



Différentiation automatique de codes mécaniques : application à l'analyse de sensibilité des tôles sandwich aux paramètres de modélisation

Komlanvi Lampoh

► To cite this version:

Komlanvi Lampoh. Différentiation automatique de codes mécaniques : application à l'analyse de sensibilité des tôles sandwich aux paramètres de modélisation. Autre. Université de Lorraine, 2012. Français. NNT : 2012LORR0220 . tel-01749313

HAL Id: tel-01749313

<https://hal.univ-lorraine.fr/tel-01749313>

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.



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>

Différentiation Automatique (DA) de codes mécaniques : application à l'analyse de sensibilité des tôles sandwich aux paramètres de modélisation

THÈSE

présentée et soutenue publiquement le 18 Septembre 2012

pour l'obtention du grade de

Docteur de l'Université de Lorraine

Spécialité : Mécanique

par

LAMPOH Komlanvi

Composition du jury

<i>Rapporteurs :</i>	Pr. Erasmo CARRERA Pr. Bruno COCHELIN	Ecole Polytechnique de Turin, Italie LMA, Ecole Centrale de Marseille, France
<i>Examineur :</i>	MCF-HDR Jean-Marc CADOU	Université de Bretagne Sud, France
<i>Directeurs :</i>	CR-HDR Isabelle CHARPENTIER Pr. El Mostafa DAYA	CNRS (LEM3), France Université de Lorraine (LEM3), France

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je tiens à remercier ma Directrice de thèse Isabelle Charpentier, HDR au LEM3 pour m'avoir confié ce travail de recherches, pour son aide, ses précieux conseils sa disponibilité et sa rigueur scientifique. Son œil critique a permis d'améliorer considérablement la rédaction de ce mémoire. C'est un immense plaisir pour moi de travailler sous votre houlette. J'ai appris beaucoup de chose avec vous tant sur le plan numérique et scientifique.

J'adresse mon remerciements au professeur DAYA El Mostafa, codirecteur de cette thèse pour sa disponibilité, ses conseils. Ses suggestions avisées ont permis d'orienter et améliorer les aspects mécaniques de ce travail.

Mes remerciements iront au Professeurs Bruno COCHELIN et Professeur Erasmo CARRERA pour l'intérêt qu'ils ont porté à mes travaux, et pour avoir accepté d'être les rapporteurs de ma thèse.

Mes remerciements iront également à Monsieur Jean-Marc CADOU, HDR à l'université de Bretagne-sud pour le temps consacré à examiner cette thèse.

Je tiens à remercier M Koutsawa Yao et M Bilasse Massamaesso pour avoir posée les prémisses de ce travail.

Je tiens à remercier M. TIEM Sonnou pour avoir mise en place cette ouverture qui nous permet de poursuivre une formation doctorale.

J'adresse mes remerciements à toute ma famille et à toutes les personnes qui ont contribué d'une manière ou d'une autre à la réussite de cette thèse. En particulier à Madame et Monsieur OBER, à ma sœur Lucie, Olivier AZOTI, Yanick Tampango.

En fin j'adresse mes remerciements à tout le personnel de LEM3 notamment les secrétaires et mes collègues doctorants pour les services rendus et pour le cadre du travail.

*Dédie cette thèse à ma mère et à mon feu père,
à Mlle OBER Cindy qui m'a accompagné
pendant tous ces années,
à tous mes frères et sœurs.*

Table des matières

Table des figures	ix
Liste des tableaux	xiii
Introduction générale	1
Chapitre 1 Etat de l'art	7
1.1 Introduction	7
1.2 Méthodes de résolution et dérivation	9
1.2.1 Principe général des méthodes de prédiction-correction	10
1.2.2 Calcul des dérivées	12
1.2.3 Méthode des différences finies	14
1.2.4 Différentiation symbolique	16
1.3 Différentiation automatique	18
1.3.1 Principe	18
1.3.2 Modes de différentiation	20
1.3.3 Implémentation et utilisation des outils de différentiation auto- matique	23
1.3.4 Test de Taylor	32
1.4 De la MAN à l'approche Diamant	33
1.4.1 Principe de la MAN	34
1.4.2 Exemple : Problème de Bratu	36
1.4.3 Approche Diamant de la MAN	38
1.4.4 Implémentation générique en Matlab	40
1.5 Conclusions	41

Chapitre 2 Analyse de sensibilité des solutions calculées par la MAN	43
2.1 Introduction	43
2.2 Aspects théoriques	45
2.2.1 Position du problème	45
2.2.2 Différentiation d'un problème résolu par Diamant	47
2.2.3 Algorithmique du calcul de sensibilité	47
2.3 ADiMat	48
2.3.1 Différentiation de code avec ADiMat	51
2.3.2 Différentiation d'un code surchargé par Diamant	53
2.3.3 Couplage ADimat-Diamant	54
2.3.4 Validation de l'algorithme de sensibilité : Problème de Bratu modifié	57
2.4 Analyse de sensibilités des matériaux sandwich	58
2.4.1 Sollicitation statique d'une poutre Verre/PVB/Verre	59
2.4.2 Sensibilité par rapport à l'épaisseur	65
2.4.3 Sensibilité par rapport au module de Young de la couche visco-élastique	69
2.4.4 Sensibilité par rapport aux éléments de la matrice de rigidité . .	70
2.5 Conclusions	73
Chapitre 3 Analyse de sensibilité des modes propres des structures sandwich viscoélastiques	75
3.1 Introduction	75
3.2 Analyse de sensibilité des modes propres et fréquences propres de vibration	77
3.2.1 Analyse de sensibilité des problèmes aux valeurs propres linéaires	77
3.2.2 Méthodes de perturbation	78
3.2.3 Méthodes hybrides	81
3.2.4 Méthodes de dérivation analytique	87
3.3 Analyse de sensibilité de problèmes aux valeurs propres non-linéaires .	91
3.3.1 Homotopie et MAN	91
3.3.2 Procédure de continuation	92
3.3.3 Calcul de sensibilité par DA	93
3.4 Analyse de sensibilité des solutions propres dépendant de la fréquence .	94
3.4.1 Analyse de sensibilité du problème réel	95
3.4.2 Analyse de sensibilité du problème complexe	96

3.4.3	Calcul de sensibilité du problème non-linéaire	98
3.4.4	Implémentation	98
3.5	modélisation directe des structures sandwich viscoélastiques en vibration libre	100
3.5.1	Lois de comportement	102
3.6	Résultats numériques	106
3.6.1	Analyse de sensibilité par rapport à l'épaisseurs et par rapport au module d'Young	106
3.6.2	Sensibilité par rapport aux éléments de la matrice de rigidité	107
3.6.3	Validation du code linéaire tangent	107
3.6.4	Analyse de sensibilité du modèle viscoélastique à module complexe constant	109
3.7	Analyse de sensibilité du modèle viscoélastique faiblement amorti dépendant de la fréquence (PVB)	118
3.7.1	Analyse de sensibilité de la poutre viscoélastique en PBV par rapport au module de cisaillement	118
3.7.2	Sensibilité par rapport à la matrice de rigidité élémentaire	123
3.7.3	Etude de sensibilité du modèle viscoélastique fortement amorti (ISD112)	123
3.7.4	Sensibilité par rapport à la matrice de rigidité élémentaire de la plaque sandwich en ISD112	129
3.8	Estimation des temps de calcul	131
3.9	Conclusion	133
	Conclusion générale et perspectives	135
	Annexe A Matrices de comportement	139
	Bibliographie	141

Table des figures

1	organigramme de calcul de sensibilité par la DA	4
1.1	Principe de la méthode de prédiction-correction.	12
1.2	Différentiation symbolique de (1.17) par rapport à E_k à l'aide de Maple	17
1.3	Différentiation symbolique de (1.17) par rapport au paramètre η_k à l'aide de Maple	18
1.4	Calcul de dérivée par surcharge d'opérateur en Matlab	26
1.5	Aperçu d'écran définissant un objet <i>TaylorSeries</i>	30
1.6	Diamant 1	31
1.7	Diamant 2	31
1.8	Branches calculées par la MAN	35
1.9	Graphe générique du code implémentant $R(u, \lambda)$ dépendant des paramètres p . La partie surchargée avec Diamant est encadrée en bleu	41
2.1	Analyse de sensibilité des solutions de la MAN	46
2.2	Création et affectation d'une variable de type <i>Adderiv</i>	50
2.3	Différentiation par ADiMat.	55
2.4	Couplage ADiMat-Diamant. Les différentes étapes du couplage : différentiation ADiMat (cadre vert en pointillés), différentiation avec Diamant (cadre bleu)	57
2.5	Analyse de sensibilité de problème de Bratu	58
2.6	Délamination dans les structures sandwich	59

2.7	Poutre sandwich en verre laminé	59
2.8	Modèle zig-zag	61
2.9	Modèle non zig-zag	61
2.10	Sensibilité (Q^d, W^d) pour une perturbation de $0.5h_2$ la couche viscoélastique.	66
2.11	Gauche : Déflexion normalisée. Droite : Sensibilité de la déflexion à une perturbation de 50% de l'épaisseur de la couche viscoélastique.	69
2.12	Sensibilité (Q^d, W^d) pour une perturbation de $\epsilon = 0.5$ sur le module de Young de la couche viscoélastique.	70
2.13	Vecteurs de sensibilité normalisés tracés sur la courbe de réponse (Q, W) pour une perturbation de 10% de la matrice de rigidité élémentaire.	71
2.14	Norme de la partie normale des vecteurs de sensibilité pour une perturbation de 10 % de la matrice de rigidité élémentaire	72
2.15	Sensibilités de la déflexion par rapport aux perturbations de la matrice de rigidité élémentaire.	73
3.1	Courbes homotopiques de la fonction g à la fonction h	84
3.2	Courbe de la fonction α_T pour le matériau ISD112,	89
3.3	Structure sandwich avec le système de cordonnées.	100
3.4	Monogramme en fréquence réduite d'un matériau viscoélastique	103
3.5	Partie réelle et imaginaire des trois premiers modes de la poutre viscoélastique à module constant avec un facteur de perte $\eta = 1.5$	115
3.6	Sensibilité des trois premiers mode de la poutre viscoélastique à module constant avec un facteur de perte $\eta = 1.5$ pour une perturbation de $\delta h_f = 10\%$	115
3.7	Sensibilité des trois premiers mode de la poutre viscoélastique à module constant avec un facteur de perte $\eta = 1.5$ pour une perturbation de $\delta h_c = 10\%$	115

3.8	Partie imaginaire du mode 1 et sa sensibilité pour la plaque sandwich viscoélastique à module constant	119
3.9	Partie réelle du mode 1 et sa sensibilité pour la plaque sandwich viscoélastique à module constant	119
3.10	Partie imaginaire du mode 2 et sa sensibilité pour la plaque sandwich viscoélastique à module constant	119
3.11	Partie réelle du mode 2 et sa sensibilité pour la plaque sandwich viscoélastique à module constant	119
3.12	Partie imaginaire du mode 4 et sa sensibilité pour la plaque sandwich viscoélastique à module constant	120
3.13	Partie réelle du mode 4 et sa sensibilité pour la plaque sandwich viscoélastique à module constant	120
3.14	Mode 1 de la poutre sandwich bi-encastrée en PVB et sa sensibilité pour une perturbation de 20% du module de cisaillement G_f	122
3.15	Sensibilité du mode 1 de la poutre sandwich bi-encastrée en PVB pour une perturbation de 20% du module de cisaillement G_f	122
3.16	Sensibilité du mode 2 de la poutre viscoélastique avec un cœur en PVB par rapport à une perturbation de 20% du module de cisaillement G_f .	122
3.17	Déflexions de la poutre et sensibilités	122
3.18	Sensibilité du mode 1 de la poutre avec un cœur en PVB pour une perturbation 10% de la matrice de rigidité élémentaire au point $x = 0.25L$	124
3.19	Mode 1 de la poutre avec un cœur en PVB	124
3.20	Sensibilité du mode 2 de la poutre avec un cœur en PVB pour une perturbation 10% de la matrice de rigidité élémentaire au point $x = 0.25L$	124
3.21	Mode 2 de la poutre avec un cœur en PVB	124
3.22	Sensibilité du mode 1 de la plaque avec un cœur en ISD112 pour une perturbation 20% de la matrice de rigidité élémentaire au point $p(0.5L, 0.5l)$ avec la condition limite CFCF.	130

3.23	Partie réelle du mode 1 de la plaque avec sa sensibilité	130
3.24	Sensibilité de la partie réelle du mode 1 de la déflexion par rapport à une perturbations de 40% de la matrice de rigidité aux points $(L/4, l/2)$	130
3.25	Sensibilité de la Partie imaginaire du mode 1 de la déflexion par rapport à une perturbations de 40% de la matrice de rigidité aux points $(L/4, l/2)$	130
3.26	Partie réelle de la dérivée du mode 1 de la déflexion de la plaque pour une double perturbations aux points $(L/4, l/2)$ et $(3L/4, l/2)$	131
3.27	Partie Imaginaire de la dérivée du mode 1 de la déflexion de la plaque pour une double perturbations aux points $(L/4, l/2)$ et $(3L/4, l/2)$. . .	131
3.28	Sensibilité de la partie réelle du mode 2 de la déflexion par rapport à une perturbations de 0.4 de la matrice de rigidité aux points $(L/4, l/2)$, $(3L/4, l/2)$	132
3.29	Sensibilité de la partie imaginaire du mode 2 de la déflexion par rapport à une perturbations de 0.4 de la matrice de rigidité aux points $(L/4, l/2)$, $(3L/4, l/2)$	132

Liste des tableaux

1.1	Ecriture générique et exemples de problèmes non-linéaires.	13
1.2	Dérivation des fonctions et opérations élémentaires.	19
1.3	Formule de récurrence pour le calcul des dérivées d'ordre k des fonctions et opérations élémentaires.	28
1.4	Variation de ratio pour un code dérivé correct.	33
1.5	Algorithme de Diamant	42
2.1	Calcul de séries pour l'algorithme Diamant sensibilité	49
2.2	Propriétés mécaniques et géométriques de la poutre sandwich	60
2.3	Test de Taylor pour une perturbation dans la direction h_2 de l'épaisseur de la couche viscoélastique.	67
3.1	Algorithme de Diamant sensibilité	99
3.2	Paramètres du modèle Maxwell généralisé pour l'ISD112 à $20^\circ C$ et $27^\circ C$	105
3.3	Propriétés mécaniques et géométriques de la plaque sandwich	108
3.4	Test de Taylor pour une perturbation h_c^d dans la direction h_c de l'épais- seur de la couche viscoélastique.	108
3.5	Caractéristiques de la poutre sandwich à cœur viscoélastique à module constant.	109
3.6	Sensibilité des fréquences propres de la poutre encastree-libre par rap- port à une perturbation $\delta h_c = 10\%$ de la couche viscoélastique à module constant.	111

3.7	Sensibilité des facteurs de perte de la poutre encastree-libre par rapport à une perturbation $\delta h_c = 10\%$ de la couche viscoélastique à module constant.	112
3.8	Sensibilité des fréquences propres d'une poutre encastree-libre par rapport à une perturbation $\delta h_f = 10\%$ de la couche viscoélastique à module constant.	113
3.9	Sensibilité des facteurs de perte de la poutre encastree-libre par rapport à une perturbation $\delta h_f = 10\%$ de la couche viscoélastique à module constante.	114
3.10	Sensibilité des fréquences propres par rapport à une perturbation $\delta E_c = 10\%$ de la couche viscoélastique à module constant.	116
3.11	Sensibilité des facteurs de perte de la poutre encastree-libre par rapport à une perturbation $\delta E_c^d = 0.1$ de la couche viscoélastique à module constant.	116
3.12	Propriétés mécaniques et géométriques de la plaque sandwich à module constant.	117
3.13	Sensibilité des fréquences propres de la plaque sandwich viscoélastique à module constant par rapport à une perturbation du module de cisaillement G_0 de 10% de la couche viscoélastique à module constant	117
3.14	Sensibilité des facteurs de perte de la plaque sandwich viscoélastique à module constant par rapport à une perturbation du module de cisaillement G_c de 10% de la couche viscoélastique à module constant	118
3.15	Propriétés mécaniques et géométriques de la poutre sandwich Aluminium/PVB/Aluminium.	120
3.16	Sensibilité des fréquences propres de la poutre sandwich en PVB par rapport à une perturbation du module de cisaillement G_f de 20%	121
3.17	Sensibilité des facteurs de perte de la poutre sandwich en PVB par rapport à une perturbation du module de cisaillement G_f de 20%	121

3.18	Evolution des fréquences propres de la poutre avec un cœur en PVB pour une perturbation de la matrice de rigidité élémentaire aux point $x = L/2$	123
3.19	Propriétés mécaniques et géométriques d'une poutre sandwich a cœur en ISD112.	125
3.20	Sensibilité des fréquences propres de la poutre encastree-libre en ISD112 par rapport à une perturbation $\delta E_f = 20\%$ de l'épaisseur des faces . . .	126
3.21	Sensibilité des facteurs de perte de la poutre sandwich encastree-libre en ISD112 rapport à une perturbation $\delta E_f = 10\%$ de l'épaisseur des faces	126
3.22	Propriétés mécaniques et géométriques de la plaque sandwich a cœur en ISD112.	127
3.23	Sensibilité des fréquences propres de la plaque sandwich en ISD112 à $27^\circ C$ par rapport à une perturbation du module de cisaillement G_f de 20%	127
3.24	Sensibilité des facteurs de perte de la plaque sandwich en IDS112 à $27^\circ C$ par rapport à une perturbation du module de cisaillement G_f de 20% .	128
3.25	Normes des cinq premiers modes propres pour une perturbation de 20% du module de cisaillement G_f pour une plaque sandwich en IDS112 à $27^\circ C$	128
3.26	Temps de calcul pour une plaque constitué de 896 éléments	132

Introduction générale

Ces dernières décades, les matériaux intelligents et les structures adaptatives ont été largement étudiées pour une meilleure connaissance, puis l'amélioration de leurs propriétés multifonctionnelles. Les structures sandwich comprenant des couches viscoélastiques et/ou piézoélectriques sont communément utilisées dans les applications industrielles pour le contrôle acoustique et vibratoire. Les vibrations sont généralement indésirables car elles sont à l'origine de phénomènes tels que la fatigue et les nuisances sonores. Les fréquences propres jouent un rôle prépondérant. Leur connaissance permet de dimensionner les structures par rapport aux sollicitations et d'éviter le phénomène de résonance. Pour atténuer ou éliminer les vibrations dans les structures, la technique la plus souvent employée est la dissipation de l'énergie en incorporant aux structures des matériaux viscoélastiques dont le pouvoir amortissant est bien connu [1, 2, 3, 4].

Leur modélisation conduit en général à des problèmes non-linéaires. A coté de cela, l'analyse de sensibilité tend à devenir un outil classique [5]. Dans le cas des structures sandwich, les applications concernent la sensibilité par rapport aux imperfections, la localisation de défauts [6] et l'analyse des vibrations [7, 8]. Plusieurs approches ont été proposées pour calculer des sensibilités et déterminer les paramètres ayant l'influence la plus forte. Le choix d'une méthode par rapport à une autre dépend en général de la complexité du modèle, de l'existence d'appareils de mesure et du nombre de paramètres à étudier. Par exemple, les sensibilités peuvent être obtenues au travers d'expériences de laboratoires [9] ou d'approches analytiques qui peuvent être entreprises sur des systèmes d'équations relativement simples [7, 10]. La plupart des études analytiques,

éventuellement basées sur une formulation éléments finis, suppose une différentiation manuelle adaptée au problème traité [8, 10]. On note que l’approche “unifiée” proposée par Choi et al. [11, 7] pour l’analyse de sensibilité des problèmes de structures non-linéaires est incomplète et ne couvre pas toutes les méthodes numériques existantes. En particulier, elle n’est pas adaptée à la Méthode Asymptotique Numérique (MAN) [12, 13, 14] utilisée dans cette thèse pour résoudre les problèmes non-linéaires par une méthode de continuation, cas statique[15] et dynamique[16, 17], issu de la modélisation des sandwich. De manière générale, dans les formulations éléments finis implantées dans les codes de simulation, les sensibilités [18, 19] peuvent être soit approchées par une méthode de différences finies (FD), soit calculées en implémentant les équations continues différentiées puis discrétisées [20, 5, 21], ou bien en codant les équations discrètes différentiées, ou bien encore par différentiation du code implémentant les équations discrètes [22]. Dans ce dernier cas, les outils de différentiation automatique (DA) [23] peuvent largement simplifier ce travail. Dans cette thèse nous exploitons le potentiel de la différentiation automatique, à un premier niveau, dans le cadre de la résolution des problèmes non-linéaires par la MAN à l’aide de l’approche Diamant, puis, à un second niveau, pour calculer la sensibilité de ces solutions par rapport à des paramètres de modélisation. Les outils développés bénéficient de la précision de la MAN en terme de résolution du problème non-linéaire, et du caractère générique et facile d’utilisation de la différentiation automatique.

Un état de l’art sur la DA, la MAN et Diamant fait l’objet du chapitre 1. Très brièvement, la DA considère un code informatique, quel que soit sa taille, comme une séquence d’opérations élémentaires, de fonctions intrinsèques, d’instructions conditionnelles et de boucles. La DA est alors effectuée en appliquant la règle de différentiation de la loi de composition des fonctions à la précédente séquence, instruction par instruction, opération par opération. En utilisant la DA, les dérivées calculées sont exactes à la précision machine près. Cette technique générique constitue également une solution fiable pour la différentiation de haut degré, comme démontré pour la version Diamant

[24] de la MAN utilisant la DA. Dans la MAN, sous l'hypothèse d'analyticit , les solutions du probl me non-lin aire discret  tudi  sont approch es sous forme de s ries de Taylor tronqu es. Celles-ci sont introduites dans ce probl me particulier. Cela conduit   une suite de probl mes lin aires impliquant la m me matrice lin aire tangente et des termes de haut degr  dans le second membre. Les  tapes de diff rentiation, calcul de la matrice lin aire tangente et termes de haut degr , ont  t  automatis  par DA dans ce que l'on appelle l'approche Diamant [24, 25, 17]. Ce chapitre discute en particulier le potentiel de Diamant en terme de g n ricit , d'efficacit  et de simplicit  d'utilisation qui permet   l'utilisateur d'entrer son syst me d' quations de mani re naturelle sans avoir   se pr occuper ni de la m thode de r solution MAN, ni de la d rivation de son probl me. Comme le montre la figure 1, Diamant a  t  utilis  dans le cadre de la continuation classique [25] et de la continuation avec homotopie [17]. Le chapitre 2 de la th se porte sur l'analyse de sensibilit  de solutions calcul es par continuation classique et a donn  lieu   une premi re publication [15]. Le 3 me chapitre concerne la sensibilit  de solutions calcul es   l'aide de la technique d'homotopie. Le chapitre 2 pr sente les aspects th oriques de l'analyse de sensibilit  des probl mes non-lin aires r solus dans le formalisme Diamant. les sensibilit s sont calcul es par rapport aux param tres de mod lisation. Comme dans le cas g n ral, le calcul de sensibilit  peut  tre effectu  de mani res diff rentes, notamment en appliquant un outil de DA sur un code MAN [26]. Cette m thode peut se r v ler peu efficace car on ne tient pas compte de la nature math matique de la MAN. Comme indiqu  dans [27], le solveur g n rique de probl mes non-lin aires Diamant peut  tre consid r  comme un solveur math matique de haut niveau tout comme le sont par exemple les solveurs lin aires et les transformations lin aires (FFT ou FWT). Diff rentier directement les instructions informatiques utilis es pour coder ces op rateurs lin aires n'est efficace. Il en va de m me pour Diamant. Dans un premier temps, ce chapitre pr sente le calcul de sensibilit  dans le formalisme Diamant en faisant appara tre les d riv es mixtes – haut degr  en le param tre de chemin, ordre 1 par rapport aux param tres de mod lisation – compl tant ainsi l' tude pr li-

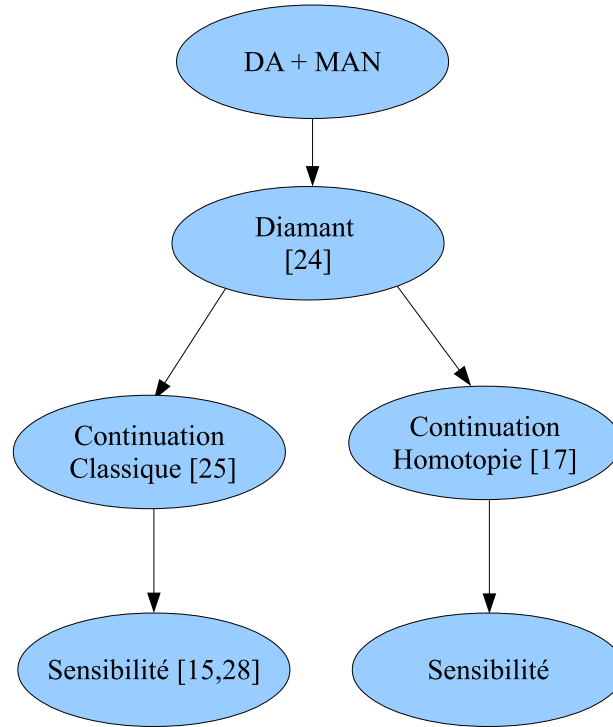


FIG. 1 – organigramme de calcul de sensibilité par la DA

minaire [28]. L'implémentation de la version Diamant-sensibilité est réalisée en Matlab de manière à s'adapter aux codes Diamant-sandwich existants [25]. Les dérivées mixtes sont calculées en deux logiciels de DA : Diamant pour la DA de haut degré et Adimat pour la dérivation d'ordre 1. Numériquement, nous avons appliqué notre méthode à la poutre sandwich viscoélastique Verre/PVB/Verre étudiée Verre/PVB/Verre [25] dans le cadre des non-linéarités géométriques (cas statique). La modélisation de cette poutre est réalisée à l'aide de la méthode des éléments finis et d'un modèle de type zigzag. Les sensibilités sont calculées par rapport à des perturbations de l'épaisseur et du module d'Young du cœur viscoélastique, mais aussi par rapport à des perturbations d'éléments de la matrice de rigidité pour simuler un défaut dans le matériau.

Le chapitre 3 s'intéresse à l'étude de la sensibilité des solutions du problème dynamique de vibration libre non-linéaire des structures sandwich viscoélastiques par rapport à des paramètres de modélisation. L'analyse de sensibilité des structures sandwich a intéressé plusieurs auteurs. Par exemple, Lima et al. [8] proposent une approche

basée sur une dérivation analytique qui implique une dérivation manuelle et une reformulation du problème pour chaque loi de comportement, ce qui en limite l'intérêt numérique. Lin et al. Des méthodes numériques basées sur la méthode de perturbation ont également été publiées [29, 30, 31], mais ces méthodes ne traitent que le cas des problèmes aux valeurs propres linéaires. Après avoir présenté les différentes méthodes d'analyse de sensibilité des problèmes aux valeurs propres, ce chapitre analyse expose notre méthode. Comme au chapitre 2, elle est basée sur une différentiation adaptée de l'algorithme Diamant impliquant l'homotopie utilisée dans [17]. Notre méthode permet d'analyser la sensibilité des valeurs propres et des modes propres de vibration non-linéaire des structures sandwich viscoélastiques pour différents modèles viscoélastiques (module viscoélastique constant, modèle faiblement amorti, et modèle fortement amorti). La sensibilité des facteurs de perte est également traitée. L'étude de sensibilité des poutres et plaques sandwich est effectué par rapport aux paramètres géométriques, matériels. L'influence de ses perturbation sur la partie réel et imaginaire des modes propres est présentée. Dans le cas de la perturbation de la matrice de rigidité élémentaire les fréquences propres de la structure ne sont pas considérablement affecté. Mais elles engendre une déformation local des modes propres.

Chapitre 1

Etat de l'art

1.1 Introduction

Le calcul des dérivées est un ingrédient essentiel en calcul scientifique. Le gradient, le Jacobien et les dérivées d'ordre supérieur sont nécessaires à la résolution des problèmes non-linéaires ou des problèmes d'optimisation par exemple. D'un côté, des codes de simulation industriels sont très souvent utilisés en mécanique pour résoudre ces problèmes d'équations aux dérivées partielles, mais ces codes sont, pour la plupart, dépourvus d'outils offrant la possibilité d'effectuer de manière simple des analyses de sensibilité. D'un autre côté, des chercheurs développent leur propres programmes informatiques soit pour répondre à une problématique donnée, soit pour tester de nouvelles méthodes numériques. De nos jours, ces codes sont généralement écrits en Fortran, C++ ou Matlab. Dans ces codes, la question du calcul des dérivées se pose régulièrement au niveau de la méthode d'approximation, de la méthode de résolution, de l'étude de la sensibilité de la solution du problème à des paramètres de modélisation, ou encore dans la résolution d'un problème optimisation par une méthode de gradient.

Lorsqu'on dispose d'un code informatique, les dérivées peuvent être calculées soit manuellement, soit numériquement par la méthode des différences finies, soit automatiquement par différentiation automatique (DA) [32, 23], soit par différentiation

symbolique. Cette dernière, limitée à quelques formules, n'est pas adaptée à la différentiation de codes de mécanique de grande taille. La méthode des différences finies a l'avantage d'être facile à mettre en œuvre, mais elle a l'inconvénient d'introduire des erreurs de troncature dans les dérivées, et donc d'avoir une précision liée à l'usage d'un pas de calcul optimal. La dérivation manuelle est possible au cours du développement du code, mais elle est source d'erreurs. Depuis quelques années, la différentiation automatique est une alternative à ces méthodes. Elle permet d'éviter les erreurs de troncatures et les erreurs humaines. Des outils de DA ont été développés notamment en Fortran, C++ et Matlab [33, 34, 35]. Ces outils permettent de calculer les dérivées des fonctions implémentées par un code informatique et plus généralement des programmes informatiques contenant des milliers de lignes [36]. Ces outils différencient les instructions d'affectation, les appels de routines, les instructions de contrôle et gèrent les trajectoires d'évaluation. Sans entrer dans les détails immédiatement (voir paragraphe 1.3), la DA permet d'augmenter un programme informatique en ajoutant des instructions pour le calcul des dérivées. Les dérivées ainsi calculées sont exactes à la précision machine près. La différentiation des codes peut se faire en mode linéaire tangent et en mode adjoint. Les dérivées de haut degré sont également accessibles. La DA a fait ses preuves dans de nombreux domaines scientifiques [37, 38]. Les apports de la DA à la simulation numérique concernent la généricité, l'efficacité et la simplicité d'implémentation du calcul des dérivées. De plus, elle offre de nouvelles opportunités pour exploiter des codes informatiques (méthodes inverses par exemple). La DA peut être utilisée pour le calcul des dérivées de haut degré.

La MAN est une méthode de continuation pour résoudre les problèmes non linéaires. Elle approche les solutions par des séries de Taylor en introduisant ces séries dans le problème non-linéaire considéré. On obtient une séquence de problèmes linéaires dépendant d'une même matrice Jacobienne et de seconds membres différents constitués de dérivées de haut degré. Comme montré au paragraphe 1.4.3, ces étapes de différentiation peuvent être automatisées en utilisant l'outil de différentiation Diamant [25, 24].

Ce chapitre est organisé en trois sections. La première, section 1.2, montre l'implication de la différentiation dans les méthodes de résolution des problèmes mécaniques non-linéaires en faisant apparaître les calculs de Jacobien et les séries de Taylor. La section 1.3 présente les principes de base de la DA et les modes de différentiation, puis décrit divers logiciels de DA. La section 1.4 discute l'automatisation de la Méthode Asymptotique Numérique [14] réalisée à l'aide de l'approche Diamant [39, 25].

1.2 Méthodes de résolution et dérivation

En ingénierie, la modélisation implique souvent la résolution d'un problème non-linéaire paramétrique s'écrivant sous la forme :

$$R(u, \dot{u}, \ddot{u}, \lambda) - F = 0, \quad (1.1)$$

Dans cette équation, le vecteur $u \in \mathbb{C}^n$ et la fonction vectorielle \mathcal{R} dépendent des variations du paramètre scalaire λ . Les vecteurs $\dot{u} = \frac{\partial u}{\partial t}$ et $\ddot{u} = \frac{\partial^2 u}{\partial t^2}$ sont les dérivées première et seconde de u par rapport au temps. Les solutions de ce système d'équations sous-déterminé définissent des branches de solutions (u, λ) . Les problèmes d'EDP munis de conditions aux limites adéquates peuvent s'écrire sous cette forme [40, 16, 41]. C'est le cas des problèmes de vibration non-linéaire génériques et libres :

$$K(\omega)u + C\dot{u} + M\ddot{u} = 0 \quad (1.2)$$

Plusieurs algorithmes de résolution peuvent être utilisés pour résoudre ces problèmes : suivi paramétrique naturel [42], méthode de prédiction-correction, méthode de continuation par pseudo-longueur d'arc [43]. Cette dernière est particulièrement intéressante car elle introduit l'équation de pseudo-longueur d'arc a

$$a = (u - u^j) \frac{\partial u}{\partial a} + (\lambda - \lambda^j) \frac{\partial \lambda}{\partial a}. \quad (1.3)$$

Cette approche permet de calculer la branche de solutions $(u(a), \lambda(a))$ à partir un point de départ (u^j, λ^j) . Parmi les méthodes figurent notamment la méthode de Newton-Raphson [44, 45] et la Méthode Asymptotique Numérique (MAN) [14, 46]. La différence

entre ces méthodes réside dans le type de prédicteur utilisé. En général, les prédicteurs sont basés sur le calcul de dérivées, et notamment celui d'un Jacobien.

1.2.1 Principe général des méthodes de prédiction-correction

Nous considérons l'équation résiduelle non-linéaire (1.1) pour illustrer ces méthodes et montrer la nécessité du calcul des dérivées dans les problèmes mécaniques. Dans (1.1), la variable u peut représenter par exemple le vecteur déplacement global et la variable λ l'effort appliqué. On suppose que (1.1) admet une solution.

Dans une méthode de prédiction-correction, on ne cherche pas une solution qui vérifie rigoureusement l'équation résiduelle (1.1), mais une solution (u, λ) vérifiant

$$\|\mathcal{R}(u, \lambda)\| < \epsilon, \quad (1.4)$$

où ϵ est la précision voulue. La résolution s'effectue itérativement en respectant les 3 phases suivantes.

Prédiction - Elle consiste à calculer une première approximation de la solution. Le point de départ est (u^j, λ^j) . Il n'est pas nécessairement solution du problème, mais on s'arrange généralement pour que ce point de départ soit proche de la solution. La prédiction tangente est définie par :

$$\begin{aligned} u^{j+1} &= u^j + \Delta u, \\ \lambda^{j+1} &= \lambda^j + \Delta \lambda, \end{aligned} \quad (1.5)$$

où les incréments de prédiction Δu et $\Delta \lambda$ doivent vérifier :

$$\mathcal{R}(u^j + \Delta u, \lambda^j + \Delta \lambda) = 0. \quad (1.6)$$

En faisant un développement limité de l'équation (1.6) au point (u^j, λ^j) on obtient

$$\mathcal{R}(u^j + \Delta u, \lambda^j + \Delta \lambda) = \mathcal{R}(u^j, \lambda^j) + \frac{\partial \mathcal{R}}{\partial u} \Delta u + \frac{\partial \mathcal{R}}{\partial \lambda} \Delta \lambda + \frac{1}{2} \frac{\partial^2 \mathcal{R}}{\partial u^2} \Delta u \Delta u + \dots, \quad (1.7)$$

où $\frac{\partial \mathcal{R}}{\partial u}$, $\frac{\partial \mathcal{R}}{\partial \lambda}$, $\frac{\partial^2 \mathcal{R}}{\partial u^2}$ sont les dérivées partielles du vecteur résidu par rapport aux variables u et λ . En particulier $\frac{\partial \mathcal{R}}{\partial u}$ est la matrice Jacobienne du vecteur résidu par rapport à la

variable vectorielle u . En négligeant les termes de haut degré dans l'équation (1.7) et en supposant que $\mathcal{R}(u^j, \lambda^j) \simeq 0$, l'incrément de prédiction peut être calculé en résolvant le système

$$\frac{\partial \mathcal{R}}{\partial u} \Delta u + \frac{\partial \mathcal{R}}{\partial \lambda} \Delta \lambda = 0. \quad (1.8)$$

Ce système comporte $(n + 1)$ inconnues pour n équations. Pour le résoudre, on lui ajoute la condition relative au choix de la longueur du pas de prédiction, par exemple l'équation de longueur d'arc (1.3). On pose alors $u^{j+1} = u^j + \Delta u$ et $\lambda^{j+1} = \lambda^j + \Delta \lambda$.

Correction - Dans le cas où la prédiction ne vérifie pas la condition $\|\mathcal{R}(u^{j+1}, \lambda^{j+1})\| < \epsilon$, on procède à une ou plusieurs corrections. La correction la plus utilisée est celle dite de Newton. Le point de départ est $(u^{(0)}, \lambda^{(0)}) = (u^{j+1}, \lambda^{j+1})$. En supposant que i corrections ont déjà été effectuées, cette phase consiste à chercher la correction $(\Delta u^{(i)}, \Delta \lambda^{(i)})$ à apporter à la prédiction $(u^{(i)}, \lambda^{(i)})$. La détermination de la correction $(\Delta u^{(i)}, \Delta \lambda^{(i)})$ est faite de manière similaire à (1.8), en résolvant le système :

$$\frac{\partial \mathcal{R}}{\partial u} \Delta u^{(i)} + \frac{\partial \mathcal{R}}{\partial \lambda} \Delta \lambda^{(i)} = -\mathcal{R}(u^{(i)}, \lambda^{(i)}), \quad (1.9)$$

auquel on ajoute une condition supplémentaire relative à la longueur d'arc imposée, comme dans le cas de la prédiction. A chaque étape de correction, on construit :

$$\begin{aligned} u^{(i+1)} &= u^{(i)} + \Delta u^{(i)}, \\ \lambda^{(i+1)} &= \lambda^{(i)} + \Delta \lambda^{(i)}. \end{aligned} \quad (1.10)$$

Lorsque la condition (1.4) est satisfaite, on met à jour (u^{j+1}, λ^{j+1}) comme étant la dernière correction $(u^{(i+1)}, \lambda^{(i+1)})$ calculée.

Continuation - Elle consiste à chercher la solution de la branche suivante en prenant un point de départ sur la branche solution précédemment calculée, puis à reprendre les deux étapes précédentes. En général, ce point de départ est pris à la fin de la branche précédente.

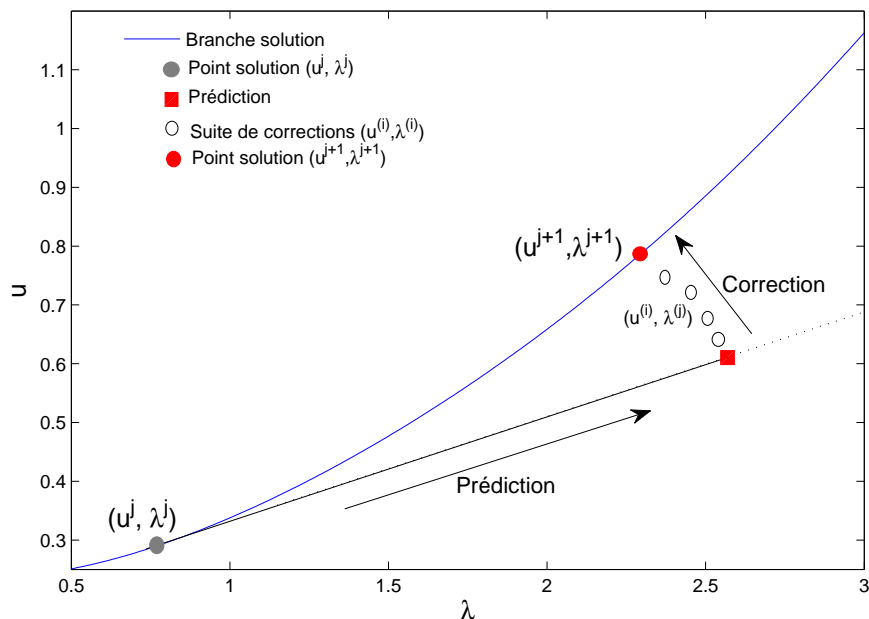


FIG. 1.1 – Principe de la méthode de prédiction-correction.

1.2.2 Calcul des dérivées

Ce bref aperçu sur les méthodes de prédiction-correction montre combien le calcul des dérivées est important en mécanique. La méthode de prédiction-correction ne garantit pas toujours une convergence vers la solution voulue, notamment lorsque le problème admet plusieurs solutions. Cette convergence dépend de la méthode de prédiction-correction utilisée. Ses performances sont fortement liées à la taille du pas de prédiction et au calcul précis de la matrice Jacobienne. Comme nous le verrons dans ce manuscrit, la différentiation automatique offre de nouvelles possibilités de calcul des dérivées dans les méthodes de résolution, mais aussi la perspective d'autres champs d'application. De premiers exemples sont présentés dans ce paragraphe pour, dès à présent, montrer le potentiel de la technique de DA.

Jacobien En mécanique, la matrice Jacobienne est le plus souvent calculée en différentiant manuellement les équations analytiques ou les équations discrétisées. Depuis

Type d'équations	Problèmes	Formulation générique	Quelques références
EDP algébriques	Valeurs limites	$\mathcal{R}(u, \lambda) = 0$	Bratu [40, 52, 53, 14, 24] Élasticité NL géométrique [39] Loi de comportement NL [14, 25] Vibration forcée NL [54] Contact [55]
	Valeurs propres	$\mathcal{P}(\lambda)u = 0$ $\mathcal{Q}(u, \lambda) = 0$	Valeurs propres NL [56] Vibration libre NL [16]
Dépendance en temps	EDO/EDA	$\mathcal{R}(u, \frac{\partial u}{\partial t}, \lambda) = 0$	Pendule [57, 51, 49] Plasticité [14] Friction [58]
	EDP	$\mathcal{R}(u, \frac{\partial u}{\partial t}) = 0$	Problème EDP d'instabilité [41]

TAB. 1.1 – Ecriture générique et exemples de problèmes non-linéaires.

quelques années, la DA offre une alternative pour le calcul du Jacobien lorsque le problème résiduel est représenté par un code informatique. Cette approche, qui reste méconnue en mécanique, facilite la mise en œuvre des méthodes de résolution des problèmes non-linéaires. Les composantes de la matrice Jacobienne sont obtenues en évaluant les codes dérivées produits par DA dans des directions de perturbation bien choisies. La DA est particulièrement efficace dans le cas des matrices creuses issues par exemple d'une discrétisation par la méthode des éléments finis.

Séries de Taylor Sous des hypothèses d'analyticit , les solutions des  quations diff rentielles ordinaires (EDO) [47, 48], des  quations diff rentielles alg briques (EDA) [49, 50, 51], des  quations aux d riv es partielles et des probl mes param triques [12, 14, 13] peuvent  tre approch es   l'aide de d veloppements en s ries de Taylor.

La diff rentiation automatique de haut degr  peut intervenir dans un grand nombre de probl mes m caniques. Le tableau 1.1 pr sente, sous forme g n rique [27], certains de ces probl mes et quelques exemple concrets r solus   l'aide de s ries de Taylor.

La m thode de suivi de courbe MAN [14] (M thode Asymptotique Num rique) est un cas particulier. Les solutions  crites sous forme de s ries tronqu es sont introduites dans le probl me discret d'EDP. Il en r sulte une s quence de probl mes lin aires pr -

sentant à chaque ordre la même matrice linéaire tangente et des termes de second membre constitués de dérivées de haut degré. Les systèmes linéaires obtenus sont résolus en utilisant un solveur direct. Les détails de cette méthode sont fournis dans la section 1.4. Les séries tronquées calculées en un point donné permettent d'approcher les solutions dans un voisinage de ce point.

Parmi les nombreux problèmes résolus par la MAN, on peut citer les problèmes incluant des non-linéarités géométriques ou matérielles [59, 60], des problèmes de mise en forme (plasticité) [61], des problèmes de vibration non-linéaire [60], des problèmes de bifurcation [61, 62], des problèmes de contact [63], les problèmes de flambement des structures plaque ou coque [13] et des problèmes de coque en grande rotation [61, 59]. Avec l'approche Diamant, la mise en œuvre de la MAN est devenue générique et simple. L'efficacité de Diamant, conférée par l'efficacité de la MAN et de la DA, peut être mise à profit pour l'analyse de la sensibilité des solutions des problèmes non-linéaires à des variations des paramètres de modélisation.

1.2.3 Méthode des différences finies

La méthode des différences finies (DF) peut être utilisée pour approcher les dérivées. Considérons un modèle mécanique défini par une fonction $M = M(p)$ où p représente l'ensemble des paramètres de modélisation, δp une direction de perturbation de norme 1 et α est un paramètre scalaire petit. Les dérivées premières de ce modèle peuvent être approchées,

$$\delta_M(\alpha\delta p) = \frac{M(p + \alpha\delta p) - M(p)}{\alpha} \quad \text{ou} \quad \delta_M(\alpha\delta p) = \frac{M(p + \alpha\delta p) - M(p - \alpha\delta p)}{2\alpha}, \quad (1.11)$$

par une différence avant (arrière) ou centrée selon les points disponibles et l'ordre de l'approximation. Ce choix dépend des conditions aux limites du problème dans le cas des méthodes de discrétisation des problèmes d'EDP. Ces approximations sont principalement utilisées pour la discrétisation et la résolution des problèmes d'EDP, et pour le calcul approché de gradients dans les méthodes d'optimisation [64].

Exemple 1 : Résolution des problèmes d'EDP - Dans les méthodes de résolution basées sur la méthode des DF, les dérivées sont remplacées par leurs approximations. Considérons le problème de diffusion défini par le système d'équations suivant :

$$\begin{cases} -\frac{\partial^2 u}{\partial x^2} + a(x)u(x) = f(x), \\ u(0) = u(1) = 0, \end{cases} \quad (1.12)$$

où $x \in [0, 1]$, $a(x)$ et $f(x)$ sont des fonctions connues à valeurs réelles, $u(x)$ est la fonction inconnue. On suppose que le problème a une solution. Définissons sur $[0, 1]$ une discrétisation équadistribuée à $n + 1$ points, de pas $h = \frac{1}{n}$. On note respectivement x_i et u_i l'abscisse et l'inconnue discrète $u(x_i)$ correspondant au nœud i . La dérivée partielle de second ordre est :

$$-\frac{\partial^2 u}{\partial x_i^2} = \frac{(2u_i - u_{i-1} - u_{i+1}))}{h^2}. \quad (1.13)$$

En remplaçant $\frac{\partial^2 u}{\partial x^2}$ par son approximation (1.13) en différences finies dans le système (1.12) on obtient le système discret (1.14), où $a_i = a(x_i)$, $f_i = f(x_i)$,

$$\begin{cases} \frac{2u_i - u_{i-1} - u_{i+1}}{h^2} + a_i u_i = f_i, & \forall \quad i = 2, \dots, n-1, \\ u_0 = u_n = 0. \end{cases} \quad (1.14)$$

Ce système peut s'écrire sous forme matricielle $KU = F$ avec

$$U = \begin{pmatrix} u_0 \\ \vdots \\ u_n \end{pmatrix}, \quad F = \begin{pmatrix} f_0 \\ \vdots \\ f_n \end{pmatrix} \quad (1.15)$$

et

$$K = \frac{1}{h^2} \begin{pmatrix} 2 + a_0 h^2 & -1 & & & \\ & -1 & 2 + a_1 h^2 & -1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 + a_{n-1} h^2 & -1 \\ & & & & -1 & 2 + a_n h^2 \end{pmatrix}. \quad (1.16)$$

La précision dépend du pas de discrétisation h choisi.

Exemple 2 : Analyse de sensibilité par calcul numérique d'une dérivée première - Le code implémentant le modèle (quelque soit la méthode de résolution employée) est exécuté deux fois. Le calcul de dérivée s'effectue en appliquant (1.11). Il a l'avantage d'être facile à mettre en œuvre. Par contre, sa précision est liée à l'utilisation d'un pas de discrétisation optimal (voir paragraphe 1.3.4).

1.2.4 Différentiation symbolique

La différentiation symbolique (DS) est proposée par certains logiciels de calcul formel comme Maple [65, 66, 67], Mathematica [68]... La fonction à différentier est supposée admettre une représentation symbolique au moyen de formules mathématiques. Pour Rall [32], le calcul des dérivées symboliques se fait à partir d'une bibliothèque d'expressions mathématiques et de leur expressions dérivées. Cette approche, appelée différentiation symbolique par transformation, associe une règle de transformation à chaque fonction mathématique élémentaire. Cela inclut la commande d'intégration "int" par exemple.

Dans un logiciel de DS, l'utilisateur entre la fonction à différentier puis lance la requête de différentiation en précisant les variables symboliques par rapport auxquelles la différentiation doit être faite. Le logiciel génère alors la fonction dérivée qui est une représentation symbolique. Un résultat numérique peut être obtenu en évaluant la formule dérivée, c'est à dire en attribuant des valeurs numériques aux variables symboliques. De manière générale, les outils de DS ne proposent que le mode direct de différentiation.

Prenons l'exemple de la formule de Maxwell généralisée

$$Maxwell(\omega, E_k, \eta_k) = E_0 + i\omega\eta_0 + \sum_{k=1}^n \frac{i\omega\eta_k}{E_k + i\omega\eta_k}, \quad (1.17)$$

pour le calcul du module opérationnel d'un matériau viscoélastique en fonction de la fréquence de vibration complexe ω , du module élastique E_0 et des paramètres caractéristiques du matériau viscoélastique η_k et E_k . Ces paramètres sont généralement

déterminés expérimentalement.

Dans l'équation (1.17), la constante i est l'unité imaginaire et le paramètre E_0 caractérise l'élasticité statique. Selon le modèle viscoélastique, le module opérationnel peut être constant ou dépendant de la température. Nous reviendrons plus en détail sur ces lois de comportements dans le troisième chapitre qui traite le cas dynamique.

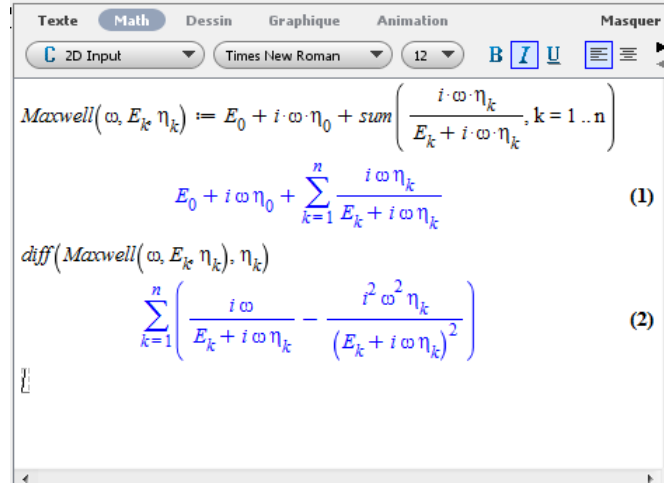
Cette équation est écrite symboliquement, puis différenciée à l'aide de Maple par rapport aux variables symboliques E_k et η_k . Les résultats sont reportés dans les figures 1.2 et 1.3. La différentiation symbolique est peu pratique pour la dérivation de codes

$$\text{Maxwell}(\omega, E_k, \eta_k) := E_0 + i \cdot \omega \cdot \eta_0 + \sum_{k=1}^n \left(\frac{i \cdot \omega \cdot \eta_k}{E_k + i \cdot \omega \cdot \eta_k} \right) \quad (1)$$

$$\text{diff}(\text{Maxwell}(\omega, E_k, \eta_k), E_k) = \sum_{k=1}^n \left(- \frac{i \cdot \omega \cdot \eta_k}{(E_k + i \cdot \omega \cdot \eta_k)^2} \right) \quad (2)$$

FIG. 1.2 – Différentiation symbolique de (1.17) par rapport à E_k à l'aide de Maple

complets et le calcul des dérivées de haut degré. En effet, la DS ne peut pas différencier des instructions contenant des instructions conditionnelles ou des boucles. Toutefois, les logiciels de DS sont de nos jours équipés une panoplie d'outils leurs conférant des capacités au-delà de la simple différentiation. La différentiation symbolique orientée-objet [69] en est un exemple.



$$\text{Maxwell}(\omega, E_k, \eta_k) := E_0 + i \cdot \omega \cdot \eta_0 + \text{sum} \left(\frac{i \cdot \omega \cdot \eta_k}{E_k + i \cdot \omega \cdot \eta_k}, k = 1 \dots n \right)$$

$$E_0 + i \omega \eta_0 + \sum_{k=1}^n \frac{i \omega \eta_k}{E_k + i \omega \eta_k} \quad (1)$$

$$\text{diff}(\text{Maxwell}(\omega, E_k, \eta_k), \eta_k)$$

$$\sum_{k=1}^n \left(\frac{i \omega}{E_k + i \omega \eta_k} - \frac{i^2 \omega^2 \eta_k}{(E_k + i \omega \eta_k)^2} \right) \quad (2)$$

FIG. 1.3 – Différentiation symbolique de (1.17) par rapport au paramètre η_k à l'aide de Maple

1.3 Différentiation automatique

1.3.1 Principe

La DA est un ensemble de techniques permettant d'augmenter un programme informatique afin de calculer des dérivées. Son principe est basé sur la différentiation de la loi de composition des fonctions. Pour une fonction F définie par $F(x) = g \circ f(x)$, où g et f sont des fonctions différentiables, cette dérivation s'écrit comme suit :

$$F'(x) = g'(f(x)) \times f'(x). \quad (1.18)$$

De manière générale, si la fonction F peut être écrite comme une composition de k fonctions élémentaires différentiables $f_i, i = 1 \dots k$:

$$F(x) = f_k(X_k) \circ \dots \circ f_i(X_i) \circ \dots \circ f_1(x), \quad (1.19)$$

alors la règle de différentiation à la chaîne s'écrit sous la forme :

$$F'(x) = f'_k(X_k) \times \dots \times f'_i(X_i) \times \dots \times f'_1(x), \quad (1.20)$$

où $X_i = f_{i-1} \circ \dots \circ f_1(x)$. Dans les outils de DA, la fonction à différentier est considérée comme une composition de fonctions élémentaires différentiables [36, 23] écrite sous

forme d'un programme informatique. Les fonctions intermédiaires f_i sont les opérations arithmétiques telles que $+$, $-$, $*$, $/$, et fonctions intrinsèques du langage de programmation telles que l'exponentielle, le logarithme et les fonctions trigonométriques. En implémentant les règles de différentiation dans les outils de DA, l'objectif est de calculer les dérivées analytiques exactes.

TAB. 1.2 – Dérivation des fonctions et opérations élémentaires.

Fonction	Dérivation	Opération	Dérivation
x	dx	$x + y$	$dx + dy$
$exp(x)$	$exp(x) * dx$	$x - y$	$dx - dy$
$Log(x)$	$\frac{dx}{x}$	$x * y$	$dx * y + x * dy$
$cos(x)$	$sinx * dx$	$\frac{x}{y}$	$\frac{dx * y - x * dy}{y^2}$
$sin(x)$	$cosx * dx$	x^n	$n * dx * x^{n-1}$ si n entier

Le tableau 1.2 présente les fonctions et opérations élémentaires avec leur dérivée. Ensuite, quelque soit de le morceau de code considéré, ce principe de dérivation est appliqué.

La DA s'applique à toutes les fonctions pouvant être représentées par un programme informatique. Les logiciels de DA utilisent des approches par transformation de source en mimant la dérivation manuelle (sans les erreurs humaines), ou par surcharge d'opérateur en tirant partie de la programmation orientée-objet proposée par certains langages informatiques. Juedes [70] a proposé une classification des outils de différentiation. Il existe deux modes de différentiation à l'ordre 1 : le mode linéaire tangent ou mode direct et le mode adjoint ou mode inverse.

1.3.2 Modes de différentiation

Mode linéaire tangent ou mode direct

Pour illustrer le mode direct de la DA, considérons la fonction :

$$\begin{aligned} f : \mathbb{R}^3 &\Rightarrow \mathbb{R}^3, \\ (x, y, z) &\mapsto f(x, y, z) = (x, y, x * y), \end{aligned} \tag{1.21}$$

faisant intervenir une seule multiplication. L'intérêt d'utiliser la fonction étendue f allant de \mathbb{R}^3 dans \mathbb{R}^3 se matérialise lors de l'écriture des équations dérivées en mode adjoint (voir paragraphe 1.3.2.0). On cherche à calculer la perturbation de sortie $\delta^s = (\delta_x^s, \delta_y^s, \delta_z^s)$ résultant d'une perturbation $\delta^e = (\delta_x^e, \delta_y^e, \delta_z^e)$ des variables d'entrée (x, y, z) de la fonction f . Pour le faire, on "construit" le Jacobien J_f de la fonction f par rapport aux variables x , y et z , pour écrire

$$\delta^s = J_f \delta^e = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ y & x & 0 \end{pmatrix} \begin{pmatrix} \delta_x^e \\ \delta_y^e \\ \delta_z^e \end{pmatrix} = \begin{pmatrix} \delta_x^e \\ \delta_y^e \\ y\delta_x^e + x\delta_y^e \end{pmatrix}, \tag{1.22}$$

et ensuite l'évaluer dans la direction d'entrée δ^e souhaitée.

Les instructions dérivées sont obtenues en remplaçant dans l'équation (1.21), les variables δ^e et δ^s par les variables informatiques (xd, yd, zd) . On obtient :

$$\begin{aligned} xd &= xd \\ yd &= yd \\ zd &= y * xd + x * yd \end{aligned} \tag{1.23}$$

Les deux premières instructions n'apportent pas d'information et ne sont pas "écrites" dans le code dérivé.

En pratique, le Jacobien n'est pas explicitement construit. L'outil de DA écrit les lignes de codes contenant les instructions dérivées et les instructions originales pour pouvoir évaluer le Jacobien dans la direction de perturbation souhaitée. Wengert [71] fut l'un des premiers à adopter cette approche pour une implémentation manuelle. Ce mode

de différentiation est particulièrement adapté au calcul des dérivées directionnelles et aux calculs de sensibilité.

Exemple : Calcul du coefficient de Lamé Considérons la formule de calcul du coefficient de Lamé λ en fonction du module d'Young E et du coefficient de Poisson ν :

$$f : \mathbb{R}^3 \Rightarrow \mathbb{R}^3, \quad f(E, \nu, \lambda) = \left(E, \nu, \frac{E\nu}{(1+\nu)(1-2\nu)} \right). \quad (1.24)$$

Il est un peu long et inutile d'écrire le Jacobien. Les équations linéaires tangentes correspondant à (1.24) sont

$$\begin{pmatrix} \frac{\partial \lambda}{\partial E} \\ \frac{\partial \lambda}{\partial \nu} \end{pmatrix} = \begin{pmatrix} \frac{\nu}{(1+\nu)(1-2\nu)} \\ \left(\frac{E}{(1+\nu)(1-2\nu)} - \frac{E\nu}{(1+\nu)^2(1-2\nu)} + \frac{2E\nu}{(1+\nu)(1-2\nu)^2} \right) \end{pmatrix} \quad (1.25)$$

et l'instruction linéaire tangente est :

$$\begin{aligned} \text{lamdad} = & ((Ed * nu + E * nud) * (1 + nu) * (1 - 2 * nu) - E * nu * (nud * (1 - 2 * nu) - (1 + nu) \\ & * 2 * nud)) / ((1 + nu) * (1 - 2 * nu)) ** 2 \end{aligned} \quad (1.26)$$

Pour calculer (1.25), il suffit d'évaluer (1.26) dans les directions (Ed, nud) égales à $(1, 0)$ pour la première composante et $(0, 1)$ pour la seconde. Cette instruction vient du code Fortran généré par Tapenade présenté au paragraphe 1.3.3. Fortran est utilisé ici car les codes dérivés produits par Tapenade sont plus simples à lire que les codes Matlab dérivés par Adimat [35] produits dans la suite de la thèse (voir 2.3).

Mode adjoint ou mode inverse

En mode adjoint, on recherche la perturbation δ^e à imposer en entrée du code pour obtenir la perturbation δ^s désirée en sortie de code. En d'autres termes, cela consiste à chercher l'opérateur Θ permettant de calculer la perturbation d'entrée $\hat{\delta}^e$ en fonction de la perturbation de sortie $\hat{\delta}^s$.

$$\hat{\delta}^e = \Theta \hat{\delta}^s. \quad (1.27)$$

En utilisant une approche de dérivées directionnelles dans l'espace adjoint, et la propriété des produits scalaires dans les espaces de Hilbert, on montre que l'opérateur Θ est exactement la transposée du Jacobien. Le mode inverse n'est pas aussi intuitif que le mode direct, mais cette astuce permet d'écrire les instructions adjointes relativement facilement.

Reprenons par exemple le cas de l'instruction $z = x * y$. Le calcul de la perturbation $\widehat{\delta}^e$ pour une perturbation $\widehat{\delta}^s$ à la sortie se fait comme suit :

$$\widehat{\delta}^e = J_f^T \cdot \widehat{\delta}^s, \quad (1.28)$$

c'est à dire :

$$\widehat{\delta}^e = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ y & x & 0 \end{pmatrix}^T \cdot \begin{pmatrix} \widehat{\delta}_x^s \\ \widehat{\delta}_y^s \\ \widehat{\delta}_z^s \end{pmatrix} = \begin{pmatrix} \widehat{\delta}_x^s + y\widehat{\delta}_z^s \\ \widehat{\delta}_y^s + x\widehat{\delta}_z^s \\ 0 \end{pmatrix}. \quad (1.29)$$

Les instructions de différentiation peuvent être déduites de l'équation précédente en remplaçant les directions $\widehat{\delta}^s$ et $\widehat{\delta}^e$ par les variables informatiques xb , yb et zb :

$$\begin{aligned} xb &= xb + y * zb \\ yb &= yb + x * zb \\ zb &= 0 \end{aligned} \quad (1.30)$$

Le suffixe b des variables dérivées fait référence au terme anglais *backward mode* qui désigne le mode adjoint. Toutes les instructions de l'équation (1.30) portent de l'information et doivent être reproduites dans le code adjoint. Le mode inverse nécessite plus d'efforts en terme de programmation que le mode linéaire tangent par contre, il est très avantageux pour le calcul des gradients de grande taille car toutes les composantes du gradient peuvent être évaluées simultanément.

1.3.3 Implémentation et utilisation des outils de différentiation automatique

DA par transformation de source

Avec les outils de différentiation par transformation de source [72, 35, 73], un code source (programme informatique de l'utilisateur) est augmenté des instructions permettant le calcul des dérivées en plus de l'évaluation du code source. Ces outils de DA peuvent différentier des programmes contenant des milliers de lignes, ce qui inclut le traitement des instructions événementielles telles que les boucles itératives et les expressions conditionnelles. La mise en œuvre de la DA par transformation de source peut toute fois nécessiter une modification du code source. C'est le cas en Matlab qui propose de nombreuses fonctions non-différentiables ou non-traitées par le logiciel de différentiation Adimat.

L'outil de DA par transformation de source peut faire une différentiation partielle ou totale. Dans cette thèse, nous utilisons principalement Matlab et l'outil de DA Adimat, mais pour ce premier exemple il est plus lisible d'utiliser Fortran et l'outil de DA Tapenade. Reprenons la formule de coefficient de Lamé (1.24) écrite en Fortran de la manière suivante :

```

1 C          Generated by TAPENADE      (INRIA, Tropics team)
2 C Tapenade 3.6 (r4343) - 10 Feb 2012 10:52
3 C
4          SUBROUTINE COEF_LAME(e, nu, lambda)
5          IMPLICIT NONE
6          REAL e, nu, lambda
7          lambda = e*nu/((1+nu)*(1-2*nu))
8          END

```

ligne de code pour le calcul du coefficient de Lamé La différentiation par transformation de source avec Tapenade de la fonction par rapport au module de Young E et au coefficient de poisson

Code linéaire tangent

```

1 C      Generated by TAPENADE      (INRIA, Tropics team)
2 C      Tapenade 3.6 (r4343) - 10 Feb 2012 10:52
3 C
4 C      Differentiation of coef_lame in forward (tangent) mode:
5 C      variations of useful results: lambda
6 C      with respect to varying inputs: e nu
7 C      RW status of diff variables: e:in nu:in lambda:out
8 C      SUBROUTINE COEF_LAME_D(e, ed, nu, nud, lambda, lambdad)
9 C      IMPLICIT NONE
10 C      REAL e, nu, lambda
11 C      REAL ed, nud, lambdad
12 C      lambdad = ((ed*nu+e*nud)*(1+nu)*(1-2*nu)-e*nu*(nud*(1-2*nu)-(1+nu)
13 C      + *2*nud))/((1+nu)*(1-2*nu))*2
14 C      lambda = e*nu/((1+nu)*(1-2*nu))
15 C      END

```

Contrairement à la différentiation symbolique, les dérivées ed et nud des paramètres d'entrée e et nu ne sont pas supposé égales à 1. Ils apparaissent dans le code différentié comme des variables. Ils définissent la direction de perturbation (ed, nud) . Pour évaluer les dérivées, il est donc nécessaire d'attribuer des valeurs aux variables linéaires tangentes : c'est ce qui permet de calculer des dérivées directionnelles.

DA par surcharge d'opérateur d'ordre un

La surcharge d'opérateur consiste à redéfinir des opérations mathématiques et les fonctions élémentaires pour un type de variable créé par le développeur. Dans le cas de la DA, le type est construit pour que les variables de ce type contiennent la valeur de la variable, et la valeur de sa dérivée. Pour le calcul des dérivées premières, nous définissons un type de variables nommé *Deriv* auquel nous associons les champs *val* et *deriv*. Considérons maintenant une variable réelle x dont la dérivée est notée xd . Avec le type *Deriv*, ces deux valeurs peuvent être regroupées en une seule variable X telle que :

$$X = \begin{cases} X.val = x \\ X.deriv = xd \end{cases} \quad (1.31)$$

En Matlab, le type *Deriv* est codé par le constructeur *Deriv.m* d'une classe appelée *@Deriv*. Le constructeur définit les propriétés des objets de cette classe. Le constructeur du type *Deriv* est donné par les lignes de code suivant.

Constructeur de la classe *Deriv*

```

1 function b=Deriv ( varargin )
2 if nargin==0
3     b.val=[];
4     b.deriv=[];
5     b=class(b, 'Deriv ');
6     return
7 end
8 if nargin==1
9     b.val=varargin{1};
10    b.deriv=1;
11    b=class(b, 'Deriv ');
12    return
13 end
14 if nargin==2
15     b.val=varargin{1};
16     b.deriv=varargin{2};
17     b=class(b, 'Deriv ');
18     return
19 end

```

Pour ce nouveau type de variable, le produit de deux variables, par exemple $Z = X * Y$, est effectué comme suit :

$$Z = \begin{cases} Z.val = X.val * Y.val \\ Z.deriv = X.deriv * Y.val + X.val * Y.deriv \end{cases} \quad (1.32)$$

La surcharge de l'opérateur est une redéfinition de l'opération de multiplication $*$ pour les variables du type *Deriv*. En Matlab, les lignes de code sont :

Code de surcharge de l'opération multiplication \times

```

1 function r=times(a,b)
2 if isa(a, 'Deriv ')
3
4     if isa(b, 'Deriv ')
5         r.val=a.val*b.val;
6         r.deriv=a.deriv*b.val+a.val*b.deriv;
7         r=Deriv(r.val, r.deriv);
8         return
9     else
10        r.val=a.val*b;
11        r.deriv=a.deriv*b;
12        r=Deriv(r.val, r.deriv);
13        return
14    end
15 else
16    r.val=a*b.val;
17    r.deriv=a*b.deriv;

```

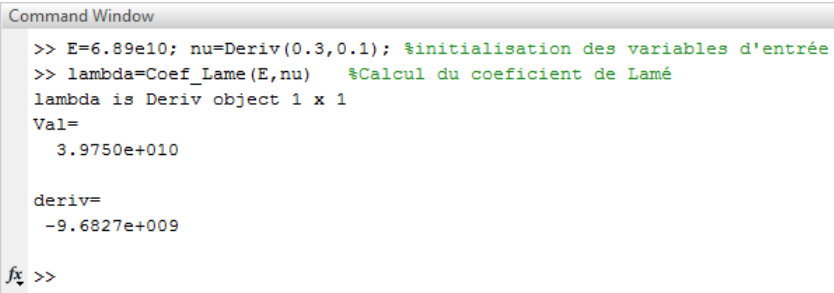
```

18     r=Deriv(r.val,r.deriv);
19     return
20 end

```

On tient compte ici de la multiplication entre deux variables de type *Deriv*, mais aussi entre une variable réelle et une variable de type *Deriv*, et vice et versa.

Les opérateurs surchargés sont regroupés dans une bibliothèque, celle-ci est écrite et validée une fois pour toutes. La différentiation est mise en place à la compilation du code surchargé (Fortran et C++), ou lors de l'interprétation dans le cas de Matlab par exemple. Elle en devient "invisible" à l'utilisateur. C'est le cas dans la figure 1.4 présentant les lignes de code nécessaires au calcul de la sensibilité du coefficient de Lamé λ par rapport à ν uniquement. La variable réelle $E = 6.89.10^{10}$ est le module d'Young, tandis que la variable ν est de type *Deriv* : sa valeur est 0.3, la perturbation est choisie égale à 0.1. On en déduit qu'une perturbation de 10% de ν induit une sensibilité négative d'environ 25% de λ . La fonction *coef_Lame* utilise implicitement la bibliothèque de surcharge d'opérateurs pour attribuer le type *Deriv* à λ et calculer à la fois sa valeur et sa dérivée.



```

Command Window
>> E=6.89e10; nu=Deriv(0.3,0.1); %initialisation des variables d'entrée
>> lambda=Coef_Lame(E,nu) %Calcul du coefficient de Lamé
lambda is Deriv object 1 x 1
Val=
    3.9750e+010

deriv=
   -9.6827e+009

fx >>

```

FIG. 1.4 – Calcul de dérivée par surcharge d'opérateur en Matlab

Utilisée par Rall [74], cette technique permet de minimiser les transformations du code source devant être effectuées par l'utilisateur avant la différentiation de celui-ci. Lors de l'évaluation des instructions, chaque opération est effectuée en fonction du type de ses opérands, il en va de même pour les fonctions.

DA d'ordre supérieur et la bibliothèque Diamant

La dérivation d'ordre supérieur obéit à des formules de récurrence (voir tableau 1.3). De nos jours, les outils de différentiation automatique qui calculent les dérivées de haut degré utilisent généralement la technique de surcharge d'opérateurs. Des formules de récurrence [23] ont été établies pour les opérations élémentaires et les fonctions intrinsèques du langage de programmation. La dérivation de haut degré implémente par exemple la formule de Leibniz pour calculer le coefficient de Taylor d'ordre k du produit de deux variables écrites sous forme de séries :

$$(u.v)_k = \sum_{j=0}^k u_j.v_{k-j} \quad (1.33)$$

où $u_r = \frac{1}{k} \frac{\partial^k u}{\partial a^k}$ et $v_r = \frac{1}{k} \frac{\partial^k v}{\partial a^k}$ et $w_k = \frac{1}{k} \frac{\partial^k w}{\partial a^k}$ sont les coefficients de Taylor de la fonction $u(a)$, $v(a)$ et $w(a)$. Dans le tableau 1.3, les termes avec le caractère \sim sont tels que $\tilde{u}_j = j u_j$ où u_j est le coefficient de Taylor de u . Notons que pour les fonctions trigonométriques \sin et \cos , le calcul de coefficient s_k et c_k se fait alternativement. L'outil de DA Diamant [24, 25] suit l'approche de programmation orienté-objet dite de surcharge d'opérateur pour cacher les aspects de la différentiation à l'utilisateur. Le type de données abstrait *TaylorSeries* est utilisé pour stocker des coefficients de Taylor jusqu'à l'ordre K . Les objets *TaylorSeries* disposent de trois champs :

- le champs *Value* qui stocke la valeur de la variable (ordre 0),
- le champs *order* qui représente l'ordre de troncature de la série,
- le champs *Coeff* qui stocke les valeurs des coefficients de Taylor (voir figure 1.5)

En Matlab, le constructeur de la classe est donné par le code présenté ci-dessous.

Code de surcharge de l'opération multiplication \times

```

1 function p=TaylorSeries(varargin)
2 if nargin==0
3     % EMPTY/DEFAULT CONSTRUCTOR
4     p.value=[];
5     p.coef=[];
6     p.order=[];
7     p=class(p, 'TaylorSeries');
```

fonction	formule de récurrence k
$w = u + v$	$w_k = u_k + v_k$
$w = u.v$	$w_k = \sum_{j=0}^k u_j v_{k-j}$
$w = \frac{u}{v}$	$w_k = \frac{1}{v_0} \left[u_k - \sum_{j=0}^k w_j v_{k-j} \right]$
$w = \sqrt{u}$	$w_k = \frac{1}{2w_0} \left[u_k - \sum_{j=0}^{k-1} w_j . w_{k-j} \right]$
$w = \exp(u)$	$\tilde{w}_k = \left[\sum_{j=1}^k w_{k-j} \tilde{u}_j \right]$
$w = \text{Log}(u)$	$\tilde{w}_k = \frac{1}{u_0} \left[\tilde{u}_k - \sum_{j=1}^k u_{k-j} \tilde{w}_j \right]$
$w = u^r$	$\tilde{w}_k = \frac{1}{u_0} \left[r \sum_{j=1}^k w_{k-j} \tilde{u}_j - \sum_{j=1}^{k-1} u_{k-j} \tilde{w}_j \right]$
$s = \sin(u)$	$\tilde{s}_k = \sum_{j=1}^k \tilde{u}_j c_{k-j}$
$c = \cos(u)$	$\tilde{c}_k = \sum_{j=1}^k -\tilde{u}_j s_{k-j}$

TAB. 1.3 – Formule de récurrence pour le calcul des dérivées d'ordre k des fonctions et opérations élémentaires.

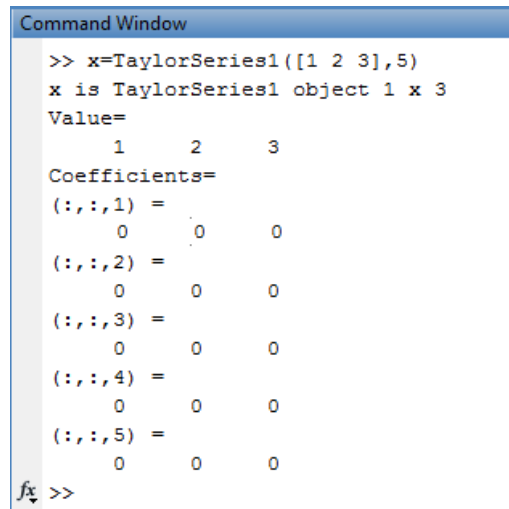

```

8         return;
9
10    elseif nargin==1
11        obj=varargin{1};
12
13        switch class(obj)
14
15            case 'TaylorSeries'
16                %COPY CONSTRUCTOR
17                p.value=obj.value;
18                p.coef=obj.coef;
19                p.order=obj.order;
20                p=class(p, 'TaylorSeries');
21                return;
22
23            case 'TaylorSeries2'
24                %CONVERSION CONSTRUCTOR
25                p.value=obj.value;
26                p.coef=obj.coef;
27                p.order=obj.order;
28                p=class(p, 'TaylorSeries');
29                return;
30
31            case 'double'
32                % CONVERT A 2-D or 1-D ARRAY TO TaylorSeries
33                p.value=obj;
34                p.order=20; %Default differentiation order
35                p.coef=zeros([size(obj) p.order]);
36                p.coef(:, :, 1)=0;
37                p=class(p, 'TaylorSeries');
38                return;
39        end
40
41    elseif nargin==2
42        obj_value=varargin{1};
43        obj_order=varargin{2};
44
45        p.value=obj_value;
46        p.order=obj_order;
47        p.coef=zeros([size(obj_value) p.order]);
48        p.coef(:, :, 1)=0;
49        p=class(p, 'TaylorSeries');
50        return;
51
52    elseif nargin==3
53        obj_value=varargin{1};
54        obj_order=varargin{2};
55        obj_coef=varargin{3};
56
57        % Check if the dimensions of coef and value match well
58        scoef=size(obj_coef);
59        svalue=size(obj_value);
60
61        if numel(scoef)==2 || scoef(1)~=svalue(1) ||
62            || scoef(2)~=svalue(2) || scoef(3)~=obj_order
63            error('Objet.value and Objet.coef dimensions mismatch')
64        end
65
66        p.value=obj_value;
67        p.order=obj_order;
68        p.coef=obj_coef;
69        p=class(p, 'TaylorSeries');

```

```

70     return;
71 end
72
73 error('Unsupported use of the Class TaylorSeries.');
```



```

Command Window
>> x=TaylorSeries1([1 2 3],5)
x is TaylorSeries1 object 1 x 3
Value=
      1      2      3
Coefficients=
(:, :, 1) =
      0      0      0
(:, :, 2) =
      0      0      0
(:, :, 3) =
      0      0      0
(:, :, 4) =
      0      0      0
(:, :, 5) =
      0      0      0
fx >>
```

FIG. 1.5 – Aperçu d'écran définissant un objet *TaylorSeries*

Issu d'une modification des premiers développements de la librairie Diamant [25], le constructeur prendre en compte les deux types définis pour les variables de *TaylorSeries* (*TaylorSeries1* et *TaylorSeries2*). Le premier développement de la boîte à outil Diamant utilise la classe *TaylorSeries1* qui recalcule à chaque ordre les coefficients de l'ordre 1 à l'ordre courant k . Cette version a été améliorée plus tard pour ne calculer que les coefficients à l'ordre courant (figures 1.6 et 1.7). La version 1 de Diamant (*TaylorSeries1*) recalcule les coefficients de l'ordre 1 à l'ordre courant k à chaque ordre alors que la version 2 de Diamant (*TaylorSeries2*) calcule les coefficients exactement à l'ordre courant k

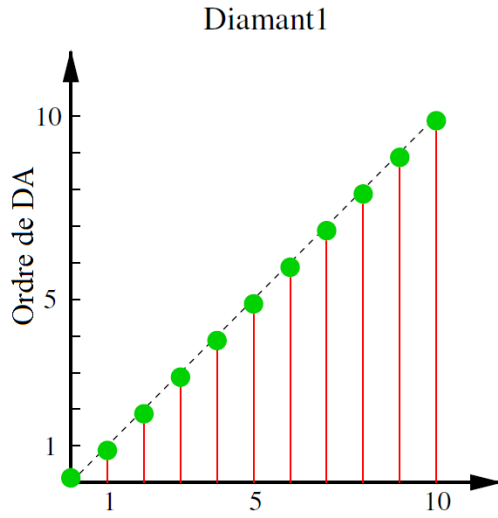


FIG. 1.6 – Diamant 1

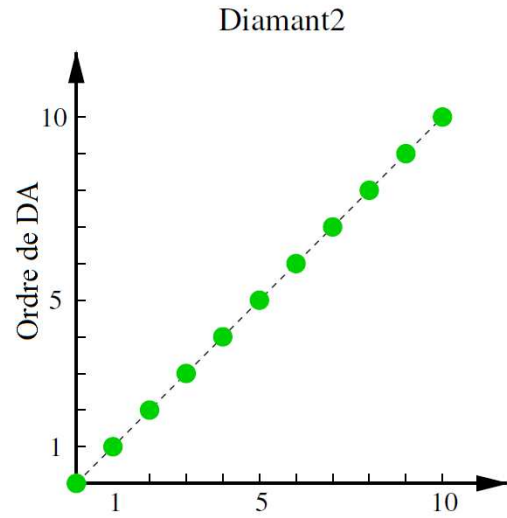


FIG. 1.7 – Diamant 2

Selon le nombre et le type des arguments en entrée, plusieurs cas sont considérés. Ces cas sont présentés dans le constructeur de classe en utilisant la syntaxe "case". Un objet *TaylorSeries* peut être créé de plusieurs manières. Par exemple, on peut le définir en fournissant une valeur pour le champ *value* et un ordre de troncature comme dans la figure 1.5. Dans ce cas, les coefficients de Taylor sont initialisés à zéro. Diamant implémente la formule de Leibniz comme suit :

Code de surcharge de l'opération multiplication \times

```

1 function r=mtimes(x,y)
2 %MTIMES objects multiplication.
3
4 global Ck %Current order of differentiation
5
6 if isa(x,'TaylorSeries')
7     if isa(y,'TaylorSeries')
8         r.value=x.value*y.value;
9         r.coef=zeros([size(r.value),x.order]);
10        r.order=x.order;
11
12        for kk=Ck:Ck
13            for ii=1:kk-1
14                r.coef(:, :, kk)=r.coef(:, :, kk)+...
15                    x.coef(:, :, ii)*y.coef(:, :, kk-ii);
16            end
17            r.coef(:, :, kk)=r.coef(:, :, kk)+...
18                x.value*y.coef(:, :, kk)+x.coef(:, :, kk)*y.value;
19        end

```

```

20         r=TaylorSeries(r.value,r.order,r.coef);
21         return;
22
23     else %x is a TaylorSeries while y is a double
24         r.value=x.value*y;
25         r.coef=zeros([size(r.value),x.order]);
26         r.order=x.order;
27
28         for kk=Ck:Ck
29             r.coef(:, :, kk)=x.coef(:, :, kk)*y;
30         end
31         r=TaylorSeries(r.value,r.order,r.coef);
32         return;
33     end
34
35 else %y is a TaylorSeries while x is a double
36
37     r.value=x*y.value;
38     r.coef=zeros([size(r.value),y.order]);
39     r.order=y.order;
40
41     for kk=Ck:Ck
42         r.coef(:, :, kk)=x*y.coef(:, :, kk);
43     end
44     r=TaylorSeries(r.value,r.order,r.coef);
45     return;
46 end

```

Lorsque tous les opérateurs et fonctions élémentaires ont été surchargés, validés et regroupés dans la bibliothèque Diamant, la dérivation d’une instruction mathématique complexe est effectuée opérateur par opérateur, fonction par fonction, en reproduisant implicitement la règle de dérivation des fonctions composées décrite plus haut.

1.3.4 Test de Taylor

Le test de Taylor est indispensable pour s’assurer que le code différentié ne comporte pas d’erreur. Ce test consiste à comparer les dérivées calculées avec le code linéaire tangent (ou le code adjoint) à celles calculées par la méthode des différences finies.

Soit un code \mathcal{M} ayant pour variable d’entrée $x \in \mathbb{R}^m$ et pour variable de sortie $y \in \mathbb{R}^n$. Pour le test, on choisit généralement une fonction objectif $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}$ à résultat scalaire et une direction de perturbation $\delta \in \mathbb{R}^m$. Puis on calcule le ratio r_α

pour des paramètres α décroissants (en général en puissances de 10 négatives).

$$r_\alpha = \frac{|\mathcal{FoM}(x + \alpha\delta) - \mathcal{FoM}(x)|}{|\Delta(\mathcal{FoM}(x)).\delta|}. \quad (1.34)$$

TAB. 1.4 – Variation de ratio pour un code dérivé correct.

α	r_α
10^{-1}	0.9741206349039363
10^{-2}	0.9977449190883242
10^{-3}	0.9997776685869196
10^{-4}	0.9999777980912296
10^{-5}	0.9999977896326924
10^{-6}	0.9999995500256210
10^{-7}	1.000000532685758
10^{-8}	0.9999827671112931
10^{-9}	0.9998133918902934
10^{-10}	0.9995718686253795

Le tableau 1.4 donne la variation du ratio r_α . Lorsque le code dérivée est correct, ce ratio tend linéairement vers 1 jusqu'à ce que α soit égal au pas optimal qui est de 10^{-6} dans ce test. le paramètre α doit être suffisamment petit pour permettre une bonne approximation des dérivées par la méthode des différences finies, et minimiser ainsi les erreurs de troncature. Par contre, lorsque α est trop petit, cela engendre des erreurs d'arrondi dues à la soustraction de nombres très proches.

1.4 De la MAN à l'approche Diamant

La MAN est basée essentiellement sur :

- la technique de perturbation qui consiste à approcher les solutions par des séries tronquées, puis à transformer le problème non-linéaire en une suite de problèmes linéaires faisant intervenir la matrice linéaire tangente et des dérivations de haut degré. Cette technique ne peut être appliquée que si la solution cherchée est analytique, et
- la technique de continuation. La branche de solutions $j + 1$ est calculée à partir de la solution (u^j, λ^j) comme suit

$$\begin{aligned} u^{j+1}(a) &= u^j + \sum_{k=1}^K a^k u_k, \\ \lambda^{j+1}(a) &= \lambda^j + \sum_{k=1}^K a^k \lambda_k, \end{aligned} \tag{1.35}$$

Elle est fréquemment associée à une technique de discrétisation qui consiste à résoudre le problème EDP non-linéaire par la méthode des éléments finis par exemple.

Les principaux freins à l'usage de la MAN sont l'hypothèse d'analyticit  (qui peut  tre lev e par r gularisation des  quations) et l' criture des termes de haut degr . L'approche Diamant, bas e sur la DA, l ve cette seconde limitation de mani re simple et efficace.

1.4.1 Principe de la MAN

La M thode Asymptotique Num rique (MAN) est une m thode de suivi de courbe d di e la r solution des  quations diff rentielles non-lin aires. Dans cette m thode, les solutions d'un probl me non-lin aire $R(U, \lambda) = 0$ suppos  analytique sont cherch es sous la forme de s ries de Taylor tronqu es   l'ordre K d velopp es par rapport   un param tre de chemin a [14]

$$u(a) = \sum_{k=0}^K a^k u_k, \quad \lambda(a) = \sum_{k=0}^K a^k \lambda_k, \tag{1.36}$$

o  les variables $u_k = \frac{1}{k!} \frac{\partial u^k}{\partial a^k}$ et $\lambda_k = \frac{1}{k!} \frac{\partial \lambda^k}{\partial a^k}$ sont les coefficients de Taylor   l'ordre k de u et λ  valu s au point $a = 0$. Le principe de la MAN est de transformer ce probl me non-lin aire en une succession de probl mes lin aires permettant de calculer it rativement ces coefficients de Taylor.

Les développements en série des inconnues u et λ sont introduits dans le problème non-linéaire pour obtenir une séquence d'équations linéaires dont la matrice linéaire tangente est commune à chaque ordre, et dont les seconds membres font intervenir des dérivées de haut degré [14]. Une seule inversion de matrice est nécessaire par branche (Fig. 1.8). Les approximations sous forme de séries sont utilisées pour calculer une branche (ou un morceau de branche) de la solution. Cette approximation est valide dans un domaine plus grand que les méthodes itératives d'ordre 1 comme la méthode de Newton-Raphson. Pour des raisons de clarté, la dépendance par rapport à a est généralement omise par la suite.

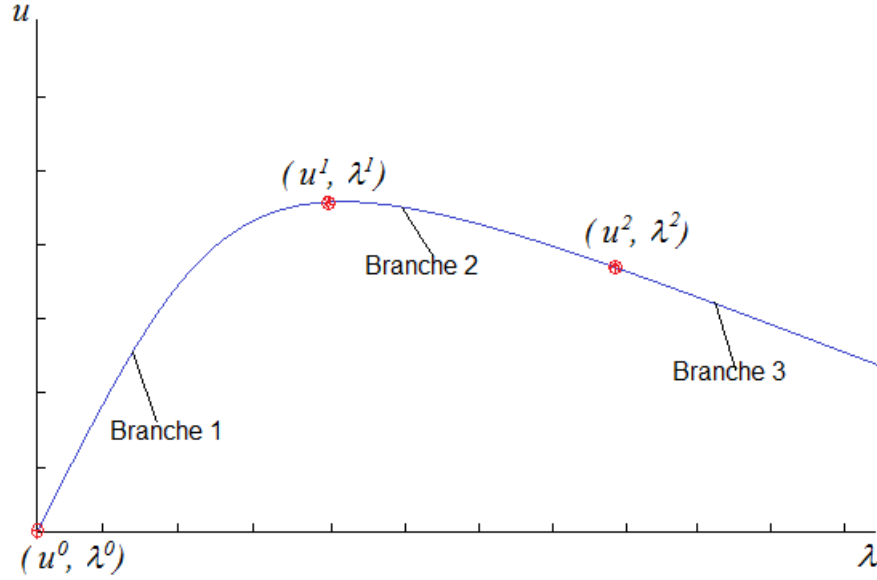


FIG. 1.8 – Branches calculées par la MAN

L'équation résiduelle (1.1) est sous-déterminée . On ajoute donc une équation de chemin à (1.1). Le problème est alors formulé par le système suivant :

$$\begin{cases} R(u, \dot{u}, \ddot{u}, \lambda) - F = 0, \\ a = \left\langle u(a) - u(0), \frac{\partial u}{\partial a}(0) \right\rangle + (\lambda(a) - \lambda(0)) \frac{\partial \lambda}{\partial a}(0). \end{cases} \quad (1.37)$$

Les inconnues u_k et λ_k sont calculées de manière itérative par la résolution des systèmes linéaires successifs. Cette étape faite, on détermine le rayon de convergence de la série

$[0, a_{max}]$ en choisissant par exemple :

$$a_{max} = \left(\epsilon \frac{\|u_1\|}{\|u_K\|} \right)^{\frac{1}{(K-1)}}, \quad (1.38)$$

où ϵ est un paramètre de précision choisi par l'utilisateur. En général, la MAN ne nécessite pas de corrections. Une façon d'améliorer l'efficacité de la MAN est de remplacer les séries par des approximations de Padé [75] pour augmenter le domaine de validité des séries.

1.4.2 Exemple : Problème de Bratu

Cet exemple classique illustre les différents aspects de la MAN :

$$R(u, \lambda) = \frac{\partial^2 u}{\partial x^2} + \lambda e^{u(x)} = 0, \quad u(0) = u(1) = 0. \quad (1.39)$$

Discretisation

Dans la MAN, pour calculer de manière simple les dérivées de haut degré, ce problème est écrit sous forme "quadratique" en introduisant la variable supplémentaire $v = \exp(u)$. Le problème devient alors :

$$R(u, \lambda) = \begin{cases} \frac{\partial^2 u(x)}{\partial x^2} + \lambda v(x) = 0, & x \in [0, 1], \\ v(x) = e^{u(x)}, \\ u(0) = u(1) = 0. \end{cases} \quad (1.40)$$

Nous procédons d'abord à la discrétisation du problème en utilisant une méthode de différences finies. Le segment $[0, 1]$ est discrétisé en utilisant $n + 1$ points équidistribués. Les conditions aux limites sont $u^0 = u^n = 0$. Le problème discret s'écrit sous la forme :

$$R([u], \lambda) = \begin{cases} K_0[u] + \lambda[v] = 0, \\ [v] - e^{[u]} = 0, \end{cases} \quad (1.41)$$

où $K_0, [u]$ et $h = \frac{1}{n}$ sont respectivement le Laplacien discret, les inconnues et le pas de discrétisation.

Perturbation de système discret

La technique de perturbation consiste à remplacer les inconnues u et λ par leurs séries tronquées à l'ordre k (1.36). Le problème à l'ordre 1 s'écrit sous la forme :

$$\begin{cases} K_0 [u_1] + \lambda^j [u_1] [v^j] + \lambda_1 [v^j] = 0, \\ [v_1] - u_1 [v^j] = 0. \end{cases} \quad (1.42)$$

où les vecteurs $[u_1]$ et $[v_1]$ sont les vecteurs de coefficients de Taylor de $[u]$ et $[v]$ à l'ordre 1.

A l'ordre k , le système s'écrit sous la forme :

$$\begin{cases} K_0 [u_k] + \lambda^j [u_k] [v^j] + \lambda_k v^j = f_k^{nl}, \\ [v_k] - u_k [v^j] = v_k^{nl}. \end{cases} \quad (1.43)$$

Dans la terminologie MAN, les termes f_k^{nl} et v_k^{nl} issus du développement en séries de λe^u et e^u vérifient les formules de récurrence :

$$\begin{aligned} f_k^{nl} &= -\lambda_0 v_k^{nl} - \sum_{r=1}^{k-1} \lambda_r v_{k-r}, \\ v_k^{nl} &= \sum_{r=1}^{k-1} \frac{k-r}{k} u_r v_{k-r}. \end{aligned} \quad (1.44)$$

Le système s'écrit sous la forme :

$$\begin{cases} K_0 [u_k] + \lambda^j [u_k] [v^j] + \lambda_k v^j = f_k^{nl}, \\ [v_k] - u_k [v^j] = v_k^{nl}, \end{cases} \quad (1.45)$$

En isolant les inconnues, les systèmes à résoudre sont :

à l'ordre 1 :

$$\begin{cases} K_t [u_1] + \lambda_1 [v^j] = 0, \\ [v_1] - u_1 [v^j] = 0. \end{cases} \quad (1.46)$$

à l'ordre k :

$$\begin{cases} K_t [u_k] + \lambda_k [v^j] = f_k^{nl}, \\ [v_k] - u_k [v^j] = v_k^{nl}, \end{cases} \quad (1.47)$$

où la matrice linéaire tangente $K_t = K_0 + K_d$ avec :

$$K_d = \begin{pmatrix} \lambda^j v^{j_0} & & \\ & \ddots & \\ & & \lambda^j v^{j_n} \end{pmatrix}. \quad (1.48)$$

Ces dérivées de haut de degré écrites sous forme de coefficients de Taylor diffèrent des formules de récurrence de dérivation classiques par l'absence des contributions linéaires tangentes de la k^{ieme} dérivée (somme sur $k - 1$). Cet exemple simple montre les difficultés d'implémentation liées à la mise en œuvre de la MAN : décomposition quadratique (introduction de la variable v) et calcul des termes de haut degré à l'aide de formules de récurrence. Même si la décomposition $L0 + L(u) + Q(u, u)$ [54] a longtemps été employée pour automatiser le calcul des formules de récurrence, le problème de la décomposition quadratique demeure. Lorsque le nombre d'opérations est important, le développement manuel devient rapidement fastidieux et source d'erreur.

L'approche Diamant, paragraphe 1.4.3, repose sur une initialisation judicieuse. Il est donc possible de calculer automatiquement ces termes par DA, sans avoir recours à la décomposition quadratique.

1.4.3 Approche Diamant de la MAN

L'objectif de Diamant est d'automatiser de manière générique et efficace le calcul des termes de haut degré f_k^{nl} résultant de la MAN grâce à la technique de la différentiation automatique.

L'approche Diamant est basée sur la formule de Faà di Bruno [76] pour le calcul des dérivées de haut degré d'une fonction composée $H = f \circ g$. La formule s'écrit :

$$\frac{\partial^n H}{\partial x^n} = \sum \frac{n!}{m_1! \dots m_n!} f^{(m_1+m_2+\dots+m_n)}(g(x)) \prod_{j=1}^n \left(\frac{g^{(j)}(x)}{j!} \right)^{m_j}, \quad (1.49)$$

où la somme est effectuée sur les partitions entières (m_1, \dots, m_n) de n . Dans cette formule, $f^{(j)}$ et $g^{(j)}$ sont respectivement les dérivées de f et g à l'ordre j . On regroupe

les termes (m_1, \dots, m_n) correspondants à un même entier et on écrit cette formule à l'aide des polynômes de Bell

$$\frac{\partial^n H}{\partial x^n} = \sum_{k=0}^n f^{(k)}(g(x)) B_{n,k}(g^{(1)}, g^{(2)}, \dots, g^{(n-k+1)}), \quad (1.50)$$

où les polynômes de Bell $B_{k,l}(g^{(1)}, \dots, g^{(k-l+1)})$, pour $l = 1 \dots k$, satisfont à :

$$B_{k,l}(g^{(1)}, \dots, g^{(n-k+1)}) = \sum \frac{k!}{i_1! \dots i_{n-k+1}!} \left(\frac{g^{(1)}}{1!} \right)^{i_1} \left(\frac{g^{(1)}}{2!} \right)^{i_2} \dots \left(\frac{g^{(n-k+1)}}{(k-l-1)!} \right)^{i_{n-k+1}}, \quad (1.51)$$

Cette somme implique toutes les suites d'entiers positifs i_1, \dots, i_{n-k+1} telles que $i_1 + i_2 + i_3 + \dots = l$ et $i_1 + 2i_2 + 3i_3 + \dots = k$.

L'approche Diamant repose sur les éléments suivants. L'équation résiduelle (1.1) est une composition de fonctions et la formule de Faà di Bruno peut être appliquée pour subdiviser de manière générique le coefficient de Taylor de \mathcal{R} d'ordre de k :

$$\mathcal{R}_k = \left\{ \mathcal{R}_{1|u_1=Id, \lambda_1=0} \right\} u_k + \left\{ \mathcal{R}_{1|u_1=0, \lambda_1=1} \right\} \lambda_k + \left\{ \mathcal{R}_{k|u_k=0, \lambda_k=0} \right\} = 0, \quad (1.52)$$

où $\left\{ \mathcal{R}_{1|u_1=Id, \lambda_1=0} \right\}$ est la matrice linéaire tangente en u , $\left\{ \mathcal{R}_{1|u_1=0, \lambda_1=1} \right\}$ est la dérivée partielle de \mathcal{R} par rapport au paramètre λ , et $\left\{ \mathcal{R}_{k|u_k=0, \lambda_k=0} \right\}$ représente la contribution des dérivées de haute degré f_k^{nl} . Dans ce nouveau formalisme, l'initialisation des coefficients de Taylor à l'ordre k de u et λ à zéro annule la contribution des termes linéaires tangents déjà pris en compte dans les deux premiers termes. Cette formule générique permet d'automatiser la MAN car les termes de l'équation (1.52) peuvent être calculés par un outil de différentiation automatique :

- Le Jacobien $\left\{ \mathcal{R}_{1|u_1=Id, \lambda_1=0} \right\}$ est le même quelque soit l'ordre de troncature. Il peut être calculé de manière efficace en tenant compte du caractère creux des matrices issues d'une discrétisation par éléments finis, et en particulier en basant sa construction sur le mécanisme d'assemblage.
- les termes contenant les dérivées d'ordre élevé et $\left\{ \mathcal{R}_{k|u_k=0, \lambda_k=0} \right\}$ peuvent être calculés automatiquement en utilisant la librairie de surcharge d'opérateur de Diamant.

Avec le formalisme Diamant, la séquence des K systèmes linéaires de la MAN s'écrit sous la forme :

$$\begin{cases} \left\{ \mathcal{R}_{1|u_1=Id, \lambda_1=0} \right\} u_k + \left\{ \mathcal{R}_{1|u_1=0, \lambda_1=1} \right\} \lambda_k = - \left\{ \mathcal{R}_{k|u_k=0, \lambda_k=0} \right\}, \\ a = (u_k - u_0)u_1 + (\lambda_k - \lambda_0)\lambda_1. \end{cases} \quad (1.53)$$

En pratique, le terme $\left\{ \mathcal{R}_{k|u_k=0, \lambda_k=0} \right\}$ et la solution du système (1.53) sont calculés de manière itérative de l'ordre 1 à K .

1.4.4 Implémentation générique en Matlab

Matlab est un environnement de calcul scientifique offrant de nombreuses boîtes à outils appelées *toolboxes* et notamment un langage de programmation orienté-objet. A la différence des langages compilés comme Fortran ou C, Matlab est un langage interprété qui propose un grand nombre de fonctions et de "toolboxes" pour simplifier le développement de nouvelles applications. De ce fait, Matlab est très utilisé pour la simulation numérique. Dans notre cas, nous sommes intéressés par la *toolbox* EDP.

Mais, aussi complet soit-il, Matlab ne dispose pas d'une boîte à outils officielle consacrée à la différentiation automatique car il est difficile de prendre en compte de toutes les fonctions différentiables de Matlab. Aujourd'hui, à l'exception d'Adimat, les outils de différentiation automatiques compatibles avec Matlab reposent sur des bibliothèques de surcharge d'opérateurs.

Notre implémentation de la méthode des éléments finis utilise les variables structurées de Matlab dans une implémentation générique des problèmes d'EDP en dimension 1, 2 ou 3. Ces variables structurées décrivent la géométrie, le maillage et les propriétés matérielles de la structure mécanique. Par exemple, les épaisseurs sont prises en compte dans la variable *geometry* alors que le module de Young est une composante de la variable *properties*. Les fonctions de forme sont fournies par l'utilisateur sous forme de fichier matlab en fonction de l'élément fini choisi pour la discrétisation. Ce choix de programmation permet à l'utilisateur de modifier facilement les paramètres de simulation, sans toucher à l'algorithme de résolution [15].

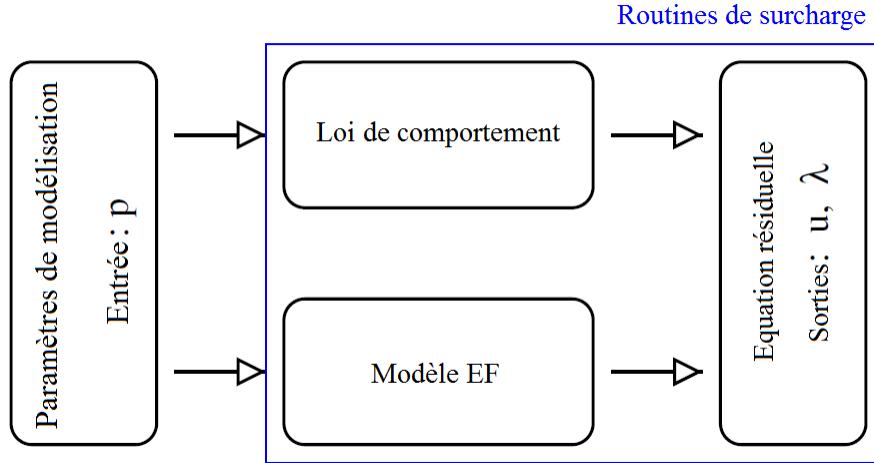


FIG. 1.9 – Graphe générique du code implémentant $R(u, \lambda)$ dépendant des paramètres p . La partie surchargée avec Diamant est encadrée en bleu .

La figure 1.9 présente le graphe générique du code implémentant le résidu $\mathcal{R}(u, \lambda, p)$. La variable p est une des variables d'entrée. La fonction \mathcal{R} est vue comme une composition de fonctions incluant la loi de comportement non-linéaire et la discrétisation éléments finis. L'algorithme de résolution de $\mathcal{R}(u, \lambda, p) = 0$ pour déterminer les variables de sortie (u, λ) est présenté dans la table 1.5.

1.5 Conclusions

L'état de l'art sur l'usage des dérivées présenté dans ce chapitre a mis en lumière le calcul de Jacobiens et de séries de Taylor dans les méthodes de résolution des problèmes mécaniques non-linéaires. Les approches traditionnelles (la dérivation manuelle, la dérivation symbolique), l'approximation par la méthode des différences finies, et la différentiation automatique sont discutées en indiquant les avantages et les inconvénients de chacune d'entre elles. Une large place est offerte à la technique de différentiation automatique comme :

- approche théorique adaptée au développement de certaines méthodes numériques.

En particulier, les principes de la DA sont introduits pour mettre en évidence sa

u_0, λ_0 connus,
Initialisation à 0 de u_k et λ_k
$\hat{u} = (u, \lambda)$
Construction de $K_T = \{\mathcal{R}_1 _{u_1=Id, \lambda_1=0}\}$
Ordre $k = 1$,
Calcul de $\mathcal{R}_1 = -F$,
Résolution de $K_T.\hat{u} = -R_1$,
Calcul de $\lambda_1 = \pm \frac{1}{\sqrt{1 + \hat{u}^t.\hat{u}}}$,
Ordre $k = 2, \dots, K$
Calcul du coefficient $\{\mathcal{R}_k _{u_1=0, \lambda_1=0}\}$,
Résolution de $K_T\hat{u} = -\{\mathcal{R}_k _{u_1=0, \lambda_1=0}\}$,
Calcul de $\lambda_k = \lambda_1.u_1.\hat{u}$, $u_k = \lambda_k.u_1 + \hat{u}$,

TAB. 1.5 – Algorithme de Diamant

généralité,

- solution technologique permettant de calculer des dérivées premières ou de haut degré de manière efficace et générique. La description des outils logiciels discute notamment les différents modes et les techniques d'implémentation de la DA dans les logiciels.

La section 1.4 traite l'automatisation de la Méthode Asymptotique Numérique [14] par DA, connue sous le nom d'approche Diamant [39, 25]. Le cadre de travail Diamant est discuté en détail car il est le support des développements autour de l'analyse de sensibilité des solutions calculées par la MAN présentée dans les Chapitres 2 et 3.

Chapitre 2

Analyse de sensibilité des solutions calculées par la MAN

2.1 Introduction

Ces dernières décennies, les matériaux intelligents et les structures adaptatives ont été intensivement étudiés dans le but d'améliorer leurs propriétés multifonctionnelles. Les structures sandwichs contenant des couches viscoélastiques ou piézoélectriques sont utilisées dans des applications industrielles pour le contrôle du son et de la vibration. De leur modélisation résulte souvent des problèmes non-linéaires.

L'analyse de sensibilité de modèles mécaniques tend à devenir un outil classique [5]. Elle permet d'évaluer l'impact de perturbations des paramètres d'entrée sur les variables de sortie d'un modèle. Ses applications concernent l'étude de sensibilité par rapport aux imperfections géométriques et matérielles des structures [20], à la localisation des endommagements [6] et à l'étude des vibrations [7, 8]. Plusieurs approches peuvent être utilisées. Les méthodes d'analyse de sensibilité sont groupées en trois classes [77] : (i) les méthodes de *screening* [78, 79, 80] qui effectuent une analyse qualitative de l'influence des paramètres d'entrée sur les variables de sortie, (ii) les méthodes d'analyse locale [81, 82] qui évaluent quantitativement l'influence d'une faible variation

des paramètres d'entrées, (iii) les méthodes d'analyse de sensibilité globale [83, 84] qui évaluent l'influence générale des paramètres d'entrée sur le comportement du modèle. Le choix d'une méthode ou l'autre dépend de la complexité des équations du modèle, des outils de mesure et du nombre de paramètres à étudier. La sensibilité peut être obtenue soit par expérimentation en laboratoire [9], soit par des approches analytiques [7, 10] ou numériques [85, 86].

Dans les codes de simulation par éléments finis, le calcul de sensibilité [18, 19] peut être effectué soit par la méthode des différences finies, soit en différentiant les équations continues ou les équations discrètes, soit en différentiant le code implémentant les équations discrètes manuellement [8, 10] ou à l'aide d'outils de différentiation automatique [23]. Même si le code à différentier a une taille importante, la DA le considère comme une séquence d'opérations élémentaires, de fonctions intrinsèques, d'instruction de contrôle et de boucles. La DA est effectuée en appliquant la règle de dérivation à la chaîne (1.20) à chaque instruction de ce code. L'utilisation de la DA permet de calculer des dérivées à la précision machine près. Cette technique générique est également fiable pour le calcul des dérivées de haut degré. L'analyse de sensibilité des solutions obtenues par la MAN est brièvement présentée dans [28] et dans [15]. Dans ce chapitre, nous proposons une discussion plus approfondie. Une attention particulière est portée sur la facilité d'utilisation, l'aspect générique, et la précision de Diamant pour la résolution d'un problème de mécanique conséquent et les calculs de sensibilité.

Ce chapitre présente les aspects théoriques de l'analyse de sensibilité des problèmes non-linéaires. Pour l'implémentation, une technique de calcul de dérivée mixte est développée en combinant deux logiciels de DA. Les résultats numériques concernent l'étude du comportement non-linéaire (non-linéarité de type géométrique) d'une poutre sandwich Verre/PVB/Verre. Les études de sensibilité sont faites par rapport aux épaisseurs des couches et par rapport à une perturbation de certains éléments de la matrice linéaire tangente.

2.2 Aspects théoriques

Comme décrit dans le Chapitre 1, la Méthode Asymptotique Numérique (MAN) a fait ses preuves dans des problèmes mécaniques incluant différents types de non-linéarités. L'objectif de cette partie est de mettre en place un algorithme de calcul de sensibilité adapté à Diamant en se basant sur les techniques de différentiation automatique.

2.2.1 Position du problème

Comme illustration, on considère le problème résiduel défini par l'équation (2.1).

$$\begin{cases} R(u(a, p), \lambda(a, p)) = 0, \\ a = \left\langle (u(a, p) - u(0, p)), \frac{\partial u(0, p)}{\partial a} \right\rangle + (\lambda(a, p) - \lambda(0, p)) \frac{\partial \lambda(0, p)}{\partial a}. \end{cases} \quad (2.1)$$

Outre la dépendance par rapport au paramètre de chemin a , on suppose que le problème dépend également d'un ensemble de paramètres p de nature géométrique, matérielle ou technique (paramètre de régularisation par exemple) sur lesquels peut se greffer une perturbation. Un paramètre et la perturbation correspondante peuvent être scalaires, vectoriels ou matriciels. Par la suite, la dépendance en p est omise pour plus de clarté dans les équations. Les systèmes linéaires issus de Diamant, voir équation (1.52),

$$\begin{cases} \left\{ \mathcal{R}_1|_{u_1=Id, \lambda_1=0} \right\} u_k + \left\{ \mathcal{R}_1|_{u_1=0, \lambda_1=1} \right\} \lambda_k = - \left\{ \mathcal{R}_k|_{u_k=0, \lambda_k=0} \right\}, \\ a = (u_k - u_0)u_1 + (\lambda_k - \lambda_0)\lambda_1, \end{cases} \quad (2.2)$$

dépendent implicitement des paramètres de modélisation p .

Les sensibilités $\frac{\partial u_k}{\partial p}$ et $\frac{\partial \lambda_k}{\partial p}$ des variables u_k et λ_k par rapport à p sont les dérivées mixtes en a et p . Elles sont notées u_k^d et λ_k^d dans la suite du manuscrit.

La figure 2.1 présente les différentes manières d'effectuer un calcul de sensibilité, par rapport aux paramètres, des solutions calculées avec la MAN ou Diamant. Dans chaque cas, le calcul de sensibilité peut être effectué de trois manières différentes. La méthode

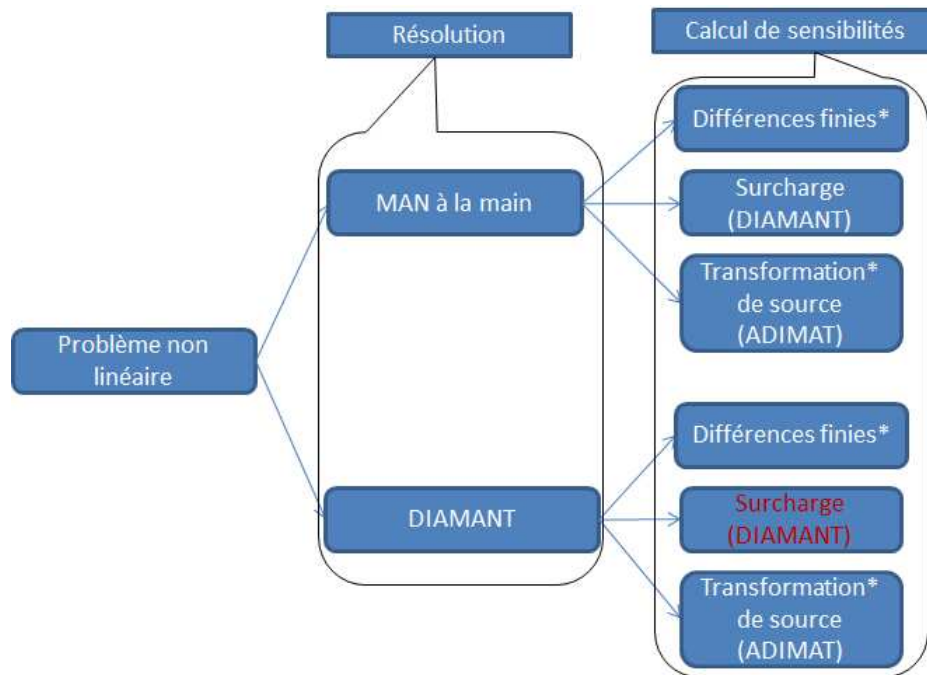


FIG. 2.1 – Analyse de sensibilité des solutions de la MAN

des différences finies présentée au chapitre 1 reste une méthode approchée. Elle ne nécessite pas de développement théorique. L'application d'un outil de DA sur un code MAN est réalisable de manière assez simple [26], mais peut se révéler peu efficace car on ne tient pas compte de la nature de la MAN. En effet, comme démontré dans [27], MAN et Diamant peuvent être considérés comme des solveurs mathématiques de haut niveau tout comme le sont par exemple les solveurs linéaires ou les transformations linéaires (FFT ou FWT). La Différentiation Automatique de Diamant, nécessaire pour le calcul des sensibilités, doit être effectuée en prenant en compte cette remarque pour être efficace.

2.2.2 Différentiation d'un problème résolu par Diamant

Pour calculer les sensibilités u_k^d et λ_k^d , il est naturel de différentier les systèmes génériques (2.2) par rapport à p . On obtient les systèmes :

$$\begin{cases} \left\{ \mathcal{R}_{1|u_1=Id, \lambda_1=0} \right\} u_k^d + \left\{ \mathcal{R}_{1|u_1=0, \lambda_1=1} \right\} \lambda_k^d = \\ - \left\{ \mathcal{R}_{1|u_1=Id, \lambda_1=0}^d \right\} u_k - \left\{ \mathcal{R}_{1|u_1=0, \lambda_1=1}^d \right\} \lambda_k - \left\{ \mathcal{R}_{1|u_k=0, \lambda_k=0}^d \right\}, \end{cases} \quad (2.3)$$

où les termes \mathcal{R}_k^d sont les dérivées du coefficient de Taylor \mathcal{R}_k avec des initialisations particulières de u_k et λ_k . Le système (2.3) dépend de la matrice linéaire tangente utilisée dans (1.52). Par contre, le second membre comporte des termes supplémentaires. Utilisée pour clore le système, l'équation de chemin (1.3) doit elle aussi être dérivée. La dérivée du paramètre de chemin satisfait à la condition :

$$a^d = (u_k^d - u_0^d)u_1 + (u_k - u_0)u_1^d + (\lambda_k^d - \lambda_0^d)\lambda_1 + (\lambda_k - \lambda_0)\lambda_1^d. \quad (2.4)$$

Le calcul des sensibilités implémenté dans le solveur Diamant est effectué en résolvant (1.53) et (2.3) de manière itérative de l'ordre 1 à l'ordre K . Les sensibilités u^d , λ^d , sont déduites des séries de Taylor par les expressions suivantes :

$$\begin{cases} u^d = u_0^d + \sum_{k=1}^K \left(k a^d a^{k-1} u_k + a^k u_k^d \right), \\ \lambda^d = \lambda_0^d + \sum_{k=1}^K \left(k a^d a^{k-1} \lambda_k + a^k \lambda_k^d \right). \end{cases} \quad (2.5)$$

Les équations (2.3) et (2.4) sont génériques elles aussi. Tous ces termes peuvent être calculés par DA.

2.2.3 Algorithmique du calcul de sensibilité

La résolution des équations (2.2)-(2.3) peut être entièrement automatisée en utilisant une combinaison d'outils de DA [15, 24] car les développements théoriques montrent la nécessité à la fois du calcul de dérivée de haut degré et des dérivées premières.

Comme Matlab a été utilisé dans le développement de codes mécaniques [87, 88, 89] pour la résolution des problèmes non-linéaires avec la MAN, il nous à sembler intéressant de compléter la version Matlab de Diamant pour pouvoir lui faire calculer des sensibilités. Pour cela, nous avons choisi de coupler les logiciels de différentiation Adimat et Diamant, Diamant pour le calcul des dérivées de haut degré de la MAN et Adimat pour la dérivation source à source des codes mécaniques par rapport au paramètre p . Cet outil est décrit en détail au paragraphe 2.3.

La mise en œuvre du calcul de sensibilité dans Diamant peut se résumer aux étapes suivantes :

1. Différentiation automatique avec Adimat du problème résiduel $\mathcal{R}(u, \lambda)$ pour obtenir le code linéaire tangent $\mathcal{R}^d(u^d, u, \lambda^d, \lambda)$,
2. Implémentation du calcul des séries (2.2) et du calcul de sensibilité (2.3),
3. Implémentation du pilote de continuation gérant la résolution des systèmes linéaires le long des branches de solutions.

Le pilote Diamant de calcul de sensibilité, tableau 2.1, est très voisin de celui de Diamant. Il est codé une fois pour toutes. Les tâches essentielles de l'utilisateur sont d'implémenter le problème résiduel comme pour Diamant, de le différentier en mode linéaire tangent avec Adimat, d'interfacer les codes résultants avec la bibliothèque Diamant de DA par surcharge d'opérateurs, d'exécuter le code, puis d'interpréter les résultats.

2.3 ADiMat

ADiMat [35] est un outil de différentiation automatique pour les codes Matlab. C'est un outil très original car il combine les techniques de transformation de source et de surcharge d'opérateurs pour produire des codes dérivés d'ordres 1 (mode linéaire tangent) et 2 (mode Hessian). Il a été développé à l'Université RWTH Aachen.

Comme Diamant, ADiMat s'appuie sur un type de variable particulier noté *Adderiv* pour effectuer la dérivation. Les champs des objets de type *Adderiv* sont :

$j = 1 ;$

$u_0^d, u_0, \lambda_0^d, \lambda_0$ connus,

initialisation à 0 de $u_k^d, u_k, \lambda_k^d, \lambda_k$, pour $k = 1, \dots, \mathcal{K}$

introduction de $\hat{u} = (u, \lambda)$ construction de $K_T = \mathcal{R}$ et de K_T^d

Ordre 1

calcul de $\mathcal{R}_1^d = -F^d$ et $\mathcal{R}_1 = -F$,

résolution de $K_T \cdot \hat{u} = -R$ et $K_T \cdot \hat{u}^d = -R_1^d - K_T^d \cdot \hat{u}$,

calcul de $\lambda_1 = \pm \frac{1}{\sqrt{1 + \hat{u}^t \cdot \hat{u}}}$; $\lambda_1^d = \pm \frac{\hat{u}^{d^t} \cdot \hat{u}}{(1 + \hat{u}^t \cdot \hat{u})^{\frac{3}{2}}}$,

Ordre $k = 2, \dots, K$,

calcul du coefficient $\{\mathcal{R}_{k|u_1=0, \lambda_1=0}\}$,

résolution de $K_T \hat{u} = -\{\mathcal{R}_{k|u_1=0, \lambda_1=0}\}$,

calcul de $\lambda_k = \lambda_1 \cdot u_1 \cdot \hat{u}$, $u_k = \lambda_k \cdot u_1 + \hat{u}$,

calcul de $-\mathcal{R}_k^d = -\left\{\mathcal{R}_{1|u_1=Id, \lambda_1=0}^d\right\} u_k - \left\{\mathcal{R}_{1|u_1=0, \lambda_1=1}^d\right\} \lambda_k - \left\{\mathcal{R}_{1|u_k=0, \lambda_k=0}^d\right\}$

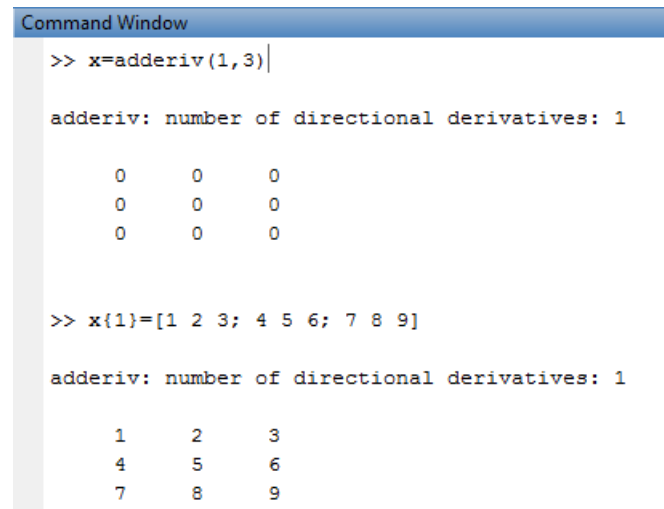
résolution de $K_T \hat{u}^d = -\mathcal{R}_k^d$,

calcul de $\lambda_k^d ; u_k^d$

TAB. 2.1 – Calcul de séries pour l'algorithme Diamant sensibilité

- *Dim* pour l'ordre de dérivation. Il est égal à 1 en mode linéaire tangent et à 2 en mode Hessien ;
- *ndd* pour le nombre de directions directionnelles propagées lors de l'évaluation du code dérivé. Lorsqu'il est supérieur à 2, les dérivées sont écrites sous forme matricielle. Cette approche est particulièrement intéressante lorsqu'on cherche à calculer un gradient ou un Jacobien en utilisant simultanément plusieurs directions de perturbation (base canonique par exemple).
- le champ *deriv* stocke la valeur numérique de la dérivée.

Dans la figure 2.2 sont présentées les instructions permettant de créer une variable de type *Adderiv* de dimension (3×3) dont le nombre de dérivées directionnelles *ndd* est égal à 1. Le champ *deriv* est tout d'abord initialisé à 0, puis affecté à l'aide de la seconde instruction.



```

Command Window
>> x=addderiv(1,3)

addderiv: number of directional derivatives: 1

    0    0    0
    0    0    0
    0    0    0

>> x{1}=[1 2 3; 4 5 6; 7 8 9]

addderiv: number of directional derivatives: 1

    1    2    3
    4    5    6
    7    8    9
  
```

FIG. 2.2 – Création et affectation d'une variable de type *Adderiv*.

Une description détaillée du constructeur de la classe *Adderiv* est fournie dans la documentation d'ADiMat. En plus des fonctions mathématiques usuelles, la classe *Adderiv* contient des méthodes spécifiques pour la différentiation de certaines fonctions mathématiques. Par exemple les méthodes *ls_mprod* et *ls_mexp* sont utilisées pour dériver l'opération de multiplication et la fonction exponentielle. Un grand nombre de

fonctions mathématiques de Matlab ont été surchargées. Les fonctionnalités de Matlab supportées par ADiMat sont présentées dans [35].

2.3.1 Différentiation de code avec ADiMat

L’objectif de ce paragraphe n’est pas de reprendre la documentation d’ADiMat, mais de décrire les bases de la différentiation de codes Matlab avec ADiMat permettant au lecteur de ce manuscrit d’éviter les difficultés que nous avons rencontrées au début. Quelques précautions doivent être prises pour faciliter la différentiation par transformation de source par le logiciel Adimat.

L’instruction pour différentier un code avec ADiMat est

$$\text{addiff}(@fname, indeps, deps) \quad (2.6)$$

où $fname$ est le nom de la fonction à différentier, $indeps$ représente les variables indépendantes (paramètres d’entrée) par rapport auxquelles la différentiation est faite, $deps$ représente les variables dépendantes (variables de sortie). Une fonction $fname$ ne peut être différentiée que si elle admet au moins une variable d’entrée et une variable de sortie.

ADiMat, appliqué au code source $\mathcal{R}(u, \lambda, p)$, permet de générer un code pour le calcul des dérivées de premier ordre par rapport aux paramètres de modélisation p . Le code linéaire tangent ainsi produit se nomme $g\text{-}\mathcal{R}(g_u, u, g_\lambda, \lambda g_p, p)$. Les variables et paramètres préfixés par $g_$ sont les dérivées premières des variables de même radical. Ainsi, g_u , g_λ et g_p désignent respectivement la dérivée de u , λ et p .

Prenons par exemple le code source permettant de calculer le module complexe d’un matériau viscoélastique en fonction de la fréquence.

Code source

```
1 function Ec=V(E0, neta0, E,w, neta, N)
2 wi=w*1j;
3 Ec=E0+wi*neta0;
4 for k=1:N
```

```

5      Ec=Ec+(wi*neta(k)*E(k))/(E(k)+wi*neta(k));
6  end

```

La différentiation de ce code par ADiMat nous donne le code linéaire tangent suivant :

Code linéaire tangent

```

1  function [g_Ec, Ec]= g_V(g_E0, E0, g_neta0, neta0, g_E,...
2      E, g_w, w, g_neta, neta, N)
3      g_wi= g_w* 1j;
4      wi= w* 1j;
5      g_tmp_V_00000= ls_mprod(g_wi, wi, g_neta0, neta0);
6      tmp_V_00000= wi* neta0;
7      g_Ec= g_E0+ g_tmp_V_00000;
8      Ec= E0+ tmp_V_00000;
9      clear tmp_V_00000 g_tmp_V_00000 ;
10     for k= 1: N
11         g_tmp_V_00001= (g_wi* neta(k)* E(k)+ wi* g_neta(k)* E(k)+...
12             wi* neta(k)* g_E(k));
13         tmp_V_00001= wi* neta(k)* E(k);
14         g_tmp_V_00002= ls_mprod(g_wi, wi, g_neta(k), neta(k));
15         tmp_V_00002= wi* neta(k);
16         g_tmp_V_00003= g_E(k)+ g_tmp_V_00002;
17         tmp_V_00003= E(k)+ tmp_V_00002;
18         g_tmp_V_00004= ls_mquot((g_tmp_V_00001), ...
19             (tmp_V_00001), (g_tmp_V_00003), (tmp_V_00003));
20         tmp_V_00004= (tmp_V_00001)/ (tmp_V_00003);
21         g_tmp_V_00005= g_Ec+ g_tmp_V_00004;
22         tmp_V_00005= Ec+ tmp_V_00004;
23         %Update detected : Ec= some.expression(Ec,...)
24         g_Ec= g_tmp_V_00005;
25         Ec= tmp_V_00005;
26     end

```

On remarque que les instructions du code source ont été décomposées en instructions élémentaires. Pour certaines opérations élémentaires “complexes” comme la multiplication et la division (leur différentiation implique plusieurs opérations), des méthodes spécifiques au type *Adderiv* sont codées dans Adimat pour “simplifier” le calcul de dérivées et “augmenter” la lisibilité du code linéaire tangent.

En pratique, la différentiation des codes avec ADiMat n’est pas aussi triviale que ne l’est la différentiation de codes fortran avec Tapenade par exemple. Il en a résulté plusieurs difficultés dans la mise en œuvre en Matlab du calcul des sensibilités des solutions MAN.

2.3.2 Différentiation d'un code surchargé par Diamant

Notre implémentation de la méthode des éléments finis utilise des variables structurées dont la différentiation est gérée par Adimat. Mais Adimat ne différentie pas les classes. C'est la principale difficulté que nous avons rencontrée car la DA de haut degré de la bibliothèque Diamant est basée sur la classe *TaylorSeries*. Deux options sont possibles.

La première solution consiste à dériver le code mécanique implémentant l'équation résiduelle avec Adimat, puis à modifier le type des variables impliquées dans les calculs de séries. Ces variables prennent le type *TaylorSeries*. Comme Matlab propage lui-même le type des variables, il suffit à l'utilisateur de modifier le type des variables au moment de leur initialisation. Ceci se fait de manière simple en dehors du code mécanique différentié par Adimat. Cette option nécessite quelques fois une gestion judicieuse des variables.

La seconde option consiste à mettre toutes les instructions relatives au Type *TaylorSeries* en commentaire, puis à différentier le code avec ADiMat. Une fois le code différentié, les instructions commentées sont rétablies. Les instructions à commenter concernent l'initialisation des variables *TaylorSeries* et les fonctions conversion des variables de type *TaylorSeries* en variables réelles.

Nous avons testé les deux options sur les codes développés dans [25, 17]. Sur des codes existants, la première option est plus facile à mettre en œuvre. Quelques imperfections d'Adimat nous sont apparues. Les difficultés rencontrées dans l'usage d'ADiMat sont exposées ci-dessous. Certaines sont reportées dans le guide utilisateur d'Adimat. Pour assurer une différentiation correcte du code, il est important d'en tenir compte.

Non-différentiation de certaines fonctions Matlab - Matlab est un langage de programmation interprété très complet, et les développeurs de l'outil Adimat n'ont pas à ce jour réussi à prendre en compte la différentiation de toutes ces fonctions. C'est le cas des instructions d'optimisation de code ou d'espace mémoire telles que *sparse*, et

de fonctions non-différentiables telles que *abs*. La présence de ces instructions génère des erreurs au cours de la différentiation du code.

Pour remédier à ce problème, ces fonctions peuvent être programmées de manière explicite.

Appels de fonctions - Des difficultés peuvent surgir lorsque la même fonction (dans le code utilisateur) est appelée plusieurs fois avec des arguments différant par leur nombre, ou par leur nature (variable active pour la différentiation, variable non-active ou constante).

Adimat génère une fonction différenciée à chaque appel de la fonction en gardant toujours le même nom pour la routine différenciée. Au final Adimat n'a généré qu'une fonction différenciée, celle correspondant chronologiquement au dernier appel. Si les arguments de la fonctions diffèrent au fil du code, le code linéaire tangent résultant est faux. Comme toujours, un test de Taylor (chapitre 1) s'impose, mais ce type d'erreur est très difficile à localiser.

2.3.3 Couplage ADimat-Diamant

L'objectif est maintenant d'organiser le calcul de la dérivation mixte de l'équation résiduelle $\mathcal{R}(u, \lambda)$ par rapport au paramètre de modélisation et par rapport au paramètre de chemin. Comme discuté dans le paragraphe 2.3.2.0.0, on a choisi d'utiliser la première option car ADiMat ne sait pas différencier les variables de type *Taylorseries*.

Principes - Le couplage ADiMat-Diamant repose sur trois éléments : la différentiation avec Adimat du code R , la surcharge du code différencié à l'aide de Diamant, la mise en œuvre de l'algorithme Diamant-sensibilité (Table 2.1) pour résoudre le problème non-linéaire et calculer la sensibilité des solutions simultanément.

La figure 2.3 indique les parties de code impactées par la différentiation linéaire tangente réalisée par rapport aux paramètres de modélisation. On observe que tout le

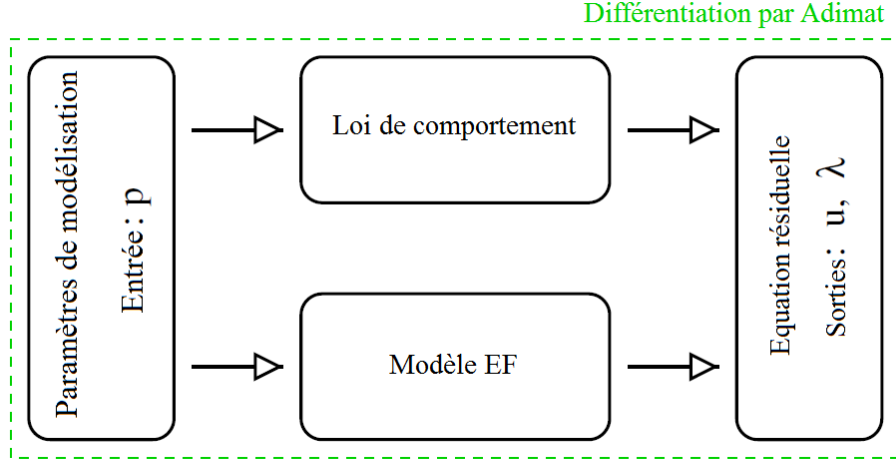


FIG. 2.3 – Différentiation par ADiMat.

modèle mécanique est concerné partant des paramètres de modélisation qui sont les “variables actives indépendantes” de la DA, aux solutions du problème non-linéaire qui sont les variables dépendantes dont la sensibilité nous intéresse. Suivant la règle de différentiation des fonctions composées, les variables dépendantes des paramètres intervenant dans la modélisation éléments finis et la loi de comportement sont également différenciées.

La différentiation du modèle mécanique $\mathcal{R}(u, \lambda, p)$ permet d’obtenir $g\text{-}\mathcal{R}(g\text{-}u, u, g\text{-}\lambda, \lambda, g\text{-}p, p)$. Nous avons vu dans le paragraphe 2.3.1 consacré à la différentiation avec Adimat que des fonctions comme *ls_mprod*, *ls_exp* spécifiques peuvent être présentes dans le code linéaire tangent $g\text{-}\mathcal{R}$. Le couplage Adimat-Diamant repose essentiellement sur la surcharge de ces fonctions pour les variables de type *TaylorSeries* de la librairie Diamant. Ceci conduit à l’extension de la librairie Diamant.

Extension de la librairie Diamant - ADiMat utilise une approche combinant les deux techniques de différentiation que sont la transformation de source et la surcharge d’opérateur. La technique utilisée pour chaque opération ou fonction dépend des dimensions des variables impliquées (matrice, réel, vecteur). Ainsi une multiplication entre variables matricielles n’est pas différenciée de la même manière qu’une multiplication

entre variables réelles. Dans Adimat, la première est faite par transformation du code en introduisant *ls_mprod*, puis utilisation de l'opérateur surchargé de *ls_mprod*, tandis que la seconde est faite par transformation de source.

La librairie Diamant est étendue à toutes les fonctions Adimat intervenant dans la différentiation linéaire tangente. Pour ce faire, nous avons consulté la liste des fonctions disponibles dans ADiMat. La surcharge de ces fonctions pour la dérivation de haut degré peut être faite explicitement en codant les formules de récurrence, ou implicitement en faisant appel aux fonctions préexistantes de Diamant. Cette seconde option est utilisée pour la surcharge de la fonction *ls_mprod*. Elle s'écrit :

$$\begin{aligned} \text{function} \quad [r] &= \text{ls_mprod}(ad, a, bd, b); \\ r &= \text{mtimes}(ad, b) + \text{mtimes}(a, bd); \end{aligned} \tag{2.7}$$

où la fonction *mtimes* est la multiplication matricielle déjà surchargée dans Diamant. Ici, les variables matricielles *a* et *b* peuvent représenter ou non une série, mais sont actives pour la différentiation. Les variables dérivées sont notées *ad* et *bd*, et sont de même nature et de même dimension que *a* et *b*. Il en va de même pour la fonction *ls_quot* d'Adimat qui calcule le quotient de deux variables qui est surchargée comme suit :

$$\begin{aligned} \text{function}[r] &= \text{ls_quot}(ad, a, bd, b) \\ r &= \frac{\text{mtimes}(ad, b) - \text{mtimes}(a, bd)}{\text{mpower}(b, 2)}; \end{aligned} \tag{2.8}$$

Couplage effectif

Le code doit ensuite être interfacé à la bibliothèque Diamant pour calculer les dérivées de haut degré par rapport au paramètre de chemin *a*. Pour le faire, nous avons opté pour une conversion des variables de type *Adderiv* en variables de type double dans les cas des paramètres, et en variables de type *TaylorSeries* pour les inconnues du problème nécessitant un développement en séries.

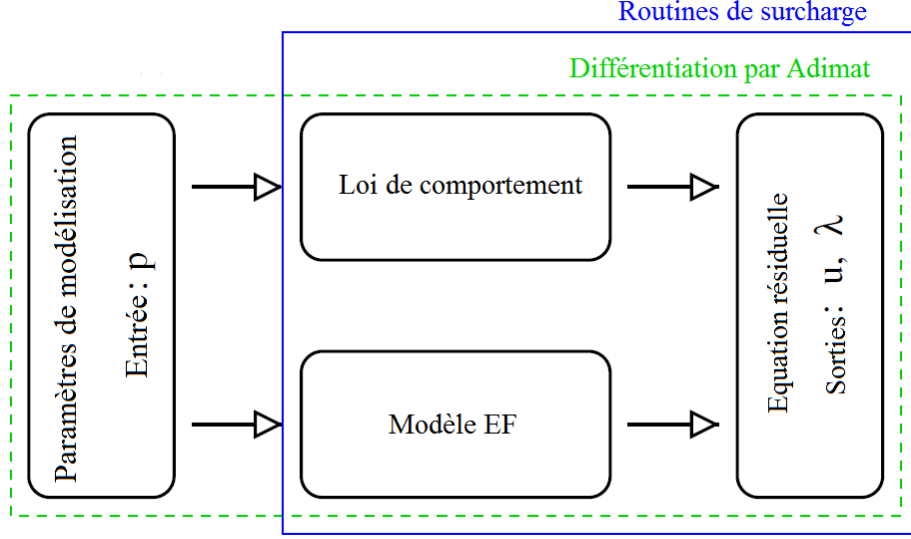


FIG. 2.4 – Couplage ADiMat-Diamant. Les différentes étapes du couplage : différenciation ADiMat (cadre vert en pointillés), différenciation avec Diamant (cadre bleu)

2.3.4 Validation de l'algorithme de sensibilité : Problème de Bratu modifié

Le problème de Bratu traité au paragraphe 1.4.2 par Diamant est utilisé pour valider nos calculs de sensibilité par comparaison avec les résultats publiés dans [28]. Ce problème est modifié en ajoutant une perturbation p sur les termes diagonaux de la matrice de rigidité :

$$\begin{cases} \frac{\partial^2 u(x, p)}{\partial x^2} + pu(x, p) + \lambda(p)e^{u(x, p)} = 0, \\ u(0, p) = u(1, p) = 0. \end{cases} \quad (2.9)$$

Ce problème a été étudié à la fois par utilisation de l'algorithme Diamant sensibilité et le couplage Adimat-Diamant, mais aussi par la différenciation par Adimat du code MAN codant les équations à l'aide de la décomposition quadratique [14]. Comme attendu, les deux approches donnent exactement les mêmes résultats numériques. La figure 2.5 compare la solution obtenue par l'approche Diamant (courbe en trait bleu) à celle obtenue en utilisant un code MAN (losange rouge). Ces deux solutions se

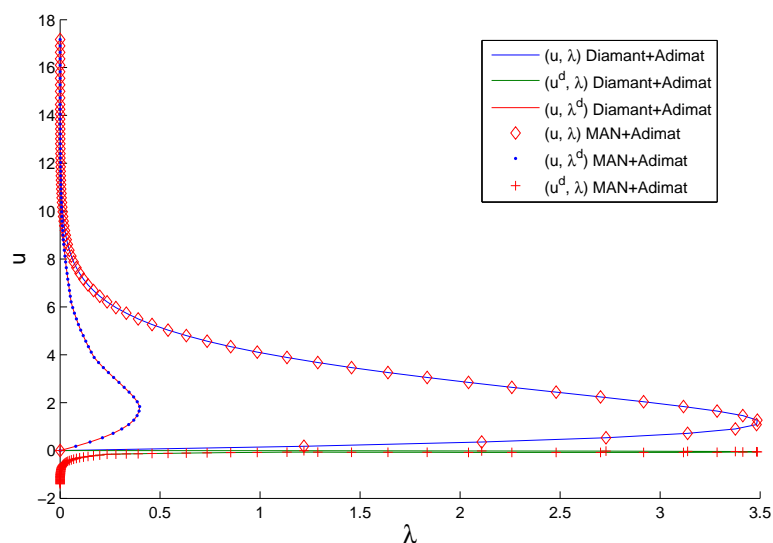


FIG. 2.5 – Analyse de sensibilité de problème de Bratu

superposent parfaitement. Il en va de même pour leurs sensibilités.

2.4 Analyse de sensibilités des matériaux sandwich

Les structures sandwich, au travers de leurs propriétés mécaniques, ont prouvé leur utilité dans des domaines tels que l'aéronautique, la construction automobile ou le bâtiment. Un matériau sandwich est constitué de trois couches au moins, des parements élastiques (en particulier en face extérieure), assemblés entre eux par au moins une autre couche. Dans un matériau sandwich viscoélastique tricouche, la couche intermédiaire est généralement un matériau polymère. Ces matériaux peuvent présenter des défauts liées au procédé de fabrication, ou résultant du chargement. Ces défauts ont une influence sur le comportement mécanique des matériaux et sont à l'origine de problème de délamination [90, 91, 92, 93] (voir figure 2.6). L'impact de la géométrie, épaisseurs asymétriques [94], sur les vibrations libre des matériaux sandwich a également fait l'objet d'études. Ces différents problèmes peuvent être étudiés au moyen d'analyse de sensibilité.

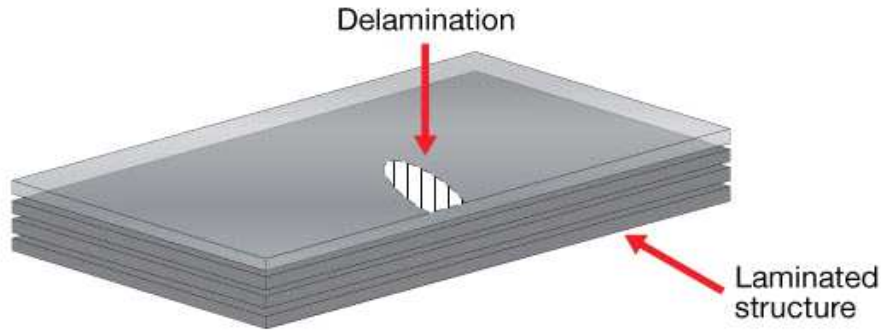


FIG. 2.6 – Délamination dans les structures sandwich

Cette section a pour objectif d'étudier la sensibilité du matériau à une variation de l'épaisseur d'une des couches, ou en présence d'un défaut pouvant engendrer une perturbation sur la matrice de rigidité. Le problème étudié est décrit dans le paragraphe 2.4.1.

2.4.1 Sollicitation statique d'une poutre Verre/PVB/Verre

Considérons une poutre sandwich en verre feuilleté tricouche (Verre/PVB/Verre) (figure 2.7) constitué de deux couches de verre assemblées entre elles par une couche viscoélastique de polyvinyl butyral (PVB) en sollicitation statique. Les propriétés mécaniques et géométriques sont présentées dans le tableau 2.4.1. La poutre est encastree des deux côtés. Une force concentrée $F = \lambda Q$ est appliquée au milieu de la poutre.

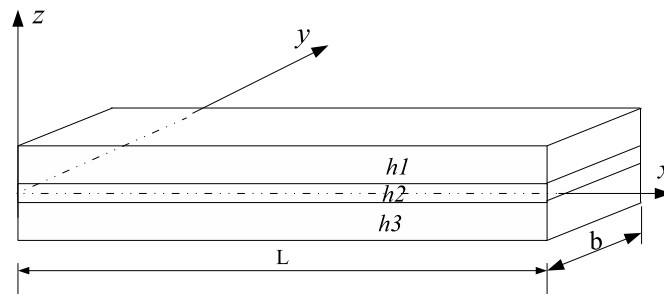


FIG. 2.7 – Poutre sandwich en verre laminé

Faces élastiques	Module de Young $E_f = 6.45^{10} Pa$
	Coefficient de Poisson $\nu_f = 0.22$
	Masse volumique $\rho_f = 2737 Kg.m^{-3}$
	Epaisseur $h_1 = h_3 = 5 \times 10^{-3} m$
Cœur viscoélastique	Module d'Young $E_c = 2.980 \times 10^6 Pa$
	Coefficient de Poisson $\nu_c = 0.4$
	Masse volumique $\rho_c = 999 Kg.m^{-3}$
	Epaisseur $h_2 = 0.76 \times 10^{-3} m$
Poutre	Longueur $L = 1 m$
	Largeur $b = 0.1 m$

TAB. 2.2 – Propriétés mécaniques et géométriques de la poutre sandwich

Ce paragraphe présente successivement des modèles cinématiques et la formulation du problème résiduel.

Modèles cinématiques

Le caractère amortissant apporté par la couche viscoélastique se traduit par un effet de cisaillement dans cette couche. Les premiers modèles cinématiques développés sont basés sur les approximations de Kirchhoff-Love [95] ou de Reissner-Mindlin [96]. L'hypothèse faite est une déformation de cisaillement négligeable par rapport aux autres déformations. Cette hypothèse revient à considérer une variation linéaire du champ de déplacement dans l'épaisseur du matériau sandwich qui est alors considéré comme mono-couche. Cette hypothèse conduit à la théorie classique des stratifiés qui n'est pas adaptée aux matériaux sandwich dont le cœur viscoélastique est considéré mou. D'autres modèles ont été proposés pour tenir compte du cisaillement dans la couche viscoélastique. Il existe des modèles linéaires et des modèles non-linéaires. Mindlin [97] a proposé un modèle permettant la variation linéaire du champs de déplacement dans l'épaisseur et la prise en compte d'une déformation de cisaillement dans le cœur. Reddy

[98] a proposé une description cubique du champ de déplacement dans l'épaisseur de la structure sandwich. Touratier [99] quant à lui propose une description sinusoïdale du champ déplacement. Ces deux modèles non-linéaires ont été utilisés dans plusieurs études sur les matériaux stratifiés [100, 101], et notamment les matériaux sandwich à cœur viscoélastique. Celles-ci ont montré l'efficacité de ces modèles pour l'estimation des propriétés mécaniques des sandwiches notamment la rigidité, la fréquence propre et l'amortissement. Deux approches peuvent être considérées. Une approche mono-couche dite non-zigzag, figure 2.9, considère le matériau sandwich comme un matériau mono-couche. Ce choix permet d'affecter aux trois couches la même description cinématique et d'assurer la continuité champ de déplacement aux interfaces. Pour améliorer et rendre plus précise la modélisation des sandwiches, une description du champ de déplacement pour chaque couche est plus intéressante. Les modèles utilisant cette approche sont appelés modèles zigzag, figure 2.8. Dans cette approche, la continuité du déplacement est imposée aux interfaces.

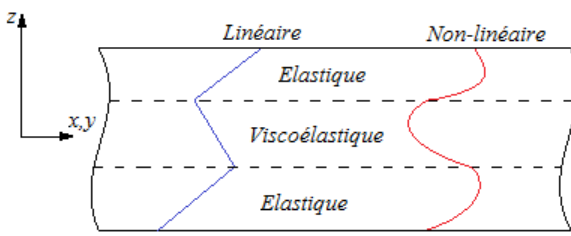


FIG. 2.8 – Modèle zig-zag

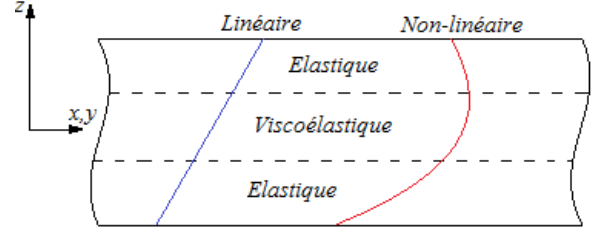


FIG. 2.9 – Modèle non zig-zag

Modèles non-zigzag Considérons par exemple une structure de type poutre sandwich à trois couches. Notons h_1 , h_2 , h_3 les épaisseurs des 3 couches, le cœur portant le numero 2 (figure 2.7). L'épaisseur totale est notée h . Le champ de déplacement du modèle non-zigzag peut être exprimé par la formule suivante :

$$U(x, z, t) = U^0(x, t) - z \frac{\partial w(x, t)}{\partial x} + f(z) \beta(x, t), \quad (2.10)$$

où U est le déplacement longitudinal, U^0 le déplacement de la ligne moyenne, β la rotation de la normale à la ligne moyenne, $w(x, t)$ est le déplacement transversal supposé le même suivant l'épaisseur et f la fonction de cisaillement qui définit le type d'approximation :

- $f(z) = 0$ le modèle d'Euler-Bernoulli [102] dans le cas des poutres et le modèle de Kirchhoff-Love [95] dans le cas des plaques ;
- $f(z) = z$ pour le modèle de Timoshenko [103] et le modèle de Mindlin [97] pour les plaques ;
- $f(z) = z - \frac{4z^3}{3h^2}$ pour le modèle de Reddy [98] ;
- $f(z) = \frac{h}{\pi} \sin\left(\frac{\pi z}{h}\right)$ pour le modèle de Touratier [99].

Modèles zigzag

Les modèles zigzag utilisent des cinématiques différentes pour le déplacement dans les faces et dans le cœur. Classiquement le modèle de Bernoulli (poutre) et le modèle Kirchhoff-Love (plaque) sont utilisés pour les faces. Un modèle adapté est utilisé pour prendre en compte le cisaillement. Dans le cas des poutres sandwich, le champ de déplacement est alors écrit pour chaque couche sous la forme suivante :

$$\begin{cases} U_1(x, z, t) = U_1^0(x, t) - (z - \frac{h_1 + h_2}{2}) \frac{\partial w(x, t)}{\partial x}, & \frac{h_2}{2} < z < \frac{h}{2} \\ U_2(x, z, t) = U_2^0(x, t) - z \frac{\partial w(x, t)}{\partial x} + \sum_{i=1}^n f_i(z) \beta_i(x, t), & -\frac{h_2}{2} < z < \frac{h_2}{2} \\ U_3(x, z, t) = U_3^0 - (z + \frac{h_1 + h_2}{2}) \frac{\partial w(x, t)}{\partial x}, & -\frac{h}{2} < z < -\frac{h_2}{2} \end{cases} \quad (2.11)$$

où U_i est le déplacement longitudinal dans la couche $i = 1, 2, 3$, U_i^0 est le déplacement de la ligne médiane de la couche i , les β_i sont les rotations additionnelles. Le cœur a pour indice 2. Les fonctions f_i sont telles que :

- $n = 1$, $f_1(z) = z$ pour le modèle de Rao [104] ;
- $n = 1$, $f_1(z) = z - \frac{4z^3}{3h^2}$ pour le modèle de Bickford [98] ;
- $n = 2$, $f_1(z) = z$, $f_2(z) = z^3$ pour le modèle mixte.

L'efficacité de ces modèles dans la description du comportement mécanique des structures sandwichs dépend des propriétés mécaniques et géométriques du matériau sandwich. Ainsi Hu et al [105] ont montré que le modèle zigzag de Rao est performant lorsque les épaisseurs et les modules d'Young des faces et du cœur vérifient respectivement $\frac{h_2}{h_1} \leq 10$ et $\frac{E_c}{E_f} \leq 0.01$. La poutre sandwich que nous considérons peut être étudiée à l'aide de ce modèle zigzag.

Formulation de l'équation résiduelle

Dans le cas statique, le problème est indépendant du temps. Le modèle zigzag de Rao est utilisé. La déformation de cisaillement est prise en compte dans la couche viscoélastique. En considérant la théorie classique des poutres d'Euler-Bernoulli [106, 4], le champ de déplacement peut être écrit sous la forme :

$$\begin{aligned} \overrightarrow{OM}(x, z) &= U_i = \begin{pmatrix} u_i(x) + (z - z_i)\beta_i(x) \\ W(x) \end{pmatrix}, \\ \beta_i &= \frac{\partial w}{\partial x}, \quad i = 1, 3, \\ \beta_2 &= \beta(x), \end{aligned} \tag{2.12}$$

où U_i est le vecteur déplacement de la couche i , u_i est le déplacement longitudinal dans le plan moyen des faces, z_i est la cote du plan moyen de la couche i , $\beta_2 = \beta(x)$ est une rotation additionnelle dans la couche viscoélastique qui permet de prendre en compte le cisaillement dans cette couche. Le déplacement transversal $W(x)$ est commun aux trois couches. Le repère considéré est tel que le plan (x, y) coïncide avec la plan médian de la couche viscoélastique. On a :

$$z_1 = \frac{h_1 + h_2}{2}, \quad z_2 = 0, \quad \text{et} \quad z_3 = \frac{h_3 + h_2}{2}. \tag{2.13}$$

La convention $\zeta_{,x} = \frac{\partial \zeta}{\partial x}$ et la condition de continuité du déplacement aux interfaces nous permettent d'écrire :

$$\begin{aligned} u_2 &= \frac{u_1 + u_3}{2} + \frac{h_1 - h_3}{4} w_{,x}, \\ \beta_2 &= \frac{u_1 - u_3}{h_2} + \frac{h_1 + h_3}{2h_2} w_{,x}, \end{aligned} \tag{2.14}$$

où $w_{,x} = \frac{\partial W}{\partial x}$. L'équation résiduelle est formulée en utilisant le principe des travaux virtuels :

$$\begin{aligned}\delta p_{int} + \delta p_{ext} &= 0, \\ \delta p_{int} &= \int_0^L \sum_{i=1}^3 \{\delta \Gamma_i\}^T \{S_i\} dx, \\ \delta p_{ext} &= \int_0^L \lambda q \delta W dx,\end{aligned}\tag{2.15}$$

où $\delta \Gamma$ représente la déformation virtuelle, S_i est le tenseur des contraintes défini par la relation :

$$S_i = [D_i] \{\Gamma_i\} = [D_i] \{\Gamma_i^l + \Gamma_i^{nl}\},\tag{2.16}$$

avec $\Gamma_i^l = \{u_{i,x}, \beta_{i,x}, W_{,x} + \beta_{,x}\}^T$ et $\Gamma_i^{nl} = \{\frac{1}{2}(w_{,x})^2, 0, 0\}^T$.

D'un point de vue numérique, le problème est discrétisé en utilisant des éléments poutres. En remplaçant les variables u_2 et β_2 par leur expression en fonction de u_1 , u_3 , $w_{,x}$ (2.14), chaque nœud a quatre degrés de liberté. Le vecteur nodal est égal à

$$v = \begin{Bmatrix} u_1 \\ u_3 \\ w \\ w_{,x} \end{Bmatrix}. \text{ L'équation résiduelle discrète est :}$$

$$\mathcal{R}(v, \lambda) = \sum_e \delta v^t \left(\int_{x_e} \sum_1^3 \left(B_i^l + B_i^{nl}(v) \right)^t S_i(v) dx - \lambda f_e \right) = 0,\tag{2.17}$$

avec

$$S_i = D_i \left(B_i^l + \frac{1}{2} B_i^{nl} v \right) v,\tag{2.18}$$

où v est le vecteur des déplacements nodaux et f_e est le vecteur des efforts nodaux. Dans l'équation (2.17), les variables B_i^l et B_i^{nl} sont respectivement la partie linéaire et la partie non-linéaire de la matrice d'interpolation des fonctions de forme. Ces fonctions de forme sont données dans [107]. La matrice de comportement mécanique D_i est donnée

par la formule :

$$D_i = \begin{bmatrix} E_i A_i & 0 & 0 \\ 0 & E_i I_i & 0 \\ 0 & 0 & \mu_i A_i \end{bmatrix}, \quad (2.19)$$

où E_i , μ_i , A_i et I_i sont respectivement le module de Young, le module de cisaillement, la section droite et le moment quadratique de la $i^{\text{ème}}$ couche.

Implémentation et différentiation

L'équation résiduelle $\mathcal{R}(v, \lambda)$ (2.17) est implémentée en Matlab, puis différenciée par rapport aux paramètres de modélisation par AdiMat. Le code est ensuite interfacé avec Diamant. L'ordre de troncature des séries de Taylor est $K = 20$, le paramètre ϵ qui intervient dans le calcul du rayon de convergence a_{max} (1.38) est pris égal à 10^{-6} .

Dans la figure 2.10, la déflexion W normalisée calculée avec le solveur Diamant est tracée et comparée à la solution obtenue dans [108, 102, 25] pour une épaisseur $h_2 = 0.76 \times 10^{-3}$.

2.4.2 Sensibilité par rapport à l'épaisseur

Nous nous sommes intéressés particulièrement à l'effet de la couche viscoélastique car elle peut favoriser les non-linéarités géométriques.

Test de Taylor

Tout d'abord, nous avons procédé à un test de Taylor pour vérifier l'exactitude du code linéaire tangent, conformément à la description faite dans le premier chapitre. Les résultats sont présentés dans le tableau 2.3. Pour ce faire, la perturbation ϵ_d est appliquée suivant l'équation :

$$h = h_0(1 + \epsilon_d); \quad (2.20)$$

où h_0 est l'épaisseur initiale

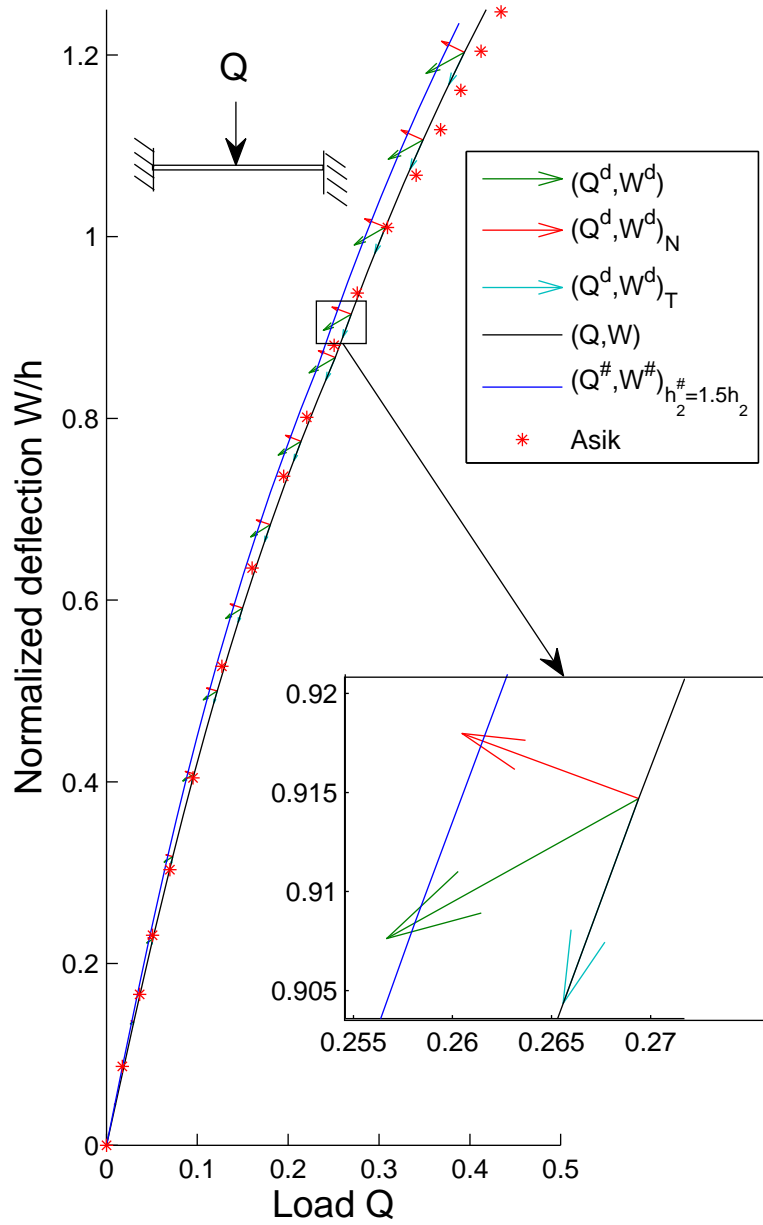


FIG. 2.10 – Sensibilité (Q^d, W^d) pour une perturbation de $0.5h_2$ la couche viscoélastique.

TAB. 2.3 – Test de Taylor pour une perturbation dans la direction h_2 de l'épaisseur de la couche viscoélastique.

α	$r_\alpha = \left(\frac{Q_{FD}^d}{Q_{AD}^d} \right)^h$	$r_\alpha = \left(\frac{W_{FD}^d}{W_{AD}^d} \right)^h$
1.00e-04	1.12747181414205e+00	1.12601796420455e+00
1.00e-05	1.01267409471703e+00	1.01253047198618e+00
1.00e-06	1.00126674295063e+00	1.00125240152212e+00
1.00e-07	1.00012666722343e+00	1.00012523528360e+00
1.00e-08	1.00001266101696e+00	1.00001254231613e+00
1.00e-09	1.00000131458209e+00	1.00000146450978e+00
1.00e-10	9.99997298080056e-01	9.99997648328058e-01
1.00e-11	9.99980260786093e-01	1.00002297228905e+00
1.00e-12	1.00007424037155e+00	1.00018373134196e+00
1.00e-13	1.00068548512531e+00	1.00234530587274e+00

Ce tableau présente les résultats du test de Taylor pour le paramètre de charge Q et la déflexion maximale $W = \max(w)$ au centre de la poutre. On observe que le ratio r_α tend linéairement vers 1 jusqu'à la valeur α égale à 10^{-9} qui est, pour cet exemple, le pas optimal pour l'approximation par la méthode des différences finies. Ce résultat montre que le code linéaire tangent est correct et peut être utilisé pour le calcul de sensibilité.

Résultats de sensibilité à l'épaisseur de la couche viscoélastique

La figure 2.11 trace les sensibilités de la poutre à une variation de l'épaisseur de la couche viscoélastique. La courbe (Q, W) (ligne solide noir) est la solution obtenue suite à la modélisation et à la résolution par Diamant [25]. Elle reproduit les résultats expérimentaux exposés dans [108] indiqués par une *.

En un point de la courbe, les sensibilités calculées (Q^d, W^d) sont décomposées en

une partie tangentielle $(Q^d, W^d)_T$ et une partie normale $(Q^d, W^d)_N$. Les flèches sur la figure 2.11 les représentent sous forme de vecteurs issus de la solution (Q, W) . Comme les sensibilités sont petites, elles sont présentées pour une perturbation $\delta h_2 = 0.5h_2$. Comme attendu, on observe une bonne concordance entre les pointes de flèches des vecteurs de sensibilité issus de la courbe et la solution perturbée $(Q^\#, W^\#)$ calculée à l'aide de Diamant pour un cœur d'épaisseur $h_2^\# = 1.5h_2$.

La sensibilité est d'ordre 1 et la perturbation de 50%, ce qui explique les écarts dans le zoom de la figure 2.11. Dans le zoom, les courbes sont tracées dans un repère orthonormé pour permettre une mesure visuelle des angles. La partie tangentielle indique la manière dont la perturbation va allonger ou raccourcir la branche solution calculée par la méthode de résolution MAN. La partie normale informe sur l'écart que cette perturbation crée entre la branche de solution "exacte" et la branche de solutions perturbée.

L'analyse de sensibilité par l'approche Diamant permet également de tracer la sensibilité de la déflexion 2.11. La figure de gauche compare la solution initiale avec la solution perturbée comme souvent pratiqué en mécanique pour l'analyse de sensibilité. On n'observe pas de différences majeures, cette approche comparative est limitée. La figure de droite montre l'évolution de la sensibilité de la déflexion avec un niveau de détails qui ne peut être obtenu ni par comparaison de courbes, ni par une approximation par la méthode des différences finies.

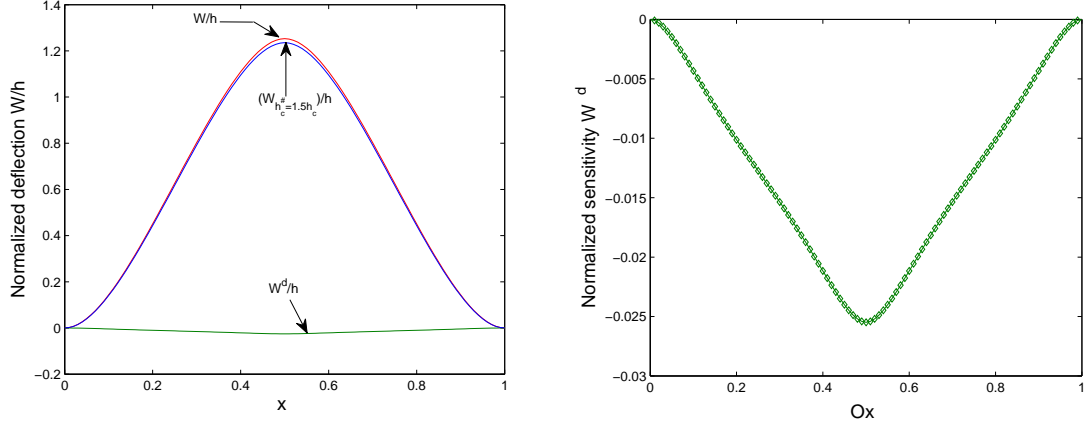


FIG. 2.11 – Gauche : Déflexion normalisée. Droite : Sensibilité de la déflexion à une perturbation de 50% de l'épaisseur de la couche viscoélastique.

2.4.3 Sensibilité par rapport au module de Young de la couche viscoélastique

Dans le code, une perturbation ϵ est ajoutée au module Young de la couche viscoélastique E_c à l'aide de l'instruction :

$$Ec = Ec * (1 + \epsilon); \quad (2.21)$$

pour mesurer son influence sur la solution. L'instruction linéarisée est

$$Ecd = Ec * (\epsilon) + Ecd * (1 + \epsilon); \quad (2.22)$$

Le code linéaire tangent est évalué avec $(Ec, \epsilon, Ecd, \epsilon) = (2.98e+6, 0, 0, 1)$. Il suffit alors de multiplier le résultat du code linéaire tangent par la valeur de la perturbation ϵ pour obtenir les sensibilités correspondant à ϵ .

Les vecteurs sensibilité (Q^d, W^d) et leur décomposition tangentielle/normale sont tracés sur la courbe solution (Q, W) dans la figure (2.12) pour un $\epsilon = 0.5$. Comme pour le cas précédent, ces résultats sont comparés à la solution perturbée $(\lambda^\#, W^\#)$ calculée avec un module de Young $E_c^\# = 1.5E_c$ par le solveur Diamant. On remarque que l'écart entre la tête de la flèche normale des vecteurs de sensibilité et la solution

perturbée est très faible : les dérivées de premier ordre fournissent à nouveau une bonne approximation de la solution perturbée.

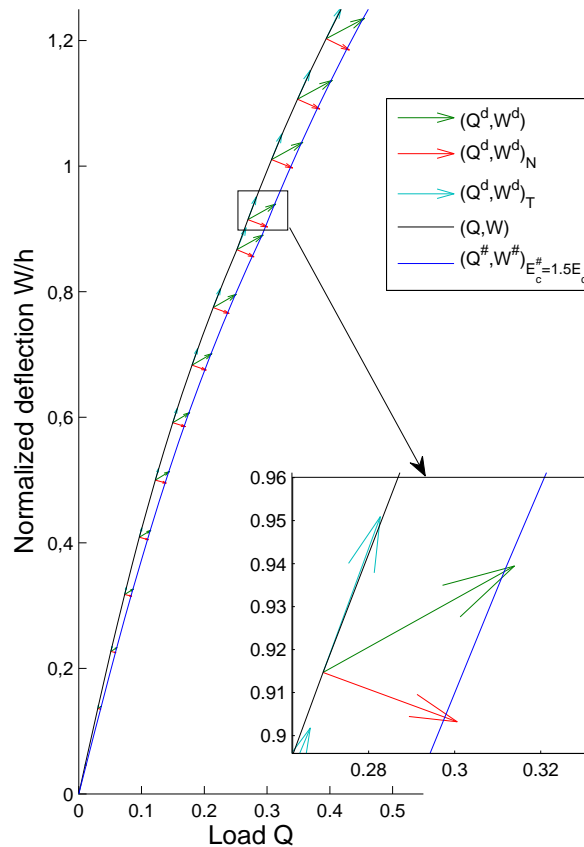


FIG. 2.12 – Sensibilité (Q^d, W^d) pour une perturbation de $\epsilon = 0.5$ sur le module de Young de la couche viscoélastique.

On note aussi, comme prévu, qu'une perturbation positive sur l'épaisseur de cœur viscoélastique agit comme une perturbation négative sur le module de Young. En d'autres termes, une perturbation positive sur l'épaisseur du cœur assouplit la poutre.

2.4.4 Sensibilité par rapport aux éléments de la matrice de rigidité

Les matériaux composites peuvent présenter des imperfections localisées dans les zones de collage pauvre aux interfaces cœur/face imparfaites [20], ou résulter d'un

alignement imparfait pour les composites à fibre [109]. Dans ces différents cas, le calcul de sensibilité peut produire des informations quantitative et qualitative [20, 109].

Du point de vue numérique, la présence d'hétérogénéités dans le matériau est prise en compte par une perturbation locale de la matrice de rigidité. Pour étudier l'influence de cette perturbation, nous avons implémenté le défaut δ comme une perturbation de certains éléments de la matrice de rigidité élémentaire :

$$K^e(\delta) = (1 + \delta)K_0^e, \quad (2.23)$$

où K_0^e est la matrice de rigidité élémentaire non perturbée. Cette modification de la matrice élémentaire est introduite dans le problème résiduel.

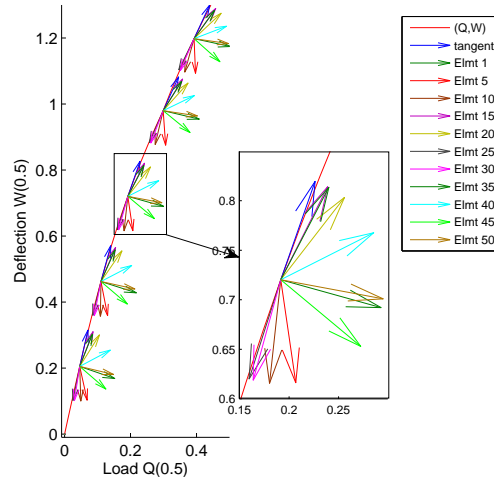


FIG. 2.13 – Vecteurs de sensibilité normalisés tracés sur la courbe de réponse (Q, W) pour une perturbation de 10% de la matrice de rigidité élémentaire.

Le code informatique implémentant le problème résiduel est alors différentié par rapport à δ par l'outil Adimat et évalué avec $\delta = \delta^d = 0$ pour tous les éléments, et $\delta^d = 1$ pour l'élément que l'on veut perturbé. Les calculs ont été faits pour les éléments de numéros $5i$ ($i = 1, 2, \dots, 10$). Pour une meilleure visualisation, nous avons décidé de tracer des vecteurs normalisés S_{5i}^V ($i = 1, 2, \dots, 10$) construits tels que :

$$S_{5i}^v = \left\{ \frac{(Q^d, W^d)_{5i}}{\|(Q^d, W^d)_{5i}\|} \right\} \quad (2.24)$$

Dans la figure 2.13, on observe que l'orientation du vecteur de sensibilité dépend de la position de l'élément perturbé.

Les vecteurs de sensibilité S_{5i}^V présentés sur la figure 2.13 sont aussi décomposés en une partie tangentielle et une partie normale. La norme de la partie normale est présentée sur la figure 2.14. On remarque une discontinuité à la fin de la première

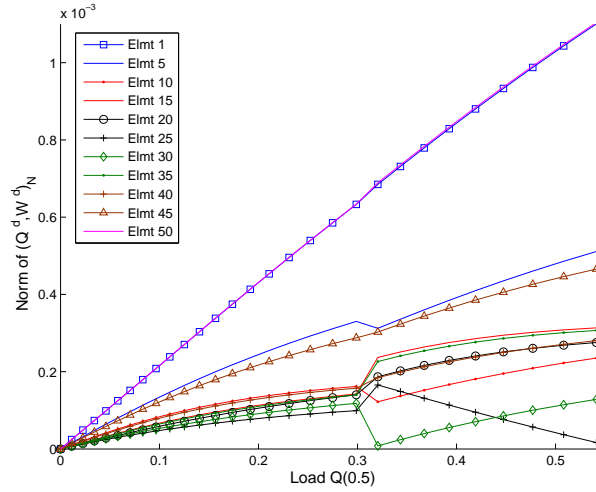


FIG. 2.14 – Norme de la partie normale des vecteurs de sensibilité pour une perturbation de 10 % de la matrice de rigidité élémentaire

branche de la MAN. C'est un résultat attendu car la MAN assure la continuité de la solution, pas celle de la sensibilité. Cette norme permet, selon la position du défaut et le chargement λ , de quantifier l'influence de la perturbation sur le comportement mécanique de la poutre.

La figure 2.15 présente la sensibilité de la déflexion. Les courbes de gauche sont relatives à un défaut localisé dans le milieu de la poutre (pour $x \in [0.35, 0.65]$). La courbe de sensibilité y apparaît comme la réponse de la structure à une faible sollicitation. Cet effet est moins visible sur les courbes de la figure de droite où l'amplitude des sensibilités dépend de la position de l'élément perturbé par rapport au bord encastré de la poutre. Ce phénomène peut être expliqué par l'effet de bord qui atténue l'amplitude de la sensibilité.

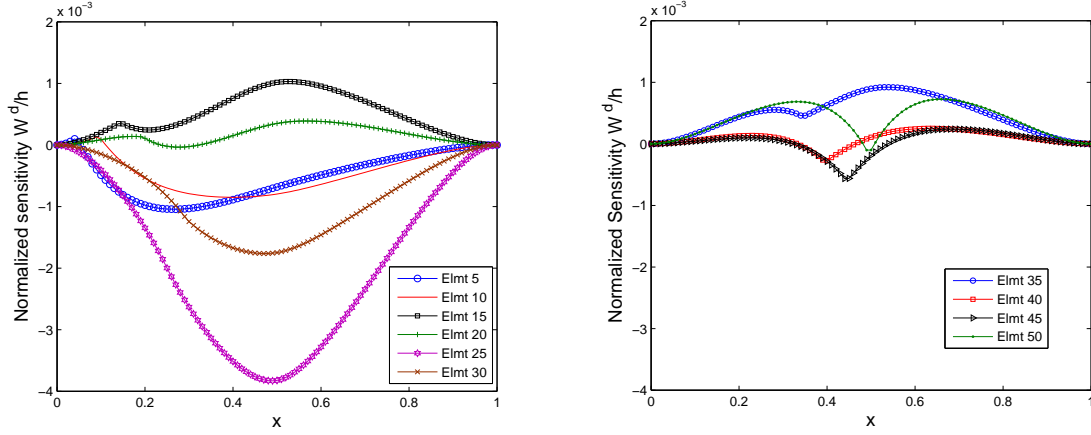


FIG. 2.15 – Sensibilités de la déflexion par rapport aux perturbations de la matrice de rigidité élémentaire.

2.5 Conclusions

Les études de sensibilité sont classiques en mécanique comme présenté dans le paragraphe 2.1. Elles peuvent être déduites de données expérimentales ou de simulations numériques réalisées en faisant varier les paramètres d'entrée, puis en traçant des courbes pour comparaison visuelle ou bien encore, dans le cas numérique, en calculant des sensibilités approchées au moyen de la méthode des différences finies. D'un côté, les comparaisons visuelles ne permettent pas de rendre compte de manière fine des variations survenant d'une expérience à l'autre car, dans le cas de perturbations faibles, les courbes peuvent se confondre. De l'autre côté, la méthode des différences finies reste une méthode approchée et difficile à mettre en œuvre lorsqu'on résout des problèmes non-linéaires par une méthode de continuation.

Dans ce chapitre et dans l'article [15], nous avons développé une méthode d'analyse de sensibilité pour les solutions de problèmes non-linéaires en exploitant le cadre de travail offert par Diamant. La sensibilité des solutions de la MAN par rapport aux paramètres de modélisation demande de différentier les équations du modèle par rapport à ces paramètres, mais aussi par rapport au paramètre de chemin. Pour le faire de manière efficace, nous avons dérivé la formule de base de l'approche Diamant afin

de construire le système d'équations permettant de calculer les sensibilités. Celui-ci présente le même Jacobien que le système original, et des termes de haut-degré qui peuvent être calculés par DA. Nous les avons calculés automatiquement en couplant Adimat avec Diamant pour calculer les dérivées mixtes.

Pour prouver l'intérêt de nos travaux, nous avons étudié une poutre sandwich visco-élastique dans le cadre des non-linéarités géométriques (cas statique). La modélisation de cette poutre est réalisée à l'aide de la méthode des éléments finis et d'un modèle de type zigzag. La sensibilité des solutions est étudiée par rapport à des perturbation de l'épaisseur et du module d'Young du cœur viscoélastique, mais aussi par rapport à des perturbations d'éléments de la matrice de rigidité pour simuler un défaut dans le matériau. Les résultats obtenus sont originaux et très précis. Ils montrent la pertinence de notre approche et de nos calculs, et notamment leurs capacités à fournir des informations complémentaires.

En conclusion, l'analyse de sensibilité, si elle bien est faite, fournit des informations qualitatives et quantitatives de premier ordre sur le comportement mécanique des structures. Elle offre aussi un accès aux gradients qui peuvent être utilisés dans le cadre de problèmes d'identification et d'optimisation. Nous avons montré que ces informations précieuses peuvent être obtenues avec un coût de développement faible en appliquant des outils de différentiation automatique au code informatique concerné.

Chapitre 3

Analyse de sensibilité des modes propres des structures sandwich viscoélastiques

3.1 Introduction

Au chapitre 2, nous avons montré que la DA permet d’aborder de manière efficace l’analyse de sensibilité des problèmes mécaniques en statique. Nous nous intéressons à présent au problème des vibrations libres des structures sandwich viscoélastiques dont la résolution permet de caractériser les propriétés amortissantes de ces structures [17, 16, 110]. Dans les structures mécaniques, les vibrations sont généralement indésirables car elles sont à l’origine de phénomènes tels que la fatigue et les nuisances sonores. La connaissance des fréquences propres permet de dimensionner les structures par rapport aux sollicitations et d’éviter le phénomène de résonance. Pour atténuer ou éliminer les vibrations dans les structures, la technique la plus souvent employée est la dissipation de l’énergie en incorporant aux structures des matériaux viscoélastiques dont le pouvoir amortissant est bien connu.

Les modes propres et les fréquences propres peuvent être obtenus par des approches

analytiques, expérimentales, ou numériques. La détermination des facteurs de perte qui caractérisent l'amortissement de ces structures est aussi importante. Un grand nombre d'études s'intéressent aux vibrations dans les structures sandwich [111, 112, 60, 17, 16, 110, 1, 2, 3, 4]. Toutes ces études ont proposé des méthodologies pour faciliter la modélisation de ces structures qui doivent en prenant en compte des lois viscoélastiques non-linéaires, des non-linéarités géométriques et des cinématiques complexes. Les travaux d'Asik [108] ont abordé le problème de la non-linéarité géométrique sans toutefois tenir en compte la dépendance en fréquence de la loi de comportement. Des travaux [16, 113, 110] ont montré que la MAN est capable de résoudre les problèmes aux valeurs propres non-linéaires dépendant de la fréquence. La version Diamant de cet algorithme de continuation a été implémentée ultérieurement dans [88].

En ingénierie des structures, pour assurer de bonnes propriétés dans le domaine de vibration, il devient important d'estimer avec précision le comportement dynamique de ces structures lorsqu'elles sont soumises à des perturbations. L'analyse de sensibilité des structures sandwich a intéressé plusieurs auteurs. Lima et al. [8] proposent une approche d'analyse de sensibilité basée sur une dérivation analytique. Elle nécessite pour chaque loi de comportement une dérivation manuelle et une reformulation du problème, ce qui en limite l'intérêt numérique eu égard à la complexité des lois viscoélastiques. Lin et al. [114] ont également proposé une approche d'analyse de sensibilité des structures amorties basée sur une dérivation analytique des équations effectuée en supposant que les modes propres dérivés sont une combinaison linéaire des modes propres. D'autres méthodes numériques basés sur la méthode de perturbation sont proposées [29, 30, 31]. Cependant ces méthodes ne traitent que le cas des problèmes aux valeurs propres linéaires.

Dans ce chapitre, nous proposons une approche d'analyse de sensibilité des problèmes aux valeurs propres non-linéaires basée sur la DA. Après avoir présenté un état de l'art des méthodes d'analyse de sensibilité des problèmes aux valeurs propres, ce chapitre expose notre contribution, puis étudie la sensibilité des structures sandwich

viscoélastiques modélisées à l'aide de divers modèles viscoélastiques. La sensibilité des fréquences propres, des modes propres et des facteurs de perte à des perturbations géométriques et matérielles est traitée.

3.2 Analyse de sensibilité des modes propres et fréquences propres de vibration

L'analyse de sensibilité des modes propres de vibration permet de caractériser le comportement dynamique des structures lorsqu'elles sont soumises à des perturbations [29]. Le problème linéaire de vibration libre peut être écrit sous la forme :

$$[K - \omega^2 M]u = 0, \quad (3.1)$$

où K et M sont respectivement la matrice de rigidité et la matrice de masse ; ω et U sont la fréquence propre et le vecteur propre. Plusieurs méthodes sont proposées dans la littérature pour l'étude de sensibilité des solutions de ce problème linéaire [115, 8, 29]. La plupart de ces méthodes ne permettent pas de traiter la sensibilité des problèmes aux valeurs propres non-linéaires écrits sous la forme :

$$[K(\omega) - \omega^2 M]u = 0, \quad (3.2)$$

représentant le problème des vibrations des structures viscoélastiques dont la loi de comportement est une fonction complexe dépendant de la fréquence. Dans cette section, nous présentons un état de l'art sur l'étude de sensibilité des modes propres et des fréquences propres des structures viscoélastiques (3.1).

3.2.1 Analyse de sensibilité des problèmes aux valeurs propres linéaires

Dans le cas du problème à valeur propre linéaire (3.1), la méthode de Lanczos [116] et la méthode des sous espaces [117] permettent de déterminer les solutions de

l'équation (3.1) avec la condition d'orthogonalité des modes propres par rapport à la matrice de rigidité.

$${}^t u^j K u^i = 0, \quad (3.3)$$

ou en normalisant les modes propres par rapport à la matrice de masse.

$$u^i M u^i = 1. \quad (3.4)$$

L'étude de sensibilité des structures en dynamique permet de caractériser le comportement dynamique des structures lorsque les paramètres de modélisation sont modifiés ou soumis à une perturbation. Elle fait partie des méthodes SDM [115] (Structural dynamic modification) qui regroupent l'ensemble des méthodes permettant de caractériser le comportement dynamique des structures lorsque les paramètres de modélisation sont modifiés.

Dans la littérature, deux démarches sont présentées :

1. la sensibilité des modes propres et des fréquences propres est effectuée par une étude de la réponse fréquentielle de la structure [8] ;
2. la sensibilité des modes propres et des fréquences propres est étudiée directement [30, 31, 10].

Ces deux approches peuvent être mises en œuvre soit par une méthode de perturbation, soit par une méthode de dérivation analytique. Nous présentons dans ce qui suit une revue bibliographique des méthodes utilisés pour étudier la sensibilité des problèmes aux valeurs propres linéaire.

3.2.2 Méthodes de perturbation

Pour réduire les temps de calcul, plusieurs auteurs ont proposé d'utiliser des méthodes de perturbations.

Méthode de perturbation matricielle

Proposée par Chen [118, 30], cette approche est assez facile à mettre en œuvre et très utilisée pour déterminer les solutions propres d'une structure dont les paramètres de modélisation modifiés affectent la matrice de masse et la matrice de rigidité. Les solutions perturbées sont déterminées à partir de la solution non perturbée. Pour illustrer cette méthode, nous considérons le problème générique (3.1) dont on suppose que $\lambda_0 = \omega_0^2$ et u_0 sont solutions. On applique à la matrice de rigidité et à la matrice de masse une perturbation de la forme :

$$\begin{aligned}\bar{K} &= K + \varepsilon \Delta K, \\ \bar{M} &= M + \varepsilon \Delta M,\end{aligned}\tag{3.5}$$

où $\varepsilon = \max\{\Delta\alpha_1, \Delta\alpha_2, \dots, \Delta\alpha_n\}$. Les $\Delta\alpha_i$ sont les paramètres de modélisation modifiés, et \bar{K} et \bar{M} sont respectivement les matrices de rigidité et la matrice de masse perturbées. Le problème linéaire aux valeurs propres à résoudre est alors écrit sous la forme :

$$[\bar{K} - \bar{\lambda}^i \bar{M}] \bar{u}^i = 0.\tag{3.6}$$

Les solutions de ce problème perturbé (3.6) sont cherchées sous la forme de séries :

$$\begin{cases} \bar{\lambda}^i = \lambda_0^i + \varepsilon \lambda_1^i + \varepsilon^2 \lambda_2^i + \dots + \varepsilon^n \lambda_n^i, \\ \bar{u}^i = u_0^i + \varepsilon u_1^i + \varepsilon^2 u_2^i + \dots + \varepsilon^n u_n^i. \end{cases}\tag{3.7}$$

Les termes des séries λ_k^i et u_k^i peuvent être calculés en introduisant (3.7) dans (3.5). Par exemple, les termes sont :

à l'ordre 1,

$$\begin{aligned}\lambda_1^i &= ({}^t u_0)^i \Delta K u_0^i - \lambda_0^i ({}^t u_0^i) \Delta M u_0^i, \\ u_1^i &= \sum_{j=1, j \neq i}^n \frac{1}{\lambda_0^i - \lambda_0^j} \left[({}^t u_0^j) \Delta K u_0^i - \lambda_0^i ({}^t u_0^j) \Delta M u_0^i \right] u_0^j - \frac{1}{2} \left[({}^t u_0^i) \Delta M u_0^i \right] u_0^i,\end{aligned}\tag{3.8}$$

à l'ordre 2,

$$\begin{aligned}\lambda_2^i &= ({}^t u_0^i) \Delta K u_1^i - \lambda_0^i ({}^t u_0^i) \Delta M u_1^i - \lambda_1^i ({}^t u_0^i) M_0 u_1^i - \lambda_1^i ({}^t u_0^i) \Delta M u_0^i, \\ u_2^i &= \sum_{j=1, j \neq i}^n \frac{1}{\lambda_0^i - \lambda_0^j} \left[({}^t u_0^j) \Delta K u_1^i - \lambda_0^i ({}^t u_0^j) \Delta M u_1^i - \lambda_1^i ({}^t u_0^j) M_0 u_1^i - \lambda_1^i ({}^t u_0^j) \Delta M u_0^i \right] u_0^j.\end{aligned}$$

(3.9)

Ces formules, issues du développement en séries d'opérations multiplicatives peuvent être fastidieuses à obtenir par un développement manuel, ce qui limite en pratique l'usage de la méthode aux premiers ordres. On remarquera que le calcul des dérivées d'ordre plus élevé pourrait être effectué par différentiation automatique. Par ailleurs, cette méthode ne prend en compte que la perturbation maximale alors que, dans la réalité, l'influence d'un paramètre n'est pas forcément proportionnelle à sa valeur.

Méthode de perturbation matricielle optimisée

Pour améliorer la précision de l'approximation de premier ordre de la matrice de masse et de la matrice de rigidité, Bickford [31] propose une méthode basée sur le calcul de coefficient de Rayleigh. Contrairement à la méthode de perturbation matricielle proposée dans [118], Bickford utilise une approximation de premier ordre pour calculer les solutions du système modifié sous la forme :

$$\begin{aligned}\lambda_w^i &= \lambda_0^i + \Delta\lambda_w^i, \\ u_w^i &= u_0^i + \Delta u_w^i,\end{aligned}\tag{3.10}$$

où (λ_w^i, u_w^i) est la $i^{\text{ème}}$ solution propre du système modifié. Pour obtenir les approximations de premier ordre, on divise l'équation (3.8) par le terme $({}^t u_0^i) \overline{M} u_0^i$, pour en déduire les approximations du premier ordre suivant l'équation :

$$\begin{aligned}\Delta\lambda_w^i &= \frac{\lambda_1^i}{({}^t u_0^i) \overline{M} u_0^i}, \\ \Delta u_w^i &= \sum_{j=1}^n c_{ij}^w u_0^j,\end{aligned}\tag{3.11}$$

où les coefficients c_{ij}^w sont définis par :

$$\begin{aligned}c_{ij}^w &= \frac{({}^t u_0^i) \Delta K u_0^i - \lambda_0^i ({}^t u_0^i) \Delta M u_0^i}{(\lambda_0^i - \lambda_0^j) ({}^t u_0^i) \overline{M} u_0^i}, \quad i \neq j, \\ c_{ii}^w &= -\frac{({}^t u_0^i) \Delta M u_0^i}{2({}^t u_0^i) \overline{M} u_0^i}.\end{aligned}\tag{3.12}$$

En utilisant l'équation (3.6), Bickford a montré que la $i^{\text{ème}}$ valeur propre du système perturbé peut être directement calculée sous la forme du quotient de Rayleigh :

$$\lambda_w^i = \frac{({}^t u_w^i) \overline{K} u_w^i}{({}^t u_w^i) \overline{M} u_w^i}. \quad (3.13)$$

Cette méthode limitée à l'ordre 1 est peu précise. Elle a été améliorée dans [30] en utilisant les formules suivantes :

$$\lambda_c^i = \frac{({}^t u_c^i)(K + \varepsilon \Delta K u_c^i)}{({}^t u_c^i)(M + \varepsilon \Delta M u_c^i)}, \quad (3.14)$$

$$\begin{aligned} c_{ij}^c &= \frac{({}^t u_0^i)(\varepsilon \Delta K) u_1^i - \lambda_0({}^t u_0^i)(\varepsilon \Delta M u_1^i - \Delta \lambda_1^i({}^t u_0^i) \overline{M} u_1^i)}{(\lambda_0^i - \lambda_0^j)} & i \neq j, \\ c_{ii}^c &= \frac{({}^t u_1^i)(\varepsilon \Delta M) u_1^i + ({}^t u_1^i) M_0 u_1^i}{2}, & i = j, \end{aligned} \quad (3.15)$$

à la place des formules (3.12), (3.13). Une méthode similaire a été utilisée dans [115] pour le calcul de sensibilité en vibration libre d'une structure poutre sandwich contenant une couche viscoélastique à module complexe constant.

3.2.3 Méthodes hybrides

Les calculs numériques ont montré que les méthodes de Chen [118, 30] et de Bickford [31] basées sur des approximations d'ordre faible sont efficaces pour prédire les solutions propres dans le cas de perturbations faibles. Pour des perturbations élevées, des méthodes dites hybrides permet d'avoir des résultats plus précis. Ces méthodes hybrides associent à la méthode de perturbation une méthode supplémentaire. Par exemple dans le cas de l'approximation de Padé [119], la décomposition en fraction rationnelle est associé à la méthode de perturbation. La méthode de Kirsch [120], quant à elle utilise une combinaison linéaire des termes de la série calculée par la méthode de perturbation.

Méthode d'approximation de Padé

Proposée dans [119, 121, 122], elle combine le développement en série et le développement en fraction rationnelle. L'approximation de Padé est utilisée pour mieux

restructurer les vecteurs de base qui sont utilisés pour l'approximation de la valeur propre à partir du quotient de Rayleigh. Cela permet d'augmenter le rayon de convergence des séries. La méthode de perturbation approche les Δu et $\Delta \lambda$ sous la forme de séries :

$$\begin{aligned}\Delta u^i &= \varepsilon u_1 + \varepsilon^2 u_2 + \varepsilon^3 u_3 + \dots, \\ \Delta \lambda^i &= \varepsilon \lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3 + \dots\end{aligned}\tag{3.16}$$

Ici cette approximation en séries est remplacée par une approximation en fractions rationnelles (approximation de Padé), le développement en série du vecteur u est :

$$u = \varepsilon V_1 + \varepsilon^2 V_2 + \varepsilon^3 V_3 + \dots,\tag{3.17}$$

où les vecteurs V_p sont définis à partir d'une base orthogonale de Schmidt telle que :

$$V_p = \sum_{k=1}^p \alpha_p^k V'_k,\tag{3.18}$$

avec

$$\begin{aligned}\alpha_p^k &= \frac{\langle V'_p, V'_k \rangle}{\langle V'_p, V'_p \rangle}, \quad p \neq k, \\ \alpha_p^p &= 1,\end{aligned}\tag{3.19}$$

où V'_k représente un vecteur de base et $\langle \cdot, \cdot \rangle$ est le produit scalaire. En utilisant (3.18) dans (3.17), le vecteur u peut être écrit sous la forme

$$u = \sum_{j=1}^{\infty} \varepsilon^j f_j(\varepsilon) V'_j, \quad \text{avec} \quad f_j(\varepsilon) = \sum_{i=j}^{\infty} \alpha_i^j \varepsilon^{i-j}.\tag{3.20}$$

En remplaçant $f_j(\varepsilon)$ par son approximation de Padé $P_j \left[\frac{L_j}{M_j} \right] (\varepsilon)$, où L_j et M_j sont les ordres des polynômes au numérateur et au dénominateur de la fraction rationnelle, le vecteur u peut être écrit sous la forme :

$$u(\varepsilon) = \sum_{j=1}^{\infty} \varepsilon^j P_j \left[\frac{L_j}{M_j} \right] (\varepsilon) V'_j.\tag{3.21}$$

Ces développements sont appliqués pour calculer Δu . Puis, la $i^{\text{ème}}$ valeur propre du système perturbé est calculée en utilisant le quotient de Rayleigh :

$$\lambda_p^i = \frac{({}^t(u_0^i) + \Delta u^i) \overline{K}(u_0^i + \Delta u^i)}{({}^t(u_0^i) + \Delta u^i) \overline{M}(u_0^i + \Delta u^i)}.\tag{3.22}$$

Méthode de Kirsch étendue

La méthode de Kirsch est initialement développée pour étudier les problèmes statiques [120, 123], cependant elle peut être facilement étendue aux cas des problèmes dynamiques comme par exemple les problèmes aux valeurs propres. Dans cette méthode, une perturbation de second ordre est utilisée pour déterminer les modes propres et les fréquences propres d'une structure dont les paramètres de modélisation sont modifiés. Les vecteurs propres sont cherchés sous la forme :

$$u^i = y_1 u_0^i + y_2 (\varepsilon u_1^i) + y_3 (\varepsilon^2 u_2^i) = u_B^i y^i, \quad (3.23)$$

où la matrice u_B^i et le vecteur y^i sont donnés par

$$u_B^i = \{u_0^i, \varepsilon u_1^i, \varepsilon^2 u_2^i\}, \quad \text{et} \quad ({}^t y^i) = \{y_1, y_2, y_3\}. \quad (3.24)$$

Les variables εu_1^i et $\varepsilon^2 u_2^i$ sont respectivement les perturbations de premier ordre et de second ordre obtenues par les équations (3.8) et (3.9).

En remplaçant (3.24) dans le problème perturbé (3.6) et en multipliant par $({}^t u^i)$, on obtient :

$$({}^t u_B)(K_0 + \Delta K)u_B^i y^i = \lambda^i ({}^t u_B^i)(M_0 + \varepsilon \Delta M)u_B^i y^i. \quad (3.25)$$

En introduisant les notations :

$$\begin{aligned} K_R^i &= ({}^t u_B)(K_0 + \varepsilon \Delta K)u_B^i, \\ M_R^i &= ({}^t u_B^i)(M_0 + \varepsilon \Delta M)u_B^i, \end{aligned} \quad (3.26)$$

le vecteur propre peut être déterminé en résolvant l'équation

$$K_R^i y^i = \lambda^i M_R^i y^i. \quad (3.27)$$

La valeur propre λ^i est calculée en utilisant le quotient de Rayleigh (3.22). Dans cette approche, seule l'approximation de premier ordre est utilisée pour calculer le vecteur propre u^i car l'expérience a montré qu'elle donne des résultats plus précis que les autres. Cette approche a été étendue dans [124] pour prendre en compte les problèmes aux valeurs propres linéaires complexes.

Méthode par homotopie

Rappel de la technique d'homotopie – Depuis les travaux des pionniers [125, 126, 127], une bibliographie conséquente explique en détail la relation entre l'homotopie et les méthodes de continuation [128]. Dans les méthodes de continuation [27], l'homotopie peut être aussi utilisée pour déterminer une solution initiale par exemple. Pour décrire l'homotopie, nous considérons le problème

$$h(x) = g(x) + f(x) = 0, \quad (3.28)$$

qui est supposé difficile à résoudre. On suppose que le problème $g(x) = 0$ admet une solution connue. L'idée de base de l'homotopie est de voir le problème (3.28) comme un problème paramétrique écrit sous la forme :

$$H(x, a) = g(x) + af(x), \quad \text{avec} \quad a \in [0, 1], \quad (3.29)$$

où a est le paramètre d'homotopie. Cela permet de déterminer la solution de $H(x, 1) = g(x) + f(x) = 0$ à partir de la solution connue pour $H(x, 0) = g(x) = 0$. La figure

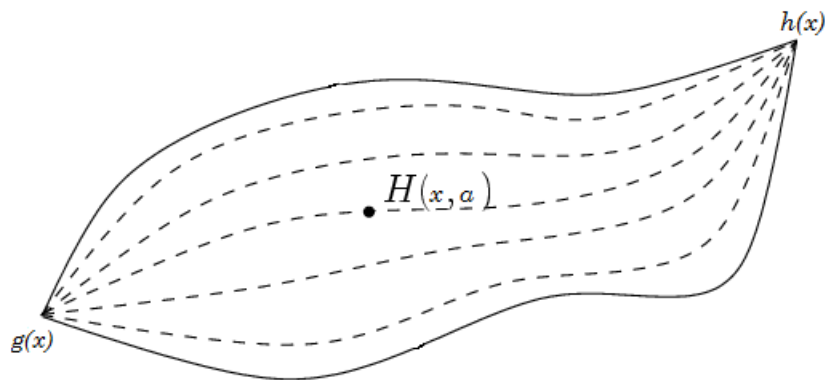


FIG. 3.1 – Courbes homotopiques de la fonction g à la fonction h .

3.1 présente un exemple de courbes homotopiques allant de la fonction g à la fonction f . L'homotopie est souvent associée à la méthode de continuation. Nous détaillerons l'usage de l'homotopie pour les problèmes de type (3.52) dans le paragraphe 3.3.1.

La méthode – La méthode d’analyse de sensibilité des solutions propres linéaires par homotopie [10] utilise la technique de perturbation et le développement en série. A la différence des méthodes présentées plus haut, elle n’est pas associée au calcul du quotient de Rayleigh. Cette technique est utilisée par Sliva [10] pour l’étude de sensibilité du problème aux valeurs propres linéaire suivant :

$$\begin{cases} [K_0 - \lambda_0^i M_0] u_0^i = 0, \\ ({}^t u_0^i) M_0 u_0^i = 1, \end{cases} \quad (3.30)$$

où M_0 et K_0 sont respectivement la matrice de masse et la matrice de rigidité de la structure initiale dont les vecteurs propres et les fréquences propres sont u_0 et λ_0 . Les perturbations de la structure sont exprimées sous la forme :

$$E = E_0(1 + \Delta_E), \quad \rho = \rho_0(1 + \Delta_\rho), \quad h = h_0(1 + \Delta_h). \quad (3.31)$$

Avec ces perturbations, la structure initiale et la structure perturbée sont liées par les relations

$$\begin{cases} K = K_0 + a\delta K_0, \\ M = M_0 + a\delta M_0, \end{cases} \quad (3.32)$$

où M et K sont la matrice de masse et la matrice de rigidité de la structure perturbée, et a le paramètre d’homotopie. On considère que ces matrices sont soumises à la même perturbation, ce qui suppose que les perturbations Δ_E , Δ_ρ et Δ_h soient égales. Le problème à résoudre s’écrit alors :

$$\begin{cases} [K - \lambda^i M] u^i = 0, \\ ({}^t u^i) M u^i = 1, \end{cases} \quad (3.33)$$

En utilisant (3.32) dans (3.33), le problème d’homotopie s’écrit :

$$\begin{cases} [K_0 + a(K - K_0)] v^i - \beta^i [M_0 + a(M - M_0)] v^i = 0, \\ (1 - a)({}^t v^i) M_0 v^i + a({}^t v^i) M v^i = 1. \end{cases} \quad (3.34)$$

A nouveau, on part de la solution initiale correspondant à $a = 0$ pour calculer déterminer la solution du système perturbé correspondant à $a = 1$. Pour ce faire, les inconnues

que sont la fréquence propre β^i et le mode propre v^i , sont cherchés sous forme de séries :

$$\begin{cases} \beta^i = \lambda_0 + \sum_{p=1}^P a^p \beta_p^i, \\ v^i = u_0 + \sum_{p=1}^P a^p v_p^i. \end{cases} \quad (3.35)$$

En introduisant (3.35) dans (3.34), le système à résoudre s'écrit :

à l'ordre $p = 1$

$$\begin{cases} (K_0 - \lambda_0^i M_0) v_1^i = F_1^i, \\ F_1^i = \beta_1^i M_0 u_0^i - (K - K_0) u_0^i + \lambda_0^i (M - M_0) u_0^i, \\ 2({}^t u_0^i) M_0 v_1^i = 1 - ({}^t u_0^i) M u_0^i. \end{cases} \quad (3.36)$$

à l'ordre $p \geq 2$

$$\begin{cases} (K_0 - \lambda_0^i M_0) v_p^i = F_p^i, \\ F_p^i = \beta_p^i M_0 u_0^i - (K - K_0) v_{p-1}^i + \sum_{k=1}^{p-1} \beta_k^i M_0 v_{p-k}^i + \sum_{k=0}^{p-1} \beta_k^i (M - M_0) v_{p-1-k}^i, \\ 2({}^t u_0^i) M_0 v_p^i = F_p', \\ F_p' = \sum_{k=0}^{p-1} v_k^i (M - M_0) v_{p-1-k}^i - \sum_{k=1}^{p-1} ({}^t v_k^i) M_0 v_{p-k}^i. \end{cases} \quad (3.37)$$

Les inconnues β_p^i et v_p^i sont déterminées en résolvant les systèmes (3.36) et (3.37). Le coefficient de Taylor à l'ordre 1 est calculé en premier :

$$\beta_1 = ({}^t u_0^i) K u_0^i - \lambda_0^i ({}^t u_0^i) M u_0^i. \quad (3.38)$$

A partir de l'ordre 2, on résout séquentiellement :

$$\beta_p^i = ({}^t u_0^i) (K - K_0) v_p^i - \sum_{k=1}^{p-1} \beta_k^i ({}^t u_0^i) M_0 v_{p-k}^i - \sum_{k=0}^{p-1} \beta_k^i ({}^t u_0^i) (M - M_0) v_{p-1-k}^i \quad (3.39)$$

Ensuite on procède au calcul des vecteurs propres. Pour ce faire, on utilise une méthode de projection suivant l'équation :

$$v_p^i = \alpha_p^i u_0^i + \sum_{j=1, j \neq i}^n \alpha_p^j u_0^j = F_p^i, \quad 1 \leq n \leq N. \quad (3.40)$$

Dans cette équation, N est le nombre de modes propres recherchés, pour lever la singularité de la matrice $[K_0 - \lambda_0 M_0]$. Le problème revient à déterminer les coefficients α_k^i en résolvant l'équation :

$$(K_0 - \lambda_0 M_0) (\alpha_p^i u_0^i + \sum_{j=1, j \neq i}^n \alpha_p^j u_0^j). \quad (3.41)$$

En multipliant (3.41) par $({}^t u_0^k)$, les coefficients α_p^k peuvent être calculés suivant :

$$\begin{aligned}\alpha_p^k &= \frac{({}^t u_0^k) F_p^i}{\lambda_0^k - \lambda_0^i}, & k \neq i, \\ \alpha_1^i &= \frac{1 - ({}^t u_0^i) M u_0^i}{2}, & \alpha_p^i = \frac{1}{2} F_p'.\end{aligned}\tag{3.42}$$

La méthode d'homotopie proposée par Sliva et al. [10] est une approche assez facile à mettre en œuvre. Les limites de cette méthode sont, d'une part, qu'elle ne traite que le cas des valeurs propres linéaires et d'autre part, que les perturbations des paramètres de modélisation Δ_E , $\Delta\rho$ et Δh doivent être égales. On notera que le calcul des dérivées de haut degré pourrait être rendu complètement transparent en utilisant la DA. Nous reviendrons sur cet aspect au paragraphe 3.3.1.

3.2.4 Méthodes de dérivation analytique

Les méthodes de dérivation analytique sont basées sur la différentiation du problème linéaire de valeurs propres :

$$\begin{aligned}K u^i - \lambda^i M u^i &= 0, \\ ({}^t u^i M u^i) &= 1,\end{aligned}\tag{3.43}$$

par rapport aux paramètres de modélisation p . Dans cette équation, l'exposant i signifie que l'on considère la $i^{\text{ème}}$ solution propre. Il en résulte un système d'équations

$$(K - \lambda^i M)(u^i)^d = -(K^d - \lambda^i M^d)u^i + (\lambda^i)^d M u^i,\tag{3.44}$$

où $K^d = \frac{\partial K}{\partial p}$, $M^d = \frac{\partial M}{\partial p}$, $(u^i)^d = \frac{\partial u^i}{\partial p}$, $(\lambda^i)^d = \frac{\partial \lambda^i}{\partial p}$ sont respectivement les dérivées premières de la matrice de rigidité, de la matrice de masse, du vecteur propre et de la valeur propre. En multipliant (3.44) par $({}^t u^i)$, on obtient la dérivée de la valeur propre :

$$(\lambda^i)^d = {}^t (u^i) (K^d - \lambda^i M^d) u^i.\tag{3.45}$$

La matrice $(K - \lambda^i M)$ étant singulière, la dérivée du vecteur propre ne peut être calculée par une inversion matricielle. Plusieurs méthodes sont disponibles dans la littérature pour résoudre ce type de problèmes. Par exemple, la méthode de Nelson

[129] et la méthode de Rudisill et Chu [130] permettent de résoudre de manière efficace ce problème. Une approche algébrique basée sur la dérivation analytique est proposée dans [131, 132]. Dans cette approche, le calcul de sensibilité se résume à la résolution du système linéaire :

$$\begin{pmatrix} K - \lambda^i M & -Mu \\ -(^t u^i)M & 0 \end{pmatrix} \begin{Bmatrix} u^{di} \\ \lambda^{di} \end{Bmatrix} = \begin{Bmatrix} [K^d - \lambda^i M^d] u^i \\ \frac{1}{2} (^t u^i) M^d u^i \end{Bmatrix} \quad (3.46)$$

Pour des problèmes mécaniques, d'autres études ont opté pour une dérivation analytique des équations constitutives du modèle numérique ou analytique. Comme dans le cas de méthodes de perturbation, la solution est cherchée sous forme d'approximation de Taylor de premier ordre ou d'ordre n , ce dernier étant généralement égal 2. En adoptant cette approche, Lima et al [8] se sont intéressés à l'étude de sensibilité de la réponse fréquentielle des structures plaques sandwich contenant des couches viscoélastiques. Leur approche basée sur la dérivation analytique permet de calculer la sensibilité par rapport aux paramètres de modélisation et par rapport à la température. Elle prend en compte la dépendance en fréquence, la dépendance en température et le module complexe en considérant une loi viscoélastique écrite sous la forme :

$$G(\omega, T) = G'(\omega, T) + iG''(\omega, T) = G'(\omega, T)[1 + i\eta_G(\omega, T)], \quad (3.47)$$

où ω et T représentent la fréquence d'excitation et la température respectivement, $G(\omega, T)$ représente le module de cisaillement, i le nombre complexe unité. Le paramètre $\eta_G(\omega, T) = \frac{G'(\omega, T)}{G''(\omega, T)}$ est appelé facteur de perte. Moreau [133] a montré que pour les PVC (Polyvinyl chloride) généralement utilisés, le module de cisaillement viscoélastique peut être lié au module d'Young viscoélastique par la relation :

$$G(\omega, T) = \frac{E(\omega, T)}{2(1 + \nu)}, \quad (3.48)$$

où ν est le coefficient de cisaillement. La dépendance en température est prise en compte en exprimant la fréquence réduite ω_r à une température T en fonction de la température de référence T_0 suivant la relation $\omega_r(T) = \alpha_T(T)\omega$ où $\alpha_T(T)$ est une

fonction variable [8]. Cette fonction $\alpha_T(T)$ peut être obtenue expérimentalement. Pour un matériau spécifique, ces paramètres peuvent être obtenus expérimentalement ou à partir des courbes de fréquences réduites (courbes maitresse). Par exemple, pour le matériau ISD112, la courbe présentée dans la figure 3.2 permet de déterminer la valeur de α_T pour une température de référence T donnée.

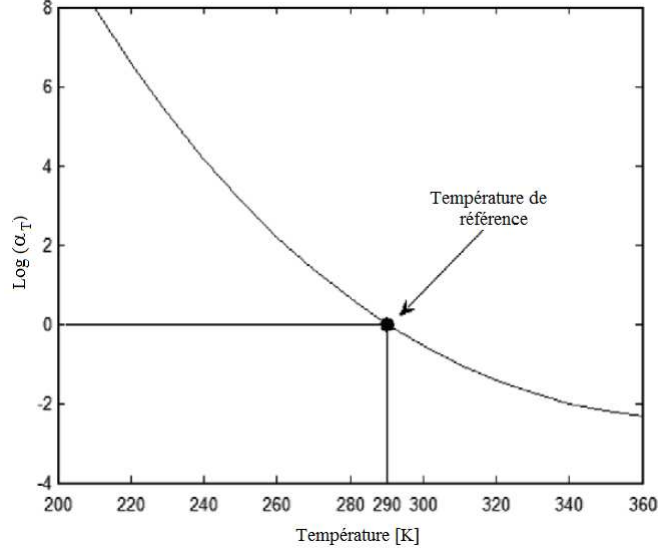


FIG. 3.2 – Courbe de la fonction α_T pour le matériau ISD112,

Considérons par exemple un problème de vibration forcée écrit sous la forme :

$$[Ke + K_v(\omega, T) - \omega^2 M]u(\omega, T) = F(\omega), \quad (3.49)$$

où $K(\omega, T)$ et M sont respectivement la matrice de rigidité et la matrice de masse obtenues à partir du modèle éléments finis. On suppose qu'elles dépendent des paramètres de modélisation p . En réponse fréquentielle, le problème de sensibilité conduit au calcul de la matrice dérivée

$$\frac{\partial H(\omega, T, p)}{\partial p} = -H(\omega, T_0, p) \left[\frac{\partial K(\omega, T_0, p)}{\partial p} - \omega^2 \frac{\partial M(p)}{\partial p} \right] H(\omega, T_0, p), \quad (3.50)$$

où

$$H(\omega, T_0, p) = [Ke + K_v(\omega, T) - \omega^2 M]^{-1}. \quad (3.51)$$

Comme nous l'avons évoqué dans le chapitre 1, la dérivation analytique de (3.50) peut être fastidieuse et source d'erreur dans le cas des lois viscoélastiques complexes. La méthode proposée dans [8] peut être améliorée en calculant ces dérivées automatiquement. Par ailleurs, cette approche nécessite l'inversion de la matrice $H(\omega, T) = [Ke + K_v(\omega, T) - \omega^2 M]$ pour chaque fréquence, ce qui engendre un temps de calcul élevé. Ce temps de calcul peut être réduit en adoptant la méthode de perturbation proposée dans [134]. Une seule inversion matricielle est alors nécessaire.

Dans le même cadre, Aryana et al [135] ont proposé une méthode d'analyse de sensibilité des modes propres de vibration basée sur l'approximation de second ordre. Leur approche utilise une discrétisation éléments finis. Dans cette approche, le problème aux valeurs propres (3.4) est différentié par rapport aux paramètres de modélisation d'un élément j du modèle éléments finis. L'équation de premier ordre obtenue est différentiée pour obtenir le problème de second ordre. Cette approche est essentiellement destinée à étudier l'influence de la présence de défauts dans les structures. Cette méthode est applicable à n'importe quelle structure, mais nécessite un effort de développement considérable sans utilisation de la différentiation automatique.

En utilisant la méthode de dérivation, Lin [114] a étudié les sensibilités pour l'identification des paramètres modaux de deux modes proches. Pour séparer les deux modes, il introduit une perturbation dans les paramètres de modélisation qui impactent soit la matrice de rigidité, soit la matrice de masse. Dans cette approche, les modes complexes sont traités en considérant une matrice de rigidité complexe de la forme $\bar{K} = K + iD$. Cette approche est d'intérêt limité car, dans le cas de structures contenant des matériaux viscoélastiques, elle ne peut traiter que le cas de module constant, ne tenant pas compte de la dépendance en fréquence de la matrice de rigidité.

Pour conclure, la précision de ces méthodes dépend de la taille de la perturbation. L'étude proposée dans [136] montre que, pour des perturbations de l'ordre de $\varepsilon = 0.1$, les méthodes d'approximation de premier ordre comme celles de Chen [30] ou de Bickford [31] sont efficaces. Par contre, pour des perturbations élevées, par exemple $\varepsilon = 0.3$ il

est nécessaire d'opter au minimum pour les méthodes d'approximation de second ordre ou la méthode d'approximation de Padé.

3.3 Analyse de sensibilité de problèmes aux valeurs propres non-linéaires

Ce chapitre propose, une méthode d'analyse de sensibilité des problèmes aux valeurs propres non-linéaire écrits sous la forme :

$$[K(\omega) - \omega^2 M]u = 0. \quad (3.52)$$

La non-linéarité matérielle induite par la loi viscoélastique ne permet pas d'utiliser les méthodes classiques de résolution des problèmes aux valeurs propres. Dans cette thèse nous proposons une analyse de sensibilité des problème de vibration des structures viscoélastiques écrit sous la forme (3.52). Notre méthode utilise la technique de l'homotopie, la MAN et la différentiation automatique.

3.3.1 Homotopie et MAN

Dans le chapitre 2, nous avons montré que les problèmes non-linéaires peuvent être résolus de manière générique et efficace par l'approche Diamant de la MAN. Pour résoudre un problème aux valeurs propres dépendant de la fréquence, on peut utiliser une méthode d'homotopie et évaluer la branche de solutions représentant cette déformation continue par la MAN [16] pour obtenir, en fin de branche, la solution de ce problème.

Pour illustrer la technique d'homotopie dans la résolution d'un problème mécanique non-linéaire, nous considérons le problème générique [17] :

$$\mathcal{R}(u, \lambda) = \mathcal{S}(u, \lambda) + \mathcal{T}(u, \lambda) = 0, \quad (3.53)$$

où $\mathcal{S}(u, \lambda)$ est un problème dont on sait calculer les solutions analytiquement ou numériquement, et $\mathcal{T} = \mathcal{R} - \mathcal{S}$ est le problème incluant les non-linéarités. La procédure

d'homotopie consiste à introduire un paramètre a dans le problème non-linéaire :

$$\mathcal{R}(u, \lambda, a) = \mathcal{S}(u\lambda) + a\mathcal{T}(u, \lambda) = 0. \quad (3.54)$$

Dans cette équation, le paramètre a qui pilote l'homotopie devient le paramètre de chemin au sens de la MAN. L'idée est de partir de la solution connue pour le problème linéaire $\mathcal{R}(u, \lambda, 0) = \mathcal{S}(u, \lambda) = 0$ pour obtenir la solution du problème non-linéaire $\mathcal{R}(u, \lambda, 1) = \mathcal{S}(u, \lambda) + \mathcal{T}(u, \lambda) = 0$. D'un point de vue numérique, cela consiste à déterminer une suite $\{a_{(j)}\}_{j=1, \dots, J}$ telle que $a_{(1)} = 0$ et $a_{(J)} = 1$ où l'indice J est déterminé lors de la procédure de continuation.

3.3.2 Procédure de continuation

Pour que la suite $a_{(j)}$ atteigne $a_{(J)} = 1$, on introduit le paramètre δa tel que $a_{(j+1)} = a_{(j)} + \delta a$. Le problème résiduel à chaque itération j est :

$$\mathcal{R}(u, \lambda) = \mathcal{S}(u, \lambda) + (a_{(j)} + \delta a)\mathcal{T}(u, \lambda). \quad (3.55)$$

On cherche les solutions u et λ sous la forme de séries développées par rapport au paramètre δa :

$$\begin{aligned} u(a_{(j)} + \delta a) &= U_0(a_{(j)}) + \sum_{k=1}^K \delta a^k u_k, \\ \lambda(a_{(j)} + \delta a) &= \lambda_0(a_{(j)}) + \sum_{k=1}^K \delta a^k \lambda_k. \end{aligned} \quad (3.56)$$

Ces approximations en séries permettent d'écrire les parties linéaire et non-linéaire \mathcal{S} et \mathcal{T} sous la forme de séries :

$$\begin{aligned} \mathcal{S}(a_{(j)} + \delta a) &= \mathcal{S}_0(a_{(j)}) + \sum_{k=1}^K \delta a^k \mathcal{S}_k, \\ \mathcal{T}(a_{(j)} + \delta a) &= \mathcal{T}_0(a_{(j)}) + \sum_{k=1}^K \delta a^k \mathcal{T}_k. \end{aligned} \quad (3.57)$$

En introduisant ces approximations dans (3.55) et en reprenant le formalisme Diamant, l'équation (3.55) peut être écrite sous la forme d'un système linéaire générique :

$$\begin{aligned} \left(\left\{ \mathcal{S}_1|_{u_1=Id, \lambda_1=0} \right\} + a_{(j)} \left\{ \mathcal{T}_1|_{u_1=Id, \lambda_1=0} \right\} \right) u_k &+ \left(\left\{ \mathcal{S}_1|_{u_1=0, \lambda_1=1} \right\} + a_{(j)} \left\{ \mathcal{T}_1|_{u_1=0, \lambda_1=1} \right\} \right) \lambda_k = \\ &- \left\{ \mathcal{S}_k|_{u_k=0, \lambda_k=0} \right\} - a_{(j)} \left\{ \mathcal{T}_k|_{u_k=0, \lambda_k=0} \right\} - \mathcal{T}_{k-1}. \end{aligned}$$

(3.58)

Les termes $\left(\left\{\mathcal{S}_1|_{u_1=Id, \lambda_1=0}\right\} + a_{(j)} \left\{\mathcal{T}_1|_{u_1=Id, \lambda_1=0}\right\}\right)$ désignent le Jacobien relatif à la branche j . Les variables $\left\{\mathcal{S}_k|_{u_k=0, \lambda_k=0}\right\}$ et $\left\{\mathcal{T}_k|_{u_k=0, \lambda_k=0}\right\}$ sont les coefficients de Taylor de \mathcal{S} et \mathcal{T} à l'ordre k , respectivement, calculés en initialisant les coefficients de Taylor u_k et λ_k à zéro. Contrairement à l'approche Diamant présentée dans le premier chapitre, il apparaît dans l'équation (3.57) le coefficient supplémentaire \mathcal{T}_{k-1} qui est une conséquence de l'homotopie. Le système sous-déterminé (3.57) peut être résolu en ajoutant une condition d'orthogonalité [137]. Ces coefficients peuvent être calculés de manière efficace avec la boîte à outils Diamant. En associant l'homotopie avec la MAN [16, 138] la solution est obtenue en un nombre d'itérations faibles (2 à 5 itérations).

Dans les précédents travaux, l'implémentation de l'homotopie [139] n'utilise qu'occasionnellement la différentiation automatique pour le calcul des Jacobiens [140]. A nouveau, l'usage de la DA facilite le calcul des dérivées de haut degré. L'efficacité du couplage homotopie-MAN-DA a été démontrée par Bilasse dans [17, 60]. La prise en compte de l'homotopie dans Diamant réside dans l'implémentation des formules (3.57) et (3.58) et la dérivation automatique des termes non-linéaires de haut degré. Cette implémentation a été présentée dans [17, 138] et utilisée dans la résolution de problèmes de vibration non-linéaire dépendant de la fréquence. Dans cette thèse, nous proposons en complément une approche d'analyse de sensibilité basée sur la DA et la MAN.

3.3.3 Calcul de sensibilité par DA

Comme dans le cas de l'approche Diamant sensibilité, le système linéaire (3.57) peut être différentié par rapport aux paramètres de modélisation. Les problèmes linéaires à

résoudre sont alors :

$$\begin{aligned}
 & \left(\left\{ \mathcal{S}_{1|U_1=id, \lambda_1=0} \right\} + a_{(j)} \left\{ \mathcal{T}_{1|U_1=id, \lambda_1=0} \right\} \right) U_k^d + \left(\left\{ \mathcal{S}_{1|U_1=0, \lambda_1=1} \right\} + a_{(j)} \left\{ \mathcal{T}_{1|U_1=0, \lambda_1=1} \right\} \right) \lambda_k^d = \\
 & - \left\{ \mathcal{S}_{k|U_k=0, \lambda_k=0}^d \right\} - a_{(j)}^d \left\{ \mathcal{T}_{k|U_k=0, \lambda_k=0} \right\} - a_{(j)} \left\{ \mathcal{T}_{k|U_k=0, \lambda_k=0}^d \right\} - \\
 & \left(\left\{ \mathcal{S}_{1|U_1=id, \lambda_1=0}^d \right\} + a_{(j)} \left\{ \mathcal{T}_{1|U_1=id, \lambda_1=0}^d \right\} + a_{(j)}^d \left\{ \mathcal{T}_{1|U_1=id, \lambda_1=0} \right\} \right) U_k - \\
 & \left(\left\{ \mathcal{S}_{1|U_1=id, \lambda_1=1}^d \right\} + a_{(j)} \left\{ \mathcal{T}_{1|U_1=id, \lambda_1=1}^d \right\} + a_{(j)}^d \left\{ \mathcal{T}_{1|U_1=id, \lambda_1=1} \right\} \right) \lambda_k - \mathcal{T}_{k-1}^d.
 \end{aligned} \tag{3.59}$$

La matrice linéaire tangente de (3.59) est le même que celle de (3.57)

La difficulté réside dans le calcul des dérivées mixtes apparaissant au second membre. On adopte la technique du couplage AdiMat-Diamant de la différentiation par transformation de source et la surcharge d'opérateur décrite au chapitre 1 et dans [15, 28].

3.4 Analyse de sensibilité des solutions propres dépendant de la fréquence

Considérons un problème aux valeurs propres non-linéaire écrit sous la forme :

$$[K(\lambda) - \lambda M]u = 0, \quad \lambda = \omega^2, \tag{3.60}$$

où K et M sont respectivement la matrice de rigidité et la matrice de masse. On suppose que la matrice de rigidité peut être décomposée sous la forme :

$$K(\lambda) = K_0 + K_\omega(\lambda), \tag{3.61}$$

où K_0 et K_ω représentent respectivement la partie linéaire et la partie non-linéaire de la matrice de rigidité. Le problème non-linéaire (3.61) peut être écrit sous la forme :

$$[K_0 - \lambda M]u + K_\omega(\lambda)u = 0. \tag{3.62}$$

A partir de cette formulation [88], la technique d'homotopie peut être appliquée en choisissant [17] :

$$\mathcal{S} = [K_0 - \lambda M]u, \quad \text{et} \quad \mathcal{T} = K_\omega(\lambda)u. \tag{3.63}$$

Pour des raisons de clarté, la dépendance par rapport au paramètre de modélisation p a été omise dans le problème de vibration (3.63). La modélisation du problème de vibration linéaire s'écrit donc :

$$S(u(p), \lambda(p)) = [K_0(p) - \lambda(p)M(p)]u(p) = 0. \quad (3.64)$$

Par la suite, nous n'indiquons plus le paramètre p .

La solution du problème linéaire étant indispensable pour le calcul de la solution non-linéaire, nous allons tout d'abord procéder à l'analyse de sensibilité de la solution linéaire.

3.4.1 Analyse de sensibilité du problème réel

La résolution du problème aux valeurs propres réel ne pose pas de difficulté majeure. Il peut être résolu par les méthodes numériques de calcul de valeurs propres et de vecteurs propres. Cette sensibilité est calculée en déterminants des dérivées ω_0^d et u_0^d qui sont respectivement les sensibilités de la fréquence propre ω_0 et du mode propre u_0 du problème linéaire. Elles peuvent être calculées en utilisant l'une des méthodes proposées au paragraphe 3.2.1. Pour des résultats plus précis, nous utilisons la méthode de dérivation analytique (voir paragraphe 3.2.4), en effectuant les dérivations avec l'outil de DA Adimat.

Pour le calcul des sensibilités, l'équation (3.64) est différenciée par rapport au paramètre de modélisation p . On obtient l'équation :

$$[K - \omega_0^2 M] u_0^d + [K^d] u_0 - 2\omega_0^d \omega_0 [M] u_0 - \omega_0^2 [M^d] u_0 = 0, \quad (3.65)$$

où $[K^d]$ et $[M^d]$ sont les matrices de rigidité $[K]$ et de masse $[M]$ dérivées. Dans la MAN, le problème mécanique est résolu par un programme informatique. Dans ce cas, au lieu de procéder à une dérivation analytique ou symbolique pour calculer ces matrices (3.46) comme dans [114], il est plus judicieux et facile de procéder à une différentiation automatique de tout le code de résolution. Ceci permet d'obtenir à la fois les matrices

dérivées, mais aussi les dérivées de tous les paramètres intervenant dans la résolution [22].

Cette équation peut être écrite pour chaque solution propre sous la forme :

$$\left[K - \omega_p^2 M \right] u_0^{di} + \left[K^d \right] u_0^i - 2\omega_0^{di} \omega_0^i [M] u_0^i - \omega_0^{i2} [M^d] u_0^i = 0, \quad \forall i \in \{1, 2, \dots, n\}. \quad (3.66)$$

Ceci permet d'obtenir un système à n équations pour $n+1$ inconnues. Pour résoudre ce système, on ajoute la condition de normalisation des modes propres par rapport à la matrice de masse (3.4). Cette condition de normalité des modes propres est différenciée par rapport au paramètre de modélisation pour faire apparaître la dérivée du mode propre recherché. On obtient :

$$2({}^t u_0^{di})[M]u_0^i + u_0^i[M^d]u_0^i = 0. \quad (3.67)$$

En associant les équations (3.67) et (3.66), on obtient un système linéaire à $n+1$ équations :

$$\begin{bmatrix} \left[K - \omega_0^{i2} M \right] & 2\omega_0^i [M] \{u_0^i\} \\ {}^t \{2 [M] \{u_0^i\}\} & 0 \end{bmatrix} \begin{Bmatrix} u_0^{di} \\ \omega_0^{di} \end{Bmatrix} = \begin{Bmatrix} \omega_0^{i2} [M^d] \\ ({}^t \{u_0^i\}) [M^d] \{u_0^i\} \end{Bmatrix}. \quad (3.68)$$

La résolution de ce système permet de calculer la sensibilité des modes propres réels. Ici, la différentiation automatique intervient dans le calcul des dérivées K^d et M^d . Ces dérivées sont obtenues facilement par une différentiation source à source du code implémentant le modèle éléments finis $\mathcal{R}(u, \lambda)$. La différentiation avec Adimat de l'équation $\mathcal{R}(u, \lambda)$ permet d'obtenir ces matrices.

3.4.2 Analyse de sensibilité du problème complexe

Pour bien comprendre la démarche adoptée pour le calcul de sensibilité, il est important de présenter d'abord l'approche de résolution du problème de modes propres et valeurs propres non-linéaires [60] approchant les solutions sous forme de séries. Dans cette partie l'exposant i désignant la $i^{\text{ème}}$ solution propre sera omis pour éviter de surcharger les termes des séries de Taylor.

En appliquant la technique d'homotopie décrite au paragraphe 3.3.1, le problème non-linéaire à résoudre peut être écrit à l'ordre k sous la forme :

$$\begin{aligned} & \left[\left\{ \mathcal{S}_{0|u_0=id} \right\} + a_j \left\{ \mathcal{T}_{0|u_0=id} \right\} \right] u_0 = 0, \quad k = 0, \\ & \left[\left\{ \mathcal{S}_{1|u_1=id, \lambda_1=0} \right\} + a_j \left\{ \mathcal{T}_{1|u_1=id, \lambda_1=0} \right\} \right] u_k = - \left\{ \mathcal{S}_{k|u_k=0} \right\} - a_j \left\{ \mathcal{T}_{k|u_k=0} \right\} - \mathcal{T}_{k-1}, \quad k \geq 1. \end{aligned} \quad (3.69)$$

Notons

$$A = \left[\left\{ \mathcal{S}_{0|u_0=id} \right\} + a_j \left\{ \mathcal{T}_{0|u_0=id} \right\} \right] \quad (3.70)$$

Le problème à l'ordre $k = 0$ montre que la matrice linéaire tangente A est singulière. Pour lever la singularité, on prend en compte la condition d'orthogonalité ${}^t u_0 u_k = 0$ des modes propres en introduisant le multiplicateur de Lagrange χ , le système linéaire est donné par :

$$\begin{bmatrix} A & u_0 \\ ({}^t u_0) & 0 \end{bmatrix} \begin{bmatrix} u_n \\ \chi \end{bmatrix} = \begin{bmatrix} - \left\{ \mathcal{S}_{k|u_k=0} \right\} - \left\{ \mathcal{T}_{k|u_k=0} \right\} + \mathcal{T}_{k-1} \\ 0 \end{bmatrix}. \quad (3.71)$$

La singularité de la matrice A implique que le problème à l'ordre $k \geq 1$ n'admet de solution que si l'équation (3.72) est vérifiée :

$${}^t u_0 \left[\left\{ \mathcal{S}_{k|u_k=0} \right\} - a_j \left\{ \mathcal{T}_{k|u_k=0} \right\} - \mathcal{T}_{k-1} \right] = 0, \quad (3.72)$$

Pour faire apparaître le coefficient de Taylor λ_n dans cette équation, les termes $\left\{ \mathcal{S}_{k|u_k=0} \right\}$ et $\left\{ \mathcal{T}_{k|u_k=0} \right\}$ sont réécrits en utilisant la décomposition du principe générique de l'approche diamant présenté dans [17] :

$$\begin{aligned} \left\{ \mathcal{T}_{k|u_k=0} \right\} &= \left\{ \mathcal{S}_{1|\lambda_1=1} \right\} \lambda_n + \left\{ \mathcal{T}_{n|\lambda_n=0} \right\}, \\ \left\{ \mathcal{S}_{k|u_k=0} \right\} &= \left\{ \mathcal{S}_{1|\lambda_1=1} \right\} \lambda_n + \left\{ \mathcal{S}_{n|\lambda_n=0} \right\}, \end{aligned} \quad (3.73)$$

les coefficients de Taylors λ_k peuvent être calculés comme suit :

$$\lambda_k = \frac{{}^t u_0 \left[\left\{ \mathcal{S}_{k|u_k=0, \lambda_k=0} \right\} - a_j \left\{ \mathcal{T}_{k|u_k=0, \lambda_k=0} \right\} - \mathcal{T}_{k-1} \right]}{{}^t u_0 \left[\left\{ \mathcal{S}_{1|u_1=0, \lambda_1=1} \right\} - a_j \left\{ \mathcal{T}_{1|u_1=0, \lambda_1=1} \right\} \right]}. \quad (3.74)$$

3.4.3 Calcul de sensibilité du problème non-linéaire

Le calcul de sensibilité est fait en différentiant les équations (3.74) et (3.71) par rapport aux paramètres de modélisation p . L'objectif est de montrer que les termes intervenant dans la résolution du problème de sensibilité peuvent être calculés facilement au utilisant le couplage Adimat-Diamant. Pour le calcul de la sensibilité de la fréquence propre λ_k , nous posons :

$$\begin{aligned}\phi_k &= \left\{ \mathcal{S}_{k|u_k=0, \lambda_k=0} \right\} - a_j \left\{ \mathcal{T}_{k|u_k=0, \lambda_k=0} \right\} - \mathcal{T}_{k-1} \\ \psi &= \left\{ \mathcal{S}_{1|u_1=0, \lambda_1=1} \right\} - a_j \left\{ \mathcal{T}_{1|u_1=0, \lambda_1=1} \right\},\end{aligned}\tag{3.75}$$

où ϕ_k et ψ sont respectivement les termes du numérateur et du dénominateur de l'équation (3.74). La sensibilité de la valeur propre s'écrit en fonction de ψ et ϕ_k comme suit :

$$\lambda_k^d = \frac{({}^t u_0^d \phi_k + {}^t u_0 \phi_k^d)({}^t u_0 \psi) - ({}^t u_0 \phi_k)({}^t u_0^d \psi + {}^t u_0 \psi^d)}{({}^t u_0 \psi)^2},\tag{3.76}$$

où ϕ_k^d et ψ^d représente les dérivées par rapport à p de ϕ_k et ψ . Ces dérivées sont données par :

$$\begin{aligned}\phi_k^d &= \left\{ \mathcal{S}_{k|u_k=0, \lambda_k=0}^d \right\} - a_j^d \left\{ \mathcal{T}_{k|u_k=0, \lambda_k=0}^d \right\} - a_j \left\{ \mathcal{T}_{k|u_k=0, \lambda_k=0}^d \right\} - \mathcal{T}_{k-1}^d \\ \psi_k^d &= \left\{ \mathcal{S}_{1|u_1=0, \lambda_1=1}^d \right\} - a_j^d \left\{ \mathcal{T}_{1|u_1=0, \lambda_1=1}^d \right\} - a_j \left\{ \mathcal{T}_{1|u_1=0, \lambda_1=1}^d \right\}.\end{aligned}\tag{3.77}$$

Le mode propre non-linéaire dérivé peut être calculé en différentiant l'équation (3.74) par rapport à p . Le système à résoudre est linéaire :

$$\begin{bmatrix} A & u_0 \\ {}^t u_0 & 0 \end{bmatrix} \begin{bmatrix} u_k^d \\ \chi^d \end{bmatrix} = \begin{bmatrix} -\left\{ \mathcal{S}_{k|u_k=0}^d \right\} - \left\{ \mathcal{T}_{k|u_k=0}^d \right\} + \mathcal{T}_{k-1}^d \\ 0 \end{bmatrix} - \begin{bmatrix} A^d & u_0^d \\ {}^t u_0^d & 0 \end{bmatrix} \begin{bmatrix} u_k \\ \chi \end{bmatrix}.\tag{3.78}$$

3.4.4 Implémentation

Comme au chapitre 2, les dérivées mixtes \mathcal{S}_k^d et \mathcal{T}_k^d de $S(U, \lambda)$ et $T(U, \lambda)$ peuvent être calculées de manière automatique en utilisant le couplage Adimat-Diamant. En pratique, cette approche nécessite moins de développement car les opérations de dérivation sont faites automatiquement. L'algorithme de calcul de sensibilité (table 3.1)

est le même quelque soit le modèle viscoélastique. La DA peut donc être appliquée localement à la loi viscoélastique, ce qui permet de prendre en compte plusieurs modèles viscoélastiques sans redifférentier entièrement le code. Celui-ci peut donc être implémenté dans une fonction Matlab.

1. Sensibilité des solutions propres réelles :

Différentiation du problème résiduel $R(u, \lambda) = \mathcal{S}(u, \lambda) + a\mathcal{T}(u, \lambda) \Rightarrow$

$$g_-R(g_-u, ug_- \lambda, \lambda) = g_-S(-u, u, g_- \lambda, \lambda) + T(-u, u, g_- \lambda, \lambda)$$

Calcul des solution propres réelles (U_0, λ_0) solution de $S(U, \lambda) = 0$ et $\omega_0 = \sqrt{\lambda_0}$

Calcul de (u_0^d, ω_0^d) solution de (3.68)

2. Calcul des séries pour $k = 1, \dots, K$.

Intinialisation de u et λ comme des variables TaylorSeries

Evaluation de $\{\mathcal{S}_1|_{u_1=id, \lambda_1=0}\}$, $\{\mathcal{T}_1|_{u_1=id, \lambda_1=0}\}$ puis de A suivant (3.70),

Calcul de ϕ_k , ψ , ϕ_k^d et ψ^d suivant les équations (3.75), (3.77)

Calcul de λ_k suivant l'équation (3.74) et λ_k^d suivant (3.76),

Calcul de u_k et u_k^d solution respective de (3.71) et (3.78),

3. Test de convergence :

Calcul de δa_{max} ,

Calcul $a_{j+1} = a_j + \delta a_{max}$

Si $a_{j+1} < 1$ retour au étape 2 avec $u_0 = u(j)$, $\lambda_0 = \lambda(j)$

TAB. 3.1 – Algorithme de Diamant sensibilité

Dans cet algorithme, $u(j)$ et $\lambda(j)$ représentent la solution calculée à l'itération j de la procédure de continuation. Cet algorithme est implémenté en Matlab.

3.5 modélisation directe des structures sandwich viscoélastiques en vibration libre

Considérons une structure sandwich symétrique constituée de trois couches. Les faces élastiques sont assemblées entre elles par une couche viscoélastique. Les propriétés amortissantes de ces structures proviennent d'un écart entre les déplacements longitudinaux dans les faces rigides et dans la couche viscoélastique soumise à un cisaillement important. Pour cette raison, le modèle cinématique de type zigzag est utilisé [141, 142, 4].

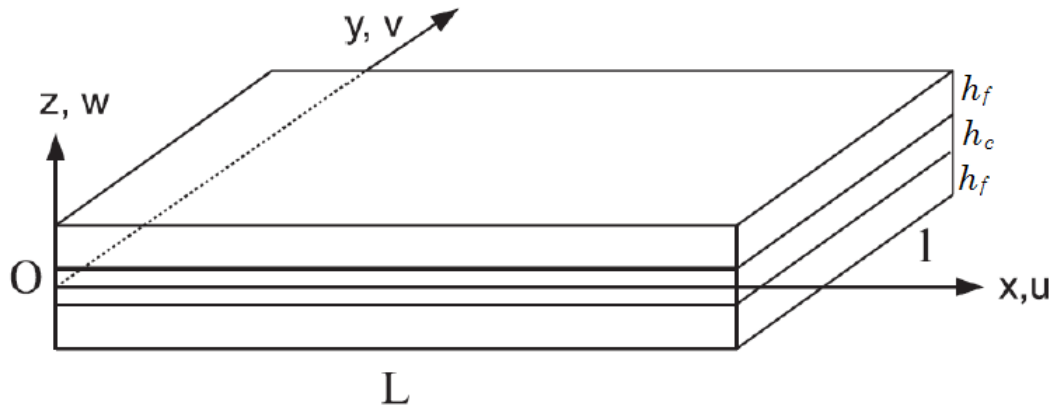


FIG. 3.3 – Structure sandwich avec le système de coordonnées.

La figure 3.3 présente une structure sandwich avec le système de coordonnées (x, y, z) ainsi que les déplacements suivant chaque axe. Les hypothèses cinématiques sont identiques à celles de la poutre sandwich traitée dans le deuxième chapitre :

- les trois couches subissent la même déflexion w ,
- les interfaces sont supposées parfaites, ce qui implique la continuité des déplacements aux interfaces,
- les faces sont supposées homogènes et isotropes,
- la couche viscoélastique a un module d'Young ou de cisaillement complexe dépendant de la fréquence.

Pour traiter à la fois le cas des structures poutre et plaque, nous avons opté pour une formulation en deux dimensions. Sous ces hypothèses, dans la théorie des plaques de Kirchhoff-love, les champs de déplacements pour les faces sont donnés par :

$$\begin{aligned} U_i(x, y, z, t) &= u_i(x, y, t) - (z - z_i) \frac{\partial w(x, y, t)}{\partial x}, \\ V_i(x, y, z, t) &= v_i(x, y, t) - (z - z_i) \frac{\partial w(x, y, t)}{\partial y}, \quad i = 1, 3, \\ W(x, y, z, t) &= w(x, y, t), \end{aligned} \quad (3.79)$$

où u_i et v_i sont les déplacements dans le plan médian de la couche i suivant les axes ox et oy , et w est le déplacement suivant l'axe (oz). En positionnant l'origine du repère dans le plan médian de la couche centrale viscoélastique, nous pouvons écrire : $z_1 = -z_3 = \frac{h_f + h_c}{2}$, $z_2 = 0$. En utilisant le tenseur de Green-Lagrange, le tenseur de déformation peut être écrit sous la forme :

$$\varepsilon_i = \Gamma_i + (z - z_i)\kappa, \quad i = 1, 3. \quad (3.80)$$

Les tenseurs Γ et κ sont respectivement le vecteur de déformation de membrane, le vecteur de déformation en flexion. Pour les faces, ces vecteurs de déformation sont donnés par :

$$\Gamma_i = \begin{Bmatrix} u_{i,x} \\ v_{i,y} \\ u_{i,x} + v_{i,y} \end{Bmatrix}, \quad \kappa = \begin{Bmatrix} -w_{,xx} \\ -w_{,yy} \\ -2w_{,xy} \end{Bmatrix}, \quad (3.81)$$

où la convention $\xi_{,x} = \frac{\partial \xi}{\partial x}$ est utilisée. Le champ de déplacements dans la couche viscoélastique $i = 2$ est donné par :

$$\begin{aligned} U_2(x, y, z, t) &= u(x, y, t) - (z - z_i)\beta_x, \\ V_2(x, y, z, t) &= v(x, y, t) - (z - z_i)\beta_y, \\ W_2(x, y, z, t) &= w(x, y, t). \end{aligned} \quad (3.82)$$

Les tenseurs de déplacement qui en découlent sont :

$$\Gamma_2 = \begin{Bmatrix} u_{,x} \\ v_{,y} \\ u_{,x} + v_{,y} \end{Bmatrix}, \quad \kappa_2 = \begin{Bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \beta_{x,y} + \beta_{y,x} \end{Bmatrix}, \quad \xi = \begin{Bmatrix} w_{,x} + \beta_x \\ w_{,y} + \beta_y \end{Bmatrix}. \quad (3.83)$$

3.5.1 Lois de comportement

Les lois de comportement de la plaque sandwich expriment les efforts de membrane résultants et les moments de flexion résultants dans les faces. En utilisant la loi de Hooke généralisée, les efforts de membrane N_i et les efforts de flexion sont définis dans les faces par :

$$\begin{aligned} N_i &= \begin{Bmatrix} N_{ixx} \\ N_{iyy} \\ N_{ixy} \end{Bmatrix} = [C_m] \{\Gamma_i\}, \quad i = 1, 3, \\ M_i &= \begin{Bmatrix} M_{ixx} \\ M_{iyy} \\ M_{ixy} \end{Bmatrix} = [C_b] \{\kappa_i\}, \end{aligned} \quad (3.84)$$

où $[C_m]$ et $[C_b]$ sont respectivement les matrices symétriques de rigidité en membrane et en flexion des faces élastiques. Ces matrices sont fournies en Annexe A. Pour la couche centrale ($i = 2$) dont la viscoélasticité est supposée linéaire et dépendante de la fréquence, un effort de cisaillement est engendré en plus des efforts de membrane et de flexion. Ces efforts peuvent être écrits sous la forme :

$$\begin{aligned} N_2 &= \begin{Bmatrix} N_{2xx} \\ N_{2yy} \\ N_{2xy} \end{Bmatrix} = [C_{2m}(t)] \frac{\partial}{\partial t} * \{\Gamma_2\}, \\ M_2 &= \begin{Bmatrix} M_{2xx} \\ M_{2yy} \\ M_{2xy} \end{Bmatrix} = [C_{2b}(t)] \frac{\partial}{\partial t} * \{\kappa_2\}, \\ Q &= \begin{Bmatrix} Q_{xz} \\ Q_{yz} \end{Bmatrix} = [C_{2s}(t)] \frac{\partial}{\partial t} * \{\xi\}. \end{aligned} \quad (3.85)$$

Dans cette équation, l'opérateur $*$ représente le produit de convolution. Les matrices $[C_{2m}]$, $[C_{2b}]$ et $[C_{2s}]$ sont respectivement les matrices rigidité en membrane, en flexion et en cisaillement de la couche viscoélastique. Selon le matériau viscoélastique, ces matrices peuvent être fonction d'un module d'Young complexe constant, ou d'une fonction

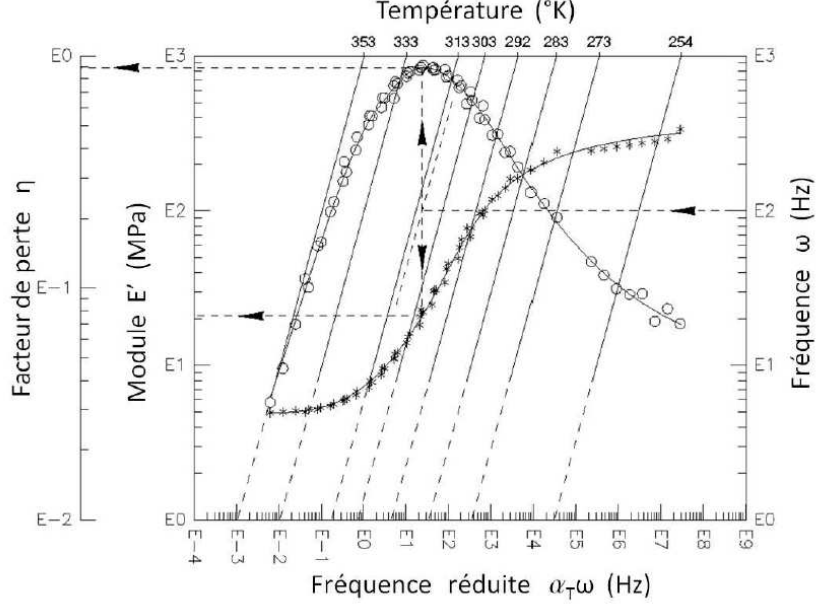


FIG. 3.4 – Monogramme en fréquence réduite d'un matériau viscoélastique

viscoélastique dont les paramètres sont déterminés expérimentalement. Elles peuvent également être fonction de la température. Mais cette dernière n'est pas prise en compte dans notre étude. Par exemple, le monogramme représenté sur la figure 3.4, issu de données expérimentales, permet de déterminer la variation du module d'Young E^* en fonction du facteur de perte η et du module de stockage E' .

L'équation du mouvement de vibration des plaques sandwich symétriques est formulée de manière variationnelle à l'aide du principe des travaux virtuels d'un système non-conservatif :

$$\delta p_{int} + \delta p_{ext} + \delta p_{acc} = 0, \quad (3.86)$$

où δp_{int} , δp_{ext} et δp_{acc} sont respectivement la puissance virtuelle des efforts intérieurs, la puissance virtuelle des efforts extérieurs et la puissance virtuelle des efforts d'accélération. Leur expression pour une structure plaque sandwich symétrique est donné par :

$$\begin{aligned}
 \delta p_{int} &= - \int_s \int_1^3 (\sum_1^3 ({}^t\{N_i\})\{\delta\Gamma_i\} + \sum_{i=1,3} {}^t\{M\}\{\delta\kappa\} + {}^t\{M_2\}\{\delta\kappa_2\} + \{Q\}\{\xi_2\}) ds \\
 \delta p_{ext} &= \int_s \int F(x, y, t) \delta w ds \\
 \delta p_{acc} &= - \int_s \int (2\rho_f * h_f + \rho_c h_c) \frac{\partial^2 w}{\partial t^2} \delta w ds.
 \end{aligned} \tag{3.87}$$

Dans l'équation (3.87), $F(x, y, t)$ représente les efforts extérieurs transversaux appliqués, les constantes ρ_f et ρ_c représentent la densité massique respective des faces et de la couche viscoélastique. La surface de la structure sandwich considérée est notée s . Dans cette formulation, seule l'inertie du déplacement transversal w est considérée. Les inerties des déplacements en membrane u et v sont négligées. En utilisant une discrétisation par éléments finis, le problème non-linéaire de vibration libre est écrit sous la forme :

$$[K_0 + E(\omega)K_v - \omega^2 M]u = 0. \tag{3.88}$$

Des éléments quadratiques ont été utilisés avec des fonctions de forme de type Hermite pour la variable le déplacement transversal w . Les fonctions de forme classiques sont utilisées pour les déplacements en membrane u et v . Notre approche est appliquée pour l'étude de sensibilité des poutres et plaques sandwich. Pour la couche viscoélastique, plusieurs matériaux ont été considérés :

- un matériau viscoélastique à module complexe constant [4] dont la loi viscoélastique est écrite sous la forme :

$$E(\omega) = E_0(1 + i\eta_c), \tag{3.89}$$

où $E(0) = E_0$ est le module d'Young d'élasticité retardé, η est le facteur de perte. Cette formule n'est valable que dans le cas où le module de stockage du matériau viscoélastique et le facteur de perte varient très peu en fonction de la fréquence. Dans ce cas, le problème (3.88) est ramené à un problème aux valeurs propres linéaire complexe.

	20°C			27°C		
j	$G_0(Pa)$	Δ_j	$\omega_j(rad.s^{-1})$	$G_0(Pa)$	Δ_j	$\omega_j(rad.s^{-1})$
1	0.510 ⁶	2.8164	31.1176	0.0511.10 ⁶	0,746	468.7
2		13.1162	446.4542		3.265	4742.4
3		45.46655	5502.5318		43.284	71532.5

TAB. 3.2 – Paramètres du modèle Maxwell généralisé pour l'ISD112 à 20°C et 27°C

- Un matériau viscoélastique, le PVB (polyvinyl de butyral) [60] par exemple étudié pour son amortissement faible, dont la loi viscoélastique est donnée par le module de cisaillement :

$$G_c^*(\omega) = G_\infty + (G_0 - G_\infty)[1 + (i\omega\tau)^{1-\alpha}]^{-\beta}, \quad (3.90)$$

où, α , β et τ sont des coefficients dépendant de la température associés à la loi, G_0 est le module de cisaillement instantané ($t=0$) et G_∞ le module de cisaillement à $t = \infty$. Dans nos applications, le PVB est considéré à 20°C. Les coefficients associés sont $G_0 = 479.10^3 Pa$, $G_\infty = 2.3510^8 Pa$, $\tau = 0.3979$, $\alpha = 0.46$ et $\beta = 0.1946$. Le module d'Young complexe est donné par $E_c^*(\omega) = 2(1 + \nu_c)G_c^*(\omega)$.

- Le polymère ISD112 choisi pour son amortissement élevé. La loi d'évolution du module de cisaillement est donnée par la formule de Maxwell généralisée [143, 144] :

$$G_c^*(\omega) = G_0 \left(1 + \sum_{j=1}^3 \frac{\Delta_j \omega}{\omega - i\omega_j} \right), \quad (3.91)$$

où G_0 est le module de cisaillement d'élasticité retardée, Δ_j et ω_j sont des paramètres du modèle obtenus par lissage de la courbe maitresse. Ces coefficients sont fonction de la température. Dans notre exemple nous allons les considérer aux températures de 20°C [144], et de 27°C [143]. Les paramètres correspondants à cette température sont regroupés dans le tableau 3.2 pour une structure sandwich aluminium/ISD112/aluminium.

3.6 Résultats numériques

Le code de résolution basé sur l'algorithme 3.1 et implémentant le problème résiduel (3.88) a été différentié par rapport au paramètre de modélisation p par Adimat. Pour les structures sandwichs viscoélastique, les paramètre p peut être l'épaisseur des couches, le module de de Young ou le module de cisaillement ou encore une rigidité élémentaire.

3.6.1 Analyse de sensibilité par rapport à l'épaisseurs et par rapport au module d'Young

Pour effectuer le calcul de sensibilité par rapport à l'épaisseur, nous introduisons une perturbation dans le code de résolution sous la forme :

$$h_c = h_{c0}(1 + \delta h), \quad h_f = h_{f0}(1 + \delta h), \quad (3.92)$$

où h_{c0} , h_{f0} sont respectivement les épaisseurs initiales respectives de la couche viscoélastique et des faces, et δh est la perturbation appliquée. Le code ainsi modifié est différentié par rapport à l'épaisseur. Les sensibilités sont obtenues en calculant les dérivées directionnelles par rapport à la perturbation δh . De la même manière, les sensibilités par rapport au module d'Young ou par rapport au module de cisaillement sont calculées en introduisant une perturbation δE ou δG comme suit :

$$E = E_0(1 + \delta E), \quad \text{ou} \quad G = G_0(1 + \delta G), \quad (3.93)$$

où E_0 et G_0 représentent respectivement le module d'Young et le module de cisaillement du matériau considéré. Comme dans le cas de la perturbation de l'épaisseur, les sensibilités sont obtenues en calculant les dérivées directionnelles par rapport à δE ou δG .

3.6.2 Sensibilité par rapport aux éléments de la matrice de rigidité

La sensibilité par rapport à la matrice de rigidité élémentaire est très peu étudiée dans le cas des modes propres de vibration. Une approche analytique a été proposée dans [135]. Elle peut être utilisée pour la détection de défaut ou d'un endommagement dans une structure. Dans cette thèse, nous avons utilisé notre approche pour l'étude de sensibilité des solutions propres par rapport à une perturbation de la matrice de rigidité élémentaire. La perturbation est appliquée à un élément de la discrétisation en modifiant sa matrice de rigidité K_{e0} suivant l'équation :

$$K_e = (1 + \delta K_e)K_{e0}. \quad (3.94)$$

En pratique, la perturbation est appliquée en considérant un point déterminé par sa position dans la structure. Une procédure permet de localiser l'élément contenant ce point. La perturbation est alors appliquée à cet élément. La perturbation de la matrice de rigidité élémentaire peut être appliquée simultanément à plusieurs éléments. Comme dans les autres cas, le code modifié est différentié par rapport à la perturbation δK_e .

3.6.3 Validation du code linéaire tangent

L'objectif du test de Taylor, voir paragraphe 1.3.4, est de valider l'exactitude des dérivées K^d et M^d . Pour ce faire, nous avons considéré une plaque sandwich dont les caractéristiques sont fournies dans le tableau 3.3. La loi de comportement viscoélastique du matériau DYAD606 est similaire à celle de ISD112.

Les résultats du test de Taylor réalisé sur la matrice de masse M , la matrice de rigidité K et sur la valeur propre λ_0 sont donnés dans le tableau 3.4. Notons que les solutions propres calculées sont conformes à celles obtenues dans les précédentes études [60, 17, 134].

Faces élastiques	Module de Young $E_f = 6.89^{10} Pa$ Coefficient de Poisson $\nu_f = 0.3$ Masse volumique $\rho_f = 2740 Kg.m^{-3}$ Epaisseur $h_f = 0.762mm$
Cœur viscoélastique DYAD606	Module d'Young $E_0 = 1794 \times 10^3 Pa$ Coefficient de Poisson $\nu_c = 0.3$ Masse volumique $\rho_c = 1104 Kg.m^{-3}$ Epaisseur $h_c = 0.254 \times 10^{-3}m$
Poutre	Longueur $L = 348mm$ Largeur $l = 304mm$

TAB. 3.3 – Propriétés mécaniques et géométriques de la plaque sandwich

α	$r_\alpha = \left(\frac{K_{0FD}^d}{K_{0AD}^d} \right)^h$	$r_\alpha = \left(\frac{M_{FD}^d}{M_{AD}^d} \right)^h$	$r_\alpha = \left(\frac{\lambda_{0FD}^d}{\lambda_{0AD}^d} \right)^h$
1.00e-00	8.537652632359007e-001	8.537652632161403e-001	5.877009072111190e-001
1.00e-01	9.813764965541341e-001	9.813764957225397e-001	9.322946226195408e-001
1.00e-02	9.980760653204313e-001	9.980760612223713e-001	9.927552138746980e-001
1.00e-03	9.998069452609518e-001	9.998069133406696e-001	9.992703969355088e-001
1.00e-04	9.999803023019962e-001	9.999804914417448e-001	9.999276504913326e-001
1.00e-05	9.999971110185502e-001	9.999993457435168e-001	9.999979782184000e-001
1.00e-06	1.000022955523101e+000	9.999373786573207e-001	9.999482297737214e-001
1.00e-07	1.000262744559671e+000	9.998594679727527e-001	9.9987129762335198e-001
1.00e-08	1.003048720213047e+000	9.980805836957792e-001	9.9840710840176365e-001
1.00e-09	1.015916020230264e+000	9.936846703452384e-001	9.874760567142390e-001

TAB. 3.4 – Test de Taylor pour une perturbation h_c^d dans la direction h_c de l'épaisseur de la couche viscoélastique.

Faces élastiques	Module de Young $E_f = 6.9^{10} Pa$
	Coefficient de Poisson $\nu_f = 0.3$
	Masse volumique $\rho_f = 2766 Kg.m^{-3}$
	Epaisseur $h_f = 1.524mm$
Cœur polymère PVB	Module d'Young, loi (3.89), $G_0 = 479 \times 10^3 Pa$;
	Coefficient de Poisson $\nu_c = 0.3$
	Masse volumique $\rho_c = 999 Kg.m^{-3}$
	Epaisseur $h_c = 0.127mm$
Dimension	Longueur $L = 177.8mm$
	Largeur $l = 12.7mm$

TAB. 3.5 – Caractéristiques de la poutre sandwich à cœur viscoélastique à module constant.

3.6.4 Analyse de sensibilité du modèle viscoélastique à module complexe constant

Nous avons utilisé notre approche pour calculer la sensibilité des solutions propres des structures sandwich poutre et plaque. Le cœur est constitué d'un matériau polymère dont le module d'Young viscoélastique vérifie (3.89).

Cas de la poutre encastree-libre

Les caractéristiques de la poutre sandwich considérée sont données dans le tableau (3.5). Cette poutre a fait l'objet de plusieurs études [138, 115]. Les faces élastiques sont en aluminium. Les sensibilités sont étudiées par rapport aux épaisseurs et aux modules de Young.

Nous nous intéressons principalement à l'évolution des fréquences propres, des facteurs de perte et des modes propres par rapport à une perturbation d'un paramètre de modélisation. Les résultats numériques calculés pour divers valeurs du facteur de perte

sont présentés dans les tableaux 3.6 et 3.7 pour une poutre encastree-libre à laquelle une perturbation de l'épaisseur de la couche viscoélastique est appliquée.

La première colonne des tableaux 3.6 présente les fréquences propres de la poutre non perturbée. La deuxième colonne présente les sensibilités calculées par notre méthode pour une perturbation de $\delta h_c = 10\%$ de la couche viscoélastique. La troisième colonne présente les fréquences propres obtenues par un calcul direct en prenant une épaisseur $h_c = h_{c0}(1 + \delta h_c)$. La colonne 4 des tableaux est la somme des valeurs de la première colonne et de celles de la deuxième colonne. Cette colonne montre que la sensibilité calculée prédit bien la fréquence propre et le facteur de perte de la poutre lorsque l'épaisseur de la couche viscoélastique est modifiée.

Le test réalisé en perturbant l'épaisseur des faces donne des résultats similaires. Ces résultats sont présentés dans les tableaux 3.8 et 3.9. Il apparait que le comportement dynamique de la poutre est plus sensible à une perturbation de l'épaisseur des faces qu'à celle de l'épaisseur de la couche viscoélastique. Ceci est attendu car, de manière générale, les fréquences propres d'une structure sandwich viscoélastique sont déterminées par les paramètres des faces élastiques.

L'importance de la sensibilité de la poutre à une perturbation des faces élastiques peut être également visualisée en traçant les modes propres pour un facteur de perte $\eta = 1.5$. Sur les figures 3.6 et 3.7, on remarque que l'amplitude de la déflexion dérivée W^d tracée pour les trois premiers modes est plus importante dans le cas de la perturbation des faces que dans le cas de la perturbation du cœur.

Cas de la plaque

Notre approche a été appliquée à une structure plaque sandwich à module viscoélastique constant. Les caractéristiques de la plaque sandwich sont données dans le tableau 3.12. La plaque considérée est en appui simple sur les quatre cotés. Les résultats sont présentés dans les tableaux 3.13 et 3.14 pour une perturbation du module de cisaillement de la couche viscoélastique.

η	$\omega_{\delta h_c=0}$	$\omega_{\delta h_c=0.1}^d$	$\omega_{\delta h_c=0.1}$	$(\omega_{\delta h_c=0} + \Omega_{\delta h_c=0.1}^d)$	erreur (%)
0,1	64,3664271	-0,65547738	63,7391884	63,7109497	0,04430358
	298,051035	-3,08765918	295,169873	294,963376	0,06995862
	748,836614	-5,32411409	743,906147	743,5125	0,05291628
	1406,59922	-6,23282985	1400,81986	1400,36639	0,03237216
	2288,01509	-7,13150066	2281,36897	2280,88359	0,02127606
0,6	65,7796882	-0,66042272	65,1454695	65,1192655	0,04022379
	300,545861	-3,23880021	297,524457	297,307061	0,07306841
	750,613734	-5,55987471	745,471795	745,05386	0,05606316
	1407,56282	-6,36624218	1401,66541	1401,19657	0,03344874
	2288,65995	-7,22925048	2281,92753	2281,43069	0,02177256
1	67,7086322	-0,64430348	67,0870079	67,0643287	0,03380558
	304,441252	-3,45998669	301,216192	300,981266	0,07799274
	753,754164	-5,97262891	748,24149	747,781535	0,06147139
	1409,24652	-6,60398971	1403,13919	1402,64253	0,03539635
	2289,80689	-7,40464035	2282,9198	2282,40225	0,02267052
1,5	70,1874838	-0,59042782	69,6143177	69,597056	0,02479627
	310,526552	-3,80268181	306,989789	306,72387	0,08662164
	759,585384	-6,72453253	753,394942	752,860852	0,07089114
	1412,36104	-7,05908628	1405,85264	1405,30196	0,03917086
	2291,97099	-7,74010002	2284,78806	2284,23089	0,0243863

TAB. 3.6 – Sensibilité des fréquences propres de la poutre encastree-libre par rapport à une perturbation $\delta h_c = 10\%$ de la couche viscoélastique à module constant.

η	$\eta_{\delta h_c=0}$	$\eta_{\delta h_c=0.1}^d$	$\eta_{\delta h_c=0.1}$	$(\eta_{\delta h_c=0} + \eta_{\delta h_c=0.1}^d)$	erreur (%)
0,1	0,28083492	0,00991482	0,29018783	0,29074974	0,19363752
	0,24162385	-0,00573525	0,23614663	0,2358886	0,10926735
	0,15314392	-0,00828936	0,14539672	0,14485456	0,37288238
	0,08804717	-0,0060067	0,08251445	0,08204047	0,57441573
	0,05646453	-0,00418484	0,05262843	0,05227969	0,66265497
0,6	0,2454124	0,00992822	0,2548445	0,25534062	0,19467612
	0,23157798	-0,00507387	0,2267259	0,22650411	0,09782134
	0,15173204	-0,00804726	0,14419764	0,14368478	0,35566755
	0,08774191	-0,00596417	0,08224657	0,08177773	0,57003378
	0,05637737	-0,00417045	0,05255379	0,05220692	0,66003797
1	0,2016454	0,00941202	0,21065016	0,21105742	0,19333507
	0,21703561	-0,00425741	0,2129533	0,2127782	0,08222071
	0,1493269	-0,00763616	0,14215394	0,14169075	0,32583929
	0,08725562	-0,00589466	0,08182082	0,08136095	0,56204051
	0,056234	-0,00414614	0,05243143	0,05208785	0,65529445
1,5	0,15260553	0,00825048	0,16054538	0,16085601	0,19348442
	0,19695116	-0,00328239	0,19377304	0,19366877	0,05381012
	0,14507962	-0,00691446	0,13854265	0,13816516	0,27247056
	0,0864629	-0,00577174	0,08113384	0,08069116	0,54561486
	0,05598957	-0,00410218	0,05222477	0,05188739	0,64600427

TAB. 3.7 – Sensibilité des facteurs de perte de la poutre encastree-libre par rapport à une perturbation $\delta h_c = 10\%$ de la couche viscoélastique à module constant.

η	$\Omega_{\delta h_f=0}$	$\omega_{\delta h_f=0.1}^d$	$\omega_{\delta h_f=0.1}$	$(\omega_{\delta h_f=0} + \omega_{\delta h_f=0.1}^d)$	erreur (%)
0,1	64,3664271	5,27647092	69,6009917	69,642898	0,06020932
	298,051035	25,67217	323,670142	323,723205	0,01639407
	748,836614	68,7223278	817,563169	817,558941	0,00051706
	1406,59921	134,498545	1541,14148	1541,09776	0,00283683
	2288,01509	223,007374	2511,08489	2511,02246	0,00248638
0,6	65,7796882	5,38878156	71,1232095	71,1684698	0,06363651
	300,545861	25,71245	326,201956	326,258311	0,01727596
	750,613733	68,6131278	819,238244	819,22686	0,00138954
	1407,5628	134,432807	1542,04546	1541,99561	0,00323261
	2288,65994	222,954801	2511,68198	2511,61474	0,00267682
1	67,7086321	5,56359777	73,2238066	73,2722298	0,06613045
	304,441252	25,7938365	330,174976	330,235089	0,01820629
	753,754164	68,4244179	822,201729	822,178582	0,00281527
	1409,24651	134,313123	1543,62097	1543,55963	0,00397331
	2289,80689	222,859668	2512,74262	2512,66656	0,00302692
1,5	70,1874837	5,81743026	75,9543449	76,004914	0,06657825
	310,526552	25,9323536	336,396761	336,458905	0,01847344
	759,585385	68,0905705	827,717113	827,675955	0,00497251
	1412,36105	134,075982	1546,52123	1546,43703	0,00544475
	2291,97099	222,675389	2514,73947	2514,64637	0,00370188

TAB. 3.8 – Sensibilité des fréquences propres d’une poutre encastree-libre par rapport à une perturbation $\delta h_f = 10\%$ de la couche viscoélastique à module constant.

η	$\eta_{\delta h_f=0}$	$\eta_{\delta h_f=0.1}^d$	$\eta_{\delta h_f=0.1}$	$(\eta_{\delta h_f=0} + \eta_{\delta h_f=0.1}^d)$	erreur (%)
0,1	0,28083492	0,00662724	0,28694672	0,28746216	0,17963111
	0,24162385	-0,01039833	0,23173186	0,23122552	0,21850161
	0,15314392	-0,0121322	0,1419026	0,14101172	0,62781136
	0,08804717	-0,00845064	0,08033919	0,07959653	0,92439738
	0,05646453	-0,00581708	0,05118889	0,05064745	1,05772254
0,6	0,2454124	0,00704822	0,25200162	0,25246061	0,18213988
	0,23157798	-0,00944549	0,22257617	0,22213249	0,19933746
	0,15173204	-0,01183093	0,14075085	0,13990111	0,60372045
	0,08774191	-0,00839587	0,08008121	0,07934604	0,91802614
	0,05637738	-0,00579921	0,051117	0,05057817	1,05411057
1	0,2016454	0,00701503	0,20827574	0,20866043	0,18470366
	0,21703561	-0,00822429	0,20917697	0,20881132	0,17480278
	0,1493269	-0,0113184	0,13878854	0,1380085	0,56203082
	0,08725562	-0,00830625	0,07967165	0,07894937	0,90656808
	0,056234	-0,00576903	0,05099924	0,05046496	1,04762442
1,5	0,15260553	0,0064049	0,15869895	0,15901043	0,19627595
	0,19695116	-0,0067273	0,19048547	0,19022387	0,13733623
	0,14507962	-0,0104177	0,13532178	0,13466192	0,48762089
	0,0864629	-0,00814906	0,07901195	0,07831384	0,88355518
	0,05598957	-0,00571472	0,05080063	0,05027486	1,03498074

TAB. 3.9 – Sensibilité des facteurs de perte de la poutre encastree-libre par rapport à une perturbation $\delta h_f = 10\%$ de la couche viscoélastique à module constante.

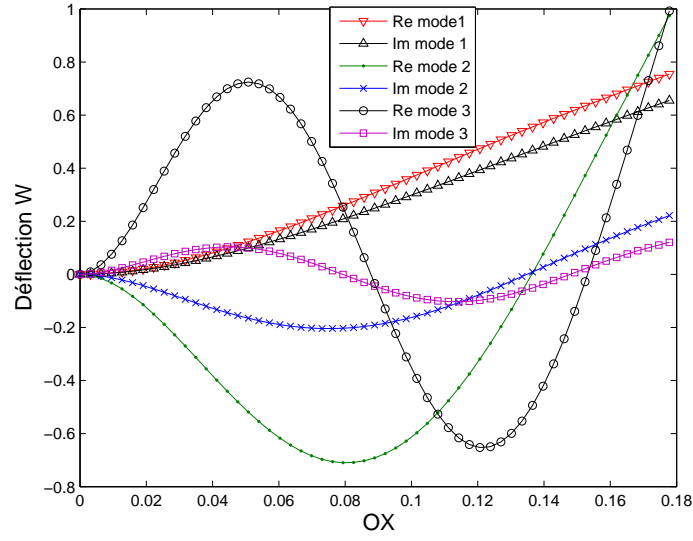


FIG. 3.5 – Partie réelle et imaginaire des trois premiers modes de la poutre viscoélastique à module constant avec un facteur de perte $\eta = 1.5$

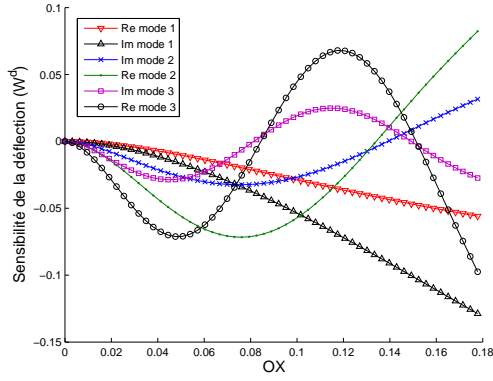


FIG. 3.6 – Sensibilité des trois premiers mode de la poutre viscoélastique à module constant avec un facteur de perte $\eta = 1.5$ pour une perturbation de $\delta h_f = 10\%$

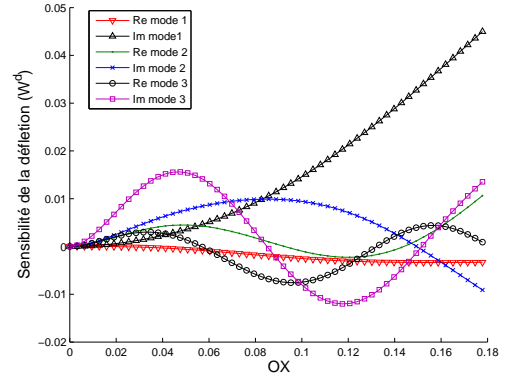


FIG. 3.7 – Sensibilité des trois premiers mode de la poutre viscoélastique à module constant avec un facteur de perte $\eta = 1.5$ pour une perturbation de $\delta h_c = 10\%$

$\eta = 1.5$					
Mode	$\omega_{\delta E_c=0}$	$\omega_{\delta E_c=0.1}^d$	$\omega_{\delta E_c=0.1}$	$(\omega_{\delta E_c=0} + \omega_{\delta E_c=0.1}^d)$	erreur (%)
1	70,187484	0,89587357	71,0254079	71,0833576	0,08159019
2	310,526552	4,46150176	314,880595	314,988053	0,03412663
3	759,585386	7,29628544	766,825947	766,881671	0,00726692
4	1412,36103	7,10967596	1419,45995	1419,47071	0,00075779
5	2291,971	7,13102597	2299,09768	2299,10202	0,00018878

TAB. 3.10 – Sensibilité des fréquences propres par rapport à une perturbation $\delta E_c = 10\%$ de la couche viscoélastique à module constant.

$\eta = 1.5$					
Mode	$\eta_{\delta E_c=0}$	$\eta_{\delta E_c=0.1}^d$	$\eta_{\delta E_c=0.1}$	$(\eta_{\delta E_c=0} + \eta_{\delta E_c=0.1}^d)$	erreur (%)
1	0,15260553	-0,00732769	0,14562059	0,14527784	0,235371
2	0,19695116	0,00500487	0,20157747	0,20195603	0,18780242
3	0,14507962	0,00866613	0,15335043	0,15374576	0,25779346
4	0,0864629	0,00696048	0,09328696	0,09342338	0,14623352
5	0,05598957	0,00490854	0,06083795	0,06089811	0,09889251

TAB. 3.11 – Sensibilité des facteurs de perte de la poutre encastree-libre par rapport à une perturbation $\delta E_c^d = 0.1$ de la couche viscoélastique à module constant.

Faces élastiques	Module de Young $E_f = 6.9^{10} Pa$
	Coefficient de Poisson $\nu_f = 0.3$
	Masse volumique $\rho_f = 2766 Kg.m^{-3}$
	Epaisseur $h_f = 0.762$
Cœur polymère	Module d'Young, loi (3.90), $E_c = 2670.08 \times 10^3 Pa$
	Coefficient de Poisson $\nu_c = 0.49$
	Masse volumique $\rho_c = 999 Kg.m^{-3}$
	Epaisseur $h_c = 0.254 mm$
Dimension	Longueur $L = 348 mm$
	Largeur $l = 304.8 mm$

TAB. 3.12 – Propriétés mécaniques et géométriques de la plaque sandwich à module constant.

Mode	$\omega_{\delta E_c=0}$	$\omega_{\delta E_c=0.1}^d$	$\omega_{\delta E_c=0.1}$	$(\omega_{\delta E_c=0} + \omega_{\delta E_c=0.1}^d)$	erreur (%)
1	59,9406461	1,23976034	61,1120797	61,1804065	0,11180569
2	114,284411	2,82715805	117,030314	117,111569	0,06943125
2	129,208781	3,13699581	132,273152	132,345776	0,05490516
3	175,058699	3,61117968	178,61138	178,669879	0,03275197

TAB. 3.13 – Sensibilité des fréquences propres de la plaque sandwich viscoélastique à module constant par rapport à une perturbation du module de cisaillement G_0 de 10% de la couche viscoélastique à module constant

Mode	$\eta_{\delta G_c=0}$	$\eta_{\delta G_c=0.1}^d$	$\eta_{\delta G_c=0.1}$	$(\eta_{\delta G_c=0} + \eta_{\delta G_c=0.1}^d)$	Erreur (%)
1	0,24032605	-0,00865431	0,23192446	0,23167173	0,1089697
2	0,30458011	-0,0001107	0,30376662	0,30446942	0,23136142
3	0,30908117	0,0027787	0,31099077	0,31185987	0,27946135
4	0,28130199	0,00761492	0,28800957	0,28891691	0,31503808

TAB. 3.14 – Sensibilité des facteurs de perte de la plaque sandwich viscoélastique à module constant par rapport à une perturbation du module de cisaillement G_c de 10% de la couche viscoélastique à module constant

Ces résultats montrent que dans le cas des structures plaques viscoélastique à module constant, les sensibilités prédisent bien les fréquences propres et les facteurs de perte. Pour cette plaque, les figures 3.8, 3.9, 3.10, 3.11, 3.12 et 3.13 présentent la partie réelle et imaginaire des modes propres et leur sensibilités. Ces figures montrent que la perturbation du module de cisaillement affecte plus la partie imaginaire que la partie réelle du mode propre et donc l'amortissement de la structure.

3.7 Analyse de sensibilité du modèle viscoélastique faiblement amorti dépendant de la fréquence (PVB)

3.7.1 Analyse de sensibilité de la poutre viscoélastique en PBV par rapport au module de cisaillement

Pour étudier la sensibilité des modèles dépendant de la fréquence, nous avons d'abord considéré une poutre sandwich dont le cœur est en PVB. La loi de comportement du matériau PVB est donnée par l'équation (3.90). Les faces sont en aluminium.

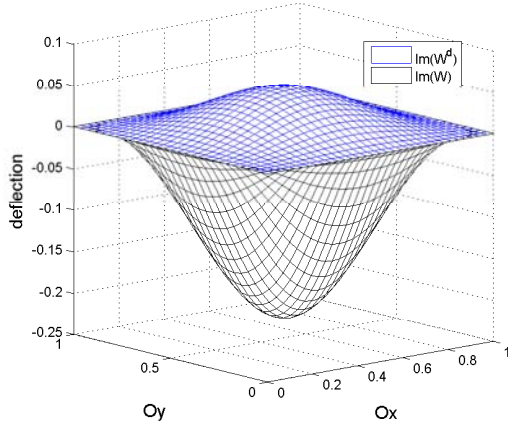


FIG. 3.8 – Partie imaginaire du mode 1 et sa sensibilité pour la plaque sandwich viscoélastique à module constant

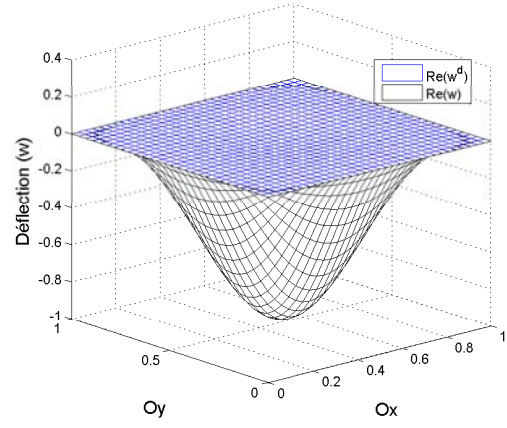


FIG. 3.9 – Partie réelle du mode 1 et sa sensibilité pour la plaque sandwich viscoélastique à module constant

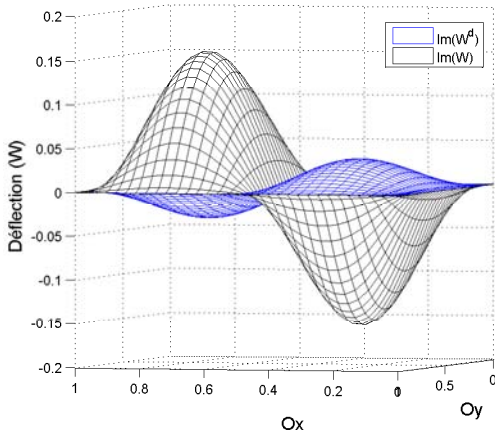


FIG. 3.10 – Partie imaginaire du mode 2 et sa sensibilité pour la plaque sandwich viscoélastique à module constant

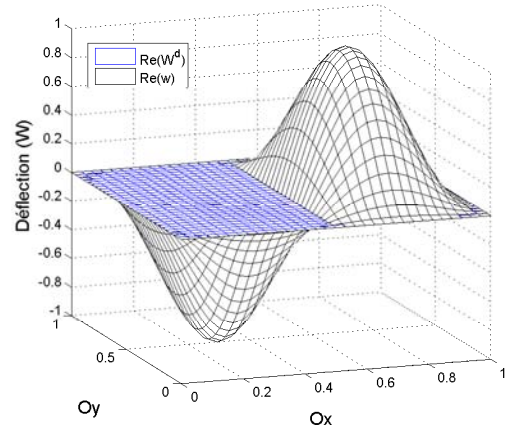


FIG. 3.11 – Partie réelle du mode 2 et sa sensibilité pour la plaque sandwich viscoélastique à module constant

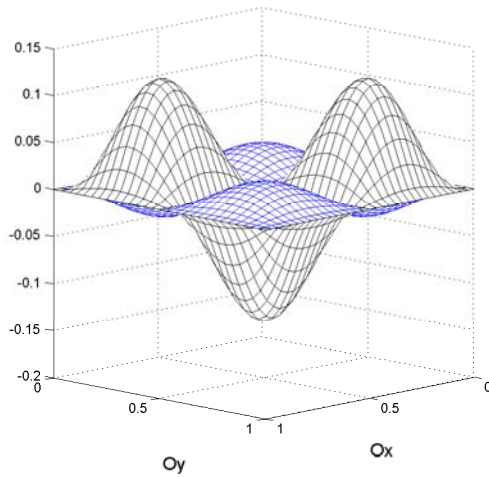


FIG. 3.12 – Partie imaginaire du mode 4 et sa sensibilité pour la plaque sandwich viscoélastique à module constant

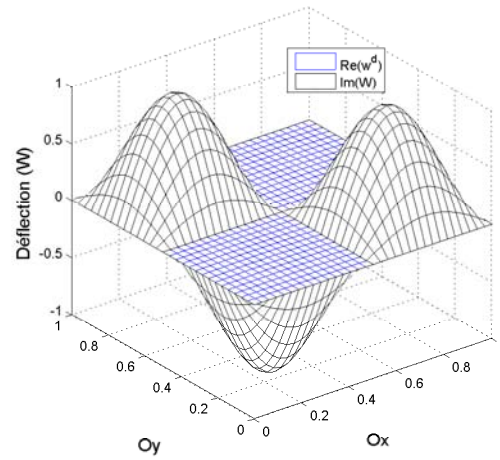


FIG. 3.13 – Partie réelle du mode 4 et sa sensibilité pour la plaque sandwich viscoélastique à module constant

Les caractéristiques de la poutre sont données dans le tableau 3.15. Les propriétés modales de cette poutre ont été étudiées dans [60].

Faces élastiques	Module de Young $E_f = 6.9^{10} Pa$
	Coefficient de Poisson $\nu_f = 0.3$
	Masse volumique $\rho_f = 2766 Kg.m^{-3}$
	Epaisseur $h_f = 1.524 mm$
Cœur polymère PVB	Module d'Young, loi (3.90), $E_0 = 1794 \times 10^3 Pa$
	Coefficient de Poisson $\nu_c = 0.4$
	Masse volumique $\rho_c = 999 Kg.m^{-3}$
	Epaisseur $h_c = 0.127 mm$
Dimension	Longueur $L = 177.8 mm$
	Largeur $l = 12.7 mm$

TAB. 3.15 – Propriétés mécaniques et géométriques de la poutre sandwich Aluminium/PVB/Aluminium.

Mode	$\omega_{\delta E_f=0}$	$\omega_{\delta E_f=0.2}^d$	$\omega_{\delta E_c=0.2}$	$(\omega_{\delta E_f=0} + \omega_{\delta E_f=0.2}^d)$	Erreurs (%)
1	82,0656713	8,14892995	89,8400265	90,2146013	0,41693534
2	506,285317	48,8558355	552,682262	555,141152	0,44490123
3	1387,91815	128,870981	1509,62851	1516,78913	0,4743303
4	2646,38065	234,234116	2866,10217	2880,61477	0,50635306

TAB. 3.16 – Sensibilité des fréquences propres de la poutre sandwich en PVB par rapport à une perturbation du module de cisaillement G_f de 20%

Mode	$\eta_{\delta E_f=0}$	$\eta_{\delta E_f=0.2}^d$	$\eta_{\delta E_c=0.2}$	$(\eta_{\delta E_f=0} + \eta_{\delta E_f=0.2}^d)$	Erreurs (%)
1	0,00140375	0,00023054	0,0016057	0,00163429	1,78066676
2	0,00518539	0,00074894	0,00596024	0,00593433	0,4347023
3	0,0088319	0,00116999	0,01005762	0,01000189	0,55412091
4	0,01274523	0,00147845	0,01432436	0,01422367	0,70289739

TAB. 3.17 – Sensibilité des facteurs de perte de la poutre sandwich en PVB par rapport à une perturbation du module de cisaillement G_f de 20%

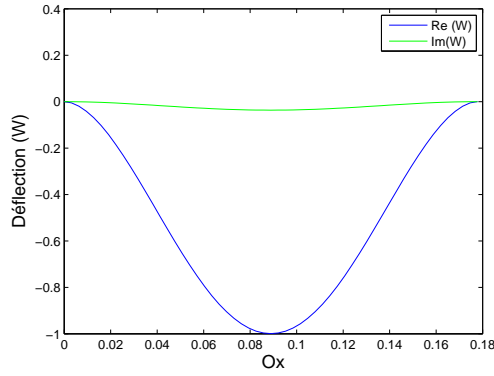


FIG. 3.14 – Mode 1 de la poutre sandwich bi-encastrée en PVB et sa sensibilité pour une perturbation de 20% du module de cisaillement G_f

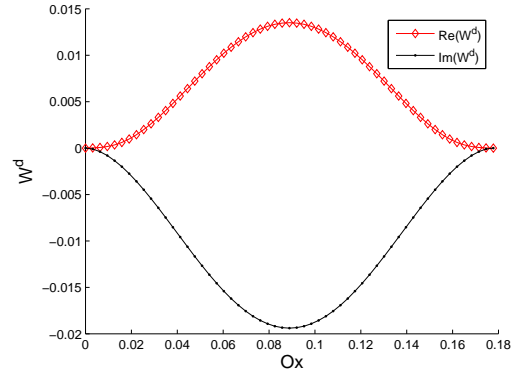


FIG. 3.15 – Sensibilité du mode 1 de la poutre sandwich bi-encastrée en PVB pour une perturbation de 20% du module de cisaillement G_f

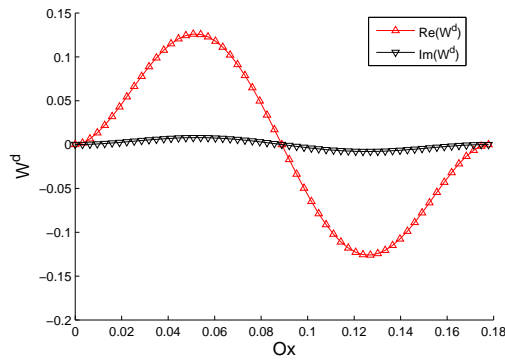


FIG. 3.16 – Sensibilité du mode 2 de la poutre viscoélastique avec un cœur en PVB par rapport à une perturbation de 20% du module de cisaillement G_f

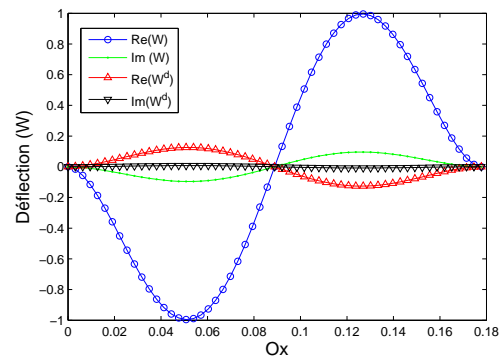


FIG. 3.17 – Déflexions de la poutre et sensibilités

Mode	$\omega_{\delta K_e=0}$	$\omega_{\delta K_e=0.1}^d$	$\omega_{\delta K_e=0.1}$	$\omega_{\delta K_e=0} + \omega_{\delta K_e=0.1}^d$
1	82,0656711	0,02722943	82,0951424	82,0929005
2	506,285305	0,03236072	506,305245	506,317666
3	1387,9181	-0,05159893	1388,32621	1387,8665
4	1628,83903	4,29708145	1629,38696	1633,13612
5	2646,38051	1,27150339	2647,06358	2647,65201

TAB. 3.18 – Evolution des fréquences propres de la poutre avec un cœur en PVB pour une perturbation de la matrice de rigidité élémentaire aux point $x = L/2$

3.7.2 Sensibilité par rapport à la matrice de rigidité élémentaire

Nous reprenons la poutre à cœur en PVB et dont les caractéristiques sont données dans le tableau 3.15. La perturbation étant localisée, elle n’affecte pas considérablement la fréquence propre de la structure. Le tableau 3.18 donne pour la poutre encastree-libre l’évolution des fréquences propres pour une perturbation de 10% d’une matrice de rigidité élémentaire. On note que les variations sont faibles, pour des perturbations de faible valeur les modes perturbés et les modes non perturbés se superposent figure 3.19, et 3.18. L’effet de la perturbation est très visible sur les courbes de sensibilité des modes propres. Les figures 3.18 et 3.20 présentent la partie réelle et la partie imaginaire de sensibilité des modes 1 et 2.

3.7.3 Etude de sensibilité du modèle viscoélastique fortement amorti (ISD112)

Pour montrer l’efficacité de notre approche dans le cas du modèle de Maxwell généralisé, nous considérons les structures poutre et plaque sandwich dont la couche viscoélastique est constituée de ISD112 dont la loi de comportement est donnée par la

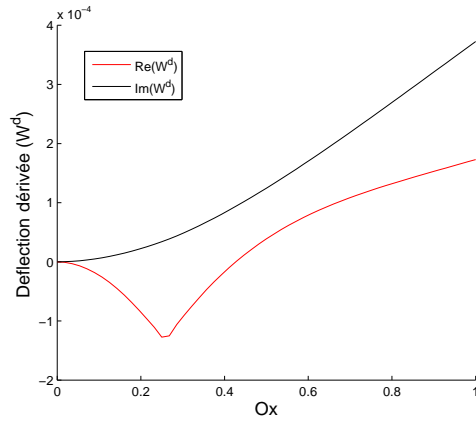


FIG. 3.18 – Sensibilité du mode 1 de la poutre avec un cœur en PVB pour une perturbation 10% de la matrice de rigidité élémentaire au point $x = 0.25L$

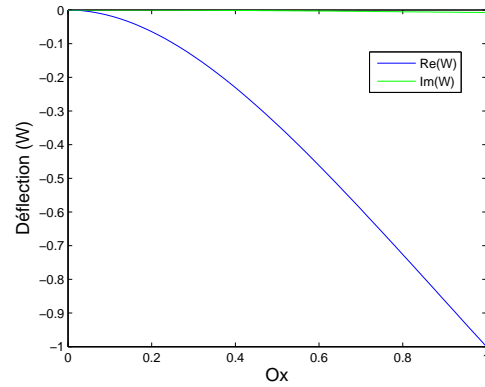


FIG. 3.19 – Mode 1 de la poutre avec un cœur en PVB

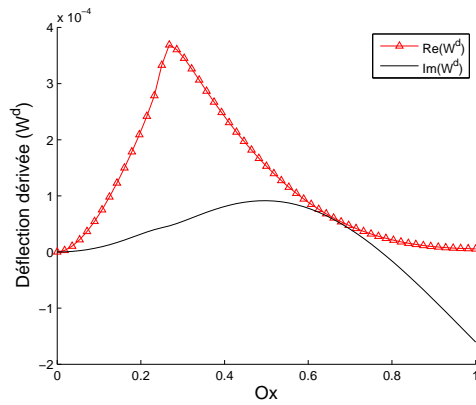


FIG. 3.20 – Sensibilité du mode 2 de la poutre avec un cœur en PVB pour une perturbation 10% de la matrice de rigidité élémentaire au point $x = 0.25L$

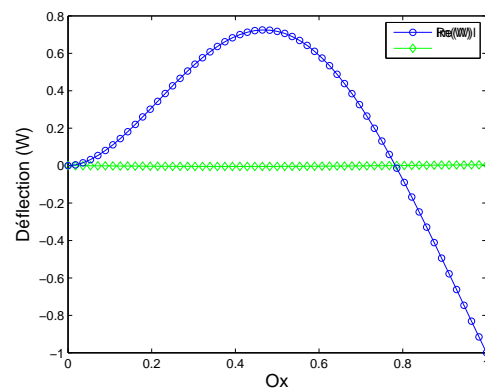


FIG. 3.21 – Mode 2 de la poutre avec un cœur en PVB

Faces élastiques	Module de Young $E_f = 6.89^{10} Pa$
	Coefficient de Poisson $\nu_f = 0.3$
	Masse volumique $\rho_f = 2766 Kg.m^{-3}$
	Epaisseur $h_f = 1.524$
Cœur IDS112	Module d'Young, loi (3.91), $E_c = 2670.08 \times 10^3 Pa$
	Coefficient de Poisson $\nu_c = 0.5$
	Masse volumique $\rho_c = 1600 Kg.m^{-3}$
	Epaisseur $h_c = 0.127 mm$
Dimension	Longueur $L = 177.8 mm$
	Largeur $l = 12.7 mm$

TAB. 3.19 – Propriétés mécaniques et géométriques d'une poutre sandwich a cœur en IDS112.

formule (3.91). Ces structures ont fait l'objet de plusieurs études [138, 4]. Les caractéristiques de la poutre et de la plaque sont données dans les tableaux 3.19 et 3.22.

Cas de la poutre

La poutre est prise à une température de $27^\circ C$. Les résultats de sensibilité calculés par rapport au module de Young E_f et par rapport à la perturbation de la matrice de rigidité élémentaire sont présentés dans les tableaux 3.20 et 3.21. Ils montrent que notre approche prédit bien les fréquences propres de la structure pour une modification du module de Young de la couche des faces.

cas de la plaque en IDS112

Pour vérifier la précision de l'approximation des modes propres, nous avons procédé à la comparaison de la norme. Le tableau 3.25 présente le résultat pour les cinq premiers modes.

Mode	$\omega_{\delta E_f=0}$	$\omega_{\delta E_f=0.1}^d$	$\omega_{\delta E_f=0.1}$	$(\omega_{\delta E_f=0} + \omega_{\delta E_f=0.1}^d)$	erreur (%)
1	70,1874838	2,61350048	72,7016525	72,8009842	0,13662927
2	310,526552	11,0648258	321,327127	321,591377	0,08223721
3	759,585384	30,6829838	789,656148	790,268368	0,07752991
4	1412,36103	63,5083764	1474,50846	1475,86941	0,09229857
5	2291,97098	107,467524	2397,03673	2399,4385	0,10019739

TAB. 3.20 – Sensibilité des fréquences propres de la poutre encastree-libre en ISD112 par rapport à une perturbation $\delta E_f = 20\%$ de l'épaisseur des faces

Mode	$\omega_{\delta E_f=0}$	$\omega_{\delta E_f=0.1}^d$	$\omega_{\delta E_f=0.1}$	$(\omega_{\delta E_f=0} + \omega_{\delta E_f=0.1}^d)$	erreur (%)
1	0,15260553	0,00732769	0,15958137	0,15993322	0,22048105
2	0,19695116	-0,00500487	0,19205472	0,19194629	0,0564579
3	0,14507962	-0,00866613	0,13685442	0,13641349	0,32218788
4	0,0864629	-0,00696048	0,08001996	0,07950243	0,64675543
5	0,05598957	-0,00490854	0,05147672	0,05108104	0,76866042

TAB. 3.21 – Sensibilité des facteurs de perte de la poutre sandwich encastree-libre en ISD112 rapport à une perturbation $\delta E_f = 10\%$ de l'épaisseur des faces

Faces élastiques	Module de Young $E_f = 6.89^{10}Pa$
	Coefficient de Poisson $\nu_f = 0.3$
	Masse volumique $\rho_f = 2740Kg.m^{-3}$
	Epaisseur $h_f = 0.762$
Cœur IDS112	Module d'Young, loi (3.90), $E_c = 2670.08 \times 10^3Pa$
	Coefficient de Poisson $\nu_c = 0.5$
	Masse volumique $\rho_c = 1600Kg.m^{-3}$
Dimension	Epaisseur $h_c = 0.254mm$
	Longueur $L = 348mm$
	Largeur $l = 304.8mm$

TAB. 3.22 – Propriétés mécaniques et géométriques de la plaque sandwich a cœur en IDS112.

Conditions limites	$\omega_{\delta E_f=0}$	$\omega_{\delta E_f=0.2}^d$	$\omega_{\delta E_c=0.2}$	$(\omega_{\delta E_f=0} + \omega_{\delta E_f=0.2}^d)$	Erreurs (%)
CFCF	46,2850669	1,40747083	47,6827853	47,6925377	0,02045282
	57,9900444	1,9482101	59,8861389	59,9382545	0,08702461
	111,766466	3,76182697	115,328268	115,528293	0,17343935
	116,701912	3,91631183	120,528147	120,618223	0,0747346
CSCS	66,364885	2,12074168	68,4233827	68,4856267	0,09096896
	132,502467	4,04261231	136,492801	136,54508	0,03830172
	136,097809	4,57769497	140,570396	140,675504	0,07477259
	194,800766	6,48737692	201,32216	201,288143	-0,0168968
SSSS	53,4911585	1,70857945	55,1394449	55,1997379	0,10934645
	109,594261	3,31881058	112,8218	112,913071	0,08089854
	125,675659	3,81295985	129,390772	129,488619	0,07562075

TAB. 3.23 – Sensibilité des fréquences propres de la plaque sandwich en IDS112 à 27°C par rapport à une perturbation du module de cisaillement G_f de 20%

Conditions limites	$\eta_{\delta E_f=0}$	$\eta_{\delta E_f=0.2}^d$	$\eta_{\delta E_c=0.2}$	$(\eta_{\delta E_f=0} + \eta_{\delta E_f=0.2}^d)$	Erreurs (%)
CFCF	0,24423287	0,00520737	0,24396968	0,24944024	2,24231115
	0,2328597	-0,00092149	0,23166869	0,23193821	0,11633709
	0,2708639	0,0064725	0,27361056	0,2773364	1,36173147
	0,27987118	-0,00576209	0,27411653	0,27410909	-0,00271383
CSCS	0,26192189	0,00052736	0,26203086	0,26244925	0,15967223
	0,32528184	0,0071949	0,32189125	0,33247674	3,2885297
	0,28906242	-0,00529676	0,2837237	0,28376567	0,01479254
	0,32516665	0,00029044	0,31849328	0,32545709	2,18648364
SSSS	0,23959682	0,00604353	0,2450316	0,24564035	0,24843875
	0,31291022	-0,00058435	0,31184944	0,31232587	0,15277549
	0,32682515	-0,00116927	0,32427491	0,32565588	0,42586426
	0,33742872	0,00616013	0,33265105	0,34358885	3,28807121

TAB. 3.24 – Sensibilité des facteurs de perte de la plaque sandwich en IDS112 à 27°C par rapport à une perturbation du module de cisaillement G_f de 20%

Mode	$ W_{\delta G_f=0} + W_{\delta G_f=0.2}^d $	$ W_{\delta G_f=0.2} $	Erreur (%)	
1	47,8506333	45,8515087	45,9517877	0,04359997
2	47,9104593	45,5427386	45,6872728	0,04776614
3	47,4667424	45,1056514	45,2512974	0,04926628
4	46,6263525	44,2230581	44,4535826	0,05148434
5	46,2496857	43,8032300	44,1208015	0,05217717

TAB. 3.25 – Normes des cinq premiers modes propres pour une perturbation de 20% du module de cisaillement G_f pour une plaque sandwich en IDS112 à 27°C

Dans le tableau 3.25, $W_{\delta G_f=0}$ représente le mode propre sans la perturbation, $W_{\delta G_f=0.2}^d$ la sensibilité pour une perturbation de 20% du module de cisaillement des faces, $W_{\delta G_f=0.2}$ représente le mode propre obtenu par un calcul direct en prenant un module de cisaillement $G_f = (1 + 0.2)G_f$. Ce tableau montre que pour une perturbation de 20%, les dérivées de premier ordre calculées approchent bien la réponse de la

structure pour ces cinq modes.

3.7.4 Sensibilité par rapport à la matrice de rigidité élémentaire de la plaque sandwich en ISD112

L'étude est réalisée à une température de 20°C . Les paramètres relatifs à cette température sont donnés dans le tableau 3.2. Le calcul de sensibilité est tout d'abord mené sur une plaque encastree de deux cotés (CFCF). La présence d'une perturbation locale se traduit par une déflexion dérivée w^d présentant une déformation locale plus accentuée dans la zone de perturbation. On voit remarque bien sur la figure 3.22 l'influence locale de la perturbation appliquée au point $(L/2, l/2)$ de la plaque. Sur les figures, les dimensions de la plaque sont normalisées pour mettre en évidence le positionnement de l'élément ou des éléments perturbés. Comme dans le cas de la poutre PVB, cet effet local est plus prononcé sur la partie réelle de la sensibilité de la déflexion. Ceci peut être expliqué par le fait que les paramètres mécaniques des faces élastiques sont plus importants que ceux de la couche viscoélastique. En utilisant un calcul direct, ces effets sont difficiles à déceler (voir figure 3.23). Le même test effectué sur une plaque dont les quatre cotés sont encastres (CCCC) confirme la présence d'une zone de forte déformation locale sur la partie réelle de la dérivée du mode 1. Sur la figure 3.25 qui représente la partie imaginaire, cette zone est moins prononcée pour la partie imaginaire. Elle est cerclée en rouge.

Pour aller plus loin dans notre étude, nous avons appliqué la perturbation en deux points différents de la plaque. Sur les figures 3.26 et 3.27, les perturbations sont appliquées aux points $(L/4, l/2)$ et $(3L/4, L/2)$. Le caractère local de l'influence de la perturbation peut toujours être observé même si leurs effets se superposent. La partie réelle de la déflexion dérivée W^d reste toujours la plus affectée. L'effet local de la perturbation peut être utile dans la détection des défauts comme la délamination ou des défauts de collage dans les structures sandwich. Il serait intéressant de confirmer cela par une étude qui associe nos résultats avec des données expérimentales. Par ailleurs,

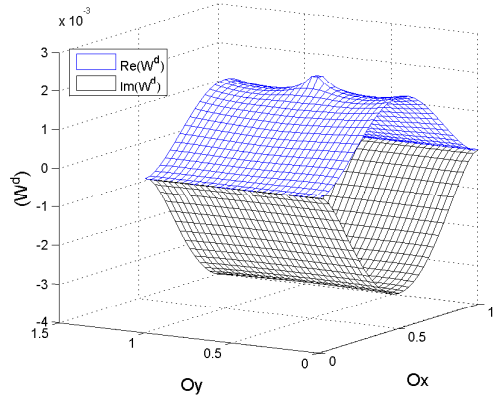


FIG. 3.22 – Sensibilité du mode 1 de la plaque avec un cœur en ISD112 pour une perturbation 20% de la matrice de rigidité élémentaire au point $p(0.5L, 0.5l)$ avec la condition limite CFCF.

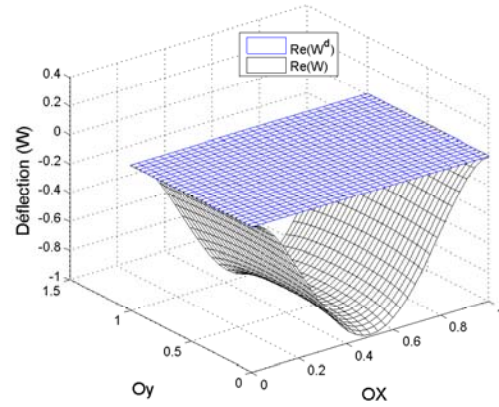


FIG. 3.23 – Partie réelle du mode 1 de la plaque avec sa sensibilité

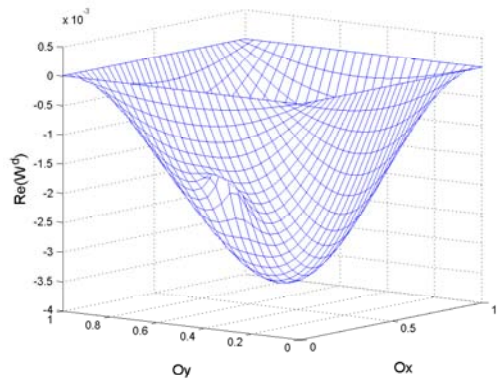


FIG. 3.24 – Sensibilité de la partie réelle du mode 1 de la déflexion par rapport à une perturbations de 40% de la matrice de rigidité aux points $(L/4, l/2)$

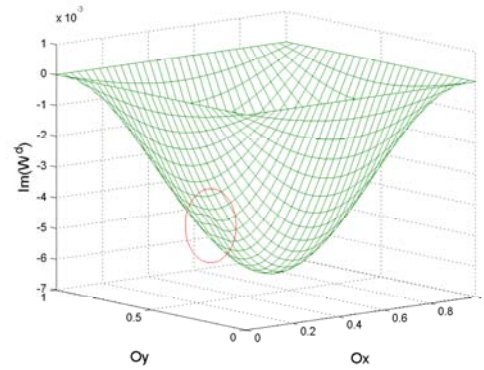


FIG. 3.25 – Sensibilité de la Partie imaginaire du mode 1 de la déflexion par rapport à une perturbations de 40% de la matrice de rigidité aux points $(L/4, l/2)$

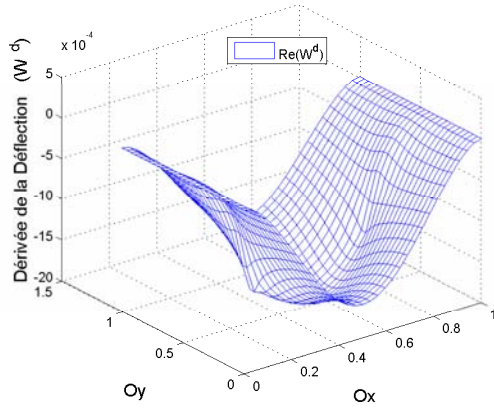


FIG. 3.26 – Partie réelle de la dérivée du mode 1 de la déflexion de la plaque pour une double perturbations aux points $(L/4, l/2)$ et $(3L/4, l/2)$

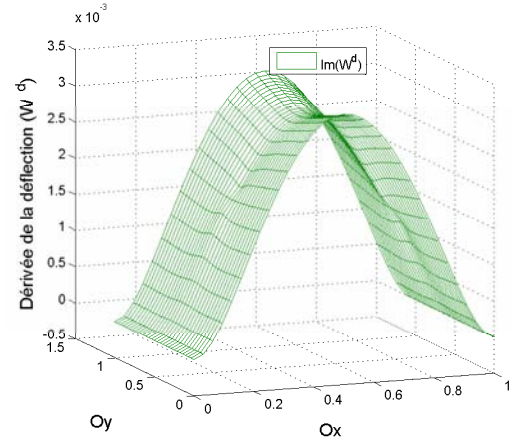


FIG. 3.27 – Partie Imaginaire de la dérivée du mode 1 de la déflexion de la plaque pour une double perturbations aux points $(L/4, l/2)$ et $(3L/4, l/2)$

en plaçant ces perturbations en des points bien précis, on peut agir sur l'amplitude d'un mode propre bien défini. Par exemple en appliquant la perturbation aux points $(L/4, l/2)$ et $(3L/4, L/2)$, l'amplitude du mode 2 de la plaque est directement affecté (figures 3.28 et 3.29). Ces résultats confirment les résultats obtenus dans le cas statique présentés au chapitre 2

3.8 Estimation des temps de calcul

Notre algorithme de calcul de sensibilité peut être subdivisé en deux parties. la première partie calcule la sensibilité de la solution propre réelle, la second calcule la sensibilité la solution propre non-linéaire ou complexe par l'homotopie. les temps de calcul moyens estimés pour ces deux parties sont résumés dans le tableau 3.26. La deuxième colonne donne le temps de calcul pour la solution du problème réel La deuxième colonne donne le temps de calcul de la solution complexe et sa sensibilité. La première ligne du tableau donne le temps du calcul direct sans la sensibilité. La deuxième ligne

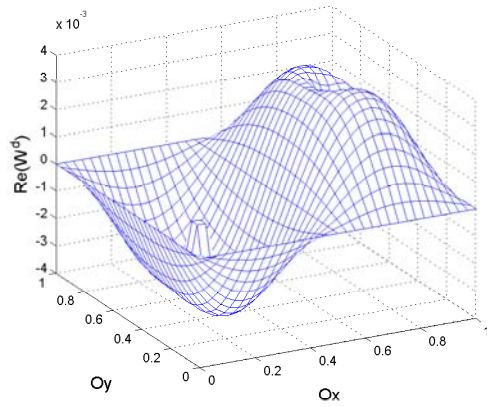


FIG. 3.28 – Sensibilité de la partie réelle du mode 2 de la déflexion par rapport à une perturbations de 0.4 de la matrice de rigidité aux points $(L/4, l/2)$, $(3L/4, l/2)$

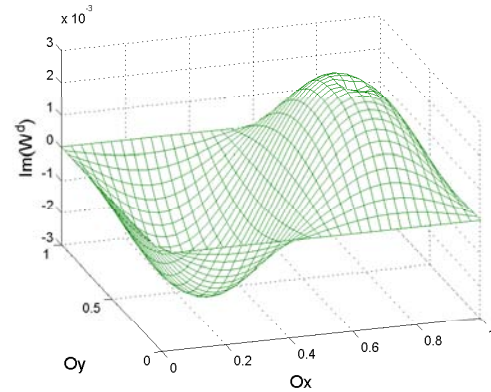


FIG. 3.29 – Sensibilité de la partie imaginaire du mode 2 de la déflexion par rapport à une perturbations de 0.4 de la matrice de rigidité aux points $(L/4, l/2)$, $(3L/4, l/2)$

du tableau donne les temps de notre approche et la dernière ligne donne les temps du calcul par la méthodes des différences finies. Par rapport à la Méthode des différences finies, notre approche permet de gagner 25% du temps de calcul. ce gain de temps est essentiellement du à la méthode utilisée pour calculer la solution réel et à la mise en donne du modèle élément finis qui est fait une fois dans notre approche alors qu'elle est répétée dans la méthode de différence finis.

	Problème réel	Problème complexe
Calcul direct	15,33s	27,47s
Calcul direct + sensibilité	15,5s	42,64s
Différences finies	30,74s	56,22s

TAB. 3.26 – Temps de calcul pour une plaque constitué de 896 éléments

3.9 Conclusion

Dans ce chapitre, nous proposons une méthode d'analyse de sensibilité des problèmes mécaniques aux valeurs propres non-linéaires basée la différentiation automatique, la MAN et la technique d'homotopie. Dans cette approche, les non-linéarités matérielles de type viscoélastique ont été prises en compte. Comme dans le cas statique, la modélisation de la vibration libre non-linéaire de la structure sandwich est effectuée par la méthode des éléments finis et le modèle de type zigzag. Des résultats sont produits pour des structures poutre et plaque sandwich à cœur viscoélastique. Pour le cœur viscoélastique, trois types de lois complexes ont été utilisés : une loi viscoélastique à module complexe constant, une loi viscoélastique induisant un amortissement faible et une loi viscoélastique induisant un amortissement élevé. La prise en compte de ces lois est facilitée par l'usage de la différentiation par transformation de source et le couplage Diamant-ADiMat. Les résultats obtenus par un calcul de sensibilité sont comparés à ceux obtenus par un calcul direct. Ces résultats montrent que l'approximation de premier ordre permet de prédire de manière efficace les fréquences propres, les facteurs de perte et les modes propres des structures sandwich viscoélastiques lorsque les paramètres de modélisation sont modifiés. Les sensibilités calculées par la DA permettent d'obtenir l'impact "exact" d'une perturbation matérielle ou géométrique localisée ou non sur les propriétés amortissantes et de manière plus visible sur les modes propres. Notre approche permet de lever les difficulté induit par la dépendance en fréquence des lois de comportement viscoélastique, de réduire le temps de calcul. elle peut être utilisée pour résoudre d'autre type de problèmes non-linéaires.

Conclusion générale et perspectives

De nos jours, les matériaux sandwich viscoélastiques sont utilisés dans des domaines de la mécanique tels que l'aéronautique, le génie civil, ou l'automobile pour leurs propriétés amortissantes. La dépendance en fréquence et leur modélisation en grande déplacement induit des non-linéarités dont la modélisation nécessite des outils numériques adéquats. Dans cette thèse, la modélisation des structures sandwich est basée sur la méthode des éléments finis et des lois viscoélastiques. Les problèmes non-linéaires discrets sont résolus par la méthode de continuation MAN (Méthode Asymptotique Numérique) dans sa version Diamant qui utilise les techniques de Différentiation Automatique (DA). Les informations obtenues par résolution directe du modèle peuvent être complétées par des analyses de sensibilité. De telles études sont de plus en plus utilisées dans les processus de modélisation afin de mieux comprendre l'impact des différents paramètres de modélisation sur le comportement mécanique du modèle pour, par exemple, simplifier ou améliorer le modèle mécanique.

Dans cette thèse, nous montrons l'intérêt de la différentiation automatique dans les codes éléments finis mécaniques pour le calcul de sensibilité des problèmes non-linéaires par rapport aux paramètres de la modélisation. Cette étude est abordée de manière générique en exploitant le potentiel de la DA utilisée, à un premier niveau, pour le calcul de la solution du problème non-linéaire par la MAN dans sa version Diamant et, à un second niveau pour le calcul de sensibilité.

Au chapitre 1, un état de l'art présente en détail la DA, la MAN et sa version Diamant. Les avantages et les inconvénients des différentes méthodes de calcul des dé-

rivées sont discutées. Une description du principe de la DA, ses modes de différentiation (linéaire tangent, adjoint, haut degré) et ses techniques d'implémentation (transformation de source et surcharge d'opérateur) est proposée pour, notamment, démontrer que la DA est générique, permettant d'occulter les étapes de différentiation à l'utilisateur tout en calculant les dérivées de manière efficaces à la précision machine près. C'est le cas pour le calcul des dérivées de haut degré apparaissant dans la MAN qui ont été rendu transparent à l'utilisateur dans la version Diamant de la MAN.

Au chapitre 2, nous avons exploité la généricité de Diamant pour proposer une méthode d'analyse de sensibilité des solutions des problèmes non-linéaires. Tout comme pour la résolution directe des problèmes non-linéaires, les aspects théoriques, numériques et algorithmiques que nous décrivons sont génériques, c'est-à-dire indépendant des équations non-linéaires à résoudre et en particulier du modèle éléments finis utilisé et de la loi viscoélastique. Les développements théoriques font apparaître des dérivées mixtes : de degré 1 par rapport aux paramètres de modélisation, de haut degré par rapport au paramètre de continuation. L'implémentation est réalisée en Matlab. L'originalité repose sur le couplage des outils de différentiation automatique - ADiMat et Diamant - pour le calcul de dérivées mixtes. Pour montrer l'efficacité de notre approche, nous avons étudié une poutre sandwich viscoélastique dans le cas des non-linéarités géométriques (cas statique). La modélisation de cette poutre est réalisée à l'aide de la méthode des éléments finis et d'un modèle de type zigzag. La sensibilité des solutions est étudiée par rapport à des perturbations de l'épaisseur et du module d'Young mais aussi par rapport à des perturbations d'éléments de la matrice de rigidité pour simuler un défaut dans le matériau. Cette étude a fait l'objet d'un article publié dans une revue internationale [15].

Pour l'étude du cas dynamique, au chapitre 3, le couplage Diamant-ADiMat est associé à la technique d'homotopie pour prendre en compte les non-linéarités matérielles induites par la dépendance en fréquence des lois viscoélastiques. Un algorithme a été implémenté pour le calcul des sensibilités des solutions propres complexes de vibration

des structures (poutre et plaque) sandwich viscoélastiques. Dans cette approche, la sensibilité des modes réels est d'abord calculée en différentiant le problème de valeur propre classique et la condition d'orthogonalité des modes propre par rapport à la matrice de masse. Les résultats du mode réel permettent d'initialiser l'homotopie pour le calcul de la sensibilité des modes complexes. L'aspect générique de la méthode proposée permet de prendre en compte plusieurs lois viscoélastiques sans développement supplémentaire. La seule tâche qui revient à l'utilisateur est de fournir la fonction implémentant le modèle viscoélastique. Comparé à l'approximation par différences finies souvent utilisée en mécanique pour le calcul de sensibilité, notre approche fournit des résultats plus précis pour la prédiction de la réponse d'une structure sandwich en vibration libre lorsque les paramètres de conception sont modifiés. Cet algorithme permet de réduire le temps de calcul. Les résultats obtenus sont très précis. Ils montrent la pertinence de notre approche et de nos calculs, et notamment leurs capacités à fournir des informations complémentaires.

Le caractère générique des outils développés dans cette thèse ouvrent de nombreuses perspectives. En fonction de sa problématique de recherche, l'utilisateur fournit le modèle éléments finis et la loi de comportement, puis résout son problème non-linéaire avec Diamant. Pour procéder à une étude de sensibilité, l'utilisateur différentie son modèle, puis utilise une des implémentations que nous avons proposées. Les résultats de sensibilité peuvent aussi être associés aux données expérimentales pour résoudre des problèmes d'optimisation ou d'identification.

Annexe A

Matrices de comportement

Les matrices de comportement utilisées dans le chapitre 3 sont données pour les faces élastiques $i = 1, 3$:

$$[C_m] = \frac{E_f h_f}{1 - \nu_f^2} \begin{bmatrix} 1 & \nu_f & 0 \\ \nu_f & 1 & 0 \\ 0 & 0 & \frac{1 - \nu_f}{2} \end{bmatrix}, \quad [C_b] = \frac{h_f^2}{12} [C_m] \quad (\text{A.1})$$

où E_f , ν_f et h_f sont respectivement le module d'Young, le coefficient de Poisson et l'épaisseur des faces élastiques. Les matrices de comportement pour le cœur viscoélastique $i = 2$ sont :

$$[C_{2m}^*(\omega)] = i\omega \int_0^{+\infty} [C_{2m}(t)] e^{-i\omega t} dt = \frac{E_c^*(\omega) h_c}{1 - \nu_c^2} \begin{bmatrix} 1 & \nu_c & 0 \\ \nu_c & 1 & 0 \\ 0 & 0 & \frac{1 - \nu_c}{2} \end{bmatrix} \quad (\text{A.2})$$

$$[C_{2b}^*(\omega)] = i\omega \int_0^{+\infty} [C_{2b}(t)] e^{-i\omega t} dt = \frac{h_c^2}{12} [C_{2m}^*(\omega)]$$

$$[C_{2s}^*(\omega)] = i\omega \int_0^{+\infty} [C_{2s}(t)] e^{-i\omega t} dt = \frac{E_c^*(\omega) h_c}{2(1 + \nu_c)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

où $E_c^*(\omega)$, ν_c et h_c sont respectivement le module d'Young complexe, le coefficient de Poisson et l'épaisseur de la couche centrale. En écrivant le module d'Young complexe

sous la forme $E_c^*(\omega) = E_c^*(0) + E(\omega)$, les matrices de comportement (A.2) peuvent être écrites sous la forme :

$$\begin{cases} [C_{2b}^*(\omega)] = [C_{2b}^*(0)] + E(\omega) [C_{2b}] \\ [C_{2s}^*(\omega)] = [C_{2s}^*(0)] + E(\omega) [C_{2s}] \end{cases} \quad (\text{A.3})$$

où les nouvelles matrices $[C_{2b}]$ et $[C_{2s}]$ sont données par :

$$[C_{2b}] = \frac{h_c^3}{12(1 - \nu_c^2)} \begin{bmatrix} 1 & \nu_c & 0 \\ \nu_c & 1 & 0 \\ 0 & 0 & \frac{1 - \nu_c}{2} \end{bmatrix}, \quad [C_{2s}] = \frac{h_c}{2(1 + \nu_c)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{A.4})$$

Bibliographie

- [1] K. Ramkumar and N. Ganesan. Vibration and damping of composite sandwich box column with viscoelastic/electrorheological fluid core and performance comparison. *Materials & Design*, 30(8) :2981 – 2994, 2009.
- [2] Jong Hee Yim, Shee Yong Cho, Yun Jong Seo, and Bor Z. Jang. A study on material damping of 0° laminated composite sandwich cantilever beams with a viscoelastic layer. *Composite Structures*, 60(4) :367 – 374, 2003.
- [3] J. Vaswani, N.T. Asnani, and B.C. Nakra. Vibration and damping analysis of curved sandwich beams with a viscoelastic core. *Composite Structures*, 10(3) :231 – 245, 1988.
- [4] E.M. Daya, L. Azrar, and M. Potier-Ferry. An amplitude equation for the non-linear vibration of viscoelastically damped sandwich beams. *Journal of Sound and Vibration*, 271(3-5) :789 – 813, 2004.
- [5] J. S. Moita, J. Infante Barbosa, C. M. Mota Soares, and C. A. Mota Soares. Sensitivity analysis and optimal design of geometrically non-linear laminated plates and shells. *Computers & Structures*, 76(1-3) :407 – 420, 2000.
- [6] Russell Biagi and Hilary Bart-Smith. Imperfection sensitivity of pyramidal core sandwich structures. *International Journal of Solids and Structures*, 44(14-15) :4690 – 4706, 2007.

- [7] J.L.T. Santos K. K. Choi. Design sensitivity analysis of non-linear structural systems part i : Theory. *International Journal for Numerical Methods in Engineering*, 24 :2039–2055, 1987.
- [8] A.M.G. de Lima, A.W. Faria, and D.A. Rade. Sensitivity analysis of frequency response functions of composite sandwich plates containing viscoelastic layers. *Composite Structures*, 92(2) :364 – 376, 2010.
- [9] J. A. Kepler and P. H. Bull. Sensitivity of structurally loaded sandwich panels to localized ballistic penetration. *Composites Science and Technology*, 69(6) :696 – 703, 2009.
- [10] G. Sliva, A. Brezillon, J.M. Cadou, and L. Duigou. A study of the eigenvalue sensitivity by homotopy and perturbation methods. *Journal of Computational and Applied Mathematics*, 234(7) :2297 – 2302, 2010. Fourth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2008).
- [11] N. H. Kim K.K. Choi. *Structural Sensitivity Analysis and Optimization 2 Non-linear Systems and Applications*, volume 2. Springer-Verlag, 2004.
- [12] A. Walker J. Thompson. The nonlinear perturbation analysis of structural systems. *J. Solid Struct*, 4 :757–768, 1968.
- [13] L. Azrar, B. Cochelin, N. Damil, and M. Potier-Ferry. An asymptotic-numerical method to compute the postbuckling behavior of elastic plates and shells. *Int. J. Numer. Meth. Eng.*, 36 :1251–1277, 1993.
- [14] M Potier-Ferry B Cochelin, N Damil. *Méthode asymptotique numérique :une technique de résolution des équations non linéaires*. Hermes Science Publications, 2007.
- [15] Komlanvi Lampoh, Isabelle Charpentier, and El Mostafa Daya. A generic approach for the solution of nonlinear residual equations. part iii : Sensitivity com-

-
- putations. *Computer Methods in Applied Mechanics and Engineering*, 200(45-46) :2983 – 2990, 2011.
- [16] E.M. Daya and M. Potier-Ferry. A numerical method for nonlinear eigenvalue problems application to vibrations of viscoelastic structures. *Computers & Structures*, 79(5) :533 – 541, 2001.
- [17] Massamaesso Bilasse, Isabelle Charpentier, El Mostafa Daya, and Yao Koutsawa. A generic approach for the solution of nonlinear residual equations. part ii : Homotopy and complex nonlinear eigenvalue method. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52) :3999 – 4004, 2009.
- [18] F. Van Keulen, R.T. Haftka, and N.H. Kim. Review of options for structural design sensitivity analysis. part 1 : Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33) :3213 – 3243, 2005. Structural and Design Optimization.
- [19] Matthijs Langelaar and Fred van Keulen. Sensitivity analysis of shape memory alloy shells. *Computers & Structures*, 86(9) :964 – 976, 2008. Composites.
- [20] Alessandro Beghini, Zdenek P. Bazant, Anthony M. Waas, and Shiladitya Basu. Postcritical imperfection sensitivity of sandwich or homogenized orthotropic columns soft in shear and in transverse deformation. *International Journal of Solids and Structures*, 43(18-19) :5501 – 5524, 2006.
- [21] Joseph M. Pajot and Kurt Maute. Analytical sensitivity analysis of geometrically nonlinear structures based on the co-rotational finite element method. *Finite Elements in Analysis and Design*, 42(10) :900 – 913, 2006.
- [22] Imen Elkhaldi, Isabelle Charpentier, and El Mostafa Daya. A gradient method for viscoelastic behaviour identification of damped sandwich structures. *Comptes Rendus Mécanique*, (0) :–, 2012.

- [23] A. Griewank. *Evaluating Derivatives, principles and Techniques of Algorithmic Differentiation*, volume 19. Frontiers in Applied Mathematics, Philadelphia, 2000.
- [24] Isabelle Charpentier and Michel Potier-Ferry. Différentiation automatique de la méthode asymptotique numérique typée : l'approche diamant. *Comptes Rendus Mécanique*, 336(3) :336–340, 2008.
- [25] Yao Koutsawa, Isabelle Charpentier, El Mostafa Daya, and Mohammed Cherkaoui. A generic approach for the solution of nonlinear residual equations. Part I : the Diamant toolbox. *Computer Methods in Applied Mechanics and Engineering*, 198(3-4) :572–577, December 2008.
- [26] Imen Elkadi. *Application de la Différentiation Automatique pour l'identification, l'optimisation et l'étude de sensibilité dans quelques problèmes mécaniques*. PhD thesis, Université Paul Verlaine de Metz, 2011.
- [27] I. Charpentier. On higher-order differentiation in non-linear mechanics. *Computer Methods in Applied Mechanics and Engineering*, pages 1–12, 2011.
- [28] I. Charpentier. Sensitivity of solutions computed through the asymptotic numerical method. *Comptes Rendus Mecanique*, 336 :788–793, 2008.
- [29] J.M. Cadou, M. Potier-Ferry, and B. Cochelin. A numerical method for the computation of bifurcation points in fluid mechanics. *European Journal of Mechanics - B/Fluids*, 25(2) :234 – 254, 2006.
- [30] S. H. Chen. Eigensolution reanalysis of modified structures using perturbations and raleigh quotients. *Comm. Num Meth Eng*, 10 :111–119, 1994.
- [31] W. B. Bickford. An improved computational thechnic for perturbations of the generalized symmetric linear algebraic eigenvalue problem. *int. j. Numer. meth. Eng*, 24 :529–541, 1987.

-
- [32] L. B. Rall. *Automatic differentiation : Techniques and Applications*. Springer-Verlag, 1981.
 - [33] Laurent Hascoët Valérie Pascual. TAPENADE for C. In *Advances in Automatic Differentiation*. Springer, 2008.
 - [34] J. Utke A. Griewank, D. Jude. Adol-c, a package for the automatic differentiation of algrithms written in c/c++. *ACM T. Math Software*, pages 131–167, 1996.
 - [35] A.Vehreschild C.Bischof, B.Lang. Automatic differentiation for matlab programs. *Proceedings in Applied Mathemaics and Mechanix*, 2 :50–53, 2003.
 - [36] H. M. Buecker, G. Corliss, P. Hovland, U. Naumann, and B Norris. Automatic differentiation : Applications, theory, and tools. *Computational Science and Engineering*, 2005.
 - [37] Catherine MISCHLER. *Les concepts et les outils de la différentiation automatique appliqués aux simulateurs de procédés*. PhD thesis, Institut national polytechnique de Toulouse, Toulouse, FRANCE, 1996.
 - [38] Rainer Steiger, Christian H. Bischof, Bruno Lang, and Walter Thiel. Using automatic differentiation to compute derivatives for a quantum-chemical computer program. *Future Generation Computer Systems*, 21(8) :1324 – 1332, 2005.
 - [39] M. Potier-Ferry I. Charpentier, A. Lejeune. The diamant approach for an efficient automatic differentiation of the asymptotic numerical method. *Advance in Automatic Differentiation*, 64 :139–149, 2008.
 - [40] M. Potier-Ferry E. Doedel. A numerical analysis and control of bifurcation in finite problems (i) bifurcationin finite dimension. *J Bifurcat Chao*, 1 :493–520, 1991.

- [41] M. Jamal, B. Braikat, S. Boutmir, N. Damil, and M. Potier-Ferry. A high ordre implicit algorithm for solving nonlinear problems. *Comput Mech*, 28 :375–380, 2002.
- [42] K. Georg E. L. Allgower. Simplicial and continuation methods for approximations fixed point solutions to systems of equations. *SIAM Rev*, 22 :28–85, 1980.
- [43] H. B. Keller. Numerical solution of bifurcation and eignvalue problems, in applications of bifurcation theory. *P. Rabinowitz, ed. Academic Press, New York*, pages 359–384, 1977.
- [44] John W. and Leonard. Newton-raphson iterative method applied to circularly towed cable-body system. *Engineering Structures*, 1(2) :73 – 80, 1979.
- [45] Robert and Kao. A comparison of newton-raphson methods and incremental procedures for geometrically nonlinear analysis. *Computers & Structures*, 4(5) :1091 – 1097, 1974.
- [46] H. Zahrouni, M. Potier-Ferry, H Elasmr, and N. Damil. Asymptotic numerical method for nonlinear constitutive laws. *Revue européenne des éléments finis*, 7 :841–869, 1998.
- [47] Martin Berz, Kyoko Makino, Khodr Shamseddine, Georg H. Hoffstätter, and Weishi Wan. COSY INFINITY and its applications in nonlinear dynamics. In Martin Berz, Christian Bischof, George Corliss, and Andreas Griewank, editors, *Computational Differentiation : Techniques, Applications, and Tools*, pages 363–365. SIAM, Philadelphia, PA, 1996.
- [48] Y.F. Chang and George F. Corliss. Solving ordinary differential equations using taylor series. *ACM Trans. Math. Software*, 8 :114–144, 1982.
- [49] K. Makino J. Hoefkens, M. Berz. Efficient high-order methods for odes and daes

-
- in automatic differentiation 2000 : From simulation to optimization. *Springer*, pages 343–348, 2002.
- [50] J. D. Pryce N. S. Nedialkov. Solving differential-algebraic equations by taylor series (i) : Computing taylor coefficients,. *BIT*, 45 :561–591, 2005.
- [51] J. D. Pryce N. S. Nedialkov. Soliving differential-algebraic equations by taylor series (iii) : The daets code. *J. Numer Anal Indsdust Math*, 3 :61–80, 2008.
- [52] B. M. Averick, R. G. Carter, J. J. Moré, G. L. Xue, and The minpack-2 test problem collection. The minpack-2 test problem collection. *Preprint MSC-P153-0694 Argonne National Laboratory*, 1992.
- [53] R. Seydel. *Paratical Bifurcation and Stability Analysis, Number 5*. Interdisciplinary Applied Mathematics, 2009.
- [54] L. Azrar, E.H. BOUTYOUR, and M. POTIER-FERRY. Non-linear forced vibrations of plates by an asymptotic numerical method. *Journal of Sound and Vibration*, 252(4) :657 – 674, 2002.
- [55] H. Boudaoud, A. Lejeune, M. Potier-Ferry, H. Zahrouni, and I. Charpentier. Différentiation automatique et man : Application au contact unilatéral. *Acte du 9ième colloque national en calcul de structure*, 2 :281–286, 2009.
- [56] Ph Guillaume. Nonlinear eigenproblems. *SIAM J. Matrix Anal A.*, 20 :575–595, 199.
- [57] J. D. Pryce. A simple structural analysis method for deas. *BIT*, 4 :364–384, 2001.
- [58] M. Potier-Ferry F. Béchet, A. Lejeune. Taylor series to solve friction problems. *C R Mecanique*, 338 :327–332, 2010.
- [59] E.H. Boutyour, L. Azrar, and M. Potier-Ferry. Vibration of buckled elastic structures with large rotations by an asymptotic numerical method. *Computers & Structures*, 84(3-4) :93 – 101, 2006.

- [60] M. Bilasse, L. Azrar, and E.M. Daya. Complex modes based numerical analysis of viscoelastic sandwich plates vibrations. *Computers & Structures*, 89(7-8) :539 – 555, 2011.
- [61] E.H.E. H. Boutyour, H. Zahrouni, M. Potier-Ferry, and M. Boudi. Asymptotic-numerical method for buckling analysis of shell structures with large rotations. *Journal of Computational and Applied Mathematics*, 168(1-2) :77 – 85, 2004. Selected Papers from the Second International Conference on Advanced Computational Methods in Engineering (ACOMEN 2002).
- [62] J.M. Cadou, M. Potier-Ferry, and B. Cochelin. A numerical method for the computation of bifurcation points in fluid mechanics. *European Journal of Mechanics - B/Fluids*, 25(2) :234 – 254, 2006.
- [63] Wassila Aggoune, Hamid Zahrouni, and Michel Potier-Ferry. High-order prediction-correction algorithms for unilateral contact problems. *Journal of Computational and Applied Mathematics*, 168(1-2) :1 – 9, 2004. Selected Papers from the Second International Conference on Advanced Computational Methods in Engineering (ACOMEN 2002).
- [64] J. Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal. *Numerical Optimization - Theoretical and Practical Aspects*. Universitext, 2006.
- [65] A.D. and Fitt. Symbolic computation of hyperbolicity regions for systems of two-phase flow conservation laws using maple. *Journal of Symbolic Computation*, 8(3) :305 – 308, 1989.
- [66] R and Taylor. Thermodynamics with maple : I- symbolic computation. *Mathematics and Computers in Simulation*, 45(1-2) :101 – 119, 1998.
- [67] D. Villard and B. Arnaldi. Symbolic differentiation library for simulation of

-
- multibody rigid systems. *Mathematics and Computers in Simulation*, 42(4-6) :659 – 673, 1996.
- [68] Steffen Seeger and Karl Heinz Hoffmann. On symbolic derivation of the cumulant equations. *Computer Physics Communications*, 168(3) :165 – 176, 2005.
- [69] Bruno Arnaldi Laurent Cogné. Différentiation symbolique orientée objets. Master’s thesis, Institut National de Recherche en Informatique et en Automatique, 1995.
- [70] David W. Juedes. *A Taxonomy of Automatic Differentiation Tools*. In A. Griewank, G. F. Corliss Automatic differentiation of algorithm, theory, implementation and application, 1991.
- [71] Robert Edwin Wengert. A simple automatic derivative evaluation programs. In *Communication of ACM*, 1964.
- [72] Laurent Hascoët and Valérie Pascual. TAPENADE 2.1 user’s guide. Rapport technique 300, INRIA, Sophia Antipolis, 2004.
- [73] Christian H. Bischof, Alan Carle, George F. Corliss, Andreas Griewank, and Paul D. Hovland. ADIFOR : Generating derivative codes from Fortran programs. *Scientific Programming*, 1(1) :11–29, 1992.
- [74] L. B. Rall. Differentiation in pascal-sc : Type gradient. *ACM T. transactions on Mathematics Software*, 10 No 2 :161–184, 1984.
- [75] Ahmad Elhage-Hussein, Michel Potier-Ferry, and Nouredine Damil. A numerical continuation method based on padé approximants. *International Journal of Solids and Structures*, 37(46-47) :6981 – 7001, 2000.
- [76] Aimin Xu and Chengjing Wang. Some extensions of faà di bruno’s formula with divided differences. *Computers & Mathematics with Applications*, 59(6) :2047 – 2052, 2010.

- [77] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis : The primer*. Wiley, 2008.
- [78] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling & Software*, 22(10) :1509 – 1518, 2007.
- [79] Francesca Campolongo, Andrea Saltelli, and Jessica Cariboni. From screening to quantitative sensitivity analysis. a unified approach. *Computer Physics Communications*, 182(4) :978 – 988, 2011.
- [80] Stig Morten and Thorsen. Screening groups of parameters using the morris method to select parameters for variance based sensitivity analysis. *Procedia - Social and Behavioral Sciences*, 2(6) :7755 – 7756, 2010.
- [81] M.P.R. Haaker and P.J.T. Verheijen. Local and global sensitivity analysis for a reactor design with parameter uncertainty. *Chemical Engineering Research and Design*, 82(5) :591 – 598, 2004.
- [82] Yutaka Tanaka, Fanghong Zhang, and Yuichi Mori. Local influence in principal component analysis : relationship between the local influence and influence function approaches revisited. *Computational Statistics & Data Analysis*, 44 :143–160, 2003.
- [83] J. Arogo Ogejo, R.S. Senger, and R.H. Zhang. Global sensitivity analysis of a process-based model for ammonia emissions from manure storage and treatment structures. *Atmospheric Environment*, 44(30) :3621 – 3629, 2010.
- [84] S. Oladyshkin, F.P.J. de Barros, and W. Nowak. Global sensitivity analysis : A flexible and efficient framework with an example from stochastic hydrogeology. *Advances in Water Resources*, 37(0) :10 – 22, 2012.

-
- [85] Jacques E.V. Peter and Richard P. Dwight. Numerical sensitivity analysis for aerodynamic optimization : A survey of approaches. *Computers & Fluids*, 39(3) :373 – 391, 2010.
- [86] V. Magnenet, A. Giraud, and F. Homand. Parameter sensitivity analysis for a drüker-prager model following from numerical simulations of indentation tests. *Computational Materials Science*, 44(2) :385 – 391, 2008.
- [87] Yao Koutsawa. *Modélisation et conception multiéchelles des matériaux : de la description atomique discrète aux modèles du continu. Application aux propriétés amortissantes des pare-brises*. PhD thesis, Université Paul Verlaine de metz, 2008.
- [88] Massamaesso BILASSE. Modélisation numérique des vibrations linéaires et non linéaires des structures sandwichsà âme viscoélastique. Master’s thesis, Université Paul Verlaine de Metz., 2010.
- [89] Bruno Cochelin and Christophe Vergez. A high order purely frequency-based harmonic balance formulation for continuation of periodic solutions. *Journal of Sound and Vibration*, 324(1-2) :243 – 262, 2009.
- [90] Sanjib Goswami and Wilfried Becker. The effect of facesheet/core delamination in sandwich structures under transverse loading. *Composite Structures*, 54(4) :515 – 521, 2001.
- [91] L. Lagunegrand, Th. Lorriot, R. Harry, and H. Wagnier. Design of an improved four point bending test on a sandwich beam for free edge delamination studies. *Composites Part B : Engineering*, 37(2-3) :127 – 136, 2006.
- [92] D. and Shu. Vibration of sandwich beams with double delaminations. *Composites Science and Technology*, 54(1) :101 – 109, 1995.
- [93] Thomas S. Gates, Xiaofeng Su, Frank Abdi, Gregory M. Odegard, and Helen M.

- Herring. Facesheet delamination of composite sandwich materials at cryogenic temperatures. *Composites Science and Technology*, 66(14) :2423 – 2435, 2006.
- [94] Buket Okutan Baba and Srinivasa Thoppul. Experimental evaluation of the vibration behavior of flat and curved sandwich composite beams with face/core debond. *Composite Structures*, 91(1) :110 – 119, 2009.
- [95] A. E. H. Love. *The mathematical theory of elasticity, 4th edition*. Cambridge University press, 1927.
- [96] R. D. Mindlin. Influence of transversase shear deformation on the bending of elastic plates. *Journal of Applied Mechanics*, 12 :69–76, 1945.
- [97] R. D. Mindlin. Influence of ratatory inertia effect and shear deformation in flexural motions of isotopic elastic plates . *Journal of Applied Mechanics*, 18 :1936–1951, 1951.
- [98] J. N. Reddy. A simple heigher-order theory of laminated composite plate. *Journal of Applied Mechanics*, 51 :745–752, 1984.
- [99] M. Touratier. An efficient standard plate theory. *International Journal of Engi-neerring Science*, 51 :745–752, 1984.
- [100] P. Ribeiro and M. Petyt. Non-linear vibration of composite laminated plates by the hierarchical finite element method. *Composite Structures*, 46(3) :197 – 208, 1999.
- [101] Ronald S. Harichandran and Mohamad K. Naja. Random vibration of laminated composite plates with material non-linearity. *International Journal of Non-Linear Mechanics*, 32(4) :707 – 720, 1997.
- [102] M. Z. AsIk and Selim Tezcan. Laminated glass beams : Strength factor and temperature effect. *Computers & Structures*, 84(5-6) :364 – 373, 2006.

-
- [103] P. S. Timoshenko. *History of strength of Materials*. Dover Publication, 1953.
- [104] D. K. Rao. Frequency and loss factor of sandwich beams under various boundary conditions. *Mechanical Engineering Science*, 20 :271–282, 1978.
- [105] Heng Hu, Salim Belouettar, Michel Potier-Ferry, and El Mostafa Daya. Review and assessment of various theories for modeling sandwich composites. *Composite Structures*, 84(3) :282 – 292, 2008.
- [106] Yao Koutsawa and El Mostafa Daya. Static and free vibration analysis of laminated glass beam on viscoelastic supports. *International Journal of Solids and Structures*, 44(25-26) :8735 – 8750, 2007.
- [107] Emmanuel Francois Gouri Dhatt, Gilbert Touzot. *Méthode des éléments finis*. Lavoisier, 2007.
- [108] M. Z. AsIk and Selim Tezcan. A mathematical model for the behavior of laminated glass beams. *Computers & Structures*, 83(21-22) :1742 – 1753, 2005.
- [109] John S. Tomblin, Ever J. Barbero, and Luis A. Godoy. Imperfection sensitivity of fiber microbuckling in elastic-nonlinear polymer-matrix composites. *International Journal of Solids and Structures*, 34(13) :1667 – 1679, 1997.
- [110] Laëtitia Duigou, El Mostafa Daya, and Michel Potier-Ferry. Iterative algorithms for non-linear eigenvalue problems. application to vibrations of viscoelastic shells. *Computer Methods in Applied Mechanics and Engineering*, (11-12) :1323 – 1335, 2003.
- [111] J.R. Banerjee, C.W. Cheung, R. Morishima, M. Perera, and J. Njuguna. Free vibration of a three-layered sandwich beam using the dynamic stiffness method and experiment. *International Journal of Solids and Structures*, 44(22-23) :7543–7563, 2007.

- [112] Y. Frostig and O.T. Thomsen. On the free vibration of sandwich panels with a transversely flexible and temperature-dependent core material part i : Mathematical formulation. *Composites Science and Technology*, 69(6) :856 – 862, 2009.
- [113] M. Potier-Ferry E. M. Daya L. Azrar. Modélisation par éléments finis des vibrations non-linéaires des plaques sandwich viscoélastique. *Mécanique ind*, 6 :13–20, 2005.
- [114] R.M. Lin and M.K. Lim. Modal analysis of close modes using perturbative sensitivity approach. *Engineering Structures*, 19(6) :397 – 406, 1997.
- [115] S.S.A. Ravi, T.K. Kundra, and B.C. Nakra. Single step eigen perturbation method for structural dynamic modification. *Mechanics Research Communications*, 22(4) :363 – 369, 1995.
- [116] D.T. Nguyen, C.F. Bunting, K.J. Moeller, H. Runesha, and J. Qin. Subspace and lanczos sparse eigen-solvers for finite element structural and electromagnetic applications. *Advances in Engineering Software*, 31(8-9) :599 – 606, 2000.
- [117] S.A. Arjmandi and V. Lotfi. Computing mode shapes of fluid-structure systems using subspace iteration methods. *Scientia Iranica*, 18(6) :1159 – 1169, 2011.
- [118] S. H. Chen. Matrix perturbation theory in structural dynamics. *Beijing : International Academic Publisher*, 1993.
- [119] M. Potier-Ferry B. Cochelin, N. Damil. Asymptotic numerical methods and padé approximants for non-linear elastic structures. *International Journal for Numerical Methods in Engineering*, 37 :1187–1213, 1994.
- [120] S. H. Liu U. Kirsch. Structural reanalysis for general layout modifications. *AIAA JOURNAL*, 35 :382–388, 1997.
- [121] S. X. Piao B. S. Wu. Application of padé approximation to mécanique. *Mechanics and Practice*, 18 :27–29, 1996.

-
- [122] XIAOWEI YANG, SUHUAN CHEN, and BAISHENG WU. Eigenvalue reanalysis of structures using perturbations and pade approximation. *Mechanical Systems and Signal Processing*, 15(2) :257 – 263, 2001.
 - [123] Uri Kirsch. Efficient sensitivity analysis for structural optimization. *Computer Methods in Applied Mechanics and Engineering*, 117(1-2) :143 – 156, 1994.
 - [124] Liang Ma, Su Huan Chen, and Guang Wei Meng. Combined approximation for reanalysis of complex eigenvalues. *Computers & Structures*, 87(7-8) :502 – 506, 2009.
 - [125] J. A. Yorke S. N Chow, J.Malletparret. Finding zeros of maps - homotopy methods that are constructive with probability one. *Math Comput*, 32 :887–899, 1978.
 - [126] H. B. Keller. Global homotopies and newton method in recent in numerical analysis, g. h. globub and c. de boor. *Academic Press, New York*, pages 73–94, 1978.
 - [127] W. Zangwill C. Gracia. *Pathways to solutions fixed points and Equilibria*. Prentice-Hall Cliff, 1981.
 - [128] L. T. Watson. Globally convergent homotopi method- a tutorial. *Appl Math Comput*, 31 :369–396, 1989.
 - [129] R. B. Nelson. Simplified calculations of eigenvector derivatives. *AIAA JOURNAL*, 14 :1201–1205, 1976.
 - [130] Y. Chu C. S. Rudisill. Numerical methods for evaluating the derivatives of eigenvalues and eigenvectors. *AIAA JOURNAL*, 13 :834–837, 1975.
 - [131] In-Won Lee and Gil-Ho Jung. An efficient algebraic method for the computation of natural frequency and mode shape sensitivities part ii. multiple natural frequencies. *Computers & Structures*, 62(3) :437 – 443, 1997.

- [132] In-Won Lee and Gil-Ho Jung. An efficient algebraic method for the computation of natural frequency and mode shape sensitivities part i. distinct natural frequencies. *Computers & Structures*, 62(3) :429 – 435, 1997.
- [133] Robert A. Moreau. Identification de propriétés viscoélastiques de matériaux polymères par mesures de réponse en fréquences de structures. Master’s thesis, INSA Rouen, France, 2007.
- [134] F. Abdoun, L. Azrar, E.M. Daya, and M. Potier-Ferry. Forced harmonic response of viscoelastic structures by an asymptotic numerical method. *Computers & Structures*, 87(1-2) :91 – 100, 2009.
- [135] F. Aryana and H. Bahai. Sensitivity analysis and modification of structural dynamic characteristics using second order approximation. *Engineering Structures*, 25(10) :1279 – 1287, 2003.
- [136] H. D. Lian S. H. Chen, X. W. Yang. Comparison of several eigenvalue reanalysis methods for modified structures. *StructMultidisc Optim*, 20 :253–259, 2000.
- [137] M. Bilasse, L. Azrar, and E.M. Daya. Complex modes based numerical analysis of viscoelastic sandwich plates vibrations. *Computers & Structures*, 89(7-8) :539 – 555, 2011.
- [138] M. Bilasse, E.M. Daya, and L. Azrar. Linear and nonlinear vibrations analysis of viscoelastic sandwich beams. *Journal of Sound and Vibration*, 329(23) :4950 – 4969, 2010.
- [139] A. Morgan L. Watson, S. Billups. Hompack, a suite of codes for globally convergent homotopy algorithms. *ACM T. Math Software*, 13 :281–310, 1987.
- [140] R. Kalaba and L. Tesfatsion. Solving nonlinear equations by adaptive homotopy continuation. *Applied Mathematics and Computation*, 41(2) :99 – 115, 1991.

-
- [141] E Carrera. On the use of the murakami's zig-zag function in the modeling of layered plates and shells. *Computers and Structures*, 82(7-8) :541 – 554, 2004.
- [142] S. Brischetto, E. Carrera, and L. Demasi. Improved bending analysis of sandwich plates using a zig-zag function. *Composite Structures*, 89(3) :408 – 415, 2009.
- [143] R. Ohayon. M. A. Trindade, A. Benjeddou. Modeling of frequency dependent viscoelastic materials for active-passive vibration damping. *Journal of Vibration and Acoustics*, 122 (2) :169–174, 2000.
- [144] A. M. Trindade. Reduce-order finite element models of viscoelastically damped beams though internal variables projection. *Journal of Vibration and Acoustics*, 128 :501–508, 2006.

Résumé

En ingénierie, pour mieux comprendre le comportement mécanique d'une structure soumise à une certaine perturbation des paramètres de conception, on procède souvent à une analyse de sensibilité. Celle-ci fournit des informations quantitatives et qualitatives sur le comportement du modèle étudié et offre un accès aux gradients utilisables dans des méthodes d'identification et d'optimisation. Dans cette thèse, nous démontrons que ces informations peuvent être obtenues à un coût de développement faible en appliquant un outil de Différentiation Automatique (DA) au code informatique qui implémente le modèle. Nous adaptons la technique de DA à la Méthode Asymptotique Numérique, dans sa version Diamant, pour le calcul de la sensibilité des solutions numériques de problèmes non-linéaires discrétisés par la méthode des éléments finis. Nous discutons de manière générique à la fois les aspects théoriques et l'implémentation de plusieurs algorithmes écrits en Matlab. Les applications concernent des poutres et des plaques sandwich dans les cas statique et dynamique (vibration libre). Les sensibilités sont calculées par rapport aux paramètres géométriques, mécanique et par rapport à des matrices de rigidité élémentaires. La généralité de nos développements permet de prendre en compte plusieurs lois viscoélastiques sans effort supplémentaire. Trois types de modèles viscoélastiques sont étudiés : module complexe constant, faible amortissement et fort amortissement. Comparée à l'approximation par différences finis souvent utilisée en mécanique, notre approche fournit des résultats plus précis pour la sensibilité de la réponse d'une structure lorsque les paramètres de conception sont perturbés. Elle permet aussi de réduire le temps de calcul.

Mots-clés: calcul de sensibilité, différentiation automatique, sandwich viscoélastique, vibration libre non linéaire, amortissement, élément fini, fréquence propre, mode réel, mode complexe, méthode asymptotique numérique, facteur de perte, Diamant, ADiMat, série de Taylor

Abstract

in engineering, for a better understanding of the mechanical behavior of a structure submitted to some perturbation of the modelling parameters, one often proceed to a sensitivity analysis. This provides quantitative and qualitative information on the behavior of the model under study and gives access to gradients that may be used in identification and optimization methods. In this thesis, we demonstrate that this information may be obtained at a low development effort by applying an Automatic Differentiation (AD) tool to the computer code that implements the model. We adapt the AD techniques to the Asymptotic Numerical Method (ANM), in its Diamant version, for sensitivity computations of numerical solutions of nonlinear problems discretized through a finite element method. We discuss in a generic manner both the theoretical aspects and the implementation of several algorithms written in Matlab. Applications are concerned with sandwich beams and sandwich plates in both the static and dynamic (free vibration) cases. Sensitivities are computed with respect to geometric and mechanical parameters, and with respect to elementary stiffness matrix. The generality of our developments allows to take into account several viscoelastic laws with no additional effort. Three kind of viscoelastic models are studied : constant complex modulus, low damping, higher damping. In comparison with the finite difference approximation often used in mechanics, our approach provides more accurate results for the sensitivity of the structure response to a perturbation of the modelling parameters. It also allows a reduction of the computation effort.

Keywords: sensitivity computation, automatic differentiation, viscoelastic sandwich, non-linear free vibration, damping, finite element, eigenfrequency, real eigenmode, complex eigenmode, asymptotic numerical method,lost facteur, Diamant, ADiMat, Taylor expansion.