1}{\eta\varepsilon^0}\left[\varepsilon(t) - \dfrac{\sigma(t)}{E'(\omega)}\right]. \end{cases} \tag{3.33}$$

Ainsi, à chaque fréquence on a $E'(\omega) = E' = \text{cste}$, ceci se traduit dans le plan $(\varepsilon - \sigma)$ par une relation contrainte/déformation elliptique d'équation :

$$\left[\frac{1}{\eta\varepsilon^0}\left(\varepsilon - \frac{\sigma}{E'}\right)\right]^2 + \left(\frac{\varepsilon}{\varepsilon^0}\right)^2 = 1. \tag{3.34}$$

Le travail dissipé :

$$E_d = \int_0^T \sigma\dot{\varepsilon}\mathrm{d}t = \eta\sigma^0\varepsilon^0\omega\int_0^{\frac{2\pi}{\omega}} \sin^2(\omega t)\mathrm{d}t = \eta\pi\sigma^0\varepsilon^0 \tag{3.35}$$

où $\sigma^0 = E'\varepsilon^0$, au cours d'un cycle ($T = 2\pi/\omega$) de déformation est l'aire de l'hystérèsis (Fig. 3.2)

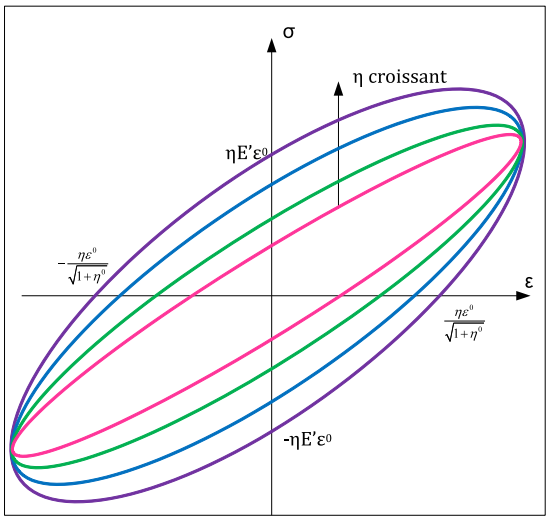

FIG. 3.2 – Hystérésis contrainte-déformation en fonction de η

L'énergie potentielle maximale vérifie alors :

$$E_p = \frac{1}{2}\sigma^0\varepsilon^0. \tag{3.36}$$

En injectant l'équation (3.36) dans (3.35), on obtient le coefficient de frottement interne η :

$$\eta = \frac{E_d}{2\pi E_p}. \tag{3.37}$$

Comme on peut le remarquer dans la figure 3.2, plus η est grand, plus la surface de l'hystérésis et l'énergie dissipée E_d sont importantes. Le coefficient de frottement interne η caractérise alors la capacité d'un matériau VE à dissiper une partie de l'énergie de déformation reçue sous forme de chaleur.

e) Effet de la fréquence et de la température sur un matériau VE

La température est le facteur d'environnement qui influence le plus les caractéristiques des matériaux viscoélastiques [Nashif *et al.*, 1985]. Les matériaux VE peuvent se présenter dans quatre états différents. Chaque état correspond à une plage de température donnée (Fig. 3.3).

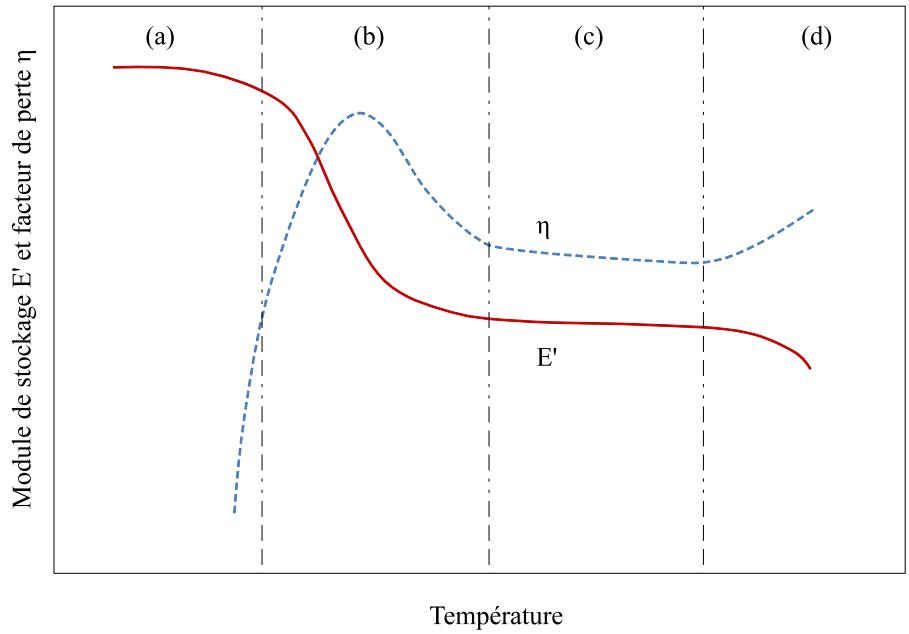

FIG. 3.3 – Variation du module et facteur de perte en fonction de la température

- L'état vitreux (a) est associé aux températures les plus faibles. Le matériau est caractérisé par un module de stockage E' maximal variant peu avec la température et un très faible facteur de perte η. De ce fait, les déformations du matériau sont très faibles.
- L'état de transition (b) est caractérisé par un module de stockage fortement décroissant avec la température et un facteur de perte présentant son maximum à la température de la transition vitreuse T_g. Cette transition vitreuse dépend de la fréquence.
- Dans l'état caoutchouteux (c), le module de stockage et le facteur de perte sont tous les deux caractérisés par des valeurs faibles. Ils varient peu avec la température.
- L'état fluide (d) est caractérisé par des instabilités dans le comportement du matériau. Cet état est très peu considéré dans les études.

Pour amortir les vibrations, il est utile de solliciter les matériaux VE au voisinage de la zone de transition (b) car le facteur de perte (responsable de l'amortissement) y atteint son maximum. Cela offre une utilisation optimale de leurs propriétés amortissantes.

Principe de superposition fréquence/température D'après des observations expérimentales [Nashif *et al.*, 1985], le profil de variation du facteur et du module de perte en fonction de la fréquence présente la même allure pour des essais effectués à différentes fréquences que pour des essais effectués à fréquence fixe en faisant varier la température. Ce

constat sur la similitude de comportement a permis de postuler une équivalence fréquence/température. Cette hypothèse se traduit par la possibilité de superposer les courbes à différentes températures sur une des courbes choisie comme référence (température de référence T_0). Cela s'effectue par une translation horizontale d'un facteur (dit facteur de décalage α_T) et correspondant à un changement d'échelle de fréquence (dite fréquence réduite $\alpha_T\omega$) sur la nouvelle courbe superposée (Fig. 3.4).

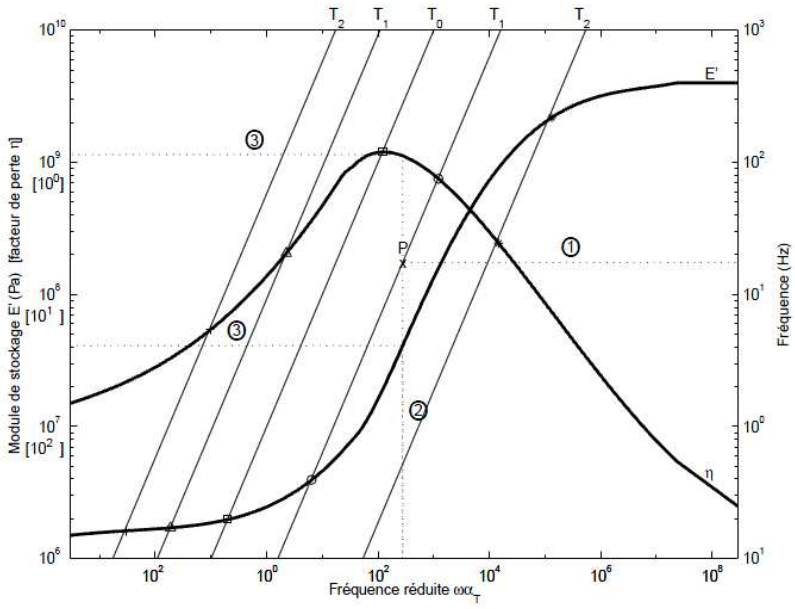

FIG. 3.4 – Courbes maitresse en fréquence réduite

La résultante de ces courbes superposées est appelée courbe maîtresse ou monogramme. Obtenue expérimentalement, elle permet de présenter sur un seul graphe, la variation du module de stockage et de facteur de perte en fonction de la température et de la fréquence. Cette courbe facilite l'obtention des propriétés du matériau en fonction de la température et de la fréquence ω. A chaque température correspond un facteur de décalage α_T qui définit une droite isotherme dans le plan $(\alpha_T\omega, \omega)$. Pour une fréquence quelconque ω_i et une température T_i, on lit l'abaque donnée par la figure 3.4 comme suit :

1. on repère le point d'intersection P de la droite horizontale ω_i et de l'isotherme oblique T_i,

2. l'abcisse du point P donne la valeur de $\omega_i\alpha_{T_i}$,

3. finalement on lit les valeurs de η et de E' directement sur la courbe.

Pour des besoins de calcul, on peut modéliser l'évolution du module et du facteur de perte

sous forme de fonctions analytiques par interpolation des courbes maitresse :

$$E^*(\alpha(T)\omega) = E'(\alpha_T\omega)\left[1 + i\eta(\alpha_T\omega)\right], \tag{3.38}$$

où $\alpha_T = \alpha(T)$ est une fonction dépendant de la température permettant le passage de l'échelle des fréquences réelles à l'échelle des fréquences réduites. Plusieurs lois empiriques permettent de déterminer le facteur de décalage α_T en échelle logarithmique, parmi celles-ci figure la loi de WLF [Williams *et al.*, 1955] (Wiliams-Landel-Ferry). Elle permet de faire la conversion de fréquence réduite $\alpha_T\omega$ en fréquence physique ω, pour des températures autres que celles mesurées expérimentalement. Cette loi est appliquée dans la zone de transition vitreuse. A des températures supérieures à T_g (température de transition vitreuse), sur une plage de température d'environ $(-T_g; T_g + 50)$, la loi de WLF décrit le facteur de décalage en température α_T comme suit :

$$log\alpha_T = \frac{C_1^0(T - T_0)}{C_2^0 + (T - T_0)}, \tag{3.39}$$

où C_1^0 et C_2^0 (donnés en $°C$) sont des cœfficients viscoélastiques qui caractérisent le matériau.

3.2.2 Modélisation rhéologique de la loi viscoélastique

Pour approcher la loi VE par un modèle analytique, le module de stockage (ou le module d'élasticité) est modélisé par un ressort qui représente un élément élastique idéal. La contrainte est alors proportionnelle à la déformation et est définie par la loi de Hooke : $\sigma(t) = E\varepsilon(t)$ (E est le module d'élasticité). Le module de perte (ou le module de viscosité) est modélisé par un amortisseur qui présente un élément visqueux dissipatif, la contrainte est alors proportionnelle à la vitesse de déformation. Elle est définie par la loi de frottement visqueux de Newton : $\sigma(t) = \eta\dot{\varepsilon}(t)$. C'est ce qu'on appelle un modèle rhéologique [Sun et Lu, 1995].

Cette combinaison peut être représentée de manière simple en associant un ressort et un amortisseur en série et/ou en parallèle (modèle de base) ou d'ordre élevé en associant plusieurs modèles de base (modèles composés ou généralisés). On trouve généralement deux grandes familles de modèles rhéologiques : les modèles standards linéaires (simples ou d'ordres élevés) et les modèles fractionnaires. Dans le cadre de ce travail, on s'intéresse aux modèles standards linéaires. Le lecteur intéressé par les modèles fractionnaires consultera la thèse de Bilasse [2010].

a) Modèles standards linéaires

Ressorts et amortisseurs sont combinés d'une façon linéaire (Fig. 3.5)

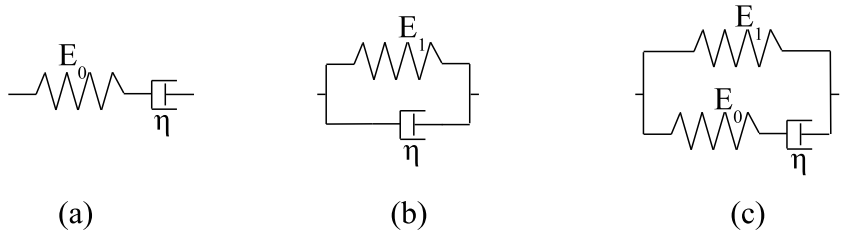

FIG. 3.5 – Modèle d'amortissement à deux ou trois paramètres : (a) Modèle de Maxwell ; (b) Modèle de Kelvin-Voigt ; (c) Modèle de Zener

.

Le matériau amortissant est approché par un modèle analytique simple qui possède peu de paramètres (deux ou trois seulement). Les lois de comportement ainsi que les modules complexes des matériaux viscoélastiques linéaires simples (Fig. 3.5) sont écrits dans la table 3.2 :

Modèle rhéologique	Loi de comportement	Module complexe $\mathcal{R}^*(i\omega)$
Maxwell	$\sigma(t) + \frac{\eta}{E_0}\dot{\sigma}(t) = \eta\dot{\varepsilon}(t)$	$\frac{i\omega\eta E_0}{E_0 + i\omega\eta}$
Kelvin-Voigt	$\sigma(t) = E_1\varepsilon(t) + \eta\dot{\varepsilon}(t)$	$E_1 + i\omega$
Zener	$\sigma(t)(E_0 + E_1) + \eta\dot{\sigma}(t)$	$\frac{E_0(E_1 + i\omega\eta)}{E_0 + E_1 + i\omega\eta}$

TAB. 3.1 – Modèles rhéologiques linéaires simples

b) Modèle d'ordre élevé

Pour généraliser les modèles précédents et obtenir des modèles d'ordre élevé, on a souvent recours soit à une fraction rationnelle :

$$\mathcal{R}^*(i\omega) = E_0 \frac{1 + \alpha_1 i\omega + ... + \alpha n_n (i\omega)^{n_n}}{1 + \beta_1 i\omega + ... + \beta_{n_d} (i\omega)^{n_d}}, \quad \text{avec } n_d, n_n \in \mathbb{N}, \tag{3.40}$$

soit à une fraction de polynômes à dérivées fractionnaires :

$$\mathcal{R}^*(i\omega) = E_0 \frac{1 + \sum_{n=1}^{\infty} b_n (i\omega)^{\beta_n}}{1 + \sum_{n=1}^{\infty} a_n (i\omega)^{\alpha_n}}, \quad \text{avec } 0 < \alpha_n \text{ et } \beta_n < 1. \tag{3.41}$$

On trouve classiquement deux représentations de modèles de la forme (3.40) : le modèle de Kelvin-Voigt généralisé et le modèle Maxwell généralisé présentés en (Fig. 3.6), ils sont basés sur une combinaison des modèles rhéologiques présentés en (Fig. 3.5).

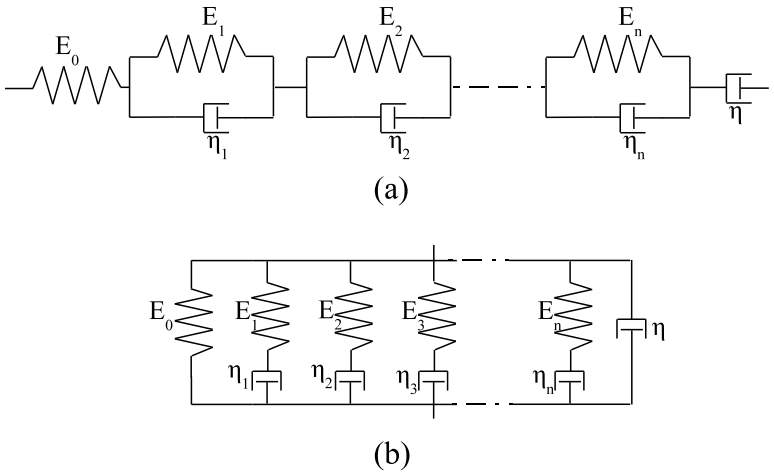

FIG. 3.6 – Modèles généralisés : (a) Kelvin-Voigt généralisé ; (b) Maxwell généralisé

Les lois de comportement ainsi que les modules complexes des matériaux viscoélastiques linéaires d'ordre élevé sont écrits dans la table 3.2 :

Modèle rhéologique	Loi de comportement	Module complexe $\mathcal{R}^*(i\omega)$
Kelvin-Voigt généralisé	$\sigma(t) + \sum_{n=1}^{N_1} a_n \frac{\mathrm{d}^n \sigma(t)}{\mathrm{d}t^n} = b_0 + \sum_{n=1}^{N_2} b_n \frac{\mathrm{d}^n \varepsilon(t)}{\mathrm{d}t^n}$	$\left[\frac{E_0 + i\omega\eta}{i\omega E_0 \eta} + \sum_{n=1}^{N} \frac{1}{E_n + i\omega\eta_n} \right]^{-1}$
Maxwell généralisé	$\sigma(t) + \sum_{n=1}^{N_1} a_n \frac{\mathrm{d}^n \sigma(t)}{\mathrm{d}t^n} = b_0 + \sum_{n=1}^{N_2} b_n \frac{\mathrm{d}^n \varepsilon(t)}{\mathrm{d}t^n}$	$E_0 + i\omega\eta + \sum_{n=1}^{N} \frac{i\omega\eta_n E_n}{E_n + i\omega\eta_n}$

TAB. 3.2 – Modèles rhéologiques linéaires d'ordre élevé

Les coefficients N, N_1 et N_2 sont entiers, a_n et b_n sont des coefficients caractéristiques du matériau. Ces paramètres sont obtenus en utilisant les courbes maitresse comme décrit par Landier [1993].

3.3 Méthodes de résolution du problème des vibrations linéaires

Généralement, la modélisation de la vibration libre linéaire d'une structure viscoélastique mène à des problèmes à valeurs propres complexes non-linéaires. Une fois la

structure discrétisée par éléments finis par exemple, ce problème aux valeurs propres peut être écrit sous la forme suivante :

$$[\mathbf{K}(\omega) - \omega^{\mathbf{2}}\mathbf{M}]\mathbf{U} = \mathbf{0}, \tag{3.42}$$

où la matrice de rigidité $\mathbf{K}(\omega)$ et le mode propre de vibration $\mathbf{U}$ sont complexes et dépendant de la fréquence ω, et la matrice de masse $\mathbf{M}$ est considérée réelle. On note $(\mathbf{U}, \omega)$ les solutions propres de ce problème. Dans l'étude des vibrations linéaires des structures sandwich viscoélastiques, les ingénieurs sont principalement intéressés par les propriétés amortissantes définies par deux paramètres : la fréquence de résonance Ω et l'amortissement η. Les solutions propres $(\mathbf{U}, \omega)$ permettent de calculer l'amortissement η et la fréquence de résonance Ω en utilisant la formule (3.43) proposée par DiTaranto et Blasingame [1966] :

$$\omega^2 = \Omega^2(1 + i\eta), \quad \text{où} \quad \eta = \frac{Im(\omega^2)}{Re(\omega^2)} \quad \text{et} \quad \Omega = \sqrt{Re(\omega^2)}. \tag{3.43}$$

Dans le cadre des vibrations linéaires, les techniques de calcul d'amortissement passif peuvent être classées selon deux grandes catégories : méthodes approchées et méthodes directes.

3.3.1 Méthodes approchées

a) Méthode d'énergie de déformation modale (Modal Strain Energy Method , MSEM).

Dans cette méthode, l'amortissement est défini comme le rapport de l'énergie de déformation dissipée dans la couche VE sur l'énergie de déformation stockée [Johnson et Kienholz, 1982]. On trouve deux versions de cette méthode :

MSER utilisant le mode réel : Dans cette version, le mode propre réel du sandwich VE est supposé proportionnel au mode propre de la structure non-amortie $\mathbf{U}_r$ qui vérifie le problème aux valeurs propres réelles

$$\begin{cases} [\mathbf{K}(\mathbf{0}) - \omega_{\mathbf{0}}^{\mathbf{2}}\mathbf{M}]\,\mathbf{U}_r = 0, \\ \mathbf{U}_r^t\mathbf{K}(\mathbf{0})\mathbf{U_r} - \omega^{\mathbf{2}}\mathbf{U_r^t}\mathbf{M}\mathbf{U_r} = \mathbf{0}. \end{cases} \tag{3.44}$$

où $\mathbf{K}(\mathbf{0})$ est la matrice de rigidité construite en considérant uniquement l'élasticité retardée du matériau VE. Le problème aux valeurs propres complexes (3.42) s'écrit alors sous la

forme :

$$\begin{cases} [\mathbf{K}(\omega) - \omega^{\mathbf{2}}\mathbf{M}]\mathbf{U_r} = \mathbf{0}, \\ \omega^2 = \Omega^2(1 + i\eta). \end{cases} \tag{3.45}$$

On approche ensuite $\mathbf{K}(\omega)$ par $\mathbf{K}(\mathbf{0})$. En utilisant la formule (3.43), on obtient une approximation de l'amortissement de la fréquence de résonance Ω après avoir approché l'amortissement par :

$$\begin{cases} \Omega^2 = \dfrac{\mathbf{U}_r^t \mathbf{K}^{\mathbf{Re}}(\mathbf{0})\mathbf{U_r}}{\mathbf{U}_r^t \mathbf{M} \mathbf{U}_r}, \\ \eta = \dfrac{\mathbf{U}_r^t \mathbf{K}^{\mathbf{Im}}(\mathbf{0})\mathbf{U_r}}{\mathbf{U}_r^t \mathbf{K}^{\mathbf{Re}}(\mathbf{0})\mathbf{U_r}}. \end{cases}$$

On notera $\mathbf{K}^{\mathbf{Re}}$ la partie réelle de la quantité $\mathbf{K}$ et $\mathbf{K}^{\mathbf{Im}}$ sa partie imaginaire.
Cette méthode n'est précise que pour des structures faiblement amorties [Daya *et al.*, 2002].

MSEC utilisant le mode complexe : Cette version utilise la même technique sauf que $\mathbf{U}$ est approché par un mode propre complexe $\mathbf{U}_c$ calculé à partir du problème de vibration forcée suivant :

$$[\mathbf{K}(\omega_{\mathbf{0}}) - \omega_{\mathbf{0}}^{\mathbf{2}}\mathbf{M}]\mathbf{U_c} = \mathbf{F}, \tag{3.46}$$

où F est une amplitude du forcing harmonique. Cette méthode donne de très bonnes estimations [Soni, 1981] avec les formules discutées dans Bilasse [2010], Bilasse *et al.* [2011]. La méthode devient inutilisable dans le cas où $\mathbf{K}$ dépend de ω [Rikards *et al.*, 1993].

b) Méthode de réponse en fréquence

Cette méthode consiste à considérer l'analyse de la réponse forcée en régime harmonique. L'équation (3.42) devient alors :

$$[\mathbf{K}(\omega) - \omega^{\mathbf{2}}\mathbf{M}]\mathbf{U}(\omega) = \mathcal{F}, \tag{3.47}$$

où $\mathcal{F}$ est le vecteur des forces sinusoïdales appliquées. On cherche ensuite la solution $\mathbf{U}(\omega)$ (courbes de réponses), ce qui impose la composition puis la décomposition de la matrice de rigidité dynamique complexe $(\mathbf{K}(\omega) - \omega^{\mathbf{2}}\mathbf{M})$ pour chaque pulsation ω. Des pics de résonance sont mis en évidence à partir des courbes de réponses en fréquence obtenues par des essais expérimentaux de type Oberst schématisés par la figure 3.7. En utilisant la

méthode de bande passante à $-3dB$ Soni [1981], il suffit de calculer le rapport : largeur à mi-hauteur de la courbe de résonance Δf sur la fréquence de résonance Ω (Fig. 3.8) pour déterminer l'amortissement expérimental pour un mode et une température donnés :

$$\eta = \frac{\Delta f}{f}. \tag{3.48}$$

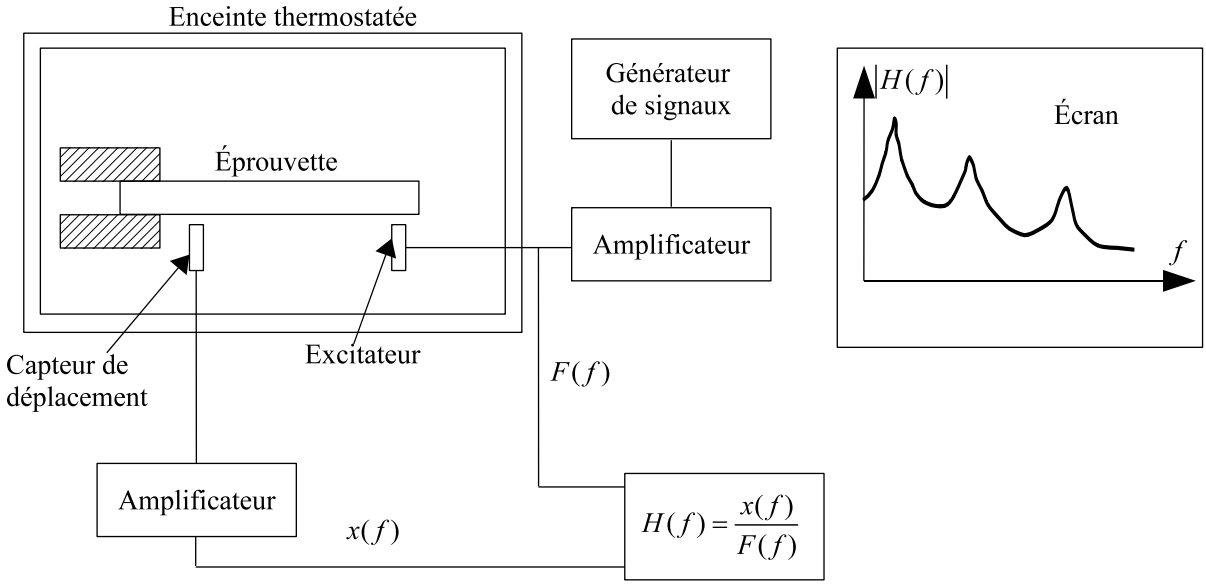

FIG. 3.7 – Fonctionnement de l'appareil de flexion Oberst

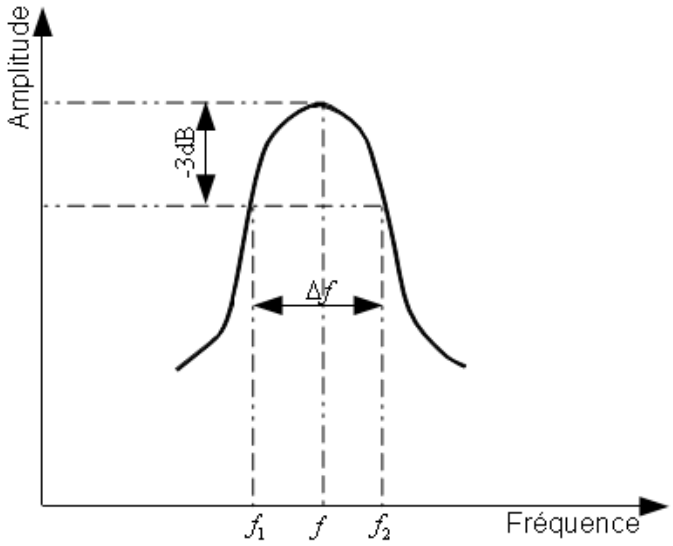

FIG. 3.8 – Courbe de résonance au voisinage d'une fréquence de résonance

L'un des inconvénients majeurs de cette méthode est que l'obtention des courbes de réponse est très coûteux lorsqu'une solution complète est exigée. Cette méthode n'est valable que pour des fréquences propres suffisamment séparées [Abdoun *et al.*, 2009, 2010].

3.3.2 Méthodes directes

Le problème non linéaire aux valeurs propres complexes (3.42) est résolu explicitement.

a) La méthode *QR*

Cette méthode permet de calculer les valeurs propres d'une matrice complexe constante en décomposant cette matrice en une matrice Q orthogonale et une matrice R triangulaire supérieure pour faciliter le calcul des valeurs propres complexes d'une façon itérative [Bathe, 1982, Brezinski et Redivo-Zaglia, 2006]. L'utilisation de cette méthode est limitée aux structures sandwich VE de petite taille ayant un module complexe indépendant de la fréquence.

b) Méthode d'itération-réduction

Proposée par Chen *et al.* [1999], cette méthode utilise une méthode de perturbation couplée à un algorithme de Lanczos et à une base réduite. Elle permet d'introduire une loi viscoélastique quelconque. Cette méthode est une variante de la méthode de Newton. Par conséquent, elle nécessite plusieurs itérations par fréquence, ce qui induit un temps de calcul très élevé pour des structures de grande taille.

c) La méthode asymptotique

Cette méthode fut initiée par Ma et He [1992]. Elle consiste à chercher le mode et la valeur propre sous la forme de séries entières du facteur de perte. Le matériau viscoélastique est supposé de module constant. La méthode est efficace pour les structures faiblement amorties. Lorsque le module varie de façon non-linéaire avec la fréquence, une procédure de continuation est nécessaire.

d) La Méthode Asymptotique Numérique (MAN)

Les algorithmes basés sur la méthode Asymptotique Numérique(MAN) couplent la technique de perturbation et l'homotopie [Daya et Potier-Ferry, 2001, Duigou *et al.*,

2003]. Ils utilisent un développement en séries de Taylor. Cette méthode sera détaillée au paragraphe 3.3.3.

e) Les méthodes de sous-espace de Krylov et d'Arnoldi non linéaires

Ces méthodes permettent de résoudre un problème aux valeurs propres complexes non linéaire creux et de grande taille sous la forme :

$$T(\omega)x = 0; \quad T(\omega) \in \mathbb{C}^{n\times n}, x \in \mathbb{C}^n. \tag{3.49}$$

On décompose le problème (3.49) en un ensemble de problèmes linéaires qu'on résout d'une manière itérative en les projetant dans des sous-espaces de type Krylov. Ces sous-espaces permettent de construire, par des opérations de type produit matrice-vecteur, produit scalaire ou combinaison linéaire de vecteurs, des sous-espaces affines pour chercher des approximations de la solution du système linéaire [Meurant, 1999].

3.3.3 Méthode Asymptotique Numérique

Basés sur la MAN avec homotopie, plusieurs algorithmes numériques ont été développés [Daya et Potier-Ferry, 2001, Duigou *et al.*, 2003] pour résoudre le problème (3.42), où la matrice de rigidité $K(\omega)$ dépendante de ω est écrite sous la forme :

$$\mathbf{K}(\omega) = \mathbf{K_0} + \mathcal{E}(\omega)\mathbf{K_c}. \tag{3.50}$$

Dans (3.50), la matrice d'élasticité $\mathbf{K_0} = \mathbf{K}(\mathbf{0}) = \mathbf{E}(\mathbf{0})\mathbf{K_c}$ liée au module d'élasticité retardé est symétrique et définie positive. La matrice K_c est une matrice constante. L'équation (3.42) s'écrit :

$$\mathbf{R}(u,\omega) = [\mathbf{K_0} + \mathbf{E}(\omega)\mathbf{K_c} - \omega^{\mathbf{2}}\mathbf{M}]\mathbf{U} = \mathbf{0}. \tag{3.51}$$

Elle est divisée en deux parties :

$$\mathbf{R}(\mathbf{U},\omega) = \mathbf{S}(\mathbf{U},\omega) + \mathbf{T}(\mathbf{U},\omega) = 0, \tag{3.52}$$

avec

$$\mathbf{S}(u,\omega) = [\mathbf{K_0} - \omega^{\mathbf{2}}\mathbf{M}]\mathbf{U}, \quad \text{et} \quad \mathbf{T}(\mathbf{u},\omega) = \mathbf{E}(\omega)\mathbf{K_c}\mathbf{U}. \tag{3.53}$$

Au contraire du problème à valeurs propres complexes, le problème à valeurs propres réelles "$S(\mathbf{U},\omega)=0$" peut être résolu par une méthode directe.
Initialement, Daya et Potier-Ferry [2001] ont proposé une méthode de continuation qui utilise une technique d'homotopie. Le problème (3.52) devient alors :

$$\mathbf{R}(\mathbf{U},\omega)=\mathbf{S}(\mathbf{U},\omega)+a\mathbf{T}(\mathbf{U},\omega)=0, \tag{3.54}$$

où $a\in[0,1]$ est un paramètre de chemin. Pour $a=0$, on retrouve le problème à valeurs propres réelles. Pour $(a=1)$, on retrouve le problème de départ (3.50) à valeurs propres complexes. Cette homotopie permet de résoudre le problème (3.42) par une méthode de continuation.
Lorsque la loi viscoélastique est analytique, les branches de solutions sont cherchées par la MAN [Daya et Potier-Ferry, 2001]. Les solutions $(\mathbf{U}(a),\omega(a))$ du problème perturbé (3.52) sont approchées par des séries de Taylor tronquées à un ordre N :

$$\begin{cases}\mathbf{U}(a+\delta_a) & =\sum_{n=0}^{N}\delta_a^n\mathbf{U}_n(a),\ \text{où}\ \mathbf{U}_n(a)=\dfrac{1}{n!}\dfrac{\partial^n\mathbf{U}}{\partial a^n}(a)\\ \omega(a+\delta_a) & =\sum_{n=0}^{N}\delta_a^n\omega_n(a),\ \text{où}\ \omega_n(a)=\dfrac{1}{n!}\dfrac{\partial^n\omega}{\partial a^n}(a),\end{cases} \tag{3.55}$$

sont les coefficients de Taylor de $\mathbf{U}$ et ω à l'ordre n évalués au point a. En injectant ces séries dans l'équation (3.54), on obtient une séquence de N systèmes linéaires dépendants ayant la même matrice linéaire tangente et un membre de droite composé de dérivées d'ordre élevé. Ces séries de Taylor peuvent être calculées explicitement [Daya et Potier-Ferry, 2001] ou évaluées par la DA [Charpentier, 2008] par le biais de l'outil Diamant.

On introduit une condition d'orthonormalisation pour former le système (3.54) :

$$\mathbf{U}_0^t(\mathbf{U}-\mathbf{U}_0)=0 \tag{3.56}$$

Le problème $(a\rightarrow\{\mathbf{U}(a),\omega(a)\})$ est alors résolu par une technique de perturbation.

La technique de perturbation : La MAN est utilisée pour résoudre le problème perturbé [Daya et Potier-Ferry, 2001]. Cette méthode consiste à approcher les inconnues

du problème $\mathbf{U}$ et ω par les séries de Taylor :

$$\begin{cases} \mathbf{U} & = \sum_{n=0}^{N} a^n U_n, \\ \omega & = \sum_{n=0}^{N} a^n \omega_n, \\ \omega^2 & = \sum_{n=0}^{N} a^n p_n, \\ 0 \leq a \leq 1. \end{cases} \tag{3.57}$$

Avec N est l'ordre de troncature des séries. La loi VE $E(\omega)$ dépendant de la fréquence s'écrit aussi à l'aide d'une série de Taylor :

$$E(\omega) = \sum_{n=0}^{N} a^n E_n. \tag{3.58}$$

En injectant les équations (3.57) et (3.58) dans l'équation (3.54) et après identification des puissances en a, on obtient une séquence de problèmes linéaires :

– problème d'ordre 1

$$\begin{cases} \mathbf{A}\mathbf{U}_0 & = 0, \\ \mathbf{A} & = \mathbf{K}(\mathbf{0}) - \mathbf{p_0}\mathbf{M}, \\ p_0 & = -\omega_0^2, \text{ c'est le problème à valeurs propres réelles.} \end{cases} \tag{3.59}$$

– problème d'ordre $(n+1)$, tel que $n > 1$

$$\begin{cases} \mathbf{A}\mathbf{U}_n & = F_n, \\ F_n & = -\sum_{k=0}^{n-1} E_k \mathbf{K}\mathbf{U}_{\mathbf{n-k-1}} + \sum_{\mathbf{k=1}}^{\mathbf{n}} \mathbf{p_k}\mathbf{M}\mathbf{U}_{\mathbf{n_k}}. \end{cases} \tag{3.60}$$

Pour les problèmes d'ordre élevé, on obtient des problèmes linéaires non-homogènes faisant tous intervenir la même matrice $\mathbf{A}$.
Les inconnues du problème à l'ordre $(n+1)$ sont $\mathbf{U}_n$ et p_n. Comme la matrice $\mathbf{A}$ est singulière, son noyau est généré à partir du mode non amortie $\mathbf{U}_0$, on a besoin d'ajouter une condition de solvabilité [Daya et Potier-Ferry, 2001] :

$$F_n^t \mathbf{U}_0 = 0. \tag{3.61}$$

Ce qui nous permet de calculer les p_n :

$$\begin{cases} p_1 &= \frac{E_0 U_0^t \mathbf{K} \mathbf{U_0}}{U_0^t \mathbf{M} U_0}, \quad n = 1, \\ p_n &= \frac{\sum_{k=0}^{n-1} E_k U_0^t \mathbf{K} \mathbf{U_{n-k-1}} - \sum_{\mathbf{k=1}}^{\mathbf{n-1}} \mathbf{p_k} \mathbf{U_0^t} \mathbf{M} \mathbf{U_{n-k}}}{U_0^t \mathbf{M} U_0}, \quad n \geq 2. \end{cases} \tag{3.62}$$

Le problème est résolu par continuation.

3.4 Problématique et constats

Les différences observées entre les simulations réalisées par [Daya et Potier-Ferry, 2001] et les essais effectués par [Landier, 1993] (Tab. 3.4), ont motivé notre étude sur l'identification des paramètres viscoélastiques des tôles acier/polymère/acier.

3.4.1 Essais expérimentaux

Durant sa thèse, Landier [1993] a réalisé chez USINOR (Arcelor Mittal Steel aujourd'hui) des essais d'Oberst (voir Fig. 3.7) pour des températures et des fréquences différentes. Les tôles sandwich utilisées dans ces essais ont été fabriquées par SOLLAC. La ligne pilote de fabrication de ces tôles sandwich est schématisée dans la Fig. 3.9. Les deux parements en acier sont tout d'abord préchauffés dans des fours à infrarouges distincts. Un film en polymère est dévidé entre les deux fours. Ensuite, les trois couches sont calandrées entre deux cylindres afin de les coller. Le maintien en température favorise ensuite l'adhérence des trois couches. Le sandwich obtenu est refroidi à l'air libre avant d'être bobiné. Les deux tôles en acier sont supposées élastiques, homogènes et isotropes. Le film en polymère est supposé viscoélastique, linéaire, non vieillissant, homogène et isotrope.

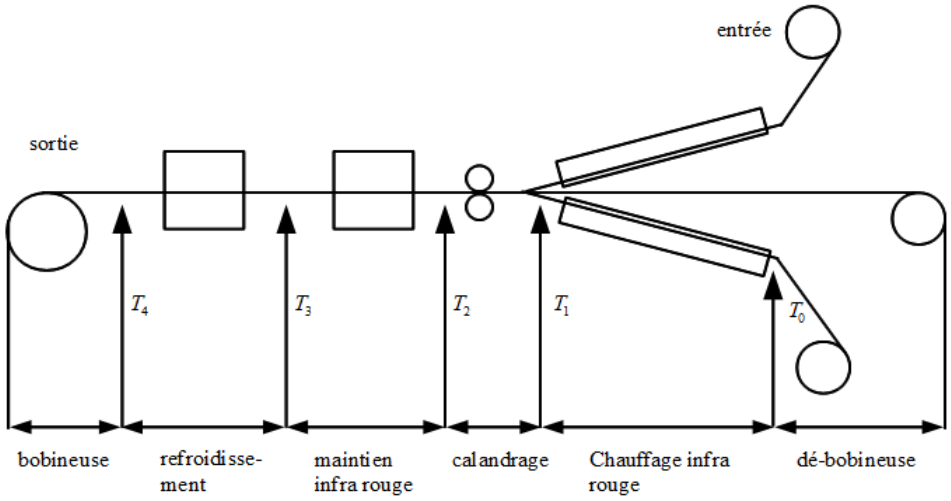

FIG. 3.9 – Ligne pilote de fabrication de tôles sandwich (Landier [1993])

Les différentes caractéristiques de l'acier et du polymère ainsi que la géométrie de la poutre utilisée pour ces essais sont résumées dans le tableau 3.3.

TAB. 3.3 – Propriétés des matériaux et géométrie de la poutre.

Couche élastique	Module de Young $E_l = 2.1 \times 10^{11} Pa$
	Coefficient de Poisson $\nu_l = 0.3$
	Densité $\rho_l = 7800 Kg.m^{-3}$
	Épaisseur $h_l = 0.6mm$
Cœur viscoélastique	Module de Young $E_c = 2.7216 \times 10^{7} Pa$
	Coefficient de Poisson $\nu_c = 0.44$
	Densité$\rho_c = 1200 Kg.m^{-3}$
	Épaisseur $h_=0.045mm$
Poutre	Longueur $L = 178mm$
	Largeur $b = 10mm$

Dans ses simulations, [Landier, 1993] a utilisé le modèle de rhéologique de Maxwell généralisé pour décrire le comportement viscoélastique du polymère (3.63) :

$$E(\omega) = \alpha_0 + i\beta_0\omega + \sum_{j=1}^{129} \frac{i\omega}{\frac{i\omega}{\alpha_j} + \frac{1}{\beta_j}}. \tag{3.63}$$

Son modèle contient 260 paramètres. Ces paramètres ont été identifiés en utilisant les courbes maitresse du polymère. Ils nous ont été fournis par l'entreprise Usinor.
Les courbes de variation du coefficients d'amortissement en fonction de la fréquence pour différentes températures sont données par le graphe (Fig. **??**).

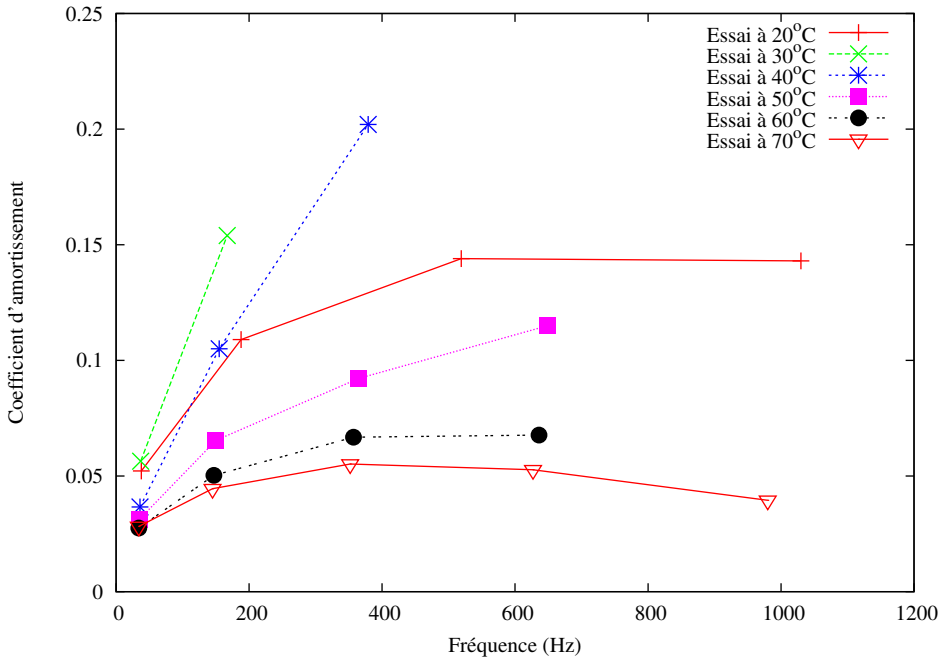

FIG. 3.10 – Amortissement de l'éprouvette sandwich : courbes issues d'essais expérimentaux (essais oberst effectués par USINOR en 1995)

3.4.2 Simulation Numérique

Daya et Potier-Ferry [2001] ont reproduit par simulation la vibration libre d'une poutre sandwich (acier/polymère/acier) dans un code Fortran basé sur la MAN. Ce code a pour entrée une structure sandwich discrétisée, un ensemble de paramètres p qui définissent la loi viscoélastique et une fréquence ω. Les sorties du code sont une fréquence de résonance Ω et un coefficient de perte modale η_m. Le modèle de Maxwell généralisé (3.63) est implémenté dans le code.

En se mettant dans le cas de l'hypothèse de contraintes planes, la structure est discrétisée en utilisant 6 éléments dans l'épaisseur et 30 éléments dans la longueur. L'élément de maillage utilisé est un élément triangulaire à trois nœuds décrit dans [Daya et Potier-Ferry, 2002]. Dans cet élément, basé sur la théorie discrète de Kirchhoff, chaque nœud possède huit degrés de liberté qui sont les déplacements longitudinaux des couches élastiques, la flèche commune et trois rotations. L'ordre de troncature des séries de la MAN est $N = 20$ alors que le paramètre ε intervenant dans l'estimation du rayon de convergence est choisi égal à 10^{-6}. Les différents paramètres géométriques sont résumés par le tableau (Tab. 3.3).

Les résultats de simulation avec le modèle de Maxwell généralisé à 260 paramètres sont comparés aux résultats expérimentaux de Landier dans le tableau 3.4

TAB. 3.4 – Fréquences naturelles et facteurs de perte de la poutre sandwich.

	Résultats expérimentaux		Résultats numériques		Erreurs relatives	
	Ω^{exp}	η_m^{exp}	Ω^{MG}	η_m^{MG}	$\frac{\Omega^{MG}-\Omega^{\text{exp}}}{\Omega^{\text{exp}}}$	$\frac{\eta_m^{MG}-\eta_m^{\text{exp}}}{\eta_m^{\text{exp}}}$
60°C	34	2.75×10^{-2}	30.60	3.58×10^{-2}	1.00×10^{-1}	3.02×10^{-1}
	147	5.03×10^{-2}	156.27	7.56×10^{-2}	6.31×10^{-2}	5.03×10^{-1}
	357	6.68×10^{-2}	382.77	7.52×10^{-2}	7.22×10^{-2}	1.26×10^{-1}
70°C	34	2.82×10^{-2}	30.25	2.92×10^{-2}	1.10×10^{-1}	3.55×10^{-2}
	145	4.45×10^{-2}	152.49	8.60×10^{-2}	5.17×10^{-2}	9.33×10^{-1}
	352	5.52×10^{-2}	376.11	7.88×10^{-2}	6.85×10^{-2}	4.26×10^{-1}

On y observe que les fréquences de résonance sont déterminées avec une erreur relative inférieure à 10%. Par contre, les erreurs sur les facteurs de perte modaux sont plus au moins importantes selon le mode de vibration. Ces erreurs peuvent être dues au fait que les différents paramètres de la loi viscoélastique utilisée ont été identifiés expérimentalement à partir des courbes maitresse du polymère seul [Landier, 1993]. Il est légitime de penser que le polymère intercalé entre les deux parements en acier de la tôle sandwich, peut ne pas se comporter de la même façon. Les erreurs de mesure ne sont pas négligeables et peuvent aussi contribuer à ces différences.

Pour améliorer les résultats de simulation, on propose dans ce travail une identification de la loi VE lorsque le matériau VE est contraint. La méthode d'optimisation retenue utilise des gradients obtenus par différentiation automatique est décrite dans le paragraphe 3.5.

3.4.3 Choix d'une loi viscoélastique pour l'identification

Pour simplifier, le module viscoélastique du cœur est écrit sous la forme :

$$E(\omega) = E_0 + \mathcal{E}(\omega), \tag{3.64}$$

où la constante E_0 est le module d'élasticité retardé et $\mathcal{E} = (E_R(\omega), E_I(\omega))$ est la fonction complexe décrivant la partie du module d'Young dépendant de la fréquence. Pour fréquence donnée, les propriétés viscoélastiques de la couche centrale satisfont la formulation du module de Young complexe "constant" [Soni, 1981, Thompson et Walker, 1968] :

$$E = Re(E)(1 + i\eta), \tag{3.65}$$

où $Re(E)$ est la partie réelle du module élastique et η est le facteur de perte du matériau VE.

En pensant à l'identification, on introduit la fonction mathématique $\mathcal{R}$:

$$\mathcal{R} \circ \mathcal{E}(\omega, p) = (\Omega(\omega, p), \eta_m(\omega, p)), \tag{3.66}$$

pour lier l'ensemble des paramètres p caractérisant la loi viscoélastique $\mathcal{E}(\omega, p)$ à la fréquence de résonance et au facteur de perte.

Les différentes étapes de calcules présentés dans le (cf. § 3.3.3) pour résoudre problème (3.52) sont données par le tableau (Tab. 3.5).

TAB. 3.5 – Solveur du problème aux valeurs propres complexes

Méthode des sous-espaces			
Pour le mode choisi			
→ Continuation	→Coef_Taylor pour $\mathcal{E}$	→ Calcul des séries	
		→ solveur linéaire	

Dans le cadre de ce travail, on propose une méthode d'identification du coefficient d'amortissement η et de la fréquence de résonance Ω qui combine simulation numérique et données expérimentales. Pour cela, on utilise le code éléments finis [Daya et Potier-Ferry, 2001] où la résolution du problème (3.42) est effectué par la MAN.

3.5 Identification de la loi viscoélastique

Le problème de contrôle de la vibration d'une structure sandwich à couches viscoélastiques ou/et piézoélectriques, est caractérisé par un grand nombre de paramètres liés au matériau (coefficient d'amortissement η et le module d'élasticité E, pour le matériau VE par exemple) et à sa géométrie (l'épaisseur des couches, leur nombre...). Dans ce contexte, l'objectif est d'identifier certains de ces paramètres afin de maximiser l'amortissement modal et la rigidité. Les paramètres à identifier peuvent être l'épaisseur et le module de cisaillement de la couche contrainte viscoélastique comme dans [Barkanov *et al.*, 2009, Baz et Ro, 1995], ou les dimensions et l'emplacement des patchs piézoélectrique comme dans [Araújo *et al.*, 2009].

3.5.1 Modélisation inverse

Les différences observées entre les résultats numériques et les résultats expérimentaux (Tab. 3.4) peuvent être corrigées en identifiant une loi viscoélastique $\mathcal{E}^*$ valable pour la structure sandwich considérée. Pour simplifier, on ignore la dépendance de Ω et η_m à ω. Dans la suite nous cherchons à identifier les paramètres $p = (E_R, E_I)$, pour différents

modes et différentes températures. Pour des données $(\Omega^{\text{exp}}, \eta^{\text{exp}})$ pour un mode particulier et à une température particulière, le critère à minimiser est une mesure $\mathcal{J}$ de l'écart entre la fréquence et de facteur de perte modal calculés et ceux observés est :

$$\mathcal{J}(\Omega, \eta_m) = \frac{(\Omega - \Omega^{\text{exp}})^2}{(\Omega^{\text{exp}})^2} + \frac{(\eta_m - \eta_m^{\text{exp}})^2}{(\eta_m^{\text{exp}})^2}, \tag{3.67}$$

pour tenir compte de la différence des ordres de grandeur. La fonction objectif est écrite sous la forme d'une fonction composée :

$$\mathcal{J}(\Omega, \eta_m) = \mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p), \tag{3.68}$$

On ajoute des poids ω_1 pour Ω et ω_2 pour η, afin de donner plus d'importance à l'un ou l'autre des paramètres. La fonction coût s'écrit alors sous la forme :

$$\mathcal{J}(\Omega, \eta_m) = \omega_1 \frac{(\Omega - \Omega^{\text{données}})^2}{(\Omega^{\text{données}})^2} + \omega_2 \frac{(\eta_m - \eta_m^{\text{données}})^2}{(\eta_m^{\text{données}})^2}. \tag{3.69}$$

Le problème de minimisation s'écrit :

$$\textit{Trouver} \quad p^* = (E_R, E_I)^* \in \mathcal{P} \quad \text{tel que} \quad \min_{p \in \mathcal{P}} \mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p) = \mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p^*), \tag{3.70}$$

où $\mathcal{P}$ est l'ensemble des paramètres viscoélastiques admissibles. Comme les fonctions ($\mathcal{J}$, $\mathcal{R}$ et $\mathcal{E}$) sont dérivables par rapport à leurs variables, le problème (3.70) peut être résolu par une méthode de gradient en cherchant une solution p^* qui annule le gradient de la fonction objectif :

$$\text{c'est à dire trouver } p^* \text{ tel que } \nabla(\mathcal{J} \circ \mathcal{R} \circ \mathcal{E})(p^*) = 0. \tag{3.71}$$

3.5.2 Calcul du gradient

Il existe plusieurs méthodes pour l'évaluation du gradient [van Keulen *et al.*, 2005]. Soit $(\delta_p \in \mathcal{P})$ une direction de perturbation. On peut approcher $\delta_J = \nabla(\mathcal{J} \circ \mathcal{R} \circ \mathcal{E})(p^0).\delta_p$ au point p^0 en utilisant un schéma de différences finis de premier ordre :

$$\delta_J = \nabla(\mathcal{J} \circ \mathcal{R} \circ \mathcal{E})(p^0).\delta_p = \frac{\mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p^0 + \varepsilon\delta_p) - \mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p^0)}{\varepsilon ||\delta_p||}, \tag{3.72}$$

où ε est un petit paramètre scalaire usuellement choisi égal à 10^{-6}.

On peut aussi différentier directement l'équation mathématique. Appliquer à l'équa-

tion (3.71) on obtient le produit de Jacobiens à implémenter :

$$\Big[\nabla\mathcal{J}(\Omega^0,\eta_m^0)\Big].\Big[\nabla\mathcal{R}(\mathcal{E}(p^0)\Big].\Big[\mathcal{E}(p^0)\Big].\delta_p=\delta_J, \tag{3.73}$$

Comme avant, le calcul du gradient $\nabla(\mathcal{J}\circ\mathcal{R}\circ\mathcal{E})$ peut être effectué en choisissant les directions de la perturbation δ_p dans la base canonique de $\mathcal{P}$.

Finalement, la technique de la Différentiation Automatique (DA) [Corliss *et al.*, 2001, Griewank et Walther, 2008] peut être appliquée au code de Daya et Potier-Ferry [2001]. Comme on l'a déjà vu au chapitre 1, en DA, tout code informatique peut être considéré comme une séquence d'opérations élémentaires, instructions de contrôle et de boucles. La différentions est alors effectuée d'une façon automatique. L'utilisation de la DA nous permet d'obtenir des dérivées exactes à la précision machine près. Cette approche générique constitue une solution fiable lorsqu'on manipule de grands codes. Elle est généralement utilisée pour les analyses de sensibilité ou des évaluations du Jacobien dans une direction particulière. Elle est proposée par tous les outils de DA existants `http://www.autodiff.org/`.

3.5.3 Différentiation Automatique

D'un point de vue informatique, une différentiation linéaire tangente par rapport aux paramètres de la loi viscoélastique est appliquée à la fois au code [Daya et Potier-Ferry, 2001] et à la fonction objectif.

Parmi les outils de DA existants, on a choisi TAPENADE ((cf. § 1.2.1)). Le mode tangent de TAPENADE permet d'obtenir un code linéaire tangent qui assure la propagation d'une perturbation suivant une direction.
En pratique, l'utilisateur fournit à l'outil : le code source, le nom de la routine principale à différentier, l'ensemble des *variables indépendantes* par rapport auxquelles on souhaite différentier le code et le mode de différentiation (Fig. 3.11).

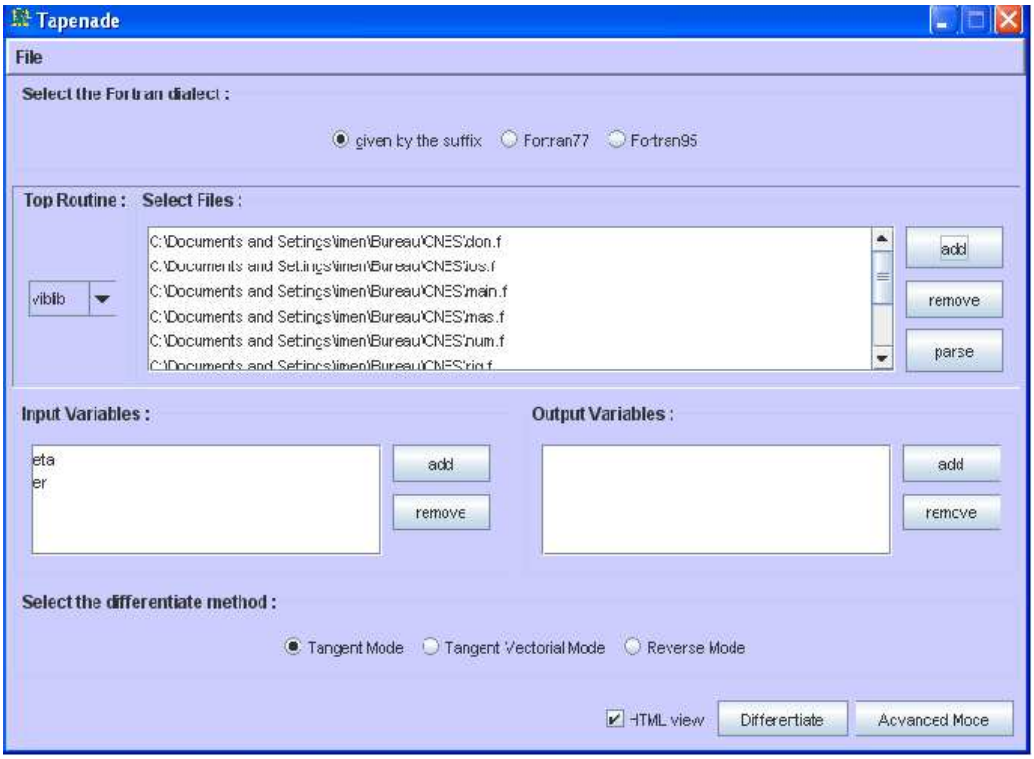

FIG. 3.11 – Chargement du code dans Tapenade

A partir de ces entrées, Tapenade génère les instructions d'un code différentié (Fig. 3.12). Dans notre cas, la routine principale est appelée *"viblib"* contenant la fonction coût notée dans nos explications J et les paramètres "E_R" et "η" notés p comme arguments. On différentie en mode linéaire tangent. Le code résultant est alors noté J_d (suivant la convention de TAPENADE), qui évalue à la fois $\mathcal{J}(p)$ et sa dérivée partielle par rapport à p dans la direction δ_p en appelant les routines différentiées du code (Chap. 1). La différentiation du code entier a prit 16 secondes en temps de calcul pour un ordinateur à processeur Intel Core2 Duo de 2GHz de fréquence et 2Go de mémoire.

La qualité du code différentié peut être améliorée si on tient en compte les particularités du notre code. En effet, on a pas besoin de différentier un solveur linéaire.
Soit

$$Ay = z \tag{3.74}$$

le système linéaire à résoudre. La différentiation de ce système par rapport au vecteur z et la matrice A donne :

$$A'y + Ay' = z'. \tag{3.75}$$

Le vecteur linéaire tangent y' est calculé alors en résolvant successivement (3.76) et (3.74) :

$$Ay' = z' - A'y. \tag{3.76}$$

avec le même solveur numérique.

D'autre part, le problème à valeurs propres réelles (lorsque $a = 0$ dans l'équation 3.54) dépend de l'élasticité retardée E_0 est indépendant des paramètres viscoélastiques (E_R, E_I). Par conséquent, la résolution de (3.53) n'a pas à être différentiée et n'est effectuée qu'une seule fois.

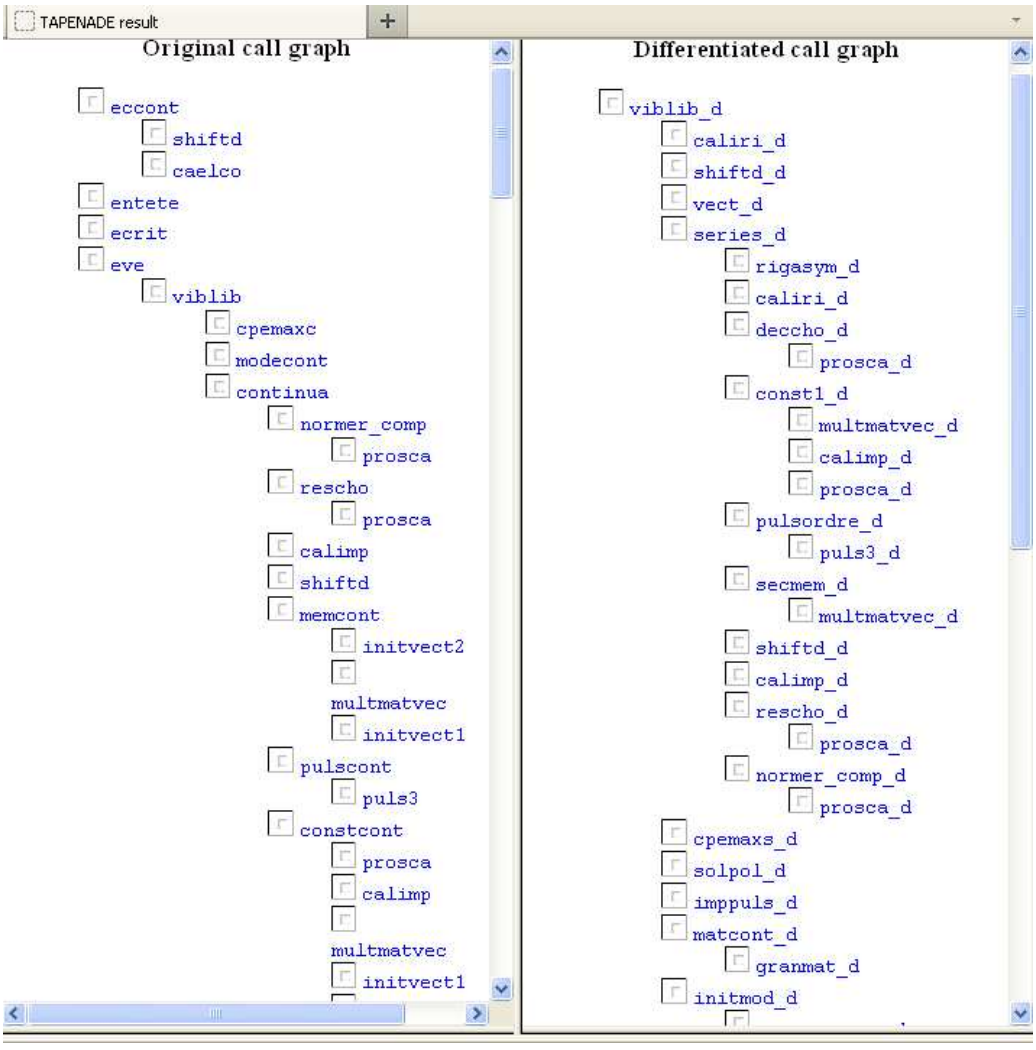

FIG. 3.12 – Différentiation du code par Tapenade

Test de validation du code différentié On vérifie le code différentié généré par TAPENADE par un test de Taylor (cf. § 1.2.3). Généralement, on vérifie les dérivées généré par

l'outil de DA pour chaque direction de perturbation δ_p de la variable x en calculant le ratio r_p :

$$r_p(\boldsymbol{\varepsilon}) = \frac{|\mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p + \boldsymbol{\varepsilon}\delta_p) - \mathcal{J} \circ \mathcal{R} \circ \mathcal{E}(p)|}{\boldsymbol{\varepsilon}\,|\nabla(\mathcal{J} \circ \mathcal{R} \circ \mathcal{E})(p).\delta_p|}. \tag{3.77}$$

Le tableau 3.6 présente le test de Taylor correspondant aux perturbations appliquées successivement aux paramètres η et E_R.

α	r_η	r_{E_R}
10^{5}		0.736476767
10^{4}		0.965834182
10^{3}		0.996480631
10^{2}		0.999646996
10^{1}	0.239671976	0.999965001
10^{0}	0.869771111	0.999995842
10^{-1}	0.991675435	0.999995814
10^{-2}	0.999294265	1.00010369
10^{-3}	0.99993093	1.00055649
10^{-4}	0.999993127	0.997910141
10^{-5}	0.999999624	1.07470489
10^{-6}	1.00000307	0.575609422
10^{-7}	1.00002637	17.0576939
10^{-8}	1.00038531	
10^{-9}	1.00290659	
10^{-10}	1.01872331	

TAB. 3.6 – Test de Taylor effectué dans les directions η et E_R.

On observe que r_η et r_{E_R} tendent linéairement vers 1 jusqu'à des valeurs de troncature optimale pour le calcul par différences finis, correspondant respectivement à $\boldsymbol{\varepsilon} = 10^{-5}$ pour η et $\boldsymbol{\varepsilon} = 1$ pour E_R. Puis, lorsque les erreurs d'arrondi dominent l'erreur de troncature, on s'éloigne de 1. Un test de Taylor ayant un tel comportement nous permet de conclure que le code différentié est correct.

3.5.4 Pilote d'optimisation

Le pilote d'optimisation pour l'identification de la loi viscoélastique en utilisant le code [Daya et Potier-Ferry, 2001] peut ainsi être implémenté comme expliquer par le tableau 3.7

TAB. 3.7 – Processus d'optimisation pour un mode donné.

Méthode des sous-espaces
Pilote d'optimisation
→ nouvel itéré p
→ Si ($J >$ un seuil)
→ Continuation_d → $\mathcal{E}$_d → séries_d pour $\mathcal{E}$_d → solveur linéaire } calcul du gradient
→ J_d
→ Si non : solution trouvée

Les itérations sont calculées à l'aide de l'optimiseur L-BFGS-B [Byrd *et al.*, 1995] basé sur une méthode de quasi-Newton. Le critère d'arrêt du processus d'optimisation est la différence relative entre deux itérations successives :

$$|J(p) - J(p-1)| \leq 10^{-3}. \tag{3.78}$$

3.6 Résultats de l'identification

Les paramètres de la loi viscoélastiques sont identifiés au moyen de l'optimiseur L-BFGS-B [Byrd *et al.*, 1995] (§ 1.3.6). Les résultats de l'identification numériques sont proposés pour la poutre sandwich à cœur viscoélastique étudiée dans [Daya et Potier-Ferry, 2001].

Les résultats de l'identification sont présentés dans le tableau 3.8. On remarque qu'on a réussi à identifier l'amortissement avec une erreur inférieur à 2.10^{-3} ce qui est satisfaisant. Pour la fréquence de résonance, on a une erreur en générale de l'ordre de 10^{-2}. Néanmoins, pour le premier mode, on a une erreur de l'ordre de 10^{-1}. Cela peut être expliqué par le fait que, pour de faibles fréquences ($f < 50Hz$), Landier [1993] a remarqué que la cohérence du signal est inférieur à 1. D'autres erreurs peuvent être liées à des imprécisions sur les épaisseurs des différentes couches, le module de Young de l'acier et la modification du module complexe du polymère due à la mise en sandwich ($\pm 10\%$) [Landier, 1993].

TAB. 3.8 – Fréquences et facteurs de perte. A : Données $(\Omega^{exp}, \eta_m^{exp})$, B : Valeurs identifiées (Ω^*, η_m^*), C : Erreurs relatives. Poids $w_1 = 10^{-6}$ et $w_2 = 1$

	A		B		C	
	Ω^{exp}	η_m^{exp}	Ω^*	η_m^*	$\frac{\Omega^* - \Omega^{exp}}{\Omega^{exp}}$	$\frac{\eta_m^* - \eta_m^{exp}}{\eta_m^{exp}}$
20°C	38	5.22×10^{-2}	32.40	5.22×10^{-2}	1.47×10^{-1}	1.03×10^{-5}
	188	1.09×10^{-1}	186.68	1.09×10^{-1}	6.87×10^{-3}	1.59×10^{-3}
	519	1.44×10^{-1}	558.60	1.44×10^{-1}	7.63×10^{-2}	1.94×10^{-4}
	1030	1.43×10^{-1}	1030.00	1.43×10^{-1}	6.02×10^{-7}	7.35×10^{-5}
30°C	37	5.64×10^{-2}	30.64	5.64×10^{-2}	1.72×10^{-1}	3.40×10^{-5}
	167	1.54×10^{-1}	169.99	1.54×10^{-1}	1.79×10^{-2}	8.63×10^{-4}
40°C	36	3.66×10^{-2}	30.87	3.66×10^{-2}	1.42×10^{-1}	1.02×10^{-3}
	155	1.05×10^{-1}	154.63	1.05×10^{-1}	2.41×10^{-3}	6.67×10^{-8}
	379	2.02×10^{-1}	388.11	2.02×10^{-1}	2.40×10^{-2}	1.46×10^{-5}
50°C	35	3.13×10^{-2}	30.54	3.13×10^{-2}	1.28×10^{-1}	4.80×10^{-5}
	150	6.54×10^{-2}	154.65	6.54×10^{-2}	3.10×10^{-2}	1.67×10^{-4}
	365	9.22×10^{-2}	391.64	9.22×10^{-2}	7.30×10^{-2}	3.17×10^{-4}
	649	1.15×10^{-1}	653.31	1.15×10^{-1}	6.65×10^{-3}	9.42×10^{-5}
60°C	34	2.75×10^{-2}	30.34	2.75×10^{-2}	1.08×10^{-1}	7.93×10^{-5}
	147	5.03×10^{-2}	153.81	4.98×10^{-2}	4.63×10^{-2}	9.25×10^{-3}
	357	6.68×10^{-2}	382.75	6.68×10^{-2}	7.21×10^{-2}	4.72×10^{-5}
	636	6.77×10^{-2}	636.12	6.77×10^{-2}	1.86×10^{-4}	2.22×10^{-6}
70°C	34	2.82×10^{-2}	30.20	2.82×10^{-2}	1.12×10^{-1}	1.63×10^{-3}
	145	4.45×10^{-2}	147.83	4.45×10^{-2}	1.95×10^{-2}	5.66×10^{-4}
	352	5.52×10^{-2}	368.12	5.52×10^{-2}	4.58×10^{-2}	1.79×10^{-5}
	627	5.27×10^{-2}	634.47	5.26×10^{-2}	1.19×10^{-2}	1.05×10^{-3}

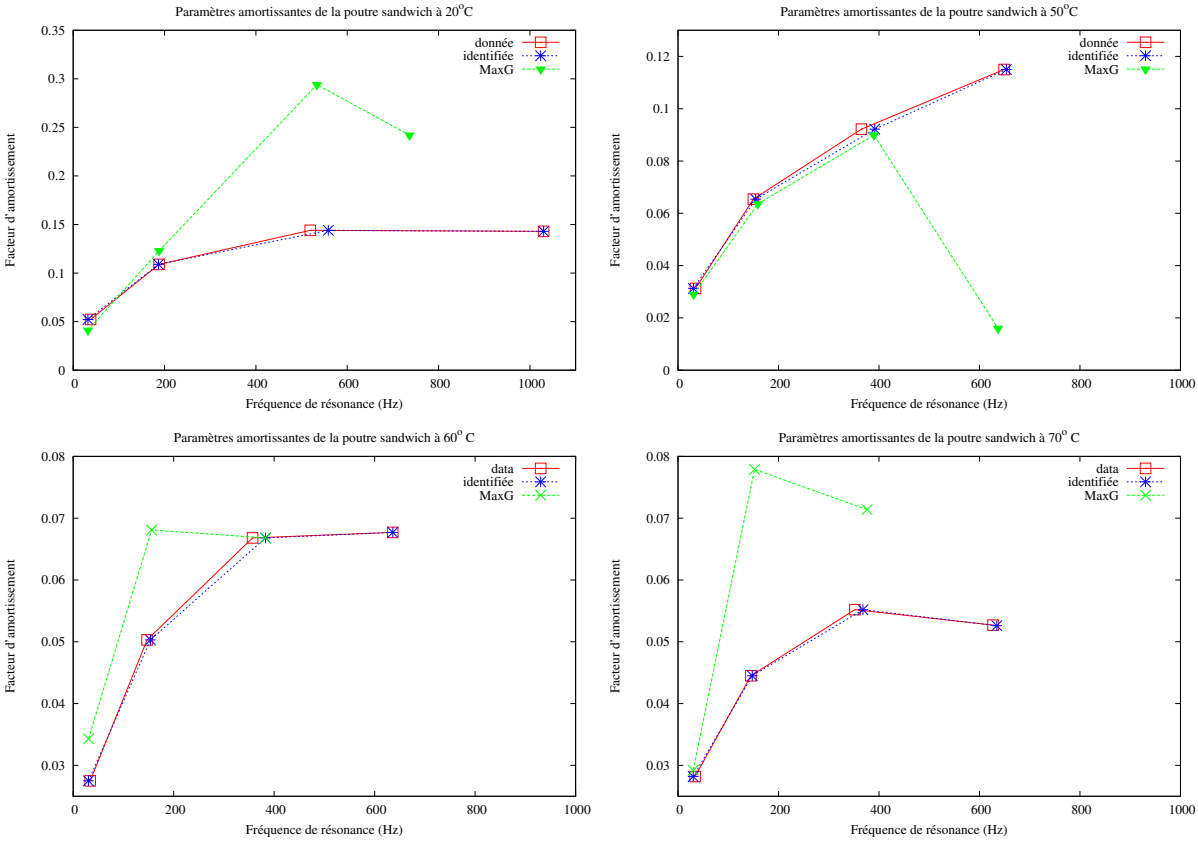

FIG. 3.13 – Variation du facteur d'amortissement en fonction de la fréquence de résonance

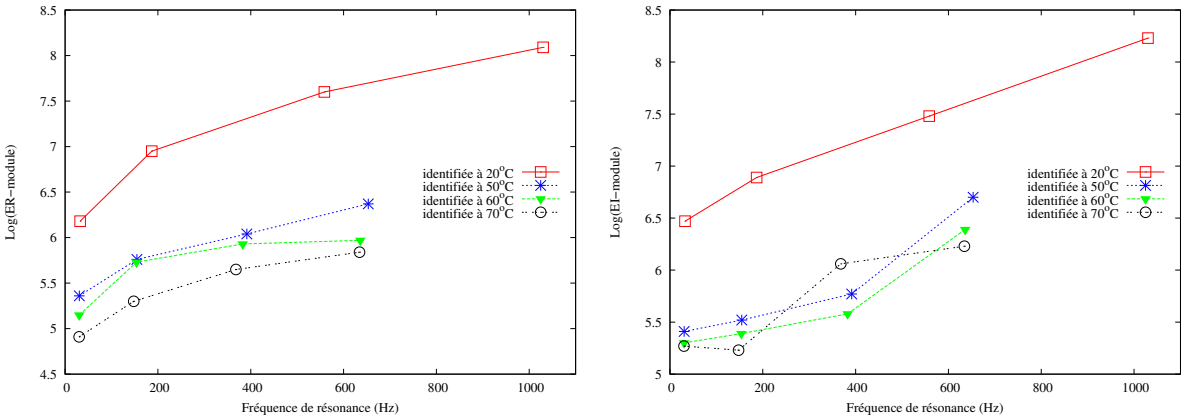

FIG. 3.14 – Identification des paramètres ER et EI à 20°C, 50°C, 60°C et 70°C

3.7 Conclusion

On a proposé dans ce chapitre, une identification de la loi de comportement de Maxwell généralisé implémentée dans le code éléments finis [Daya et Potier-Ferry, 2001]

qui modélise la vibration libre d'une structure sandwich (acier/ploymère/acier) de type plaque ou poutre. Les paramètres de la loi de comportement sont fournis par l'industriel USINOR pour le polymère seul. Les différences observées entre la simulation et les essais effectués par [Landier, 1993] ont motivé le choix d'une méthode inverse pour identifier la loi de comportement VE. La fonction objectif est choisie comme étant la différence entre la simulation et les observations. Comme le modèle est différentiable par rapport aux paramètres de la loi VE, le minimum peut être obtenu en cherchant une solution pour laquelle le gradient de cette fonction est nul. Pour obtenir le gradient de la fonction objectif, on différentie le code éléments finis par rapport aux paramètres à identifier par l'outil de DA Tapenade [Hascoët *et al.*, 2005a] en mode linéaire tangent. Le problème de minimisation est résolu par l'algorithme d'optimisation LBFGS [Byrd *et al.*, 1995]. Les résultats obtenus nous permettent de valider la méthode proposée pour l'identification de la loi de comportement de la tôle sandwich à cœur viscoélastique entière.

Conclusion

Dans cette thèse, on s'intéresse à l'application de la technique de Différentiation Automatique pour l'identification des lois de comportement non-linéaires indispensables pour la prédiction du comportement des tôles en grandes déformations. Mes travaux ont pour but de développer une stratégie d'identification modèle-expérience en utilisant des essais expérimentaux, la simulation par éléments finis, une méthode de gradient et la technique de Différentiation Automatique. On a traité dans le cadre de cette thèse, deux problèmes d'identification.

Le premier chapitre décrit la technique de différentiation automatique de codes complets et des méthodes de gradient pour l'identification. Le principal avantage d'une utilisation de la DA est le développement très rapide de codes linéaires tangents (ou adjoints) pour le calcul des gradients exacts (au sens du code) nécessaires pour résoudre le problème inverse. La DA offre une aide précieuse en automatisant le calcul des dérivées des fonctions présentées par des codes informatiques. La DA peut être utile dans toute méthode utilisant le calcul des dérivées. Charpentier [2011] a notamment montré l'efficacité de la DA pour automatiser la MAN (Méthode Asymptotique Numérique)[Cochelin *et al.*, 2007]. Cependant, il faut noter que contrairement à d'autres outils d'estimation des dérivées, il faut disposer du code source des fonctions à différentier. Dans cette thèse, on génère automatiquement les codes linéaires tangents pour deux codes mécaniques (FLagSHyP et [Daya et Potier-Ferry, 2001]) pour identifier deux lois de comportement.

Dans le (Chap. 2), le problème consiste à identifier la loi d'écrouissage des tôles métalliques à partir d'un essai de traction uni-axiale. Pour cela, des essais de traction uni-axiale ont été réalisés aux laboratoires d'ArcelorMittal pour deux type d'acier(un acier multiphasé et un acier doux). Un code libre de simulation par éléments finis écrit en FORTRAN a été utilisé pour simuler ces essais. Une loi d'écrouissage empirique à 6 paramètres, utilisée par les industriels afin d'offrir une certaine flexibilité aux extrapolations en grandes déformations, ainsi qu'une fonction coût calculant la différence entre les simu-

lations et les essais ont été implémentés dans le code. Ensuite, le gradient de la fonction coût a été calculé automatiquement par le logiciel de DA par surcharge d'opérateur TAPENADE [Hascoët *et al.*, 2005a]. Le gradient calculé automatiquement en quelques secondes est fourni à l'optimiseur L-BFGS-B [Byrd *et al.*, 1995] interfacé avec le code éléments finis afin de résoudre le problème de minimisation posé. Le potentiel de la méthode est prouvé. Cependant, on a été confronté à plusieurs difficultés. Le code utilisé est académique et ne permet pas de travailler avec un maillage fin (temps de calcul très élevé). Pour essayer de comprendre les différences observées entre les résultats expérimentaux et les simulations au niveau de la chute de la force (à la rupture), on a réalisé des simulations avec un maillage fin avec Abaqus. Les simulations effectuées avec la même loi de comportement reproduisent la chute de force, associée à des champs de déformation mettant en évidence la striction diffuse, puis la striction localisée. Néanmoins la chute de force est, selon l'acier considéré, trop précoce ou un peu trop lente par rapport à l'expérimental. Cet écart peut être attribué, respectivement, à la non prise en compte de la sensibilité à la vitesse de déformation dans la loi de comportement ou à l'endommagement. Pour améliorer la qualité de nos résultats, il est nécessaire d'utiliser un code source plus performant. En second lieu, on propose d'implémenter une loi de comportement tenant compte d'un plus grand nombre de phénomènes physiques menant à la rupture à savoir l'écrouissage, la sensibilité à la vitesse et l'endommagement. Enfin, il serait utile d'enrichir la base des données utilisées pour l'identification en déterminant expérimentalement les champs cinématiques de déplacement et de déformation grâce à des techniques d'analyse d'images numériques.

Dans le (Chap. 3), le problème consiste à identifier la loi de comportement d'une poutre sandwich acier/polymère/acier en vibrations libres à partir de fréquences de résonance et de facteurs d'amortissement expérimentaux. La loi de comportement de Maxwell généralisée est implémentée dans un code éléments finis écrit en FORTRAN dans le cadre des travaux de Daya et Potier-Ferry [2001]. Les paramètres de la loi de comportement ont été fournis par l'industriel USINOR pour le polymère seul. Les différences observées entre les simulations et les essais réalisés par Landier [1993] ont motivé cette étude. La loi de comportement a été identifiée par la stratégie proposée dans le (Chap. 2). La fonction coût implémentée dans le code mesure la différence entre les simulations et les observations. Le code est différentié automatiquement par TAPENADE [Hascoët *et al.*, 2005a]. Des modifications ont été apportée afin d'éviter la différentiation du solveur linéaire. La résolution du problème de minimisation est effectuée par l'optimiseur L-BFGS-B [Byrd *et al.*, 1995]. La méthode d'optimisation mise en place à permis d'identifier la loi de comportement de la poutre sandwich à cœur viscoélastique. On a trouvé une très bonne concordance avec les essais pour l'identification de l'amortissement (erreur inférieure à 2.10^{-3}) et une erreur

inférieur à 10^{-2} pour la fréquence de résonance excepté pour le premier mode. Pour celui-ci, on a eu une erreur de l'ordre de 10^{-1} pour la fréquence de résonance. Cette différence peut être expliquée par les erreurs expérimentales liées aux faibles fréquences. Des essais plus sophistiquées permettraient d'améliorer la qualité des résultats.

Bibliographie

Abdoun, F., Azrar, L. et DAya, E. (2010). Damping and forced vibration analyses of viscoelastic shells. *International Journal for Computational Methods in Engineering Science and Mechanics*, 11:109–122.

Abdoun, F., Azrar, L., Daya, E. et Potier-Ferry, M. (2009). Forced vibrations of sandwich viscoelastic beams and plates by an asymptotic numerical method. *Computers and Structures*, 87:91–100.

Appel, W. (2006). *Mathématiques pour la physique et les physiciens*. H&K Editions.

Araújo, A., Soares, C. M., Herskovits, J. et Pedersen, P. (2002). Development of a finite element model for the identification of mechanical and piezoelectric properties through gradient optimisation and experimental vibration data. *Composite Structures*, 58:307–318.

Araújo, A., Soares, C. M., Herskovits, J. et Pedersen, P. (2009). Estimation of piezoelectric and viscoelastic properties in laminated structures. *Composite Structures*, 87:168–174.

Aubert, P., Di Césaré, N. et Pironneau, O. (2001). Automatic differentiation in C++ using expression templates and application to a flow control problem. *Computing and Visualization in Science*, 3:197–208.

Barkanov, E., Skukis, E. et Petitjean, B. (2009). Characterization of viscoelastic layers in sandwich panels via an inverse method. *Journal of Sound and Vibration*, 327:402–412.

Bathe, K. (1982). *Finite Element Procedures in Engineering Analysis*. Prentice-Hall.

Baz, A. et Ro, J. (1995). Optimum design and control of active constrained layer damping. *Journal of Mechanical Engineering Design*, 117:135–144.

Bendtsen, C. et Stauning, O. (1996). FADBAD, a flexible C++ package for automatic differentiation. Technical Report IMM–REP–1996–17, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Benson, S. et Moré, J. J. (2001). A limited memory variable metric method for bound constrained minimization. Rapport technique, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois.

Bert, C. (1973). Material damping : An introductory review of mathematical models, measures, and experimental techniques. *Journal of Sound and Vibration*, 29(2):129–153.

Betcke, T. et Voss, H. (2002). A Jacobi-Davidson-type projection method for nonlinear eigenvalue problems. Rapport technique, Technical University of Hamburg, Department of Mathematics. http://doku.b.tu-harburg.de/volltexte/2006/220/.

Bilasse, M. (2010). *Modélisation numérique des vibrations linéaires et non linéaires des structures sandwichs à âme viscoélastique.* Thèse de doctorat, Université Paul Verlaine de Metz.

Bilasse, M., Azrar, L. et Daya, E. (2011). Complex modes based numerical analysis of viscoelastic sandwich plates vibrations. *Computers and Structures*, 89(7–8):539–555.

Bischof, C. H., Carle, A., Khademi, P. M., Mauer, A. et Hovland, P. (1994). ADIFOR 2.0 user's guide. Technical Memorandum ANL/MCS–TM–192, Mathematics and Computer Science Division, Argonne National Laboratory.

Bonnans, J., Gilbert, J. et Lemarechal, C. (2002). *Numerical optimization : Theoretical and practical aspects.*

Bonnet, J. et Wood, R. (2002). Numerical simulation of the superplastic forming of dental and medical prostheses. *Biomechanics and Modeling in Mechanobiology*, 1:177 – 196.

Bonnet, J. et Wood, R. (Janvier 2008). *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press.

Brezinski, C. et Redivo-Zaglia, M. (2006). *Méthodes numériques itératives, 1ère édition.* Ellipses.

Byrd, R., Lu, P., Nocedal, J. et Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208.

Castello, D., Rochinha, F., Roitman, N. et Magluta, C. (2008). Constitutive parameter estimation of a viscoelastic model with internal variables. *Mechanical Systems and Signal Processing*, 22:1840–1857.

Charpentier, I. (2008). Sensitivity of solutions computed through the asymptotic numerical method. *Comptes Rendus Mecanique*, 336:788–793.

Charpentier, I. (2011). On higher-order differentiation in nonlinear mechanics. *Optimization Methods and Software*, 0(0):1–12.

Charpentier, I. et Utke, J. (2009,). Fast higher-order derivative tensors with Rapsodia. *Optimization Methods Software,*, 24,:1–14.

Chen, X., Chen, H. et Le, H. (1999). Damping prediction of sandwich structures by order-reduction-iterative approach. *Journal Sound and Vibration*, 222 (5):803–812.

Cochelin, B., Damil, N. et Potier-Ferry, M. (2007). *Méthode Asymptotique Numérique : une technique de résolution des équations non linéaires.* Hermès - Lavoisier.

Corliss, G., Faure, C., Griewank, A., Hascoët, L. et Naumann, U. (2001). *Automatic Differentiation of algorithms : From simulation to optimization*. Springer.

Daya, E., Azrar, L. et Potier-Ferry, M. (2002). Modélisation des vibrations nonlinéaires des structures sandwich viscoélastiques. *In Septièmes Journées d'Analyse Numérique et d'Optimisation, Tanger-Maroc.*

Daya, E. et Potier-Ferry, M. (2001). A numerical method for nonlinear eigenvalue problems application to vibrations of viscoelastic structures. *Computer and structures*, 79:533–541.

Daya, E. et Potier-Ferry, M. (2002). A shell finite element for viscoelastically damped sandwich structures. *Revue européenne des éléments finis,*, 11:39–56.

Diot, S., Gavrus, A., Guines, D. et Ragneau., E. (2003.). Identification du comportement d'un acier en compression : du quasi-statique au dynamique : Identification of a steel compression behaviour : from quasi static approach to dynamic one. *Mécanique et Industries,*, 4(5):519–524.

DiTaranto, R. et Blasingame, W. (1966). Composite loss factors of selected laminated beams. *The Journal of the Acoustical Society of America*, 40:187–194.

Dournaux, J., Bouvier, S., Aouafi, A. et Vacher, P. (2009). Full-field measurement technique and its application to the analysis of materials behavior under plane-strain mode. *Mat. Sci. Engng*, A500:47–62.

Duigou, L., Daya, E. et Potier-Ferry, M. (2003). Iterative algorithms for non-linear eigenvalue problems. application to vibrations of viscoelastic shells. *Computer Methods in Applied Mechanics and Engineering*, 192:1323–1335.

Duncan, J. (1965). *Bull. Mech. Engng Educ.*, 4:29–37.

Fried, J. (2003). *Polymer Science and Technology*. Prentice-Hall.

Gavrus, A., Massoni, E. et Chenot, J. L. (1996). An inverse analysis using a finite element model for identification of rheological parameters. *Journal of Materials Processing Technology*, 60:447–454.

Ghouati, O. et Gelin, J. (1998). Identification of material parameters directly from metal forming processes. *Journal of Materials Processing Technology*, 80-81:560–564.

Gilbert, J.-C. et Lemaréchal, C. (1995). The modules M1QN3 and N1QN3. version 2.0c. Rapport technique, INRIA.

Givoli, D. et Patlashenko, I. (2000). Solution of static optimal control problems in non-linear elasticity via quadratic programming. *Communications In Numerical Methods In Engineering*, 16:877–890.

Golla, D. et Hughes, P. (1985). Dynamics of viscoelastic structures : a time domain finite element formulation. *Journal of Applied Mechanics*, 52:897–906.

Griewank, A., Juedes, D. et Utke, J. (1996). Algorithm 755 : Adol-c : A package for the automatic differentiation of algorithms written in c/c++. *ACM Transactions on Mathematical Software*, 22(2):131–167.

Griewank, A. et Walther, A. (2008). *Evaluating derivatives : Principles and techniques of algorithmic differentiation*. Other Titles in Applied Mathematics. SIAM, Philadelphia, PA, 2nd édition.

Hascoët, L., Greborio, R. et Pascual, V. (2005a). *Computing adjoints by Automatic Differentiation with TAPENADE*. Springer.

Hascoët, L., Nauman, U. et Pascual, V. (October 2005b). "To be recorded" analysis in reverse-mode automatic differentiation. *Future Generation Computer Systems*, 21: 1401–1417.

Hendricks., M. (1991). *Identification of the mechanical behavior of solid materials.* Thèse de doctorat, Eindhoven University of Technology.

Herskovits, J. (1998). Feasible direction interior-point technique for nonlinear optimization. *Journal of Optimization Theory and Applications*, 99:121–146.

Hill, R. (1948). A theory of the yielding and plastic flow of anisotropic metals. *Proceedings of the Royal Society*, pages 281–297.

Hill, R. (1950). *The Mathematical Theory of Plasticity.* Oxford Classic.

Johnson, C. et Kienholz, D. (1982). Finite element prediction of damping in structures with constrained viscoelastic layers. *American Institute of Aeronautics and Astronautics Journal*, 20(9):1284–1290.

Kajberga, J., Sundina, K., Melinb, L. et Stahlea, P. (2004). High strain-rate tensile testing and viscoplastic parameter identification using microscopic high-speed photography. *International Journal of Plasticity,*, 20(4–5):561–575.

Karafilis, A. et Boyce., M. (1993). A general anisotropic yield criterion using bounds and a trasformation weighting tensor. *Journal of the Mechanics and Physics of Solids*, 41:1859–1886.

Kim, S. et Lee, D. (2009). Identification of fractional-derivative-model parameters of viscoelastic materials from measured frfs. *Journal of Sound and Vibration*, 324:570–586.

Kobayashi, A. S. (1993). *Handbook on Experimental Mechanics.* Society for Experimental Mechanics, 2nd rev. édition.

Landier, J. (1993). *Modélisation et étude expérimentale des proriétés amorissantes des tôles sandwich.* Thèse de doctorat, Université de Metz.

Lemaitre, J. (1986). Local approach of fracture. *Engineering Fracture Mechanics*, 25: 523–53.

Lemoine, X., Iancu, A. et Ferron, G. (2011). Flow curve determination at large plastic strain levels : Limitations of the membrane theory in the analysis of the hydraulic bulge test. *AIP Conference Proceedings*, 1353(1):1411–1416.

Lesieutre, G. et Bianchini, E. (1995). Time domain modeling of linear viscoelasticity using augmenting thermodynamic fields. *Journal of Vibration and Acoustics*, 117:424–430.

Lin, C. et Moré, J. (2006). Newton's method for large bound-constrained optimization problems. *SIAM J. Optim*, 9:1100–1127.

Ma, B.-A. et He, J.-F. (1992). A finite element analysis of viscoelastically damped sandwich plates. *Journal of Sound and Vibration*, 152(1):107–123.

Mahnken, R. et Stein, E. (1996). A unified approach for parameter identification of inelastic material models in the frame of the finite element method. *Computer Methods in Applied Mechanics and Engineering,*, 136:225–258.

Mahnken, R. et Stein, E. (1997). Parameter identification for finite deformation elasto-plasticity in principal directions. *Computer Methods in Applied Mechanics and Engineering*, 147(1-2):17–39.

Matter, M., Gmür, T., Cugnoni, J. et Schorderet, A. (2009). Numerical-experimental identification of the elastic and damping properties in composite plates. *Composite Structures*, 90:180–187.

Meurant, G. (1999). Computer solution of large linear systems. *Studies in Mathematics and its Applications*, 28.

Meuwissen, M., Oomens, C., Baaijens, F., Petterson, R. et Janssen., J. (1998). Determination of the elasto-plastic properties of aluminium using a mixed numerical-experimental method. *Journal of Materials Processing Technology,*, 75:204–211.

Nashif, A., Jones, D. et Henderson, J. (1985). *Vibration Damping*. John Wiley and Sons.

N.Bartoli et Moral, P. D. (2001). *Simulation et algorithmes stochastiques : une introduction avec applications*. Toulouse : Cépaduès.

Persoz, B. (1969). *La rhéologie*. Masson and Cie.

PRAGER, W. (1955). The theory of plasticity : a survey of recent achievements. *Proceedings of the Institution of Mechanical Engineers, London*, 169:41–57.

RAUCH, E. (1998). Plastic anisotropy of sheet metals determined by simple shear tests. *Mat. Sci. Engng*, A241:179–183.

RICE, J. et TRACEY, D. (1969). On the ductile enlargement of voids in triaxial stress fields. *Journal of the Mechanics and Physics of Solids*, 17:201–217.

RIKARDS, R., CHATE, A. et BARKANOV, E. (1993). Finite element analysis of damping the vibrations of laminated composites. *Computers & structures*, 47:1005–1015.

ROCHA, M. (2008). *Un modelo de acoplamento eletromecânico do tecido cardiaco através do método dos elementos finitos*. Thèse de doctorat, Universidade Federal de Juiz de Fora.

SALENÇON, J. (2009). *Viscoélasticité pour le calcul des structures*. Les Éditions de l'École polytechnique , Presses de l'école nationale des Ponts et Chaussées (ENPC).

SANTOS, A. D., TEIXEIRA, P., da ROCHA, A. B. et BARLAT, F. (2010). On the determination of flow stress using bulge test and mechanical measurement. *AIP Conference Proceedings*, 1252(1):845–852.

SCHNUR, D. S. et ZABARAS, N. (1992). An inverse method for determining elastic material properties and a material interface. *International Journal for Numerical Methods in Engineering*, 33:2039–2057.

SEMIATIN, S. et JONAS, J. (1984.). *Formability and workability of metals : Plastic instability and flow localization.* American Society for Metals (Metals Park, Ohio),.

SONI, M. (1981). Finite element analysis of viscoelastically damped sandwich structures. *Shock and vibration bulletin*, 55:97–109.

SUN, C. et LU, Y. (1995). *Vibration Damping of Structural Elements*. Prentice-Hall PTR.

SURREL, Y. (2005). Les techniques optiques de mesure de champ : essai de classification. *Instrumentation,* M*esure,* M*étrologie*, 4(3–4):11–42.

TAYLOR, G. et QUINNEY, H. (1931). *Philosophical Transactions of the Royal Society.*, 230:323.

THOMPSON, J. et WALKER, A. (1968). The nonlinear perturbation analysis of discrete structural systems. *Int. J. Solids & Structures*, 4:757–768.

van Keulen, F., Hafka, R. et Kim, N. (2005). Review of options for structural design sensitivity analysis. part 1 : Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194:3213–3243.

Von-Mises (1913). Mechanik der festen korper in plastich-deformablen zustand, nachr konigl ges wiss gottingen. *Math Phys*, pages 582–592.

Wagoner, R. H. (1981). Comparison of plane-strain and tensile work hardening in two sheet steel alloys. *Metallurgical and Materials Transactions.*, 12:877–882.

Wengert, R. (1964). A simple automatic derivative evaluation program. *Communications of the ACM*, 7:463–464.

Williams, M., Landel, R. et Ferry, J. (1955). The temperature dependance of relaxation mechanisms in amorphous polymers and other glass forming liquids. *Journal of American Chemical Society*, 77:3701–3707.

Woodford, D. (1969). *Transactions of the American Society for Metals*, 62:291.

Yoshida, F., Urabe, M. et Toropov, V. (1998). Identification of material parameters in constitutive model for sheet metals from cyclic bending tests. *International Journal of Mechanical Science.*, 40(2-3):249–327.

Yoshida, F., Urabeb, M., Hinoa, R. et Toropov., V. V. (2003). Inverse approach to identification of material parameters of cyclic elasto plasticity for component layers of a bimetallic sheet. *International Journal of Plasticity*, 19:2149–2170.

Zienkiewicz, O. et Taylor, R. (2005). *The Finite Element Method Set*. A Butterworth-Heinemann Title, 6 édition.

Résumé

La méthode des éléments finis permet de déterminer les états de contrainte à l'intérieur d'une structure mécanique soumise à une sollicitation. Des procédés de fabrication peuvent être modélisés. De nos jours, la réduction des temps de calcul permet d'aborder des problèmes inverses pour identifier des paramètres du matériau ou de la géométrie de la structure, optimaux pour une problématique donnée. Le problème inverse fait intervenir un modèle mécanique et une fonction objectif mesurant un écart entre la réponse observée expérimentalement et les réponses obtenues à partir de jeux de paramètres différents.

Cette thèse décrit une méthode de gradient et la technique de différentiation automatique pour résoudre deux problèmes inverses issus de la mécanique. Le premier concerne l'identification d'une loi d'écrouissage à partir d'essais de traction uni-axiale réalisés sur deux aciers différents et du code académique FLagSHyP . Le second s'intéresse à l'identification d'une loi viscoélastique pour un matériau sandwich à partir de données expérimentales mesurées sur une poutre sandwich en vibrations libres et du solveur aux valeurs propres complexes développé au LPMM. Ces deux modèles sont différentiables. Pour chaque problème, le jeu de paramètres optimal est identifié comme un jeu pour lequel le gradient de la fonction objectif s'annule. Nous utilisons la technique de différentiation automatique pour générer des codes pour le calcul de ces gradients de manière simplifiée. Les résultats obtenus démontrent l'intérêt de cette approche.

Abstract

The finite element method allows for the determination of the states of the stress in a mechanical structure under sollicitation. Manufacturing processes may be modeled. Nowadays, the reduction of computational costs allows to tackle inverse problems for the identification of some material or geometrical parameters of the structure that are optimal to a given problem. The inverse problem makes use of a mechanical model and

an objective function measuring the discrepancies between the observed response and responses resulting from differents sets of parameters.

This thesis describes a gradient method and the automatic differentiation technique for the solution of two inverse problems in mechanics. The first one is concerned with the identification of a strain-hardening law from uniaxial tension test performed on two different steels and from the academical code FLagSHyP . The second one deals with the identification of a viscoelastic law for a sandwich material from experimental data measured on a simply supported sandwich beam and a complex eigenvalue solver developed at LPMM. These two models are differentiable. For each problem, the optimal parameter set is identified as a set for which the gradient of the objective function is zero. We use the automatic differentiation technique to generate codes for the computations of these gradients in a simplified manner. Results we obtained demonstrate the interest of such an approach.