

Titre: Méthode des équations de sensibilités continues d'ordre élevé pour les paramètres de valeur
Title:

Auteur: Corinne Belley
Author:

Date: 2015

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Belley, C. (2015). Méthode des équations de sensibilités continues d'ordre élevé pour les paramètres de valeur [Master's thesis, École Polytechnique de Montréal].
Citation: PolyPublie. <https://publications.polymtl.ca/1749/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1749/>
PolyPublie URL:

Directeurs de recherche: Dominique Pelletier
Advisors:

Programme: Génie mécanique
Program:

UNIVERSITÉ DE MONTRÉAL

MÉTHODE DES ÉQUATIONS DE SENSIBILITÉS CONTINUES D'ORDRE ÉLEVÉ
POUR LES PARAMÈTRES DE VALEUR

CORINNE BELLEY
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
JUN 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MÉTHODE DES ÉQUATIONS DE SENSIBILITÉS CONTINUES D'ORDRE ÉLEVÉ
POUR LES PARAMÈTRES DE VALEUR

présenté par : BELLEY Corinne

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GARON André, Ph. D., président

M. PELLETIER Dominique, Ph. D., membre et directeur de recherche

M. OZELL Benoît, Ph. D., membre

DÉDICACE

À Michael, pour son dévouement, sa patience et sa confiance.

REMERCIEMENTS

Tout d'abord, je tiens à remercier mon directeur de recherche Dominique Pelletier pour la qualité de son encadrement, la pertinence de ses conseils ainsi que pour la confiance qu'il m'a accordée dans la réalisation de ce projet. Je souhaite également adresser mes remerciements à Alexander Hay pour ses judicieux conseils, ses commentaires constructifs et pour son importante implication dans ce projet. Ensemble, ils m'ont apporté le soutien technique nécessaire pour me faire progresser et mener ce projet à terme.

Enfin, je tiens à souligner l'appui financier du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) et du Fonds de recherche du Québec – Nature et technologies (FQRNT) tout au long de ma maîtrise. Les bourses octroyées par ces deux organismes m'ont permis de réaliser mes ambitions personnelles et professionnelles.

RÉSUMÉ

Ce mémoire présente une méthode générique permettant de générer automatiquement les équations de sensibilités continues d'ordre arbitrairement élevé n pour un nombre quelconque de paramètres q . Une sensibilité est le taux de variation des variables d'écoulement par rapport à un paramètre. Par exemple, la sensibilité de la vitesse u par rapport à un paramètre quelconque α est exprimée par : $\frac{\Delta u}{\Delta \alpha}$. Les sensibilités d'ordre n sont donc les dérivées n -ièmes des variables d'écoulement par rapport à un ou plusieurs paramètres. Dans le cadre de ce mémoire, nous abordons uniquement les paramètres de valeur.

L'optimisation de la performance de systèmes mécaniques ou aéronautiques nécessite généralement de nombreuses évaluations d'une fonction objectif F mesurant la performance du système à optimiser. Ces évaluations ont pour but d'identifier les valeurs des paramètres qui conduisent à une performance optimale. Habituellement, cela requiert la solution numérique des équations différentielles régissant le système (*e.g.* : Navier-Stokes). Plus le système est complexe, plus cela exige des ressources de calcul importantes. Le coût de chacune des simulations est habituellement la résolution d'un système non-linéaire. Il s'agit d'une opération fort coûteuse. Ce mémoire explore donc la faisabilité d'utiliser une série de Taylor d'ordre élevé n dans l'espace des paramètres comme substitut à la résolution numérique d'équations aux dérivées partielles afin de réduire le coût du design optimal. L'utilisation de la série de Taylor évite d'avoir à refaire des simulations complètes du système qui sont généralement très coûteuses.

Une analyse simple indique que le nombre de termes dans l'équation des sensibilités croît rapidement avec l'ordre n des sensibilités. Pour être utile en pratique, cette approche doit automatiser le processus de différentiation ainsi que le processus de génération des systèmes algébriques résultant de la discrétisation par éléments finis. Nous avons élaboré un algorithme générique basé sur le multinôme de Newton pour déterminer la structure des équations différentielles des sensibilités. Nous avons aussi conçu un algorithme pour générer automatiquement le système d'équations algébriques. L'innovation de ce travail réside dans la formulation générale et automatique des équations différentielles des sensibilités d'ordre élevé n qui privilégie une approche continue plutôt que discrète. Cela signifie que les équations aux dérivées partielles sont d'abord différentiées à l'ordre souhaité. Puis, elles sont discrétisées afin d'être résolues numériquement par la méthode des éléments finis. Cette formulation générale déve-

loppée pour générer automatiquement les équations de sensibilités d'ordre n est basée sur le multinôme de Newton.

Notre méthode de sensibilités a été implémentée et vérifiée par la méthode des solutions manufacturées. Ensuite, sa fonction principale de calculer les sensibilités d'ordre n a été validée par la comparaison des sensibilités numériques calculées aux sensibilités obtenues par la méthode des différences finies. Ce qui confirme que notre méthode calcule correctement les sensibilités. Notre méthode des sensibilités d'ordre n remplit donc son rôle adéquatement. Enfin, la possibilité d'approximer les solutions par la série de Taylor dans l'espace des paramètres a été étudiée. Les résultats obtenus révèlent que certaines conditions doivent être réunies pour obtenir une approximation précise sur une large plage de valeurs des paramètres avec la série de Taylor. Par exemple, l'intervalle de convergence de la série doit être suffisamment grand. La solution ne doit pas varier trop rapidement avec les paramètres, sinon l'approximation par la série de Taylor produit des résultats erronés. Dans certaines conditions, la série de Taylor peut ne pas converger. Cette technique d'approximation n'est donc pas sans faille lorsqu'on extrapole une solution par la série de Taylor.

L'évaluation rapide de solutions perturbées par la série de Taylor s'avère très efficace lorsqu'elle peut être utilisée sur une plage de valeurs raisonnablement grande. En effet, les coûts de calculs qui s'y rattachent sont suffisamment négligeables comparativement aux coûts engendrés par les simulations complètes d'un système non-linéaire.

ABSTRACT

This work introduces a generic method for the automatic development and resolution of continuous sensitivity equations of arbitrary high order n for any number q of parameters. A sensitivity is defined as the dependence of fluid variables (e.g.: velocity field, pressure) with respect to parameters such as the geometric components or physical properties. In other words, a sensitivity of order n is the n^{th} derivative of the flow variables with respect to parameter(s). For instance, the n^{th} sensitivity of the velocity u with respect to the parameter α is expressed by $\frac{\partial^n u}{\partial \alpha^n}$. For this work, we study only value parameters (e.g.: physical properties) which do not influence on the geometry.

The sensitivity is useful in the context of design optimization. Performance optimization of mechanical and aeronautical systems typically requires many evaluations of an objective function which measures the performance of the system to be optimized. These evaluations are achieved in order to identify the parameter values that lead to an optimal performance. In general, these evaluations require the numerical solution of the partial differential equations governing the system (e.g.: Navier-Stokes). These systems are usually nonlinear. Consequently, the cost of each simulation is significant. Also, the cost related to computing resources increases with the complexity of the system. This work studies therefore the possibility of using the Taylor series of high order n in parameter space to approximate the numerical solution of partial differential equations. This represents a cost effective alternative to solving partial differential equations.

A simple analysis shows that the number of terms in the n^{th} order sensitivity equation grows quickly with n . To be of practical value, this approach must provide an automatic and generic method for the determination of the structure of the differential equations for the sensitivities. More importantly, we also need to develop a generic algorithm to generate the finite element systems for the sensitivity equations. The contribution of this project is the general formulation and automatic development of the sensitivity equations of high order n which favors an approach continuous rather than discrete. This means that the partial differential equations are first differentiated. They are then discretized in order to be solved numerically using the finite element method. This general expression built to automatically produce sensitivity equations is based on the multinomial Newton.

Our continuous sensitivity method has been implemented and verified by the method of manufactured solutions. Then, the main role of this method, which is to compute sensitivities of order n , has been validated through comparison of its numerical sensitivities with those obtained by the finite difference method. This has confirmed that our method correctly calculates the sensitivities of order n . Finally, the possibility to approximate solutions by the Taylor series in the parameter space was studied. The results obtained show that the approximation by Taylor series on wide range of parameter values is accurate only when certain conditions are met. For instance, one of this condition is that the series convergence interval should not be small. Another condition is that the solution should not vary too quickly with the parameters. Otherwise, erroneous results may be obtained by the Taylor series when this condition is not met. The technique of extrapolation by the Taylor series is not robust as it sometimes fails to converge.

The fast evaluation of solutions for problem at perturbed values of the parameters by Taylor series expansion is very effective when it can be used over a quite large range of values. Indeed, the cost computations attached to it are sufficiently negligible compared to costs of the simulations of a nonlinear system to be considered as a potential alternative.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xv
LISTE DES SIGLES ET ABRÉVIATIONS	xvii
LISTE DES NOTATIONS ET SYMBOLES	xviii
LISTE DES ANNEXES	xxi
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 REVUE DE LITTÉRATURE	6
2.1 Définitions et concepts de base	6
2.1.1 Variable dépendante	6
2.1.2 Paramètre	7
2.1.3 Sensibilité	7
2.2 Applications et utilités des sensibilités	9
2.2.1 Design optimal	9
2.2.2 Évaluation rapide de solutions perturbées	10
2.2.3 Analyse de sensibilités	12
2.3 Techniques d'évaluation des sensibilités	13
2.3.1 Différences Finies	13
2.3.2 Méthode d'Équations des Sensibilités	14
2.3.3 Différentiation automatique	15
2.4 Comparaison des méthodes	16

2.5	Objectifs de recherche	19
CHAPITRE 3 MÉTHODE DE L'ÉQUATION DES SENSIBILITÉS D'ORDRE ARBITRAIRE POUR UN PROBLÈME 1D		
		21
3.1	Problème 1D	21
3.2	Formulation générale des équations de sensibilités à l'ordre n	22
3.2.1	Le multinôme de Newton	26
3.2.2	Analogie entre le théorème multinomial et la différentiation d'ordre n	30
3.2.3	Formulation générale de l'équation de sensibilités à un paramètre	33
3.2.4	Formulation générale de l'équation de sensibilités avec q paramètre(s)	38
3.3	Application de la formulation générale de l'équation de sensibilités aux équations de Navier-Stokes	48
CHAPITRE 4 ALGORITHME POUR LA GÉNÉRATION ET LA RÉOLUTION DES SENSIBILITÉS D'ORDRE ARBITRAIRE		
		54
4.1	Problème 1D	54
4.2	Données d'entrée	55
4.3	Procédure générale implémentée	56
4.4	Algorithme pour déterminer les systèmes de sensibilités à évaluer	57
4.5	Algorithme pour générer les équations de sensibilités d'un système	64
4.5.1	Exemple de fonctionnement des algorithmes 3 et 4	68
4.5.2	Enregistrement des équations de sensibilités dans une structure de données générique	70
4.6	Solveur des équations de sensibilités d'ordre n	76
4.6.1	Formulation faible	77
4.6.2	Discrétisation	81
4.6.3	Résolution par la méthode de Newton	84
4.6.4	Jacobien numérique	85
4.6.5	Jacobien analytique	86
4.7	Généricité de l'algorithme des sensibilités	88
CHAPITRE 5 VÉRIFICATION ET VALIDATION		
		89
5.1	Problème test	89
5.2	Vérification du code	90
5.3	Comparaison avec la méthode des Différences Finies	94
5.4	Évaluation rapide par la série de Taylor de solutions perturbées	100

CHAPITRE 6 CONCLUSION	106
6.1 Limitations de la solution proposée	107
6.2 Améliorations futures	108
RÉFÉRENCES	111
ANNEXES	115

LISTE DES TABLEAUX

Tableau 2.1	Tableau résumant les diverses techniques de calcul des sensibilités. . .	19
Tableau 3.1	Nombre de termes résultant de la dérivée à l'ordre n d'un produit de 1 à 4 fonctions.	23
Tableau 3.2	Nombre de termes pour chacune des Équations aux Dérivées Partielles selon l'ordre n	24
Tableau 3.3	Analogie entre les composantes de $\frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2}$ et celles de $(b_1 + b_2 + b_3)^2$.	34
Tableau 4.1	Forme générale du tableau des solutions des r systèmes de sensibilités pour une variable particulière.	70
Tableau 4.2	Forme générale du tableau <i>CombSens</i>	71
Tableau 4.3	Forme générale du tableau <i>InfoSens</i>	72
Tableau 4.4	Combinaisons des systèmes de sensibilités d'ordre 0 à 3 avec 2 paramètres.	74
Tableau 4.5	Information sur les sensibilités d'ordre 0 à 3 avec 2 paramètres. . . .	74
Tableau 4.6	Numéro de combinaisons pour $d(j, :, k)$ des termes numéro 1 à 9. . . .	75
Tableau 4.7	Numéro de combinaisons pour $d(j, :, k)$ des termes numéro 10 à 18. . .	75
Tableau 5.1	Valeur de référence des sensibilités d'ordre $n = 0$ à 6.	101
Tableau F.1	Intégration numérique sur l'intervalle de référence (voir Fortin et Garon, 2014).	138
Tableau H.1	Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P1$	147
Tableau H.2	Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P1$	147
Tableau H.3	Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P1$	148
Tableau H.4	Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P1$	148
Tableau H.5	Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P2$	149
Tableau H.6	Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P2$	149
Tableau H.7	Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P2$	150
Tableau H.8	Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P2$	150
Tableau H.9	Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P3$	151
Tableau H.10	Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P3$	151
Tableau H.11	Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P3$	151
Tableau H.12	Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P3$	152
Tableau H.13	Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P4$	152

Tableau H.14	Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P4$	152
Tableau H.15	Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P4$	153
Tableau H.16	Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P4$	153
Tableau H.17	Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P5$	153
Tableau H.18	Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P5$	154
Tableau H.19	Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P5$	154
Tableau H.20	Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P5$	154
Tableau H.21	Erreurs entre les solutions éléments finis et celles calculées par Différences Finies (DF) avec des éléments $P1$ ($n = 1$).	155
Tableau H.22	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 2$).	155
Tableau H.23	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 3$).	156
Tableau H.24	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 4$).	156
Tableau H.25	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 5$).	157
Tableau H.26	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 6$).	157
Tableau H.27	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 1$).	158
Tableau H.28	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 2$).	158
Tableau H.29	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 3$).	159
Tableau H.30	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 4$).	159
Tableau H.31	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 5$).	160
Tableau H.32	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 6$).	160
Tableau H.33	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 1$).	161

Tableau H.34	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 2$).	161
Tableau H.35	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 3$).	162
Tableau H.36	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 4$).	162
Tableau H.37	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 5$).	163
Tableau H.38	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 6$).	163
Tableau H.39	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 1$).	164
Tableau H.40	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 2$).	164
Tableau H.41	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 3$).	165
Tableau H.42	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 4$).	165
Tableau H.43	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 5$).	166
Tableau H.44	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 6$).	166
Tableau H.45	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 1$).	167
Tableau H.46	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 2$).	167
Tableau H.47	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 3$).	168
Tableau H.48	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 4$).	168
Tableau H.49	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 5$).	169
Tableau H.50	Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 6$).	169

LISTE DES FIGURES

Figure 2.1	Problème de conduction de chaleur.	7
Figure 2.2	Courbe de la température T en fonction de x	8
Figure 2.3	Zone de recirculation - Extrapolation en $\alpha_0 = 0^\circ$ (Mahieu, 2003).	12
Figure 2.4	Approches des méthodes de calcul des sensibilités.	15
Figure 3.1	Arbre des possibilités de combinaisons entre les combinaisons obtenues pour les ensembles indépendants E_1 et E_2	44
Figure 4.1	Schéma de la méthode implémentée.	57
Figure 4.2	Discrétisation avec 3 éléments de type P_1 ($p = 1$).	81
Figure 4.3	Discrétisation avec 3 éléments de type P_2 ($p = 2$).	81
Figure 4.4	Discrétisation avec 3 éléments de type P_3 ($p = 3$).	81
Figure 4.5	Discrétisation avec 3 éléments de type P_4 ($p = 4$).	82
Figure 5.1	Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités d'ordre n où $n = 0$ à 1.	94
Figure 5.2	Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités d'ordre n où $n = 2$ à 3.	95
Figure 5.3	Sensibilités numériques versus sensibilités des DF (éléments P_1).	98
Figure 5.4	Sensibilités numériques versus sensibilités des DF (éléments P_4).	99
Figure 5.5	Comparaison des solutions numériques versus les solutions extrapolées par série de Taylor.	102
Figure 5.6	Erreur des solutions extrapolées par série de Taylor.	104
Figure 5.7	Solutions numériques perturbées.	105
Figure A.1	Itinéraires pour aller de A vers B, puis B vers C.	118
Figure I.1	Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 1$ à 3.	172
Figure I.2	Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 4$ à 6.	173
Figure I.3	Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 7$ à 9.	174
Figure I.4	Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 10$	175
Figure I.5	Comparaison de la méthode proposée avec la méthode des DF centrées (éléments de type P_2).	176

Figure I.6	Comparaison de la méthode proposée avec la méthode des DF centrées (éléments de type $P3$).	177
Figure I.7	Comparaison de la méthode proposée avec la méthode des DF centrées (éléments de type $P5$).	178
Figure I.8	Comparaison des solutions numériques versus les solutions extrapolées par série de Taylor.	179
Figure I.9	Comparaison des solutions numériques versus les solutions extrapolées par série de Taylor.	180

LISTE DES SIGLES ET ABRÉVIATIONS

CFD	Computational Fluid Dynamics
CSEM	Continuous Sensitivity Equation Method
DA	Différentiation Automatique
DF	Différences Finies
DSEM	Discontinuous Sensitivity Equation Method
EDP	Équation aux Dérivées Partielles
MEF	Méthode des Éléments Finis
MES	Méthode des Équations des Sensibilités
MESC	Méthode des Équations des Sensibilités Continues
MESCE	Méthode des Équations des Sensibilités Continues Eulériennes
MESLE	Méthode des Équations des Sensibilités Continues Lagrangiennes
MESD	Méthode des Équations des Sensibilités Discrètes
MFN	Mécanique des Fluides Numérique
MSM	Méthode des Solutions Manufacturées
MVA	Méthode des Variables Adjointes
MVF	Méthode des Volumes Finis
RANS	Reynolds-Averaged Navier-Stokes

LISTE DES NOTATIONS ET SYMBOLES

Caractères usuels

b_j	Fonction ou monôme numéro j
c	Vecteur contenant les coefficients de l'équation de sensibilités
c_p	Chaleur massique à pression constante
C	Constante
d	Tableau contenant les ordres des dérivées des monômes de chacun des s termes
D	Diffusité thermique égale à $\frac{\lambda}{\rho c_p}$
f_j	Fonction numéro j
\mathbf{f}	Champ de forces volumiques
F	Fonction objectif
\mathbf{G}	Équations d'état (contraintes)
J_R	Matrice jacobienne associée au résidu R
K	Élément du maillage ($K = [x_1^K, x_2^K]$)
k_i	Valeur de l'exposant ou de la dérivée du i -ième terme ou monôme
$k_{i,j}$	Ordre de la dérivée du j -ième terme ou monôme par rapport au i -ième paramètre
L	Fonctionnelle de Lagrange
m	Nombre de monômes ou de fonctions dans un terme spécifique
n	Ordre de sensibilité
n_x	Nombre de composantes du vecteur \mathbf{x}
n_α	Nombre de composantes du vecteur $\boldsymbol{\alpha}$
$nddl$	Nombre total de degrés de liberté
ndp	Nombre de noeuds total pour un élément fini
nel	Nombre d'éléments finis sur le domaine
nn	Nombre de noeuds total sur l'ensemble du domaine
os	Ordre courant des sensibilités étudiées
p	Degré des polynômes pour la discrétisation spatiale
P	Pression
q	Nombre de paramètres
R	Résidu
r	Nombre total de systèmes de sensibilités
Re	Nombre de Reynolds
s	Nombre total de termes ou de combinaisons possibles

$S_{FD}^{(n)}$	Sensibilités d'ordre n calculée par Différences Finies
$S_u^{(n)}$	Sensibilité d'ordre n de u
$S_T^{(n)}$	Sensibilité d'ordre n de T
$S_u^{(\alpha^{(n)})}$	Sensibilité d'ordre n de u par rapport au paramètre α
$S_T^{(\alpha^{(n)})}$	Sensibilité d'ordre n de T par rapport au paramètre α
T	Température
u	Vitesse selon l'axe des abscisses
\mathbf{u}	Champ de vitesse
\mathbf{U}	Vecteur des degrés de liberté (pour u et T et leurs sensibilités)
x	Abscisse
\mathbf{X}	Vecteur des variables d'état (variables dépendantes)

Notations mathématiques

∇	Gradient
\mathbb{N}	Ensemble des nombres naturels incluant 0 ; <i>i.e.</i> $\mathbb{N} = \{0, 1, 2, \dots\}$
\mathbb{R}	Ensemble des nombres réels
$\ \cdot\ _2$	Norme euclidienne (norme 2)
$\ \cdot\ _{1,\Omega}$	Norme sur l'espace $H^1(\Omega)$
$\frac{\partial}{\partial \cdot}$	Dérivée partielle
$\langle \cdot, \cdot \rangle$	Produit scalaire

Caractères grecs

α	Paramètre et angle d'attaque
$\boldsymbol{\alpha}$	Vecteur de paramètres
β	Coefficient de dilatation thermique
δ	Très petite perturbation
$\Delta\alpha$	Amplitude de variation de α
ϵ	Critère de convergence
Γ	Frontière du domaine Ω
Γ_D	Partie de la frontière $\delta\Omega$ où l'on impose les conditions de Dirichlet
Γ_N	Partie de la frontière $\delta\Omega$ où l'on impose les conditions de Neumann
λ	Conductivité thermique
μ	Viscosité dynamique

ν	Viscosité cinématique équivalente à $\frac{\mu}{\rho}$
Ω	Domaine de calcul
ϕ	Fonction d'interpolation
ρ	Masse volumique
ξ	Variable adjointe
$\boldsymbol{\xi}$	Vecteur de variables adjointes

Indices

<i>bc</i>	Annotation pour les conditions aux frontières (Boundary condition)
<i>exa</i>	Annotation pour la solution exacte
<i>extr</i>	Annotation pour la solution extrapolée
<i>h</i>	Annotation pour la solution éléments finis
<i>i</i>	Numéro de la variable
<i>j</i>	Numéro de la variable
<i>x</i>	Composante en <i>x</i> d'un vecteur
<i>y</i>	Composante en <i>y</i> d'un vecteur
<i>1x</i>	Dérivée d'ordre 1 par rapport à <i>x</i>

Exposants

$(\alpha_i^{(n)})$	Dérivée partielle d'ordre <i>n</i> par rapport au paramètre α_i S'il y a plusieurs α_i , il s'agit d'une dérivée partielle croisée
()	Ordre de la dérivée
<i>k</i>	Numéro de l'itération
<i>K</i>	Élément du maillage

LISTE DES ANNEXES

Annexe A	ANALYSE COMBINATOIRE	115
Annexe B	OPÉRATEUR DE SOMMATION	119
Annexe C	FORMULATION GÉNÉRALE DES ÉQUATIONS DE SENSIBILITÉS DE NAVIER-STOKES	121
Annexe D	FORMULE DE LEIBNIZ	129
Annexe E	SOUS-ROUTINE EXPLICITE ÉQUIVALENTE À LA SOUS-ROUTINE DE L'ALGORITHME 4	132
Annexe F	INTÉGRATION NUMÉRIQUE	137
Annexe G	ESPACES FONCTIONNELS	144
Annexe H	TABLEAUX DES RÉSULTATS	147
Annexe I	RÉSULTATS SUPPLÉMENTAIRES	170

CHAPITRE 1 INTRODUCTION

Avec l'avènement des ordinateurs modernes dans les années 1960, la Mécanique des Fluides Numérique (MFN), couramment désignée par le terme anglais Computational Fluid Dynamics (CFD), a connu un essor fulgurant (Ammara, 2005). Entre les années 1960 et 2000, le but des industriels était de simuler des écoulements dans divers contextes. Il s'agissait donc de résoudre numériquement les équations qui régissaient le comportement des écoulements. L'évolution rapide de la puissance des ordinateurs a graduellement permis à l'industrie de mettre en place des solveurs pour résoudre numériquement des problèmes d'écoulement de plus en plus complexes. Au tout début, nous nous contentions de résoudre des écoulements potentiels et des problèmes de couche limite. Puis, à partir des années 1970, la simulation des écoulements d'Euler est devenue accessible grâce aux progrès qu'ont connus les ordinateurs. Dans les années 1980, des ordinateurs plus puissants, pouvant désormais résoudre numériquement les équations de Navier-Stokes, ont fait leur apparition. Enfin, la simulation numérique des écoulements turbulents décrits par les équations de Navier-Stokes moyennées (Reynolds-Averaged Navier-Stokes (RANS)) a fait ses débuts dans les années 1990. Aux alentours des années 1990-2000, les progrès engendrés par l'évolution continue des outils de calculs numériques ont amené les chercheurs à s'interroger sur la précision des simulations numériques en vue de les améliorer. Des processus de vérification et de validation (Roache, 1998) ont alors été développés pour respectivement quantifier l'erreur découlant de la résolution numérique des Équation(s) aux Dérivées Partielles (EDP) (e.g. : analyse de convergence) et vérifier que nous résolvions les EDP appropriées (i.e. : vérifier l'exactitude du modèle utilisé). Dans le but d'obtenir des simulations numériques plus précises, des techniques de raffinement de maillage et d'adaptation de maillages ont été mises en place. Encore aujourd'hui, nous cherchons à augmenter la précision et la fiabilité de la simulation des écoulements, puisqu'elle joue un rôle fondamental dans la conception des systèmes mécaniques et aéronautiques. En effet, elle constitue un outil d'analyse essentiel qui permet de repérer les problèmes de conception, de valider et d'optimiser des designs industriels.

Traditionnellement, avant l'arrivée des ordinateurs, l'étude des systèmes était principalement effectuée par le biais d'approches analytique et expérimentale (voir Ammara, 2005). L'approche analytique était limitée aux écoulements et géométries simples, tandis que l'approche expérimentale, bien que précise et fiable, était très dispendieuse et pouvait s'avérer très difficile à réaliser sur le plan technique. Avec l'arrivée des premiers ordinateurs, une nouvelle approche, soit l'approche numérique, a vu le jour. En raison de sa rapidité et de sa flexi-

bilité, elle constitue désormais un outil indispensable pour l'amélioration et l'optimisation des performances des systèmes mécaniques et aéronautiques. Dans le cadre de l'optimisation, le problème consiste à trouver l'optimum d'une fonction objectif qui sert de critère pour déterminer la meilleure solution du problème d'optimisation. Il existe deux types d'optimisation : l'optimisation unidisciplinaire et l'optimisation multi-critères. Dans le premier cas, pour trouver la solution optimale, nous avons recours à des algorithmes de gradient tels que ceux utilisés par la Méthode des Variables Adjointes (MVA) et la Méthode d'Équations des Sensibilités (MES). La résolution de ce problème implique quelques dizaines, voire des centaines, d'évaluations de la fonction objectif, ce qui n'est pas négligeable si nous considérons que chaque évaluation équivaut à la résolution d'un système non-linéaire (équations de Navier-Stokes). Quant à l'optimisation multi-critères, elle fait généralement appel à des algorithmes génétiques pour évaluer l'optimum global. Cela occasionne des centaines, voire des milliers, d'évaluations des fonctions objectifs (problèmes non-linéaires). Pour ces deux types d'optimisation, plus le système à optimiser est complexe plus l'évaluation de la fonction objectif est onéreuse. En effet, plus l'écoulement devient complexe, plus sa simulation nécessite de ressources de calcul (e.g. : maillage plus fin). Le coût engendré par chacune des simulations est la résolution d'un système non-linéaire. Pour l'optimisation multi-critères, ces coûts sont encore plus importants, puisque celle-ci requiert la résolution d'un plus grand nombre d'évaluations et ce, pour plusieurs fonctions objectifs.

Nous proposons une alternative pour réduire le nombre d'évaluations de l'écoulement. Il s'agit d'évaluer l'écoulement en l'extrapolant à partir de l'écoulement de référence $\mathbf{u}(\mathbf{x}, \alpha_0)$ par le biais de la série de Taylor dans l'espace des paramètres :

$$T^n(\mathbf{u}) = \mathbf{u}(\mathbf{x}, \alpha_0) + \sum_{i=1}^n S_u^{(\alpha^{(i)})}(\mathbf{x}, \alpha_0) \frac{(\Delta\alpha)^i}{i!} + O(\Delta\alpha^{n+1}), \quad (1.1)$$

où $T^n(\mathbf{u})$ est la configuration de base perturbée, $\mathbf{u}(\mathbf{x}, \alpha_0)$ est la configuration de base, $\Delta\alpha$ représente la perturbation du paramètre α et $S_u^{(\alpha^{(i)})}(\mathbf{x}, \alpha_0)$ est la sensibilité d'ordre i de u par rapport au paramètre α . Évidemment, l'évaluation de $\mathbf{u}(\mathbf{x}, \alpha_0)$ nécessite la résolution d'un problème non-linéaire. Toutefois, l'évaluation des sensibilités $S_u^{(\alpha^{(i)})}(\mathbf{x}, \alpha_0)$, pour $i = 1$ à n implique la résolution d'un problème linéaire. Le coût d'évaluation d'une sensibilité est donc une simple fraction du coût de calcul pour la résolution de u . À titre d'exemple, supposons que le coût de calcul de $S_u^{(\alpha^{(i)})}(\mathbf{x}, \alpha_0)$ représente 10% du coût d'évaluation de $\mathbf{u}(\mathbf{x}, \alpha_0)$. Nous pouvons alors évaluer les sensibilités d'ordre 1 à 10 pour un coût équivalent à celui de la résolution d'un écoulement. Comparativement à la fonction objectif dont le coût de réso-

lution est égale au coût d'un écoulement, ces sensibilités nous permettent d'approximer les solutions d'écoulement (i.e. : u) par le biais de la série de Taylor à l'ordre 10 à un coût moindre.

Pour améliorer l'efficacité et réduire les coûts de calcul permettant d'analyser un système, nous nous sommes donc orientés vers le calcul des sensibilités d'écoulement. Mise à part l'optimisation, l'étude des sensibilités fournit des informations utiles pour de nombreuses applications. Il suffit de songer au design optimal, à l'analyse d'incertitude, à l'évaluation rapide de solutions pour une gamme spécifique de valeurs dans l'espace des paramètres et à l'analyse des sensibilités (Sobieszczanski-Sobieski, 1990; Turgeon, 2001; Putko et al., 2002; Mahieu, 2003). Cette dernière consiste en l'identification des paramètres clés qui contrôlent la réponse des variables du système et définissent les zones sensibles.

Il est important de ne pas confondre les sensibilités d'écoulement, qui sont traitées dans ce travail, avec les sensibilités d'un design optimal. En CFD, la sensibilité est la dérivée de la solution de l'écoulement par rapport aux paramètres d'intérêt. Nous parlons alors de sensibilités d'écoulement. Pour l'optimisation, la sensibilité désigne le gradient de la fonction objectif que l'on peut calculer par la MVA ou la MES. Les variables adjointes de la MVA servent à évaluer le gradient de la fonction objectif. Par conséquent, elles ont du sens uniquement dans le cadre d'un problème d'optimisation et ne sont d'aucune utilité pour la caractérisation d'un écoulement. Dans le cas d'un problème d'optimisation, la MVA résout un seul système par fonction objectif peu importe le nombre q de paramètres de design. Pour un design multi-objectif (plusieurs fonctions objectifs), il y aura donc autant de systèmes adjoints que de fonctions objectifs. Quant à la MES, elle résout q systèmes d'équations de sensibilités peu importe le nombre de fonctions objectifs. Cela signifie que pour un problème d'optimisation multi-objectifs, le nombre de systèmes d'équations de sensibilités est indépendant du nombre de fonctions objectifs. La MVA est alors plus avantageuse lorsque le nombre q de paramètres est supérieur au nombre de fonctions objectifs. Par contre, il est préférable de recourir à la MES dans le cas où le nombre de fonctions objectifs est très grand par rapport à q .

De nombreuses techniques ont été développées pour calculer les sensibilités d'écoulement pour les problèmes de mécanique des fluides. La majorité d'entre elles portent sur le calcul des sensibilités d'ordre 1 (Gao et al., 1995; Turgeon, 2001; Putko et al., 2002; Bakr et al., 2009; Asli et Tousei, 2011). Nous recensons seulement quelques méthodes pour le calcul des sensibilités d'ordre 2 (Sherman et al., 1996; Mahieu, 2003; Ilinca et al., 2008), alors que très peu s'attaquent aux sensibilités d'ordre plus élevé (Sobieszczanski-Sobieski, 1990; Papadimi-

triu et Giannakoglou, 2013; Soulat et al., 2013). Il semble n’y avoir encore aucun travail qui traite les sensibilités continues à un ordre élevé arbitraire. Ce projet se concentre donc sur le développement et l’implémentation d’un algorithme pour calculer des sensibilités d’ordre n arbitraire dans le continu.

Nous pouvons regrouper les approches de calcul des sensibilités d’écoulement en 3 catégories distinctes :

- Méthode des Différences Finies (DF) ;
- Méthode d’Équations des Sensibilités (MES) ;
- Différentiation Automatique (DA).

Ici, nous définissons brièvement les techniques de calcul des sensibilités. Ces méthodes seront abordées avec plus de détails au chapitre 2.

Méthodes des Différences Finies : La méthode des DF est la plus simple à appliquer, mais pas forcément la plus efficace. En effet, cette technique est très coûteuse en termes de calculs, car elle requiert des résolutions complètes du système (sensibilités d’ordre 0) pour au moins deux valeurs pour chaque paramètre d’intérêt (voir Allaire, 2005).

Méthode d’Équations des Sensibilités : Cette technique se divise en 2 types d’approches (Turgeon et al., 2002; Mahieu et al., 2005; Pelletier et al., 2008) selon l’ordre dans lequel nous effectuons les opérations de discrétisation et de différentiation. Dans le cas «discrétisation-puis-différentiation», nous discrétisons les équations différentielles régissant le comportement du système d’intérêt pour obtenir un système d’équations algébriques. Puis, nous différencions ces équations discrétisées pour obtenir les sensibilités des variables discrétisées (e.g. : $\frac{\partial u_h}{\partial \alpha}$). Nous parlons de Méthode d’Équations des Sensibilités Discrètes (MESD). Dans le second cas, «différentiation-puis-discrétisation», nous commençons par dériver les EDP par rapport au paramètres d’intérêt pour obtenir un système régissant le comportement de la sensibilité continue de la solution (e.g. : $\frac{\partial u}{\partial \alpha}$). Ensuite, nous discrétisons les EDP différenciées. Nous obtenons $\frac{\partial u}{\partial \alpha}|_h$, où u est la solution du système et α est le paramètre d’intérêt. Nous parlons alors de la Méthode d’Équations des Sensibilités Continues (MESD).

Différentiation Automatique : La façon courante de procéder est de discrétiser, dans un premier temps, le système d’équations. Ensuite, nous utilisons les outils de Différentiation Automatique qui appliquent automatiquement la règle de dérivation des fonctions composées au code pour les équations discrètes. Ces outils calculent les dérivées exactes du système discret à des coûts assez importants.

Le présent mémoire est structuré comme suit. Dans le chapitre 2, nous présentons une revue

bibliographique complète sur les méthodes de résolution et les applications des sensibilités. Nous y présentons les diverses techniques utilisées pour calculer les sensibilités. Puis, nous les comparons afin de mettre en évidence les avantages et désavantages de chacune. L'intérêt de recueillir les sensibilités d'ordre élevé y est également clairement exposé. Cela nous amène alors à définir clairement les objectifs du présent travail. Par la suite, dans le chapitre 3, nous décrivons la formulation générique développée pour générer automatiquement les équations de sensibilités d'ordre élevé n à q paramètres de valeur (dérivées croisées). Le chapitre 4 présente l'algorithme générique développé pour générer et résoudre automatiquement les équations de sensibilités d'ordre n arbitrairement élevé. Nous y expliquons brièvement la façon dont nous ajustons le code de la Méthode des Éléments Finis (MEF) pour résoudre les sensibilités d'ordre n . Cela permet alors de mieux percevoir la généricité sur l'ensemble de la méthode développée et implémentée. Au chapitre 5, nous décrivons les processus de vérification et de validation de la méthode proposée que nous réalisons sur un problème non-linéaire 1D. Nous y dévoilons également les résultats obtenus pour chaque cas d'étude.

CHAPITRE 2 REVUE DE LITTÉRATURE

Afin de bien situer la contribution scientifique apportée par le présent projet, nous passons en revue les travaux de recherches antérieures qui traitent les équations de sensibilités ainsi que les principales applications des sensibilités. Cela permet de donner une vue globale des connaissances acquises dans ce domaine. Jusqu'à ce jour, de nombreuses méthodes de calcul des sensibilités ont été développées. On peut regrouper ces méthodes de calcul des sensibilités en 3 grandes catégories (Pelletier et al., 2008; Turgeon et al., 2002) :

1. Différences Finies (DF),
2. Méthode d'Équations des Sensibilités (MES),
3. Différentiation Automatique (DA).

Ces méthodes sont exploitées dans de multiples domaines tels que l'aérodynamique (Papadimitriou et Giannakoglou, 2013; Duvigneau et Pelletier, 2006; Godfrey et Cliff, 1998; Borggaard et Burns, 1997), la thermodynamique (Dowding et Blackwell, 2001; Haftka et Malkus, 1981), la biomécanique (Etienne et al., 2006; Katz et al., 2000) et la mécanique des structures (Lu et al., 2013; Van Keulen et al., 2005; Rao et Rahman, 2005; Mahieu et al., 2005).

Ce chapitre est structuré comme suit. Dans un premier temps, nous définissons les concepts de base. Puis, nous présentons le contexte d'utilisation des sensibilités pour bien faire ressortir l'utilité et l'intérêt des sensibilités d'ordre élevé. Ensuite, nous abordons et comparons les principales techniques d'évaluation des sensibilités. Enfin, nous énonçons les objectifs établis pour ce projet.

2.1 Définitions et concepts de base

Dans le but de faciliter la compréhension du sujet, cette section présente la terminologie courante pour les sensibilités.

2.1.1 Variable dépendante

Les variables dépendantes correspondent aux inconnues apparaissant dans les équations différentielles qui décrivent le comportement d'un écoulement. Par exemple, le champ de vitesse, la température et la pression de l'écoulement sont des variables dépendantes (voir Turgeon et al., 2002). On parle aussi de «variables d'état».

2.1.2 Paramètre

Il existe deux types de paramètres, soient les paramètres de valeur et les paramètres de forme (Mahieu et al., 2005; Turgeon et al., 2002; Turgeon, 2001). Un paramètre de valeur se définit comme étant une caractéristique mesurable qui n'affecte pas la géométrie du domaine de calcul. Par exemple, en mécanique des fluides, les propriétés physiques telles que la masse volumique et la viscosité sont des paramètres de valeur. Contrairement aux paramètres de valeur, les paramètres de forme ont une incidence directe sur la géométrie du domaine de calcul et affectent donc aussi les conditions aux limites, puisque la position des frontières du domaine varie (i.e. : déplacement des frontières). Par exemple, l'épaisseur d'une aile d'avion peut être considérée comme un paramètre de forme pour l'écoulement autour d'une aile d'avion, puisque lorsqu'elle varie, la forme de l'aile change également.

2.1.3 Sensibilité

Une sensibilité d'ordre n se définit comme la dérivée d'ordre n d'une variable dépendante par rapport aux paramètres d'intérêt. Les équations de sensibilités d'ordre n correspondent donc à la différentiation d'ordre n des équations d'état du système.

Afin d'illustrer le principe des sensibilités, considérons le problème unidimensionnel de conduction de chaleur dans un matériau composite représenté par la figure 2.1 (voir Mahieu, 2003).

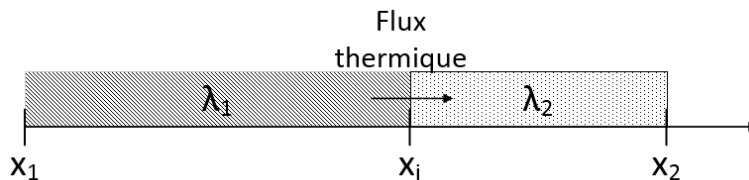


Figure 2.1 Problème de conduction de chaleur.

Ce problème est défini par :

$$\frac{d}{dx} \left(\lambda \frac{dT}{dx} \right) = 0, \quad (2.1)$$

où les conditions limites sont :

$$T(x_1) = T_1, \quad (2.2)$$

$$T(x_2) = T_2. \quad (2.3)$$

La température est donc une fonction de x linéaire par morceaux, telle qu'illustrée par la figure 2.2, dont la solution analytique est :

$$T(x) = \begin{cases} A_1x + B_1 & \text{si } x \in [x_1, x_i], \\ A_2x + B_2 & \text{si } x \in [x_i, x_2], \end{cases} \quad (2.4)$$

où A_1 , A_2 , B_1 et B_2 sont des constantes.

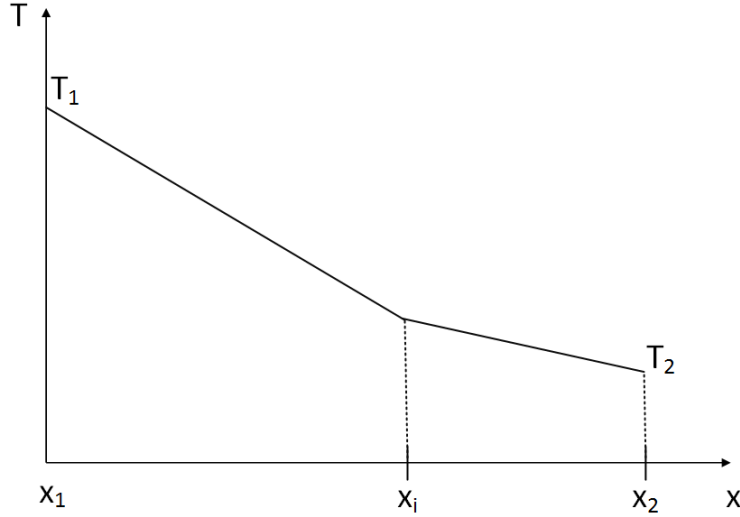


Figure 2.2 Courbe de la température T en fonction de x .

La température T dépend non seulement de l'abscisse, mais également des paramètres du problème. Le champ T s'écrit donc sous la forme :

$$T = T(x; T_1, T_2, \lambda_1, \lambda_2, x_1, x_i, x_2), \quad (2.5)$$

où T est la variable dépendante, x est la variable indépendante et T_1 , T_2 , λ_1 , λ_2 , x_1 , x_i ainsi que x_2 sont les paramètres.

La sensibilité d'ordre 1 par rapport à l'un de ces paramètres α s'écrit :

$$S_T^{(\alpha^{(1)})} = \frac{\partial T}{\partial \alpha} \text{ où } \alpha \in \{T_1, T_2, \lambda_1, \lambda_2, x_1, x_i, x_2\}. \quad (2.6)$$

Elle représente la réponse du système à une perturbation unitaire du paramètre α (voir série de Taylor d'ordre 1).

2.2 Applications et utilités des sensibilités

Afin de mieux saisir l'intérêt que l'on porte à l'étude des sensibilités, nous passons en revue les principales applications des sensibilités. Les applications abordées sont : l'optimisation de design en ingénierie, l'évaluation rapide de solutions perturbées ainsi que l'analyse de sensibilités (voir Mahieu, 2003).

2.2.1 Design optimal

L'optimisation consiste à déterminer la configuration idéale des paramètres pour maximiser ou minimiser une fonction objectif représentant l'indice de performance du système. Un problème d'optimisation typique comporte les éléments suivants (Gunzburger, 2000) :

- \mathbf{X} , les variables d'état ;
- $\boldsymbol{\alpha}$, les variables de contrôle ou paramètres ;
- $F(\mathbf{X}(\boldsymbol{\alpha}); \boldsymbol{\alpha})$, la fonction objectif ;
- $\mathbf{G}(\mathbf{X}; \boldsymbol{\alpha}) = \mathbf{0}$, les équations d'état (contraintes).

L'optimisation (design optimal) consiste à trouver les valeurs des paramètres $\boldsymbol{\alpha}$ minimisant ou maximisant la fonction objectif :

$$\mathcal{F}(\boldsymbol{\alpha}) = F(\mathbf{X}(\boldsymbol{\alpha}); \boldsymbol{\alpha}), \quad (2.7)$$

où $\mathbf{X} \in \mathbb{R}^{n_x}$, $\boldsymbol{\alpha} \in \mathbb{R}^{n_\alpha}$ et $\mathcal{F}(\boldsymbol{\alpha}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\alpha} \rightarrow \mathbb{R}$. Les variables d'état \mathbf{X} correspondent aux variables dépendantes. Pour les problèmes d'écoulement, les variables d'état peuvent être associées au champ de vitesse, à la température, à la pression, la masse volumique et/ou à d'autres éléments qui décrivent le mouvement du fluide (Gunzburger, 2000). Par exemple, dans les équations de Navier-Stokes incompressible, \mathbf{X} représente le champ de vitesse \mathbf{u} et la pression P . La dépendance implicite de \mathbf{X} (variables d'état) par rapport à $\boldsymbol{\alpha}$ est déterminée par le biais des contraintes \mathbf{G} (équations d'état, *e.g.* : les équations de Navier-Stokes) auxquelles le problème est soumis :

$$\mathbf{G}(\mathbf{X}; \boldsymbol{\alpha}) = \mathbf{0}, \quad (2.8)$$

où $\mathbf{G} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\alpha} \rightarrow \mathbb{R}^{n_x}$.

Les techniques d'optimisation se distinguent par le type d'information auquel elles ont recours pour trouver un optimum. Par exemple, une technique d'ordre 0 requiert uniquement

l'évaluation de la fonction objectif elle-même, alors qu'une technique d'ordre 2 nécessite l'évaluation des dérivées d'ordre 0 à 2 de la fonction $\mathcal{F}(\boldsymbol{\alpha})$.

Par exemple, pour une technique d'ordre 1, nous devons évaluer la fonction objectif et ses dérivées premières par rapport aux paramètres $\boldsymbol{\alpha}$ qui sont représentées symboliquement par :

$$\frac{d\mathcal{F}}{d\boldsymbol{\alpha}} = \frac{\partial F}{\partial \boldsymbol{\alpha}} + \frac{\partial F}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\boldsymbol{\alpha}}, \quad (2.9)$$

où $\frac{\partial F}{\partial \boldsymbol{\alpha}}$ et $\frac{\partial F}{\partial \mathbf{X}}$ sont des dérivées explicites connues, tandis que $\frac{d\mathbf{X}}{d\boldsymbol{\alpha}}$ correspond aux sensibilités qui sont inconnues. Les dérivées explicites se déduisent facilement, car F est exprimée en fonction de \mathbf{X} et $\boldsymbol{\alpha}$. Par contre, le terme $\frac{d\mathbf{X}}{d\boldsymbol{\alpha}}$ de l'équation (2.9) étant inconnu, nous devons générer et résoudre les équations de sensibilités d'ordre 1 afin d'évaluer la dérivée première de la fonction objectif par rapport à $\boldsymbol{\alpha}$. Les façons de procéder pour la génération et la résolution de sensibilités font l'objet de la prochaine section (voir section 2.3).

Généralement, nous observons que les techniques d'optimisation d'ordre élevé sont plus efficaces (Mahieu, 2003). Toutefois, ces techniques requièrent le calcul des sensibilités d'ordre plus élevé. Il est donc approprié de s'intéresser aux calculs des sensibilités d'ordre élevé n afin d'améliorer l'efficacité de l'optimisation de design en ingénierie.

2.2.2 Évaluation rapide de solutions perturbées

Dans certains cas, nous nous intéressons aux effets que peut avoir la perturbation d'un paramètre α sur une variable d'état d'un problème donné. Par exemple, pour un profil NACA0012, nous cherchons à connaître l'influence de l'angle d'attaque sur l'écoulement laminaire \mathbf{u} autour du profil. Afin d'éviter de résoudre à chaque fois le système d'équations d'état pour des valeurs différentes de l'angle d'attaque α , nous proposons d'approximer la nouvelle solution $\mathbf{u}(\mathbf{x}, \alpha_0 + \Delta\alpha)$ par le biais d'une série de Taylor développée autour du point α_0 . La solution perturbée est donc obtenue par :

$$\mathbf{u}(\mathbf{x}, \alpha_0 + \Delta\alpha) = \mathbf{u}(\mathbf{x}, \alpha_0) + \sum_{i=1}^n S_u^{(\alpha^{(i)})}(\mathbf{x}, \alpha_0) \frac{(\Delta\alpha)^i}{i!} + O(\Delta\alpha^{n+1}), \quad (2.10)$$

où α_0 est la valeur de référence de l'angle d'attaque α et $O(\Delta\alpha^{n+1})$ désigne l'erreur de troncature à l'ordre $n + 1$. Il est à noter qu'il est possible d'approximer une solution par le biais de la série de Taylor lorsqu'on fait varier plus d'un paramètre à la fois. La série de Taylor est alors exprimée sous la forme suivante :

$$u(\mathbf{x}, \boldsymbol{\alpha}_0 + \Delta\boldsymbol{\alpha}) = \mathbf{u}(\mathbf{x}, \alpha_0) + \sum_{i=1}^n \frac{(\Delta\boldsymbol{\alpha} \cdot \nabla)^i u(\mathbf{x}, \boldsymbol{\alpha}_0)}{i!} + O(\Delta\boldsymbol{\alpha}^{n+1}), \quad (2.11)$$

où $\Delta\boldsymbol{\alpha} = (\Delta\alpha_1, \Delta\alpha_2, \dots, \Delta\alpha_q)$ et $\nabla = (\frac{\partial}{\partial\alpha_1}, \frac{\partial}{\partial\alpha_2}, \dots, \frac{\partial}{\partial\alpha_q})$.

Pour extrapoler la solution à l'ordre n , il faut donc résoudre les systèmes d'équations d'état et des sensibilités d'ordre 1 à n en (\mathbf{x}, α_0) . Dans le cas de la mécanique des fluides, les systèmes d'équations d'état sont généralement non-linéaires, alors que les équations des sensibilités sont toujours linéaires. Les coûts de calculs engendrés par les sensibilités sont donc nettement inférieurs à ceux pour la résolution du système d'équations d'état. L'extrapolation par série de Taylor devient alors une option avantageuse dans le cas où nous cherchons à évaluer l'effet de la variation d'un paramètre quelconque sur une variable d'état d'un système, puisque cela évite de résoudre à chaque fois (pour chaque $\Delta\alpha$) le système non-linéaire des équations d'état.

Mahieu (Mahieu, 2003) compare les séries de Taylor d'ordre 1 et 2 pour extrapoler le champ de ligne de courant autour d'un profil NACA0012. Il utilise les solutions et les sensibilités calculées en $\alpha_0 = 0^\circ$ pour évaluer les séries de Taylor d'ordre 1 et 2 à des angles d'attaque de 3° , 4° , 5° et 6° . La figure 2.3, tirée de (Mahieu, 2003), illustre les lignes de courant qui représentent le champ de vitesse extrapolé à 3° , 4° , 5° et 6° . La première colonne présente les résultats obtenus par extrapolation linéaire, alors que la deuxième colonne présente l'extrapolation quadratique. Quant à la troisième colonne, elle présente les solutions obtenues par la résolution des équations de Navier-Stokes en $\alpha = \alpha_0 + \Delta\alpha$.

En considérant les résultats de Mahieu pour l'extrapolation de la solution, nous constatons que plus l'ordre d'extrapolation est élevé, plus la solution approximée est précise et fiable (voir figure 2.3). En effet, l'extrapolation d'ordre 2 est beaucoup plus proche de la solution exacte que l'extrapolation d'ordre 1. L'extrapolation d'ordre 2, à partir de $\alpha = 0^\circ$, est particulièrement précise pour une perturbation de α inférieure ou égale à 3° . Par contre, la précision se détériore lorsqu'on considère des perturbations plus importantes.

En augmentant l'ordre n de la série de Taylor, il est possible d'augmenter la plage de valeurs des paramètres tout en maintenant une erreur de troncature négligeable (Soulat et al., 2013). Néanmoins, cela ne fonctionne pas toujours. Dans certains cas, l'intervalle de convergence de la série de Taylor est très petit. Il s'ensuit que la plage de valeurs des paramètres pour laquelle nous obtenons une approximation précise est très petite également. Pour augmenter

l'ordre de la série de Taylor, il est nécessaire de calculer les sensibilités d'ordre n plus élevé. Il semble alors tout à fait pertinent et utile de développer une méthode permettant de calculer les sensibilités d'ordre n arbitrairement élevé.

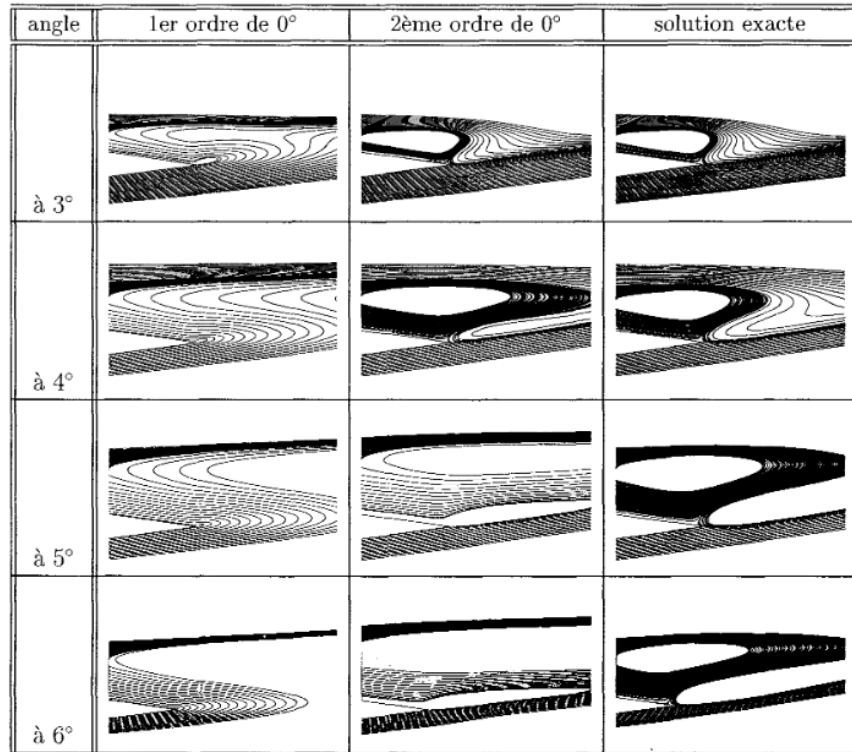


Figure 2.3 Zone de recirculation - Extrapolation en $\alpha_0 = 0^\circ$ (Mahieu, 2003).

2.2.3 Analyse de sensibilités

En calculant les sensibilités, nous cherchons à identifier de façon claire et précise les zones sensibles du domaine en fonction des variations des paramètres du système. Cela permet de déterminer les paramètres qui ont une forte influence sur les variables d'état du modèle ou inversement les paramètres qui ont une faible influence. Il devient alors possible de cartographier le comportement des variables d'état par rapport aux paramètres en se concentrant sur une plage de valeurs spécifiques.

Le calcul des sensibilités d'ordre élevé a pour avantage de décrire les interactions et/ou les liens de dépendance entre les paramètres considérés (dérivées croisées). Par exemple, si nous prenons le terme $\frac{\partial^3 u}{\partial \alpha_1 \partial \alpha_2 \partial \alpha_3}$, nous obtenons alors des informations sur la façon dont les paramètres α_1 , α_2 et α_3 interagissent entre eux.

Les sensibilités d'ordre élevé permettent d'enrichir la base d'information à laquelle l'ingénieur concepteur ou l'ingénieur analyste a accès. En effet, les informations qui découlent des sensibilités permettent d'améliorer la qualité de la caractérisation du modèle étudié et de mieux appréhender ce dernier.

2.3 Techniques d'évaluation des sensibilités

Il existe peu de méthodes pour calculer les sensibilités d'écoulement d'ordre élevé. En fait, la majorité des méthodes rencontrées en mécanique des fluides se concentrent sur le calcul de sensibilités d'ordre 1 (Sherman et al., 1996; Turgeon, 2001; Putko et al., 2002; Bakr et al., 2009). Quelques études abordent les sensibilités d'ordre 2 (Sherman et al., 1996; Mahieu, 2003), tandis qu'un nombre négligeable se penche sur l'analyse des sensibilités d'ordre égal ou supérieur à 3 (Papadimitriou et Giannakoglou, 2013). Puisque l'utilisation de sensibilités d'ordre élevé permet d'améliorer la précision de la solution dans divers contextes (se référer à la section 2.2), l'objectif de cet ouvrage est de présenter une méthode des sensibilités continues d'ordre arbitrairement élevé n . Dans cette section, nous présentons les principales méthodes de calcul des sensibilités (dérivées) : Différences Finies, Méthode d'Équations des Sensibilités et Différentiation Automatique. Comme ce travail ne s'intéresse pas spécialement aux problèmes d'optimisation, la Méthode des Variables Adjointes (Gunzburger, 1999, 2000) n'y est pas abordée.

2.3.1 Différences Finies

La méthode des DF est une des techniques les plus simples pour évaluer la sensibilité d'un système d'équations. Pour ce faire, il s'agit de résoudre le système d'équations en fixant d'abord le paramètre considéré α_i à une valeur donnée afin d'évaluer les variables d'état. Puis, selon l'ordre souhaité pour les sensibilités, nous introduisons successivement des perturbations $\Delta\alpha_i$ au niveau du paramètre α_i . On recalcule alors les valeurs des variables d'état (*e.g.* : la vitesse u) pour chacun de ces systèmes perturbés. Dans le cas des sensibilités d'ordre 1 par rapport au paramètre α pour la vitesse u , nous utilisons l'équation (2.12) qui correspond aux DF centrées :

$$S_u^{(\alpha^{(1)})} = \frac{\partial u}{\partial \alpha} = \frac{u(\alpha_0 + \Delta\alpha) - u(\alpha_0 - \Delta\alpha)}{2\Delta\alpha}. \quad (2.12)$$

Bien que cette méthode soit simple, le choix du pas de variation du paramètre $\Delta\alpha$ peut s'avé-

rer délicat. Si la perturbation est trop grande, les sensibilités approximées seront imprécises, tandis qu'une valeur trop petite de $\Delta\alpha$ introduira des erreurs numériques d'arrondi (Turgeon, 2001; Iott et al., 1985) dans (2.12).

Plus l'ordre des sensibilités est élevé, plus les dérivées qu'on tente de calculer sont d'ordre élevé. En se référant aux formules des DF (voir Allaire, 2005), nous constatons que cela revient à résoudre le système des équations d'état pour un grand nombre de valeurs du paramètre α . Cela entraîne donc des coûts très importants.

Cette technique s'avère donc très coûteuse en temps de calcul, car elle nécessite la résolution du problème pour de nombreuses valeurs pour chaque paramètre d'intérêt. Dans le cas des paramètres de forme, des difficultés techniques risquent d'apparaître, car le domaine de calcul, de même que le maillage, ne sont pas forcément identiques pour des valeurs différentes d'un paramètre spécifique.

2.3.2 Méthode d'Équations des Sensibilités

La MES est conçue de manière à générer et résoudre numériquement les équations de sensibilités continues. Cette technique se subdivise en deux types d'approches : les *sensibilités discrètes* (MESD) et les *sensibilités continues* (MESC) (Pelletier et al., 2008; Mahieu et al., 2005; Turgeon et al., 2002; Stanley, 2000). La principale différence entre ces deux approches réside dans l'ordre dans lequel nous effectuons les opérations de discrétisation et de différentiation sur les EDP. La figure 2.4 illustre ces deux approches.

La première approche (MESD) consiste d'abord à discrétiser les équations décrivant l'état (*e.g.* : l'écoulement) pour, ensuite, les différentier par rapport aux paramètres (Pelletier et al., 2008; Komkov et al., 1986). La résolution de ces équations donne les sensibilités exactes du système discret. On parle alors de sensibilités discrètes. Quant à la seconde approche (MESC), elle différentie en premier les EDP. Puis, elle procède à la discrétisation pour résoudre numériquement les EDP des sensibilités (voir Borggaard et Burns, 1997). Cette approche, basée sur les équations continues, permet donc d'approximer les sensibilités exactes de la solution analytique.

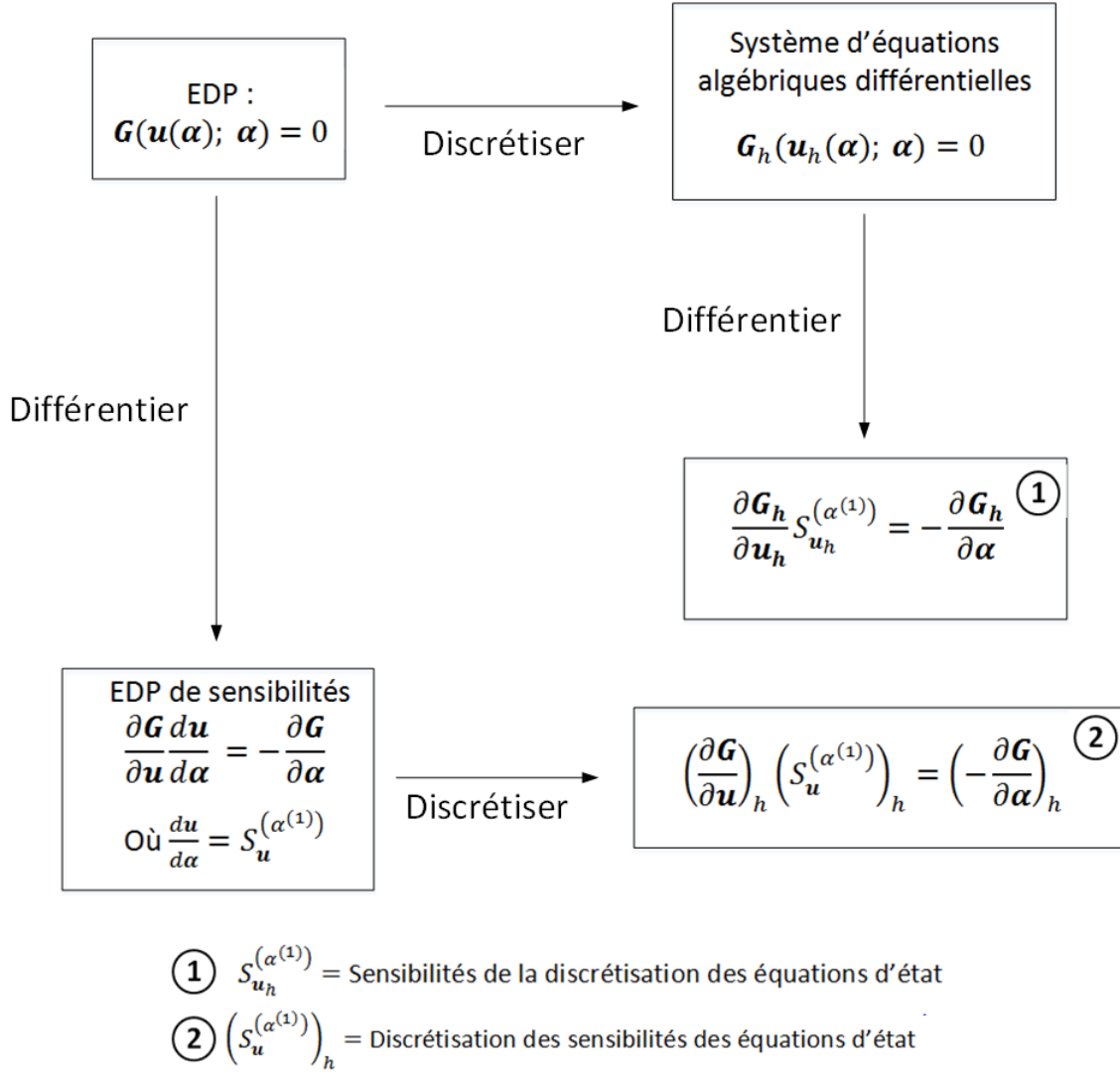


Figure 2.4 Approches des méthodes de calcul des sensibilités.

2.3.3 Différentiation automatique

La Différentiation Automatique consiste à discrétiser le problème, pour ensuite le différencier et générer les équations de sensibilités discrètes (voir Van Keulen et al., 2005). Dans un premier temps, les équations du système à résoudre sont donc discrétisées. Puis, les dérivées de ces équations, par rapport aux paramètres, sont évaluées grâce à un logiciel de différenciation qui assure l'application automatique de la règle de la dérivation en chaîne. Cet outil s'appuie sur le fait que tous les programmes informatiques sont constitués d'un ensemble d'opérateurs (+, ×, etc.) et de fonctions élémentaires (sin, log, etc.) qui sont définis par le langage de programmation utilisé (voir Homescu, 2011). Subséquemment, l'outil de DA spécifie les dérivées

usuelles de ces fonctions.

Il existe de multiples outils pour la Différentiation Automatique développés dans divers langages. On retrouve ADIFOR, TAPENADE, ADOL-C, etc. Dans de nombreux cas, les outils de DA prennent en charge le calcul des sensibilités d'ordre supérieur à 2 (voir Cotte, 2010). C'est le cas de l'outil développé par l'entreprise Fluorem et l'École Centrale de Lyon (Soulat et al., 2013) qui traite les sensibilités d'ordre n arbitrairement élevé.

La DA se distingue de la méthode par DF par le fait que les dérivées calculées par la DA utilisent les formules de dérivées exactes (voir Gilbert et al., 1991). Cela implique donc uniquement des erreurs d'arrondi. De plus, contrairement aux outils de différentiation symbolique (*e.g.* : Maple), les outils de DA permettent de dériver des fonctions complexes qui comportent plusieurs variables. Bien que ce type de méthode s'applique aisément peu importe la longueur et le niveau de complexité du code (voir Bischof et al., 1996), elle sollicite néanmoins beaucoup de ressources informatiques (Bischof et al., 1996; Homescu, 2011).

2.4 Comparaison des méthodes

En mécanique des fluides, la relation entre une variable d'écoulement et la variation d'un paramètre d'intérêt α est généralement non-linéaire. Les sensibilités d'ordre plus élevé n permettent de mieux caractériser cette relation de même que les dépendances entre divers paramètres du système, ce qui explique l'intérêt de les calculer (pour plus de détails se référer à la section 2.2). Dans le cadre de ce projet, nous nous intéressons donc au développement d'une méthode générique qui différentie les équations d'un système jusqu'à un ordre n pour un nombre q de paramètres pour ensuite résoudre les sensibilités qui s'y rattachent. Ici, la méthode développée se limite aux sensibilités par rapport aux paramètres de valeur. Toutefois, nous souhaitons développer une méthode qui pourra être éventuellement adaptée aux cas des paramètres de forme. Nos travaux sont orientés de manière à pouvoir appliquer cette méthode au cas particulier des équations de Navier-Stokes qui modélisent les écoulements de fluides. Cependant, la méthode que nous proposons au sein de ce mémoire pourrait aisément être adaptée à n'importe quel système d'équations différentielles. Dans cette section, nous comparons les différentes techniques de calcul des sensibilités présentées dans la section 2.3. Cela permet d'expliquer les raisons qui nous motivent à privilégier la MES pour le développement d'une méthode générique calculant les sensibilités d'ordre n .

Si nous considérons les Différences Finies pour le calcul des sensibilités d'ordre élevé, celles-ci s'avèrent peu efficaces et très coûteuses, puisqu'elles exigent de nombreuses résolutions exhaustives du système d'équations d'état (sensibilités d'ordre 0) pour différentes valeurs du ou des paramètres d'intérêt. Dans le cas où le système est non-linéaire, la méthode de résolution itérative peut conduire à un nombre considérable d'itérations lors de la résolution de chacun des systèmes. Cette méthode n'est donc pas performante, puisqu'elle engendre des coûts de calculs substantiels et que le choix de valeurs des perturbations, pour s'assurer de la fiabilité des résultats, peut s'avérer une tâche ardue, voire impossible.

Quant à la MES, elle nécessite la résolution du système des sensibilités pour chaque paramètre considéré. Toutefois, dans le cas des paramètres de valeur, le membre de gauche des systèmes d'équations des sensibilités est identique peu importe l'ordre. Il est alors possible de factoriser et conserver la matrice pour éviter de la recalculer à chaque fois. Il est à noter que cela n'est pas applicable lors de l'utilisation de la MESC avec une approche lagrangienne et des paramètres de forme.

Tel que mentionné dans la sous-section 2.3.3, il existe des outils de différentiation (voir Soulat et al., 2013) qui résolvent des problèmes de sensibilités d'ordre élevé. Ceux-ci sont puissants, car ils génèrent automatiquement le code de calcul pour les sensibilités discrètes (Mahieu et al., 2005). Néanmoins, ces outils ne sont pas forcément efficaces, puisque les ressources informatiques utilisées ainsi que le temps de calcul pour la différentiation et la résolution des sensibilités sont considérables (Bischof et al., 1996; Verma, 2000; Heimbach et al., 2005; Homescu, 2011). Le temps de calcul pour résoudre les équations de sensibilités peut même être plus significatif que celui pour résoudre les équations d'état. De plus, dans certains problèmes discrets, les fonctions impliquées sont non-différentiables (*e.g.* : limiteur, stabilisateur, etc.). L'utilisation de la différentiation automatique devient alors complexe.

Par ailleurs, pour les techniques qui travaillent avec les sensibilités du système discret (*e.g.* : MESD), les sensibilités de maillage doivent être considérées. Cela engendre un effort de calcul supplémentaire. Subséquemment, la discrétisation du problème par le biais d'un maillage non structuré adaptatif est pratiquement impossible, car il devient alors complexe d'estimer les sensibilités de mailles qui sont établies par la différence entre deux maillages. Cela provient du fait que le maillage, pour un paramètre perturbé, ne respecte pas forcément la topologie initiale, particulièrement quand il s'agit d'un paramètre de forme. En effet, le domaine et les conditions aux limites du problème varient avec les paramètres de forme. Le traitement de

sensibilités discrètes présente donc un inconvénient significatif, car l’adaptation de maillage permet habituellement d’obtenir des maillages de qualité et de raffiner les éléments aux endroits où il y a une plus forte variation pour ainsi obtenir des résultats plus précis (Turgeon, 2001).

Contrairement à l’approche discrète, l’approche en continu (MESC) a pour avantage d’éliminer la nécessité de considérer les sensibilités du maillage (Borggaard et Burns, 1997; Borggaard et Verma, 2000; Turgeon et al., 2002). Effectivement, la différentiation du système ayant lieu avant la discrétisation, il n’y a pas de dérivées du maillage, mais plutôt un maillage des dérivées (sensibilités). Cela est beaucoup plus commode dans l’éventualité où l’on souhaite traiter un système comportant des paramètres de forme. Actuellement, nous exploitons la MESC principalement à l’ordre 1 et 2 (Turgeon, 2001; Mahieu, 2003). L’utilisation de cette méthode à des ordres plus élevés est beaucoup plus rare, puisque cela entraîne des équations qui sont lourdes à manipuler et à gérer. Malgré cette difficulté liée à l’écriture des équations de sensibilités, nous constatons que la Méthode d’Équations des Sensibilités Continues est préférable, puisque la différentiation des équations est directe. Contrairement aux sensibilités discrètes, il n’est pas essentiel d’exprimer la dépendance du maillage en fonction du paramètre d’intérêt afin de différentier les termes qui y sont associés.

Le tableau 2.1 résume les points apportés dans les paragraphes précédents. Il permet de visualiser et de comparer rapidement les avantages et les limites de chacune des techniques de calcul des sensibilités. De plus, ce tableau donne un indicatif pour la précision et le coût de calcul de ces techniques. La notation $Coût(u)$ correspond au coût engendré par la technique pour calculer les variables d’état (*e.g.* : la vitesse u), alors que $Coût(S_u^{(i)})$ représente le coût de calcul pour les sensibilités d’ordre i .

Tableau 2.1 Tableau résumant les diverses techniques de calcul des sensibilités.

Technique	Avantages	Limites	Précision des calculs	Coût approximatif
Méthode des DF	-Simplicité de l'implémentation.	-Choix difficile du pas $\Delta\alpha$. -Non applicable aux maillages qui varient avec les paramètres d'intérêt.	-Erreur de troncature. -Erreur numérique provenant du calcul des variables d'état	$\geq 2 * Coût(u)$, selon l'ordre des sensibilités calculées
MESC	-Linéarité des équations des sensibilités. -Discretisation séparée pour les équations d'état et les équations de sensibilités. -Pas de sensibilités de mailles.	-Pas de méthodes développées à un ordre élevé, car les équations de sensibilités sont lourdes à gérer.	-Erreur numérique des solutions et des sensibilités d'ordre inférieur à n .	$\approx Coût(u) + Coût(S_u^{(1)}) + Coût(S_u^{(2)}) + \dots + Coût(S_u^{(n)})$, où $Coût(u) \gg Coût(S_u^{(i)})$ pour $i = 1, \dots, n$
MESD	-Consistance des sensibilités. -Linéarité des équations des sensibilités.	-Présence de sensibilités de maillage. -Maillage identique pour écoulement et sensibilités.	-Sensibilités exactes du système discret. -Précision dépendante de la discrétisation.	Variable selon la technique utilisée pour la différentiation.
DA	-Exactitude des dérivées (pas d'erreur de troncature). -Génération automatique du code pour calculer les sensibilités.	-Utilisation impossible d'un maillage non-structuré adaptatif.	-Erreur numérique provenant du calcul des variables d'état. -Précision dépendante de la discrétisation.	$\approx Coût(u) + 5 * Coût(u) * n$, où n est l'ordre maximal des sensibilités étudiées.

Ici, pour notre méthode générique traitant les équations de sensibilités jusqu'à un ordre arbitraire n , on privilégie une méthode basée sur les équations de sensibilités continues, soit la MESC. Ce choix provient essentiellement du fait que la résolution des équations d'état (sensibilités d'ordre 0) et le calcul des sensibilités se réalisent dans des procédés séparés. Un autre avantage avec la MESC est que nous n'avons pas à considérer les sensibilités de mailles, puisqu'on dérive les EDP après les avoir discrétisées et non l'inverse. En revanche, dans le cas de la MESD, les sensibilités discrètes dépendent directement du choix de discrétisation des variables d'état diminuant ainsi la précision et le contrôle sur le calcul des sensibilités. Enfin, la MESC nous permet également de discrétiser le problème avec un maillage non structuré adaptatif ce qui permet d'obtenir des résultats plus précis en raffinant les éléments aux endroits où les variations sont plus importantes. Cela ne peut être envisagé avec la MESD.

2.5 Objectifs de recherche

L'objectif principal de ce projet est de concevoir un algorithme générique qui génère et stocke les équations des sensibilités continues d'ordre élevé n , où n est arbitraire, pour ensuite les résoudre par la Méthode des Éléments Finis. Pour ce faire, les composantes des équations de sensibilités sont automatiquement définies dans une structure de données générique et

résolues successivement de l'ordre 0 jusqu'à l'ordre n qui, lui, est déterminé par l'utilisateur. Dans un premier temps, ce projet se concentre uniquement sur les sensibilités pour les paramètres de valeur, mais éventuellement nous projetons d'étendre l'algorithme aux paramètres de forme.

Le projet sera donc réalisé en respectant les objectifs spécifiques suivants :

- Développer l'algorithme général pour le calcul des sensibilités d'ordre n ,
- Implanter l'algorithme dans un programme qui utilise la MEF,
- Vérifier et valider l'implémentation de l'algorithme au moyen de la Méthode des Solutions Manufacturées (MSM),
- Expérimenter la méthode sur des problèmes classiques en 1D,
- Démontrer l'efficacité de la méthode par rapport aux autres méthodes existantes par l'extrapolation de solutions par la série de Taylor.

L'algorithme pour le calcul des sensibilités d'ordre n a pour but d'accroître considérablement la précision des solutions extrapolées par la série de Taylor dans l'espace des paramètres et ce, sur de plus larges plages de variation des paramètres. Par ailleurs, nous visons également l'amélioration de la qualité de l'analyse, de la caractérisation et de l'optimisation du comportement des écoulements considérés. Les coûts généraux (ressources, effort de calcul, etc.) engendrés par cet algorithme doivent néanmoins demeurer raisonnables.

CHAPITRE 3 MÉTHODE DE L'ÉQUATION DES SENSIBILITÉS D'ORDRE ARBITRAIRE POUR UN PROBLÈME 1D

Ce chapitre présente une formulation générique de l'équation des sensibilités continues. Cette formulation est dite générique, car elle permet d'écrire automatiquement les équations de sensibilités d'ordre arbitrairement élevé n pour un nombre quelconque q de paramètre(s). Cette formulation est la base de l'algorithme générique que nous avons implémenté pour la génération automatique des équations de sensibilités d'ordre n . Nous expliciterons cet algorithme au chapitre 4.

La première section de ce chapitre présente les équations modèles 1D que nous utilisons pour expliquer et illustrer l'approche générique. Dans la seconde section, nous présentons la formulation générique pour les dérivées d'ordre n . Nous décrivons d'abord le théorème du multinôme de Newton, car il joue un rôle important dans l'approche générique que nous proposons. Ensuite, nous présentons la formulation générale des équations de sensibilités d'ordre n en faisant le lien entre celle-ci et le théorème du multinôme de Newton. Nous considérons d'abord les sensibilités par rapport à un seul paramètre. Puis, nous abordons les sensibilités avec des paramètres mixtes (dérivées croisées). Finalement, dans la dernière section, nous prenons les équations de Navier-Stokes pour montrer comment nous pouvons appliquer la formulation générale des dérivées pour générer les équations de sensibilités.

3.1 Problème 1D

Afin d'illustrer l'intérêt que présente une formulation générique pour la MES à l'ordre arbitraire n , nous utilisons un problème régi par un système d'équations 1D présentant plusieurs des caractéristiques des équations de Navier-Stokes. Ce problème est décrit par l'équation de Burgers couplée avec l'équation linéaire de convection-diffusion similaire à l'équation d'énergie. Ces équations doivent donc être résolues de manière couplée. Le problème est défini par le système suivant :

$$\rho u \cdot \nabla u = \nabla \cdot (\mu \nabla u) - \beta T + f_s, \quad (3.1)$$

$$\rho c_p u \cdot \nabla T = \nabla \cdot (\lambda \nabla T) + q_s, \quad (3.2)$$

où u et T sont les variables dépendantes inconnues, tandis que ρ , μ et c_p sont les propriétés physiques et f_s et q_s sont des termes sources qui s'avèrent très utiles pour la vérification du code par la méthode des solutions manufacturées (voir chapitre 5). Nous désignons l'équa-

tion (3.1) comme étant l'EDP de u et l'équation (3.2) comme étant l'EDP de T .

En plus de dépendre des coordonnées spatiales, les variables dépendantes, les propriétés physiques et les termes sources peuvent aussi dépendre de manière explicite ou implicite d'un ou plusieurs paramètres. Dans le présent travail, nous considérons uniquement les paramètres de valeur, c'est-à-dire ceux qui n'affectent pas la géométrie du domaine de calcul. En d'autres mots, les coordonnées spatiales ne dépendent pas des paramètres α . Nous pouvons donc exprimer les dépendances des variables dépendantes de la manière suivante :

$$u = u(x; \alpha), \quad (3.3)$$

$$T = T(x; \alpha), \quad (3.4)$$

et celle des propriétés physiques de manière similaire :

$$\rho = \rho(x; \alpha), \quad (3.5)$$

$$\mu = \mu(x; \alpha), \quad (3.6)$$

$$\beta = \beta(x; \alpha), \quad (3.7)$$

$$c_p = c_p(x; \alpha), \quad (3.8)$$

$$\lambda = \lambda(x; \alpha). \quad (3.9)$$

3.2 Formulation générale des équations de sensibilités à l'ordre n

La plupart des termes du système défini par (3.1) et (3.2) prennent la forme de produits de fonctions. Lorsque nous dérivons ces équations, la règle de la dérivée d'un produit conduit à une prolifération des termes dont le nombre augmente avec l'ordre de la dérivée. Considérons à titre d'exemple le terme βT , un produit de 2 fonctions (β et T). Ses dérivées par rapport au paramètre α sont :

$$\text{ordre 1 : } \frac{d(\beta T)}{d\alpha} = \frac{\partial \beta}{\partial \alpha} T + \beta \frac{\partial T}{\partial \alpha}, \quad (3.10)$$

$$\text{ordre 2 : } \frac{d^2(\beta T)}{d\alpha^2} = \frac{\partial^2 \beta}{\partial \alpha^2} T + 2 \frac{\partial \beta}{\partial \alpha} \frac{\partial T}{\partial \alpha} + \beta \frac{\partial^2 T}{\partial \alpha^2}, \quad (3.11)$$

$$\text{ordre 3 : } \frac{d^3(\beta T)}{d\alpha^3} = \frac{\partial^3 \beta}{\partial \alpha^3} T + 3 \frac{\partial^2 \beta}{\partial \alpha^2} \frac{\partial T}{\partial \alpha} + 3 \frac{\partial \beta}{\partial \alpha} \frac{\partial^2 T}{\partial \alpha^2} + \beta \frac{\partial^3 T}{\partial \alpha^3}. \quad (3.12)$$

Le tableau 3.1 illustre la croissance du nombre de termes dans la n -ième dérivée de produits de 1 à 4 fonctions. Ici, m désigne le nombre de fonctions dans un produit et n l'ordre de

la dérivée. Ainsi, chaque ligne du tableau, à l'exclusion de la première, contient le nombre de termes obtenus pour les dérivées d'ordre 0 à 10 d'un produit de m fonctions. Quant à la valeur de m , elle est indiquée dans la première colonne de chacune de ces lignes. À l'inverse, chaque colonne du tableau donne le nombre de termes résultant de la dérivation à l'ordre n d'un produit de m fonctions. La valeur de cet ordre n est indiquée dans la première ligne de chacune de ces colonnes. Pour une dérivée d'ordre donné, nous observons donc que le nombre de termes obtenus augmente avec m . Par exemple, la dérivée d'ordre 10 sur un produit de 3 fonctions donne 66 termes, alors que la dérivée du même ordre appliquée à un produit de 4 fonctions en crée 286.

Tableau 3.1 Nombre de termes résultant de la dérivée à l'ordre n d'un produit de 1 à 4 fonctions.

		n : Ordre de la dérivée										
		0	1	2	3	4	5	6	7	8	9	10
m	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	2	3	4	5	6	7	8	9	10	11
	3	1	3	6	10	15	21	28	36	45	55	66
	4	1	4	10	20	35	56	84	120	165	220	286

À partir du tableau 3.1, nous pouvons évaluer le nombre de termes apparaissant dans les équations de sensibilités d'ordre 0 à 10 du système régi par (3.1) et (3.2). Nous dénombrons d'abord les termes qui découlent des dérivées d'ordre n pour chacun des produits de fonctions apparaissant dans les équations (3.1) et (3.2) et puis, nous en faisons la somme. Le tableau 3.2 illustre le nombre de termes des équations de sensibilités d'ordre $n = 0$ à 10 pour les EDP de u (équation (3.1)) et de T (équation (3.2)). À titre d'exemple, l'EDP de u :

$$\rho u \cdot \nabla u = \nabla \cdot (\mu \nabla u) - \beta T + f_s, \quad (3.13)$$

est constituée des produits de fonctions suivants :

- $\rho u \cdot \nabla u$: produit de 3 fonctions, soient ρ , u et ∇u ,
- $\nabla \cdot (\mu \nabla u)$: produit de 2 fonctions, soient μ et ∇u ,
- βT : produit de 2 fonctions, soient β et T ,
- f_s : une seule fonction.

Considérons les sensibilités d'ordre 2 de cette EDP (dérivée seconde), nous obtenons :

- 6 termes pour $\rho u \cdot \nabla u$,

- 3 termes pour $\nabla \cdot (\mu \nabla u)$,
- 3 termes pour βT ,
- 1 terme pour f_s ,

pour un total de 13 termes.

Tableau 3.2 Nombre de termes pour chacune des Équations aux Dérivées Partielles selon l'ordre n .

n	Nombre de termes dans $\frac{\partial^n}{\partial \alpha^n}$ (3.1)	Nombre de termes dans $\frac{\partial^n}{\partial \alpha^n}$ (3.2)
0	4	3
1	8	7
2	13	14
3	19	25
4	26	41
5	34	63
6	43	92
7	53	129
8	64	175
9	76	231
10	89	298

Le tableau 3.2 illustre clairement la croissance fulgurante de la complexité des EDP des sensibilités lorsque l'ordre n augmente. Cette croissance est encore plus rapide si l'on considère 2 paramètres ou plus en raison des dérivées croisées qui apparaissent. Il est à noter qu'on associe la complexité au nombre de termes dans les équations de sensibilités.

Reprenons l'exemple du produit βT et considérons les sensibilités par rapport à 2 paramètres α_1 et α_2 . Pour les sensibilités d'ordre 1, nous avons :

$$\frac{d(\beta T)}{d\alpha_1} = \frac{\partial \beta}{\partial \alpha_1} T + \beta \frac{\partial T}{\partial \alpha_1}, \quad (3.14)$$

$$\frac{d(\beta T)}{d\alpha_2} = \frac{\partial \beta}{\partial \alpha_2} T + \beta \frac{\partial T}{\partial \alpha_2}. \quad (3.15)$$

À l'ordre 2, nous avons :

$$\frac{d^2(\beta T)}{d\alpha_1^2} = \frac{\partial^2 \beta}{\partial \alpha_1^2} T + 2 \frac{\partial \beta}{\partial \alpha_1} \frac{\partial T}{\partial \alpha_1} + \beta \frac{\partial^2 T}{\partial \alpha_1^2}, \quad (3.16)$$

$$\frac{d^2(\beta T)}{d\alpha_1 d\alpha_2} = \frac{\partial^2 \beta}{\partial \alpha_1 \partial \alpha_2} T + \frac{\partial \beta}{\partial \alpha_1} \frac{\partial T}{\partial \alpha_2} + \frac{\partial \beta}{\partial \alpha_2} \frac{\partial T}{\partial \alpha_1} + \beta \frac{\partial^2 T}{\partial \alpha_1 \partial \alpha_2}, \quad (3.17)$$

$$\frac{d^2(\beta T)}{d\alpha_2^2} = \frac{\partial^2 \beta}{\partial \alpha_2^2} T + 2 \frac{\partial \beta}{\partial \alpha_2} \frac{\partial T}{\partial \alpha_2} + \beta \frac{\partial^2 T}{\partial \alpha_2^2}. \quad (3.18)$$

Finalement, à l'ordre 3, nous avons les expressions suivantes :

$$\frac{d^3(\beta T)}{d\alpha_1^3} = \frac{\partial^3 \beta}{\partial \alpha_1^3} T + 3 \frac{\partial^2 \beta}{\partial \alpha_1^2} \frac{\partial T}{\partial \alpha_1} + 3 \frac{\partial \beta}{\partial \alpha_1} \frac{\partial^2 T}{\partial \alpha_1^2} + \beta \frac{\partial^3 T}{\partial \alpha_1^3}, \quad (3.19)$$

$$\frac{d^3(\beta T)}{d\alpha_1^2 d\alpha_2} = \frac{\partial^3 \beta}{\partial \alpha_1^2 \partial \alpha_2} T + 2 \frac{\partial^2 \beta}{\partial \alpha_1 \partial \alpha_2} \frac{\partial T}{\partial \alpha_1} + \frac{\partial^2 \beta}{\partial \alpha_1^2} \frac{\partial T}{\partial \alpha_2} + 2 \frac{\partial \beta}{\partial \alpha_1} \frac{\partial^2 T}{\partial \alpha_1 \partial \alpha_2} + \frac{\partial \beta}{\partial \alpha_2} \frac{\partial^2 T}{\partial \alpha_1^2} + \beta \frac{\partial^3 T}{\partial \alpha_1^2 \partial \alpha_2}, \quad (3.20)$$

$$\frac{d^3(\beta T)}{d\alpha_1 d\alpha_2^2} = \frac{\partial^3 \beta}{\partial \alpha_1 \partial \alpha_2^2} T + 2 \frac{\partial^2 \beta}{\partial \alpha_1 \partial \alpha_2} \frac{\partial T}{\partial \alpha_2} + \frac{\partial^2 \beta}{\partial \alpha_2^2} \frac{\partial T}{\partial \alpha_1} + 2 \frac{\partial \beta}{\partial \alpha_2} \frac{\partial^2 T}{\partial \alpha_1 \partial \alpha_2} + \frac{\partial \beta}{\partial \alpha_1} \frac{\partial^2 T}{\partial \alpha_2^2} + \beta \frac{\partial^3 T}{\partial \alpha_2^2 \partial \alpha_1}, \quad (3.21)$$

$$\frac{d^3(\beta T)}{d\alpha_2^3} = \frac{\partial^3 \beta}{\partial \alpha_2^3} T + 3 \frac{\partial^2 \beta}{\partial \alpha_2^2} \frac{\partial T}{\partial \alpha_2} + 3 \frac{\partial \beta}{\partial \alpha_2} \frac{\partial^2 T}{\partial \alpha_2^2} + \beta \frac{\partial^3 T}{\partial \alpha_2^3}. \quad (3.22)$$

Nous constatons que le fait de considérer les sensibilités mixtes (2 paramètres ou plus) accroît considérablement le nombre de termes résultant de la règle de dérivation du produit. Par exemple, à l'ordre 3 le terme $\frac{\partial^3(\beta T)}{\partial \alpha_1^2 \alpha_2}$ produit 6 termes alors que $\frac{\partial^3(\beta T)}{\partial \alpha_1^3}$ en engendre 4.

Il s'avère donc inconcevable d'écrire explicitement les EDP des sensibilités pour des ordres supérieurs à 2 et de coder explicitement le solveur requis comme l'ont fait Mahieu et Ilinca et al. pour les sensibilités d'ordre 2. Il faut donc automatiser tout le processus de différentiation des EDP et celui de la construction des systèmes d'éléments finis pour résoudre les EDP des sensibilités d'ordre n . Pour ce faire, il faut déterminer une méthode générale pour générer automatiquement les EDP de sensibilités d'ordre n .

Dans cette section, nous nous concentrons donc sur la formulation générale des équations de sensibilités d'ordre n à q paramètres. Dans un premier temps, nous présentons la formule du multinôme de Newton en raison de ses ressemblances avec la forme des termes des EDP de sensibilités à l'ordre n à un paramètre. Puis, nous explicitons cette analogie au moyen d'un exemple. Cela nous conduit alors à la formulation générale des équations de sensibilités

d'ordre n à un paramètre. Enfin, nous expliquons comment nous nous sommes inspirés de cette analogie pour formuler de façon générique les équations de sensibilités d'ordre n à plusieurs paramètres.

3.2.1 Le multinôme de Newton

La formule du multinôme de Newton (Nyblom, 1996; Flusser et Francia, 2000) est une relation donnant le développement d'une somme d'un nombre fini m de termes (monômes) lorsque celle-ci est élevée à une puissance entière n :

$$(b_1 + b_2 + \dots + b_m)^n = \sum_{\substack{k_1, k_2, \dots, k_m \in \mathbb{N} \\ k_1 + k_2 + \dots + k_m = n}} \frac{n!}{k_1! k_2! \dots k_m!} b_1^{k_1} b_2^{k_2} \dots b_m^{k_m}. \quad (3.23)$$

Dans l'équation (3.23), k_1, k_2, \dots, k_m sont des entiers, b_j représente le j -ième monôme et k_j est l'exposant affecté au monôme b_j de $(b_1 + b_2 + \dots + b_m)^n$. On appelle *coefficient multinomial* (Mazkewitsch, 1963; Krantz et Parks, 2002) l'entier $\frac{n!}{k_1! k_2! \dots k_m!}$. Pour démontrer que le coefficient multinomial est un entier, nous l'écrivons sous la forme équivalente suivante :

$$\frac{n!}{k_1! k_2! \dots k_m!} = \frac{n!}{k_1! (n - k_1)!} \cdot \frac{(n - k_1)!}{k_2! (n - k_1 - k_2)!} \cdot \dots \cdot \frac{(n - k_1 - \dots - k_{m-1})!}{k_m! (n - k_1 - \dots - k_{m-1} - k_m)!}. \quad (3.24)$$

Sachant que $n = k_1 + k_2 + \dots + k_m$, nous obtenons :

$$\frac{n!}{k_1! k_2! \dots k_m!} = \frac{n!}{k_1! (n - k_1)!} \cdot \frac{(n - k_1)!}{k_2! (n - k_1 - k_2)!} \cdot \dots \cdot \frac{(n - k_1 - \dots - k_{m-1})!}{k_m! 0!}. \quad (3.25)$$

Par définition, $C_k^n = \frac{n!}{k!(n-k)!} \in \mathbb{N}$ (voir annexe A). À partir de l'équation (3.24), nous déduisons que le coefficient multinomial est équivalent à un produit d'entier :

$$\frac{n!}{k_1! k_2! \dots k_m!} = C_{k_1}^n \cdot C_{k_2}^{n-k_1} \cdot \dots \cdot C_{k_m}^{n-k_1-\dots-k_{m-1}}. \quad (3.26)$$

Il est important de se rappeler que le coefficient multinomial est toujours un entier.

L'opérateur de sommation dans l'équation (3.23) inclut toutes les combinaisons possibles des exposants k_j telles que $\sum_{j=1}^m k_j = n$. Ici, le terme *combinaison* s'interprète au sens de l'analyse combinatoire (se référer à l'annexe A). Lorsque nous parlons des combinaisons des exposants k_j telles que $\sum_{j=1}^m k_j = n$, cela fait référence à des combinaisons avec répétition. Ce problème se traduit par la recherche des façons de disposer n unités de puissance parmi m monômes

(donne la valeur des exposants).

Nous pouvons interpréter le problème d'une autre façon : nous cherchons les manières de multiplier n monômes en les sélectionnant dans l'ensemble $E = \{b_1, b_2, \dots, b_m\}$ constitué de m monômes en supposant qu'un même monôme peut être répété. L'ordre dans lequel apparaissent les monômes est sans importance. Il s'agit donc de combinaisons avec répétitions. Notre problème consiste alors à choisir n éléments, pouvant être répétés, parmi les m éléments de l'ensemble $E = \{b_1, b_2, \dots, b_m\}$. Le nombre de répétitions d'un élément b_j , pour $j = 1$ à m , est l'exposant affecté à ce même monôme b_j . Par conséquent, cela indique la valeur que prend l'exposant k_j .

À titre d'exemple, considérons l'expression $(b_1 + b_2 + b_3)^2$ et cherchons les combinaisons possibles des exposants k_j des monômes b_j telles que $k_1 + k_2 + k_3 = 2$ en vue d'écrire la forme polynomiale selon l'équation (3.23). Pour ce faire, nous trouvons alors l'ensemble des combinaisons avec répétitions de 2 éléments parmi les 3 éléments de l'ensemble $E = \{b_1, b_2, b_3\}$:

$$\begin{array}{lll} \{b_1, b_1\}, & \{b_2, b_2\}, & \{b_3, b_3\}, \\ \{b_1, b_2\}, & \{b_1, b_3\}, & \{b_2, b_3\}. \end{array}$$

Ces combinaisons correspondent aux façons de multiplier 2 monômes b_j pour $j = 1$ à 3. Maintenant, faisons le lien entre ces combinaisons et la valeur associée à chacun des exposants k_j pour $j = 1$ à m . Sachant que le nombre de répétitions observées pour chaque b_j équivaut à la valeur de k_j , nous obtenons :

$$\begin{array}{ll} \{b_1, b_1\} & \rightarrow k_1 = 2; k_2 = 0; k_3 = 0; \\ \{b_2, b_2\} & \rightarrow k_1 = 0; k_2 = 2; k_3 = 0; \\ \{b_3, b_3\} & \rightarrow k_1 = 0; k_2 = 0; k_3 = 2; \\ \{b_1, b_2\} & \rightarrow k_1 = 1; k_2 = 1; k_3 = 0; \\ \{b_1, b_3\} & \rightarrow k_1 = 1; k_2 = 0; k_3 = 1; \\ \{b_2, b_3\} & \rightarrow k_1 = 0; k_2 = 1; k_3 = 1. \end{array}$$

Comme la sommation de l'équation (3.23) porte sur la totalité des ensembles de k_j précé-

demment obtenus, la formule du multinôme de Newton nous donne :

$$(b_1 + b_2 + b_3)^2 = \frac{2!}{2!0!0!}b_1b_1 + \frac{2!}{0!2!0!}b_2b_2 + \frac{2!}{0!0!2!}b_3b_3 + \frac{2!}{1!1!0!}b_1b_2 + \frac{2!}{1!0!1!}b_1b_3 + \frac{2!}{0!1!1!}b_2b_3. \quad (3.27)$$

Dont la forme polynomiale compacte est :

$$(b_1 + b_2 + b_3)^2 = (b_1)^2 + (b_2)^2 + (b_3)^2 + 2b_1b_2 + 2b_1b_3 + 2b_2b_3. \quad (3.28)$$

Le nombre total s de ces combinaisons avec répétition s'obtient à l'aide de la formule (A.6). Pour l'exemple précédent, le nombre de combinaisons avec répétition lorsqu'on prend $k = 2$ éléments dans un ensemble contenant $n = 3$ éléments équivaut à :

$$C_2^{3+2-1} = \frac{(3+2-1)!}{2!(3-1)!} = 6.$$

Il est pertinent de remarquer que le nombre total de combinaisons avec répétition correspond au nombre total de termes additionnés dans le multinôme de Newton représenté par l'équation (3.23).

Par ailleurs, nous déduisons la valeur du coefficient multinomial à partir des valeurs trouvées pour chacun des ensembles de k_j ($j = 1$ à m). La forme générale du coefficient multinomial provient de la formule générale utilisée pour dénombrer les arrangements (voir définition dans la section A) d'un ensemble de n éléments parmi m éléments distincts dont les multiplicités respectives sont k_1, k_2, \dots, k_m . Le nombre d'arrangements pour ce type d'ensemble se calcule à partir de la formule :

$$\frac{n!}{k_1!k_2! \dots k_m!}. \quad (3.29)$$

Par exemple, si nous prenons $(b_1 + b_2 + b_3)^3$, nous avons $C_3^{3+3-1} = 10$ combinaisons de 3 monômes choisis parmi les 3 éléments de l'ensemble $E = \{b_1, b_2, b_3\}$, puisqu'il y a possibilité de répétition. Voici une de ces combinaisons :

$$\{b_1, b_1, b_2\} \rightarrow k_1 = 2; k_2 = 1; k_3 = 0.$$

Nous souhaitons connaître le nombre de façons d'arranger les $n = 3$ éléments de ce sous-ensemble, sachant qu'il contient l'élément b_1 , qui est répété 2 fois, et l'élément b_2 . En s'appuyant sur

l'équation (3.29), nous trouvons :

$$\frac{3!}{2!1!0!} = 3, \quad (3.30)$$

qui est égal au nombre d'arrangements. Ces 3 arrangements correspondent en fait à :

$$\{b_1, b_1, b_2\}, \quad \{b_1, b_2, b_1\}, \quad \{b_2, b_1, b_1\}. \quad (3.31)$$

Lorsqu'on développe la puissance n de la somme de m termes, on procède par distributivité. Ainsi, on distribue chaque monôme b_j (pour $j = 1$ à m) de la première parenthèse à chacun des m monômes de la deuxième parenthèse et ainsi de suite. Par exemple, pour $(b_1 + b_2 + b_3)^3$, nous avons :

$$(b_1 + b_2 + b_3)^3 = (b_1 + b_2 + b_3)(b_1 + b_2 + b_3)(b_1 + b_2 + b_3).$$

En distribuant les éléments de la première parenthèse à ceux de la seconde, nous obtenons :

$$(b_1 + b_2 + b_3)^3 = (b_1b_1 + b_1b_2 + b_1b_3 + b_2b_1 + b_2b_2 + b_2b_3 + b_3b_1 + b_3b_2 + b_3b_3)(b_1 + b_2 + b_3).$$

Enfin, en multipliant chacun des éléments obtenus précédemment avec ceux de la dernière parenthèse, nous retrouvons :

$$\begin{aligned} (b_1 + b_2 + b_3)^3 = & b_1b_1b_1 + \boxed{b_1b_1b_2} + b_1b_1b_3 + \\ & \boxed{b_1b_2b_1} + b_1b_2b_2 + b_1b_2b_3 + \\ & b_1b_3b_1 + b_1b_3b_2 + b_1b_3b_3 + \\ & \boxed{b_2b_1b_1} + b_2b_1b_2 + b_2b_1b_3 + \\ & b_2b_2b_1 + b_2b_2b_2 + b_2b_2b_3 + \\ & b_2b_3b_1 + b_2b_3b_2 + b_2b_3b_3 + \\ & b_3b_1b_1 + b_3b_1b_2 + b_3b_1b_3 + \\ & b_3b_2b_1 + b_3b_2b_2 + b_3b_2b_3 + \\ & b_3b_3b_1 + b_3b_3b_2 + b_3b_3b_3. \end{aligned} \quad (3.32)$$

Le développement explicite de la sommation des 3 monômes b_j élevé à la puissance 3 de l'équation (3.32) nous permet de retrouver les arrangements évoqués à l'équation (3.31). Nous les avons d'ailleurs encadrés. De plus, nous constatons que l'ensemble des termes de l'équation (3.32) correspond aux divers arrangements associés à chacune des 10 combinaisons

avec répétition de 3 éléments choisis dans l'ensemble $E = \{b_1, b_2, b_3\}$:

$$\begin{array}{ccccc} \{b_1, b_1, b_1\}, & \{b_1, b_1, b_2\}, & \{b_1, b_1, b_3\}, & \{b_1, b_2, b_2\}, & \{b_1, b_2, b_3\}, \\ \{b_1, b_3, b_3\}, & \{b_2, b_2, b_2\}, & \{b_2, b_2, b_3\}, & \{b_2, b_3, b_3\}, & \{b_3, b_3, b_3\}. \end{array}$$

Comme la multiplication est commutative, cela implique que l'ordre des n monômes multipliés dans un même terme est sans importance. Nous considérons donc que les arrangements qui contiennent les n mêmes éléments sont des termes identiques, bien que l'ordre dans lequel nous retrouvons leurs éléments soit différent. Nous pouvons donc les regrouper ensemble :

$$\begin{aligned} (b_1 + b_2 + b_3)^3 = & b_1 b_1 b_1 + 3b_1 b_1 b_2 + 3b_1 b_1 b_3 + 3b_1 b_2 b_2 + 6b_1 b_2 b_3 + \\ & 3b_1 b_3 b_3 + b_2 b_2 b_2 + 3b_2 b_2 b_3 + 3b_2 b_3 b_3 + b_3 b_3 b_3. \end{aligned} \quad (3.33)$$

En indiquant la puissance, de chacun des monômes b_j , nous obtenons :

$$\begin{aligned} (b_1 + b_2 + b_3)^3 = & (b_1)^3 + 3(b_1)^2 b_2 + 3(b_1)^2 b_3 + 3b_1 (b_2)^2 + 6b_1 b_2 b_3 + \\ & 3b_1 (b_3)^2 + (b_2)^3 + 3(b_2)^2 b_3 + 3b_2 (b_3)^2 + (b_3)^3. \end{aligned} \quad (3.34)$$

Pour chaque terme spécifique, le nombre de termes répétés correspond au coefficient multinomial devant ce même terme dans la forme polynomiale (voir (3.34)). Le nombre d'arrangements calculé pour chacune des combinaisons avec répétition des monômes b_j correspond donc au coefficient $\frac{n!}{k_1! k_2! \dots k_m!}$ placé devant chacun des termes $b_1^{k_1} b_2^{k_2} \dots b_m^{k_m}$ qui se retrouvent dans la forme développée du multinôme de Newton (voir équation (3.23)).

3.2.2 Analogie entre le théorème multinomial et la différentiation d'ordre n

La méthode que nous présentons génère les dérivées d'ordre n de produits de m fonctions en se basant sur la formule du multinôme de Newton (Mazkewitsch, 1963). Dans la présente sous-section, nous établissons d'abord l'analogie entre la différentiation des EDP pour les sensibilités d'ordre n à un seul paramètre et la puissance n de la somme de termes exprimée par la formule du multinôme de Newton. Ensuite, nous dévoilons la formulation générale qui en résulte pour la différentiation des termes des EDP par rapport à un seul paramètre. Cela nous permet alors de déduire la forme générale de l'équation de sensibilités d'ordre n à plusieurs paramètres.

Notre étude se limite aux paramètres de valeur. Les coordonnées définissant l'espace sont donc indépendantes du paramètre α . Cela implique que les paramètres étudiés n'ont aucun

impact sur la géométrie du domaine. L'opérateur ∇ qui effectue la différentiation par rapport aux coordonnées spatiales n'est donc pas touché par l'opérateur $\frac{\partial^n}{\partial \alpha^n}$. Pour l'exemple donné par les équations (3.37) et (3.38), la différentiation est réalisée uniquement sur la vitesse u sur laquelle le gradient opère, puisque les paramètres d'intérêt sont des paramètres de valeur $(\frac{\partial}{\partial \alpha}(\nabla u) = \nabla \frac{\partial u}{\partial \alpha})$.

Tout d'abord, nous présentons un exemple simple afin de clarifier comment notre méthode utilise la formule du multinôme de Newton. Considérons l'évaluation de la dérivée seconde du terme $\rho u \cdot \nabla u$ de l'équation (3.1) par rapport au paramètre α . Pour cet exemple, la variable correspond à la vitesse u et la propriété est la masse volumique ρ . Rappelons que n désigne l'ordre de différentiation considéré et m le nombre de fonctions dans le produit pour un terme donné. Pour obtenir la 2^{ième} dérivée par rapport à α , nous appliquons deux fois ($n = 2$) la règle de la dérivée du produit, par rapport à un paramètre α , au terme $\rho u \cdot \nabla u$ constitué du produit de trois fonctions ($m = 3$). La dérivée d'ordre 1 s'écrit :

$$\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha} = \frac{\partial \rho}{\partial \alpha} u \cdot \nabla u + \rho \frac{\partial u}{\partial \alpha} \cdot \nabla u + \rho u \cdot \nabla \frac{\partial u}{\partial \alpha}. \quad (3.35)$$

Puis, la dérivée d'ordre 2 :

$$\begin{aligned} \frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2} &= \frac{\partial^2 \rho}{\partial \alpha^2} u \cdot \nabla u + 2 \frac{\partial \rho}{\partial \alpha} \frac{\partial u}{\partial \alpha} \cdot \nabla u + 2 \frac{\partial \rho}{\partial \alpha} u \cdot \nabla \frac{\partial u}{\partial \alpha}, \\ &+ \rho \frac{\partial^2 u}{\partial \alpha^2} \cdot \nabla u + 2 \rho \frac{\partial u}{\partial \alpha} \cdot \nabla \frac{\partial u}{\partial \alpha} + \rho u \cdot \nabla \frac{\partial^2 u}{\partial \alpha^2}. \end{aligned} \quad (3.36)$$

Pour alléger l'écriture, nous adoptons la notation suivante :

$$\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha} = \rho^{(1)} u^{(0)} \cdot \nabla u^{(0)} + \rho^{(0)} u^{(1)} \cdot \nabla u^{(0)} + \rho^{(0)} u^{(0)} \cdot \nabla u^{(1)}, \quad (3.37)$$

$$\begin{aligned} \frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2} &= \rho^{(2)} u^{(0)} \cdot \nabla u^{(0)} + 2 \rho^{(1)} u^{(1)} \cdot \nabla u^{(0)} + 2 \rho^{(1)} u^{(0)} \cdot \nabla u^{(1)}, \\ &+ \rho^{(0)} u^{(2)} \cdot \nabla u^{(0)} + 2 \rho^{(0)} u^{(1)} \cdot \nabla u^{(1)} + \rho^{(0)} u^{(0)} \cdot \nabla u^{(2)}. \end{aligned} \quad (3.38)$$

où l'ordre des dérivées est indiqué par un exposant entre parenthèses. Par exemple, $u^{(1)} = \frac{\partial u}{\partial \alpha}$ et $u^{(2)} = \frac{\partial^2 u}{\partial \alpha^2}$ correspondent respectivement aux sensibilités d'ordre 1 et 2 de la vitesse u par rapport au paramètre α , tandis que $u^{(0)}$ est la vitesse elle-même.

Le multinôme de Newton nous permet de déterminer le nombre de termes résultant d'une

dérivée ainsi que les coefficients qui les affectent. En effet, il existe une analogie entre la différentiation d'ordre n d'un terme représenté par un produit de m fonctions et la puissance n d'une somme de m monômes. Considérons la somme elle-même ($n = 1$) de 3 monômes, b_1 , b_2 et b_3 ($m = 3$), et l'évaluation de son carré ($n = 2$) :

$$(b_1 + b_2 + b_3) = b_1^1 b_2^0 b_3^0 + b_1^0 b_2^1 b_3^0 + b_1^0 b_2^0 b_3^1, \quad (3.39)$$

$$(b_1 + b_2 + b_3)^2 = b_1^2 b_2^0 b_3^0 + 2b_1 b_2 b_3^0 + 2b_1 b_2^0 b_3 + b_1^0 b_2^2 b_3^0 + 2b_1^0 b_2 b_3 + b_1^0 b_2^0 b_3^2. \quad (3.40)$$

où $b_j^0 = 1$ pour $j = 1$ à 3. Si nous remplaçons respectivement les variables ρ , u et ∇u par b_1 , b_2 et b_3 dans les équations (3.37) et (3.38), nous pouvons aisément relever les correspondances entre les équations (3.37) et (3.39) ainsi que celles entre les équations (3.38) et (3.40). Pour le premier cas ($n = 1$), nous remarquons que chaque forme développée comporte 3 termes ayant les mêmes coefficients. Par ailleurs, pour chacun de ces termes, nous notons que les exposants de (3.39) et les ordres des dérivées des fonctions correspondantes de (3.37) coïncident :

$$\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha} : (b_1 + b_2 + b_3),$$

puisque :

$$\rho^{(1)} u^{(0)} \cdot \nabla u^{(0)} : b_1^1 b_2^0 b_3^0,$$

$$\rho^{(0)} u^{(1)} \cdot \nabla u^{(0)} : b_1^0 b_2^1 b_3^0,$$

$$\rho^{(0)} u^{(0)} \cdot \nabla u^{(1)} : b_1^0 b_2^0 b_3^1.$$

Il en va de même pour les équations (3.40) et (3.38) ($n = 2$) :

$$\frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2} : (b_1^2 + 2b_1 b_2 + 2b_1 b_3 + b_2^2 + 2b_2 b_3 + b_3^2),$$

puisque :

$$\rho^{(2)} u^{(0)} \cdot \nabla u^{(0)} : b_1^2 b_2^0 b_3^0,$$

$$\rho^{(1)} u^{(1)} \cdot \nabla u^{(0)} : b_1^1 b_2^1 b_3^0,$$

$$\rho^{(1)} u^{(0)} \cdot \nabla u^{(1)} : b_1^1 b_2^0 b_3^1,$$

$$\rho^{(0)} u^{(2)} \cdot \nabla u^{(0)} : b_1^0 b_2^2 b_3^0,$$

$$\rho^{(0)} u^{(1)} \cdot \nabla u^{(1)} : b_1^0 b_2^1 b_3^1,$$

$$\rho^{(0)} u^{(0)} \cdot \nabla u^{(2)} : b_1^0 b_2^0 b_3^2.$$

Cette analogie, entre la différentiation d'ordre n et la formule du multinôme de Newton, est valide pour toute dérivée d'ordre arbitraire n d'un produit d'un nombre quelconque m de fonctions par rapport à un paramètre donné (se référer à Mazkewitsch, 1963). Le multinôme de Newton nous permet donc de calculer le nombre s de termes qui découlent de la dérivée d'un produit de m fonctions. De plus, il permet de déterminer l'ordre des dérivées de chacune des m fonctions de même que les coefficients affectant chacun de ces s termes grâce à la formule du coefficient multinomial (voir (3.29)). Nous présentons dans les sous-sections 3.2.3 et 3.2.4 les formulations générales pour les dérivées de produits de m fonctions que nous avons déduites à partir de cette analogie entre le multinôme de Newton et la différentiation de produits de fonctions.

3.2.3 Formulation générale de l'équation de sensibilités à un paramètre

Afin de trouver la formulation générale de la dérivée d'ordre n_i d'un produit de m fonctions par rapport à un paramètre α_i , nous nous basons sur l'analogie précédente. Nous transposons alors les composantes du multinôme de Newton sur la dérivée explicite : nombre s de termes, coefficients de chacun des termes et puissance de chacun des m monômes. Reprenons l'exemple de la dérivée d'ordre 2 du terme $\rho u \cdot \nabla u$ de la section 3.2.2. En se basant sur l'équation (3.40) décrite par le multinôme de Newton (voir équation (3.23)) pour $m = 3$ et $n = 2$, nous trouvons les combinaisons de k_j (pour $j = 1$ à 3) et de ce fait, les coefficients $\frac{n!}{k_1! k_2! k_3!}$. Nous présentons ces éléments dans le tableau 3.3. Ce tableau contient également les termes de la dérivée d'ordre 2 du produit de 3 fonctions $\rho u \cdot \nabla u$ obtenus grâce à la transposition des composantes correspondantes du multinôme de Newton.

Tableau 3.3 Analogie entre les composantes de $\frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2}$ et celles de $(b_1 + b_2 + b_3)^2$.

No. terme	Termes du multinôme de Newton	Exposant k_j	Coefficient $\frac{n!}{k_1!k_2!k_3!}$	Transposition des composantes sur les termes de la dérivée
1	$b_1^2 b_2^0 b_3^0$	$k_1 = 2; k_2 = 0;$ $k_3 = 0;$	$\frac{2!}{2!0!0!} = 1$	$\rho^{(2)} u^{(0)} \cdot \nabla u^{(0)}$
2	$b_1^1 b_2^1 b_3^0$	$k_1 = 1; k_2 = 1;$ $k_3 = 0;$	$\frac{2!}{1!1!0!} = 2$	$2\rho^{(1)} u^{(1)} \cdot \nabla u^{(0)}$
3	$b_1^1 b_2^0 b_3^1$	$k_1 = 1; k_2 = 0;$ $k_3 = 1;$	$\frac{2!}{1!0!1!} = 2$	$2\rho^{(1)} u^{(0)} \cdot \nabla u^{(1)}$
4	$b_1^0 b_2^2 b_3^0$	$k_1 = 0; k_2 = 2;$ $k_3 = 0;$	$\frac{2!}{0!2!0!} = 1$	$\rho^{(0)} u^{(2)} \cdot \nabla u^{(0)}$
5	$b_1^0 b_2^1 b_3^1$	$k_1 = 0; k_2 = 1;$ $k_3 = 1;$	$\frac{2!}{0!1!1!} = 2$	$2\rho^{(0)} u^{(1)} \cdot \nabla u^{(1)}$
6	$b_1^0 b_2^0 b_3^2$	$k_1 = 0; k_2 = 0;$ $k_3 = 2;$	$\frac{2!}{0!0!2!} = 1$	$\rho^{(0)} u^{(0)} \cdot \nabla u^{(2)}$

L'analogie entre les termes du multinôme de Newton et ceux apparaissant dans l'expression explicite de la dérivée d'ordre n d'un produit de m fonctions, nous permet d'écrire la dérivée d'ordre 2 du produit $\rho u \cdot \nabla u$ ($m = 3$) à l'aide des composantes du multinôme de Newton (i.e. : les coefficients et les exposants k_j) pour le cas où $n = 2$ et $m = 3$:

$$\begin{aligned} \frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2} &= \frac{2!}{2!0!0!} \rho^{(2)} u^{(0)} \cdot \nabla u^{(0)} + \frac{2!}{1!1!0!} \rho^{(1)} u^{(1)} \cdot \nabla u^{(0)} + \frac{2!}{1!0!1!} \rho^{(1)} u^{(0)} \cdot \nabla u^{(1)} \\ &+ \frac{2!}{0!2!0!} \rho^{(0)} u^{(2)} \cdot \nabla u^{(0)} + \frac{2!}{0!1!1!} \rho^{(0)} u^{(1)} \cdot \nabla u^{(1)} + \frac{2!}{0!0!2!} \rho^{(0)} u^{(0)} \cdot \nabla u^{(2)}, \end{aligned} \quad (3.41)$$

où l'exposant entre parenthèses correspond à l'ordre des dérivées des fonctions. Nous attribuons l'ordre de la dérivée de chaque fonction (i.e. : ρ , u et ∇u) incluse dans chaque terme selon la valeur de l'exposant k_j qui lui est associé. Nous associons respectivement les exposants k_1 , k_2 et k_3 aux ordres des dérivées des fonctions ρ , u et ∇u . Afin de s'assurer que l'équation (3.41) correspond bien à la dérivée seconde de $\rho u \cdot \nabla u$, nous comparons les termes avec ceux obtenus dans l'équation (3.38) dans la sous-section précédente. Nous constatons

que les 2 équations sont identiques. Nous pouvons alors réécrire l'équation (3.41) au moyen d'une notation similaire à celle du multinôme de Newton :

$$\frac{\partial^2(\rho u \cdot \nabla u)}{\partial \alpha^2} = \sum_{k_1+k_2+k_3=2} \frac{2!}{k_1!k_2!k_3!} \rho^{(k_1)} u^{(k_2)} \cdot \nabla u^{(k_3)}, \quad (3.42)$$

où la sommation s'effectue sur l'ensemble des combinaisons de k_j , pour $j = 1$ à 3, telles que $\sum_{j=1}^3 k_j = 2$. Les exposants k_j entre parenthèses indiquent l'ordre de la dérivée de chacune des 3 fonctions.

À partir de l'équation (3.42), nous sommes en mesure d'établir l'expression générale de la dérivée d'ordre n_i par rapport à un paramètre α_i d'un produit de m fonctions :

$$\frac{\partial^{n_i}}{\partial \alpha_i^{n_i}} \left(\prod_{j=1}^m f_j \right) = \sum_{k_{i,1}+k_{i,2}+\dots+k_{i,m}=n_i} \frac{n_i!}{k_{i,1}!k_{i,2}!\dots k_{i,m}!} \prod_{j=1}^m f_j^{(\alpha_i^{(k_{i,j})})}, \quad (3.43)$$

où $k_{i,j}$ représente l'ordre de la dérivée de la fonction f_j par rapport au paramètre d'intérêt α_i . Par exemple, la valeur de $k_{1,2}$ correspond à l'ordre de la dérivée partielle de la fonction f_2 par rapport au paramètre α_1 . Notez que la sommation de l'équation (3.43) permet d'additionner les termes $\frac{n_i!}{k_{i,1}!k_{i,2}!\dots k_{i,m}!} \prod_{j=1}^m f_j^{(\alpha_i^{(k_{i,j})})}$ pour tous les sous-ensembles de $k_{i,j}$ tels que $\sum_{j=1}^m k_{i,j} = n_i$. Comme dans le cas du multinôme de Newton, nous représentons ces sous-ensembles au moyen d'un problème combinatoire. Il s'agit alors de déterminer les combinaisons constituées de n_i éléments pris parmi m éléments dans l'ensemble $E = \{f_1^{(\alpha_i)}, f_2^{(\alpha_i)}, \dots, f_m^{(\alpha_i)}\}$ en admettant qu'un même élément peut être répété. On parle alors de combinaisons avec répétition. Nous supposons que le nombre de répétitions d'un élément $f_j^{(\alpha_i)}$ équivaut à la valeur de $k_{i,j}$ et que, dans le cas où il y a 0 répétition de $f_j^{(\alpha_i)}$, nous considérons tout simplement que $k_{i,j} = 0$. Nous dénombrons ces combinaisons en utilisant directement la formule (A.6) pour $C_{n_i}^{n_i+m-1}$. Notons que l'équation (3.43) correspond à la formule de Leibniz décrivant le relation des dérivées d'ordre n de produit de m fonctions. Cette formule est démontrée à l'annexe D.

Clarifions l'application de ce procédé à une dérivée plutôt qu'à la puissance d'une somme comme dans le cas du multinôme de Newton. À titre d'exemple, nous cherchons la forme explicite de la dérivée $\frac{\partial^3}{\partial \alpha_1^3}(f_1 f_2 f_3)$ en utilisant l'équation (3.43). Pour ce faire, nous devons trouver les sous-ensembles de $k_{1,j}$ tels que $\sum_{j=1}^{m=3} k_{1,j} = n_1 = 3$. C'est-à-dire que nous cherchons les sous-ensembles de 3 éléments que l'on peut former en choisissant ceux-ci dans l'ensemble

$E = \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\}$ ($n_1 = 3$ et $m = 3$). Nous avons alors $C_3^{3+3-1} = 10$ combinaisons :

$$\begin{aligned} & \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_1^{(\alpha_1)}\}, \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_2^{(\alpha_1)}\}, \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_3^{(\alpha_1)}\}, \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\}, \\ & \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\}, \{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\}, \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\}, \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\}, \\ & \{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\}, \{f_3^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\}. \end{aligned}$$

Prenons la combinaison $\{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\}$. On sait que le nombre de répétitions observées pour chaque élément $f_j^{(\alpha_i)}$ d'une combinaison est égal à la valeur de l'exposant $k_{i,j}$ de l'équation (3.43). Par conséquent, les exposants $k_{1,j}$ ($j = 1$ à 3) associés à la combinaison $\{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\}$ sont :

$$\{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\} \rightarrow k_{1,1} = 1; k_{1,2} = 2; k_{1,3} = 0;$$

Le terme $\frac{n_1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \prod_{j=1}^3 f_j^{\binom{k_{1,j}}{\alpha_1^{(k_{1,j})}}}$ de l'équation (3.43) lié à cette combinaison est donc :

$$\frac{3!}{1!2!0!} f_1^{\binom{1}{\alpha_1^{(1)}}} f_2^{\binom{2}{\alpha_1^{(2)}}} f_3^{\binom{0}{\alpha_1^{(0)}}}.$$

L'exposant entre parenthèses affectant α_i est l'ordre de la dérivée par rapport à α_i de la fonction f_j . Ainsi, pour notre exemple, nous avons :

$$f_1^{\binom{1}{\alpha_1^{(1)}}} f_2^{\binom{2}{\alpha_1^{(2)}}} f_3^{\binom{0}{\alpha_1^{(0)}}} = \frac{\partial f_1}{\partial \alpha_1} \frac{\partial^2 f_2}{\partial \alpha_1^2} f_3. \quad (3.44)$$

Nous pouvons alors écrire explicitement la dérivée de $\frac{\partial^3}{\partial \alpha_1^3} (f_1 f_2 f_3)$ à partir de l'équation (3.43) en déterminant les valeurs des $k_{1,j}$ associées à chacune des combinaisons obtenues précédemment. Pour ce faire, nous comptons le nombre de répétitions de chaque élément $f_j^{(\alpha_1)}$:

$$\begin{aligned} \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_1^{(\alpha_1)}\} & \rightarrow k_{1,1} = 3; k_{1,2} = 0; k_{1,3} = 0; \\ \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_2^{(\alpha_1)}\} & \rightarrow k_{1,1} = 2; k_{1,2} = 1; k_{1,3} = 0; \\ \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_3^{(\alpha_1)}\} & \rightarrow k_{1,1} = 2; k_{1,2} = 0; k_{1,3} = 1; \\ \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\} & \rightarrow k_{1,1} = 1; k_{1,2} = 2; k_{1,3} = 0; \\ \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\} & \rightarrow k_{1,1} = 1; k_{1,2} = 1; k_{1,3} = 1; \\ \{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\} & \rightarrow k_{1,1} = 1; k_{1,2} = 0; k_{1,3} = 2; \\ \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\} & \rightarrow k_{1,1} = 0; k_{1,2} = 3; k_{1,3} = 0; \end{aligned}$$

$$\begin{aligned}
\{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 2; k_{1,3} = 1; \\
\{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 1; k_{1,3} = 2; \\
\{f_3^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 0; k_{1,3} = 3.
\end{aligned}$$

Ainsi, nous effectuons la sommation $\sum_{k_{1,1}+k_{1,2}+k_{1,3}=3}$ sur tous ces sous-ensembles de $k_{1,j}$. La dérivée d'ordre 3 par rapport au paramètre α_1 est constituée de la somme de 10 termes (produits de fonctions) :

$$\begin{aligned}
\frac{\partial^3}{\partial \alpha_1^3} \left(\prod_{j=1}^3 f_j \right) &= \frac{3!}{3!0!0!} f_1^{(\alpha_1^{(3)})} f_2^{(\alpha_1^{(0)})} f_3^{(\alpha_1^{(0)})} + \frac{3!}{2!1!0!} f_1^{(\alpha_1^{(2)})} f_2^{(\alpha_1^{(1)})} f_3^{(\alpha_1^{(0)})} + \\
&\frac{3!}{2!0!1!} f_1^{(\alpha_1^{(2)})} f_2^{(\alpha_1^{(0)})} f_3^{(\alpha_1^{(1)})} + \frac{3!}{1!2!0!} f_1^{(\alpha_1^{(1)})} f_2^{(\alpha_1^{(2)})} f_3^{(\alpha_1^{(0)})} + \\
&\frac{3!}{1!1!1!} f_1^{(\alpha_1^{(1)})} f_2^{(\alpha_1^{(1)})} f_3^{(\alpha_1^{(1)})} + \frac{3!}{1!0!2!} f_1^{(\alpha_1^{(1)})} f_2^{(\alpha_1^{(0)})} f_3^{(\alpha_1^{(2)})} + \\
&\frac{3!}{0!3!0!} f_1^{(\alpha_1^{(0)})} f_2^{(\alpha_1^{(3)})} f_3^{(\alpha_1^{(0)})} + \frac{3!}{0!2!1!} f_1^{(\alpha_1^{(0)})} f_2^{(\alpha_1^{(2)})} f_3^{(\alpha_1^{(1)})} + \\
&\frac{3!}{0!1!2!} f_1^{(\alpha_1^{(0)})} f_2^{(\alpha_1^{(1)})} f_3^{(\alpha_1^{(2)})} + \frac{3!}{0!0!3!} f_1^{(\alpha_1^{(0)})} f_2^{(\alpha_1^{(0)})} f_3^{(\alpha_1^{(3)})}.
\end{aligned} \tag{3.45}$$

L'équation (3.43) considère la dérivée d'ordre n_i par rapport à un seul paramètre α_i pour un terme (i.e. : produit de m fonctions) de l'EDP. Nous devons appliquer cette formulation générale à chacun des produits de fonctions présents dans l'EDP. Ainsi, nous devons dériver à l'ordre n_i chacun des termes contenus dans l'EDP par rapport au paramètre α_i et les assembler en les séparant par les mêmes opérateurs (+ ou -) qui se trouvent dans l'équation d'état (ordre 0). Par exemple, si nous considérons l'EDP (3.1), la dérivée d'ordre n_i par rapport au paramètre α_i s'écrit :

$$\frac{\partial^{n_i}(\rho u \cdot \nabla u)}{\partial \alpha_i^{n_i}} = \frac{\partial^{n_i}(\nabla \cdot (\mu \nabla u))}{\partial \alpha_i^{n_i}} - \frac{\partial^{n_i}(\beta T)}{\partial \alpha_i^{n_i}} + \frac{\partial^{n_i}(f_s)}{\partial \alpha_i^{n_i}}.$$

En appliquant la formulation générale donnée par l'équation (3.43) à chacun des produits de fonctions ($\rho u \cdot \nabla u$, $\nabla \cdot (\mu \nabla u)$, βT et f_s), nous obtenons l'équation de sensibilités d'ordre n_i par rapport à un paramètre α_i pour l'EDP (3.1). Dans le cadre de ce travail, la forme de l'équation (3.43) est fondamentale, car c'est sur elle que s'appuie notre algorithme pour générer automatiquement les équations de sensibilités d'ordre n et enregistrer leurs composantes dans une structure de données générique.

3.2.4 Formulation générale de l'équation de sensibilités avec q paramètre(s)

La difficulté supplémentaire pour les équations de sensibilités avec 2 paramètres ou plus ($q \geq 2$) réside dans l'apparition des dérivées croisées. Pour générer les équations des sensibilités avec un nombre quelconque q de paramètres, nous devons appliquer tour à tour chaque dérivée partielle d'ordre n_i par rapport aux différents paramètres d'intérêt α_i pour $i = 1$ à q . Cela signifie que la formule (3.43), basée sur la formule du multinôme de Newton, est appliquée successivement pour $i = 1$ à q . Notons que nous supposons que les fonctions dérivées sont suffisamment régulières pour que l'ordre d'évaluation des dérivées croisées n'influence pas le résultat (e.g. : $\frac{\partial^2}{\partial \alpha_1 \partial \alpha_2} = \frac{\partial^2}{\partial \alpha_2 \partial \alpha_1}$).

//Il suffit en fait que les dérivées partielles existent au voisinage d'un point a et soient continues en a pour pouvoir intervertir l'ordre de dérivation. //Si f et ses dérivées partielles sont continues, l'ordre des dérivées partielles n'a pas d'importance (équivalent)

Par exemple, considérons les sensibilités obtenues en appliquant la dérivée partielle $\frac{\partial^{1+2}}{\partial \alpha_1 \partial \alpha_2^2}$ aux termes de l'EDP (3.1). Nous devons alors différentier tous les termes de l'EDP par $\frac{\partial}{\partial \alpha_1}$, puis $\frac{\partial^2}{\partial \alpha_2^2}$. Ici, n_1 et n_2 valent respectivement 1 et 2. Il est donc question d'un système de sensibilités d'ordre 3 à 2 paramètres ($n = n_1 + n_2 = 3$ et $q = 2$). Nous appliquons les dérivées partielles uniquement au terme $\rho u \cdot \nabla u$ dans le but d'alléger le contenu de l'exemple. Les autres termes de sensibilités s'obtiennent de façon similaire. Dans un premier temps, nous appliquons la dérivée $\frac{\partial}{\partial \alpha_1}$:

$$\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha_1} = \rho^{(\alpha_1^{(1)})} u^{(\alpha_1^{(0)})} \cdot \nabla u^{(\alpha_1^{(0)})} + \rho^{(\alpha_1^{(0)})} u^{(\alpha_1^{(1)})} \cdot \nabla u^{(\alpha_1^{(0)})} + \rho^{(\alpha_1^{(0)})} u^{(\alpha_1^{(0)})} \cdot \nabla u^{(\alpha_1^{(1)})}. \quad (3.46)$$

Pour une fonction donnée, l'exposant affecté à l'exposant α_1 indique l'ordre de la dérivée de cette même fonction par rapport au paramètre α_1 . Par exemple, $u^{(\alpha_1^{(1)})}$ et $u^{(\alpha_1^{(0)})}$ représentent respectivement la dérivée $\frac{\partial u}{\partial \alpha_1}$ et la vitesse u elle-même.

Nous écrivons la dérivée décrite par l'équation (3.46) sous forme générale (se référer à l'équation (3.43)) :

$$\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha_1} = \sum_{k_{1,1}+k_{1,2}+k_{1,3}=1} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \rho^{(\alpha_1^{(k_{1,1})})} u^{(\alpha_1^{(k_{1,2})})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})})}. \quad (3.47)$$

Ensuite, nous dérivons cette dernière équation 2 fois par rapport au paramètre α_2 tel qu'in-

diqué :

$$\frac{\partial^2}{\partial \alpha_2^2} \left(\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha_1} \right) = \frac{\partial^2}{\partial \alpha_2^2} \left(\sum_{k_{1,1}+k_{1,2}+k_{1,3}=1} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \rho^{(\alpha_1^{(k_{1,1})})} u^{(\alpha_1^{(k_{1,2})})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})})} \right). \quad (3.48)$$

Selon les règles de dérivation, la formule (3.43) doit être appliquée à chacun des 3 termes résultant de la dérivée première par rapport à α_1 (équation (3.46)). L'équation (3.49) vient illustrer ces propos :

$$\frac{\partial^2}{\partial \alpha_2^2} \left(\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha_1} \right) = \sum_{k_{1,1}+k_{1,2}+k_{1,3}=1} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \cdot \frac{\partial^2}{\partial \alpha_2^2} \left(\rho^{(\alpha_1^{(k_{1,1})})} u^{(\alpha_1^{(k_{1,2})})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})})} \right). \quad (3.49)$$

Nous pouvons sortir le terme $\frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!}$ de la dérivée seconde, puisqu'il s'agit d'un coefficient constant. Nous obtenons alors :

$$\begin{aligned} \frac{\partial^2}{\partial \alpha_2^2} \left(\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha_1} \right) &= \sum_{k_{1,1}+k_{1,2}+k_{1,3}=1} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times \\ &\left[\rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(2)})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(0)})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(0)})} + \right. \\ &2\rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(1)})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(1)})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(0)})} + \\ &2\rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(1)})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(0)})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(1)})} + \\ &\rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(0)})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(2)})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(0)})} + \\ &2\rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(0)})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(1)})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(1)})} + \\ &\left. \rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(0)})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(0)})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(2)})} \right], \end{aligned} \quad (3.50)$$

où l'exposant affecté à l'exposant α_i donne l'ordre de la dérivée d'une fonction donnée par rapport au paramètre α_i ($i = 1$ à 2).

L'expression générale pour l'équation (3.50) est :

$$\begin{aligned} \frac{\partial^2}{\partial \alpha_2^2} \left(\frac{\partial(\rho u \cdot \nabla u)}{\partial \alpha_1} \right) = & \\ \sum_{k_{1,1}+k_{1,2}+k_{1,3}=1} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times & \\ \left[\sum_{k_{2,1}+k_{2,2}+k_{2,3}=2} \frac{2!}{k_{2,1}!k_{2,2}!k_{2,3}!} \cdot \rho^{(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})})} u^{(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})})} \cdot \nabla u^{(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})})} \right]. & \end{aligned} \quad (3.51)$$

Notons que les exposants $(\alpha_i^{(k_{i,j})})$ donnent l'ordre des dérivées partielles par rapport à un paramètre α_i pour les fonctions f_j . Ici, $j = 1$ à 3 et $i = 1$ à 2. Les fonctions ρ , u et ∇u sont respectivement associées aux fonctions f_1 , f_2 et f_3 . Pour exprimer les dérivées partielles mixtes d'une fonction donnée f_j , nous les écrivons sous la notation $(\alpha_1^{(k_{1,j})} \alpha_2^{(k_{2,j})} \dots \alpha_q^{(k_{q,j})})$. Ainsi $f_j^{(\alpha_1^{(k_{1,j})} \alpha_2^{(k_{2,j})} \dots \alpha_q^{(k_{q,j})})}$ représente la dérivée $\frac{\partial^{k_{1,j}+k_{2,j}+\dots+k_{q,j}}}{\partial \alpha_1^{k_{1,j}} \partial \alpha_2^{k_{2,j}} \dots \partial \alpha_q^{k_{q,j}}}(f_j)$.

À partir de ce principe, nous avons défini la formulation générale des équations de sensibilités avec q paramètres. Dans le cas du système où l'on souhaite évaluer les sensibilités obtenues en appliquant $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ aux EDP, nous utilisons la formule générale suivante sur chacun des produits de m fonctions des EDP :

$$\begin{aligned} \frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}} \left(\prod_{j=1}^m f_j \right) = & \\ \sum_{k_{1,1}+k_{1,2}+\dots+k_{1,m}=n_1} \frac{n_1!}{k_{1,1}!k_{1,2}! \dots k_{1,m}!} \times & \\ \sum_{k_{2,1}+k_{2,2}+\dots+k_{2,m}=n_2} \frac{n_2!}{k_{2,1}!k_{2,2}! \dots k_{2,m}!} \times & \\ \dots \times & \\ \sum_{k_{q,1}+k_{q,2}+\dots+k_{q,m}=n_q} \frac{n_q!}{k_{q,1}!k_{q,2}! \dots k_{q,m}!} \prod_{j=1}^m f_j^{(\alpha_1^{(k_{1,j})} \alpha_2^{(k_{2,j})} \dots \alpha_q^{(k_{q,j})})}, & \end{aligned} \quad (3.52)$$

où q est le nombre total de paramètres α_i considérés dans le système, $i = 1$ à q , m est le nombre de fonctions dans le produit considéré et n_i est l'ordre de la dérivée partielle par rapport au paramètre α_i du produit $\prod_{j=1}^m f_j$. L'exposant $(\alpha_1^{(k_{1,j})} \alpha_2^{(k_{2,j})} \dots \alpha_q^{(k_{q,j})})$ identifie la dérivée partielle mixte pour la fonction f_j d'un terme donné. La valeur $k_{i,j}$ apparaissant dans cette expression donne l'ordre de la sensibilité de la fonction f_j pour le paramètre α_i . Par exemple, $f_j^{(\alpha_1^{(1)} \alpha_2^{(2)} \alpha_3^{(0)})}$ représente la dérivée $\frac{\partial^3}{\partial \alpha_1 \partial \alpha_2^2}$ de la fonction f_j . Nous avons donc une

sensibilité d'ordre 3 pour f_j . On note que le nombre d'opérateurs de sommation $\sum_{k_{i,1}+k_{i,2}+\dots+k_{i,m}}$ dans l'expression générale (3.52) est égal au nombre q de paramètres considérés pour les sensibilités (i.e. : $i = 1$ à q). Chaque sommation s'effectue sur un ensemble de sous-ensembles de $k_{i,j}$ propre à chaque paramètre α_i , puisque ces sous-ensembles dépendent de la valeur de l'ordre n_i de la dérivée partielle par rapport au paramètre α_i . En effet, nous devons avoir $\sum_{j=1}^m k_{i,j} = n_i$ pour tous les sous-ensembles associés au i -ième opérateur de sommation ($i = 1$ à q), où i indique le numéro du paramètre α_i concerné.

Ces sous-ensembles, sur lesquels s'effectue chacune des sommations, sont définis de manière similaire à ceux de la sous-section 3.2.3. Par exemple, $\sum_{k_{2,1}+k_{2,2}+k_{2,3}=2}$ se traduit par le problème combinatoire suivant : trouver les combinaisons à répétition formées de 2 éléments tirés dans l'ensemble $E_2 = \{f_1^{(\alpha_2)}, f_2^{(\alpha_2)}, f_3^{(\alpha_2)}\}$ qui contient 3 éléments. À partir de la formule (A.6), nous savons qu'il existe 6 combinaisons de ce type, soient :

$$\begin{array}{lll} \{f_1^{(\alpha_2)}, f_1^{(\alpha_2)}\}, & \{f_2^{(\alpha_2)}, f_2^{(\alpha_2)}\}, & \{f_3^{(\alpha_2)}, f_3^{(\alpha_2)}\}, \\ \{f_1^{(\alpha_2)}, f_2^{(\alpha_2)}\}, & \{f_1^{(\alpha_2)}, f_3^{(\alpha_2)}\}, & \{f_2^{(\alpha_2)}, f_3^{(\alpha_2)}\}. \end{array}$$

Dans la formule (3.52), la totalité des opérateurs de sommations sont multipliés entre eux. La multiplication de ces opérateurs de sommations fait référence au principe multiplicatif de l'analyse combinatoire (voir annexe A). Nous avons q ensembles $E_i = \{f_1^{(\alpha_i)}, f_2^{(\alpha_i)}, \dots, f_m^{(\alpha_i)}\}$, avec $i = 1$ à q , soit autant d'ensembles qu'il y a de paramètres d'intérêt. Il faut choisir n_i éléments parmi les m éléments de chacun de ces ensembles E_i . Chacun des éléments peut être sélectionné plusieurs fois. L'ordre dans lequel nous choisissons ces éléments n'a aucune importance. Il s'agit donc de combinaisons avec répétition. Le nombre de répétitions d'un élément $f_j^{(\alpha_i)}$, pour $i = 1$ à q et $j = 1$ à m , nous indique l'ordre de la dérivée partielle par rapport au paramètre α_i pour la fonction f_j . Le nombre de combinaisons possibles pour chaque ensemble E_i est :

$$C_{n_i}^{m_i+m-1} = \frac{(n_i + m - 1)!}{n_i!(m - 1)!}. \quad (3.53)$$

Une fois que nous avons déterminé les façons de choisir n_i éléments pour chacun des ensembles E_i , nous examinons le nombre total s de possibilités pour combiner ces diverses façons de choisir les n_i éléments parmi E_i (principe multiplicatif). Par exemple, si nous avons 3 ensembles, E_1 , E_2 et E_3 , les combinaisons de n_1 , n_2 et n_3 éléments respectivement obtenues pour le premier ensemble, le deuxième ensemble et le troisième ensemble sont combinées entre

elles. Il s'agit alors de multiplier les possibilités pour chaque cas. Autrement dit, le nombre total s de combinaisons (donc de termes) pour l'équation (3.52) est équivalent à :

$$s = \prod_{i=1}^q C_{n_i}^{m_i+m-1}. \quad (3.54)$$

Les sous-ensembles de $k_{i,j}$ associés à chaque opérateur de sommation sont trouvés de façon indépendante. La multiplication des sommations se traduit donc par une sommation sur la totalité des combinaisons de $k_{i,j}$ qui existent pour $i = 1$ à q et $j = 1$ à m telles que $\sum_{j=1}^m k_{i,j} = n_i$. En se basant sur l'écriture de l'équation (B.6), nous pouvons réécrire l'équation (3.52) de la façon suivante :

$$\begin{aligned} \frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}} \left(\prod_{j=1}^m f_j \right) = & \sum_{\substack{k_{1,1}+k_{1,2}+\dots+k_{1,m}=n_1 \\ k_{2,1}+k_{2,2}+\dots+k_{2,m}=n_2 \\ \dots \\ k_{q,1}+k_{q,2}+\dots+k_{q,m}=n_q}} \frac{n_1!}{k_{1,1}!k_{1,2}!\dots k_{1,m}!} \times \frac{n_2!}{k_{2,1}!k_{2,2}!\dots k_{2,m}!} \times \dots \\ & \times \frac{n_q!}{k_{q,1}!k_{q,2}!\dots k_{q,m}!} \prod_{j=1}^m f_j \left(\alpha_1^{(k_{1,j})} \alpha_2^{(k_{2,j})} \dots \alpha_q^{(k_{q,j})} \right). \end{aligned} \quad (3.55)$$

Illustrons le fonctionnement de l'équation (3.55) au moyen d'un exemple. Nous cherchons à obtenir le développement de $\frac{\partial^3}{\partial \alpha_1^2 \partial \alpha_2} (f_1 f_2 f_3)$ ($m = 3$ et $q = 2$). Pour ce faire, nous devons trouver l'ensemble des combinaisons de $k_{i,j}$, pour $i = 1$ à 2 et $j = 1$ à 3 , sur lesquelles ont lieu les sommations de l'équation (3.52). Dans un premier temps, nous devons trouver les combinaisons de n_i éléments choisis parmi chacun des ensembles $E_i = \{f_1^{(\alpha_i)}, f_2^{(\alpha_i)}, f_3^{(\alpha_i)}\}$ formés de 3 éléments chacun ($m = 3$). Pour notre exemple, nous avons des sensibilités à 2 paramètres ($q = 2$). Cela implique que $i = 1$ à 2 . Nous avons donc 2 ensembles E_i soient : $E_1 = \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\}$ et $E_2 = \{f_1^{(\alpha_2)}, f_2^{(\alpha_2)}, f_3^{(\alpha_2)}\}$. De plus, comme la dérivée partielle par rapport à α_1 est d'ordre 2 et celle par rapport à α_2 est d'ordre 1, n_1 et n_2 valent respectivement 2 et 1. Commençons par chercher toutes les combinaisons de $n_1 = 2$ éléments que l'on peut obtenir à partir des éléments de l'ensemble $E_1 = \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\}$ lorsque ceux-ci peuvent être répétés :

$$\begin{aligned} & \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}\}, \quad \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}\}, \quad \{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}\} \\ & \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}\}, \quad \{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}\}, \quad \{f_3^{(\alpha_1)}, f_3^{(\alpha_1)}\}. \end{aligned}$$

Nous avons un total de 6 combinaisons dont le nombre peut être validé par la formule (3.53) :

$C_2^{2+3-1} = 6$. Ensuite, nous déterminons les combinaisons formées de $n_2 = 1$ élément choisi parmi les éléments de l'ensemble $E_2 = \{f_1^{(\alpha_2)}, f_2^{(\alpha_2)}, f_3^{(\alpha_2)}\}$. On dénombre 3 combinaisons possibles (voir l'équation (3.53)) :

$$\{f_1^{(\alpha_2)}\}, \quad \{f_2^{(\alpha_2)}\}, \quad \{f_3^{(\alpha_2)}\}.$$

Par la suite, nous évaluons toutes les possibilités de combinaisons entre les combinaisons respectivement obtenues à partir des ensembles E_1 et E_2 . Nous illustrons ces possibilités par le biais d'un arbre à la figure 3.1. Les combinaisons de la première colonne de l'arbre correspondent aux sous-ensembles de 2 éléments choisis parmi les éléments de l'ensemble E_1 , tandis que la deuxième colonne correspond aux combinaisons formées d'un seul élément provenant de l'ensemble E_2 . On note que chaque possibilité de la première colonne peut être combinée à une des trois combinaisons de 1 élément obtenues à partir de l'ensemble E_2 .

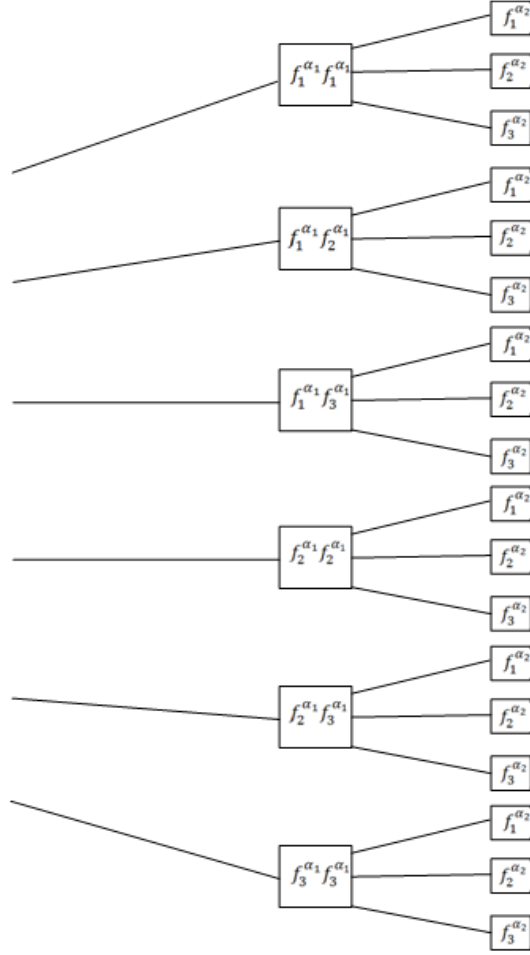


Figure 3.1 Arbre des possibilités de combinaisons entre les combinaisons obtenues pour les ensembles indépendants E_1 et E_2 .

Par conséquent, la figure 3.1 nous permet d'observer un total de $C_2^{2+3-1} \cdot C_1^{1+3-1} = 6 \cdot 3 = 18$ combinaisons (se référer à l'équation (3.54)) :

$$\begin{aligned}
 & \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_1^{(\alpha_2)}\}, & \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_2^{(\alpha_2)}\}, & \{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_3^{(\alpha_2)}\} \\
 & \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_1^{(\alpha_2)}\}, & \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_2)}\}, & \{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_2)}\} \\
 & \{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_1^{(\alpha_2)}\}, & \{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_2^{(\alpha_2)}\}, & \{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_2)}\} \\
 & \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_1^{(\alpha_2)}\}, & \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_2)}\}, & \{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_2)}\} \\
 & \{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}, f_1^{(\alpha_2)}\}, & \{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}, f_2^{(\alpha_2)}\}, & \{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_2)}\} \\
 & \{f_3^{(\alpha_1)}, f_3^{(\alpha_1)}, f_1^{(\alpha_2)}\}, & \{f_3^{(\alpha_1)}, f_3^{(\alpha_1)}, f_2^{(\alpha_2)}\}, & \{f_3^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_2)}\}.
 \end{aligned}$$

Nous associons ces combinaisons aux sous-ensembles de $k_{i,j}$ appropriés. Sachant que le nombre de répétitions d'un élément $f_j^{(\alpha_i)}$ donne la valeur de l'exposant $k_{i,j}$, nous obtenons donc les sous-ensembles suivants :

$$\begin{aligned}
\{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_1^{(\alpha_2)}\} &\rightarrow k_{1,1} = 2; k_{1,2} = 0; k_{1,3} = 0; \\
&k_{2,1} = 1; k_{2,2} = 0; k_{2,3} = 0; \\
\{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_2^{(\alpha_2)}\} &\rightarrow k_{1,1} = 2; k_{1,2} = 0; k_{1,3} = 0; \\
&k_{2,1} = 0; k_{2,2} = 1; k_{2,3} = 0; \\
\{f_1^{(\alpha_1)}, f_1^{(\alpha_1)}, f_3^{(\alpha_2)}\} &\rightarrow k_{1,1} = 2; k_{1,2} = 0; k_{1,3} = 0; \\
&k_{2,1} = 0; k_{2,2} = 0; k_{2,3} = 1; \\
\{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_1^{(\alpha_2)}\} &\rightarrow k_{1,1} = 1; k_{1,2} = 1; k_{1,3} = 0; \\
&k_{2,1} = 1; k_{2,2} = 0; k_{2,3} = 0; \\
\{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_2)}\} &\rightarrow k_{1,1} = 1; k_{1,2} = 1; k_{1,3} = 0; \\
&k_{2,1} = 0; k_{2,2} = 1; k_{2,3} = 0; \\
\{f_1^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_2)}\} &\rightarrow k_{1,1} = 1; k_{1,2} = 1; k_{1,3} = 0; \\
&k_{2,1} = 0; k_{2,2} = 0; k_{2,3} = 1; \\
\{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_1^{(\alpha_2)}\} &\rightarrow k_{1,1} = 1; k_{1,2} = 0; k_{1,3} = 1; \\
&k_{2,1} = 1; k_{2,2} = 0; k_{2,3} = 0; \\
\{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_2^{(\alpha_2)}\} &\rightarrow k_{1,1} = 1; k_{1,2} = 0; k_{1,3} = 1; \\
&k_{2,1} = 0; k_{2,2} = 1; k_{2,3} = 0; \\
\{f_1^{(\alpha_1)}, f_3^{(\alpha_1)}, f_3^{(\alpha_2)}\} &\rightarrow k_{1,1} = 1; k_{1,2} = 0; k_{1,3} = 1; \\
&k_{2,1} = 0; k_{2,2} = 0; k_{2,3} = 1; \\
\{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_1^{(\alpha_2)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 2; k_{1,3} = 0; \\
&k_{2,1} = 1; k_{2,2} = 0; k_{2,3} = 0; \\
\{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_2^{(\alpha_2)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 2; k_{1,3} = 0; \\
&k_{2,1} = 0; k_{2,2} = 1; k_{2,3} = 0; \\
\{f_2^{(\alpha_1)}, f_2^{(\alpha_1)}, f_3^{(\alpha_2)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 2; k_{1,3} = 0; \\
&k_{2,1} = 0; k_{2,2} = 0; k_{2,3} = 1; \\
\{f_2^{(\alpha_1)}, f_3^{(\alpha_1)}, f_1^{(\alpha_2)}\} &\rightarrow k_{1,1} = 0; k_{1,2} = 1; k_{1,3} = 1;
\end{aligned}$$

trouvées grâce à l'arbre de la figure 3.1 :

$$\begin{aligned}
& \frac{\partial^3}{\partial \alpha_1^2 \partial \alpha_2} \left(\prod_{j=1}^3 f_j \right) = \\
& \frac{2!}{2!0!0!} \frac{1!}{1!0!0!} f_1^{(\alpha_1^{(2)} \alpha_2^{(1)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(0)})} + \frac{2!}{2!0!0!} \frac{1!}{0!1!0!} f_1^{(\alpha_1^{(2)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(1)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(0)})} + \\
& \frac{2!}{2!0!0!} \frac{1!}{0!0!1!} f_1^{(\alpha_1^{(2)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(1)})} + \frac{2!}{1!1!0!} \frac{1!}{1!0!0!} f_1^{(\alpha_1^{(1)} \alpha_2^{(1)})} f_2^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(0)})} + \\
& \frac{2!}{1!1!0!} \frac{1!}{0!1!0!} f_1^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(1)} \alpha_2^{(1)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(0)})} + \frac{2!}{1!1!0!} \frac{1!}{0!0!1!} f_1^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(1)})} + \\
& \frac{2!}{1!0!1!} \frac{1!}{1!0!0!} f_1^{(\alpha_1^{(1)} \alpha_2^{(1)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(1)} \alpha_2^{(0)})} + \frac{2!}{1!0!1!} \frac{1!}{0!1!0!} f_1^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(1)})} f_3^{(\alpha_1^{(1)} \alpha_2^{(0)})} + \\
& \frac{2!}{1!0!1!} \frac{1!}{0!0!1!} f_1^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(1)} \alpha_2^{(1)})} + \frac{2!}{0!2!0!} \frac{1!}{1!0!0!} f_1^{(\alpha_1^{(0)} \alpha_2^{(1)})} f_2^{(\alpha_1^{(2)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(0)})} + \\
& \frac{2!}{0!2!0!} \frac{1!}{0!1!0!} f_1^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(2)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(1)})} + \frac{2!}{0!2!0!} \frac{1!}{0!0!1!} f_1^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(2)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(0)} \alpha_2^{(1)})} + \\
& \frac{2!}{0!1!1!} \frac{1!}{1!0!0!} f_1^{(\alpha_1^{(0)} \alpha_2^{(1)})} f_2^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(1)} \alpha_2^{(0)})} + \frac{2!}{0!1!1!} \frac{1!}{0!1!0!} f_1^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(1)} \alpha_2^{(1)})} f_3^{(\alpha_1^{(1)} \alpha_2^{(0)})} + \\
& \frac{2!}{0!1!1!} \frac{1!}{0!0!1!} f_1^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(1)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(1)} \alpha_2^{(1)})} + \frac{2!}{0!0!2!} \frac{1!}{1!0!0!} f_1^{(\alpha_1^{(0)} \alpha_2^{(1)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(2)} \alpha_2^{(0)})} + \\
& \frac{2!}{0!0!2!} \frac{1!}{0!1!0!} f_1^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(1)})} f_3^{(\alpha_1^{(2)} \alpha_2^{(0)})} + \frac{2!}{0!0!2!} \frac{1!}{0!0!1!} f_1^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_2^{(\alpha_1^{(0)} \alpha_2^{(0)})} f_3^{(\alpha_1^{(2)} \alpha_2^{(1)})}.
\end{aligned} \tag{3.56}$$

L'expression générale (3.52) développe la dérivée par rapport à q paramètres d'un seul produit de fonctions apparaissant dans l'EDP. Comme dans le cas des sensibilités avec 1 paramètre, il est important d'utiliser l'équation (3.52) sur tous les produits de fonctions de l'EDP lorsqu'on souhaite obtenir l'équation des sensibilités d'ordre $n = n_1 + n_2 + \dots + n_q$ par rapport aux q paramètres. Prenons l'EDP (3.1) formé de 4 produits d'une ou plusieurs fonctions :

$$(\rho u \cdot \nabla u)^{(0)} = (\nabla \cdot (\mu \nabla u))^{(0)} - (\beta T)^{(0)} + (f_s)^{(0)}. \tag{3.57}$$

L'exposant (0) de l'équation (3.57) fait référence à la dérivée d'ordre 0. Par exemple, $(\rho u \cdot \nabla u)^{(0)}$ correspond au produit $\rho u \cdot \nabla u$ lui-même.

La dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}$ de cette équation est donc :

$$\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}(\rho\mathbf{u} \cdot \nabla\mathbf{u}) = \frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}(\nabla \cdot (\mu\nabla\mathbf{u})) - \frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}(\beta T) + \frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}(f_s). \quad (3.58)$$

En appliquant la formule générale (3.55) à chacun des termes de l'EDP, nous obtenons donc l'équation de sensibilités de (3.1) d'ordre n avec q paramètres : $\alpha_1, \alpha_2, \dots, \alpha_q$.

3.3 Application de la formulation générale de l'équation de sensibilités aux équations de Navier-Stokes

Ce projet vise ultimement l'application de la MESC d'ordre n sur des problèmes d'écoulements. Nous prenons donc les EDP de Navier-Stokes incompressible et stationnaire avec l'approximation de Boussinesq et montrons comment appliquer la formulation générale de l'équation de sensibilités d'ordre n . Le comportement d'un écoulement régi par les équations de Navier-Stokes est défini comme suit :

$$\nabla \cdot \mathbf{u} = 0, \quad (3.59)$$

$$\rho\mathbf{u} \cdot \nabla\mathbf{u} = -\nabla P + \nabla \cdot [\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] - \rho\mathbf{g}\beta(\mathbf{T} - \mathbf{T}_0) + \mathbf{f}, \quad (3.60)$$

$$\rho c_p \mathbf{u} \cdot \nabla T = \nabla \cdot (\lambda \nabla T) + q_s, \quad (3.61)$$

où le vecteur vitesse \mathbf{u} , la température T et la pression P sont des variables dépendantes, alors que ρ, μ, β, c_p et λ sont les propriétés physiques du fluide, \mathbf{g} est le vecteur gravité, T_0 est la température de référence, \mathbf{f} est une force volumique et q_s est une source de chaleur. L'équation (3.59) correspond à l'équation de continuité, l'équation (3.60) représente l'équation de mouvement et l'équation (3.61) est l'équation d'énergie.

Nous devons définir certains éléments lorsqu'on souhaite générer les équations de sensibilités à partir de la formule générale (3.55). Dans un premier temps, nous déterminons le nombre q de paramètres du système. Puis, nous fixons les valeurs n_1, n_2, \dots, n_q en fonction du système de sensibilité étudié. Ensuite, nous évaluons le nombre exact d'équations d'état du système, ainsi que le nombre de produits de fonctions qui se trouvent dans chacune de ces équations. Enfin, nous devons déterminer le nombre de fonctions de chacun des produits contenus dans chacune des équations du système. Cela nous indique alors les valeurs successives de m dans

l'équation (3.55). Avec ses informations, nous pouvons utiliser l'équation (3.55) pour générer les équations de sensibilités. Nous appliquons celle-ci aux divers produits de chacune des équations, puis nous assemblons les dérivées explicites ainsi obtenues pour chaque produit.

À titre d'exemple, pour écrire les équations de sensibilités d'ordre 2 obtenues en appliquant $\frac{\partial^2}{\partial\alpha_1\partial\alpha_3}$ aux équations de Navier-Stokes, nous utilisons la formule générale (3.55). Cela implique que $q = 3$ et $n_1 = 1$, $n_2 = 0$ et $n_3 = 1$. Notez que tous les paramètres considérés sont des paramètres de valeur. Pour le problème des équations de Navier-Stokes 2D, on compte 4 équations. En effet, l'équation de mouvement (3.60) est une EDP vectorielle qui comporte 2 composantes, soient en x et en y :

$$\begin{aligned} \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = & -\frac{\partial P}{\partial x} + 2\mu \frac{\partial^2 u}{\partial x^2} + \mu \frac{\partial^2 v}{\partial x \partial y} + \mu \frac{\partial^2 u}{\partial y^2} + 2 \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial x} + \\ & \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial y} \frac{\partial u}{\partial y} - \rho g_x \beta (T - T_0) + f_x, \end{aligned} \quad (3.62)$$

$$\begin{aligned} \rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} = & -\frac{\partial P}{\partial y} + 2\mu \frac{\partial^2 v}{\partial y^2} + \mu \frac{\partial^2 u}{\partial x \partial y} + \mu \frac{\partial^2 v}{\partial x^2} + 2 \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial y} + \\ & \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial x} - \rho g_y \beta (T - T_0) + f_y, \end{aligned} \quad (3.63)$$

où u et v correspondent respectivement aux composantes en x et en y du vecteur vitesse \mathbf{u} , g_x et g_y sont les composantes en x et en y du vecteur gravité \mathbf{g} et f_x et f_y représentent les composantes en x et en y du champ de force \mathbf{f} .

Pour obtenir les équations de sensibilités d'ordre 2 du système $\frac{\partial^2}{\partial\alpha_1\partial\alpha_3}$, nous devons appliquer la dérivée $\frac{\partial^2}{\partial\alpha_1\partial\alpha_3}$ aux équations (3.59), (3.62), (3.63) et (3.61). Pour l'équation (3.59), nous obtenons :

$$\frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial u}{\partial x} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial v}{\partial y} \right) = 0.$$

Nous appliquons alors la formulation générale (3.55) à chacun des 2 produits contenus dans cette équation, soient :

- $\frac{\partial u}{\partial x}$ qui est formé d'une seule fonction,
- $\frac{\partial v}{\partial y}$ qui est formé d'une seule fonction.

Pour $\frac{\partial u}{\partial x}$, où $m = 1$, nous obtenons :

$$\frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} \left(\frac{\partial u}{\partial x} \right) = \sum_{\substack{k_{1,1}=1 \\ k_{2,1}=0 \\ k_{3,1}=1}} \frac{1!}{k_{1,1}!} \cdot \frac{0!}{k_{2,1}!} \cdot \frac{1!}{k_{3,1}!} \left(\frac{\partial u}{\partial x} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)}.$$

La forme de la dérivée est similaire pour $\frac{\partial v}{\partial y}$, car les composantes m , q , n_1 , n_2 et n_3 sont identiques :

$$\frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} \left(\frac{\partial v}{\partial y} \right) = \sum_{\substack{k_{1,1}=1 \\ k_{2,1}=0 \\ k_{3,1}=1}} \frac{1!}{k_{1,1}!} \cdot \frac{0!}{k_{2,1}!} \cdot \frac{1!}{k_{3,1}!} \left(\frac{\partial v}{\partial y} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)}.$$

On remarque que même si l'ordre n_2 par rapport au paramètre α_2 vaut 0, nous notons explicitement la présence du paramètre α_2 dans l'équation, car nous considérons les sensibilités avec $q = 3$ paramètres. Ainsi, pour les divers systèmes d'ordre 2 étudiés, ce ne sont pas tous les paramètres qui apparaissent dans les dérivées croisées tel qu'on peut l'observer pour le cas particulier $\frac{\partial^2}{\partial \alpha_1 \partial \alpha_3}$.

Si nous assemblons les formes générales obtenues pour les dérivées des produits $\frac{\partial u}{\partial x}$ et $\frac{\partial v}{\partial y}$, nous obtenons la notation suivante pour la sensibilité de l'équation de continuité :

$$\begin{aligned} \frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} (3.59) &= \sum_{\substack{k_{1,1}=1 \\ k_{2,1}=0 \\ k_{3,1}=1}} \frac{1!}{k_{1,1}!} \cdot \frac{0!}{k_{2,1}!} \cdot \frac{1!}{k_{3,1}!} \left(\frac{\partial u}{\partial x} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} + \\ &\quad \sum_{\substack{k_{1,1}=1 \\ k_{2,1}=0 \\ k_{3,1}=1}} \frac{1!}{k_{1,1}!} \cdot \frac{0!}{k_{2,1}!} \cdot \frac{1!}{k_{3,1}!} \left(\frac{\partial v}{\partial y} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)}. \end{aligned}$$

Appliquons la dérivée $\frac{\partial^2}{\partial\alpha_1\partial\alpha_3}$ à l'équation de mouvement selon x (3.62) :

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\rho u \frac{\partial u}{\partial x} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\rho v \frac{\partial u}{\partial y} \right) = & - \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial P}{\partial x} \right) + 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 u}{\partial x^2} \right) + \\ & \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 v}{\partial x \partial y} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 u}{\partial y^2} \right) + \\ & 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial x} \frac{\partial u}{\partial x} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} \right) + \\ & \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial y} \frac{\partial u}{\partial y} \right) - \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (\rho g_x \beta (T - T_0)) + \\ & \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (f_x). \end{aligned} \tag{3.64}$$

Faisons de même pour l'équation de mouvement selon y (3.63) :

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\rho u \frac{\partial v}{\partial x} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\rho v \frac{\partial v}{\partial y} \right) = & - \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial P}{\partial y} \right) + 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 v}{\partial y^2} \right) + \\ & \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 u}{\partial x \partial y} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 v}{\partial x^2} \right) + \\ & \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial x} \frac{\partial u}{\partial y} \right) + \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial x} \frac{\partial v}{\partial x} \right) + \\ & 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial y} \frac{\partial v}{\partial y} \right) - \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (\rho g_y \beta (T - T_0)) + \\ & \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (f_y). \end{aligned} \tag{3.65}$$

Nous n'écrivons pas la formulation générale des dérivées des termes de ces équations de sensibilités dans ce chapitre, car ceux-ci génèrent beaucoup de termes. Toutefois, le principe est le même que pour l'équation de continuité. Il suffit de connaître le nombre m de fonctions pour chacun des produits de l'équation. Nous utilisons cette valeur pour initialiser la variable m de l'équation (3.55). À partir de cela, on applique l'équation (3.52) ou l'équation (3.55) à chacun des produits de fonctions. Le lecteur intéressé peut retrouver le développement complet des termes de sensibilités des équations (3.64) et (3.65) en se référant à l'annexe C.

Pour l'équation de sensibilités du système $\frac{\partial^2}{\partial\alpha_1\alpha_3}$ de l'équation (3.61), nous obtenons :

$$\frac{\partial^2}{\partial\alpha_1\alpha_3}(\rho c_p \mathbf{u} \cdot \nabla T) = \frac{\partial^2}{\partial\alpha_1\alpha_3}(\nabla \cdot (\lambda \nabla T)) + \frac{\partial^2}{\partial\alpha_1\alpha_3}(q_s).$$

L'équation d'énergie (3.61) comporte 3 produits de fonctions, soient :

- $\rho c_p \mathbf{u} \cdot \nabla T$ qui est formé de 4 fonctions : ρ , c_p , \mathbf{u} et ∇T ,
- $\nabla \cdot (\lambda \nabla T)$ qui est formé de 2 fonctions : λ et ∇T ,
- q_s qui est formé d'une seule fonction.

Nous appliquons donc la formule générale pour développer la dérivée de chacun de ces 3 produits. Commençons par $\rho c_p \mathbf{u} \cdot \nabla T$. La formulation générale de sa dérivée est :

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\alpha_3}(\rho c_p \mathbf{u} \cdot \nabla T) = & \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}+k_{1,4}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}+k_{2,4}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}+k_{3,4}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!k_{1,4}!} \cdot \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!k_{2,4}!} \cdot \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!k_{3,4}!} \\ & \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)_{c_p} \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)_{\mathbf{u}} \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right) \cdot \nabla T \left(\alpha_1^{(k_{1,4})} \alpha_2^{(k_{2,4})} \alpha_3^{(k_{3,4})} \right). \end{aligned}$$

Pour le produit $\nabla \cdot (\lambda \nabla T)$, sa dérivée $\frac{\partial^2}{\partial\alpha_1\alpha_3}$ s'écrit sous la forme générale suivante :

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\alpha_3}(\nabla \cdot (\lambda \nabla T)) = & \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \cdot \frac{0!}{k_{2,1}!k_{2,2}!} \cdot \frac{1!}{k_{3,1}!k_{3,2}!} \\ & \nabla \cdot \left(\lambda \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) \nabla T \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \right). \end{aligned}$$

Enfin, la dérivée du terme q_s est similaire à celle des dérivées des termes de l'équation de continuité, puisque $m = 1$:

$$\frac{\partial^2}{\partial\alpha_1\alpha_3}(q_s) = \sum_{\substack{k_{1,1}=1 \\ k_{2,1}=0 \\ k_{3,1}=1}} \frac{1!}{k_{1,1}!} \cdot \frac{0!}{k_{2,1}!} \cdot \frac{1!}{k_{3,1}!} q_s \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right).$$

Nous obtenons la forme générale de l'équation de sensibilités pour l'équation d'énergie en

assemblant les formules générales des dérivées des produits de fonctions qui la composent :

$$\begin{aligned}
\frac{\partial^2}{\partial \alpha_1 \alpha_3} (3.61) = & \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}+k_{1,4}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}+k_{2,4}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}+k_{3,4}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!k_{1,4}!} \cdot \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!k_{2,4}!} \cdot \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!k_{3,4}!} \\
& \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)_{C_p} \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)_{\mathbf{u}} \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right) \cdot \nabla T \left(\alpha_1^{(k_{1,4})} \alpha_2^{(k_{2,4})} \alpha_3^{(k_{3,4})} \right) = \\
& \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \cdot \frac{0!}{k_{2,1}!k_{2,2}!} \cdot \frac{1!}{k_{3,1}!k_{3,2}!} \nabla \cdot \left(\lambda \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) \nabla T \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \right) + \\
& \sum_{\substack{k_{1,1}=1 \\ k_{2,1}=0 \\ k_{3,1}=1}} \frac{1!}{k_{1,1}!} \cdot \frac{0!}{k_{2,1}!} \cdot \frac{1!}{k_{3,1}!} q_s \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right).
\end{aligned}$$

Notons que le principe utilisé pour générer automatiquement les équations de sensibilités est le même pour toutes les EDP. Nous devons d'abord déterminer les éléments suivants :

- Nombre de paramètres (q) ;
- Ordre des sensibilités (n_1, n_2, \dots, n_q) ;
- Nombre d'équations différentielles du système ;
- Nombre de produits dans chaque équation ;
- Nombre de fonction dans chaque produit (m).

En connaissant ces éléments, il est alors possible d'appliquer la formule générale (3.55) pour générer les équations de sensibilités. En effet, ces données permettent de trouver les composantes des équations de sensibilités d'ordre n avec q paramètres, soient :

- Nombre de termes des équations des sensibilités d'ordre n où $n = n_1 + n_2 + \dots + n_q$;
- L'ordre des dérivées partielles ($k_{i,j}$) de chaque fonction f_j des produits ;
- La valeur des coefficients de chacun des termes.

Nous présentons au chapitre 4 l'algorithme qui permet de générer automatiquement ces composantes.

CHAPITRE 4 ALGORITHME POUR LA GÉNÉRATION ET LA RÉSOLUTION DES SENSIBILITÉS D'ORDRE ARBITRAIRE

Dans ce chapitre, nous détaillons l'implémentation et le fonctionnement de l'algorithme qui génère et résout de manière générique les équations de sensibilités d'ordre arbitraire n . Notre approche est dite générique, car elle permet d'écrire les équations de sensibilités d'ordre arbitrairement élevé n pour un nombre quelconque q de paramètre(s) sans avoir à modifier le code de résolution. Cette approche dérive automatiquement l'EDP du système étudié jusqu'à l'ordre n , où $n \in \mathbb{N}$, et résout numériquement les équations de sensibilités qui en découlent. Nous testons cette méthode générique sur un problème non-linéaire unidimensionnel. Cela nous permet alors de valider et d'analyser les résultats de cette méthode. La validation et la vérification de la méthode implémentée sont abordées plus en détail au chapitre 5.

Tout d'abord, nous présentons le problème unidimensionnel sur lequel nous appliquons notre algorithme. Ensuite, nous indiquons les données d'entrées que l'utilisateur doit fournir à notre algorithme pour générer et résoudre les équations de sensibilités d'ordre n à q paramètre(s). Puis, nous décrivons explicitement la procédure pour générer les équations de sensibilités d'ordre n et en particulier, pour déterminer la nature et le nombre des termes impliqués. Premièrement, nous expliquons comment notre algorithme trouve les systèmes de sensibilités à évaluer. Deuxièmement, nous détaillons l'algorithme implémenté pour la génération des composantes des équations de sensibilités d'ordre n . Enfin, nous décrivons le solveur utilisé pour résoudre les équations de sensibilités générées.

4.1 Problème 1D

Nous utilisons un problème test régi par un système d'équations 1D présentant plusieurs caractéristiques des équations de Navier-Stokes 2D afin de vérifier l'implémentation de notre algorithme :

$$u \cdot \nabla u = \nabla \cdot (\nu \nabla u) - \beta T + f_s, \quad (4.1)$$

$$u \cdot \nabla T = \nabla \cdot (D \nabla T) + q_s, \quad (4.2)$$

où u et T sont les variables dépendantes inconnues, tandis que ν , β et D sont les propriétés physiques et f_s et q_s sont des termes sources. Ces derniers s'avèrent très utiles pour la vérification du code par la MSM décrite au chapitre 5. L'équation (4.1) désigne l'équation de u et l'équation (4.2) représente l'équation de T .

Comme nous traitons uniquement les paramètres de valeur, nous exprimons les dépendances des variables de la façon suivante :

$$u = u(x; \boldsymbol{\alpha}), \quad (4.3)$$

$$T = T(x; \boldsymbol{\alpha}). \quad (4.4)$$

Nous écrivons les dépendances des propriétés physiques de manière similaire :

$$\nu = \nu(x; \boldsymbol{\alpha}), \quad (4.5)$$

$$\beta = \beta(x; \boldsymbol{\alpha}), \quad (4.6)$$

$$D = D(x; \boldsymbol{\alpha}). \quad (4.7)$$

Pour ce problème non-linéaire 1D, nous avons choisi un domaine de calcul unitaire $\Omega \in [0, 1]$. Dans ce projet, nous considérons seulement les conditions de Dirichlet. Cela signifie que nous imposons des conditions de Dirichlet pour les deux variables (u et T) aux deux extrémités du domaine ($x = 0$ et $x = 1$) :

$$u = u_{bc} \quad \text{sur } \Gamma_D, \quad (4.8)$$

$$T = T_{bc} \quad \text{sur } \Gamma_D. \quad (4.9)$$

Nous avons donc $\Gamma_D = \Gamma = \partial\Omega$.

4.2 Données d'entrée

Pour résoudre les sensibilités du problème test 1D, l'utilisateur doit fournir un minimum de données qui sont nécessaires à la discrétisation du problème et à la résolution des sensibilités.

Ces données sont :

- L'ordre n des sensibilités à résoudre ;
- Le nombre q de paramètres ;
- Le nombre nel d'éléments finis ;
- Le facteur de distribution des éléments ;
- Le type des éléments finis utilisés (i.e. : la valeur de p) qui dépend de l'ordre du polynôme de Lagrange utilisé pour les fonctions d'interpolation ;
- Le nombre de points de Gauss npg pour les quadratures de Gauss ;
- Le critère de convergence ϵ pour la méthode de Newton ;

- Les fonctions régissant les propriétés physiques (*e.g.* : ρ , ν , etc.);
- Les fonctions régissant les sensibilités des propriétés physiques d'ordre 1 à n ;
- Les fonctions régissant les termes sources f_s et q_s et leurs sensibilités d'ordre 1 à n ;
- Les conditions limites et leurs sensibilités.

4.3 Procédure générale implémentée

Dans ce travail, nous proposons une méthode des sensibilités continues d'ordre n par rapport à q paramètres. Celle-ci comprend les étapes suivantes :

1. Déterminer les systèmes de sensibilités d'ordre os à évaluer pour os variant de 0 à n ;
2. Générer les composantes des équations de sensibilités pour chaque système d'ordre os à évaluer ;
3. Enregistrer ces composantes dans une structure de données générique ;
4. Résoudre les équations de sensibilités de chaque système d'ordre os ;
5. Refaire les étapes 4 et 5 jusqu'à ce que os soit égal à n .

La première étape consiste à déterminer les systèmes de sensibilités d'ordre os à évaluer. Cela signifie que nous devons identifier et dénombrer les différentes façons de dériver les EDP à l'ordre os par rapport à q paramètres d'intérêt pour $os = 0$ à n . Chaque façon de dériver correspond à un système de sensibilités. Autrement dit, nous devons trouver les ordres n_i ($i = 1$ à q) des dérivées partielles par rapport à chacun des q paramètres de manière à avoir $\sum_{i=1}^q n_i = os$. Nous enregistrons les données associées aux dérivées partielles des systèmes de sensibilités d'ordre 0 à n dans le tableau *CombSens*, tandis que les autres données pertinentes (*e.g.* : nombre de systèmes de sensibilités d'ordre os) sont stockées dans le tableau *InfoSens*. Par la suite, pour chaque ordre os , allant de 0 à n , nous devons générer et résoudre successivement chacun des systèmes associés à l'ordre courant os . Rappelons que l'ordre 0 représente les EDP elles-mêmes. Il existe donc un seul système de sensibilités d'ordre 0. La figure 4.1 donne une vision globale du fonctionnement de notre méthode. Notons que l'emplacement *InfoSens*(2, os) du tableau *InfoSens* contient le nombre de systèmes d'ordre os à résoudre. Dans les sections qui suivent, nous abordons en détail les étapes de génération et de résolution des équations de sensibilités nécessaires à notre algorithme pour calculer les sensibilités d'ordre n par rapport à q paramètres.

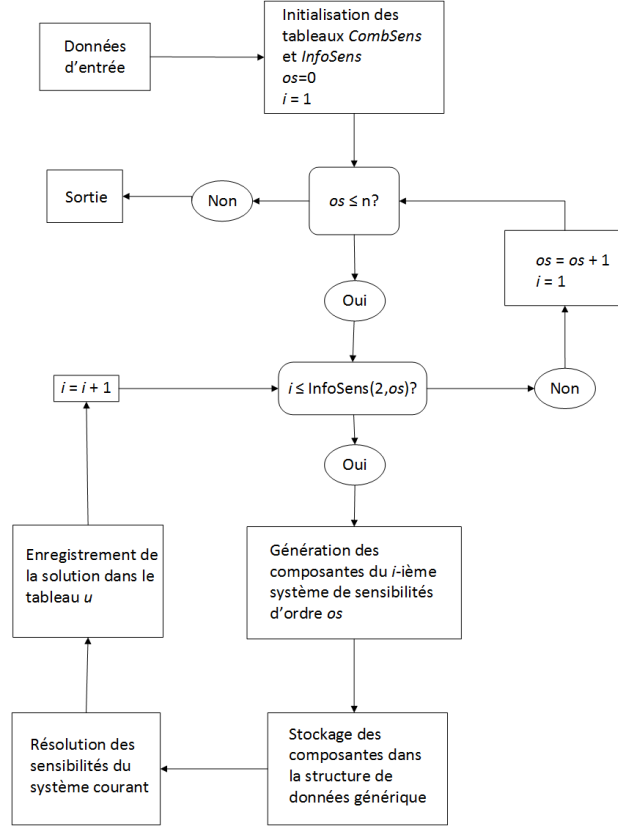


Figure 4.1 Schéma de la méthode implémentée.

4.4 Algorithme pour déterminer les systèmes de sensibilités à évaluer

Lorsque nous considérons les sensibilités d'ordre n par rapport à q paramètres, nous devons d'abord déterminer les dérivées que nous devons évaluer pour chaque variable d'état. Cela nous indique alors le nombre de systèmes de sensibilités d'ordre n que nous devons résoudre. Nous devons appliquer ces dérivées aux EDP et résoudre les équations obtenues pour obtenir les sensibilités d'ordre n . Par exemple, pour les sensibilités d'ordre 2 avec 3 paramètres (α_1 , α_2 et α_3), les dérivées sont :

$$S^{(\alpha_1(2))} = \frac{\partial^2}{\partial \alpha_1^2}, \quad S^{(\alpha_2(2))} = \frac{\partial^2}{\partial \alpha_2^2}, \quad S^{(\alpha_3(2))} = \frac{\partial^2}{\partial \alpha_3^2}, \quad (4.10)$$

$$S^{(\alpha_1(1)\alpha_2(1))} = \frac{\partial^2}{\partial \alpha_1 \partial \alpha_2}, \quad S^{(\alpha_1(1)\alpha_3(1))} = \frac{\partial^2}{\partial \alpha_1 \partial \alpha_3}, \quad S^{(\alpha_2(1)\alpha_3(1))} = \frac{\partial^2}{\partial \alpha_2 \partial \alpha_3}. \quad (4.11)$$

Pour ce cas spécifique, en plus des 3 systèmes d'ordre 1 (i.e. : $\frac{\partial}{\partial \alpha_1}$, $\frac{\partial}{\partial \alpha_2}$ et $\frac{\partial}{\partial \alpha_3}$), nous devons résoudre les 6 systèmes d'ordre 2 (équations (4.10) et (4.11)). L'équation (4.10) représente

les dérivées d'ordre 2 par rapport à un seul paramètre, alors que l'équation (4.11) définit les dérivées croisées (paramètres multiples). Notons que nous considérons que les fonctions dérivées sont suffisamment régulières pour que l'ordre d'évaluation des dérivées croisées n'affecte pas le résultat de l'opérateur de dérivation (i.e. : $\frac{\partial^2}{\partial\alpha_1\partial\alpha_2} = \frac{\partial^2}{\partial\alpha_2\partial\alpha_1}$).

Afin de déterminer le nombre de façons de dériver les équations à l'ordre n par rapport à q paramètres, nous utilisons un problème de combinatoire semblable à ceux de la section 3.2.1. Celui-ci consiste à chercher le nombre de façons de positionner n unités de dérivée parmi q paramètres. Une autre façon de voir ce problème est de chercher les différentes manières d'appliquer les dérivées partielles $\frac{\partial^{n_i}}{\partial\alpha_i}$ à l'EDP, pour $i = 1$ à q , de manière à ce que $n = n_1 + n_2 + \dots + n_q$ et où l'ordre dans lequel les dérivées sont appliquées est sans importance. Nous devons donc trouver les combinaisons de n éléments sélectionnés dans l'ensemble $E = \left\{ \frac{\partial}{\partial\alpha_1}, \frac{\partial}{\partial\alpha_2}, \dots, \frac{\partial}{\partial\alpha_q} \right\}$ contenant q éléments (se référer à la section A). Ces éléments peuvent être choisis plus d'une fois. Cela signifie qu'ils peuvent être répétés. Le nombre de répétitions de chaque élément $\frac{\partial}{\partial\alpha_i}$ indique l'ordre n_i de la dérivée par rapport au paramètre α_i . Si l'élément $\frac{\partial}{\partial\alpha_i}$ n'apparaît pas dans la combinaison, nous en déduisons que n_i vaut 0. Par ailleurs, comme l'ordre dans lequel les dérivées partielles sont appliquées est sans importance, l'ordre des éléments choisis est également sans importance. Il s'agit donc de combinaisons avec répétition. Le nombre total de ces combinaisons s'obtient à l'aide de la formule de combinaison avec répétition (voir équation (A.6)) :

$$C_n^{m+q-1} = \frac{(n+q-1)!}{n!(q-1)!}. \quad (4.12)$$

Prenons un exemple. Lorsque nous cherchons les systèmes de sensibilités d'ordre 5 avec 3 paramètres, il existe un total de $\frac{(5+3-1)!}{5!2!} = 21$ combinaisons possibles ($n = 5$ et $q = 3$). Voici une de ces combinaisons :

$$\left\{ \frac{\partial}{\partial\alpha_2}, \frac{\partial}{\partial\alpha_3}, \frac{\partial}{\partial\alpha_2}, \frac{\partial}{\partial\alpha_1}, \frac{\partial}{\partial\alpha_2} \right\},$$

ou de façon équivalente, puisque l'ordre des éléments n'a aucune importance :

$$\left\{ \frac{\partial}{\partial\alpha_1}, \frac{\partial}{\partial\alpha_2}, \frac{\partial}{\partial\alpha_2}, \frac{\partial}{\partial\alpha_2}, \frac{\partial}{\partial\alpha_3} \right\}.$$

Ainsi, lorsque nous calculons les sensibilités d'ordre 5 par rapport à 3 paramètres, un des 21 systèmes de sensibilités à étudier correspond à la dérivée partielle $\frac{\partial^5}{\partial\alpha_1\partial\alpha_2^3\partial\alpha_3}$ des EDP.

Une fonction récursive permet à notre algorithme de déterminer toutes les combinaisons possibles pour dériver les EDP à l'ordre n par rapport à q paramètres distincts. Chaque combinaison s'obtient en résolvant le système d'EDP pour cette sensibilité. Pour chaque système de sensibilités d'ordre n , la fonction récursive évalue l'ordre n_i des dérivées partielles par rapport à chacun des q paramètres α_i de telle sorte que $\sum_{i=1}^q n_i = n$. L'ordre des dérivées par rapport à chacun des paramètres α_i peut donc varier de 0 à n .

L'algorithme 1 résume le processus récursif utilisé pour déterminer les ordres n_i des dérivées par rapport à chacun des q paramètres α_i pour chaque système. Expliquons globalement comment l'algorithme 1 fonctionne. Tout d'abord, il appelle la fonction $Calc_nbrSens(q, n)$ afin d'évaluer le nombre r de systèmes d'équations de sensibilités que nous devons résoudre à l'ordre n avec q paramètres. Comme chaque système de sensibilités correspond à une combinaison avec répétition de n éléments tirés dans un ensemble de q éléments, cette fonction utilise la formule des combinaisons avec répétition (voir équation (4.12)). Ensuite, l'algorithme initialise la matrice d qui emmagasine les ordres des dérivées partielles par rapport à chaque paramètre pour chacune des r combinaisons. Cela implique que $d \in \mathbb{N}^{q \times r}$. L'algorithme initialise également les variables temporaires *compteur* et $d2$ qu'il utilise pour les appels récursifs à la sous-routine $CalcSensParam$. Enfin, l'algorithme appelle la sous-routine récursive $CalcSensParam$ pour calculer les ordres des dérivées partielles par rapport à chaque paramètre de chacune des r combinaisons.

Pour mieux comprendre le fonctionnement de l'algorithme 1, montrons comment il détermine les systèmes de sensibilités d'ordre 2 avec 3 paramètres. Nous avons donc $n = 2$ et $q = 3$. Lors de l'étape d'initialisation, nous avons :

$$r = C_2^{2+3-1} = 6; \quad \text{compteur} = 0;$$

$$d = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}; \quad d2 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}.$$

Nous appelons une première fois la sous-routine $CalcSensParam(3,2,d,d2,compteur)$. L'algorithme entre alors dans la boucle **Pour** $i = 2$ à 0. À ce moment, la variable i vaut 2. Nous

Algorithme 1 : Pseudocode pour déterminer les combinaisons des dérivées partielles

Entrée: n et q

Sortie: $d(:, :)$ et r

Initialisation;

$r \leftarrow \text{Calc_nbrSens}(q, n)$

$d(1 : q, 1 : r) \leftarrow n$

$\text{compteur} \leftarrow 0$

$d2(1 : q) \leftarrow n$

$d, \text{compteur} \leftarrow \text{CalcSensParam}(q, n, d, d2, \text{compteur})$

si ($\text{compteur} \neq r$) **alors**

| message d'erreur;

sinon

| **retourner** r et $d(:, :)$;

fin

Sous-routine $\text{CalcSensParam}(q, n, d, d2, r)$

Déclaration de variables

temp_q ;

temp_n ;

pour $i = n$ à 0 **faire**

| $d2(q) = i$

| **si** $q > 2$ **alors**

| | $\text{temp_q} = q - 1$

| | $\text{temp_n} = n - i$

| | **Appel** $\text{CalcSensParam}(\text{temp_q}, \text{temp_n}, d, d2, r)$;

| **fin**

| **sinon si** $q \leq 2$ **alors**

| | $r = r + 1$

| | **si** $q = 1$ **alors**

| | | $d(:, r) = d2(:, :)$

| | | Sortie;

| | **fin**

| | $d2(1) = n - i$

| | $d(:, r) = d2(:, :)$

| **fin**

fin

obtenons alors :

$$d2 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} .$$

Comme $q = 3$, l'algorithme entre dans la boucle conditionnelle **si** $q > 2$ **alors**. On trouve donc :

$$\begin{aligned} temp_q &= 3 - 1 = 2, \\ temp_n &= 2 - 2 = 0. \end{aligned}$$

L'algorithme appelle alors $CalcSensParam(2,0,d,d2,0)$. L'algorithme entre dans la boucle **Pour** $i = 0$ à 0 dans laquelle il fera une seule itération, soit en $i = 0$. Il y trouve :

$$d2 = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} .$$

L'algorithme choisit la condition **sinon si** $q \leq 2$, car q vaut 2. Il incrémente alors le compteur, désigné par r dans la sous-routine $CalcSensParam$. La valeur de r est donc 1, puisqu'elle valait 0 lors de l'appel de la sous-routine. De plus, $d2(1) = n - i = 0 - 0$, ce qui signifie que :

$$r = 1; \quad d2 = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}; \quad d = \begin{bmatrix} 0 & 2 & 2 & 2 & 2 & 2 \\ 0 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix} .$$

On poursuit dans la première boucle *Pour* $i = 2$ à 0 que l'algorithme a initialement appelée avec $i = 1$ et ainsi de suite. Voici les valeurs que nous obtenons, en suivant l'algorithme pas à pas, pour la variable r , le vecteur $d2$ et le tableau d :

$$r = 2; \quad d2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}; \quad d = \begin{bmatrix} 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 & 2 \\ 2 & 1 & 2 & 2 & 2 & 2 \end{bmatrix};$$

$$r = 3; \quad d2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}; \quad d = \begin{bmatrix} 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 0 & 2 & 2 & 2 \\ 2 & 1 & 1 & 2 & 2 & 2 \end{bmatrix};$$

$$r = 4; \quad d2 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}; \quad d = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 & 2 \\ 0 & 1 & 0 & 2 & 2 & 2 \\ 2 & 1 & 1 & 0 & 2 & 2 \end{bmatrix};$$

$$r = 5; \quad d2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}; \quad d = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 & 1 & 2 \\ 2 & 1 & 1 & 0 & 0 & 2 \end{bmatrix};$$

$$r = 6; \quad d2 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}; \quad d = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

En prenant la matrice finale obtenue pour d ($r = 6$) et en faisant le lien entre les valeurs de celle-ci et l'ordre des dérivées partielles par rapport aux paramètres α_i appropriés ($i = 1$ à 3), nous retrouvons les systèmes d'ordre 2 avec 3 paramètres (voir les équations (4.10) et (4.11)). En effet, chaque colonne nous donne les ordres des dérivées partielles par rapport à chacun des 3 paramètres α_i ($i = 1$ à 3) pour un système spécifique. Ici, il y a 6 colonnes donc 6 systèmes ($r = 6$) d'ordre 2. La matrice d est donc formée de 6 colonnes qui correspondent à chacun des 6 systèmes de sensibilités et de 3 lignes qui correspondent à chacun des 3 paramètres α_i .

Pour la colonne $d(:, 2) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$, nous avons la combinaison :

$$\left\{ \frac{\partial}{\partial \alpha_2}, \frac{\partial}{\partial \alpha_3} \right\}.$$

Cette combinaison est associée au deuxième système, soit $\frac{\partial^2}{\partial \alpha_2 \partial \alpha_3}$. Par conséquent, les combi-

naisons et les dérivées partielles associées à chacune des colonnes de la matrice d sont :

$$d(:, 1) = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \rightarrow \left\{ \frac{\partial}{\partial \alpha_3}, \frac{\partial}{\partial \alpha_3} \right\} \text{ soit } \frac{\partial^2}{\partial \alpha_3^2};$$

$$d(:, 2) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \rightarrow \left\{ \frac{\partial}{\partial \alpha_2}, \frac{\partial}{\partial \alpha_3} \right\} \text{ soit } \frac{\partial^2}{\partial \alpha_2 \partial \alpha_3};$$

$$d(:, 3) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \rightarrow \left\{ \frac{\partial}{\partial \alpha_1}, \frac{\partial}{\partial \alpha_3} \right\} \text{ soit } \frac{\partial^2}{\partial \alpha_1 \partial \alpha_3};$$

$$d(:, 4) = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \rightarrow \left\{ \frac{\partial}{\partial \alpha_2}, \frac{\partial}{\partial \alpha_2} \right\} \text{ soit } \frac{\partial^2}{\partial \alpha_2^2};$$

$$d(:, 5) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \rightarrow \left\{ \frac{\partial}{\partial \alpha_1}, \frac{\partial}{\partial \alpha_2} \right\} \text{ soit } \frac{\partial^2}{\partial \alpha_1 \partial \alpha_2};$$

$$d(:, 6) = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \rightarrow \left\{ \frac{\partial}{\partial \alpha_1}, \frac{\partial}{\partial \alpha_1} \right\} \text{ soit } \frac{\partial^2}{\partial \alpha_1^2}.$$

Il existe une approche plus simple pour déterminer les combinaisons de n éléments $\frac{\partial}{\partial \alpha_i}$ pris dans l'ensemble $E = \left\{ \frac{\partial}{\partial \alpha_1}, \frac{\partial}{\partial \alpha_2}, \dots, \frac{\partial}{\partial \alpha_q} \right\}$ en admettant qu'il y a possibilité de répétition. Il s'agit d'utiliser des boucles imbriquées qui sont dépendantes les unes des autres (i.e. : **Pour** $param_i = n - param_q - \dots - param_{i+2} - param_{i+1}$). Ce principe est illustré par l'intermédiaire de l'algorithme 2. Toutefois, notre méthode ne serait pas générique si nous faisons usage de ce type d'approche. Le problème avec les boucles imbriquées c'est que leur nombre doit être adapté selon le nombre q de paramètres. La sous-routine *CalcSensParam*

est donc avantageuse, car elle ne requiert pas de modifications en fonction de l'ordre n ou du nombre de paramètres q . Elle appelle simplement les boucles de l'algorithme 2 de façon récursive jusqu'à ce que la variable q soit égale à 2. Il s'agit de la dernière boucle, car la sous-routine *CalcSensParam* considère les boucles pour $param_q$ jusqu'à $param_2$. Cette façon de procéder permet au programme de savoir quand il doit arrêter de faire des appels récursifs. Lorsque les boucles appelées récursivement se terminent, nous retrouvons dans le tableau d toutes les combinaisons associées aux systèmes de sensibilités d'ordre n avec q paramètres.

Algorithme 2 : Boucle équivalente à la sous-routine récursive *CalcSensParam*

Entrée: n

Sortie: $d(:, :)$

```

pour  $param_q = n$  à 0 faire
  pour  $param_{q-1} = (n - param_q)$  à 0 faire
    ...
    pour  $param_2 = (n - param_q - param_{q-1} - \dots - param_3)$  à 0 faire
       $compteur \leftarrow compteur + 1$ 
       $param_1 = n - param_q - param_{q-1} - \dots - param_2$ 
       $i \leftarrow 0$ 
      tant que  $(i < q)$  faire
         $i \leftarrow i + 1$ 
         $d(i, compteur) = param_i$ 
      fin
    fin
  fin
fin

```

4.5 Algorithme pour générer les équations de sensibilités d'un système

Chacun des systèmes de sensibilités d'ordre n à q paramètres doit être généré et résolu individuellement. Dans cette section, nous présentons la méthode générale mise en place pour générer les équations de sensibilités continues d'un système de sensibilités spécifique d'ordre arbitraire n . Cette méthode détermine les composantes de chacun des termes des équations de sensibilités, soient les coefficients et les ordres des dérivées des fonctions (i.e. : les variables et les propriétés). Pour développer et implémenter cette méthode, nous nous sommes basés sur les formulations générales des équations de sensibilités (voir les équations (3.43) et (3.55)) définies au chapitre précédent.

Nous présentons la méthode pour calculer les composantes des dérivées des produits de fonc-

tions sous forme de pseudocode dans les algorithmes 3 et 4. Ces composantes permettent de retracer les termes de sensibilités découlant de la dérivée d'un produit de fonctions particulier de l'EDP. Pour chaque produit de m fonctions contenu dans l'EDP (ordre 0), il est alors possible d'évaluer le nombre total s de termes produits par la dérivée partielle appliquée au système ainsi que les coefficients et les ordres de dérivées de chaque fonction pour chacun de ces s termes. Pour obtenir la totalité des composantes des termes qui composent l'équation de sensibilités d'ordre n , nous devons calculer ces composantes (*i.e.* : nombre de termes, valeurs des coefficients et valeurs des ordres de dérivées partielles) pour chaque produit de fonctions présent dans l'EDP. On suppose que les fonctions représentant les variables et les propriétés physiques des EDP sont infiniment différentiables.

Algorithme 3 : Pseudocode pour calculer les composantes des équations de sensibilités
- Partie 1

Entrée: q , m et $n(1 : q)$

Sortie: $c(1 : s)$ et $d(1 : m, 1 : q, 1 : s)$

Initialisation;

$s \leftarrow Nt_multinomial(m, n, q)$

$c(1 : s) \leftarrow 1$

$compteur \leftarrow 0$

$m_max \leftarrow m$

pour $i= 1$ à q **faire**

| $temp_c(i) \leftarrow (n(i))!$

fin

pour $i= 1$ à m **faire**

| $temp_d(i, 1 : q) \leftarrow n(1 : q)$

| **pour** $j= 1$ à s **faire**

| | $d(i, 1 : q, j) \leftarrow n(1 : q)$

| **fin**

fin

$c, d, compteur \leftarrow multinomeNewton(m, q, n, d, temp_d, compteur, c, temp_c, m_max)$

si ($compteur \neq s$) **alors**

| message d'erreur;

sinon

| **retourner** s , $c(:)$ et $d(:, :, :)$;

fin

Considérons l'algorithme 3. Celui-ci calcule les diverses composantes de la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ appliquée à un produit de m fonctions. Les variables d'entrées de l'algorithme 3 lui fournissent les informations suivantes :

- q : le nombre de paramètres de sensibilité ;

Algorithme 4 : Pseudocode pour calculer les composantes des équations de sensibilités

- Partie 2

Entrée: $m, q, n(1 : q), m_max, d2(1 : m, 1 : q), s, coeff(:)$

Sortie: $c(1 : s), d(1 : m, 1 : q, 1 : s), d2(1 : m, 1 : q), s, coeff(:)$

Sous-routine multinomeNewton($m, q, n, d, d2, s, c, coeff, m_max$)

Initialisation;

$temp_coeff(1 : q) \leftarrow coeff(1 : q)$

$temp_m \leftarrow 0$

$temp_n(1 : q) \leftarrow n(1 : q)$

$temp_q \leftarrow q$

pour $i = n(q)$ à 0 **faire**

$d2(m, q) = i$

$temp_coeff(q) = \frac{coeff(q)}{i!}$

si $m > 2$ **alors**

$temp_m = m - 1$

$temp_n(q) = n(q) - i$

Appel multinomeNewton($temp_m, temp_q, temp_n, d, d2, s, c,$
 $temp_coeff, m_max$) ;

fin

sinon si $m \leq 2$ **alors**

si $q > 1$ **alors**

$temp_q = q - 1$

$temp_m = m_max$

$d2(1, q) = n(q) - i$

$d(:, q, s + 1) = d2(:, q)$

$temp_coeff(q) = \frac{temp_coeff(q)}{(d2(1, q))!}$

$temp_coeff(temp_q) = temp_coeff(q) \cdot coeff(temp_q)$

Appel multinomeNewton($temp_m, temp_q, temp_n, d, d2, s, c,$
 $temp_coeff, m_max$) ;

fin

sinon si $q = 1$ **alors**

$s = s + 1$

$d2(1, q) = n(q) - i$

$d(:, :, s) = d2(:, :)$

$c(s) = \frac{temp_coeff(q)}{(d2(1, q))!}$

fin

fin

fin

- m : le nombre de fonctions multipliées dans le produit qu'on dérive ;
- $n(1 : q)$: l'ordre des dérivées partielles par rapport à chacun des q paramètres.

L'algorithme 3 appelle la fonction $Nt_multinomial(m, n, q)$ pour calculer le nombre total s de termes, composés de m fonctions, découlant de la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}$ appliquée à un produit de m fonctions (se référer à (3.54)). Ensuite, il initialise le vecteur c , qui stocke les coefficients de ces s termes, ainsi que le tableau tridimensionnel d , qui contient les ordres de dérivées des fonctions de chacun de ces s termes. Il est à noter que $s \in \mathbb{N}^*$, $c \in \mathbb{N}^s$ et $d \in \mathbb{N}^{m \times q \times s}$. L'algorithme 3 initialise également les variables temporaires, soient le vecteur $temp_c \in \mathbb{N}^q$, la matrice $temp_d \in \mathbb{N}^{m \times q}$ et la variable $compteur \in \mathbb{N}$. Ces variables temporaires sont utilisées pour les appels récursifs à la sous-routine $multinomeNewton$ qui calcule les coefficients des s termes et l'ordre des dérivées de chacune des m fonctions de ces s termes. En se référant à l'équation (3.55), la variable $temp_c(i)$ donne le coefficient $\frac{n_i!}{k_{i,1}k_{i,2}\dots k_{i,m}}$ affecté à un terme découlant de la dérivée d'ordre n_i par rapport au paramètre α_i , tandis que $temp_d(j, i)$ donne l'ordre $k_{i,j}$ de la dérivée de la fonction numéro j par rapport au paramètre α_i pour le terme de sensibilités courant dans la fonction récursive. Comme nous calculons les coefficients par le biais de $\frac{n_1!}{k_{1,1}!k_{1,2}!\dots k_{1,m}!} \frac{n_2!}{k_{2,1}!k_{2,2}!\dots k_{2,m}!} \dots \frac{n_q!}{k_{q,1}!k_{q,2}!\dots k_{q,m}!}$, nous initialisons $temp_c(i)$ avec la factorielle de n_i . La multiplication entre les coefficients $temp_c(i) = \frac{n_i!}{k_{i,1}!k_{i,2}!\dots k_{i,m}!}$ (pour $i = 1$ à q) se fait plus tard dans la sous-routine $multinomeNewton$ avec la condition **si** $q > 1$ **alors** par le biais de $temp_coeff(temp_q) = temp_coeff(q) \cdot coeff(temp_q)$. Le coefficient $\frac{n_i!}{k_{i,1}!k_{i,2}!\dots k_{i,m}!}$ étant un entier (i est le numéro du paramètre α_i), nous initialisons la variable $temp_coeff(i)$ qui lui associée avec son numérateur ($n_i!$). Cela évite d'avoir temporairement une fraction contenue dans $temp_coeff(i)$ lors de la division par $k_{i,j}$ avec $j = 1$ à m . Autrement, il y aurait une erreur lors de l'exécution du code, puisque la variable entière $temp_coeff(i)$ contiendrait un nombre réel.

Pour un système de sensibilités $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\alpha_2^{n_2}\dots\alpha_q^{n_q}}$, cette même dérivée appliquée à 2 produits distincts de m fonctions engendre des composantes de sensibilités qui sont exactement identiques lorsque la valeur de m pour ces 2 produits est identique. Par exemple, si nous avons 2 produits de 3 fonctions et que nous leur appliquons exactement la même dérivée, le nombre de termes, la valeur des coefficients et l'ordre des dérivées des fonctions f_1 , f_2 et f_3 de chacun de ces termes sont égaux pour la dérivée de ces 2 produits.

La sous-routine $multinomeNewton$ appelée dans l'algorithme 3 est récursive. Cela signifie qu'elle s'appelle elle-même. Cette sous-routine, détaillée dans l'algorithme 4, permet de calculer et de retourner les composantes des termes de sensibilités découlant de la dérivée d'un produit de fonctions particulier. Ces composantes correspondent aux ordres des dérivées des

fonctions et aux coefficients des s termes. Il est à noter que la sous-routine *multinomeNewton* est indépendante de la routine principale de l'algorithme 3. Ainsi, bien que certains paramètres de cette sous-routine aient des appellations identiques à certaines variables locales du procédé principal, ceux-ci sont distincts ; ils ne pointent pas sur le même emplacement.

Le lecteur intéressé peut également retrouver dans l'annexe E les détails de la sous-routine explicite équivalente à la sous-routine récursive *multinomeNewton*. Nous présentons cette forme explicite par le biais de l'algorithme 5 (voir annexe E). Dans le cas de l'algorithme 5, celui-ci est constitué de boucles imbriquées permettant de calculer la valeur des composantes des termes de sensibilités découlant d'un produit de fonctions. La sous-routine récursive décrite par l'algorithme 4 exécute le même procédé, mais de façon récursive. En fait, l'algorithme 4 appelle récursivement les boucles **Pour** que l'on retrouve dans l'algorithme 5

La récursivité de la sous-routine *multinomeNewton* assure la généralité du processus. Cela n'est pas possible avec l'utilisation explicite des boucles imbriquées (algorithme 5). En effet, la récursivité est essentielle pour permettre à l'algorithme de déterminer les composantes des termes découlant de l'application des dérivées partielles d'ordre $n(1 : q)$ sur un produit de m monômes par rapport à q paramètres pour des valeurs arbitraires de m , q et $n(1 : q)$. Dans le cas des boucles imbriquées, celles-ci doivent être adaptées selon les valeurs de m et q : nous devons avoir $q \cdot (m - 1)$ boucles imbriquées.

4.5.1 Exemple de fonctionnement des algorithmes 3 et 4

Nous présentons un exemple simple dans le but d'aider à saisir la façon dont fonctionnent les algorithmes 3 et 4 pour générer les composantes des termes découlant de la dérivée d'ordre n (i.e. : $\frac{\partial^n}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$) appliquée à un produit de m fonctions. Nous détaillons le déroulement de la procédure des algorithmes pour la dérivée troisième $\frac{\partial^3}{\partial \alpha_1 \partial \alpha_2^2}$ appliquée au terme $\rho u \cdot \nabla u$ de l'équation (3.1). Les données d'entrée du programme principal (voir algorithme 3) sont donc :

$$q = 2, \quad m = 3 \quad \text{et} \quad n = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

où q est le nombre de paramètres des sensibilités, m est le nombre de fonctions dans le produit $\rho u \cdot \nabla u$ et le vecteur n contient les ordres pour les dérivées partielles du système par rapport aux différents paramètres α_i ($i = 1$ à 2). Pour cet exemple, l'algorithme 3 s'exécute de la façon qui suit. Premièrement, il appelle la fonction *Nt_multinomial*(3, n , 2), où n est un vecteur

de dimension q . La fonction *Nt_multinomial* retourne le nombre total s de termes résultant de l'application de la dérivée $\frac{\partial^3}{\partial\alpha_1\partial\alpha_2^2}$ au terme $\rho u \cdot \nabla u$ selon l'équation (3.54), soit :

$$s = C_1^{1+3-1} \cdot C_2^{2+3-1} = 18.$$

L'algorithme alloue alors la mémoire pour le vecteur c et la matrice d , contenant respectivement les coefficients et les ordres de dérivées, de manière à ce que $c \in \mathbb{N}^{18}$ et $d \in \mathbb{N}^{3 \times 2 \times 18}$. Par la suite, la sous-routine récursive *multinomeNewton* est appelée. Les valeurs retournées par cette dernière sont les suivantes :

$$d(1, :, :) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \end{bmatrix};$$

$$d(2, :, :) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix};$$

$$d(3, :, :) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$c^T = [1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1];$$

$$\text{compteur} = 18.$$

La valeur enregistrée dans l'emplacement $d(j, i, k)$ correspond à l'ordre de la dérivée partielle de la j -ième fonction par rapport au paramètre α_i pour le k -ième terme résultant de la dérivée appliquée à un terme particulier de l'EDP (se référer à l'équation (3.52)). L'indice i correspond au numéro de la rangée de la matrice $d(j, :, :)$, alors que l'indice k réfère au numéro de la colonne de la même matrice. Par exemple, si nous prenons la cinquième colonne ($k = 5$) des matrices $d(1, :, :)$, $d(2, :, :)$ et $d(3, :, :)$, nous obtenons alors les composantes du cinquième terme provenant de la dérivée $\frac{\partial^3}{\partial\alpha_1\partial\alpha_2^2}$ de $\rho u \cdot \nabla u$ de l'EDP. Ainsi, la première fonction correspond à la solution de base de ρ puisqu'elle est affectée des dérivées d'ordre 0 (*i.e.* : $\frac{\partial^0}{\partial\alpha_1^0\partial\alpha_2^0}$). Pour la seconde fonction, soit u , nous avons la dérivée $\frac{\partial^2}{\partial\alpha_1\partial\alpha_2}$, alors que pour la troisième fonction, soit ∇u , nous avons la dérivée $\frac{\partial}{\partial\alpha_2}$. Le tableau tridimensionnel d et le vecteur c nous conduisent donc aux termes de l'équation de sensibilités d'ordre 3 du système $\frac{\partial^3}{\partial\alpha_1\partial\alpha_2^2}$ qui découlent de l'application de la dérivée au produit $\rho u \cdot \nabla u$. On peut aisément remarquer que les composantes placées dans le vecteur c et le tableau d par le biais des

algorithmes 3 et 4 sont égales à celles trouvées par les boucles imbriquées de l'algorithme 6 (voir sous-section E). En se basant sur la formulation générale (3.55) établie pour les équations de sensibilités d'ordre n avec q paramètre, nous obtenons donc :

$$\frac{\partial^3}{\partial \alpha_1 \partial \alpha_2^2} (\rho u \cdot \nabla u) = \sum_{k=1}^{s=18} c(k) \left(\rho \left(\alpha_1^{(d(1,1,k))} \alpha_2^{(d(1,2,k))} \right) u \left(\alpha_1^{(d(2,1,k))} \alpha_2^{(d(2,2,k))} \right) \cdot \nabla u \left(\alpha_1^{(d(3,1,k))} \alpha_2^{(d(3,2,k))} \right) \right). \quad (4.13)$$

4.5.2 Enregistrement des équations de sensibilités dans une structure de données générique

Les informations sur les équations de sensibilités et leur résolution sont conservées dans une structure de données générique. Nous avons mis en place un système permettant de classer les sensibilités d'ordre n , puisqu'il existe plusieurs systèmes de sensibilités d'ordre n lorsque q est différent de 1. Dans un premier temps, nous identifions pour chaque ordre os variant de 0 à n le nombre de systèmes à évaluer. En additionnant le nombre de systèmes pour $os = 0$ à n , nous trouvons le nombre total r de systèmes de sensibilités à résoudre incluant l'écoulement lui-même (ordre 0). Par la suite, nous attribuons un numéro variant de 1 à r pour chaque système de sensibilités (ou dérivée partielle). Cela permet d'attribuer un numéro unique à chacune des sensibilités pour les variables, soient u et T dans le cas des équations (4.1) et (4.2). En effet, nous créons un tableau tridimensionnel u de dimension $m \times nn \times r$ qui contient les solutions pour tous les systèmes de sensibilités, où m est le nombre de variables inconnues et nn est le nombre total de noeuds sur le domaine. Par exemple, pour une variable spécifique ce tableau ressemble à :

Tableau 4.1 Forme générale du tableau des solutions des r systèmes de sensibilités pour une variable particulière.

		k				
		1	2	3	...	r
No. noeud	1	$u(j, \vec{x}_1, 1)$	$u(j, \vec{x}_1, 2)$	$u(j, \vec{x}_1, 3)$	\dots	$u(j, \vec{x}_1, r)$
	2	$u(j, \vec{x}_2, 1)$	$u(j, \vec{x}_2, 2)$	$u(j, \vec{x}_2, 3)$	\dots	$u(j, \vec{x}_2, r)$
	3	$u(j, \vec{x}_3, 1)$	$u(j, \vec{x}_3, 2)$	$u(j, \vec{x}_3, 3)$	\dots	$u(j, \vec{x}_3, r)$
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	nn	$u(j, \vec{x}_{nn}, 1)$	$u(j, \vec{x}_{nn}, 2)$	$u(j, \vec{x}_{nn}, 3)$	\dots	$u(j, \vec{x}_{nn}, r)$

où k est le numéro du système de sensibilités. La donnée $u(j, \vec{x}_i, k)$ représente la solution de la variable numéro j au noeud i pour le k -ième système de sensibilités, où \vec{x}_i correspond aux coordonnées du noeud numéro i . Lors de la résolution des EDP d'ordre 0 à n , cette structure de données générique s'avère utile pour l'identification des sensibilités pour chacune des variables du problème.

De surcroît, nous stockons les combinaisons des dérivées partielles pour chaque système de sensibilités dans le tableau *CombSens*. Ce tableau est alloué dynamiquement, car le nombre de colonnes qu'on y retrouve varie en fonction du nombre q de paramètres d'intérêt du problème étudié. De plus, le nombre de lignes de *CombSens* dépend du nombre total de systèmes de sensibilités qui lui-même dépend de q et n . En effet, nous allouons la mémoire de ce tableau de manière à avoir $CombSens \in \mathbb{N}^{r \times q}$:

Tableau 4.2 Forme générale du tableau *CombSens*.

No. combinaison	n_1	n_2	\dots	n_q
1	0	0	\dots	0
2	\dots	\dots	\dots	\dots
\vdots	\vdots	\vdots	\vdots	\vdots
r	\dots	\dots	\dots	\dots

La colonne portant le titre *No. combinaison* indique le numéro des rangées du tableau *CombSens*. Les autres colonnes contiennent les ordres n_i ($i = 1$ à q) des dérivées partielles par rapport aux q paramètres α_i pour chacune des r combinaisons associées aux systèmes de sensibilités d'ordre 0 à n . L'insertion des valeurs du tableau *CombSens* se fait par le biais de l'algorithme 1 qui est exécuté sur l'ensemble des ordres de 0 à n . En réalité, il n'existe pas de colonne prévue pour contenir les numéros de combinaisons; cela est superflu. Ici, nous l'ajoutons afin d'illustrer clairement le principe d'utilisation du tableau.

Par ailleurs, nous créons le tableau *InfoSens* qui permet de stocker certaines informations sur les combinaisons des sensibilités (dérivées partielles) :

Tableau 4.3 Forme générale du tableau *InfoSens*.

Ordre	Nbr sensibilités	No. début	No. fin
0	1	1	1
1	C_1^{1+q-1}	2	$1 + C_1^{1+q-1}$
\vdots	\vdots	\vdots	\vdots
n	C_n^{n+q-1}	No. fin(n-1) +1	No. fin(n-1)+ C_n^{n+q-1}

La première colonne indique l'ordre os du système, la seconde colonne fournit le nombre de sensibilités d'ordre os lorsqu'on a q paramètres, tandis que la troisième colonne et la quatrième colonne nous donnent l'intervalle des numéros de combinaisons associées aux systèmes de sensibilités d'ordre os . Par exemple, la première rangée nous donne les informations pour le système d'ordre 0, soit le système d'équations d'état. Évidemment, nous y dénombrons un seul système. Par conséquent, nous avons une seule combinaison sur l'intervalle. Cela explique pourquoi le numéro du début de l'intervalle est le même que le numéro indiquant sa fin.

Notre approche prend les composantes de chacun des termes résultant de la dérivée d'ordre n d'un produit de fonctions de l'EDP et les associe aux numéros de combinaisons du tableau *CombSens* qui coïncident avec elles. Il s'agit alors de prendre les valeurs des ordres des dérivées partielles par rapport aux paramètres α_i ($i = 1$ à q) stockées dans le tableau d (voir algorithmes 3 et 4) et de comparer $d(j, :, k)$ aux combinaisons des dérivées partielles du tableau *CombSens*. Les indices j et k indiquent respectivement les numéros de la fonction et du terme de sensibilités concernés. Une fois que nous avons trouvé la combinaison qui coïncide avec les données de $d(j, :, k)$, nous faisons le lien entre le numéro d'identification de cette combinaison et la sensibilité définie dans $d(j, :, k)$. Pour ce faire, nous avons créé le tableau *NoComb*(1 : m , 1 : s) dans lequel nous transférons les numéros de combinaisons auxquels sont associés les ordres des dérivées du tableau d pour chacune des m fonctions de chacun des s termes des sensibilités. Ainsi, *NoComb*(j, k) contient le numéro de la combinaison qui coïncide avec les ordres des dérivées de $d(j, :, k)$. Nous reprenons alors l'équation (3.55) et la modifions pour obtenir la forme suivante :

$$\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}} \left(\prod_{j=1}^m f_j \right) = \sum_{k=1}^s c(k) \cdot \prod_{j=1}^m f_j^{(NoComb(j,k))}, \quad (4.14)$$

où s indique le nombre total de termes dans la sommation et m le nombre de fonctions dans le produit considéré. Pour chaque terme, les m fonctions ont chacune leur propre com-

binaison de dérivées partielles par rapport aux q paramètres (dérivée mixte). La variable $NoComb(j, k)$ mise en exposant, indique le numéro de cette combinaison pour la fonction f_j du k -ième terme résultant de la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\partial\alpha_q^{n_q}}(\prod_{j=1}^m f_j)$.

Nous utilisons le tableau *InfoSens* pour définir l'intervalle de recherche pour trouver les numéros de combinaisons que nous enregistrons dans le tableau *NoComb*. Pour ce faire, nous trouvons l'ordre $n_{j,k}$ de la sensibilité de la j -ième fonction du k -ième terme désignée par $d(j, :, k)$. Nous devons donc additionner l'ordre des dérivées partielles par rapport à chaque paramètre. Nous trouvons alors :

$$n_{j,k} = \sum_{i=1}^q d(j, i, k). \quad (4.15)$$

Par la suite, nous sélectionnons la rangée du tableau *InfoSens* dont la valeur est égale à $n_{j,k}$. Notez que les numéros des lignes du tableau *InfoSens*, représentant l'ordre, varient de 0 à n . Puis, nous prenons les valeurs des colonnes *No. début* et *No. fin* pour restreindre notre recherche dans le tableau *CombSens*. Nous comparons donc les données de $d(j, :, k)$ avec les données des rangées dont le numéro est compris entre les valeurs de *No. début* et *No. fin*. Par exemple, si nous avons $d(j, :, k) = [1 \ 2 \ 1]$, nous accédons à la ligne numéro 4 du tableau *InfoSens*, car $n_{j,k} = 1 + 2 + 1 = 4$. Supposons que, pour cette ligne, la valeur dans la colonne *No. début* soit 15 et celle dans la colonne *No. fin* soit 25. Pour trouver le numéro de combinaison coïncidant avec $d(j, :, k)$, nous effectuons une comparaison avec les combinaisons situées entre les lignes 15 et 25 du tableau *CombSens*, où le numéro de ligne correspond en fait au numéro de la combinaison.

Afin de montrer explicitement comment notre algorithme utilise cette structure de données générique, nous reprenons l'exemple de la dérivée $\frac{\partial^3}{\partial\alpha_1\partial\alpha_2^2}$ du terme $\rho u \cdot \nabla u$ ($q = 2$ et $m = 3$). Nous représentons les données du tableau *CombSens* par le tableau 4.4 ($n = 3$ et $q = 2$). Quant au tableau *InfoSens*, nous illustrons ses valeurs dans le tableau 4.5. Les informations du tableau *InfoSens* peuvent être validées en regardant directement les données du tableau *CombSens*. Nous notons que le numéro de combinaison de la sensibilité d'ordre 0 est bel et bien 1. Par la suite, pour les sensibilités d'ordre 1, le tableau 4.5 montre qu'il y a 2 systèmes de sensibilités dont les numéros de combinaisons sont compris entre 2 et 3. D'après le tableau 4.4, ces combinaisons sont $\frac{\partial}{\partial\alpha_2}$ et $\frac{\partial}{\partial\alpha_1}$, dont les numéros de combinaisons respectifs sont 2 et 3, ce qui coïncide avec les renseignements fournis par le tableau 4.5. Le tableau 4.5 indique également qu'il existe 3 systèmes de sensibilités d'ordre 2 ainsi que 4 systèmes d'ordre

3 avec 2 paramètres. Les intervalles de numéros de combinaisons assignés pour ces systèmes sont respectivement : 4 à 6 et 7 à 10. Nous remarquons alors dans le tableau 4.4 qu'il y a concordance; le tableau *CombSens* attribue bel et bien les numéros de combinaisons 4 à 6 pour les 3 systèmes d'ordre 2 (*i.e* : $\frac{\partial^2}{\partial\alpha_2^2}, \frac{\partial^2}{\partial\alpha_1\partial\alpha_2}, \frac{\partial^2}{\partial\alpha_1^2}$), tandis qu'il assigne les numéros 7 à 10 pour les 4 systèmes d'ordre 3 (*i.e* : $\frac{\partial^3}{\partial\alpha_2^3}, \frac{\partial^3}{\partial\alpha_1\partial\alpha_2^2}, \frac{\partial^3}{\partial\alpha_1^2\partial\alpha_2}, \frac{\partial^3}{\partial\alpha_1^3}$).

Tableau 4.4 Combinaisons des systèmes de sensibilités d'ordre 0 à 3 avec 2 paramètres.

No. combinaison	n_1	n_2
1	0	0
2	0	1
3	1	0
4	0	2
5	1	1
6	2	0
7	0	3
8	1	2
9	2	1
10	3	0

Tableau 4.5 Information sur les sensibilités d'ordre 0 à 3 avec 2 paramètres.

Ordre	Nbr sensibilités	No. début	No. fin
0	1	1	1
1	$C_1^{1+2-1} = 2$	2	3
2	$C_2^{2+2-1} = 3$	4	6
3	$C_3^{3+2-1} = 4$	7	10

Nous reprenons le tableau d trouvé dans la section 4.5.1 :

$$d(1, :, :) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \end{bmatrix};$$

$$d(2, :, :) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix};$$

$$d(3, :, :) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Nous prenons chaque dérivée partielle définie par $d(j, :, k)$, pour $j = 1$ à 3 et $k = 1$ à 18, et nous les comparons aux combinaisons du tableau 4.4. Nous obtenons alors le tableau *NoComb* de taille 3 par 18. Nous représentons celui-ci par les tableaux 4.6 et 4.7.

Tableau 4.6 Numéro de combinaisons pour $d(j, :, k)$ des termes numéro 1 à 9.

		k								
		1	2	3	4	5	6	7	8	9
j	1	1	1	3	1	1	3	2	2	5
	2	1	3	1	2	5	2	1	3	1
	3	8	4	4	5	2	2	5	2	2

Tableau 4.7 Numéro de combinaisons pour $d(j, :, k)$ des termes numéro 10 à 18.

		k								
		10	11	12	13	14	15	16	17	18
j	1	1	1	3	2	2	5	4	4	8
	2	4	8	4	2	5	2	1	3	1
	3	3	1	1	3	1	1	3	1	1

Dans les tableaux 4.6 et 4.7, j et k indiquent respectivement le numéro des fonctions f_j et le numéro du terme résultant de la dérivée de $\rho u \cdot \nabla u$. Quant à lui, j désigne les numéros des lignes du tableau *NoComb* et k désigne les numéros des colonnes de ce même tableau. Ainsi, la donnée $NoComb(j, k)$ représente le numéro de la combinaison associée à la dérivée mixte de la fonction f_j du k -ième terme de sensibilités. Par exemple, la donnée $NoComb(2, 5)$, localisée dans le tableau 4.6, donne le numéro de la combinaison associée à la 2^{ième} fonction pour le 5^{ième} terme de sensibilités. Ici, celui-ci est 5 et réfère à la dérivée $\frac{\partial^2}{\partial \alpha_1 \partial \alpha_1}$ (voir tableau 4.4).

Cela signifie que le 5^{ième} terme de sensibilités inclus la sensibilité $\frac{\partial^2 f_2}{\partial \alpha_1 \partial \alpha_2}$ dans son produit.

Pour chaque produit de m fonctions, nous devons enregistrer dans le tableau *NoComb* les numéros de combinaisons associées à chaque fonction pour tous les termes de sensibilités résultant de la dérivée qui lui est appliquée. Dans notre problème test, les produits de fonctions impliqués dans les EDP sont tous constitués de 2 fonctions (voir (4.1) et (4.2)) à l'exception des termes sources. Nous avons donc associé les combinaisons seulement pour le cas où m vaut 2. Pour un cas tel que celui défini par les équations (3.1) et (3.2), où il y a des produits de fonctions dont le nombre m de fonctions est différent, nous devons ajuster le tableau *NoComb*.

La structure de données générique présentée dans la présente sous-section est efficace dans le contexte de la résolution des équations de sensibilités. Elle nous permet d'accéder rapidement au tableau de sensibilités des variables (solutions). Nous présentons à la sous-section 4.6 la façon dont le solveur l'utilise.

4.6 Solveur des équations de sensibilités d'ordre n

Lors de la résolution des sensibilités, nous résolvons d'abord le système d'EDP régissant les variables d'état. Puis, nous résolvons successivement tous les systèmes de sensibilités d'ordre $os = 1$ à n . Avant de passer à la résolution des systèmes d'ordre supérieur à os , il est primordial de résoudre tous les systèmes de sensibilités associés à l'ordre os et aux ordres inférieurs à os . Dans le cas du problème test 1D (voir section 4.1), les deux variables d'écoulement u et T correspondent aux inconnues. Ce sont donc leurs sensibilités que nous tentons de calculer.

Les EDP et leurs équations de sensibilités ne peuvent être résolues analytiquement. On adopte donc un code de calcul numérique afin d'approximer les variables inconnues ainsi que leurs sensibilités. Il existe de multiples méthodes pour résoudre numériquement les EDP dont la méthode des Différences Finies, la Méthode des Volumes Finis et la Méthode des Éléments Finis. Dans le cadre de ce projet, nous favorisons la MEF (Fortin et Garon, 2014).

Dans cette section, nous détaillons le solveur utilisé par notre approche pour résoudre les sensibilités d'ordre arbitraire n . En théorie, ce solveur peut résoudre des équations de sensibilités d'ordre élevé n , où n pourrait valoir, par exemple, 15. En raison de la généralité de notre méthode, les équations de sensibilités subissent un traitement identique à celui des équations d'état peu importe l'ordre n . Ici, nous décrivons la structure de la MEF que nous

avons implantée pour la résolution des équations de sensibilités d'ordre n . Nous présentons d'abord la formulation faible, en lien avec les équations des sensibilités d'ordre n . Puis, nous décrivons la discrétisation de cette forme faible. Cela permet d'illustrer clairement la généralité de l'algorithme que nous proposons, non seulement pour la génération, mais aussi pour la résolution des équations de sensibilités d'ordre arbitraire n .

4.6.1 Formulation faible

Pour obtenir la forme faible, on multiplie l'équation de sensibilités par une fonction test ϕ et on l'intègre sur le domaine Ω . Par exemple, pour les EDP (sensibilités d'ordre 0) du problème test 1D régi par les équations (4.1) et (4.2), nous définissons la forme intégrale par :

$$\int_{\Omega} u \cdot \nabla u \phi \, d\Omega = \int_{\Omega} (\nabla \cdot (\nu \nabla u) - \beta T) \phi \, d\Omega, \quad (4.16)$$

$$\int_{\Omega} u \cdot \nabla T \phi \, d\Omega = \int_{\Omega} \nabla \cdot (D \nabla T) \phi \, d\Omega, \quad (4.17)$$

$\forall \phi \in H_0^1(\Omega)$, puisque nous imposons des conditions de Dirichlet sur toute la frontière Γ . Il est à noter que les termes sources des équations (4.1) et (4.2) sont considérés nuls pour cet exemple d'où leur absence dans leurs formes faibles respectivement représentées par (4.16) et (4.17).

Comme le système est non-linéaire, nous nous intéressons aux résidus élémentaires \mathbf{R} , car nous utilisons la méthode de Newton pour linéariser le problème. Nous modifions donc la forme faible de manière à obtenir $R(u, T, \phi) = 0$. La forme faible de chacun des résidus R_1 et R_2 de l'exemple précédent correspond respectivement à :

$$\int_{\Omega} (u \cdot \nabla u - \nabla \cdot (\nu \nabla u) + \beta T) \phi \, d\Omega = 0, \quad (4.18)$$

$$\int_{\Omega} (u \cdot \nabla T - \nabla \cdot (D \nabla T)) \phi \, d\Omega = 0. \quad (4.19)$$

En intégrant par parties les termes du deuxième ordre apparaissant dans la forme intégrale des résidus de l'exemple précédent, nous obtenons la forme faible :

$$\int_{\Omega} (u \cdot \nabla u + \beta T) \phi \, d\Omega + \int_{\Omega} \nu \nabla u \cdot \nabla \phi \, d\Omega = 0, \quad (4.20)$$

$$\int_{\Omega} (u \cdot \nabla T) \phi \, d\Omega + \int_{\Omega} D \nabla T \cdot \nabla \phi \, d\Omega = 0. \quad (4.21)$$

En fait, nous intégrons par parties uniquement les termes $\int_{\Omega} \nabla \cdot (\nu \nabla u) \phi \, d\Omega$ et $\int_{\Omega} \nabla \cdot$

$(D\nabla T)\phi d\Omega$ des équations (4.18) et (4.19). Nous obtenons la forme explicite de leur intégration à partir du théorème de divergence (voir Fortin et Garon, 2014) :

$$\int_{\Omega} \nabla \cdot (\nu \nabla u) \phi d\Omega = \int_{\Gamma} \phi \nu \nabla u \cdot \mathbf{n} d\Gamma - \int_{\Omega} \nu \nabla u \cdot \nabla \phi d\Omega, \quad (4.22)$$

$$\int_{\Omega} \nabla \cdot (D\nabla T) \phi d\Omega = \int_{\Gamma} \phi D\nabla T \cdot \mathbf{n} d\Gamma - \int_{\Omega} D\nabla T \cdot \nabla \phi d\Omega. \quad (4.23)$$

Étant donné que $\phi \in H_0^1(\Omega)$, ϕ est nul sur le bord. Cela signifie que le terme intégré sur la frontière Γ s'annule :

$$\int_{\Omega} \nabla \cdot (\nu \nabla u) \phi d\Omega = - \int_{\Omega} \nu \nabla u \cdot \nabla \phi d\Omega, \quad (4.24)$$

$$\int_{\Omega} \nabla \cdot (D\nabla T) \phi d\Omega = - \int_{\Omega} D\nabla T \cdot \nabla \phi d\Omega. \quad (4.25)$$

Nous pouvons exprimer de façon générique les résidus du système de sensibilités $\frac{\partial^n}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ ($n = n_1 + n_2 + \dots + n_q$) du problème test (voir (4.1) et (4.2)) :

$$R_1(\Omega) = \int_{\Omega} \left(t_{1,1}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} + t_{1,2}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} - f_s^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} \right) \phi d\Omega - \quad (4.26)$$

$$\int_{\Omega} \nabla \cdot \left(t_{1,3}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} \right) \phi d\Omega = 0,$$

$$R_2(\Omega) = \int_{\Omega} \left(t_{2,1}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} - q_s^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} \right) \phi_i - \quad (4.27)$$

$$\int_{\Omega} \nabla \cdot \left(t_{2,2}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} \right) \phi d\Omega = 0,$$

où $t_{i,j}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$ correspond à l'ensemble des termes de sensibilités obtenus en appliquant la dérivée $\frac{\partial^n}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ sur le j -ième produit de m fonctions de l'équation numéro i du système étudié. Ici le cas analysé ne comporte que 2 équations, ce qui signifie que $i = 1$ à 2. Nous désignons respectivement les équation (4.1) et (4.2) par $i = 1$ et $i = 2$. L'équation numéro 1 comporte donc 3 produits de fonctions ($j = 1$ à 3), soient :

1. $t_{1,1}^{(\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)})} = u \cdot \nabla u$ est un produit de 2 fonctions, soient u et ∇u ;
2. $t_{1,2}^{(\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)})} = \beta T$ est un produit de 2 fonctions, soient β et T ;
3. $\nabla \cdot t_{1,3}^{(\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)})} = \nabla \cdot (\nu \nabla u)$ est un produit de 2 fonctions, soient ν et ∇u .

Quant à l'équation numéro 2, elle inclut 2 produits de fonctions ($j = 1$ à 2) :

1. $t_{2,1}^{(\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)})} = u \cdot \nabla T$ est un produit de 2 fonctions, soient u et ∇T ;
2. $\nabla \cdot t_{2,2}^{(\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)})} = \nabla \cdot (D \nabla T)$ est un produit de 2 fonctions, soient D et ∇T .

Notez que nous considérons les termes sources f_s et q_s par leur notation habituelle plutôt que par $t_{i,j}$.

À titre d'exemple, nous écrivons les résidus pour la dérivée $\frac{\partial^2}{\partial \alpha_1^2}$ appliquée aux EDP (4.1) et (4.2). Le résidu de l'équation de sensibilités de l'équation (4.1) est :

$$\int_{\Omega} \left(t_{1,1}^{(\alpha_1^{(2)})} + t_{1,2}^{(\alpha_1^{(2)})} - f_s^{(\alpha_1^{(2)})} \right) \phi \, d\Omega + \int_{\Omega} \left(t_{1,3}^{(\alpha_1^{(2)})} \right) \cdot \nabla \phi \, d\Omega = 0. \quad (4.28)$$

Les dérivées $\frac{\partial^2}{\partial \alpha_1^2}$ de chacun des termes de l'équation (4.1) sont :

$$t_{1,1}^{(\alpha_1^{(2)})} = u^{(\alpha_1^{(2)})} \cdot \nabla u^{(\alpha_1^{(0)})} + 2u^{(\alpha_1^{(1)})} \cdot \nabla u^{(\alpha_1^{(1)})} + u^{(\alpha_1^{(0)})} \cdot \nabla u^{(\alpha_1^{(2)})}, \quad (4.29)$$

$$t_{1,2}^{(\alpha_1^{(2)})} = \beta^{(\alpha_1^{(2)})} T^{(\alpha_1^{(0)})} + 2\beta^{(\alpha_1^{(1)})} T^{(\alpha_1^{(1)})} + \beta^{(\alpha_1^{(0)})} T^{(\alpha_1^{(2)})}, \quad (4.30)$$

$$t_{1,3}^{(\alpha_1^{(2)})} = \nu^{(\alpha_1^{(2)})} \nabla u^{(\alpha_1^{(0)})} + 2\nu^{(\alpha_1^{(1)})} \nabla u^{(\alpha_1^{(1)})} + \nu^{(\alpha_1^{(0)})} \nabla u^{(\alpha_1^{(2)})}, \quad (4.31)$$

$$f_s^{(\alpha_1^{(2)})} = f_s^{(\alpha_1^{(2)})}. \quad (4.32)$$

Nous obtenons la forme explicite de (4.28) en substituant les termes $t_{1,j}^{(\alpha_1^{(2)})}$ par leurs dérivées respectives ($j = 1$ à 3).

Pour l'équation de sensibilités de l'équation (4.2), le principe reste le même. Le résidu de l'équation de sensibilités d'ordre 2 pour l'équation (4.2) est donné par :

$$\int_{\Omega} \left(t_{2,1}^{(\alpha_1^{(2)})} - q_s^{(\alpha_1^{(2)})} \right) \phi \, d\Omega + \int_{\Omega} \left(t_{2,2}^{(\alpha_1^{(2)})} \right) \cdot \nabla \phi \, d\Omega = 0. \quad (4.33)$$

Les dérivées d'ordre 2 par rapport au paramètre α_1 , pour chacun des termes, sont :

$$t_{2,1}^{(\alpha_1^{(2)})} = u^{(\alpha_1^{(2)})} \cdot \nabla T^{(\alpha_1^{(0)})} + 2u^{(\alpha_1^{(1)})} \cdot \nabla T^{(\alpha_1^{(1)})} + u^{(\alpha_1^{(0)})} \cdot \nabla T^{(\alpha_1^{(2)})},$$

$$\begin{aligned} t_{2,2}^{(\alpha_1^{(2)})} &= D^{(\alpha_1^{(2)})} \nabla T^{(\alpha_1^{(0)})} + 2D^{(\alpha_1^{(1)})} \nabla T^{(\alpha_1^{(1)})} + D^{(\alpha_1^{(0)})} \nabla T^{(\alpha_1^{(2)})}, \\ q_s^{(\alpha_1^{(2)})} &= q_s^{(\alpha_1^{(2)})}. \end{aligned}$$

Pour obtenir la forme explicite de l'équation (4.33), il suffit de remplacer les termes $t_{2,j}$ par leurs dérivées explicites ($j = 1$ à 2).

Les fonctions des termes de ces 2 équations dépendent des paramètres $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_q\}$. Comme le projet ne considère que les paramètres de valeurs, les coordonnées spatiales ne dépendent pas des paramètres. Le gradient n'est donc pas affecté par les variations des paramètres $\boldsymbol{\alpha}$. Ainsi, le gradient étant indépendant des q paramètres α_i , il s'ensuit que l'intégration par parties pour les sensibilités d'ordre n se fait de la même façon que celle du cas de base des équations (4.24) et (4.25). La forme faible des sensibilités d'ordre n des termes $\nabla \cdot (\nu \nabla u)$ et $\nabla \cdot (D \nabla T)$ s'exprime donc de la façon suivante :

$$\int_{\Omega} \frac{\partial^n (\nabla \cdot (\nu \nabla u))}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}} \phi \, d\Omega = - \int_{\Omega} \sum_{k=1}^s c(k) \nu^{(NoComb(1,k))} \nabla u^{(NoComb(2,k))} \cdot \nabla \phi \, d\Omega, \quad (4.34)$$

$$\int_{\Omega} \frac{\partial^{(n)} (\nabla \cdot (D \nabla T))}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}} \phi \, d\Omega = - \int_{\Omega} \sum_{k=1}^s c(k) D^{(NoComb(1,k))} \nabla T^{(NoComb(2,k))} \cdot \nabla \phi \, d\Omega, \quad (4.35)$$

où la limite s de l'opérateur de sommation est le nombre de termes de sensibilités découlant de la dérivée, $NoComb(1, k)$ et $NoComb(2, k)$ sont les numéros de combinaisons respectivement associées aux dérivées de la première fonction et de la seconde fonction du k -ième terme résultant de la dérivée d'ordre n de $\nabla \cdot (\nu \nabla u)$ et de $\nabla \cdot (D \nabla T)$ (se référer à la la section 4.5.2).

Reprenons la formulation générale des dérivées de produits de fonctions que nous avons adaptée pour notre algorithme générique (voir équation (4.14)). Nous pouvons alors écrire les résidus $R_1(\Omega)$ et $R_2(\Omega)$ (voir (4.26) et (4.27)) de la façon suivante :

$$\begin{aligned} &\int_{\Omega} \left(\sum_{k=1}^s c(k) u^{(NoComb(1,k))} \cdot \nabla u^{(NoComb(2,k))} + \sum_{k=1}^s c(k) \beta^{(NoComb(1,k))} T^{(NoComb(2,k))} - f_s^{(os)} \right) \phi \, d\Omega + \\ &\int_{\Omega} \sum_{k=1}^s c(k) D^{(NoComb(1,k))} \nabla T^{(NoComb(2,k))} \cdot \nabla \phi \, d\Omega = 0, \end{aligned} \quad (4.36)$$

$$\int_{\Omega} \left(\sum_{k=1}^s c(k) u^{(NoComb(1,k))} \cdot \nabla T^{(NoComb(2,k))} - q_s^{(os)} \right) \phi \, d\Omega + \int_{\Omega} \sum_{k=1}^s c(k) D^{(NoComb(1,k))} \nabla T^{(NoComb(2,k))} \cdot \nabla \phi \, d\Omega = 0, \quad (4.37)$$

où $c(k)$ est le coefficient devant le k -ième terme découlant de la dérivée $\frac{\partial^n}{\partial \alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}$ appliquée au produit de fonctions, la limite s de l'opérateur de sommation est le nombre total de termes de sensibilités pour un produit de fonctions particulier, $(NoComb(j, k))$ est la combinaison associée à la dérivée de la j -ième fonction du k -ième terme de sensibilités et (os) est la combinaison associée au système de sensibilités (i.e. : $\frac{\partial^n}{\partial \alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}$).

4.6.2 Discrétisation

Pour la discrétisation du problème, l'algorithme accède à un fichier dans lequel l'utilisateur fournit les données requises : le type de l'élément fini (l'ordre p du polynôme), le nombre nel d'éléments et le facteur de distribution des éléments. Par exemple, nous divisons notre domaine en 3 éléments ($nel = 3$) : K_1 , K_2 et K_3 . La discrétisation obtenue pour des éléments de types P_1 , P_2 , P_3 et P_4 (i.e. : $p = 1, 2, 3$ et 4) est alors illustrée par le biais des figures 4.2, 4.3, 4.4 et 4.5.

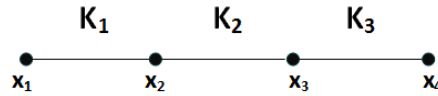


Figure 4.2 Discrétisation avec 3 éléments de type P_1 ($p = 1$).

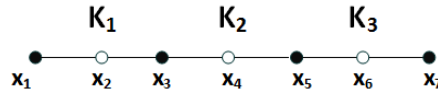


Figure 4.3 Discrétisation avec 3 éléments de type P_2 ($p = 2$).

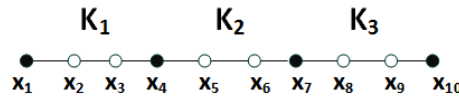


Figure 4.4 Discrétisation avec 3 éléments de type P_3 ($p = 3$).

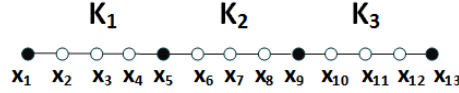


Figure 4.5 Discrétisation avec 3 éléments de type P_4 ($p = 4$).

En discrétisant la forme faible des équations des sensibilités d'ordre n , avec la Méthode des Éléments Finis nous obtenons un système discret de la forme :

$$A^K \mathbf{U}^K = \mathbf{F}^K + \mathbf{S}^K, \quad (4.38)$$

sur l'ensemble des éléments K , où $K = [x_1^K, x_2^K]$. A^K est la matrice élémentaire, \mathbf{U}^K est le vecteur des degrés de liberté élémentaires et $(\mathbf{F}^K + \mathbf{S}^K)$ est le vecteur des sollicitations élémentaires.

De même, le résidu considéré pour la méthode de Newton est discrétisé sur l'ensemble des éléments K comme suit :

$$\mathbf{R}(\mathbf{U}^K) = A^K \mathbf{U}^K - \mathbf{F}^K - \mathbf{S}^K = 0. \quad (4.39)$$

Comme les membres de gauche (i.e : A^K) des équations de sensibilités sont identiques pour tout ordre $n \in \mathbb{N}$, il est possible de calculer la matrice A et de la factoriser une fois pour toutes et de réutiliser la factorisation pour chaque membre de gauche.

En s'appuyant sur les équations (4.36) et (4.37), nous pouvons écrire les formules des résidus élémentaires $Res_i^K(1)$ et $Res_i^K(2)$:

$$\begin{aligned} & \int_{x_1^K}^{x_2^K} \left(\sum_{k=1}^s c(k) u^{(NoComb(1,k))} \cdot \nabla u^{(NoComb(2,k))} + \sum_{k=1}^s c(k) \beta^{(NoComb(1,k))} T^{(NoComb(2,k))} \right) \phi_i^K dx \\ & - \int_{x_1^K}^{x_2^K} f_s^{(os)} \phi_i^K dx + \int_{x_1^K}^{x_2^K} \sum_{k=1}^s c(k) D^{(NoComb(1,k))} \nabla T^{(NoComb(2,k))} \cdot \nabla \phi_i^K dx = 0, \end{aligned} \quad (4.40)$$

$$\begin{aligned} & \int_{x_1^K}^{x_2^K} \left(\sum_{k=1}^s c(k) u^{(NoComb(1,k))} \cdot \nabla T^{(NoComb(2,k))} - q_s^{(os)} \right) \phi_i^K dx + \\ & \int_{x_1^K}^{x_2^K} \sum_{k=1}^s c(k) D^{(NoComb(1,k))} \nabla T^{(NoComb(2,k))} \cdot \nabla \phi_i^K dx = 0, \end{aligned} \quad (4.41)$$

pour $i = 1$ à ndp , où ndp est le nombre total de noeuds d'un élément fini. Les fonctions ϕ_i^K sont les fonctions d'interpolation de l'élément K qui dépendent du type de l'élément fini (voir Fortin et Garon, 2014).

Interpolation de la solution

L'interpolation de la solution est directement liée au type de l'élément choisi par l'utilisateur. En effet, la solution de u s'écrit sous la forme :

$$u\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)(x_k) = \sum_{j=1}^{ndp} u\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)(x_j)\phi_j(x_k), \quad (4.42)$$

$$\frac{du\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)(x_k)}{dx} = \sum_{j=1}^{ndp} \frac{du\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)(x_j)}{dx} \frac{d\phi_j(x_k)}{dx}, \quad (4.43)$$

où $u\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)(x_k)$ est la solution de la sensibilité $\frac{\partial^n}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\alpha_q^{n_q}}$ de u en x_k avec $n = n_1 + n_2 + \dots + n_q$ et $\phi_j(x_k)$ est la fonction d'interpolation j en x_k . Les valeurs représentées par $u\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)(x_j)$ correspondent aux valeurs des degrés de liberté de l'élément K (MEF).

Nous reprenons les équations précédentes et nous les écrivons sous la notation suivante pour éviter d'alourdir les équations :

$$S_u^{(noSyst)}(x_k) = \sum_{j=1}^{ndp} S_u^{(noSyst)}(x_j)\phi_j(x_k), \quad (4.44)$$

$$\frac{dS_u^{(noSyst)}(x_k)}{dx} = \sum_{j=1}^{ndp} \frac{dS_u^{(noSyst)}(x_j)}{dx} \frac{d\phi_j(x_k)}{dx}, \quad (4.45)$$

où nous remplaçons $u\left(\alpha_1^{(n_1)}\alpha_2^{(n_2)}\dots\alpha_q^{(n_q)}\right)$ par $S_u^{(noSyst)}$. L'exposant $(noSyst)$ affecté à S_u est le numéro de la combinaison associée au système $\frac{\partial^n}{\partial\alpha_1^{n_1}\partial\alpha_2^{n_2}\dots\alpha_q^{n_q}}$.

Il en va de même pour la solution de T :

$$S_T^{(noSyst)}(x_k) = \sum_{j=1}^{ndp} S_T^{(noSyst)}(x_j)\phi_j(x_k), \quad (4.46)$$

$$\frac{dS_T^{(noSyst)}(x_k)}{dx} = \sum_{j=1}^{ndp} \frac{dS_T^{(noSyst)}(x_j)}{dx} \frac{d\phi_j(x_k)}{dx}, \quad (4.47)$$

où $S_T^{(noSyst)}$ est la sensibilité de T du système numéro *noSyst* que nous résolvons.

4.6.3 Résolution par la méthode de Newton

Comme la plupart des systèmes d'équations différentielles en mécanique des fluides, le système d'EDP que nous traitons est non-linéaire. Nous le linéarisons par la méthode de Newton. Même si les équations de sensibilités d'ordre 1 à n , pour $n > 0$, sont linéaires, nous utilisons néanmoins la méthode de Newton pour la résolution des équations des sensibilités d'ordre 0 à n . Cela permet de préserver la généralité de notre algorithme.

Les systèmes de sensibilités étant linéaires, leur résolution requiert peu d'itérations comparativement au système non-linéaire pour l'écoulement. Généralement, la résolution des systèmes de sensibilités nécessite 2 itérations pour atteindre la convergence. Cependant, cela dépend du critère de convergence.

La méthode de Newton cherche à résoudre le système linéaire :

$$J_R(\mathbf{U}^k)[\delta\mathbf{U}] = -\mathbf{R}(\mathbf{U}^k), \quad (4.48)$$

où \mathbf{U}^k est le vecteur des degrés de liberté à l'itération k , J_R est le jacobien associé au résidu, $\delta\mathbf{U}$ est le vecteur des corrections et $\mathbf{R}(\mathbf{U}^k)$ est le vecteur des résidus.

Nous déterminons une approximation initiale \mathbf{U}^0 de la solution. Ensuite, on résout le problème linéaire décrit par l'équation (4.48) pour obtenir la correction $\delta\mathbf{U}$. Cela nous permet alors de corriger la solution de l'itération k avec $\delta\mathbf{U}$:

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \delta\mathbf{U}. \quad (4.49)$$

Nous vérifions alors si nous avons atteint le critère de convergence :

$$\frac{\|\delta\mathbf{U}\|}{\|\mathbf{U}^{k+1}\|} < \epsilon \text{ et } \|\mathbf{R}(\mathbf{U}^{k+1})\| < \epsilon. \quad (4.50)$$

Si la convergence n'est pas atteinte, on procède à une nouvelle itération en utilisant la solution mise à jour \mathbf{U}^{k+1} comme nouvelle condition initiale. Les valeurs des degrés de liberté U_j ($j = 1$ à *nddl*) sont mises à jour dans le tableau u à chaque itération. On effectue de nouvelles itérations jusqu'à ce qu'il y ait convergence ou encore que le nombre d'itérations

maximal soit atteint.

Le calcul de la matrice jacobienne peut se faire de façons numérique ou analytique. Évidemment, il est préférable d'avoir recours au jacobien analytique, puisqu'il considère les dérivées exactes. Nous abordons plus en détail la façon d'évaluer les jacobiens numérique et analytique dans les sous-sections 4.6.4 et 4.6.5.

4.6.4 Jacobien numérique

Le Jacobien J_R correspond à la matrice des dérivées partielles des résidus :

$$J_R(\mathbf{U}^k) = \left[\frac{\partial R_i}{\partial U_j} \right] (\mathbf{U}^k), \quad (4.51)$$

où i et j varient de 1 à $nddl$. Nous déterminons la quantité $nddl$ de degrés de liberté en considérant le nombre de variables inconnues (ici, u et T) multiplié par le nombre de degrés de liberté par variable. Dans l'équation (4.51), $\frac{\partial R_i}{\partial U_j}$ correspond à la dérivée partielle du résidu R_i par rapport à la variable associée au degré de liberté numéro j . Cette dernière correspond soit à $S_u^{(noSyst)}(x_j)$ ou à $S_T^{(noSyst)}(x_j)$ (sensibilités du système courant dont le numéro de combinaison qui lui est associé est $noSyst$) avec x_j qui est la coordonnée du noeud physique associé au degré de liberté j . Par ailleurs, la donnée $\frac{\partial R_i}{\partial U_j}$ est située dans la j -ième colonne de la i -ième ligne de la matrice J_R . Par exemple, $\frac{\partial R_5}{\partial U_3}$ correspond à la valeur située à la cinquième ligne troisième colonne dans la matrice J_R .

Dans le cas du jacobien numérique, nous approximations les dérivées $\frac{\partial R_i}{\partial U_j}$ par des Différences Finies décentrées comme suit :

$$\frac{\partial R_i}{\partial U_j} = \frac{R_i(U_1^k, U_2^k, \dots, U_j^k + \delta U_j, \dots, U_N^k) - R_i(U_1^k, U_2^k, \dots, U_j^k, \dots, U_N^k)}{\delta U_j} + O(\delta U_j), \quad (4.52)$$

où U_j est la sensibilité du système étudié pour le degré de liberté j , l'indice N est le nombre total de degrés de liberté ($nddl$) et l'exposant k indique le numéro d'itération.

Pour calculer la dérivée $\frac{\partial R_i}{\partial U_j}$, il suffit alors d'utiliser les valeurs de résidus calculées avec la solution \mathbf{U}^k et de calculer à nouveau le résidu en perturbant légèrement la valeur nodale de U_j^k en lui ajoutant δU_j . Pour un résidu spécifique, nous perturbons chacune des valeurs nodales du problème afin d'obtenir ses dérivées par rapport à toutes les valeurs nodales U_j .

Nous répétons ce processus sur tous les résidus afin d'obtenir les dérivées $\frac{\partial R_i}{\partial U_j} \forall i, j = 1$ à $nddl$. Selon le système étudié, nous prenons les valeurs U_j^k dans le tableau tridimensionnel u (voir 4.5.2).

4.6.5 Jacobien analytique

Pour le jacobien analytique, nous évaluons de manière exacte la matrice de l'équation (4.51). Nous cherchons alors à obtenir les expressions analytiques définissant les composantes de la matrice A . Les vecteurs \mathbf{F} et \mathbf{S} , les membres de droite de la forme faible, sont indépendants de \mathbf{U} . Leurs dérivées par rapport aux valeurs U_j sont donc nulles. Nous déterminons les composantes de la matrice A par le biais du membre de gauche de la formulation variationnelle des équations de sensibilités dont la structure demeure inchangée pour un système de sensibilités d'ordre $n \in \mathbb{N}$ avec un nombre q quelconque de paramètres de valeur. On suppose que ce système est associé au numéro de combinaison $noSyst$ du tableau $CombSens$. Par exemple, si nous prenons les équations du problème 1D de la section 4.1, les membres de gauche obtenus sont :

$$\int_{\Omega} (S_u^{(noSyst)} \cdot \nabla u + u \cdot \nabla S_u^{(noSyst)} + \beta S_T^{(noSyst)}) \phi d\Omega + \int_{\Omega} (\nu \nabla S_u^{(noSyst)}) \cdot \nabla \phi d\Omega, \quad (4.53)$$

$$\int_{\Omega} (S_u^{(noSyst)} \cdot \nabla T + u \cdot \nabla S_T^{(noSyst)}) \phi d\Omega + \int_{\Omega} (D \nabla S_T^{(noSyst)}) \cdot \nabla \phi d\Omega, \quad (4.54)$$

où $noSyst \in [1, 2, \dots, r]$ et représente le numéro de la combinaison associé au système de sensibilités étudié. Nous savons que la matrice A^K de l'équation (4.39) est identique peu importe l'ordre des équations des sensibilités étudiés et que seule cette partie de l'équation dépend de \mathbf{U} (sensibilités d'ordre n). Nous en déduisons que les formules analytiques pour les composantes du jacobien analytique ne changent pas avec l'ordre des sensibilités.

Le jacobien analytique de l'élément K est obtenu par :

$$J_R^K = \begin{bmatrix} J^K(u, S_u^{(noSyst)}(x_j)) & J^K(u, S_T^{(noSyst)}(x_j)) \\ J^K(T, S_u^{(noSyst)}(x_j)) & J^K(T, S_T^{(noSyst)}(x_j)) \end{bmatrix}. \quad (4.55)$$

Nous avons écrit la matrice (4.55) sous forme compacte. Voici quelques indicatifs concernant la notation que nous utilisons :

- $S_u^{(noSyst)}(x_j)$ est la sensibilité de u du système numéro $noSyst$ en x_j ;
- $S_T^{(noSyst)}(x_j)$ est la sensibilité de T du système numéro $noSyst$ en x_j ;
- $J^K(u, S_u^{(noSyst)}(x_j))$ est la dérivée du résidu de l'équation de sensibilités de u par

- rapport à la sensibilité $S_u^{(noSyst)}(x_j)$;
- $J^K(u, S_T^{(noSyst)}(x_j))$ est la dérivée du résidu de l'équation de sensibilités de u par rapport à la sensibilité $S_T^{(noSyst)}(x_j)$;
 - $J^K(T, S_u^{(noSyst)}(x_j))$ est la dérivée du résidu de l'équation de sensibilités de T par rapport à la sensibilité $S_u^{(noSyst)}(x_j)$;
 - $J^K(T, S_T^{(noSyst)}(x_j))$ est la dérivée du résidu de l'équation de sensibilités de T par rapport à la sensibilité $S_T^{(noSyst)}(x_j)$.

Afin de déterminer la forme générale des éléments de la matrice jacobienne élémentaire, nous prenons les intégrales numériques des résidus élémentaires (voir les équations (F.4) et (F.6) de l'annexe E). Nous y remplaçons $S_u^{(noSyst)}(x_k)$ et $S_T^{(noSyst)}(x_k)$ par leurs solutions interpolées respectivement données par les équations (4.45) et (4.47). Pour $J^K(u, S_u^{(noSyst)}(x_j))$ nous obtenons :

$$\begin{aligned} J^K(u, S_u^{(noSyst)}(x_j)) &= \frac{\partial \text{(F.4)}}{\partial S_u^{(noSyst)}(x_j)} \\ &= \sum_{k=1}^{npg} \left[\left(\frac{du_k}{dx} \phi_j(x_k) + u_k \frac{d\phi_j(x_k)}{dx} \right) \phi_i(x_k) + \nu_k \frac{d\phi_j(x_k)}{dx} \frac{d\phi_i(x_k)}{dx} \right] w_k. \end{aligned} \quad (4.56)$$

De même, les composantes $J^K(u, S_T^{(noSyst)}(x_j))$ correspondent à :

$$J^K(u, S_T^{(noSyst)}(x_j)) = \frac{\partial \text{(F.4)}}{\partial S_T^{(noSyst)}(x_j)} = \sum_{k=1}^{npg} [\beta_k \phi_j(x_k) \phi_i(x_k)] w_k. \quad (4.57)$$

Pour $J^K(T, S_u^{(noSyst)}(x_j))$, nous avons :

$$J^K(T, S_u^{(noSyst)}(x_j)) = \frac{\partial \text{(F.6)}}{\partial S_u^{(noSyst)}(x_j)} = \sum_{k=1}^{npg} \left[\frac{dT_k}{dx} \phi_j(x_k) \phi_i(x_k) \right] w_k. \quad (4.58)$$

Enfin, $J^K(T, S_T^{(noSyst)}(x_j))$ est défini par :

$$J^K(T, S_T^{(noSyst)}(x_j)) = \frac{\partial \text{(F.6)}}{\partial S_T^{(noSyst)}(x_j)} = \sum_{k=1}^{npg} \left[u_k \frac{d\phi_j(x_k)}{dx} \phi_i(x_k) + D_k \frac{d\phi_j(x_k)}{dx} \frac{d\phi_i(x_k)}{dx} \right] w_k, \quad (4.59)$$

où $i, j = 1$ à ndp (ndp est le nombre total de noeud par élément), l'indice k d'une fonction donnée est la valeur de cette fonction en x_k , w_k est le poids d'intégration en x_k et ϕ_i est la

fonction d'interpolation numéro i .

4.7 Généricité de l'algorithme des sensibilités

L'algorithme implémenté pour la génération et la résolution des équations de sensibilités continues à l'ordre n est générique, puisqu'il requiert uniquement la connaissance des données suivantes :

- le nombre m de fonctions pour chacun des termes traités ;
- le nombre q de paramètres ;
- l'ordre n des sensibilités à calculer.

Évidemment, l'utilisateur doit également définir les paramètres de la discrétisation, c'est-à-dire le nombre d'éléments finis, le type de ces éléments et le nombre de points de Gauss pour l'intégration numérique. Cette méthode peut donc être utilisée avec n'importe quel système d'équations différentielles dont on souhaite calculer les sensibilités d'ordre n avec q paramètres. Nous portons un intérêt à cette méthode en vue de l'appliquer éventuellement aux équations de Navier-Stokes en régime laminaire et incompressible. Dans le projet actuel, le fait de l'appliquer sur un système d'équations 1D permet de réduire la complexité du problème et de simplifier l'implémentation de la MEF, puisqu'il y a moins d'équations à gérer dans le système et que nous n'avons pas à nous préoccuper du tenseur de contraintes visqueuses.

CHAPITRE 5 VÉRIFICATION ET VALIDATION

Dans ce chapitre, nous présentons les étapes nécessaires à la vérification du code de calcul des sensibilités et les résultats que nous avons obtenus. Nous subdivisons en trois phases la méthodologie pour la validation et la vérification du code de calcul, soient :

1. La vérification du code par le biais de la Méthode des Solutions Manufacturées (voir Roache, 1998, 2002),
2. La validation du code par la comparaison des sensibilités numériques avec celles obtenues par Différences Finies,
3. La comparaison des solutions numériques avec les solutions extrapolées par le développement de la série de Taylor à l'ordre n .

Nous réalisons ces étapes de vérification sur le problème unidimensionnel décrit à la section 5.1. Nous détaillons ensuite la MSM dans la section 5.2. Puis, nous illustrons et comparons les résultats obtenus pour la vérification du code par le biais de la MSM. Dans la section 5.3, nous comparons les sensibilités d'ordre n calculées par notre algorithme à celles évaluées par la méthode des Différences Finies. On suppose que notre méthode fonctionne correctement à l'ordre 0, car cette solution apparaît dans les équations de sensibilités par Différences Finies. La comparaison entre les résultats de la méthode des DF et ceux de notre MESC a pour but de s'assurer que cette dernière approxime réellement les sensibilités. Finalement, dans la dernière section, nous appliquons la méthode des sensibilités d'ordre n à des problèmes perturbés, c'est-à-dire pour des valeurs différentes des paramètres. Nous comparons alors les prédictions par la série de Taylor dans l'espace des paramètres aux solutions obtenues par simulations pour les valeurs perturbées des paramètres.

5.1 Problème test

Afin de réaliser les étapes de validation et de vérification, nous reprenons le problème non-linéaire unidimensionnel de la section 4.1. Dans un premier temps, nous considérons le cas à 1 paramètre ($q = 1$), soit α_1 . Nous définissons les propriétés physiques dépendantes du paramètre α_1 comme suit :

$$\nu = 0.1e^{-\alpha_1 x}, \quad (5.1)$$

$$\beta = 0.01 \cos(-\alpha_1 x), \quad (5.2)$$

$$D = 0.01e^{\alpha_1 x}, \quad (5.3)$$

où $x \in [0, 1]$ et $\alpha_1 = \frac{\pi}{2}$. On remarque que ces propriétés varient dans l'intervalle $[0, 1]$.

Puis, nous procédons aux vérifications préliminaires (*i.e.* : MSM) pour $q > 1$. Nous reprenons le problème non-linéaire unidimensionnel et nous définissons les propriétés matérielles en fonction de 2 paramètres ($q = 2$), soient α_1 et α_2 :

$$\nu = 0.1e^{-\alpha_1\alpha_2}, \quad (5.4)$$

$$\beta = 0.01 \cos(-\alpha_1\alpha_2), \quad (5.5)$$

$$D = 0.01e^{\alpha_1\alpha_2}, \quad (5.6)$$

où le domaine du problème est $x \in [0, 1]$ et les paramètres α_1 et α_2 valent respectivement $\frac{\pi}{2}$ et $\frac{\pi}{4}$.

5.2 Vérification du code

La première étape de vérification consiste à s'assurer qu'il y a convergence vers la solution exacte dans l'espace $H^1(\Omega)$ (voir annexe G) avec le raffinement du maillage. Cela permet de vérifier que nous avons implémenté adéquatement notre méthode des sensibilités. Nous effectuons une analyse de convergence avec la MSM. Il s'agit alors de concevoir une solution analytique et de l'injecter dans le problème. Nous calculons l'erreur exacte entre la solution analytique connue et la solution numérique obtenue par notre méthode. Cela nous permet de vérifier que $\lim_{\Delta h \rightarrow 0} \|u_{exa} - u_h\|_{1,\Omega} = 0$ et $\lim_{\Delta h \rightarrow 0} \|T_{exa} - T_h\|_{1,\Omega} = 0$, où u_{exa} et T_{exa} sont les solutions analytiques de u et T et u_h ainsi que T_h sont les solutions éléments finis de u et T . Nous utilisons la norme $\| \cdot \|_{1,\Omega}$ pour calculer l'erreur exacte de la solution numérique :

$$\|u_{exa} - u_h\|_{1,\Omega}^2 = \int_{\Omega} (u_{exa} - u_h)^2 d\Omega + \int_{\Omega} \left(\frac{\partial(u_{exa} - u_h)}{\partial x} \right)^2 d\Omega, \quad (5.7)$$

$$\|T_{exa} - T_h\|_{1,\Omega}^2 = \int_{\Omega} (T_{exa} - T_h)^2 d\Omega + \int_{\Omega} \left(\frac{\partial(T_{exa} - T_h)}{\partial x} \right)^2 d\Omega. \quad (5.8)$$

Nous procédons de la même façon pour évaluer l'erreur des sensibilités :

$$\begin{aligned}
& \left\| \left(\frac{\partial u_{exa}}{\partial x} \right)_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)} - u_h \right\|_{1,\Omega}^2 = \\
& \int_{\Omega} \left(\left(\frac{\partial u_{exa}}{\partial x} \right)_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)} - u_h \right)^2 d\Omega + \\
& \int_{\Omega} \left(\frac{\partial \left(\frac{\partial u_{exa}}{\partial x} \right)_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)}}{\partial x} \right)^2 d\Omega,
\end{aligned} \tag{5.9}$$

$$\begin{aligned}
& \left\| \left(\frac{\partial T_{exa}}{\partial x} \right)_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)} - T_h \right\|_{1,\Omega}^2 = \\
& \int_{\Omega} \left(\left(\frac{\partial T_{exa}}{\partial x} \right)_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)} - T_h \right)^2 d\Omega + \\
& \int_{\Omega} \left(\frac{\partial \left(\frac{\partial T_{exa}}{\partial x} \right)_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)}}{\partial x} \right)^2 d\Omega,
\end{aligned} \tag{5.10}$$

où $u_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)}$ est la sensibilité de u et $T_{\left(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)} \right)}$ est la sensibilité de T pour le système où on applique la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$.

La MSM est un exercice de vérification purement mathématique. Il n'est donc pas nécessaire que la solution analytique que nous choisissons soit réaliste. Toutefois, celle-ci doit être suffisamment complexe pour faire intervenir tous les termes dans les équations du problème. Cela permet de tester toutes les parties du code de calcul. Ici, pour vérifier notre code générique, la solution manufacturée pour le problème à 1 paramètre ($q = 1$) est :

$$u_{exa}(x; \alpha_1) = \sin(\alpha_1 x), \tag{5.11}$$

$$T_{exa}(x; \alpha_1) = \cos(\alpha_1 x), \tag{5.12}$$

tandis que la solution manufacturée pour le problème avec 2 paramètres ($q = 2$) est :

$$u_{exa}(x; \alpha_1, \alpha_2) = \sin(\alpha_1 \alpha_2 x), \tag{5.13}$$

$$T_{exa}(x; \alpha_1, \alpha_2) = \cos(\alpha_1 \alpha_2 x), \quad (5.14)$$

où u_{exa} est la solution analytique de u et T_{exa} est la solution analytique de T . Rappelons que α_1 est égal à $\frac{\pi}{2}$, pour ces 2 problèmes, et que α_2 vaut $\frac{\pi}{4}$ pour le problème à 2 paramètres.

Les solutions choisies pour u et T ne satisfont pas les équations différentielles. Afin de satisfaire les équations (4.1) et (4.2), nous leur ajoutons des termes sources artificiels, soient respectivement f_s et q_s . Nous déterminons les fonctions appropriées pour f_s et q_s en injectant les fonctions u_{exa} et T_{exa} dans les équations concernées. Pour le problème test avec $q = 1$, nous avons :

$$\begin{aligned} f_s(x; \alpha_1) = & \alpha_1 \sin(\alpha_1 x) \cos(\alpha_1 x) + 0.1 \alpha_1^2 e^{-\alpha_1 x} \cos(\alpha_1 x) \\ & + 0.1 \alpha_1^2 e^{-\alpha_1 x} \sin(\alpha_1 x) + 0.01 \cos(-\alpha_1 x) \cos(\alpha_1 x), \end{aligned} \quad (5.15)$$

$$\begin{aligned} q_s(x; \alpha_1) = & -\alpha_1 \sin^2(\alpha_1 x) + 0.01 \alpha_1^2 e^{\alpha_1 x} \sin(\alpha_1 x) \\ & + 0.01 \alpha_1^2 e^{\alpha_1 x} \cos(\alpha_1 x), \end{aligned} \quad (5.16)$$

alors que pour le problème test avec $q = 2$, nous obtenons :

$$\begin{aligned} f_s(x; \alpha_1, \alpha_2) = & \alpha_1 \alpha_2 \sin(\alpha_1 \alpha_2 x) \cos(\alpha_1 \alpha_2 x) + 0.1 (\alpha_1 \alpha_2)^2 e^{-\alpha_1 \alpha_2} \sin(\alpha_1 \alpha_2 x) \\ & + 0.01 \cos(-\alpha_1 \alpha_2) \cos(\alpha_1 \alpha_2 x), \end{aligned} \quad (5.17)$$

$$q_s(x; \alpha_1, \alpha_2) = -\alpha_1 \alpha_2 \sin^2(\alpha_1 \alpha_2 x) + 0.01 (\alpha_1 \alpha_2)^2 e^{\alpha_1 \alpha_2} \cos(\alpha_1 \alpha_2 x). \quad (5.18)$$

Le processus est similaire pour les problèmes de sensibilités d'ordre n . Nous trouvons d'abord les sensibilités de la solution manufacturée (*i.e.* : u_{exa} et T_{exa}) en dérivant la solution selon la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ du système de sensibilités étudié. Puis nous l'injectons dans l'équation de sensibilités concernée. Cela nous permet alors d'isoler les sensibilités des termes sources et de déterminer leurs fonctions. Une autre manière de calculer les sensibilités des termes sources est d'appliquer directement la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ à f_s et q_s .

Nous devons vérifier notre code éléments finis qui utilise des polynômes de degrés $p = 1$ à 5 (fait référence aux degrés des polynômes de Lagrange). Nous évaluons les solutions des sensibilités pour l'ordre $n = 0$ à 3 de chacun des 2 problèmes manufacturés comportant respectivement 1 et 2 paramètre(s). Nous résolvons systématiquement les problèmes manufacturés pour différentes valeurs de Δh qui correspond à la taille des éléments du maillage. En fait, nous augmentons graduellement le nombre d'éléments du problème. Cela permet de

vérifier que l'ordre de convergence de la MEF coïncide avec ce que la théorie prévoit (voir Fortin et Garon, 2014).

Dans le cas des sensibilités à 1 paramètre, nous illustrons par les figures 5.1 et 5.2 les résultats obtenus pour les normes d'erreur exacte (voir équations (5.9) et (5.10)) en fonction du nombre de noeuds du maillage dont la taille des éléments (*i.e.* : Δh) dépend directement. Nous représentons les courbes pour chaque type d'éléments finis que nous notons $P1$, $P2$, $P3$, $P4$ et $P5$ (*i.e.* : polynômes de degré $p = 1, 2, 3, 4$ et 5). Le lecteur intéressé peut trouver les données numériques utilisées pour tracer ces courbes dans l'annexe H. Il est aisé de remarquer que les erreurs diminuent avec le raffinement du maillage à un taux de plus en plus rapide pour des ordres de discrétisation p de plus en plus élevés. Par exemple, pour $p = 2$, l'erreur est plus importante pour un même nombre de noeuds que dans le cas où nous résolvons le même problème avec $p = 3$.

Nous vérifions également la théorie concernant l'ordre de convergence qui prévoit :

$$\|u_{exa}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} - u_h^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}\|_{1,\Omega} \simeq C(\Delta h)^p, \quad (5.19)$$

$$\|T_{exa}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} - T_h^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}\|_{1,\Omega} \simeq C(\Delta h)^p, \quad (5.20)$$

où C est une constante, $\sum_{i=1}^q n_i = n$, Δh est la taille des éléments et p est le degré des polynômes utilisés pour la discrétisation. Pour ce faire, nous utilisons les échelles logarithmiques nous permettant alors de s'apercevoir que les pentes des courbes d'erreur ont bel et bien une valeur égale à la valeur de p tel qu'espéré :

$$Pente \simeq \frac{\ln |erreur_1| - \ln |erreur_2|}{\ln |\Delta h_1| - \ln |\Delta h_2|} = p. \quad (5.21)$$

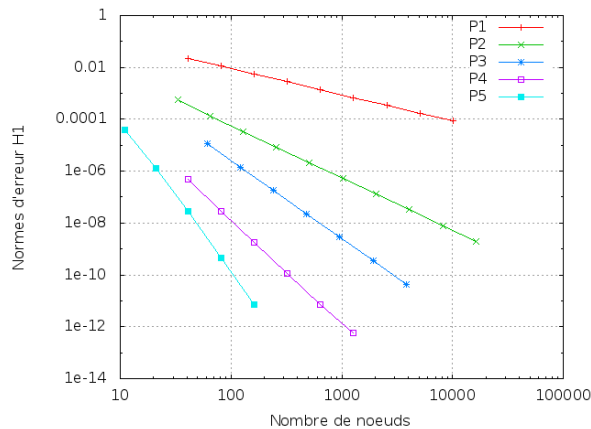
La valeur logarithmique de l'erreur correspond à :

$$\ln |erreur_i| \simeq p \ln |\Delta h_i| + \ln |C|, \quad (5.22)$$

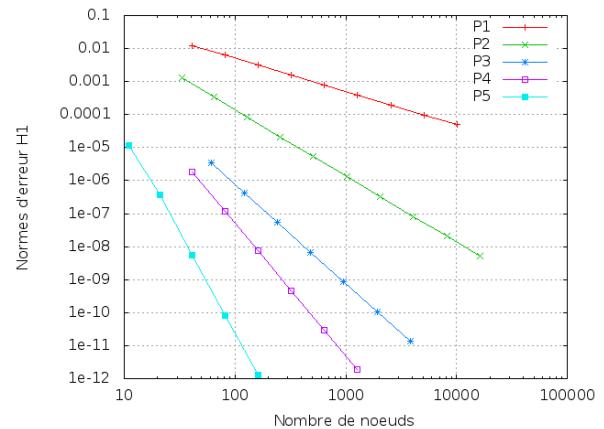
où l'indice i indique le numéro de l'erreur ou du Δh qui lui est associé. Nous calculons respectivement les erreurs pour u et T par le biais des équations (5.19) et (5.20). Nous constatons que le taux de décroissance des erreurs obtenues par la MSM est en accord avec le taux de convergence théorique pour chaque type d'élément utilisé ($p = 1$ à 5). Autrement dit, pour la résolution du problème par la MEF avec des éléments de degré p , l'ordre de convergence est

p . Par exemple, avec des éléments finis de degré 3, la solution par la MEF converge à l'ordre 3.

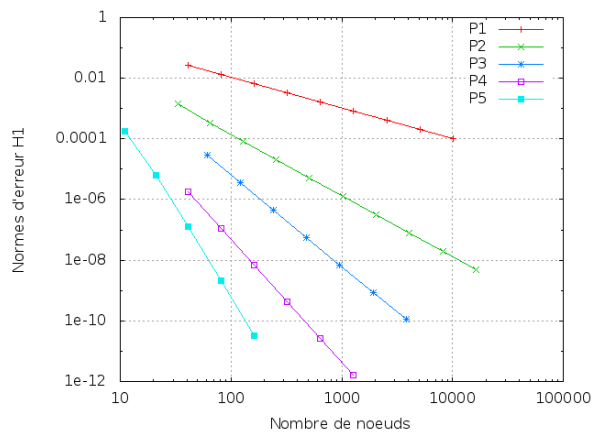
Les résultats obtenus pour le problème avec 2 paramètres sont similaires à ceux observés pour le problème à 1 paramètre. Toutefois, nous les avons répertoriés dans l'annexe I afin d'alléger la présentation. L'ensemble des résultats préliminaires obtenus démontre que l'implantation de la méthode proposée est adéquate et sans erreur. Nous pouvons donc conclure que cet exercice de vérification du code de calcul est réussi.



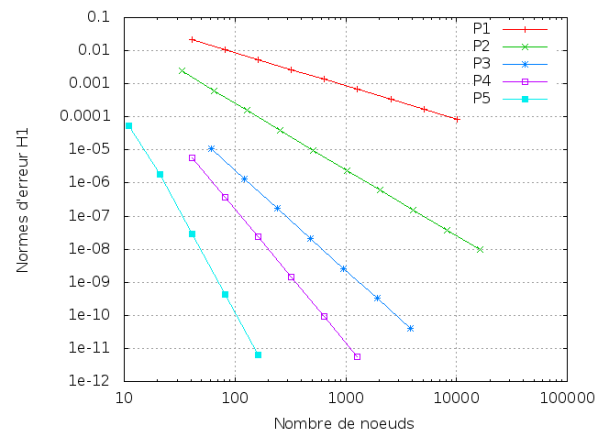
(a) Analyse de convergence pour $T(\alpha_1^{(0)})$



(b) Analyse de convergence pour $u(\alpha_1^{(0)})$.



(c) Analyse de convergence pour $T(\alpha_1^{(1)})$.



(d) Analyse de convergence pour $u(\alpha_1^{(1)})$.

Figure 5.1 Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités d'ordre n où $n = 0$ à 1.

5.3 Comparaison avec la méthode des Différences Finies

Nous consacrons cette section à la comparaison des sensibilités numériques calculées par notre code par rapport aux sensibilités évaluées par le biais de la méthode des Différences

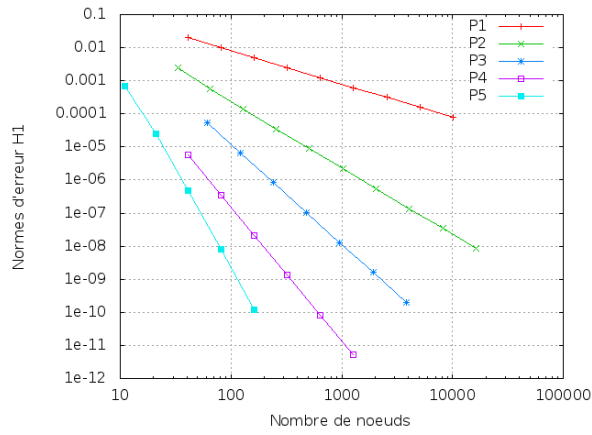
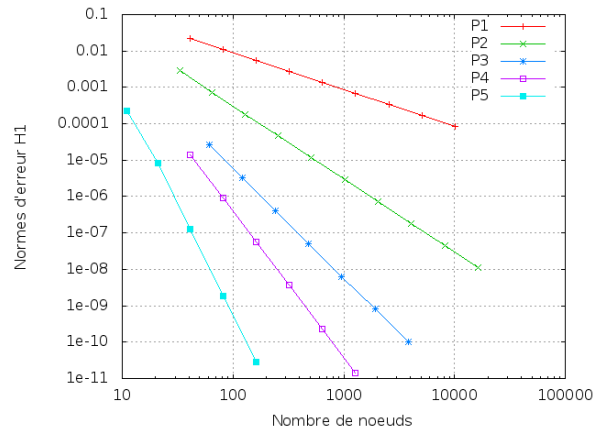
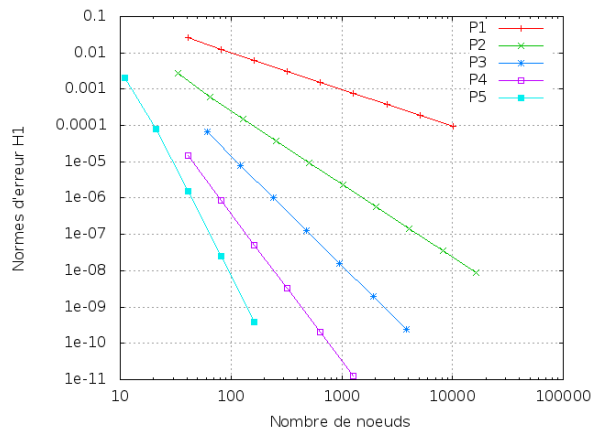
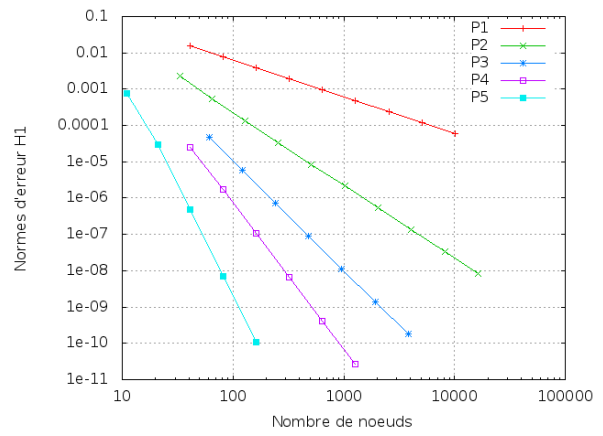
(a) Analyse de convergence pour $T(\alpha_1^{(2)})$.(b) Analyse de convergence pour $u(\alpha_1^{(2)})$.(c) Analyse de convergence pour $T(\alpha_1^{(3)})$.(d) Analyse de convergence pour $u(\alpha_1^{(3)})$.

Figure 5.2 Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités d'ordre n où $n = 2$ à 3.

Finies. Pour réaliser cette comparaison, nous considérons les sensibilités d'ordre 1 à 6 avec 1 paramètre. Ici, nous utilisons des Différences Finies centrées d'ordre 2 pour calculer les sensibilités. La formule générale des différences centrées par rapport à 1 paramètre α_1 est :

$$S_{FD}^{(\alpha_1^{(n)})}(x, \alpha_1) = \frac{S^{(\alpha_1^{(n-1)})}(x, \alpha_1 + \Delta\alpha_1) - S^{(\alpha_1^{(n-1)})}(x, \alpha_1 - \Delta\alpha_1)}{2\Delta\alpha_1} + O(\Delta\alpha_1^2), \quad (5.23)$$

où $n \in \mathbb{N}^*$, $S_{FD}^{(\alpha_1^{(n)})}$ est une sensibilité d'ordre n obtenue par DF et $S^{(\alpha_1^{(n-1)})}$ est une sensibilité numérique d'ordre $n - 1$. Afin d'approximer les sensibilités d'ordre n , on remarque qu'il est nécessaire de connaître les sensibilités d'ordre $n - 1$. Toutefois, nous pouvons réécrire les formules des DF, afin de pouvoir calculer les sensibilités d'ordre n par DF en considérant uniquement les sensibilités d'ordre 0 (*i.e.* : u et T). Par exemple, si nous prenons $u^{(\alpha_1^{(2)})}(x, \alpha_1)$, d'après l'équation (5.23) nous avons :

$$u^{(\alpha_1^{(2)})}(x, \alpha_1) = \frac{u^{(\alpha_1^{(1)})}(x, \alpha_1 + \Delta\alpha_1) - u^{(\alpha_1^{(1)})}(x, \alpha_1 - \Delta\alpha_1)}{2\Delta\alpha_1}. \quad (5.24)$$

Sachant que les sensibilités d'ordre 1 évaluées par les Différences Finies centrées sont égales à :

$$u^{(\alpha_1^{(1)})}(x, \alpha_1 + \Delta\alpha_1) = \frac{u^{(\alpha_1^{(0)})}(x, \alpha_1 + 2\Delta\alpha_1) - u^{(\alpha_1^{(0)})}(x, \alpha_1)}{2\Delta\alpha_1}, \quad (5.25)$$

$$u^{(\alpha_1^{(1)})}(x, \alpha_1 - \Delta\alpha_1) = \frac{u^{(\alpha_1^{(0)})}(x, \alpha_1) - u^{(\alpha_1^{(0)})}(x, \alpha_1 - 2\Delta\alpha_1)}{2\Delta\alpha_1}, \quad (5.26)$$

nous déduisons facilement la formule de Différences Finies pour les dérivées d'ordre 2 :

$$u^{(\alpha_1^{(2)})}(x, \alpha_1) = \frac{u^{(\alpha_1^{(0)})}(x, \alpha_1 + 2\Delta\alpha_1) - 2u^{(\alpha_1^{(0)})}(x, \alpha_1) + u^{(\alpha_1^{(0)})}(x, \alpha_1 - 2\Delta\alpha_1)}{4(\Delta\alpha_1)^2}, \quad (5.27)$$

qui s'exprime uniquement en fonction des sensibilités d'ordre 0.

Nous effectuons le même genre de manipulations pour exprimer les formules des DF des sensibilités uniquement en fonction des sensibilités d'ordre 0. Pour les sensibilités d'ordre 1, il n'y a pas de modifications à apporter, puisque l'expression des Différences Finies dépend uniquement de la solution d'ordre 0. En utilisant des formules de DF qui dépendent uniquement

de sensibilités d'ordre 0 (*i.e.* : $u^{(\alpha_1^{(0)})}$ ou $T^{(\alpha_1^{(0)})}$), le calcul des sensibilités d'ordre arbitraire n par DF requiert uniquement la résolution des EDP (*i.e.* : sensibilités d'ordre 0). En plus de l'erreur d'approximation d'ordre 2 des différences centrées, cette façon de procéder introduit une erreur supplémentaire dans les calculs qui se rapporte à l'erreur numérique découlant du calcul de la solution à l'ordre 0. Notons que dans le cas où nous calculerions les Différences Finies par les sensibilités d'ordre $n - 1$ approximées par la méthode proposée (voir les chapitres 3 et 4), les erreurs numériques liées aux sensibilités d'ordre 0 à $n - 1$ seraient alors introduites dans les calculs. Ces erreurs numériques, propagées et amplifiées, qui proviennent du calcul des sensibilités d'ordre $n - 1$ seraient alors ajoutées au calcul par Différences Finies.

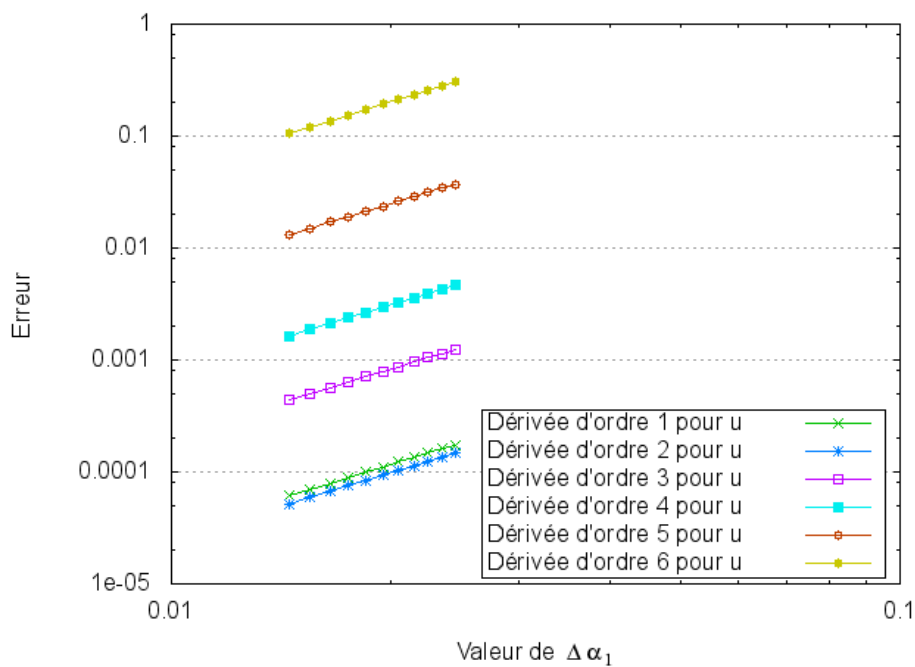
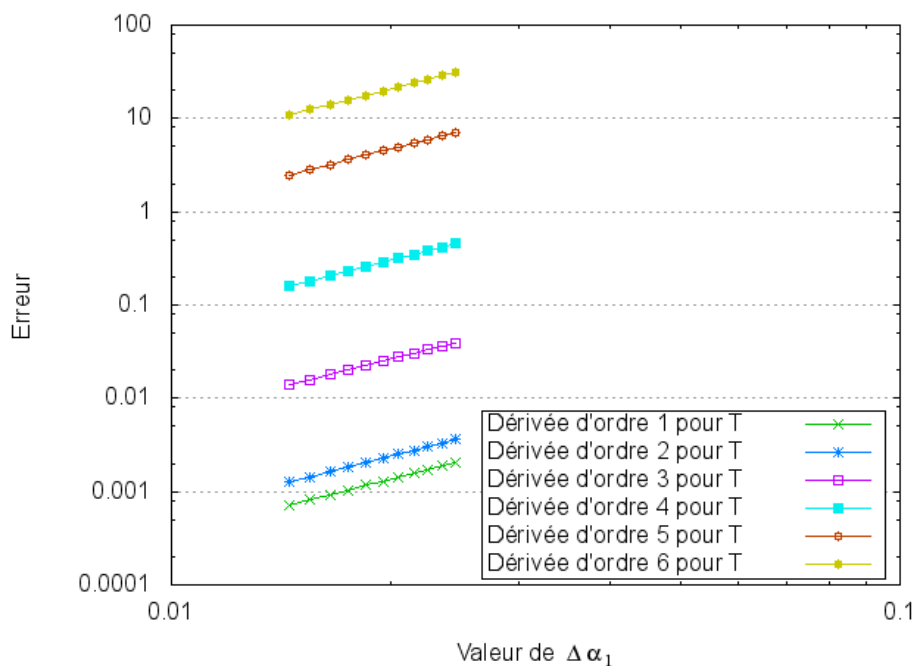
Afin de comparer les résultats obtenus par la méthode des DF et la MESC d'ordre n présentée dans ce travail, nous mesurons l'erreur existant entre les sensibilités qui découlent de ces deux approches. Par souci de simplicité, nous mesurons l'erreur entre la sensibilité numérique d'ordre n et la sensibilité approximée par Différences Finies dans la norme 2 :

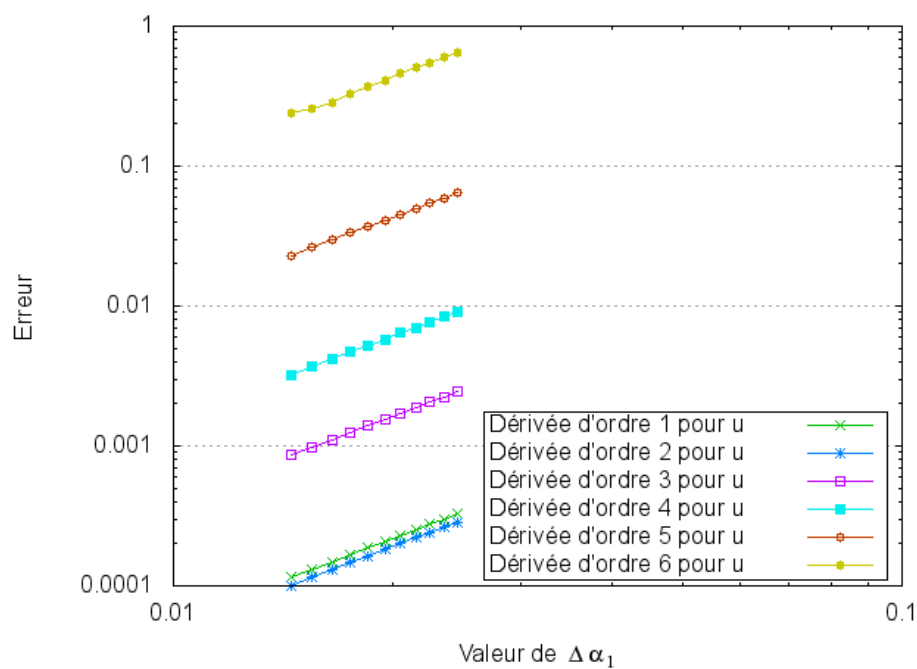
$$\|S_h^{(n)} - S_{FD}^{(n)}\|_2 = \sqrt{\sum_{i=1}^{nn} (S_h^{(n)} - S_{FD}^{(n)})^2}, \quad (5.28)$$

où $S_h^{(n)}$ est une sensibilité d'ordre n obtenue par la MEF, $S_{FD}^{(n)}$ est une sensibilité d'ordre n obtenue par la méthode de DF et nn est le nombre de noeuds de calcul du maillage.

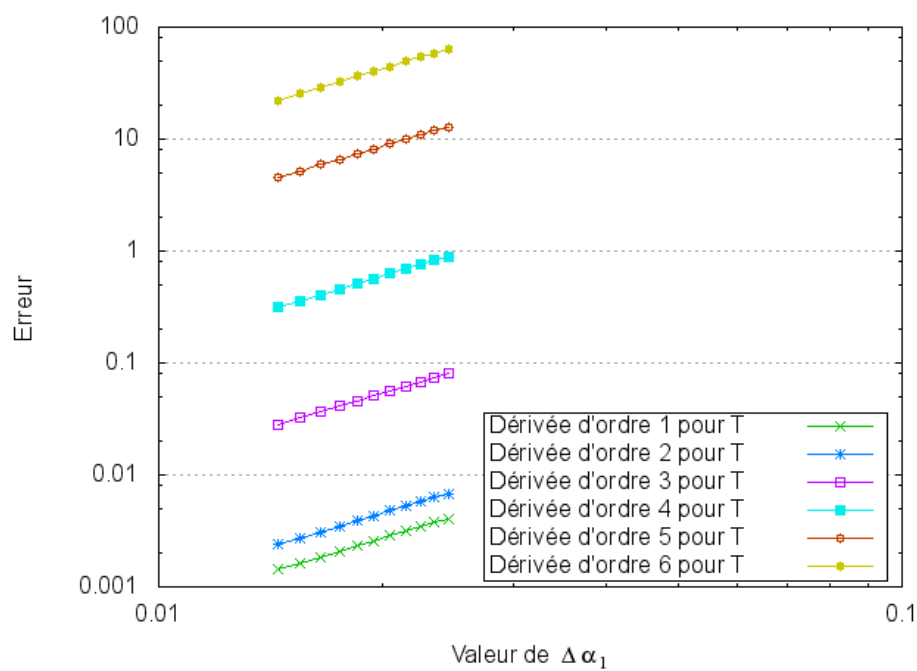
Les figures 5.3 et 5.4 montrent explicitement les erreurs observées pour les sensibilités d'ordre 1 à 6 pour chacune des variables dépendantes u et T étudiées, lorsque les éléments finis sont de degrés 1 et 4 ($p = 1$ et $p = 4$). Les courbes tracées représentent en fait l'erreur en fonction de la perturbation $\Delta\alpha_1$ choisie pour l'approximation des sensibilités par les Différences Finies. Les valeurs numériques de ces erreurs peuvent être consultées dans l'annexe H. On peut alors constater, en s'appuyant sur ces résultats, que l'erreur entre les deux méthodes diminue à un taux de 2, lorsque $\Delta\alpha_1$ décroît. Cela vient confirmer la théorie selon laquelle la formule des DF centrées converge à l'ordre 2 (voir l'équation (5.23)). Notons qu'on observe des résultats similaires avec des éléments finis de degré $p = 2, 3$ et 5. Les graphiques illustrant ces résultats sont consignés dans la section I de l'annexe I. Nous présentons également les valeurs numériques de ces résultats dans des tableaux à l'annexe H.

Nous savons que les DF approximent correctement les sensibilités d'ordre n , puisque nous évaluons les DF grâce aux sensibilités d'ordre 0 dont nous sommes assurés de la fiabilité.

(a) Erreur par rapport à $\Delta\alpha$ pour u en utilisant les éléments $P1$.(b) Erreur par rapport à $\Delta\alpha$ pour T en utilisant les éléments $P1$.Figure 5.3 Sensibilités numériques versus sensibilités des DF (éléments $P1$).



(a) Erreur par rapport à $\Delta \alpha$ pour u en utilisant les éléments $P4$.



(b) Erreur par rapport à $\Delta \alpha$ pour T en utilisant les éléments $P4$.

Figure 5.4 Sensibilités numériques versus sensibilités des DF (éléments $P4$).

L'erreur entre les sensibilités des Différences Finies et celles de notre méthode est relativement faible. On peut donc en déduire que notre programme calcule adéquatement les sensibilités. La proximité entre les résultats obtenus par notre méthode et ceux de la méthode des Différences Finies confirment l'exactitude des prédictions des sensibilités par notre méthode. Notons que pour calculer les sensibilités à paramètres multiples, le processus est identique. La seule différence réside dans le fait que les dérivées considérées sont mixtes.

5.4 Évaluation rapide par la série de Taylor de solutions perturbées

Dans cette dernière section, nous étudions la possibilité d'utiliser la série de Taylor d'ordre n dans l'espace des paramètres pour obtenir une estimation de la solution à des valeurs perturbées des paramètres. On estime donc la solution à des valeurs perturbées des paramètres en disposant uniquement de la solution et de ses sensibilités d'ordre 1 à n à un état de référence (u_0, T_0) . Cela évite donc de refaire des simulations à l'aide de notre méthode à chaque fois que les paramètres du problème sont perturbés. Par exemple, les séries de Taylor d'ordre n pour approximer les valeurs de $u(x, (\alpha_1)_0 + \Delta\alpha_1)$ et $T(x, (\alpha_1)_0 + \Delta\alpha_1)$, qui dépendent du paramètre α_1 , sont :

$$u_{extr}(x, (\alpha_1)_0 + \Delta\alpha_1) = \sum_{i=0}^n u^{(\alpha_1^{(i)})}(x, (\alpha_1)_0) \frac{(\Delta\alpha_1)^i}{i!} + O(\Delta\alpha_1^{n+1}), \quad (5.29)$$

$$T_{extr}(x, (\alpha_1)_0 + \Delta\alpha_1) = \sum_{i=0}^n T^{(\alpha_1^{(i)})}(x, (\alpha_1)_0) \frac{(\Delta\alpha_1)^i}{i!} + O(\Delta\alpha_1^{n+1}), \quad (5.30)$$

où $(\alpha_1)_0$ est la valeur de référence du paramètre α_1 et $O(\Delta\alpha_1^{n+1})$ désigne l'erreur de troncature qui est d'ordre $n + 1$.

Ici, nous étudions le comportement des série de Taylor d'ordre 0 à 6 pour les variables u et T du problème unidimensionnel à 1 paramètre de la section 5.1. Notons que la solution analytique de ce problème est inconnue et que les termes sources f_s et q_s sont nuls. Le maillage utilisé comporte 30 éléments finis de type $P3$ ($p = 3$). Nous avons reporté les valeurs des sensibilités calculées en $(x, (\alpha_1)_0)$ dans le tableau 5.1. Nous utilisons ces valeurs pour prédire les solutions perturbées dans l'espace du paramètre α_1 par la série de Taylor d'ordre 0 à 6 (équations (5.29) et (5.30)). L'expansion d'ordre 0 ne nécessite que la solution des EDP (u et T), tandis que les développements d'ordre 1 à 6 exige des sensibilités d'ordre de plus en plus élevé.

Nous comparons les solutions extrapolées par la série de Taylor d'ordre n avec les solutions

recalculées en $(x, (\alpha_1)_0 + \Delta\alpha_1)$ par la MEF. La figure 5.5 illustre les solutions extrapolées u_{extr} et T_{extr} par rapport aux solutions calculées en $(x, (\alpha_1)_0 + \Delta\alpha_1)$ par la MESC proposée sur $\alpha_1 \in [\frac{\pi}{2} - 1, \frac{\pi}{2} + 1]$. Notons que la valeur de référence $(\alpha_1)_0$ est $\frac{\pi}{2}$, ce qui signifie que $\Delta\alpha_1$ varie entre -1 et 1 (*i.e.* : $\Delta\alpha_1 \in [-1, 1]$).

Tableau 5.1 Valeur de référence des sensibilités d'ordre $n = 0$ à 6 .

Ordre n	$S_u^{(n)}$	$S_T^{(n)}$
0	0.3141	0.2448
1	-0.1997	-1.3560
2	-4.6987E-02	0.1720
3	0.4411	4.2706
4	-1.1640	-22.5311
5	1.9405	35.7519
6	-5.3961	896.8783

Les résultats obtenus montrent que la précision de solution extrapolée de u (voir figure 5.5(a)) augmente avec l'ordre de la série de Taylor sur l'intervalle $[\frac{\pi}{2} - 1, \frac{\pi}{2} + 1]$ du paramètre α_1 . Par exemple, l'erreur entre la solution numérique et la solution extrapolée à l'ordre 6 est inférieure à celle obtenue pour l'extrapolation à l'ordre 4 sur la totalité de l'intervalle de α_1 considéré (voir la figure 5.6(a)). En outre, pour la solution extrapolée de T , nous constatons une certaine amélioration de la précision avec l'ordre lorsqu'on se restreint à une plage de α_1 comprise entre 1.0 et 2.2. Nous remarquons que, pour les séries de Taylor d'ordre 4 et 5, la solution approximée pour T diverge de la solution numérique sur l'intervalle de $\alpha_1 \in [\frac{\pi}{2} - 1, 1]$, alors que pour l'ordre 6, elle oscille légèrement pour ce même intervalle. Pour l'intervalle de $\alpha_1 \in [2.2, \frac{\pi}{2} + 1]$, les solutions approximées par les séries de Taylor d'ordre 3, 4 et 6 divergent. Cela peut être dû au fait qu'on se trouve à l'extérieur de l'intervalle de convergence de la série de Taylor associée à la fonction de T ou encore au fait que T varie rapidement avec α_1 comme on peut l'observer à la figure 5.7(b). Il peut être difficile d'approximer adéquatement un changement rapide de la solution en fonction d'un paramètre pour une série de Taylor dont l'ordre n'est pas suffisamment élevé. Il serait peut-être nécessaire d'approximer T par des séries d'ordre plus élevé. Notons que la fonction de u illustrée par la figure 5.7(a) varie lentement avec α_1 comparativement à T .

En se référant à la figure 5.6, on remarque que l'ordre de précision de la solution extrapolée de u et T est bel et bien $O(\Delta\alpha_1^{n+1})$ pour les séries de Taylor d'ordre 0 à 6 lorsque $\Delta\alpha_1$ est petit

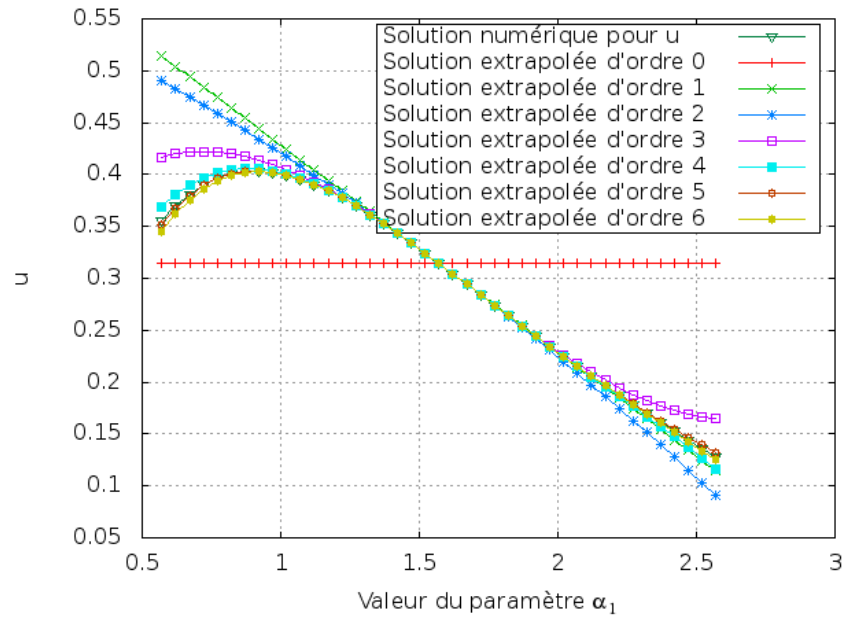
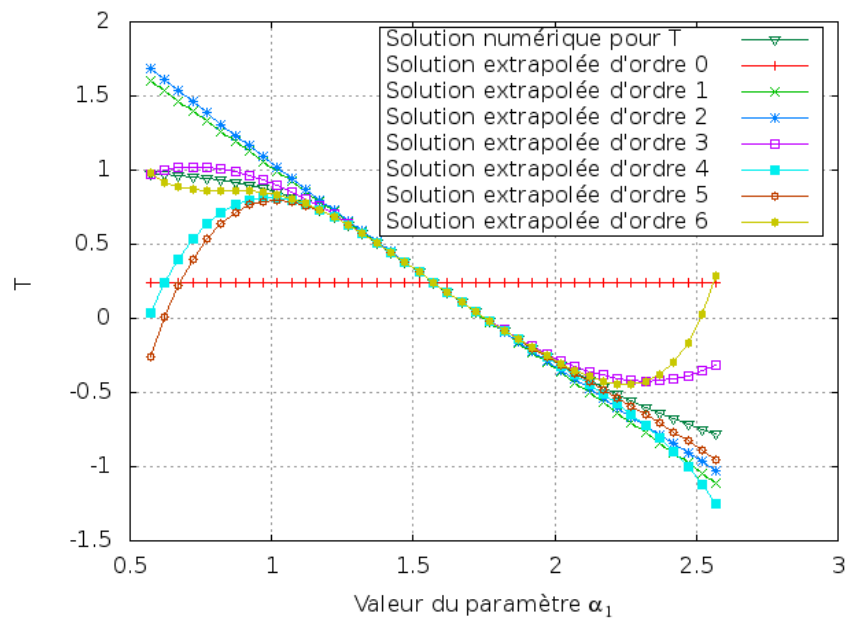
(a) Extrapolation de u en fonction de α_1 .(b) Extrapolation de T en fonction de α_1 .

Figure 5.5 Comparaison des solutions numériques versus les solutions extrapolées par série de Taylor.

tel que prédit par la théorie. Dans certains cas, en augmentant l'ordre n de la série de Taylor, on peut élargir la plage de variation du paramètre de sorte à obtenir des approximations de précision équivalente, voire meilleure, que celles des séries d'ordre inférieur. Cela dépend de l'intervalle de convergence de la série de Taylor de la solution étudiée. Lorsque l'intervalle de convergence est extrêmement grand, voire infini, cela est réalisable ; on peut élargir la plage de variation du paramètre en augmentant l'ordre n de la série de Taylor. Toutefois, si cet intervalle s'avère très petit, il est impossible d'obtenir une approximation précise par la série de Taylor en dehors de cet intervalle peu importe l'ordre de la série. Ce principe s'applique également pour le cas du développement de Taylor à variables multiples utilisé dans le cas où il y a plusieurs paramètres.

Comme l'algorithme implémenté pour résoudre les sensibilités est générique, il est possible d'extrapoler des solutions par une série de Taylor à un ordre élevé n quelconque. Rappelons que l'extrapolation par la série de Taylor, n'est pas une technique infaillible pour approximer les solutions. Par exemple, si une variable d'écoulement varie très rapidement avec le paramètre α_1 , il devient alors difficile d'obtenir une extrapolation précise sur une large plage de valeurs du paramètre α_1 (voir Soulat et al., 2013). Il peut alors être nécessaire de prédire la solution par une série d'ordre n très élevé. On doit alors déterminer à partir de quel ordre n l'approximation est précise. Par ailleurs, l'intervalle de convergence de la série de Taylor en un point de référence (x, α_0) restreint la grandeur de la plage de variation des paramètres sur laquelle l'approximation par la série de Taylor est valide. Dans certains cas, cet intervalle peut être très petit, alors que dans d'autres cas il peut être extrêmement grand, voire infini. Il faut donc prendre garde à vérifier que l'approximation semble toujours se situer dans l'intervalle de convergence. Lorsqu'on peut approximer les solutions de façon précise sur une plage de variation importante des paramètres, la série de Taylor devient un substitut économique comparée à une simulation complète aux valeurs perturbées des paramètres α_i ($i = 1$ à q). Toutefois, lorsque $|\Delta\alpha_i| > 1$ et que l'ordre n de la série est très élevé, l'erreur numérique introduite par les sensibilités dans la série de Taylor s'amplifie de façon considérable. Dans ce cas, nous croyons, qu'il est difficile de converger la série de Taylor pour $|\Delta\alpha_i| > 1$.

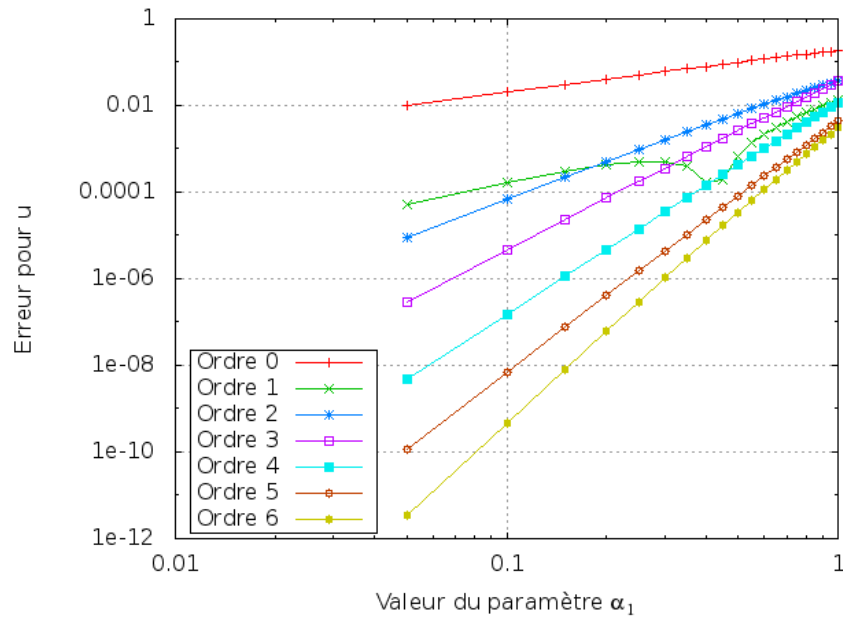
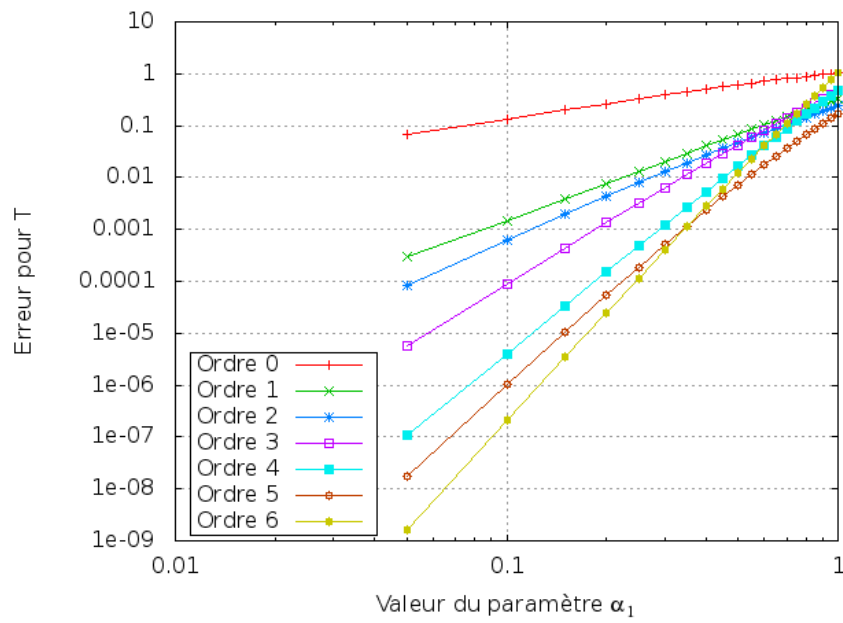
(a) Erreur de u en fonction de α_1 .(b) Erreur de T en fonction de α_1 .

Figure 5.6 Erreur des solutions extrapolées par série de Taylor.

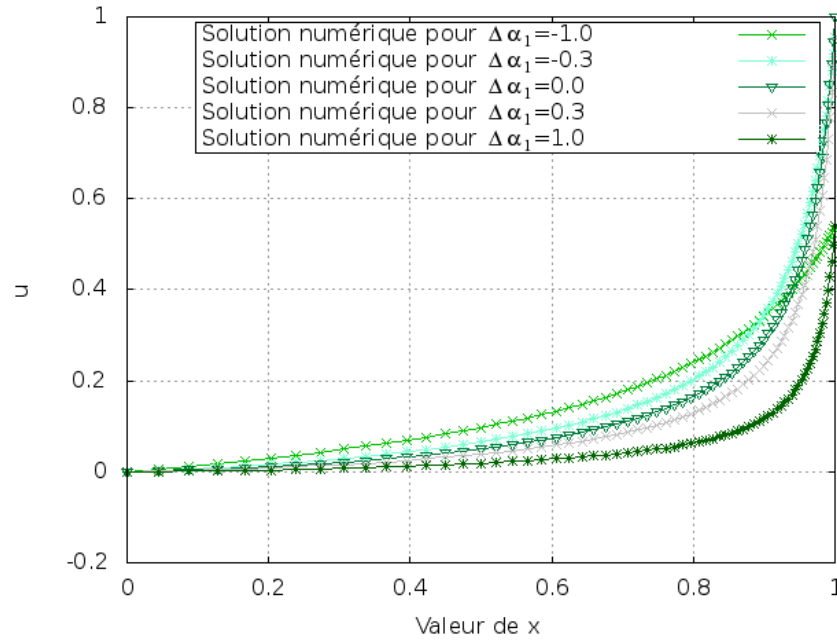
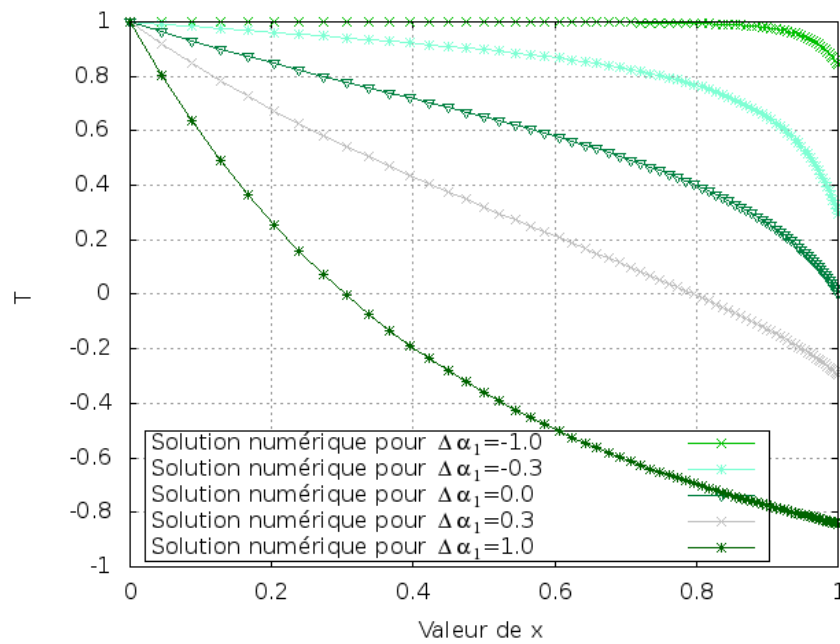
(a) Solution perturbée de u en fonction de x .(b) Solution perturbée de T en fonction de x .

Figure 5.7 Solutions numériques perturbées.

CHAPITRE 6 CONCLUSION

Initialement, nous nous étions fixés les objectifs suivants :

- Développer une méthode générale pour le calcul des sensibilités continues d'ordre arbitrairement élevé n ;
- Implémenter la méthode générale dans un code qui utilise la MEF ;
- Vérifier l'implémentation de la méthode au moyen de la MSM ;
- Valider les sensibilités calculées par la méthode par comparaison avec les sensibilités obtenues par la méthode des DF ;
- Expérimenter la méthode sur des problèmes classiques en 1D ;
- Démontrer l'efficacité de la méthode pour approximer les solutions par la série de Taylor sur de larges plages de valeurs des paramètres.

Dans le cadre de ce travail, nous sommes parvenus à :

- Développer une formulation générique et générale des équations de sensibilités d'ordre n arbitrairement élevé avec un nombre q quelconque de paramètres (voir équation (3.55)) ;
- Implémenter un algorithme générique et général pour générer automatiquement les composantes des équations de sensibilités sans avoir à les écrire explicitement à l'avance ;
- Implémenter un algorithme d'éléments finis générique qui, en plus de générer automatiquement la discrétisation des équations d'état et de sensibilités, construit automatiquement les matrices élémentaires de l'écoulement et des sensibilités d'ordre n par rapport à q paramètres ;
- Démontrer la faisabilité de la méthodologie sur un problème unidimensionnel avec des éléments finis de degré 1 à 5 ;
- Vérifier la méthodologie à l'aide de la MSM ;
- Valider la méthodologie en comparant les sensibilités calculées par notre méthode et celles obtenues par Différences Finies ;
- Montrer, sous certaines conditions de régularité de la solution, que :
 1. la série de Taylor converge vers la solution pour les petites perturbations des paramètres ;
 2. la série de Taylor d'ordre élevé approxime la solution de façon plus précise que la série de Taylor d'ordre plus faible sur une même plage de valeurs des paramètres.

Notre méthode des sensibilités est basée sur la MESC qui consiste à différentier les EDP, puis à discrétiser les équations différentielles ainsi obtenues. Notre approche peut donc être qualifiée de différentiation automatiques des EDP dans le continu.

Dans le cas d'un système non-linéaire perturbé, l'approximation des solutions par série de Taylor a pour avantage d'engendrer des coûts nettement inférieurs aux coûts habituellement produits par la résolution du système non-linéaire perturbé. La série de Taylor s'avère donc être une alternative intéressante au lieu de recalculer la solution aux valeurs perturbées des paramètres. Toutefois, nous avons conclu que cela ne peut être envisageable que sous certaines conditions, c'est-à-dire si l'intervalle de convergence de la série de Taylor n'est pas petit, si la solution ne présente pas de discontinuités, etc. Dans le cas d'une variation rapide de la solution en fonction du paramètre α_1 , une série de Taylor d'ordre supérieur ne donne pas nécessairement une approximation plus précise que celle obtenue par une série d'ordre inférieur lorsque nous considérons les résultats sur une même plage de valeurs du paramètre α_1 . Une des solutions imaginables pour remédier à cet inconvénient serait de réduire la plage de valeurs du paramètre α_1 . Cependant, cela est en contradiction avec notre objectif qui consiste à élargir la plage de valeurs des paramètres en extrapolant les solutions par la série de Taylor d'ordre n supérieur. Une autre option serait alors d'approximer la solution par le biais de la série de Taylor à un ordre très élevé en vue d'améliorer la précision des approximations pour les cas où il y a des variations très rapides de la solution en fonction d'un paramètre α_i (voir Soulat et al., 2013).

6.1 Limitations de la solution proposée

Par souci de limiter la complexité du projet, nous nous sommes restreints au traitement des paramètres de valeur. En outre, notre méthode ne nous permet pas pour l'instant de résoudre les problèmes où il y a des conditions aux frontières de Neumann. Par souci de simplicité, nous avons donc considéré un problème avec des conditions limites de Dirichlet sur les variables dépendantes.

Un autre inconvénient de notre méthode est qu'elle tient pour acquis que les sensibilités qu'elle résout sont infiniment différentiables. Toutefois, il existe certains cas en mécanique des fluides où des discontinuités peuvent apparaître dans la solution des EDP décrivant le problème d'écoulement ou encore dans ses sensibilités. Par exemple, si on prend une onde de choc, il y a discontinuité.

Lors de l'extrapolation des solutions perturbées dans l'espace des paramètres, nous nous sommes aperçus que la série de Taylor ne garantissait pas toujours une approximation adéquate de la solution perturbée. La précision de l'extrapolation par la série de Taylor dépend de certains facteurs. Dans le cas où l'intervalle de convergence de la série de Taylor est petit, il est difficile d'élargir la plage des paramètres de façon à obtenir une approximation respectant le critère de précision émis et ce, même en augmentant l'ordre de la série de Taylor. La série de Taylor éprouve également des difficultés à représenter le comportement de la solution lorsqu'elle varie rapidement avec les paramètres. Il est donc difficile d'obtenir un résultat précis par le biais de la série de Taylor pour ce cas particulier. Enfin, nous croyons que la série de Taylor peut difficilement converger pour une variation d'un paramètre supérieure à 1 (*e.g.* : $|\alpha_1| > 1$), car l'erreur numérique associée aux sensibilités introduites dans la série ne cesse d'augmenter avec l'ordre.

6.2 Améliorations futures

L'objectif principal de ce mémoire était de développer une méthode capable de générer automatiquement les équations de sensibilités d'ordre n d'un système $1D$ et de les résoudre de façon générique. Ici, nous suggérons certaines améliorations qui devraient être apportées à notre méthode afin qu'elle puisse être appliquée à un plus grand nombre de problèmes :

- À court terme, étudier le cas des problèmes avec conditions de Neumann en $1D$;
- Traiter les paramètres de forme en $1D$;
- À plus long terme, étendre l'approche aux équations de Navier-Stokes multidimensionnelle,
- À long terme, étudier l'approche de Soulat et al. pour le cas où la solution présente des quasi-discontinuités dans sa dépendance avec les paramètres,
- Appliquer la méthodologie à des problèmes de design optimal et étudier la possibilité d'utiliser la série de Taylor d'ordre élevé comme substitut aux calculs MFN aux valeurs perturbées des paramètres pour évaluer la fonction objectif et son gradient.

Pour traiter les problèmes avec des conditions de Neumann, il sera essentiel d'adapter notre Méthode d'Équations des Sensibilités Continues d'ordre n . Cela nécessitera des changements au niveau de la forme faible de la MEF, puisque c'est au niveau des intégrales sur la frontière Γ_N , où les conditions de Neumann sont imposées, que l'impact se répercutera lors des calculs.

Pour le cas des paramètres de forme, on devra dériver la forme faible dans le continuum :

$$\frac{d^n}{d\boldsymbol{\alpha}^n} \int_{\Omega(\boldsymbol{\alpha})} F(u(\boldsymbol{\alpha}), T(\boldsymbol{\alpha}); \boldsymbol{\alpha}) d\Omega, \quad (6.1)$$

où $\Omega(\boldsymbol{\alpha})$ est le domaine qui varie avec les paramètres $\boldsymbol{\alpha}$ et $F(u(\boldsymbol{\alpha}), T(\boldsymbol{\alpha}); \boldsymbol{\alpha})$ désigne la forme faible liée aux EDP dont les variables dépendantes des paramètres $\boldsymbol{\alpha}$ sont u et T .

On devra introduire un «mapping» sur un domaine Ω_0 indépendant de $\boldsymbol{\alpha}$ afin d'être en mesure d'introduire la dérivée à l'intérieur de l'intégrale :

$$\frac{d^n}{d\boldsymbol{\alpha}^n} \int_{\Omega_0} F(u(\boldsymbol{\alpha}), T(\boldsymbol{\alpha}); \boldsymbol{\alpha}) \dots d\Omega_0. \quad (6.2)$$

Suite à la transformation sur le domaine Ω_0 , l'intégrale contiendra des termes additionnels qui ne sont pas notés ici.

Éventuellement, nous souhaitons appliquer notre méthode des sensibilités aux problèmes de Navier-Stokes en 2D. Certaines modifications d'ordre technique devront être réalisées pour adapter notre code de calcul en vue de l'appliquer aux problèmes 2D. Le principe pour générer les équations de sensibilités du système à résoudre demeurera le même (voir section 3.3). Toutefois, il faudra adapter la MEF pour être en mesure de calculer les intégrales numériques des résidus dans le domaine 2D.

Il faudrait idéalement trouver un moyen de déterminer le rayon de convergence lors de l'évaluation d'une solution par la série de Taylor en $(u; \boldsymbol{\alpha}_0)$. Toutefois, cela semble se révéler une tâche plutôt ardue, puisque la solution analytique est inconnue. Par contre, il est possible d'analyser les approximations pour différents ordres n afin de déduire si la série de Taylor semble converger vers une même solution. Dans le cas où il nous apparaît que le rayon de convergence est petit, nous pourrions approximer la solution à partir d'un autre point de référence $(u; \boldsymbol{\alpha}_0)$. Cela pourrait nous permettre d'obtenir un intervalle de convergence plus important pour la série de Taylor.

Ici, nous nous sommes contentés de valider notre méthode pour le cas des sensibilités à 1 paramètre. Il faudrait valider notre méthode des sensibilités pour le cas où plusieurs paramètres sont considérés dans le système à résoudre. La méthode des DF serait un bon moyen pour s'assurer que notre méthode est en mesure de calculer les dérivées croisées des variables

d'état. Le même constat s'applique pour l'extrapolation par le biais de la série de Taylor. Il serait judicieux de prendre un cas à plusieurs paramètres et de comparer l'approximation obtenue par la série de Taylor avec la solution numérique de notre méthode lorsque plusieurs paramètres sont perturbés en simultané.

RÉFÉRENCES

- G. Allaire, *Introduction à la modélisation mathématique et à la simulation numérique*. Editions Ecole Polytechnique, 2005, book section 1, pp. 1–30.
- I. Ammara, “Optimisation de forme de profils aérodynamiques par le calcul de sensibilités discrètes obtenues par la technique des récurrences simplifiées,” Thesis, 2005.
- M. Asli et A. M. Tousi, “Sensitivity analysis of a centrifugal compressor using the linearization method,” dans *ASME-JSME-KSME 2011 Joint Fluids Engineering Conference*. American Society of Mechanical Engineers, 2011, Conference Proceedings, pp. 1785–1790.
- M. H. Bakr, P. Zhao, et N. K. Nikolova, “Adjoint first order sensitivities of transient responses and their applications in the solution of inverse problems,” *Antennas and Propagation, IEEE Transactions on*, vol. 57, no. 7, pp. 2137–2146, 2009.
- C. Bischof, P. Khademi, A. Mauer, et A. Carle, “Adifor 2.0 : Automatic differentiation of fortran 77 programs,” *Computational Science and Engineering, IEEE*, vol. 3, no. 3, pp. 18–32, 1996.
- J. Borggaard et J. Burns, “A pde sensitivity equation method for optimal aerodynamic design,” *Journal of Computational Physics*, vol. 136, no. 2, pp. 366–384, 1997.
- J. Borggaard et A. Verma, “On efficient solutions to the continuous sensitivity equation using automatic differentiation,” *SIAM Journal on Scientific Computing*, vol. 22, no. 1, pp. 39–62, 2000.
- R. Cotte, *L’enjeu de la différentiation automatique dans les méthodes de Newton d’ordres supérieurs*. Université de Sherbrooke, 2010.
- K. J. Dowding et B. F. Blackwell, “Sensitivity analysis for nonlinear heat conduction,” *Journal of Heat Transfer*, vol. 123, no. 1, pp. 1–10, 2001.
- R. Duvigneau et D. Pelletier, “A sensitivity equation method for fast evaluation of nearby flows and uncertainty analysis for shape parameters,” *International Journal of Computational Fluid Dynamics*, vol. 20, no. 7, pp. 497–512, 2006.
- S. Etienne, D. Pelletier, et A. Garon, “Application of a sensitivity equation method to generic fsi biomedical problems,” dans *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006, Conference Proceedings, pp. 2004–0694.
- P. Flusser et G. A. Francia, “Derivation and visualization of the binomial theorem,” *International Journal of Computers for Mathematical Learning*, vol. 5, no. 1, pp. 3–24, 2000.

- A. Fortin et A. Garon, “Les éléments finis : de la théorie à la pratique,” *Notes de cours*, 2014.
- D. Gao, W. R. Stockwell, et J. B. Milford, “First-order sensitivity and uncertainty analysis for a regional-scale gas-phase chemical mechanism,” *Journal of Geophysical Research : Atmospheres (1984–2012)*, vol. 100, no. D11, pp. 23 153–23 166, 1995.
- J. C. Gilbert, G. Le Vey, et J. Masse, “La différentiation automatique de fonctions représentées par des programmes,” 1991.
- A. G. Godfrey et E. M. Cliff, “Direct calculation of aerodynamic force derivatives : a sensitivity-equation approach,” *AIAA Paper*, vol. 393, p. 1998, 1998.
- M. Gunzburger, “Adjoint equation-based methods for control problems in incompressible, viscous flows,” *Flow, Turbulence and Combustion*, vol. 65, no. 3-4, pp. 249–272, 2000.
- M. D. Gunzburger, “Sensitivities, adjoints and flow optimization,” *International Journal for Numerical Methods in Fluids*, vol. 31, no. 1, pp. 53–78, 1999.
- R. T. Haftka et D. S. Malkus, “Calculation of sensitivity derivatives in thermal problems by finite differences,” *International Journal for Numerical Methods in Engineering*, vol. 17, no. 12, pp. 1811–1821, 1981.
- P. Heimbach, C. Hill, et R. Giering, “An efficient exact adjoint of the parallel mit general circulation model, generated via automatic differentiation,” *Future Generation Computer Systems*, vol. 21, no. 8, pp. 1356–1371, 2005.
- C. Homescu, “Adjoint and automatic (algorithmic) differentiation in computational finance,” *Available at SSRN 1828503*, 2011.
- F. Ilinca, D. Pelletier, et A. Hay, “First-and second-order sensitivity equation methods for value and shape parameters,” *International journal for numerical methods in fluids*, vol. 57, no. 9, pp. 1349–1370, 2008.
- J. Iott, R. T. Haftka, et H. M. Adelman, *Selecting step sizes in sensitivity analysis by finite differences*. National Aeronautics and Space Administration, Scientific and Technical Information Branch, 1985, vol. 86382.
- A. Katz, S. S. Nambi, K. Mather, A. D. Baron, D. A. Follmann, G. Sullivan, et M. J. Quon, “Quantitative insulin sensitivity check index : a simple, accurate method for assessing insulin sensitivity in humans,” *The Journal of Clinical Endocrinology and Metabolism*, vol. 85, no. 7, pp. 2402–2410, 2000.
- V. Komkov, K. K. Choi, et E. J. Haug, *Design sensitivity analysis of structural systems*. Academic press, 1986, vol. 177.
- S. G. Krantz et H. R. Parks, *A primer of real analytic functions*. Springer, 2002.

- F. J. Lu, Q. W. Yang, J. H. Li, W. M. Jin, et Y. Yao, “Structural damage assessment by an improved flexibility sensitivity method,” *Applied Mechanics and Materials*, vol. 351, pp. 532–535, 2013.
- J.-N. Mahieu, S. Etienne, D. Pelletier, et J. Borggaard, “A second-order sensitivity equation method for laminar flow,” *International Journal of Computational Fluid Dynamics*, vol. 19, no. 2, pp. 143–157, 2005.
- J.-N. Mahieu, “Méthode d’équations des sensibilités à l’ordre deux pour des écoulements laminaires,” M.Sc.A., 2003, copyright - Copyright UMI - Dissertations Publishing 2003 Dernière mise à jour - 2014-01-20 Première page - n/a.
- D. Mazkewitsch, “The n-th derivative of a product,” *American Mathematical Monthly*, pp. 739–742, 1963.
- M. Nyblom, “On a generalization of the binomial theorem,” 1996.
- D. Papadimitriou et K. Giannakoglou, “Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint,” *International Journal for Numerical Methods in Fluids*, vol. 71, no. 5, pp. 652–670, 2013.
- D. Pelletier, A. Hay, S. Etienne, et J. Borggaard, “The sensitivity equation method in fluid mechanics,” *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 17, no. 1-2, pp. 31–61, 2008.
- M. M. Putko, A. C. Taylor, P. A. Newman, et L. L. Green, “Approach for input uncertainty propagation and robust design in cfd using sensitivity derivatives,” *Journal of fluids engineering*, vol. 124, no. 1, pp. 60–69, 2002.
- B. Rao et S. Rahman, “A continuum shape sensitivity method for fracture analysis of orthotropic functionally graded materials,” *Mechanics of materials*, vol. 37, no. 10, pp. 1007–1025, 2005.
- P. J. Roache, *Verification and validation in computational science and engineering*. Hermosa Albuquerque, NM, 1998.
- , “Code verification by the method of manufactured solutions,” *Journal of Fluids Engineering*, vol. 124, no. 1, pp. 4–10, 2002.
- L. L. Sherman, A. C. Taylor III, L. L. Green, P. A. Newman, G. W. Hou, et V. M. Korivi, “First-and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods,” *Journal of Computational Physics*, vol. 129, no. 2, pp. 307–331, 1996.
- J. Sobieszczanski-Sobieski, “High order sensitivity analysis of complex, coupled systems,” *AIAA journal*, vol. 28, no. 4, pp. 756–758, 1990.

L. Soulat, P. Ferrand, S. Moreau, S. Aubert, et M. Buisson, “Efficient optimisation procedure for design problems in fluid mechanics,” *Computers & Fluids*, vol. 82, pp. 73–86, 2013.

L. G. Stanley, “Computational methods for sensitivity analysis with applications to elliptic boundary value problems,” Thesis, 2000.

E. Turgeon, D. Pelletier, et J. Borggaard, “A general continuous sensitivity equation formulation for complex flows,” *Numerical Heat Transfer : Part B : Fundamentals*, vol. 42, no. 6, pp. 485–498, 2002.

E. Turgeon, “Méthode d’éléments finis adaptative pour la résolution des sensibilités d’écoulements,” Ph.D., 2001, copyright - Copyright UMI - Dissertations Publishing 2001 Dernière mise à jour - 2014-01-22 Première page - n/a.

F. Van Keulen, R. Haftka, et N. Kim, “Review of options for structural design sensitivity analysis. part 1 : Linear systems,” *Computer methods in applied mechanics and engineering*, vol. 194, no. 30, pp. 3213–3243, 2005.

A. Verma, “An introduction to automatic differentiation,” *CURRENT SCIENCE-BANGALORE-*, vol. 78, no. 7, pp. 804–807, 2000.

ANNEXE A ANALYSE COMBINATOIRE

Cette annexe fait un rappel de la terminologie utilisée en combinatoire : arrangement, combinaison, arrangement avec répétition, combinaison avec répétition, etc.

Arrangement

Un arrangement est une liste ordonnée de k éléments choisis parmi un ensemble de n éléments distincts. Dans un arrangement, l'ordre dans lequel les objets sont sélectionnés est important. On considère donc que deux arrangements de k éléments sont distincts s'ils comportent des éléments distincts ou encore si l'ordre des éléments les composant est différent. Par exemple, si nous avons un ensemble de 3 lettres $E = \{a, b, c\}$ ($n = 3$) et que nous souhaitons connaître tous les arrangements possibles en prenant 2 lettres ($k = 2$) dans cet ensemble, nous obtenons :

$$\{a, b\}, \{b, a\}, \{a, c\}, \{c, a\}, \{b, c\}, \{c, b\}.$$

Le nombre d'arrangements de k éléments parmi n est déterminé au moyen de la formule suivante :

$$A_{n,k} = \frac{n!}{(n-k)!} \quad (\text{A.1})$$

Combinaison

Les combinaisons ne tiennent pas compte de l'ordre dans lequel nous choisissons les éléments. Ainsi, pour l'exemple précédent, les combinaisons de 2 lettres tirées parmi les 3 lettres de l'ensemble E sont :

$$\{a, b\}, \{b, c\}, \{a, c\}.$$

Le nombre de combinaisons de k objets parmi n objets distincts est donné par ($n \geq k$) :

$$C_k^n = \frac{A_{n,k}}{k!} = \frac{n!}{k!(n-k)!}. \quad (\text{A.2})$$

Nous allons démontrer par récurrence que C_k^n est un entier. Pour $n = 1$ et pour $k = 1$ ($k \leq n$),

nous obtenons :

$$C_k^1 = \frac{1!}{1!(1-1)!} = 1, \quad (\text{A.3})$$

qui est un entier.

Nous supposons ensuite que C_k^n donne un entier pour $n \geq 1$. Nous vérifions alors si c'est vrai pour $n + 1$. D'après la formule du triangle de Pascal

$$C_k^{n+1} = \frac{(n+1)!}{k!(n+1-k)!} = C_{k-1}^n + C_k^n. \quad (\text{A.4})$$

Comme nous supposons que C_k^n est un entier pour $n \geq 1$ lorsque $k \leq n$, nous avons donc une somme d'entiers dans l'équation (A.4). C_k^{n+1} est donc un entier. Nous pouvons conclure que C_k^n est un entier $\forall n \in \mathbb{N}$ lorsque $k \leq n$.

Répétition des éléments

Il existe également des configurations d'éléments où il y a répétition (ou remise) des éléments de l'ensemble E . On parle alors d'arrangements avec répétition ou de combinaisons avec répétition. Dans le cas des arrangements avec répétition, lorsqu'on choisit k éléments parmi n , ceux-ci peuvent être répétés. Il s'ensuit que le nombre d'arrangements avec répétition est égal à :

$$B_k^n = n^k \quad (\text{A.5})$$

Pour l'exemple précédent, les arrangements avec répétition sont :

$$\{a, a\}, \{a, b\}, \{b, a\}, \{b, b\}, \{a, c\}, \{c, a\}, \{c, c\}, \{b, c\}, \{c, b\}.$$

Dans le cas des combinaisons avec répétition (ou remise), on cherche les façons de sélectionner k éléments avec répétition parmi un ensemble de n éléments. Il est à noter que l'ordre de la sélection n'a pas d'importance, ni pour les combinaisons classiques (sans répétition), ni pour les combinaisons avec répétition. La différence pour les combinaisons avec répétition réside dans le fait que, parmi les k éléments choisis, il est possible d'avoir des éléments identiques. On dit alors de ces éléments qu'ils sont répétés. Dans l'exemple précédent, si nous faisons des combinaisons avec répétition de 2 éléments choisis dans l'ensemble $E = \{a, b, c\}$, nous obtenons les 6 sous-ensembles suivants

$$\{a, a\}, \{a, b\}, \{b, b\}, \{a, c\}, \{c, c\}, \{b, c\}.$$

La formule suivante calcule le nombre de combinaisons avec répétition de k éléments choisis parmi un ensemble de n éléments :

$$\Gamma_n^k = C_k^{n+k-1} = \frac{(n+k-1)!}{k!(n-1)!}. \quad (\text{A.6})$$

Nous ne nous attardons pas sur les démonstrations des formules (A.1), (A.2), (A.5) et (A.6), puisque l'objet de ce travail n'est pas l'analyse combinatoire.

Principe multiplicatif

Le principe multiplicatif est un théorème fondamental du dénombrement. On fait appel au principe multiplicatif, lorsque qu'un évènement est subdivisé en q étapes donnant lieu respectivement à s_1, s_2, \dots, s_q possibilités. Pour passer de l'étape 1 à l'étape finale q , il existe alors un nombre total s de possibilités égal à :

$$s = s_1 \cdot s_2 \cdot \dots \cdot s_q. \quad (\text{A.7})$$

Par exemple, si nous avons :

- Un évènement A qui peut se produire de n_A façons différentes ;
- Un évènement B qui peut se produire de n_B façons différentes ;
- Un évènement C qui peut se produire de n_C façons différentes.
- ...

Nous admettons que tous ces évènements sont indépendants. Par conséquent, le total s des possibilités pour combiner les évènements A, B, C, \dots correspond au produit des possibilités pour chacun des évènements, soit :

$$s = n_A \cdot n_B \cdot n_C \cdot \dots$$

Plus concrètement, prenons l'exemple d'un itinéraire pour se rendre d'un point de départ A à une destination finale C en passant par une destination intermédiaire B . Nous savons qu'il existe :

- 4 chemins partant de A vers B ;
- 2 chemins partant de B vers C .

À partir de l'équation (A.7), nous pouvons déterminer qu'il y a $4 \cdot 2 = 8$ itinéraires possibles. Nous illustrons ces chemins de la façon suivante :

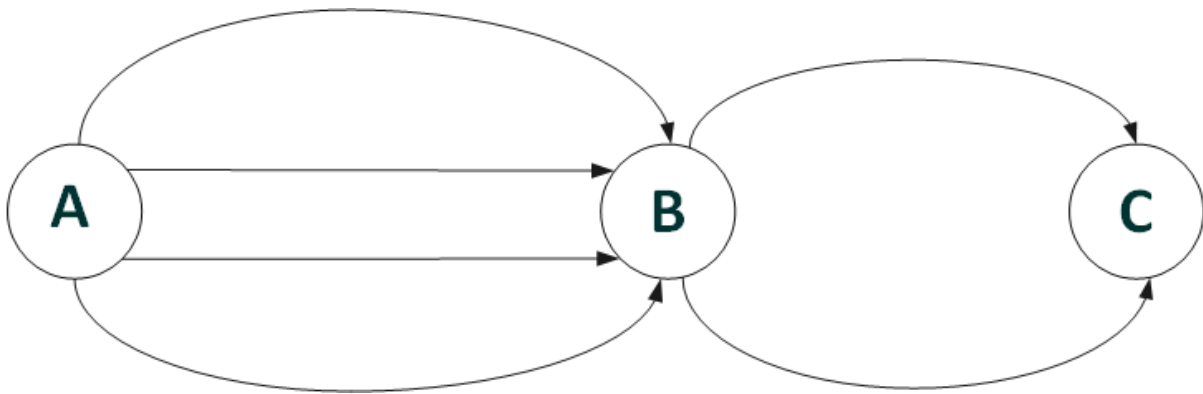


Figure A.1 Itinéraires pour aller de A vers B, puis B vers C.

ANNEXE B OPÉRATEUR DE SOMMATION

L'opérateur de sommation \sum permet de faire la sommation de plusieurs termes. Par exemple, pour $m \in \mathbb{N}^*$, nous avons :

$$\sum_{i=1}^m a_i = a_1 + a_2 + \dots + a_m. \quad (\text{B.1})$$

Par ailleurs, une somme à l'intérieur de l'opérateur peut être noté comme suit :

$$\sum_{i=1}^m (a_i + b_i) = \sum_{i=1}^m a_i + \sum_{i=1}^m b_i. \quad (\text{B.2})$$

Avec les produits intérieurs, c'est différent :

$$\sum_{i=1}^m (a_i b_i) \neq \sum_{i=1}^m a_i \sum_{i=1}^m b_i. \quad (\text{B.3})$$

En d'autres termes, la somme d'un produit n'est pas égale au produit de la somme.

Double sommation

Il est possible de faire une somme double. On définit une somme double toute somme ressemblant à :

$$\sum_{i=1}^l \sum_{j=1}^k a_{i,j}. \quad (\text{B.4})$$

Par définition, nous avons :

$$\sum_{i=1}^l S_i, \text{ où } S_i = \sum_{j=1}^k a_{i,j}. \quad (\text{B.5})$$

Comme les bornes de la seconde sommation sont indépendantes de l'indice de la première sommation, nous pouvons également noter de façon compacte cette somme :

$$\sum_{i=1}^l \sum_{j=1}^k a_{i,j} = \sum_{\substack{1 \leq i \leq l \\ 1 \leq j \leq k}} a_{i,j}. \quad (\text{B.6})$$

Dans le cas où il y a dépendance, la notation compacte serait différente de celle donnée

par (B.6). Par exemple, si nous avons la sommation :

$$\sum_{i=1}^k \sum_{j=1}^i a_{i,j}. \quad (\text{B.7})$$

La notation compacte de (B.7) est :

$$\sum_{i=1}^k \sum_{j=1}^i a_{i,j} = \sum_{1 \leq j \leq i \leq k} a_{i,j}. \quad (\text{B.8})$$

Dans le cas où la somme double s'effectue sur un produit de termes et que, parmi ceux-ci, un dépend d'un seul indice, nous pouvons le mettre en évidence dans l'équation (en le sortant des opérateurs de sommation dont il ne dépend pas) :

$$\sum_{i=1}^l \sum_j^k a_{i,j} b_j = \sum_{j=1}^k b_j \sum_{i=1}^l a_{i,j}. \quad (\text{B.9})$$

L'inversion des opérateurs de sommation n'a aucun impact puisque leurs bornes ne dépendent pas des indices i et j .

Dans le cas où les expressions dépendent d'un seul indice, nous avons :

$$\sum_i \sum_j a_i b_j = \left(\sum_i a_i \right) \left(\sum_j b_j \right). \quad (\text{B.10})$$

Les notations pour les sommations doubles sont valides pour les sommations multiples (*i.e.* : q opérateurs de sommation) pour les cas où les bornes des opérateurs ne dépendent pas des indices des autres opérateurs de l'équation.

ANNEXE C FORMULATION GÉNÉRALE DES ÉQUATIONS DE SENSIBILITÉS DE NAVIER-STOKES

Nous développons dans cette annexe les formulations générales pour les équations de sensibilités de l'équation de mouvement du problème de Navier-Stokes.

Tout d'abord, nous déterminons le nombre de produits pour l'équation de mouvement en x et celle en y . Les équations (3.62) et (3.63) se composent de 11 produits de fonctions. Pour l'équation (3.62), ceux-ci sont :

- $\rho u \frac{\partial u}{\partial x}$ qui est formé de 3 fonctions : ρ , u et $\frac{\partial u}{\partial x}$;
- $\rho v \frac{\partial u}{\partial y}$ qui est formé de 3 fonctions : ρ , v et $\frac{\partial u}{\partial y}$;
- $\frac{\partial P}{\partial x}$ qui est formé d'une fonction ;
- $2\mu \frac{\partial^2 u}{\partial x^2}$ qui est formé de 2 fonctions : μ et $\frac{\partial^2 u}{\partial x^2}$;
- $\mu \frac{\partial^2 v}{\partial x \partial y}$ qui est formé de 2 fonctions : μ et $\frac{\partial^2 v}{\partial x \partial y}$;
- $\mu \frac{\partial^2 u}{\partial y^2}$ qui est formé de 2 fonctions : μ et $\frac{\partial^2 u}{\partial y^2}$;
- $2 \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial x}$ qui est formé de 2 fonctions : $\frac{\partial \mu}{\partial x}$ et $\frac{\partial u}{\partial x}$;
- $\frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x}$ qui est formé de 2 fonctions : $\frac{\partial \mu}{\partial y}$ et $\frac{\partial v}{\partial x}$;
- $\frac{\partial \mu}{\partial y} \frac{\partial u}{\partial y}$ qui est formé de 2 fonctions : $\frac{\partial \mu}{\partial y}$ et $\frac{\partial u}{\partial y}$;
- $\rho g_x \beta (T - T_0)$ qui est formé de 4 fonctions : ρ , g_x , β et $(T - T_0)$;
- f_x qui est formé d'une fonction.

Alors que pour l'équation (3.63), ceux-ci sont :

- $\rho u \frac{\partial v}{\partial x}$ qui est formé de 3 fonctions : ρ , u et $\frac{\partial v}{\partial x}$;
- $\rho v \frac{\partial v}{\partial y}$ qui est formé de 3 fonctions : ρ , v et $\frac{\partial v}{\partial y}$;
- $\frac{\partial P}{\partial y}$ qui est formé d'une fonction ;
- $2\mu \frac{\partial^2 v}{\partial y^2}$ qui est formé de 2 fonctions : μ et $\frac{\partial^2 v}{\partial y^2}$;
- $\mu \frac{\partial^2 u}{\partial x \partial y}$ qui est formé de 2 fonctions : μ et $\frac{\partial^2 u}{\partial x \partial y}$;
- $\mu \frac{\partial^2 v}{\partial x^2}$ qui est formé de 2 fonctions : μ et $\frac{\partial^2 v}{\partial x^2}$;
- $2 \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial y}$ qui est formé de 2 fonctions : $\frac{\partial \mu}{\partial y}$ et $\frac{\partial v}{\partial y}$;
- $\frac{\partial \mu}{\partial x} \frac{\partial u}{\partial y}$ qui est formé de 2 fonctions : $\frac{\partial \mu}{\partial x}$ et $\frac{\partial u}{\partial y}$;
- $\frac{\partial \mu}{\partial x} \frac{\partial v}{\partial x}$ qui est formé de 2 fonctions : $\frac{\partial \mu}{\partial x}$ et $\frac{\partial v}{\partial x}$;
- $\rho g_y \beta (T - T_0)$ qui est formé de 4 fonctions : ρ , g_y , β et $(T - T_0)$;
- f_y qui est formé d'une fonction.

En observant chacun des produits, nous constatons que la valeur de m , pour l'équation (3.55)

varie de 1 à 4. Pour une même dérivée appliquée à deux produits différents, la forme générale est la même si la valeur de m pour ces deux produits est identique. En d'autres mots, pour un produit composé du même nombre de fonctions, les composantes de la dérivée sont les mêmes. Par conséquent, les formes générales des dérivées $\frac{\partial^2}{\partial\alpha_1\partial\alpha_3}$ de produits pour 1, 2, 3 et 4 fonction(s) correspondent respectivement à :

$$\frac{\partial^2}{\partial\alpha_1\partial\alpha_3}(f_1) = \frac{1!}{1!} \times \frac{0!}{0!} \times \frac{1!}{1!} f\left(\alpha_1^{(1)} \alpha_2^{(0)} \alpha_3^{(1)}\right), \quad (\text{C.1})$$

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\partial\alpha_3}(f_1 f_2) = \\ \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times f_1\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})}\right) f_2\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})}\right), \end{aligned} \quad (\text{C.2})$$

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\partial\alpha_3}(f_1 f_2 f_3) = \\ \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!} \times \\ f_1\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})}\right) f_2\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})}\right) f_3\left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})}\right), \end{aligned} \quad (\text{C.3})$$

$$\begin{aligned} \frac{\partial^2}{\partial\alpha_1\partial\alpha_3}(f_1 f_2 f_3 f_4) = \\ \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}+k_{1,4}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}+k_{2,4}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}+k_{3,4}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!k_{1,4}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!k_{2,4}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!k_{3,4}!} \times \\ f_1\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})}\right) f_2\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})}\right) f_3\left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})}\right) f_4\left(\alpha_1^{(k_{1,4})} \alpha_2^{(k_{2,4})} \alpha_3^{(k_{3,4})}\right). \end{aligned} \quad (\text{C.4})$$

Par conséquent, nous substituons les dérivées de chaque produit de m fonctions de l'équation de mouvement en x (voir (3.64)) par la formule générale appropriée selon la valeur de m (voir (C.1), (C.2),(C.3) et (C.4)). Nous obtenons alors les sommations de termes de sensibilités $t_{1,k}$ avec $k = 1$ à 11 qui sont définies par les équations (C.7), (C.8), (C.9), (C.10), (C.11), (C.12), (C.13),(C.14), (C.15), (C.16) et (C.17). En assemblant les formulations générales obtenues pour chacune des dérivées contenues dans l'équation de sensibilités

de mouvement en x , nous obtenons :

$$t_{1,1} + t_{1,2} = t_{1,3} + t_{1,4} + t_{1,5} + t_{1,6} + t_{1,7} + t_{1,8} + t_{1,9} + t_{1,10} + t_{1,11}. \quad (\text{C.5})$$

Il en va de même pour l'équation de mouvement en y . La forme générale des dérivées pour chaque produit est donnée par $t_{2,k}$ avec $k = 1$ à 11 des équations (C.18), (C.19), (C.20), (C.21), (C.22), (C.23), (C.24), (C.25), (C.26), (C.27), et (C.28). En se basant sur l'équation (3.65), nous assemblons les formes générales de ces dérivées pour obtenir l'équation de sensibilités du système $\frac{\partial^2}{\partial \alpha_1 \partial \alpha_2}$:

$$t_{2,1} + t_{2,2} = t_{2,3} + t_{2,4} + t_{2,5} + t_{2,6} + t_{2,7} + t_{2,8} + t_{2,9} + t_{2,10} + t_{2,11}. \quad (\text{C.6})$$

$$\begin{aligned}
t_{1,1} &= \frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} \left(\rho u \frac{\partial u}{\partial x} \right) = \\
&\sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!} \times \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) u \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \left(\frac{\partial u}{\partial x} \right) \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right), \\
\end{aligned} \tag{C.7}$$

$$\begin{aligned}
t_{1,2} &= \frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} \left(\rho v \frac{\partial u}{\partial y} \right) = \\
&\sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!} \times \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) v \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \left(\frac{\partial u}{\partial y} \right) \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right), \\
\end{aligned} \tag{C.8}$$

$$t_{1,3} = -\frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} \left(\frac{\partial P}{\partial x} \right) = -\frac{1!}{1!} \times \frac{0!}{0!} \times \frac{1!}{1!} \left(\frac{\partial P}{\partial x} \right) \left(\alpha_1^{(1)} \alpha_2^{(0)} \alpha_3^{(1)} \right), \tag{C.9}$$

$$\begin{aligned}
t_{1,4} &= 2 \frac{\partial^2}{\partial \alpha_1 \partial \alpha_3} \left(\mu \frac{\partial^2 u}{\partial x^2} \right) = 2 \times \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \mu \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) \left(\frac{\partial^2 u}{\partial x^2} \right) \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right), \\
\end{aligned} \tag{C.10}$$

$$t_{1,5} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 v}{\partial x \partial y} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \mu \binom{(k_{1,1}) \alpha_1 \quad (k_{2,1}) \alpha_2 \quad (k_{3,1}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3} \left(\frac{\partial^2 v}{\partial x \partial y} \right) \binom{(k_{1,2}) \alpha_1 \quad (k_{2,2}) \alpha_2 \quad (k_{3,2}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3}, \quad (\text{C.11})$$

$$t_{1,6} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 u}{\partial y^2} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \mu \binom{(k_{1,1}) \alpha_1 \quad (k_{2,1}) \alpha_2 \quad (k_{3,1}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3} \left(\frac{\partial^2 u}{\partial y^2} \right) \binom{(k_{1,2}) \alpha_1 \quad (k_{2,2}) \alpha_2 \quad (k_{3,2}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3}, \quad (\text{C.12})$$

$$t_{1,7} = 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial\mu}{\partial x} \frac{\partial u}{\partial x} \right) = 2 \times \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \left(\frac{\partial\mu}{\partial x} \right) \binom{(k_{1,1}) \alpha_1 \quad (k_{2,1}) \alpha_2 \quad (k_{3,1}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3} \left(\frac{\partial u}{\partial x} \right) \binom{(k_{1,2}) \alpha_1 \quad (k_{2,2}) \alpha_2 \quad (k_{3,2}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3}, \quad (\text{C.13})$$

$$t_{1,8} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial\mu}{\partial y} \frac{\partial v}{\partial x} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \left(\frac{\partial\mu}{\partial y} \right) \binom{(k_{1,1}) \alpha_1 \quad (k_{2,1}) \alpha_2 \quad (k_{3,1}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3} \left(\frac{\partial v}{\partial x} \right) \binom{(k_{1,2}) \alpha_1 \quad (k_{2,2}) \alpha_2 \quad (k_{3,2}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3}, \quad (\text{C.14})$$

$$t_{1,9} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial\mu}{\partial y} \frac{\partial u}{\partial y} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \left(\frac{\partial\mu}{\partial y} \right) \binom{(k_{1,1}) \alpha_1 \quad (k_{2,1}) \alpha_2 \quad (k_{3,1}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3} \left(\frac{\partial u}{\partial y} \right) \binom{(k_{1,2}) \alpha_1 \quad (k_{2,2}) \alpha_2 \quad (k_{3,2}) \alpha_3}{\alpha_1 \quad \alpha_2 \quad \alpha_3}, \quad (\text{C.15})$$

$$t_{1,10} = -\frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (\rho g_x \beta(T - T_0)) = - \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}+k_{1,4}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}+k_{2,4}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}+k_{3,4}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!k_{1,4}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!k_{2,4}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!k_{3,4}!} \times \\ \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) g_x \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \beta \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right) (T - T_0) \left(\alpha_1^{(k_{1,4})} \alpha_2^{(k_{2,4})} \alpha_3^{(k_{3,4})} \right), \quad (\text{C.16})$$

$$t_{1,11} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (f_x) = -\frac{1!}{1!} \times \frac{0!}{0!} \times \frac{1!}{1!} f_x \left(\alpha_1^{(1)} \alpha_2^{(0)} \alpha_3^{(1)} \right), \quad (\text{C.17})$$

$$t_{2,1} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\rho u \frac{\partial v}{\partial x} \right) = \\ \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!} \times \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) u \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \left(\frac{\partial v}{\partial x} \right) \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right), \quad (\text{C.18})$$

$$t_{2,2} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\rho v \frac{\partial v}{\partial y} \right) = \\ \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!} \times \rho \left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right) v \left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right) \left(\frac{\partial v}{\partial y} \right) \left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right), \quad (\text{C.19})$$

$$t_{2,3} = -\frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial P}{\partial y} \right) = -\frac{1!}{1!} \times \frac{0!}{0!} \times \frac{1!}{1!} \left(\frac{\partial P}{\partial y} \right)^{\left(\alpha_1^{(1)} \alpha_2^{(0)} \alpha_3^{(1)} \right)}, \quad (\text{C.20})$$

$$t_{2,4} = 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 v}{\partial y^2} \right) = 2 \times \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \mu^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} \left(\frac{\partial^2 v}{\partial y^2} \right)^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)}, \quad (\text{C.21})$$

$$t_{2,5} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 u}{\partial x \partial y} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \mu^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} \left(\frac{\partial^2 u}{\partial x \partial y} \right)^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)}, \quad (\text{C.22})$$

$$t_{2,6} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\mu \frac{\partial^2 v}{\partial x^2} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \mu^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} \left(\frac{\partial^2 v}{\partial x^2} \right)^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)}, \quad (\text{C.23})$$

$$t_{2,7} = 2 \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial \mu}{\partial y} \frac{\partial v}{\partial y} \right) = 2 \times \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \left(\frac{\partial \mu}{\partial y} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} \left(\frac{\partial v}{\partial y} \right)^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)}, \quad (\text{C.24})$$

$$t_{2,8} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial\mu}{\partial x} \frac{\partial u}{\partial y} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \left(\frac{\partial\mu}{\partial x} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} \left(\frac{\partial u}{\partial y} \right)^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)}, \quad (\text{C.25})$$

$$t_{2,9} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} \left(\frac{\partial\mu}{\partial x} \frac{\partial v}{\partial x} \right) = \sum_{\substack{k_{1,1}+k_{1,2}=1 \\ k_{2,1}+k_{2,2}=0 \\ k_{3,1}+k_{3,2}=1}} \frac{1!}{k_{1,1}!k_{1,2}!} \times \frac{0!}{k_{2,1}!k_{2,2}!} \times \frac{1!}{k_{3,1}!k_{3,2}!} \times \left(\frac{\partial\mu}{\partial x} \right)^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} \left(\frac{\partial v}{\partial x} \right)^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)}, \quad (\text{C.26})$$

$$t_{2,10} = -\frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (\rho g_y \beta (T - T_0)) = - \sum_{\substack{k_{1,1}+k_{1,2}+k_{1,3}+k_{1,4}=1 \\ k_{2,1}+k_{2,2}+k_{2,3}+k_{2,4}=0 \\ k_{3,1}+k_{3,2}+k_{3,3}+k_{3,4}=1}} \frac{1!}{k_{1,1}!k_{1,2}!k_{1,3}!k_{1,4}!} \times \frac{0!}{k_{2,1}!k_{2,2}!k_{2,3}!k_{2,4}!} \times \frac{1!}{k_{3,1}!k_{3,2}!k_{3,3}!k_{3,4}!} \times \rho^{\left(\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \alpha_3^{(k_{3,1})} \right)} g_y^{\left(\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \alpha_3^{(k_{3,2})} \right)} \beta^{\left(\alpha_1^{(k_{1,3})} \alpha_2^{(k_{2,3})} \alpha_3^{(k_{3,3})} \right)} (T - T_0)^{\left(\alpha_1^{(k_{1,4})} \alpha_2^{(k_{2,4})} \alpha_3^{(k_{3,4})} \right)}, \quad (\text{C.27})$$

$$t_{2,11} = \frac{\partial^2}{\partial\alpha_1\partial\alpha_3} (f_y) = -\frac{1!}{1!} \times \frac{0!}{0!} \times \frac{1!}{1!} f_y^{\left(\alpha_1^{(1)} \alpha_2^{(0)} \alpha_3^{(1)} \right)}. \quad (\text{C.28})$$

ANNEXE D FORMULE DE LEIBNIZ

La formule de Leibniz correspond à la dérivée d'un produit de 2 fonctions :

$$(fg)^{(n)} = \sum_{i=0}^n \frac{n!}{i!(n-i)!} f^{(i)} g^{(n-i)}, \quad (\text{D.1})$$

où $n \in \mathbb{N}$, f et g sont des fonctions n fois dérivables et $f^{(0)} = f$ et $g^{(0)} = g$. Comme f et g sont n fois dérivables, alors leur produit l'est également.

Preuve de la formule de Leibniz

Pour prouver que la formule de Leibniz s'applique à l'ordre n , nous faisons une preuve par récurrence. Si $n = 0$, la formule de Leibniz correspond à la dérivée d'ordre 0 d'un produit de fonctions :

$$(fg)^{(0)} = \frac{0!}{0!0!} f^{(0)} g^{(0)}. \quad (\text{D.2})$$

La formule pour $n = 0$ donne le résultat escompté. La proposition $P(0)$ est donc vraie.

On suppose que la formule de Leibniz à l'ordre $n \geq 0$ est vraie. Cela signifie que la proposition $P(n)$ est vraie. On doit donc vérifier si la proposition $P(n+1)$ est vraie. La proposition $P(n+1)$ est vraie, si et seulement si :

$$(fg)^{(n+1)} = \sum_{i=0}^{n+1} \frac{(n+1)!}{i!(n+1-i)!} f^{(i)} g^{(n+1-i)} = \sum_{i=0}^{n+1} \binom{n+1}{i} f^{(i)} g^{(n+1-i)}. \quad (\text{D.3})$$

Vérifions la proposition $P(n+1)$:

$$\begin{aligned} (fg)^{(n+1)} &= \left((fg)^{(n)} \right)^{(1)} = \sum_{i=0}^n \frac{n!}{i!(n-i)!} \left(f^{(i)} g^{(n-i)} \right)^{(1)}, \\ &= \sum_{i=0}^n \frac{n!}{i!(n-i)!} \left(f^{(i+1)} g^{(n-i)} + f^{(i)} g^{(n+1-i)} \right), \\ &= \sum_{i=0}^n \frac{n!}{i!(n-i)!} f^{(i+1)} g^{(n-i)} + \sum_{i=0}^n \frac{n!}{i!(n-i)!} f^{(i)} g^{(n+1-i)}, \\ &= \sum_{i=1}^{n+1} \frac{n!}{(i-1)!(n-i+1)!} f^{(i)} g^{(n-i+1)} + \sum_{i=0}^n \frac{n!}{i!(n-i)!} f^{(i)} g^{(n+1-i)}, \end{aligned}$$

$$= \sum_{i=1}^n \left(\frac{n!}{(i-1)!(n-i+1)!} + \frac{n!}{i!(n-i)!} \right) f^{(i)} g^{(n+1-i)} + \frac{n!}{n!0!} f^{(n+1)} g^{(0)} + \frac{n!}{0!n!} f^{(0)} g^{(n+1)}.$$

Nous remplaçons les coefficients binomiaux $\frac{n!}{(i-1)!(n-i+1)!}$ et $\frac{n!}{i!(n-i)!}$ par leurs notations équivalentes soient respectivement C_{i-1}^n et C_i^n :

$$(fg)^{(n+1)} = \sum_{i=1}^n (C_{i-1}^n + C_i^n) f^{(i)} g^{(n+1-i)} + f^{(n+1)} g^{(0)} + f^{(0)} g^{(n+1)}. \quad (\text{D.4})$$

D'après la formule du triangle de Pascal :

$$C_{i-1}^m + C_i^m = C_i^{m+1}. \quad (\text{D.5})$$

Nous faisons la substitution de $(C_{i-1}^n + C_i^n)$ par C_i^{n+1} dans l'équation (D.4) :

$$(fg)^{(n+1)} = \sum_{i=1}^n (C_i^{n+1}) f^{(i)} g^{(n+1-i)} + f^{(n+1)} g^{(0)} + f^{(0)} g^{(n+1)}, \quad (\text{D.6})$$

$$(fg)^{(n+1)} = \sum_{i=0}^{n+1} (C_i^{n+1}) f^{(i)} g^{(n+1-i)}. \quad (\text{D.7})$$

Nous pouvons donc conclure que la proposition $P(n+1)$ est vraie. Cela démontre que la formule de Leibniz est vraie $\forall n \in \mathbb{N}$.

Il est possible que les fonctions f et g soient composées d'un produit de plusieurs fonctions. La formule de Leibniz demeure néanmoins valide. Supposons que $f = f_1$ et $g = f_2 f_3 \dots f_m$ pour m un entier positif. Nous appliquons alors la règle de Leibniz de façon imbriquée :

$$\begin{aligned} (fg)^{(n)} &= (f_1 f_2 \dots f_m)^{(n)} \\ &= \sum_{k_1=0}^n \frac{n!}{k_1!(n-k_1)!} f_1^{(k_1)} g^{(n-k_1)} \\ &= \sum_{k_1=0}^n \frac{n!}{k_1!(n-k_1)!} f_1^{(k_1)} \sum_{k_2=0}^{n-k_1} \frac{(n-k_1)!}{k_2!(n-k_1-k_2)!} f_2^{k_2} \times \\ &\quad \sum_{k_3=0}^{n-k_1-k_2} \frac{(n-k_1-k_2)!}{k_3!(n-k_1-k_2-k_3)!} f_3^{k_3} \times \dots \times \\ &\quad \sum_{k_{m-1}=0}^{n-k_1-k_2-\dots-k_{m-2}} \frac{(n-k_1-k_2-\dots-k_{m-2})!}{k_{m-1}!((n-k_1-k_2-\dots-k_{m-2}-k_{m-1})!)} f_{m-1}^{(k_{m-1})} f_m^{(n-k_1-k_2-\dots-k_{m-1})}. \end{aligned}$$

On pose $k_m = n - k_1 - k_2 - \dots - k_{m-1}$. Puis, on simplifie les coefficients en sortant de l'opérateur de sommation ceux qui ne dépendent pas de l'indice de ce dernier :

$$(fg)^{(n)} = \sum_{k_1=0}^n \frac{n!}{k_1!(n-k_1)!} f_1^{(k_1)} \cancel{(n-k_1)!} \sum_{k_2=0}^{n-k_1} \frac{1}{k_2!(n-k_1-k_2)!} f_2^{k_2} \times$$

$$\cancel{(n-k_1-k_2)!} \sum_{k_3=0}^{n-k_1-k_2} \frac{1}{k_3!(n-k_1-k_2-k_3)!} f_3^{k_3} \times \dots \times$$

$$\cancel{(n-k_1-k_2-\dots-k_{m-2})!} \sum_{k_{m-1}=0}^{n-k_1-k_2-\dots-k_{m-2}} \frac{1}{k_{m-1}!(k_m)!} f_{m-1}^{(k_{m-1})} f_m^{(k_m)}.$$

On obtient donc l'équation suivante :

$$(fg)^{(n)} = \sum_{k_1=0}^n \frac{n!}{k_1!} f_1^{(k_1)} \sum_{k_2=0}^{n-k_1} \frac{1}{k_2!} f_2^{(k_2)} \sum_{k_3=0}^{n-k_1-k_2} \frac{1}{k_3!} f_3^{(k_3)} \dots \sum_{k_{m-1}=0}^{n-k_1-k_2-\dots-k_{m-2}} \frac{1}{k_{m-1}!k_m!} f_{m-1}^{(k_{m-1})} f_m^{(k_m)}.$$

Nous regroupons toutes les composantes de l'équation précédente à l'intérieur de l'opérateur de sommation interne. Nous obtenons alors :

$$(fg)^{(n)} = \sum_{k_1=0}^n \sum_{k_2=0}^{n-k_1} \sum_{k_3=0}^{n-k_1-k_2} \dots \sum_{k_{m-1}=0}^{n-k_1-k_2-\dots-k_{m-2}} \frac{n!}{k_1!k_2!k_3! \dots k_{m-1}!k_m!} f_1^{(k_1)} f_2^{(k_2)} f_3^{(k_3)} \dots f_{m-1}^{(k_{m-1})} f_m^{(k_m)}, \quad (\text{D.8})$$

où $k_m = n - k_1 - k_2 - \dots - k_{m-1}$.

La multiplication des opérateurs de sommations dans l'équation (D.8) permet d'évaluer tous les ensembles de k_i pour $i = 1$ à m tels que $\sum_{i=1}^m k_i = n$. Nous pouvons alors réécrire l'équation (D.8) comme suit :

$$\sum_{\substack{k_1, k_2, \dots, k_m \in \mathbb{N} \\ k_1 + k_2 + \dots + k_m = n}} \frac{n!}{k_1!k_2!k_3! \dots k_{m-1}!k_m!} f_1^{(k_1)} f_2^{(k_2)} f_3^{(k_3)} \dots f_{m-1}^{(k_{m-1})} f_m^{(k_m)}. \quad (\text{D.9})$$

Remarquons que cette équation est équivalente à l'expression générale de la dérivée d'ordre n_i par rapport à un paramètre α_i d'un produit de m fonctions que nous avons développée (voir équation (3.43)). Ainsi, la formule générale de la dérivée d'ordre n du produit de m fonctions découle de la formule de Leibniz qui est valide $\forall n \in \mathbb{N}$.

ANNEXE E SOUS-ROUTINE EXPLICITE ÉQUIVALENTE À LA SOUS-ROUTINE DE L'ALGORITHME 4

Dans cette annexe, nous détaillons la sous-routine explicite sur laquelle nous nous sommes basés pour développer la sous-routine récursive *multinomeNewton* (voir algorithme 4). Cela permet de faciliter la compréhension du processus de base de la sous-routine *multinomeNewton*. L'algorithme 5 est la forme explicite équivalente à l'algorithme 4. Dans l'algorithme 5, nous remarquons que la valeur de $k_{i,j}$ (*i.e.* : ordre de la dérivée de la j -ième fonction par rapport au paramètre α_i) dépend des valeurs des autres $k_{i,l}$ où $l > j$ de sorte à obtenir $k_{i,j} = n_i - k_{i,m} - \dots - k_{i,j+2} - k_{i,j+1}$. Ces boucles imbriquées déterminent toutes les combinaisons possibles des ordres $k_{i,j}$, pour $i = 1$ à q , de sorte à ce que $\sum_{j=1}^m k_{i,j} = n_i$. La sous-routine récursive dans l'algorithme 4 exécute le même procédé, mais de façon récursive. En fait, l'algorithme 4 appelle récursivement les boucles **Pour** que l'on retrouve dans l'algorithme 5.

Exemple du fonctionnement de l'algorithme 5

Nous expliquons brièvement le fonctionnement des boucles imbriquées de l'algorithme 5. Par exemple, si nous avons un terme formé du produit de 3 fonctions, soit $f_1 f_2 f_3$, et un système de sensibilités avec 2 paramètres, nous avons un total de 4 boucles ($m = 3$ et $q = 2$). Nous cherchons les composantes des termes produits par la dérivée $\frac{\partial^3}{\partial \alpha_1 \partial \alpha_2^2}$ appliquée à ce produit de 3 fonctions. Pour ce faire, nous pouvons utiliser un système de boucles imbriquées tel qu'illustré par l'algorithme 6.

Pour le premier passage dans les boucles de l'algorithme 6 nous avons :

$$\begin{aligned} k_{1,1} &= 0; & k_{1,2} &= 0; & k_{1,3} &= 1; \\ k_{2,1} &= 0; & k_{2,2} &= 0; & k_{2,3} &= 2. \end{aligned}$$

Par la suite, l'algorithme met à jour le tableau d et le vecteur c :

$$d(:, :, 1) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 2 \end{bmatrix}; \quad c(1) = 1.$$

Algorithme 5 : Boucle équivalente à la sous-routine récursive *multinomeNewton*

Entrée: n**Sortie:** $d(:, :, :)$ et $c(:, :)$ **Initialisation;** $d(1 : m, 1 : q, 1 : s) \leftarrow 0$ $c(1 : s) \leftarrow (n(1)! \cdot n(2)! \cdots n(q)!)$ $compteur \leftarrow 0$ **pour** $k_{q,m} = n_q$ **à 0 faire****pour** $k_{q,m-1} = (n_q - k_{q,m})$ **à 0 faire**

...

pour $k_{q,2} = (n_q - k_{q,m} - k_{q,m-1} - \dots - k_{q,3})$ **to 0 faire****pour** $k_{q-1,m} = n_{q-1}$ **à 0 faire****pour** $k_{q-1,m-1} = (n_{q-1} - k_{q-1,m})$ **à 0 faire**

...

pour $k_{q-1,2} = (n_{q-1} - k_{q-1,m} - k_{q-1,m-1} - \dots - k_{q-1,3})$ **à 0 faire**

...

pour $k_{1,m} = n_1$ **à 0 faire****pour** $k_{1,m-1} = (n_1 - k_{1,m})$ **à 0 faire**

...

pour $k_{1,2} = (n_1 - k_{1,m} - k_{1,m-1} - \dots - k_{1,3})$ **à 0 faire** $compteur \leftarrow compteur + 1$ $k_{q,1} = n_q - k_{q,m} - k_{q,m-1} - \dots - k_{q,2}$ $k_{q-1,1} = n_{q-1} - k_{q-1,m} - k_{q-1,m-1} - \dots - k_{q-1,2}$ \vdots $k_{1,1} = n_1 - k_{1,m} - k_{1,m-1} - \dots - k_{1,2}$ $i \leftarrow 0$ **tant que** $(i < q)$ **faire** $i \leftarrow i + 1$ $j \leftarrow 0$ **tant que** $(j < m)$ **faire** $j \leftarrow j + 1$ $d(j, i, compteur) = k_{i,j}$ $c(compteur) \leftarrow \frac{c(compteur)}{k_{i,j}!}$ **fin****fin****fin****fin****fin****fin****fin****fin****fin****fin****fin**

Algorithme 6 : Boucles imbriquées pour calculer les composantes des termes de sensibilités ($m = 3$ et $q = 2$)

Entrée: n

Sortie: $d(:, :, :)$ et $c(:)$

Initialisation;

$c(1 : s) \leftarrow (1! \cdot 2!)$

$compteur \leftarrow 0$

pour $k_{2,3} = 2$ à 0 **faire**

pour $k_{2,2} = (2 - k_{2,3})$ à 0 **faire**

pour $k_{1,3} = 1$ à 0 **faire**

pour $k_{1,2} = (1 - k_{1,3})$ à 0 **faire**

$compteur \leftarrow compteur + 1$

$k_{2,1} = 2 - k_{2,3} - k_{2,2}$

$k_{1,1} = 1 - k_{1,3} - k_{1,2}$

$i \leftarrow 0$

tant que $(i < 2)$ **faire**

$i \leftarrow i + 1$

$j \leftarrow 0$

tant que $(j < 3)$ **faire**

$j \leftarrow j + 1$

$d(j, i, compteur) = k_{i,j}$

$c(compteur) \leftarrow \frac{c(compteur)}{k_{i,j}!}$

fin

fin

fin

fin

fin

fin

À la seconde itération, nous obtenons :

$$\begin{aligned} k_{1,1} &= 0; & k_{1,2} &= 1; & k_{1,3} &= 0; \\ k_{2,1} &= 0; & k_{2,2} &= 0; & k_{2,3} &= 2. \end{aligned}$$

Cela implique que :

$$d(:, :, 2) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 2 \end{bmatrix}; \quad c(2) = 1;$$

où la première ligne (*i.e.* : $d(1, :, 2)$) correspond à la deuxième colonne de $d(1, :, :)$, la deuxième ligne (*i.e.* : $d(2, :, 2)$) correspond à la deuxième colonne de $d(2, :, :)$ et la troisième ligne (*i.e.* : $d(3, :, 2)$) correspond à la deuxième colonne de $d(3, :, :)$.

À la fin des itérations des boucles, nous obtenons les tableaux :

$$d(1, :, :) = \begin{bmatrix} 0 & 0 & 1 & 0 & \boxed{0} & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \boxed{0} & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \end{bmatrix};$$

$$d(2, :, :) = \begin{bmatrix} 0 & 1 & 0 & 0 & \boxed{1} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & \boxed{1} & 1 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix};$$

$$d(3, :, :) = \begin{bmatrix} 1 & 0 & 0 & 1 & \boxed{0} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & \boxed{1} & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$c^T = [1 \quad 1 \quad 1 \quad 2 \quad \boxed{2} \quad 2 \quad 2 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1].$$

Nous avons au total 18 termes de sensibilités pour les sensibilités d'ordre 3 (système $\frac{\partial^3}{\partial \alpha_1 \partial \alpha_2^2}$) du produit de 3 fonctions. On associe les valeurs $d(j, i, k)$ du tableau d à l'ordre de la dérivée partielle par rapport au paramètre α_i pour la j -ième fonction du k -ième terme des sensibilités. Ainsi, pour les tableaux $d(:, :, :)$ donnés précédemment, cela signifie que la colonne k du tableau $d(j, :, k)$ donne les ordres des dérivées partielles par rapport aux paramètres α_i pour la fonction f_j du k -ième terme de sensibilités pour $i = 1$ à 2. Inversement, les lignes de ces

tableaux, soient $d(j, i, :)$ indique les ordres de dérivées partielles d'une fonction f_j par rapport à un paramètre particulier α_i pour l'ensemble des 18 termes de sensibilités. À titre d'exemple, le cinquième terme des sensibilités ($k = 5$) est :

$$2f_1 \frac{\partial f_2}{\partial \alpha_1} \frac{\partial f_2}{\partial \alpha_2} \frac{\partial f_3}{\partial \alpha_2}.$$

En effet, nous avons $d(1, :, 5) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $d(2, :, 5) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $d(3, :, 5) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ et $c(5) = 2$ qui désignent respectivement les valeurs des ordres des dérivées partielles par rapport aux paramètres α_1 et α_2 pour les fonctions f_1 , f_2 et f_3 du cinquième terme découlant de la dérivée $\frac{\partial^3}{\partial \alpha_1 \partial \alpha_2^2}$ ainsi que la valeur du coefficient devant ce même terme.

ANNEXE F INTÉGRATION NUMÉRIQUE

Dans cette annexe, nous faisons un rappel de l'intégration numérique gaussienne. Nous y exposons également les formules d'intégration numérique pour les résidus dont l'intervalle d'intégration correspond à l'élément de référence.

Le tableau F.1 résume l'essentiel de la quadrature de Gauss (*i.e.* : point d'intégration, poids d'intégration, degré de précision). Par exemple l'intégrale d'une fonction quelconque $g(x)$ sur le domaine $[-1, 1]$ (élément de référence en $1D$) s'évalue par une expression de la forme :

$$\int_{-1}^1 g(x)dx = \sum_{k=1}^{npg} g(xe_k)w_k \quad (\text{F.1})$$

où xe_k est un point d'intégration, w_k est le poids d'intégration au k -ième point de Gauss et npg est le nombre de points de Gauss sur lesquels à lieu la quadrature. Nous évaluons la position physique du point de Gauss sur l'élément fini (*i.e.* : xe_k) comme suit :

$$xe_k = \sum_{i=1}^{ndp} x_i \phi_{i,k}, \quad (\text{F.2})$$

où x_i est la valeur de x au noeud numéro i de l'élément fini et $\phi_{i,k}$ est la fonction d'interpolation i au k -ième point de Gauss.

Tableau F.1 Intégration numérique sur l'intervalle de référence (voir Fortin et Garon, 2014).

Quadrature de Gauss (1D)			
m_G	Points d'intégration ξ_i	Poids d'intégration w_i	Degré de précision
1	0	2	1
2	-0,577 350 269 189 625 +0,577 350 269 189 625	1 1	3
3	-0,774 596 669 241 483 0,0 +0,774 596 669 241 483	0,555 555 555 555 556 0,888 888 888 888 889 0,555 555 555 555 556	5
4	-0,861 136 311 594 052 -0,339 981 043 584 856 +0,339 981 043 584 856 +0,861 136 311 594 052	0,347 854 845 137 454 0,652 145 154 862 545 0,652 145 154 862 545 0,347 854 845 137 454	7
5	-0,906 179 845 938 664 -0,538 469 310 105 683 0,0 +0,538 469 310 105 683 +0,906 179 845 938 664	0,236 926 885 056 189 0,478 628 670 499 365 0,568 888 889 888 889 0,478 628 670 499 365 0,236 926 885 056 189	9

Pour effectuer l'intégration sur l'élément de référence, nous devons remplacer $d\Omega$ par $J d\Omega_0$. Cela explique pourquoi le jacobien J_k apparaît dans les formules d'intégration numérique qui suivent. Les résidus élémentaires du problème 1D sont définis par les équations (4.40) et (4.41). Pour résoudre les sensibilités du problème test 1D, l'intégrale numérique pour le

résidu $Res_{\Omega_e}^i(1)$ de l'équation de sensibilités de u pour le système $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ s'écrit :

$$\begin{aligned}
& \sum_{k=1}^{npg} \left[\left(u_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q}}{dx} + u_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q}}{dx} \right) \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{\substack{k_{1,1}+k_{1,2}=n_1 \\ k_{2,1}+k_{2,2}=n_2 \\ \vdots \\ k_{q,1}+k_{q,2}=n_q}} \frac{n_1!}{k_{1,1}!k_{1,2}!} \frac{n_2!}{k_{2,1}!k_{2,2}!} \dots \frac{n_q!}{k_{q,1}!k_{q,2}!} u_k \binom{(k_{1,1}) \dots (k_{q,1})}{\alpha_1 \dots \alpha_q} \times \right. \\
& \quad \times \frac{du_k \binom{(k_{1,2}) \dots (k_{q,2})}{\alpha_1 \dots \alpha_q}}{dx} \phi_i(x_k) - \\
& \quad \left. \left(u_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q}}{dx} + u_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q}}{dx} \right) \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\beta_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q} T_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q} \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{\substack{k_{1,1}+k_{1,2}=n_1 \\ k_{2,1}+k_{2,2}=n_2 \\ \vdots \\ k_{q,1}+k_{q,2}=n_q}} \frac{n_1!}{k_{1,1}!k_{1,2}!} \frac{n_2!}{k_{2,1}!k_{2,2}!} \dots \frac{n_q!}{k_{q,1}!k_{q,2}!} \beta_k \binom{(k_{1,1}) \dots (k_{q,1})}{\alpha_1 \dots \alpha_q} \times \right. \\
& \quad \times T_k \binom{(k_{1,2}) \dots (k_{q,2})}{\alpha_1 \dots \alpha_q} \phi_i(x_k) - \beta_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q} T_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q} \phi_i(x_k) \left. \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\nu_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q}}{dx} \cdot \nabla \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{\substack{k_{1,1}+k_{1,2}=n_1 \\ k_{2,1}+k_{2,2}=n_2 \\ \vdots \\ k_{q,1}+k_{q,2}=n_q}} \frac{n_1!}{k_{1,1}!k_{1,2}!} \frac{n_2!}{k_{2,1}!k_{2,2}!} \dots \frac{n_q!}{k_{q,1}!k_{q,2}!} \nu_k \binom{(k_{1,1}) \dots (k_{q,1})}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(k_{1,2}) \dots (k_{q,2})}{\alpha_1 \dots \alpha_q}}{dx} \right. \\
& \quad \left. - \nu_k \binom{(0) \dots (0)}{\alpha_1 \dots \alpha_q} \frac{du_k \binom{(n_1) \dots (n_q)}{\alpha_1 \dots \alpha_q}}{dx} \right] \cdot \nabla \phi_i(x_k) J_k w_k,
\end{aligned}
\tag{F.3}$$

où $i = 1$ à ndp (ndp indique le nombre de noeuds d'un élément) et la limite npg de l'opérateur $\sum_{k=1}^{npg}$ est le nombre de points de Gauss utilisés pour l'intégration numérique. C'est l'usager qui choisit la valeur de npg . Dans l'équation (F.3) :

- ϕ_i est la fonction d'interpolation ;
- $u_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$ est la sensibilité d'ordre $n_1 + n_2 + \dots + n_q = n$ de la variable u avec q paramètres ;
- $T_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$ est la sensibilité d'ordre n de la variable T avec q paramètres ;
- x_k est le point d'intégration ;
- l'indice k utilisé pour les variables u et T fait référence au point x_k (k -ième point de Gauss) ;
- J_k est le jacobien au point d'intégration x_k ;
- w_k est le poids d'intégration ;
- l'opérateur $\sum_{\substack{k_{1,1}+k_{1,2}+\dots+k_{1,m}=n_1 \\ k_{2,1}+k_{2,2}+\dots+k_{2,m}=n_2 \\ \dots \\ k_{q,1}+k_{q,2}+\dots+k_{q,m}=n_q}}$ fonctionne façon similaire à celui de l'équation (3.55).

En l'occurrence, l'opérateur $\sum_{\substack{k_{1,1}+k_{1,2}+\dots+k_{1,m}=n_1 \\ k_{2,1}+k_{2,2}+\dots+k_{2,m}=n_2 \\ \dots \\ k_{q,1}+k_{q,2}+\dots+k_{q,m}=n_q}}$ fait la somme des termes sur tous les sous-ensembles de $k_{i,j}$ tels que $\sum_{j=1}^m = n_i$ avec $i = 1$ à q et où m est le nombre de fonctions des produits de l'EDP (ici, $m = 2$). Cet opérateur permet donc d'ajouter au résidu tous les termes de sensibilités qui découlent de la dérivée $\frac{\partial^{n_1+n_2+\dots+n_q}}{\partial \alpha_1^{n_1} \partial \alpha_2^{n_2} \dots \partial \alpha_q^{n_q}}$ appliquée à un produit de m fonctions. Pour les opérateurs de sommation s'effectuant sur les sous-ensembles de $k_{i,j}$, nous avons soustrait les termes faisant intervenir les sensibilités d'ordre n des variables dépendantes (*i.e.* : $u_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$, $\frac{du_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}}{dx}$ et $T_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$), car ceux-ci sont déjà ajoutés à l'extérieur de l'opérateur de sommation. Cela évite de les comptabiliser en double. Il est à noter que ces termes correspondent aux éléments du membre de gauche puisqu'ils font intervenir directement les sensibilités $S_u^{(n)}$ et $S_T^{(n)}$.

Dans le but d'alléger l'équation (F.3), nous l'écrivons sous une notation différente :

$$\begin{aligned}
& \sum_{k=1}^{npg} \left[\left(S_u^{(noSyst)}(x_k) \frac{du(x_k)}{dx} + u(x_k) \frac{dS_u^{(noSyst)}(x_k)}{dx} + \beta(x_k) S_T^{(noSyst)}(x_k) \right) \phi_i(x_k) + \right. \\
& \quad \left. \nu(x_k) \frac{dS_u^{(noSyst)}(x_k)}{dx} \cdot \nabla \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{t=1}^{s-2} c(t) u^{(NoComb(1,t))}(x_k) \frac{du^{(NoComb(2,t))}(x_k)}{dx} \phi_i(x_k) + \right. \\
& \quad \sum_{t=1}^{s-1} c(t) \beta^{(NoComb(1,t))}(x_k) T^{(NoComb(2,t))}(x_k) \phi_i(x_k) + \\
& \quad \left. \sum_{t=1}^{s-1} c(t) \nu^{(NoComb(1,t))}(x_k) \frac{du^{(NoComb(2,t))}(x_k)}{dx} \cdot \nabla \phi_i(x_k) \right] J_k w_k, \tag{F.4}
\end{aligned}$$

où $S_u^{(noSyst)}$ et $S_T^{(noSyst)}$ équivalent respectivement à $u_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$ et à $T_k^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}$.

L'exposant $(noSyst)$ fait référence au numéro de la combinaison associée au système suivant : $\frac{\partial^n}{\partial \alpha_1^{(n_1)} \partial \alpha_2^{(n_2)} \dots \partial \alpha_q^{(n_q)}}$. L'exposant $(NoComb(j, t))$, pour $j = 1$ à 2 , affecté à une fonction indique

la sensibilité de cette même fonction. La sommation $\sum_{t=1}^{s-2}$ additionne les termes de sensibilités pour lesquels l'ordre des dérivées associées à $(NoComb(j, t))$, pour $j = 1$ à 2 , est inférieur à n , sachant que $n = n_1 + n_2 + \dots + n_q$. La limite $s - 2$ de l'opérateur de sommation indique que, parmi les termes de sensibilités découlant d'une dérivée d'ordre n appliquée au produit $u \frac{du}{dx}$, il y en a 2 qui impliquent les sensibilités d'ordre n des variables dépendantes inconnues. Comme ceux-ci sont également ajoutés à l'extérieur de l'opérateur de sommation, cela évite de les comptabiliser en double. De plus, l'opérateur $\sum_{t=1}^{s-1}$ procède de la même manière.

Toutefois, pour les produits βT et $\nu \frac{du}{dx}$, il n'y a qu'un seul terme de sensibilités dans lequel on retrouve la $n^{ième}$ sensibilité d'une variable inconnue, puisque ces produits sont constitués de 2 fonctions chacun et que parmi celles-ci une seule correspond à une variable dépendante inconnue.

De façon similaire, le résidu $Res_{\Omega_e}^i(2)$ pour l'équation de sensibilités de T est évalué par :

$$\begin{aligned}
& \sum_{k=1}^{npg} \left[\left(u_k \binom{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}} \cdot \nabla T_k \binom{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}}{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}} + \right. \right. \\
& \quad \left. \left. u_k \binom{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}}{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}} \cdot \nabla T_k \binom{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}} \right) \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{\substack{k_{1,1}+k_{1,2}=n_1 \\ k_{2,1}+k_{2,2}=n_2 \\ \vdots \\ k_{q,1}+k_{q,2}=n_q}} \frac{n_1!}{k_{1,1}!k_{1,2}!} \frac{n_2!}{k_{2,1}!k_{2,2}!} \dots \frac{n_q!}{k_{q,1}!k_{q,2}!} \times \right. \\
& \quad \times u_k \binom{\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \dots \alpha_q^{(k_{q,1})}}{\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \dots \alpha_q^{(k_{q,2})}} \cdot \nabla T_k \binom{\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \dots \alpha_q^{(k_{q,2})}}{\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \dots \alpha_q^{(k_{q,1})}} \phi_i(x_k) - \\
& \quad \left(u_k \binom{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}} \cdot \nabla T_k \binom{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}}{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}} + \right. \\
& \quad \left. u_k \binom{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}}{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}} \cdot \nabla T_k \binom{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}} \right) \phi_i(x_k) \left. \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[D_k \binom{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}}{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}} \nabla T_k \binom{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}} \cdot \nabla \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{\substack{k_{1,1}+k_{1,2}=n_1 \\ k_{2,1}+k_{2,2}=n_2 \\ \vdots \\ k_{q,1}+k_{q,2}=n_q}} \frac{n_1!}{k_{1,1}!k_{1,2}!} \frac{n_2!}{k_{2,1}!k_{2,2}!} \dots \frac{n_q!}{k_{q,1}!k_{q,2}!} \times \right. \\
& \quad \times D_k \binom{\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \dots \alpha_q^{(k_{q,1})}}{\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \dots \alpha_q^{(k_{q,2})}} \nabla T_k \binom{\alpha_1^{(k_{1,2})} \alpha_2^{(k_{2,2})} \dots \alpha_q^{(k_{q,2})}}{\alpha_1^{(k_{1,1})} \alpha_2^{(k_{2,1})} \dots \alpha_q^{(k_{q,1})}} \cdot \nabla \phi_i(x_k) - \\
& \quad \left. D_k \binom{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}}{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}} \nabla T_k \binom{\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)}}{\alpha_1^{(0)} \alpha_2^{(0)} \dots \alpha_q^{(0)}} \cdot \nabla \phi_i(x_k) \right] J_k w_k.
\end{aligned} \tag{F.5}$$

La notation simplifiée pour l'équation (F.5) est la suivante :

$$\begin{aligned}
& \sum_{k=1}^{npg} \left[\left(S_u^{(noSyst)}(x_k) \cdot \nabla T(x_k) + u(x_k) \cdot \nabla S_T^{(noSyst)}(x_k) \right) \phi_i(x_k) + \right. \\
& \quad \left. D(x_k) \nabla S_T^{(noSyst)}(x_k) \cdot \nabla \phi_i(x_k) \right] J_k w_k + \\
& \sum_{k=1}^{npg} \left[\sum_{t=1}^{s-2} c(t) u^{(NoComb(1,t))}(x_k) \cdot \nabla T^{(NoComb(2,t))}(x_k) \phi_i(x_k) + \right. \\
& \quad \left. \sum_{t=1}^{s-1} c(t) D^{(NoComb(1,t))}(x_k) \nabla T^{(NoComb(2,t))}(x_k) \cdot \nabla \phi_i(x_k) \right] J_k w_k.
\end{aligned} \tag{F.6}$$

ANNEXE G ESPACES FONCTIONNELS

Dans cette annexe, nous abordons certaines notions sur les espaces fonctionnels. En mathématiques, un espace fonctionnel est défini comme un ensemble de fonctions (ou d'applications) d'un ensemble X vers un ensemble Y .

Un des espaces fonctionnels courants est l'espace $L^2(\Omega)$ défini comme suit :

$$L^2(\Omega) = \{u : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} (u(\mathbf{x}))^2 dv < \infty\}. \quad (\text{G.1})$$

L'espace fonctionnel $L^2(\Omega)$ est linéaire. Cela signifie que :

- Si $u \in L^2(\Omega)$ et $\beta \in \mathbb{N}$, alors $\beta u \in L^2(\Omega)$,
- Si $u \in L^2(\Omega)$ et $w \in L^2(\Omega)$, alors $(u + w) \in L^2(\Omega)$.

On introduit une norme pour mesurer la grandeur d'une fonction sur les espaces fonctionnels. Cette norme est d'une très grande utilité lors de l'analyse de convergence des MEF. Nous définissons d'abord le produit scalaire sur un espace fonctionnel linéaire S . Il s'agit en fait d'une fonction définie sur $S \times S$ qui associe un scalaire noté $((u, w))_S$ à un couple (u, w) de $S \times S$ (voir Fortin et Garon, 2014). Ce scalaire vérifie les propriétés suivantes :

1. $((u, w))_S = ((w, u))_S, \forall u, w \in S$ (commutativité) ;
2. $((\beta u, w))_S = ((u, \beta w))_S = \beta((u, w))_S, \forall \beta \in \mathbb{R}$ et $\forall u, w \in S$;
3. $((u_1 + u_2, w))_S = ((u_1, w))_S + ((u_2, w))_S \forall u_1, u_2, w \in S$ (additivité) ;
4. $((u, u))_S \geq 0$, le produit scalaire étant égal à 0 si et seulement si $u = 0$.

Par exemple, pour $(u, w) \in L^2(\Omega) \times L^2(\Omega)$, le produit scalaire est :

$$((u, w))_{0,\Omega} = \int_{\Omega} u w d\Omega. \quad (\text{G.2})$$

Les 4 propriétés précédentes peuvent être vérifiées.

Une norme consiste en une application de S dans \mathbb{R} qui, à une fonction $u \in S$, associe un scalaire que nous notons $\|u\|_S$ satisfaisant les 3 propriétés suivantes :

1. $\|u\|_S > 0, \forall u \in S ; \|u\|_S = 0 \iff u = 0 ;$

2. $\|\beta u\|_S = |\beta| \|u\|_S = \beta((u, w))_S, \forall \beta \in \mathbb{R} \text{ et } \forall u \in S;$
3. $\|u + w\|_S \leq \|u\|_S + \|w\|_S \forall u, w \in S$ (inégalité triangulaire).

Si un espace fonctionnel linéaire possède un produit scalaire, alors il possède une norme dite induite :

$$\|u\|_S = ((u, u))_S^{1/2}. \quad (\text{G.3})$$

Par exemple, pour le produit scalaire sur l'espace $L^2(\Omega)$ (voir équation (G.2)), la norme induite est :

$$\|u, u\|_{0,\Omega} = ((u, u))_{0,\Omega}^{1/2} = \left(\int_{\Omega} u^2 d\Omega \right)^{1/2}. \quad (\text{G.4})$$

Pour ce travail, l'espace fonctionnel utilisé pour l'analyse de convergence est l'espace $H^1(\Omega)$. Cet espace est linéaire. On le définit comme suit :

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \mid \frac{\partial u}{\partial x_i} \in L^2(\Omega), 1 \leq i \leq 3 \right\}, \quad (\text{G.5})$$

où x_i correspond à la i -ième coordonnée selon la dimension du problème.

On munit cet espace du produit scalaire $((u, w))_{1,\Omega}$:

$$((u, w))_{1,\Omega} = \int_{\Omega} \left(uw + \sum_{i=1}^n \frac{\partial u}{\partial x_i} \frac{\partial w}{\partial x_i} \right) d\Omega, \quad (\text{G.6})$$

où $\frac{\partial u}{\partial x_i}$ et $\frac{\partial w}{\partial x_i}$ sont les dérivées partielles par rapport aux coordonnées spatiales. Notons que $i = 1$ à 3 , pour un problème $3D$, mais que $i = 1$ à 2 pour un problème $2D$ et $i = 1$ pour un problème $1D$.

La norme induite dans l'espace $H^1(\Omega)$ est :

$$\|u\|_{1,\Omega} = ((u, u))_{1,\Omega}^{1/2} = \left(\int_{\Omega} \left(u^2 + \sum_{i=1}^3 \left(\frac{\partial u}{\partial x_i} \right)^2 \right) d\Omega \right)^{1/2}, \quad (\text{G.7})$$

où x_i correspond à la coordonnée dans un espace $3D$. En $3D$, les variables x_1, x_2 et x_3 représentent respectivement les coordonnées selon les axes x, y et z . Pour les problèmes $1D$, les coordonnées x_2 et x_3 sont inexistantes. Pour les problèmes $2D$, la coordonnée x_2 correspond à la variable y dans le plan cartésien, alors que nous ne considérons pas la coordonnée x_3 .

Il existe également un espace $H_0^1(\Omega)$, similaire à l'espace $H^1(\Omega)$, mais dont les fonctions s'annulent au bord. On écrit cet espace comme suit :

$$H_0^1(\Omega) = \{w \in H^1(\Omega) \mid w = 0 \text{ sur } \Gamma\} \quad (\text{G.8})$$

On peut également avoir $H_{\Gamma_0}^1(\Omega)$. Dans ce cas, les fonctions s'annulent uniquement sur Γ_0 qui représente une partie de la frontière Γ du domaine Ω . On définit cet espace :

$$H_{\Gamma_0}^1(\Omega) = \{w \in H^1(\Omega) \mid w = 0 \text{ sur } \Gamma_0\} \quad (\text{G.9})$$

On note alors la norme sur les espaces $H_0^1(\Omega)$ et $H_{\Gamma_0}^1(\Omega)$:

$$|u|_{1,\Omega} = \left(\int_{\Omega} \left(\sum_{i=1}^n \frac{\partial u}{\partial x_i} \right)^2 d\Omega \right) \quad (\text{G.10})$$

où n correspond à la dimension du problème.

ANNEXE H TABLEAUX DES RÉSULTATS

Dans cette annexe, nous présentons sous forme de tableaux les données numériques obtenues lors des diverses étapes de vérification et validation : MSM et comparaison avec la méthode des DF.

Résultats pour la Méthode des Solutions Manufacturées dans le cas avec 1 paramètre

Tableau H.1 Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P1$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(0)})} \ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(0)})} \ $
41	1.255E-02	2.242E-02
81	6.274E-03	1.115E-02
161	3.137E-03	5.564E-03
321	1.568E-03	2.781E-03
641	7.842E-04	1.390E-03
1281	3.921E-04	6.952E-04
2561	1.961E-04	3.476E-04
5121	9.803E-05	1.738E-04
10241	4.901E-05	8.690E-05

Tableau H.2 Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P1$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(1)})} \ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(1)})} \ $
41	2.174E-02	2.611E-02
81	1.087E-02	1.297E-02
161	5.435E-03	6.478E-03
321	2.717E-03	3.238E-03
641	1.359E-03	1.619E-03
1281	6.793E-04	8.093E-04
2561	3.397E-04	4.047E-04
5121	1.698E-04	2.023E-04
10241	8.492E-05	1.012E-04

Tableau H.3 Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P1$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{u}_{\text{h}}^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{T}_{\text{h}}^{(\alpha_1^{(2)})}\ $
41	2.188E-02	2.044E-02
81	1.094E-02	1.004E-02
161	5.468E-03	5.000E-03
321	2.734E-03	2.497E-03
641	1.367E-03	1.248E-03
1281	6.834E-04	6.241E-04
2561	3.417E-04	3.120E-04
5121	1.709E-04	1.560E-04
10241	8.543E-05	7.801E-05

Tableau H.4 Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P1$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{u}_{\text{h}}^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{T}_{\text{h}}^{(\alpha_1^{(3)})}\ $
41	1.568E-02	2.593E-02
81	7.802E-03	1.233E-02
161	3.896E-03	6.087E-03
321	1.947E-03	3.033E-03
641	9.736E-04	1.515E-03
1281	4.868E-04	7.575E-04
2561	2.434E-04	3.787E-04
5121	1.217E-04	1.894E-04
10241	6.085E-05	9.468E-05

Tableau H.5 Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P2$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $
33	1.344E-03	5.634E-04
65	3.377E-04	1.339E-04
129	8.453E-05	3.304E-05
257	2.114E-05	8.234E-06
513	5.285E-06	2.057E-06
1025	1.321E-06	5.141E-07
2049	3.303E-07	1.285E-07
4097	8.258E-08	3.213E-08
8193	2.064E-08	8.033E-09
16385	5.161E-09	2.008E-09

Tableau H.6 Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P2$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $
33	2.470E-03	1.394E-03
65	6.203E-04	3.277E-04
129	1.552E-04	8.066E-05
257	3.882E-05	2.008E-05
513	9.706E-06	5.016E-06
1025	2.462E-06	1.254E-06
2049	6.066E-07	3.134E-07
4097	1.517E-07	7.835E-08
8193	3.791E-08	1.959E-08
16385	9.479E-09	4.897E-09

Tableau H.7 Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P2$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $
33	2.941E-03	2.498E-03
65	7.377E-04	5.782E-04
129	1.846E-04	1.417E-04
257	4.515E-05	3.525E-05
513	1.154E-05	8.802E-06
1025	2.885E-06	2.200E-06
2049	7.211E-07	5.499E-07
4097	1.803E-07	1.375E-07
8193	4.507E-08	3.437E-08
16385	1.127E-08	8.592E-09

Tableau H.8 Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P2$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $
33	2.241E-03	2.750E-03
65	5.502E-04	6.135E-04
129	1.369E-04	1.490E-04
257	3.420E-05	3.699E-05
513	8.548E-06	9.231E-06
1025	2.137E-06	2.307E-06
2049	5.342E-07	5.766E-07
4097	1.335E-07	1.441E-07
8193	3.339E-08	3.604E-08
16385	8.347E-09	9.009E-09

Tableau H.9 Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P3$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $
61	3.439E-06	1.161E-05
121	4.302E-07	1.446E-06
241	5.379E-08	1.806E-07
481	6.724E-09	2.257E-08
961	8.406E-10	2.821E-09
1921	1.051E-10	3.527E-10
3841	1.341E-11	4.428E-11

Tableau H.10 Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P3$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $
61	1.070E-05	2.916E-05
121	1.339E-06	3.617E-06
241	1.674E-07	4.513E-07
481	2.092E-08	5.639E-08
961	2.615E-09	7.048E-09
1921	3.269E-10	8.810E-10
3841	4.094E-11	1.101E-10

Tableau H.11 Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P3$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $
61	2.624E-05	5.430E-05
121	3.278E-06	6.690E-06
241	4.097E-07	8.335E-07
481	5.122E-08	1.041E-07
961	6.402E-09	1.301E-08
1921	8.002E-10	1.626E-09
3841	1.000E-10	2.033E-10

Tableau H.12 Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P3$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $
61	4.659E-05	6.648E-05
121	5.804E-06	8.065E-06
241	7.250E-07	1.001E-06
481	9.061E-08	1.249E-07
961	1.133E-08	1.560E-08
1921	1.416E-09	1.950E-09
3841	1.770E-10	2.438E-10

Tableau H.13 Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P4$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $
41	1.779E-06	4.836E-07
81	1.165E-07	2.893E-08
161	7.364E-09	1.792E-09
321	4.616E-10	1.118E-10
641	2.887E-11	6.985E-12
1281	1.843E-12	5.870E-13

Tableau H.14 Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P4$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $
41	5.596E-06	1.841E-06
81	3.668E-07	1.093E-07
161	2.319E-08	6.758E-09
321	1.454E-09	4.213E-10
641	9.093E-11	2.632E-11
1281	5.691E-12	1.668E-12

Tableau H.15 Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P4$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $
41	1.397E-05	5.816E-06
81	9.175E-07	3.418E-07
161	5.803E-08	2.108E-08
321	3.638E-09	1.313E-09
641	2.275E-10	8.199E-11
1281	1.423E-11	5.155E-12

Tableau H.16 Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P4$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $
41	2.574E-05	1.457E-05
81	1.698E-06	8.434E-07
161	1.074E-07	5.180E-08
321	6.734E-09	3.223E-09
641	4.212E-10	2.013E-10
1281	2.634E-11	1.261E-11

Tableau H.17 Erreur exacte pour $u^{(\alpha_1^{(0)})}$ et $T^{(\alpha_1^{(0)})}$ avec des éléments $P5$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(0)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(0)})}\ $
11	1.134E-05	3.853E-05
21	3.719E-07	1.340E-06
41	5.673E-09	2.811E-08
81	8.261E-11	4.613E-10
161	1.284E-12	7.269E-12

Tableau H.18 Erreur exacte pour $u^{(\alpha_1^{(1)})}$ et $T^{(\alpha_1^{(1)})}$ avec des éléments $P5$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(1)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(1)})}\ $
11	5.473E-05	1.757E-04
21	1.875E-06	6.126E-06
41	2.876E-08	1.265E-07
81	4.181E-10	2.057E-09
161	6.479E-12	3.228E-11

Tableau H.19 Erreur exacte pour $u^{(\alpha_1^{(2)})}$ et $T^{(\alpha_1^{(2)})}$ avec des éléments $P5$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(2)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(2)})}\ $
11	2.255E-04	6.715E-04
21	8.219E-06	2.424E-05
41	1.273E-07	4.892E-07
81	1.847E-09	7.861E-09
161	2.861E-11	1.227E-10

Tableau H.20 Erreur exacte pour $u^{(\alpha_1^{(3)})}$ et $T^{(\alpha_1^{(3)})}$ avec des éléments $P5$.

nnodes	$\ \mathbf{u}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{u}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{exa}}^{(\alpha_1^{(3)})} - \mathbf{T}_{\mathbf{h}}^{(\alpha_1^{(3)})}\ $
11	7.644E-04	2.039E-03
21	3.056E-05	8.083E-05
41	4.816E-07	1.581E-06
81	6.966E-09	2.504E-08
161	1.079E-10	3.878E-10

Erreurs entre la méthode des DF et notre méthode

Tableau H.21 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 1$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{u}_h^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{T}_h^{(\alpha_1^{(1)})}\ $
0.0145	6.133E-05	7.160E-04
0.0155	7.008E-05	8.182E-04
0.0165	7.941E-05	9.271E-04
0.0175	8.933E-05	1.043E-03
0.0185	9.983E-05	1.165E-03
0.0195	1.109E-04	1.295E-03
0.0205	1.226E-04	1.431E-03
0.0215	1.348E-04	1.574E-03
0.0225	1.477E-04	1.724E-03
0.0235	1.611E-04	1.880E-03
0.0245	1.751E-04	2.044E-03

Tableau H.22 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 2$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{u}_h^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{T}_h^{(\alpha_1^{(2)})}\ $
0.0145	5.171E-05	1.260E-03
0.0155	5.909E-05	1.439E-03
0.0165	6.696E-05	1.631E-03
0.0175	7.532E-05	1.835E-03
0.0185	8.417E-05	2.050E-03
0.0195	9.352E-05	2.277E-03
0.0205	1.034E-04	2.517E-03
0.0215	1.137E-04	2.768E-03
0.0225	1.245E-04	3.031E-03
0.0235	1.358E-04	3.307E-03
0.0245	1.476E-04	3.594E-03

Tableau H.23 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 3$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{u}_h^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{T}_h^{(\alpha_1^{(3)})}\ $
0.0145	4.354E-04	1.379E-02
0.0155	4.976E-04	1.576E-02
0.0165	5.639E-04	1.786E-02
0.0175	6.344E-04	2.010E-02
0.0185	7.090E-04	2.246E-02
0.0195	7.877E-04	2.495E-02
0.0205	8.707E-04	2.758E-02
0.0215	9.577E-04	3.034E-02
0.0225	1.049E-03	3.323E-02
0.0235	1.144E-03	3.625E-02
0.0245	1.244E-03	3.941E-02

Tableau H.24 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 4$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{u}_h^{(\alpha_1^{(4)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{T}_h^{(\alpha_1^{(4)})}\ $
0.0145	1.632E-03	1.594E-01
0.0155	1.865E-03	1.821E-01
0.0165	2.113E-03	2.064E-01
0.0175	2.377E-03	2.322E-01
0.0185	2.656E-03	2.594E-01
0.0195	2.950E-03	2.882E-01
0.0205	3.260E-03	3.185E-01
0.0215	3.585E-03	3.503E-01
0.0225	3.926E-03	3.836E-01
0.0235	4.282E-03	4.184E-01
0.0245	4.654E-03	4.548E-01

Tableau H.25 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 5$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{u}_h^{(\alpha_1^{(5)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{T}_h^{(\alpha_1^{(5)})}\ $
0.0145	1.317E-02	2.481E-00
0.0155	1.503E-02	2.834E-00
0.0165	1.701E-02	3.209E-00
0.0175	1.911E-02	3.607E-00
0.0185	2.133E-02	4.028E-00
0.0195	2.367E-02	4.472E-00
0.0205	2.612E-02	4.939E-00
0.0215	2.869E-02	5.428E-00
0.0225	3.137E-02	5.939E-00
0.0235	3.417E-02	6.472E-00
0.0245	3.707E-02	7.028E-00

Tableau H.26 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P1$ ($n = 6$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{u}_h^{(\alpha_1^{(6)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{T}_h^{(\alpha_1^{(6)})}\ $
0.0145	1.059E-01	10.867E-00
0.0155	1.217E-01	12.415E-00
0.0165	1.374E-01	14.074E-00
0.0175	1.550E-01	15.833E-00
0.0185	1.733E-01	17.697E-00
0.0195	1.926E-01	19.667E-00
0.0205	2.129E-01	21.740E-00
0.0215	2.342E-01	23.918E-00
0.0225	2.567E-01	26.201E-00
0.0235	2.802E-01	28.588E-00
0.0245	3.047E-01	31.081E-00

Tableau H.27 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 1$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{u}_h^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{T}_h^{(\alpha_1^{(1)})}\ $
0.0145	8.092E-05	1.011E-03
0.0155	9.247E-05	1.155E-03
0.0165	1.048E-04	1.309E-03
0.0175	1.179E-04	1.472E-03
0.0185	1.317E-04	1.645E-03
0.0195	1.463E-04	1.828E-03
0.0205	1.617E-04	2.020E-03
0.0215	1.779E-04	2.222E-03
0.0225	1.948E-04	2.433E-03
0.0235	2.125E-04	2.654E-03
0.0245	2.310E-04	2.885E-03

Tableau H.28 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 2$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{u}_h^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{T}_h^{(\alpha_1^{(2)})}\ $
0.0145	7.153E-05	1.695E-03
0.0155	8.174E-05	1.937E-03
0.0165	9.263E-05	2.194E-03
0.0175	1.042E-04	2.468E-03
0.0185	1.164E-04	2.758E-03
0.0195	1.294E-04	3.064E-03
0.0205	1.430E-04	3.386E-03
0.0215	1.573E-04	3.724E-03
0.0225	1.722E-04	4.079E-03
0.0235	1.879E-04	4.449E-03
0.0245	2.042E-04	4.835E-03

Tableau H.29 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 3$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{u}_h^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{T}_h^{(\alpha_1^{(3)})}\ $
0.0145	6.050E-04	1.991E-02
0.0155	6.913E-04	2.275E-02
0.0165	7.834E-04	2.579E-02
0.0175	8.813E-04	2.901E-02
0.0185	9.850E-04	3.242E-02
0.0195	1.094E-03	3.603E-02
0.0205	1.210E-03	3.982E-02
0.0215	1.331E-03	4.380E-02
0.0225	1.457E-03	4.798E-02
0.0235	1.590E-03	5.234E-02
0.0245	1.728E-03	5.690E-02

Tableau H.30 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 4$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{u}_h^{(\alpha_1^{(4)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{T}_h^{(\alpha_1^{(4)})}\ $
0.0145	2.265E-03	2.221E-01
0.0155	2.588E-03	2.538E-01
0.0165	2.932E-03	2.875E-01
0.0175	3.298E-03	3.234E-01
0.0185	3.685E-03	3.614E-01
0.0195	4.094E-03	4.015E-01
0.0205	4.524E-03	4.437E-01
0.0215	4.975E-03	4.880E-01
0.0225	5.448E-03	5.344E-01
0.0235	5.942E-03	5.829E-01
0.0245	6.457E-03	6.335E-01

Tableau H.31 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 5$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{u}_h^{(\alpha_1^{(5)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{T}_h^{(\alpha_1^{(5)})}\ $
0.0145	1.616E-02	3.220E-00
0.0155	1.847E-02	3.677E-00
0.0165	2.091E-02	4.163E-00
0.0175	2.348E-02	4.680E-00
0.0185	2.621E-02	5.226E-00
0.0195	2.907E-02	5.802E-00
0.0205	3.208E-02	6.406E-00
0.0215	3.523E-02	7.040E-00
0.0225	3.852E-02	7.703E-00
0.0235	4.194E-02	8.395E-00
0.0245	4.550E-02	9.115E-00

Tableau H.32 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P2$ ($n = 6$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{u}_h^{(\alpha_1^{(6)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{T}_h^{(\alpha_1^{(6)})}\ $
0.0145	1.621E-01	15.781E-00
0.0155	1.844E-01	18.043E-00
0.0165	2.088E-01	20.451E-00
0.0175	2.352E-01	23.010E-00
0.0185	2.627E-01	25.722E-00
0.0195	2.923E-01	28.585E-00
0.0205	3.236E-01	31.599E-00
0.0215	3.577E-01	34.770E-00
0.0225	3.901E-01	38.090E-00
0.0235	4.256E-01	41.565E-00
0.0245	4.631E-01	45.193E-00

Tableau H.33 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 1$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{u}_h^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{T}_h^{(\alpha_1^{(1)})}\ $
0.0145	9.910E-05	1.238E-03
0.0155	1.132E-04	1.414E-03
0.0165	1.283E-04	1.603E-03
0.0175	1.444E-04	1.803E-03
0.0185	1.613E-04	2.015E-03
0.0195	1.792E-04	2.238E-03
0.0205	1.981E-04	2.474E-03
0.0215	2.179E-04	2.721E-03
0.0225	2.386E-04	2.980E-03
0.0235	2.603E-04	3.251E-03
0.0245	2.829E-04	3.533E-03

Tableau H.34 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 2$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{u}_h^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{T}_h^{(\alpha_1^{(2)})}\ $
0.0145	8.708E-05	2.076E-03
0.0155	9.951E-05	2.372E-03
0.0165	1.128E-04	2.687E-03
0.0175	1.268E-04	3.023E-03
0.0185	1.418E-04	3.378E-03
0.0195	1.575E-04	3.753E-03
0.0205	1.741E-04	4.147E-03
0.0215	1.915E-04	4.561E-03
0.0225	2.097E-04	4.995E-03
0.0235	2.288E-04	5.448E-03
0.0245	2.486E-04	5.921E-03

Tableau H.35 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 3$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{u}_h^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{T}_h^{(\alpha_1^{(3)})}\ $
0.0145	7.411E-04	2.439E-02
0.0155	8.469E-04	2.787E-02
0.0165	9.598E-04	3.159E-02
0.0175	1.080E-03	3.553E-02
0.0185	1.207E-03	3.971E-02
0.0195	1.341E-03	4.413E-02
0.0205	1.482E-03	4.877E-02
0.0215	1.630E-03	5.365E-02
0.0225	1.785E-03	5.877E-02
0.0235	1.948E-03	6.411E-02
0.0245	2.117E-03	6.967E-02

Tableau H.36 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 4$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{u}_h^{(\alpha_1^{(4)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{T}_h^{(\alpha_1^{(4)})}\ $
0.0145	2.774E-03	2.720E-01
0.0155	3.170E-03	3.108E-01
0.0165	3.591E-03	3.522E-01
0.0175	4.039E-03	3.961E-01
0.0185	4.514E-03	4.427E-01
0.0195	5.014E-03	4.918E-01
0.0205	5.541E-03	5.434E-01
0.0215	6.093E-03	5.977E-01
0.0225	6.673E-03	6.545E-01
0.0235	7.278E-03	7.139E-01
0.0245	7.909E-03	7.759E-01

Tableau H.37 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 5$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{u}_h^{(\alpha_1^{(5)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{T}_h^{(\alpha_1^{(5)})}\ $
0.0145	1.982E-02	3.942E-00
0.0155	2.263E-02	4.502E-00
0.0165	2.559E-02	5.098E-00
0.0175	2.875E-02	5.731E-00
0.0185	3.209E-02	6.400E-00
0.0195	3.560E-02	7.104E-00
0.0205	3.928E-02	7.845E-00
0.0215	4.314E-02	8.621E-00
0.0225	4.716E-02	9.433E-00
0.0235	5.136E-02	10.280E-00
0.0245	5.572E-02	11.162E-00

Tableau H.38 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P3$ ($n = 6$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{u}_h^{(\alpha_1^{(6)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{T}_h^{(\alpha_1^{(6)})}\ $
0.0145	1.993E-01	19.328E-00
0.0155	2.266E-01	22.098E-00
0.0165	2.570E-01	25.043E-00
0.0175	2.882E-01	28.184E-00
0.0185	3.228E-01	31.502E-00
0.0195	3.586E-01	35.009E-00
0.0205	3.949E-01	38.713E-00
0.0215	4.346E-01	42.594E-00
0.0225	4.780E-01	46.654E-00
0.0235	5.215E-01	50.911E-00
0.0245	5.668E-01	55.358E-00

Tableau H.39 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 1$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{u}_h^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{T}_h^{(\alpha_1^{(1)})}\ $
0.0145	1.144E-04	1.429E-03
0.0155	1.307E-04	1.633E-03
0.0165	1.482E-04	1.851E-03
0.0175	1.667E-04	2.082E-03
0.0185	1.863E-04	2.326E-03
0.0195	2.069E-04	2.584E-03
0.0205	2.287E-04	2.856E-03
0.0215	2.516E-04	3.142E-03
0.0225	2.755E-04	3.441E-03
0.0235	3.005E-04	3.753E-03
0.0245	3.267E-04	4.079E-03

Tableau H.40 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 2$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{u}_h^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{T}_h^{(\alpha_1^{(2)})}\ $
0.0145	1.003E-04	2.397E-03
0.0155	1.146E-04	2.739E-03
0.0165	1.299E-04	3.103E-03
0.0175	1.461E-04	3.491E-03
0.0185	1.636E-04	3.901E-03
0.0195	1.814E-04	4.333E-03
0.0205	2.005E-04	4.789E-03
0.0215	2.206E-04	5.267E-03
0.0225	2.416E-04	5.768E-03
0.0235	2.635E-04	6.291E-03
0.0245	2.864E-04	6.838E-03

Tableau H.41 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 3$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{u}_h^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{T}_h^{(\alpha_1^{(3)})}\ $
0.0145	8.559E-04	2.816E-02
0.0155	9.781E-04	3.218E-02
0.0165	1.108E-03	3.647E-02
0.0175	1.247E-03	4.103E-02
0.0185	1.394E-03	4.586E-02
0.0195	1.548E-03	5.095E-02
0.0205	1.711E-03	5.632E-02
0.0215	1.883E-03	6.195E-02
0.0225	2.062E-03	6.786E-02
0.0235	2.249E-03	7.403E-02
0.0245	2.445E-03	8.048E-02

Tableau H.42 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 4$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{u}_h^{(\alpha_1^{(4)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{T}_h^{(\alpha_1^{(4)})}\ $
0.0145	3.203E-03	3.141E-01
0.0155	3.659E-03	3.589E-01
0.0165	4.146E-03	4.067E-01
0.0175	4.664E-03	4.574E-01
0.0185	5.212E-03	5.111E-01
0.0195	5.790E-03	5.678E-01
0.0205	6.398E-03	6.275E-01
0.0215	7.036E-03	6.902E-01
0.0225	7.705E-03	7.558E-01
0.0235	8.404E-03	8.244E-01
0.0245	9.133E-03	8.959E-01

Tableau H.43 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 5$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{u}_h^{(\alpha_1^{(5)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{T}_h^{(\alpha_1^{(5)})}\ $
0.0145	2.285E-02	4.553E-00
0.0155	2.613E-02	5.199E-00
0.0165	2.953E-02	5.887E-00
0.0175	3.320E-02	6.617E-00
0.0185	3.705E-02	7.390E-00
0.0195	4.112E-02	8.203E-00
0.0205	4.537E-02	9.059E-00
0.0215	4.981E-02	9.955E-00
0.0225	5.447E-02	10.892E-00
0.0235	5.931E-02	11.870E-00
0.0245	6.434E-02	12.889E-00

Tableau H.44 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P4$ ($n = 6$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{u}_h^{(\alpha_1^{(6)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{T}_h^{(\alpha_1^{(6)})}\ $
0.0145	2.373E-01	22.286E-00
0.0155	2.548E-01	25.554E-00
0.0165	2.851E-01	28.985E-00
0.0175	3.260E-01	32.583E-00
0.0185	3.713E-01	36.381E-00
0.0195	4.107E-01	40.444E-00
0.0205	4.587E-01	44.687E-00
0.0215	5.028E-01	49.179E-00
0.0225	5.513E-01	53.875E-00
0.0235	6.025E-01	58.786E-00
0.0245	6.541E-01	63.924E-00

Tableau H.45 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 1$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{u}_h^{(\alpha_1^{(1)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(1)})} - \mathbf{T}_h^{(\alpha_1^{(1)})}\ $
0.0145	1.279E-04	1.598E-03
0.0155	1.462E-04	1.826E-03
0.0165	1.656E-04	2.069E-03
0.0175	1.863E-04	2.327E-03
0.0185	2.082E-04	2.601E-03
0.0195	2.314E-04	2.889E-03
0.0205	2.557E-04	3.193E-03
0.0215	2.812E-04	3.512E-03
0.0225	3.080E-04	3.847E-03
0.0235	3.360E-04	4.196E-03
0.0245	3.652E-04	4.561E-03

Tableau H.46 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 2$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{u}_h^{(\alpha_1^{(2)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(2)})} - \mathbf{T}_h^{(\alpha_1^{(2)})}\ $
0.0145	1.120E-04	2.679E-03
0.0155	1.280E-04	3.062E-03
0.0165	1.450E-04	3.469E-03
0.0175	1.632E-04	3.903E-03
0.0185	1.823E-04	4.361E-03
0.0195	2.026E-04	4.845E-03
0.0205	2.239E-04	5.354E-03
0.0215	2.463E-04	5.889E-03
0.0225	2.697E-04	6.449E-03
0.0235	2.942E-04	7.034E-03
0.0245	3.198E-04	7.645E-03

Tableau H.47 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 3$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{u}_h^{(\alpha_1^{(3)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(3)})} - \mathbf{T}_h^{(\alpha_1^{(3)})}\ $
0.0145	9.569E-04	3.149E-02
0.0155	1.094E-03	3.598E-02
0.0165	1.239E-03	4.078E-02
0.0175	1.394E-03	4.587E-02
0.0185	1.558E-03	5.127E-02
0.0195	1.731E-03	5.697E-02
0.0205	1.913E-03	6.297E-02
0.0215	2.105E-03	6.927E-02
0.0225	2.305E-03	7.587E-02
0.0235	2.515E-03	8.277E-02
0.0245	2.734E-03	8.997E-02

Tableau H.48 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 4$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{u}_h^{(\alpha_1^{(4)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(4)})} - \mathbf{T}_h^{(\alpha_1^{(4)})}\ $
0.0145	3.582E-03	3.512E-01
0.0155	4.093E-03	4.013E-01
0.0165	4.636E-03	4.547E-01
0.0175	5.215E-03	5.114E-01
0.0185	5.827E-03	5.715E-01
0.0195	6.474E-03	6.349E-01
0.0205	7.154E-03	7.016E-01
0.0215	7.867E-03	7.716E-01
0.0225	8.615E-03	8.450E-01
0.0235	9.396E-03	9.217E-01
0.0245	1.021E-02	1.002E-00

Tableau H.49 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 5$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{u}_h^{(\alpha_1^{(5)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(5)})} - \mathbf{T}_h^{(\alpha_1^{(5)})}\ $
0.0145	2.565E-02	5.089E-00
0.0155	2.927E-02	5.812E-00
0.0165	3.309E-02	6.582E-00
0.0175	3.708E-02	7.399E-00
0.0185	4.143E-02	8.262E-00
0.0195	4.596E-02	9.172E-00
0.0205	5.072E-02	10.128E-00
0.0215	5.570E-02	11.130E-00
0.0225	6.090E-02	12.178E-00
0.0235	6.631E-02	13.272E-00
0.0245	7.194E-02	14.410E-00

Tableau H.50 Erreurs entre les solutions éléments finis et celles calculées par DF avec des éléments $P5$ ($n = 6$).

$\Delta\alpha_1$	$\ \mathbf{u}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{u}_h^{(\alpha_1^{(6)})}\ $	$\ \mathbf{T}_{\text{DF}}^{(\alpha_1^{(6)})} - \mathbf{T}_h^{(\alpha_1^{(6)})}\ $
0.0145	2.780E-01	24.831E-00
0.0155	3.072E-01	28.431E-00
0.0165	3.324E-01	32.323E-00
0.0175	3.733E-01	36.381E-00
0.0185	4.151E-01	40.678E-00
0.0195	4.692E-01	45.165E-00
0.0205	5.144E-01	49.951E-00
0.0215	5.600E-01	54.995E-00
0.0225	6.162E-01	60.235E-00
0.0235	6.725E-01	65.730E-00
0.0245	7.308E-01	71.472E-00

ANNEXE I RÉSULTATS SUPPLÉMENTAIRES

Dans cette annexe, nous présentons certains résultats préliminaires pour les problèmes à 1 et 2 paramètre(s) décrits au chapitre 5. En premier lieu, nous analysons la convergence des résultats pour le problème avec 2 paramètres à l'aide de la MSM. Puis, nous montrons les erreurs qui existent entre la solution par DF et les sensibilités obtenues par le biais de notre méthode pour les éléments finis de degré 2, 3 et 5.

Analyse de convergence

Nous effectuons une analyse de convergence avec la Méthode des Solutions Manufacturées pour le problème unidimensionnel à 2 paramètres ($q = 2$). Nous avons défini les propriétés matérielles de ce problème aux équations (5.4), (5.5) et (5.6). La solution manufacturée pour le problème avec 2 paramètres est :

$$u_{exa}(x, \alpha_1, \alpha_2) = \sin(\alpha_1 \alpha_2 x), \quad (\text{I.1})$$

$$T_{exa}(x, \alpha_1, \alpha_2) = \cos(\alpha_1 \alpha_2 x). \quad (\text{I.2})$$

où α_1 et α_2 valent respectivement $\frac{\pi}{2}$ et $\frac{\pi}{4}$ et $x \in [0, 1]$.

Pour ce problème à 2 paramètres, nous résolvons les équations de sensibilités d'ordre 0 à 3. Au total nous avons $C_{3+1}^{3+2-1} = 10$ systèmes d'équations de sensibilités à résoudre, puisqu'il y a 1 seul système d'ordre 0, 2 systèmes d'ordre 1, 3 systèmes d'ordre 2 et 4 systèmes d'ordre 3 :

$$1 + 2 + 3 + 4 = 10.$$

Ces systèmes sont :

1. À l'ordre 0 : $\frac{\partial^0}{\partial \alpha_1^0 \partial \alpha_2^0}$;
2. À l'ordre 1 : $\frac{\partial^1}{\partial \alpha_1^1 \partial \alpha_2^0}$ et $\frac{\partial^1}{\partial \alpha_1^0 \partial \alpha_2^1}$;
3. À l'ordre 2 : $\frac{\partial^2}{\partial \alpha_1^2 \partial \alpha_2^0}$, $\frac{\partial^2}{\partial \alpha_1^1 \partial \alpha_2^1}$ et $\frac{\partial^2}{\partial \alpha_1^0 \partial \alpha_2^2}$;
4. À l'ordre 3 : $\frac{\partial^3}{\partial \alpha_1^3 \partial \alpha_2^0}$, $\frac{\partial^3}{\partial \alpha_1^2 \partial \alpha_2^1}$, $\frac{\partial^3}{\partial \alpha_1^1 \partial \alpha_2^2}$ et $\frac{\partial^3}{\partial \alpha_1^0 \partial \alpha_2^3}$.

Notre algorithme classe les solutions de ces systèmes dans un tableau tridimensionnel u grâce à un numéro r variant de 1 à 10 qui correspond à l'indice de la troisième dimension désignant

le numéro du système de sensibilités (se référer à la section 4.5.2).

Pour l'analyse de convergence, nous utilisons des éléments de degré $p = 1$ à 5 (polynômes de Lagrange). Ils sont notés $P1$ à $P5$. Afin de vérifier le taux de convergence, pour chaque éléments de type p , nous résolvons le problème pour des maillages de plus en plus fins. Nous illustrons les résultats obtenus pour les systèmes de sensibilités d'ordre 0 à 3 à 2 paramètres par les figures I.1, I.2, I.3 et I.4.

On observe que, peu importe le type des éléments finis, l'erreur exacte diminue avec le raffinement du maillage. L'erreur s'approche de plus en plus de 0 , lorsque la valeur de $\Delta h \rightarrow 0$:

$$\lim_{\Delta h \rightarrow 0} \|u_{\text{exa}}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} - u_h^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}\|_{1,\Omega} = 0, \quad (\text{I.3})$$

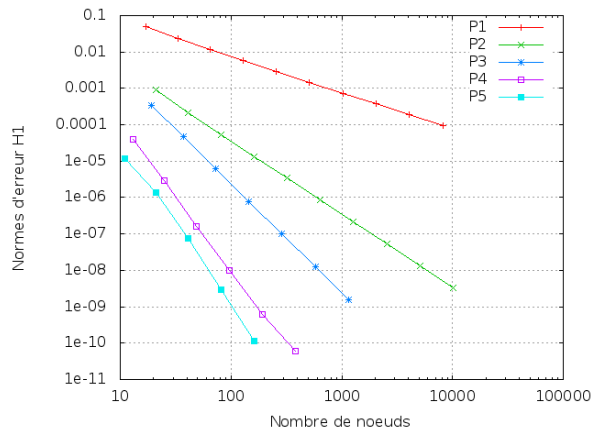
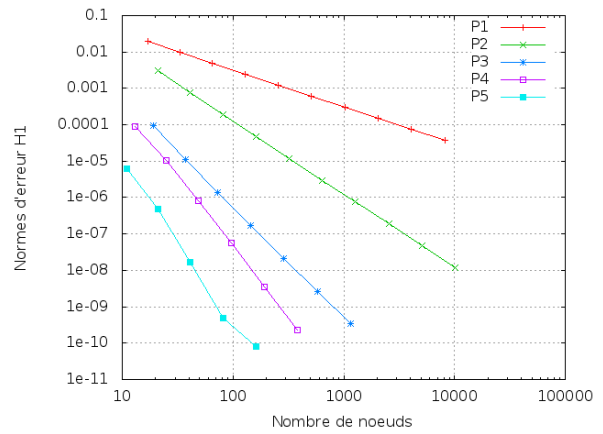
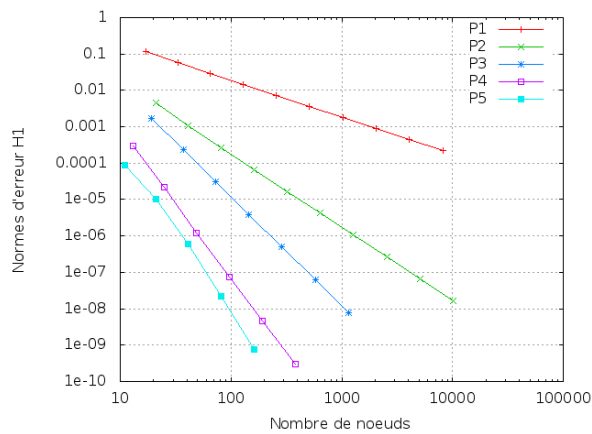
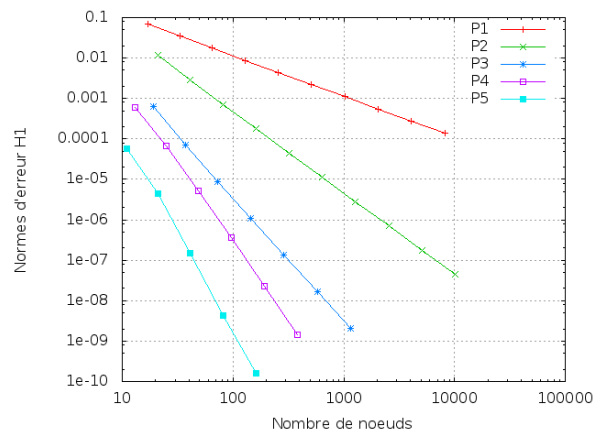
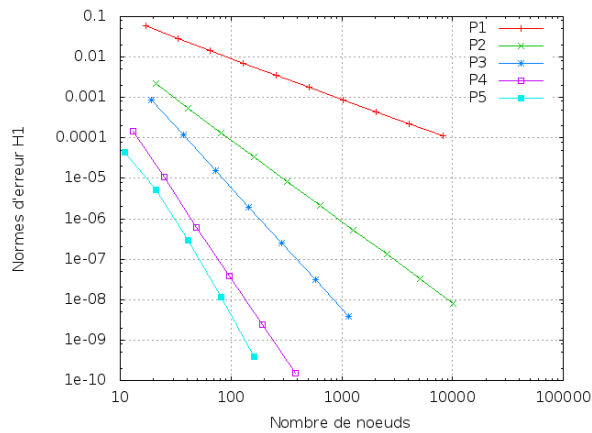
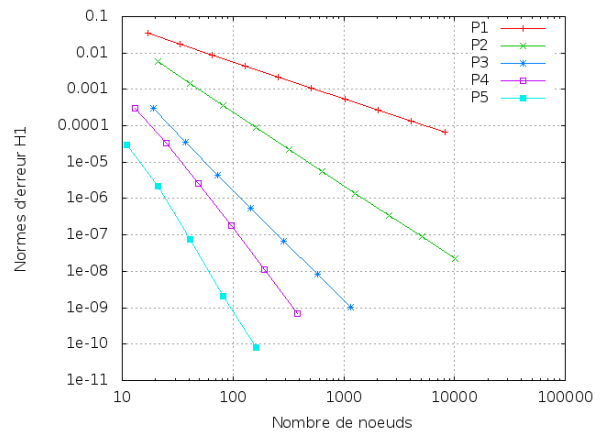
$$\lim_{\Delta h \rightarrow 0} \|T_{\text{exa}}^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})} - T_h^{(\alpha_1^{(n_1)} \alpha_2^{(n_2)} \dots \alpha_q^{(n_q)})}\|_{1,\Omega} = 0. \quad (\text{I.4})$$

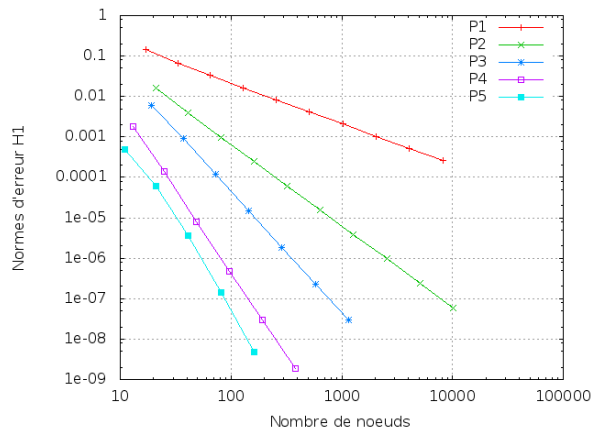
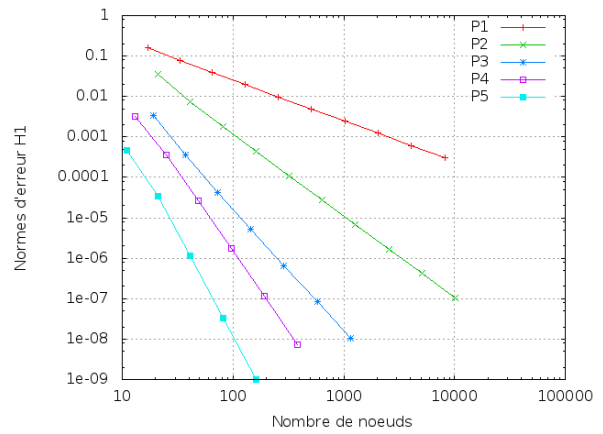
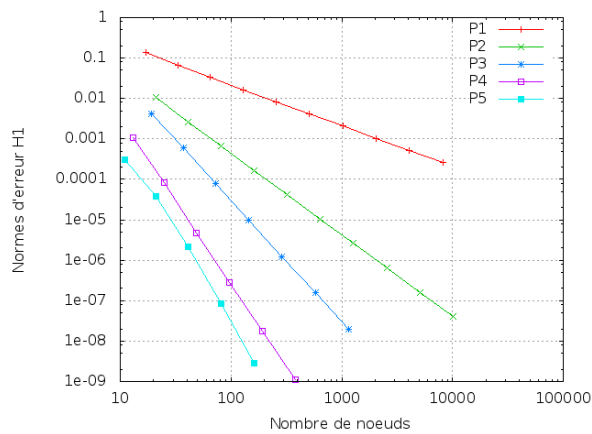
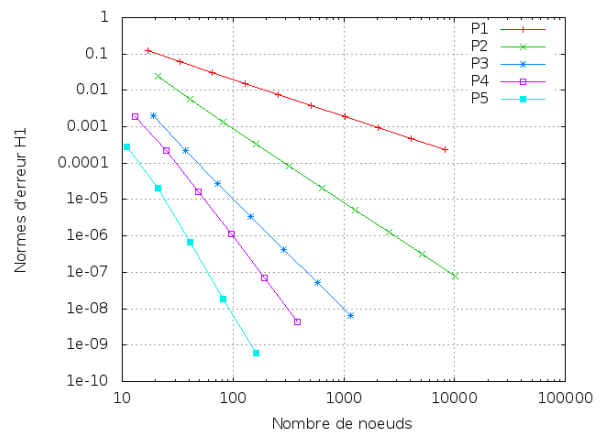
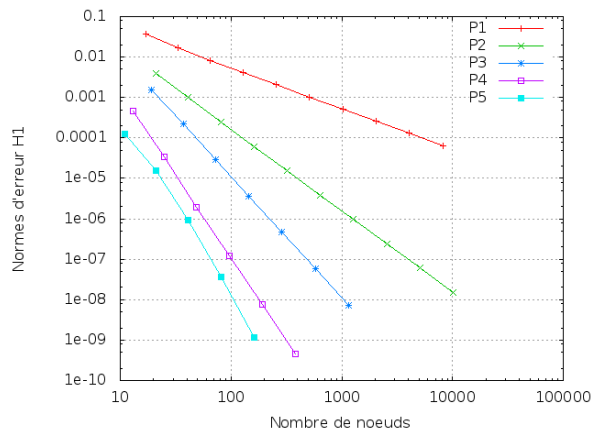
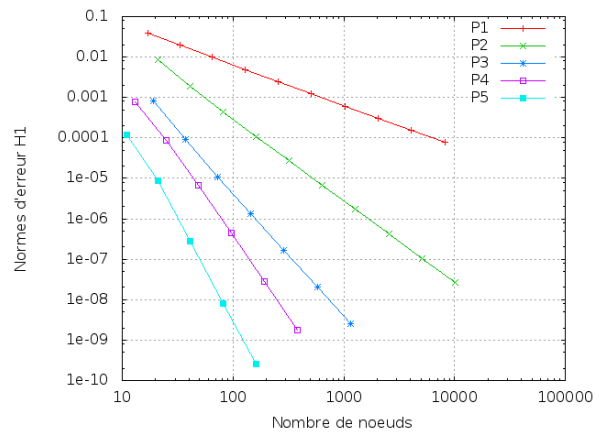
Par ailleurs, on note que l'erreur exacte, pour un même nombre de noeuds, diminue lorsque le degré p augmente. La précision de la solution numérique augmente donc avec le degré p des éléments finis. Par exemple pour 1000 noeuds, la solution éléments finis est plus précise pour des éléments de type $P3$ ($p = 3$) que pour des éléments de type $P2$ ($p = 2$). Nous remarquons également que le taux de convergence augmente avec le degré des éléments finis. En effet, la pente de l'erreur par rapport au nombre de noeuds est plus importante pour $p = 3$ (*i.e.* : $P3$) que pour $p = 2$ (*i.e.* : $P2$). Pour les éléments de types $P1$, $P2$, $P3$ et $P4$, nous trouvons respectivement des pentes s'approchant de -1 , -2 , -3 et -4 ce qui coïncide avec la théorie (se référer à l'équation (5.21)). Dans le cas des éléments de type $P5$, on note qu'il est plus difficile d'obtenir un taux de convergence égal à 5 (*i.e.* : pente de -5).

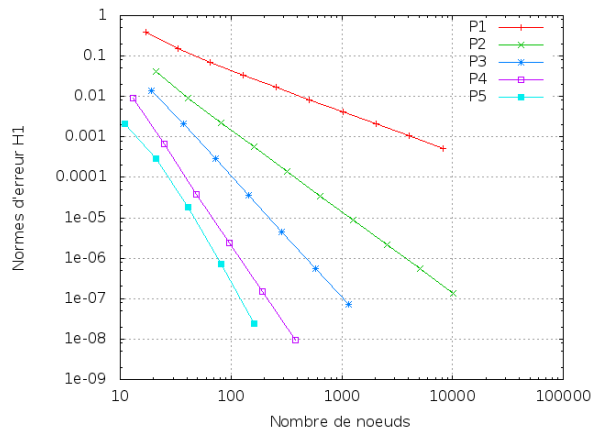
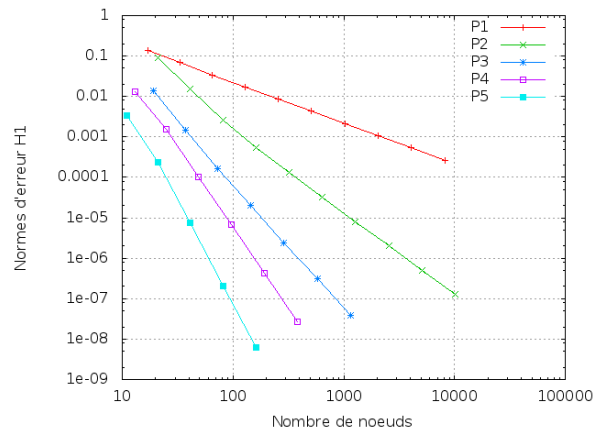
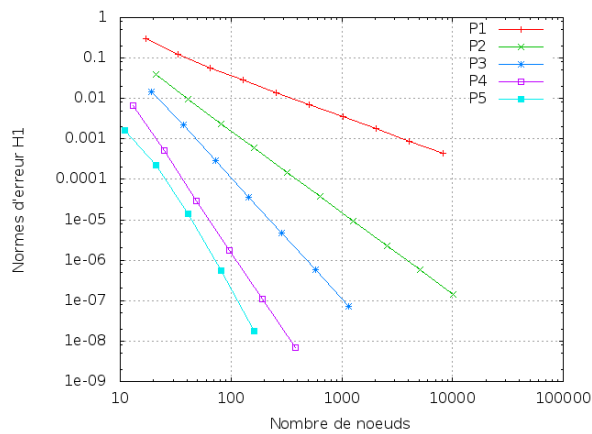
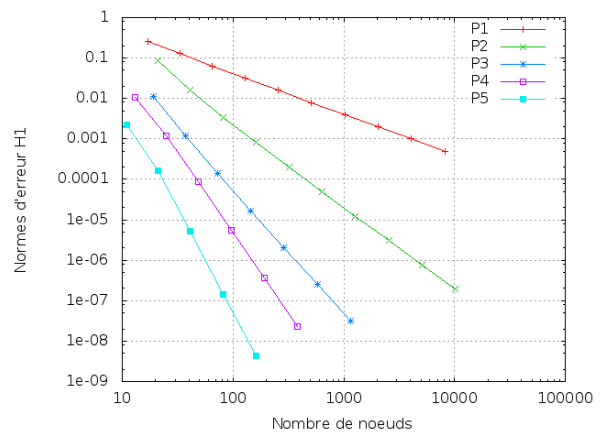
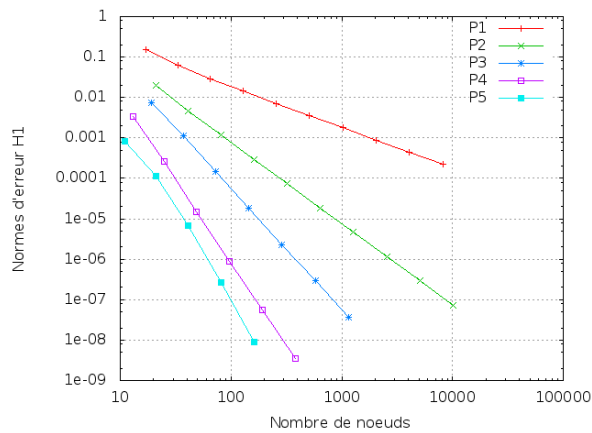
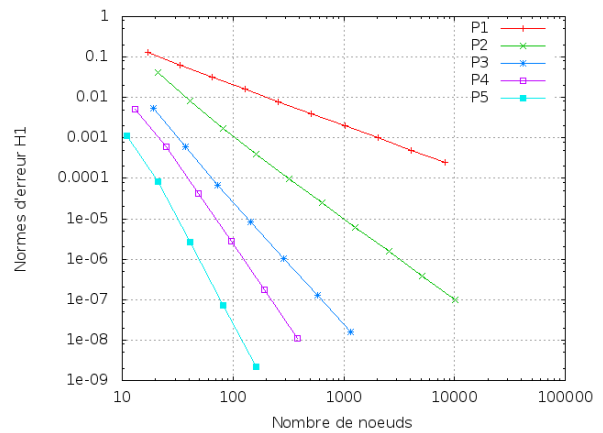
Comparaison avec la méthode des DF

Dans cette section, nous présentons les résultats obtenus pour les erreurs entre les sensibilités par DF et les sensibilités calculées par notre méthode pour des éléments finis d'ordre 2 , 3 et 5 . Nous illustrons ces résultats par les figures I.5, I.6 et I.7. Dans les graphiques qu'on y retrouve, chaque courbe représente l'erreur pour la sensibilité d'ordre n calculée par la méthode des DF en fonction du pas $\Delta\alpha_1$ choisi. L'ordre n varie de 1 à 6 .

Nous remarquons que les DF pour les sensibilités d'ordre plus élevé sont moins précises ;

(a) Analyse de convergence pour $T(\alpha_1^{(0)}\alpha_2^{(0)})$.(b) Analyse de convergence pour $u(\alpha_1^{(0)}\alpha_2^{(0)})$.(c) Analyse de convergence pour $T(\alpha_1^{(0)}\alpha_2^{(1)})$.(d) Analyse de convergence pour $u(\alpha_1^{(0)}\alpha_2^{(1)})$.(e) Analyse de convergence pour $T(\alpha_1^{(1)}\alpha_2^{(0)})$.(f) Analyse de convergence pour $u(\alpha_1^{(1)}\alpha_2^{(0)})$.Figure I.1 Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 1$ à 3.

(a) Analyse de convergence pour $T(\alpha_1^{(0)}\alpha_2^{(2)})$.(b) Analyse de convergence pour $u(\alpha_1^{(0)}\alpha_2^{(2)})$.(c) Analyse de convergence pour $T(\alpha_1^{(1)}\alpha_2^{(1)})$.(d) Analyse de convergence pour $u(\alpha_1^{(1)}\alpha_2^{(1)})$.(e) Analyse de convergence pour $T(\alpha_1^{(2)}\alpha_2^{(0)})$.(f) Analyse de convergence pour $u(\alpha_1^{(2)}\alpha_2^{(0)})$.Figure I.2 Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 4$ à 6.

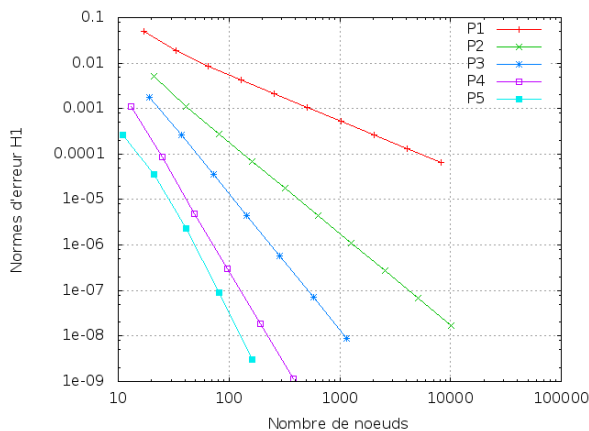
(a) Analyse de convergence pour $T(\alpha_1^{(0)}\alpha_2^{(3)})$.(b) Analyse de convergence pour $u(\alpha_1^{(0)}\alpha_2^{(3)})$.(c) Analyse de convergence pour $T(\alpha_1^{(1)}\alpha_2^{(2)})$.(d) Analyse de convergence pour $u(\alpha_1^{(1)}\alpha_2^{(2)})$.(e) Analyse de convergence pour $T(\alpha_1^{(2)}\alpha_2^{(1)})$.(f) Analyse de convergence pour $u(\alpha_1^{(2)}\alpha_2^{(1)})$.Figure I.3 Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 7$ à 9 .

les erreurs sont plus importantes. Par exemple, pour une dérivée d'ordre 4, l'erreur est plus grande que pour une dérivée d'ordre 2. En outre, nous observons que les sensibilités calculées par les DF convergent à l'ordre 2. Cela vient donc valider la théorie (voir l'équation (5.23)).

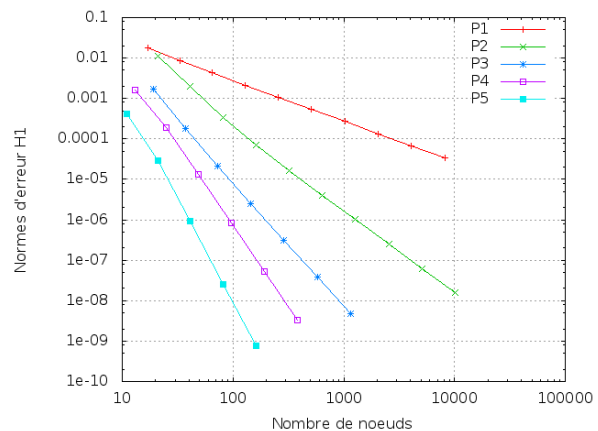
Évaluation rapide de solutions perturbées par série de Taylor

Dans cette partie, nous prenons le problème à 1 paramètre posé pour la MSM et nous extrapolons sa solution par la série de Taylor d'ordre 0 à 8 (voir sections 5.1 et 5.2). Connaissant la solution analytique de ce problème (équations (5.11) et (5.12)), nous sommes en mesure de vérifier avec certitude si l'approximation par la série de Taylor devient de plus en plus précise avec l'ordre pour ce problème spécifique.

Nous extrapolons la solution en $(\alpha_1)_0 = \frac{\pi}{2}$ à un point x donné. Nous perturbons la solution sur une plage de α_1 assez large : $\alpha_1 \in [\frac{\pi}{2} - 2, \frac{\pi}{2} + 1.5]$. Nous représentons les solutions approximées par l'intermédiaire de la série de Taylor dans la figure I.8. Globalement, la solution extrapolée semble s'approcher de plus en plus de la solution numérique recalculée lorsque l'ordre de la série de Taylor augmente. On constate que pour des valeurs de $\Delta\alpha_1$ dans le voisinage immédiat de $(\alpha_1)_0$, la solution extrapolée est très précise et, par conséquent, l'écart entre celle-ci et la solution recalculée est très faible. Nous exposons l'erreur entre la solution extrapolée et la solution numérique dans la figure I.9. Les courbes représentées par les figures I.9(a) et I.9(b) sont respectivement les erreurs pour u et pour T en fonction de la valeur du paramètre α_1 pour les séries de Taylor d'ordre 0 à 8.

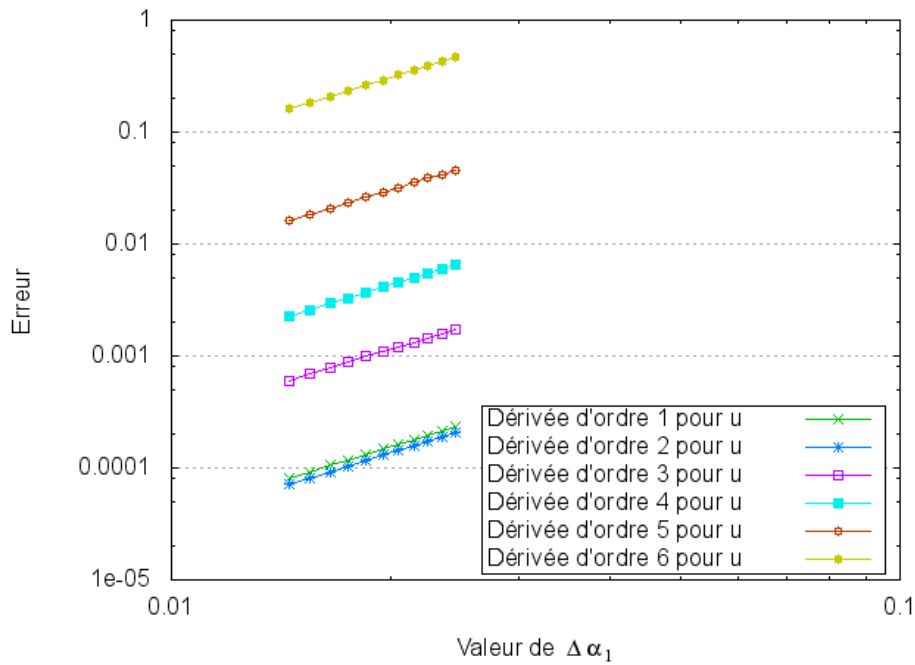
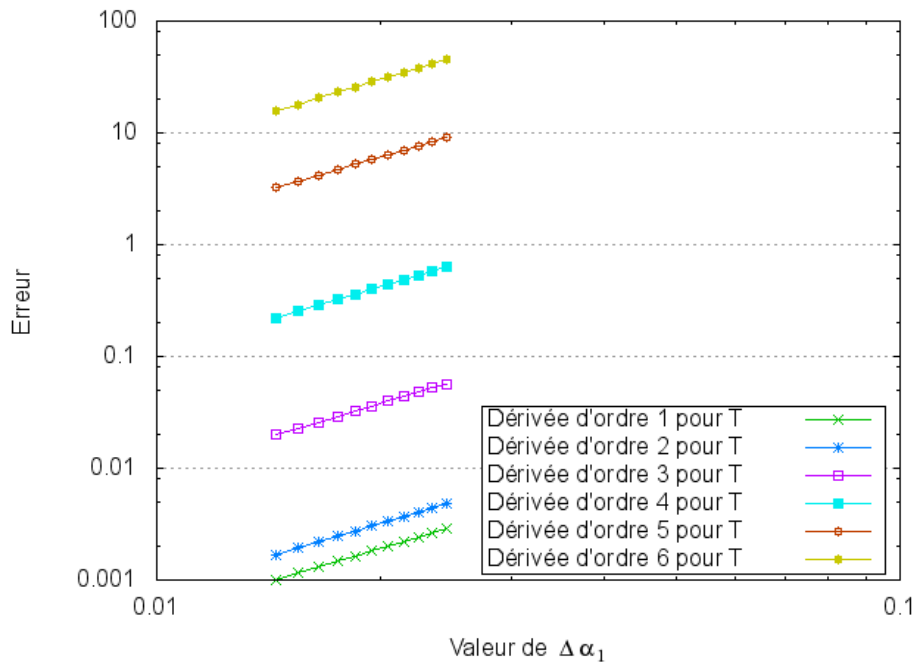


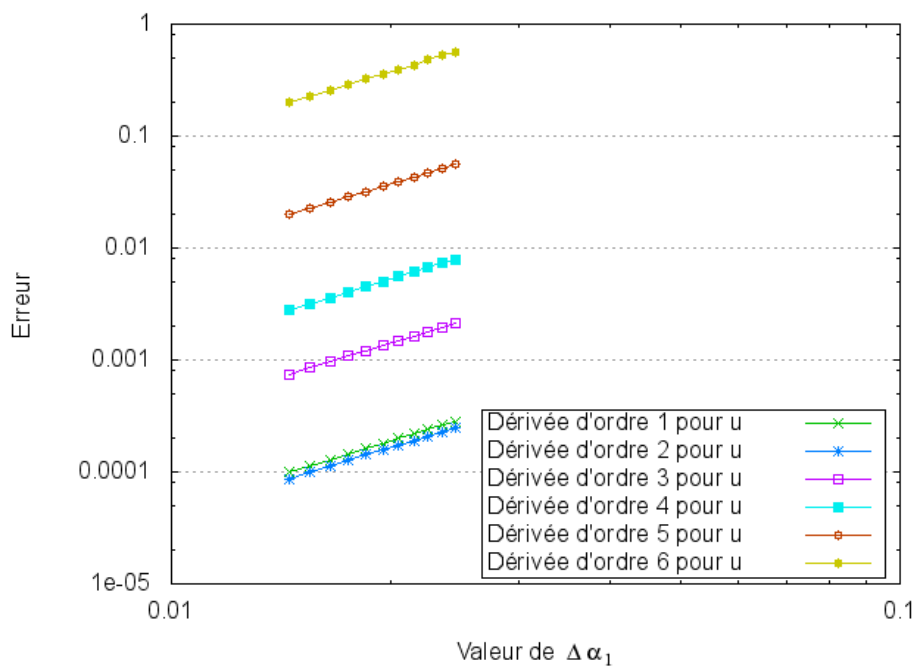
(a) Analyse de convergence pour $T(\alpha_1^{(3)} \alpha_2^{(0)})$.



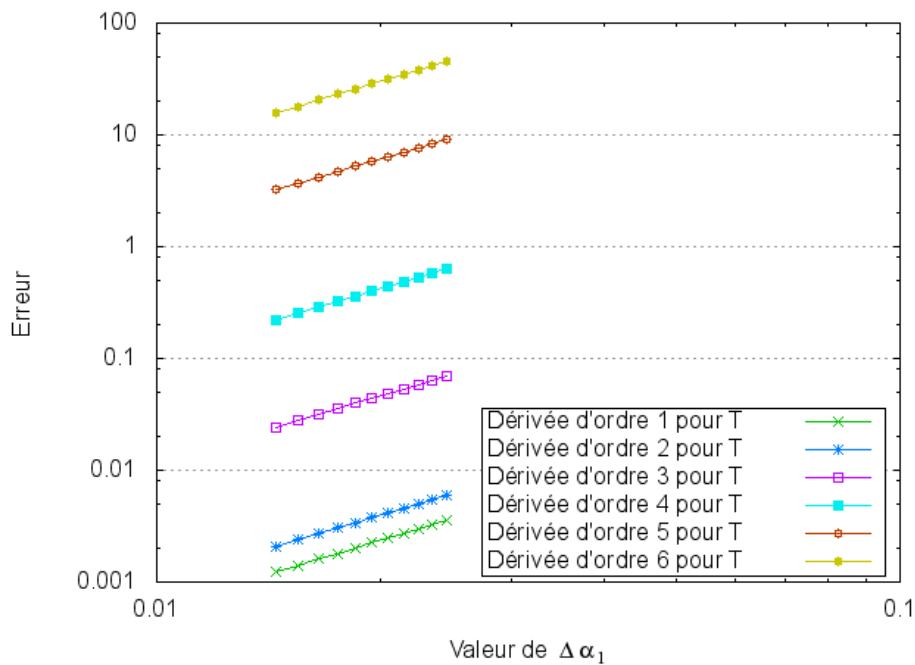
(b) Analyse de convergence pour $u(\alpha_1^{(3)} \alpha_2^{(0)})$.

Figure I.4 Erreur exacte en fonction du nombre de noeuds du maillage pour les sensibilités numéro $r = 10$.

(a) Erreur par rapport à $\Delta \alpha$ pour u en utilisant les éléments P_2 .(b) Erreur par rapport à $\Delta \alpha$ pour T en utilisant les éléments P_2 .Figure I.5 Comparaison de la méthode proposée avec la méthode des DF centrées (éléments de type P_2).

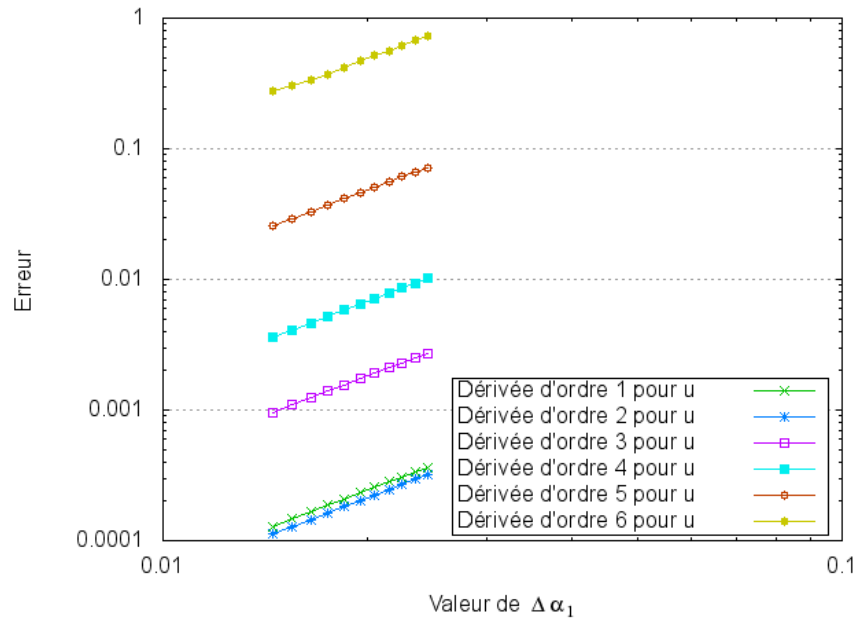
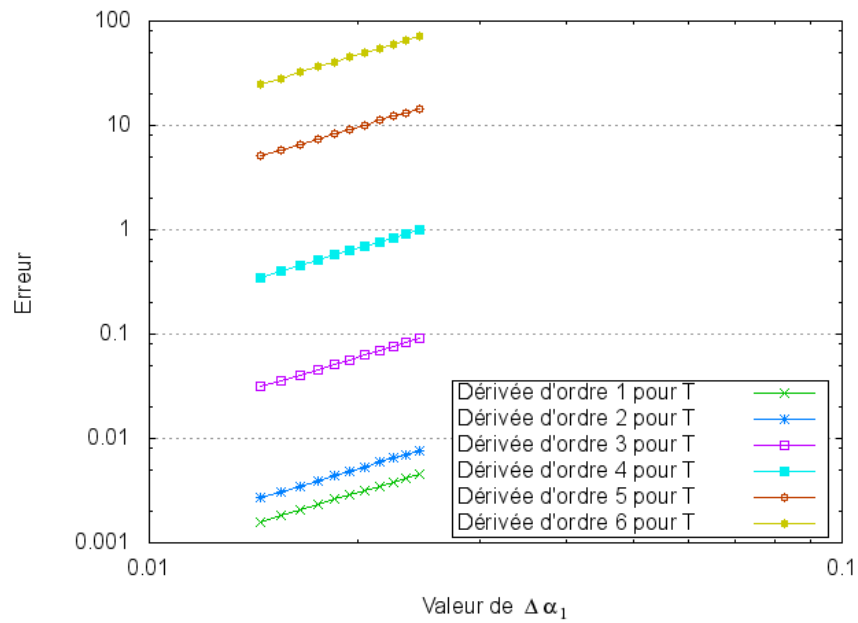


(a) Erreur par rapport à $\Delta\alpha$ pour u en utilisant les éléments P_3 .



(b) Erreur par rapport à $\Delta\alpha$ pour T en utilisant les éléments P_3 .

Figure I.6 Comparaison de la méthode proposée avec la méthode des DF centrées (éléments de type P_3).

(a) Erreur par rapport à $\Delta \alpha$ pour u en utilisant les éléments P_5 .(b) Erreur par rapport à $\Delta \alpha$ pour T en utilisant les éléments P_5 .Figure I.7 Comparaison de la méthode proposée avec la méthode des DF centrées (éléments de type P_5).

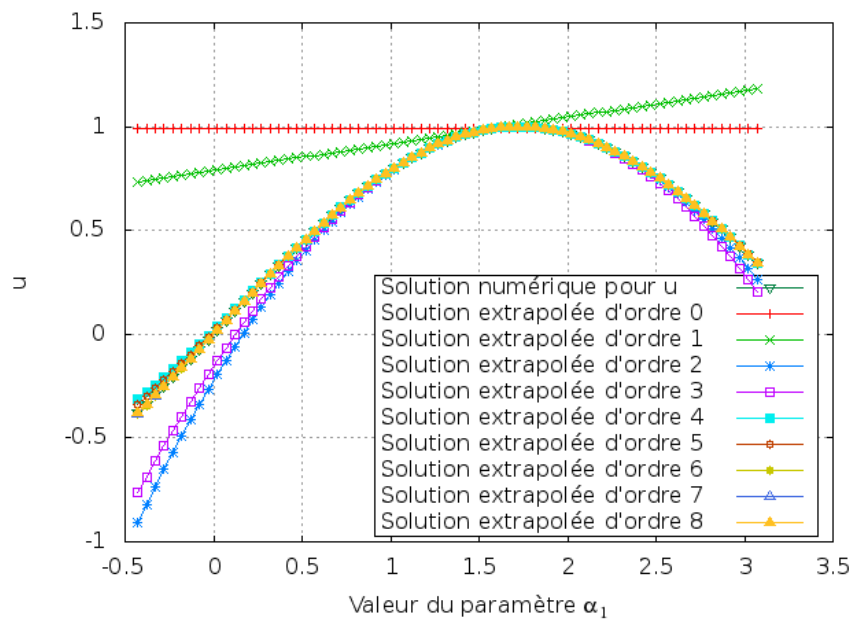
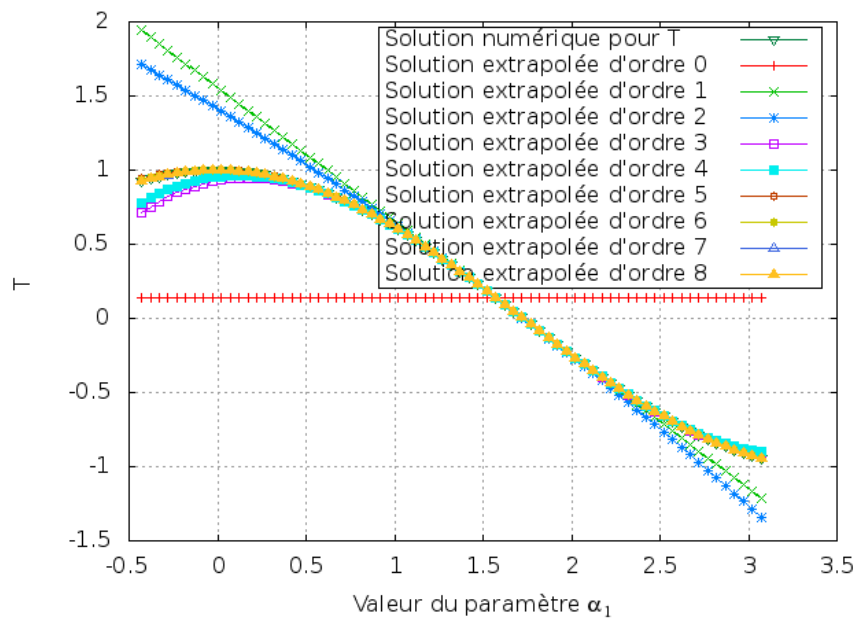
(a) Extrapolation de u en fonction de α_1 .(b) Extrapolation de T en fonction de α_1 .

Figure I.8 Comparaison des solutions numériques versus les solutions extrapolées par série de Taylor.

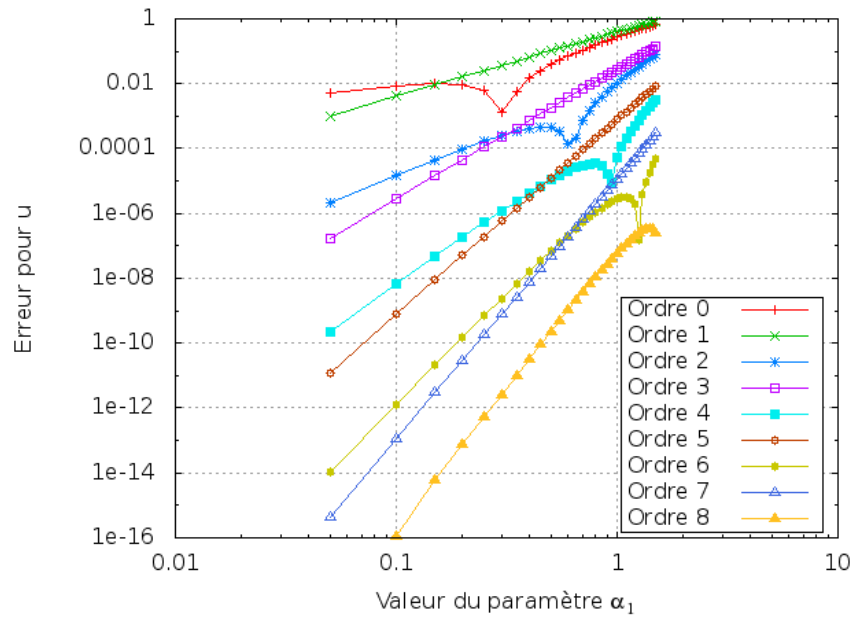
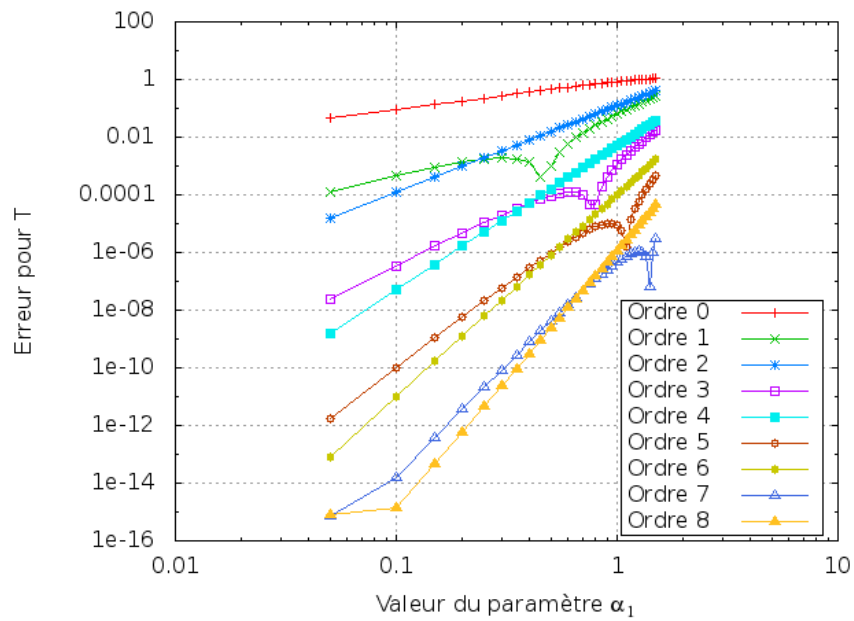
(a) Erreur de u en fonction de α_1 .(b) Erreur de T en fonction de α_1 .

Figure I.9 Comparaison des solutions numériques versus les solutions extrapolées par série de Taylor.