



HAL
open science

Simulation avancée des problèmes thermiques rencontrés lors de la mise en forme des composites

Chady Ghnatios

► To cite this version:

Chady Ghnatios. Simulation avancée des problèmes thermiques rencontrés lors de la mise en forme des composites. Matériaux et structures en mécanique [physics.class-ph]. Ecole Centrale de Nantes (ECN), 2012. Français. NNT: . tel-00867281

HAL Id: tel-00867281

<https://theses.hal.science/tel-00867281>

Submitted on 27 Sep 2013

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

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

École Centrale Nantes

École Doctorale

Science Pour l'Ingénieur, Géosciences, Architecture

Année 2011-2012

N°B.U. :

Thèse de Doctorat

SPÉCIALITÉ : GÉNIE MÉCANIQUE

Présentée et soutenue par:

CHADY GHNATIOS

Le 02 octobre 2012

à l'École Centrale Nantes

Titre

**Simulation avancée des problèmes thermiques
rencontrés lors de la mise en forme des composites**

JURY

Président	Pierre VILLON	Professeur de l'UTC, Université de Technologie de Compiègne
Rapporteurs	Fabrice SCHMIDT Alain NEVEU	Professeur des Écoles des Mines, École des Mines d'Albi-Carmaux Professeur d'Université, Université d'Évry Val D'essonne
Examineurs	Francisco CHINESTA Arnaud POITOU David NERON Pierre JOYOT	Professeur des universités, Ecole Centrale Nantes Professeur des universités, Ecole Centrale Nantes Maître de conférences, École Normale supérieure Cachan Maître de conférences, ESTIA Recherche, Biarritz
Invités	Sylvain CHATEL Adrien LEYGUE	Ingénieur R&D, EADS France Innovation Works Chargé de recherche CNRS, École Centrale Nantes
Directeur de thèse	Francisco CHINESTA	École Centrale Nantes
Co-directeur	Arnaud POITOU	Directeur de l'École Centrale Nantes
Encadrant	Adrien LEYGUE	École Centrale Nantes

N°E.D. : 498-229



Sommaire

Introduction	1
1 Les matériaux composites, mise en forme et simulation	5
1.1 Mise en forme des composites, nécessité d'une simulation "riche"	5
1.2 Méthodes de réduction de modèle	13
1.3 Positionnement du travail	23
2 La PGD, formulation tensorielle et autres formulations spécifiques	25
2.1 Formulation tensorielle de la PGD	25
2.2 Cas particuliers et formulations différentes	32
2.3 Conclusion	43
3 La PGD dans la simulation de la pultrusion	47
3.1 Introduction	47
3.2 Optimisation de la pultrusion	48
3.3 Mariage de la PGD avec la DDDAS	79
3.4 Conclusion	108
4 Modèle thermique paramétrique de la dépose de bandes thermoplastiques	109
4.1 Simulation des plaques stratifiées en 3D	110
4.2 Simulation des plaques stratifiées paramétrées	117
4.3 Simulation du drapage assisté en 3D	119
4.4 Simulation paramétrique de la dépose de bandes	127
4.5 Conclusion	132
5 La PGD dans l'analyse inverse Off-line/On-line	135
5.1 Présentation d'un algorithme original	135
5.2 Traitement du problème de Cauchy	136

5.3	Résolution par la PGD dans des géométries complexes	148
5.4	Conclusion	156
6	Optimisation et contrôle pour des modèles non-linéaires	159
6.1	Modèle de Bailleul non-linéaire paramétrique	160
6.2	Linéarisation et contrôle d'un modèle non-linéaire	170
6.3	Conclusion et perspectives	182
	Conclusion générale et perspectives	186
	Références	189
A	Exemple de construction des tenseurs élémentaires	201
A.1	Exemple d'enrichissement d'un système 2D	201
A.2	Problème multidimensionnel	203
B	Estimation d'erreur à posteriori	207
B.1	Problème modèle	207
B.2	Estimation sur un problème bidimensionnel	210

Table des figures

1.1	Exemple de la cinétique de réticulation selon le modèle de Bailleul	7
1.2	Exemple de contraintes générées au niveau global	8
1.3	Exemples de contraintes résiduelles générées au niveau des plis	9
1.4	Fissuration due aux contraintes résiduelles générées par les constituants	10
1.5	Schéma du procédé de pultrusion	10
1.6	Figure d'un dispositif de drapage assisté par robot	11
1.7	Schéma du procédé de contrôle non destructif	12
2.1	Les nœuds du domaine Ω en (x,y) , 1735 nœuds	30
2.2	Maillage dans l'épaisseur, 100 nœuds	30
2.3	Coupe de la solution 3D du problème-exemple	31
2.4	Algorithme de calcul PGD, n étant le nombre de modes calculés	33
2.5	Solution PGD dans un domaine 1D avec la position du défaut en coordonnée	36
2.6	Solution éléments finis pour toute position du défaut	37
2.7	Erreur relative entre la solution éléments finis et la solution PGD	38
2.8	Solution PGD en 2D pour deux positions possibles du défaut	38
2.9	Détection d'un défaut par la méthode proposée	39
2.10	Solution donnée par la DPGD pour $f = 0$ et $L = 1$	44
2.11	Solution donnée par la DPGD pour $f = 10$ et $L = 1$	45
3.1	Schéma global illustrant le travail	79
4.1	Solution du problème à $t = 1$ s	112
4.2	Solution du problème à $t = 10$ s	112
4.3	Flux dans la plaque pour $t = 1$ s	113
4.4	Flux dans la plaque pour $t = 10$ s	113
4.5	Mécanisme de "transport" de la chaleur dans une plaque stratifiée	114
4.6	Champ de température dans la plaque formée de quatre périodes à $t = 1$ s	114

4.7	Flux dans la plaque formée de quatre périodes à $t = 1 s$	114
4.8	Maillage de la plaque dans le domaine Ω 2D	115
4.9	Champ de température dans la plaque formée de deux plis à $t = 10 s$	115
4.10	Carte des gradients dans la plaque formée de deux plis $t = 10 s$	115
4.11	Carte des flux dans la plaque formée de deux plis $t = 10 s$	116
4.12	Champ de température pour un empilement $[0^\circ/45^\circ/90^\circ/ - 45^\circ] \times 4$	116
4.13	Carte des gradients pour un empilement $[0^\circ/45^\circ/90^\circ/ - 45^\circ] \times 4$ à $t = 10 s$	116
4.14	Les flux dans la plaque d'empilement $[0^\circ/45^\circ/90^\circ/ - 45^\circ] \times 4$ à $t = 10 s$	117
4.15	Flux dans la plaque à $0^\circ/90^\circ$, $45^\circ/-45^\circ$, $30^\circ/60^\circ$ et $30^\circ/120^\circ$ à $t = 1s$	118
4.16	Illustration du drapage assisté par robot, modèle réel	119
4.17	Schéma du modèle local du procédé de dépose de bandes	120
4.18	Simulation 1D d'une résistance thermique	122
4.19	Modélisation des résistances de contact	123
4.20	Solution à la surface supérieure de la pièce, sans stabilisation	124
4.21	Fonctions de forme utilisant la méthode "Upwind"	124
4.22	La pièce simulée en 3D	126
4.23	Coupe au milieu de la pièce drapée montrant les champs de température	126
4.24	Champs de température suivant l'épaisseur sous le jet laser	126
4.25	Zoom sur une résistance interplis	126
4.26	La pièce simulée en 3D à vitesse et résistance maximale	129
4.27	La pièce simulée en 3D à vitesse minimale et résistance maximale	129
4.28	La pièce simulée en 3D à vitesse maximale et résistance minimale	129
4.29	La pièce simulée en 3D à vitesse et résistance minimale	129
4.30	L'effet de la vitesse sur la température maximale de la pièce	130
4.31	L'effet de la source de chaleur sur la température de la pièce	130
4.32	L'effet des résistances inter-plis h_1 à gauche et h_2 à droite	131
4.33	Une application tablette exploitant la solution 8D, développée par F. Bordeu	131
4.34	La déformation d'une plaque due à un champ de température	133
4.35	Enveloppe d'une pièce déformée par l'élasticité thermique	134
5.1	Le recalage classique, \mathbf{d} étant les paramètres	137
5.2	L'idée proposée	137
5.3	Approximation de l'histoire thermique	139
5.4	Solution pour $\theta_1 = 50$, $\theta_2 = 40$, $\theta_3 = 60$ et $\theta_4 = 70$	140
5.5	Flux en $x = 0$ pour $\theta_1 = 50$, $\theta_2 = 40$, $\theta_3 = 60$ et $\theta_4 = 70$	141
5.6	Le gradient désiré est en rouge, en bleu tireté les itérations successives	142

5.7	Champ de température “consigne” du problème	143
5.8	Solution par calcul inverse du problème	143
5.9	Maillages de résolution, les cercles rouges indiquent le maillage grossier M	144
5.10	Particularisation de la solution pour un choix donné de \mathbf{d}	145
5.11	Solution reconstruite après identification des conditions aux limites	147
5.12	Illustration de l’extension de domaine	148
5.13	Exemples de solution pour différents α_i	151
5.14	Le maillage de γ	152
5.15	u_d (en rouge) contre $U _\gamma$ (en bleu)	152
5.16	Un deuxième exemple: Maillage de γ	153
5.17	Un deuxième exemple: u_d (en rouge) contre $U _\gamma$ (en bleu)	153
5.18	Partition de γ	154
5.19	La solution calculée avec $u_d = 10$ et $u_{ext} = 0$.	155
5.20	Mapping du domaine paramétrique	155
5.21	Solution PGD pour $\lambda_1 = \lambda_2 = -0.3$	157
5.22	Solution éléments finis	157
5.23	Solution PGD pour $\lambda_1 = \lambda_2 = 0.25$	157
5.24	Solution éléments finis	157
5.25	Erreur en norme L_2 pour λ_1 et $\lambda_2 \in [-0.5, 0.5]$	158
6.1	Température \mathcal{U} de la résine pour le cycle optimal	161
6.2	Réticulation α de la résine pour le cycle optimal	161
6.3	Illustration de la cuisson à optimiser	162
6.4	Schémas d’un cycle de cuisson pour un stratifié	162
6.5	Une particularisation de la solution homogène paramétrique	165
6.6	La solution $\mathcal{U}_h + \mathcal{U}_p$ sur un snapshot	167
6.7	La solution différences finies sur un snapshot	167
6.8	La solution $\mathcal{U}_h + \mathcal{U}_p$ non coïncidant avec un snapshot	167
6.9	La solution différences finies loin du snapshot	167
6.10	Erreur relative entre notre méthode et les différences finies sur un snapshot	168
6.11	Erreur relative entre notre méthode et les différences finies loin du snapshot	168
6.12	Frontière de Pareto pour une plaque de 10mm d’épaisseur	169
6.13	Frontière de Pareto pour une plaque de 20mm d’épaisseur	169
6.14	Surface de Pareto pour deux plaques ayant 10mm et 20mm d’épaisseur	170
6.15	Décomposition de la source de chaleur dans un intervalle Δt	171
6.16	Paramétrisation de la fonction \mathcal{S} en x , constante sur un pas de temps Δt	171

6.17	Maillage du temps : Δt est le pas grossier	172
6.18	La condition initiale, paramétré par les valeurs de θ_i	172
6.19	Solution homogène non convergée après 30 modes	175
6.20	Solution non convergée à gauche et la solution de référence à droite	175
6.21	L'erreur en norme L_2 en utilisant $N = 30$ à gauche et $N = 300$ à droite	176
6.22	Solution linéaire convergée avec $U_0 = 80^\circ\text{C}$	176
6.23	L'erreur en norme L_2 en négligeant U^{p_1}	178
6.24	Les solutions en PGD à gauche et en différences finies à droite	178
6.25	L'erreur en norme L_2 en tenant compte de U^{p_1}	179
6.26	Algorithme de recollage des sous domaines de temps Δt	180
6.27	Comparaison des temps de calcul entre l'évaluation PGD et différences finies	182
6.28	Solution finale après recollage pour un terme source paramétré en espace x	183
6.29	Erreur en norme L_2 de la solution recollée par rapport à la vraie solution	183
6.30	Solution différences finies en rouge et solution après recollage en couleur	184
6.31	L'erreur en norme L_2 après recollage de trois sous-domaines	184
B.1	Solution du problème modèle pour $K = 1$ et $v = 20$	208
B.2	$\hat{\sigma}$ en bleu et σ^\dagger en rouge	209
B.3	$\hat{\sigma}$ en bleu et σ^\dagger en rouge après un premier raffinement	210
B.4	η après un premier raffinement du maillage	211
B.5	\mathcal{U} après convergence de l'algorithme de raffinement	211
B.6	$\hat{\sigma}$ en bleu et σ^\dagger en rouge après convergence du raffinement	212
B.7	η en fonction du nombre d'itération de raffinement	213
B.8	La solution \mathcal{U} avec une convection suivant x et y	214
B.9	\mathcal{U} après raffinement en x et en y	214



Liste des tableaux

1.1	La comparaison des méthodes de réduction de modèle	23
2.1	Les valeurs de $\frac{Res_i}{Res_1}$ pour les deux méthodes de calcul PGD	30
2.2	Les valeurs utilisées lors de la résolution du problème posé	37
4.1	Valeurs des paramètres utilisés pour le drapage	125
4.2	Calcul nécessaire pour converger le problème de drapage paramétré	130
6.1	Les paramètres du modèle de Bailleul utilisés dans la simulation	162



Introduction

De nos jours, l'usage des matériaux composites est en perpétuelle augmentation. Ce succès est dû principalement aux propriétés mécaniques comparables à ceux des matériaux métalliques, tout en conservant une faible densité. En conséquent, l'usage industriel de tels matériaux s'est popularisé par soucis d'économie d'énergie, une ressource de plus en plus coûteuse dans la société moderne. Ainsi, de nombreux domaines tels l'aéronautique, le transport terrestre, la construction... utilisent des matériaux composites.

Plusieurs procédés sont employés pour élaborer des produits en composites. Cependant cette fabrication est délicate. En effet, cette dernière s'accompagne nécessairement de la création de contraintes résiduelles. Evidemment, ces contraintes dépendent de plusieurs paramètres matériau ou procédé, tel que la nature des constituants, l'épaisseur de la pièce, la température de cuisson... Plusieurs travaux ont déjà visé la quantification de ces contraintes résiduelles. Classiquement, on considère le point de gel de la résine comme étant le critère d'apparition des contraintes. Cependant, l'interprétation mécanique d'une telle approche reste floue (Msallem, 2008). En effet, il est plus intéressant de considérer que les contraintes commencent à se former dès le début de la cuisson. Ainsi, la compréhension de l'histoire thermique exacte en tout point de la pièce devient nécessaire afin de quantifier correctement les contraintes résiduelles générées.

La modélisation 3D étant très coûteuse du point de vue du temps de calcul, la simulation des plaques composites classiquement fait appel à des hypothèses de comportement dans l'épaisseur. Cependant, considérant la richesse du point de vue physique de ces matériaux, le comportement dans la direction de l'épaisseur n'est ni intuitif ni évident. Ces méthodes ont été donc remises en question.

Récemment, de nouvelles méthodes de calcul ont vue le jour. Ces méthodes, appelées “méthodes de réduction de modèle”, promettent la résolution de problèmes multidimensionnels sans avoir à payer le coût du calcul associé aux approches classiques. La résolution de problèmes simplement inimaginables par les techniques classiques devient alors accessible. Généralement ces méthodes se basent sur la représentation séparée des tenseurs et sur les bases réduites.

Dans le chapitre 1 de cette thèse, nous présenterons brièvement les problèmes des matériaux composites, ainsi que les solutions numériques offertes par les méthodes de réduction de modèles. Cependant, plusieurs questions se posent quand on évoque la réduction de modèles. Comment construire la base réduite pour une solution inconnue? Comment résoudre des problèmes multidimensionnels avec un temps de calcul raisonnable? La réponse réside dans la séparation des tenseurs en produits de tenseurs d’ordres inférieurs. Trois grandes familles de méthodes existent de nos jours. De plus, toutes les trois utilisent cette technique de séparation, chacune d’une façon différente. Nous présenterons brièvement les grandes idées de chaque technique. A la fin du premier chapitre, une petite comparaison des trois méthodes sera réalisée.

Dans un deuxième temps, la “PGD”, ou “Proper Generalized Decomposition”, une de ces méthodes, sera détaillée. La PGD est basée sur la séparation des coordonnées. Elle porte ainsi quelques ressemblances à la méthode nommée “LArge Time INcrement” ou “LATIN” qui a été introduite par [Ladeveze \(1985b,a\)](#) où la séparation espace-temps était une des ingrédients principaux de cette méthode ([Ladeveze, 1996](#)). La PGD sera utilisée par la suite. Par conséquent, une formulation efficace qui s’applique à plusieurs problèmes multidimensionnels sera nécessaire. Cette formulation sera présentée dans le chapitre 2. D’autres formulations plus spécifiques à des problèmes plus particuliers seront aussi présentées.

Plus tard, des simulations des champs de température rencontrés lors de la mise en forme de matériaux composites seront effectuées. De plus, ajoutant les paramètres du procédé comme coordonnées supplémentaires du problème, on obtiendra des solutions multidimensionnelles paramétriques. Ces dernières ouvrent la porte du traitement de problèmes en temps réel, tel que la DDDAS ou le contrôle de procédés. Ceci sera achevé en divisant le traitement du problème en deux parties. Premièrement, des calculs “off-line” produisent des solutions paramétriques. Dans un deuxième temps, des algorithmes d’identification “on-line” particularisent la solution. La grosse partie du calcul étant achevée “off-line”, on pourra utiliser de



plateformes de calcul légère “on-line”, tel que des téléphones portables ou des tablettes. Le chapitre 3 présentera une illustration de cette démarche dans la cadre de la pultrusion, un procédé de fabrication de matériaux composites à section constante.

D’autre part, dans le chapitre 4, on s’intéressera à la solution de problèmes thermiques multidimensionnels rencontrés lors de la mise en forme de matériaux composites par dépose de bandes. D’abord, la différence entre des solutions réellement 3D et les solutions 2D associées à des lois de comportement dans l’épaisseur sera montrée. Dans un deuxième temps, le calcul de la solution thermique multidimensionnelle de la dépose de bande sera présenté. La simulation de cette dernière a toujours été un défi pour les numériciens du fait de la complexité du domaine et de la richesse de la solution dans la direction de l’épaisseur. Une application sur tablette exploitant la solution en temps réel, ainsi que le post traitement qui en découle sera montrée.

Le chapitre 5 présentera la PGD appliquée aux domaines de l’identification et du calcul inverse. En effet, le développement rapide des méthodes de traitement d’images renforce l’intérêt pour les problèmes aux conditions limites redondantes sur une partie du bord et inconnues sur l’autre (Andrieux et Baranger, 2008), aussi connu sous le nom de “problème de Cauchy”. Le problème est formulé comme la recherche de conditions limites manquantes sur une partie du domaine à partir de conditions limites redondantes sur une autre partie. Les applications ne se limitent pas au traitement d’image mais s’étendent à l’identification de conditions aux limites inconnues, aux problèmes inverses géométriques, à l’estimation des zones de contacts et beaucoup d’autres. Cependant, la résolution d’un problème inverse par une méthode itérative demande traditionnellement la résolution de nombreux problèmes directs. Par contre, utilisant la PGD, on peut extraire la résolution du problème direct en dehors de la boucle d’optimisation. On présentera diverses formulations permettant d’utiliser la représentation séparée propre à la PGD pour des problèmes sur des géométries complexes.

La résolution de problèmes multidimensionnels non-linéaires par des techniques de réduction de modèle n’est pas encore évidente. En effet, ces problèmes sont très difficiles à résoudre du fait des itérations successives dans des espaces de haute dimensionnalité. Le chapitre 6 traitera un problème modélisé par l’équation de la chaleur non linéaire couplé à une cinétique de réticulation. La solution paramétrique obtenue sera utilisée pour réaliser une optimisation multiobjectifs. Dans un deuxième temps, un problème thermique “linéarisé” englobant de la DDDAS sera résolu, et une stratégie de contrôle sera présentée. Une comparaison avec les méthodes classiques montrera l’efficacité de la stratégie utilisée.



1

Les matériaux composites, mise en forme et simulation

1.1	Mise en forme des composites, nécessité d’une simulation “riche”	5
1.1.1	Les contraintes résiduelles	6
1.1.2	Procédés de mise en forme	9
1.1.3	Contrôle non destructif et défauts de surfaces	11
1.2	Méthodes de réduction de modèle	13
1.2.1	Proper Orthogonal Decomposition	14
1.2.2	Reduced Basis	16
1.2.3	Proper Generalized Decomposition	17
1.2.4	Comparaison des différentes techniques de réduction de modèles	21
1.3	Positionnement du travail	23

Dans ce chapitre nous résumons l’état de l’art concernant la thèse. Dans un premier temps, nous parlerons des matériaux composites et de quelques problèmes de fabrication. La thèse portera sur quelques niches de fabrication de ces matériaux, tel que la dépose de bande ou la pultrusion par exemple. De plus, des techniques de réduction de modèle sont utilisées pour la simulation des différents procédés, ainsi nous présenterons brièvement ces différentes techniques en les comparant.

1.1 Mise en forme des composites, nécessité d’une simulation “riche”

Un matériau composite est un assemblage d’au moins deux matériaux non miscibles, ayant une capacité d’adhésion (Ashby et Jones, 1991). Depuis longtemps, les matériaux composites sont très utilisés. On pourra citer le matériau le plus populaire qui est le béton, un composite de ciment, sable, granulats et fer. De nos jours, les matériaux composites ont acquis de plus en plus de place dans les agendas des chercheurs, puisqu’ils disposent d’atouts importants par

rapport aux matériaux traditionnels (Berreur *et al.*, 2002). En effet, ils apportent de nombreux avantages fonctionnels comme la légèreté tout en gardant la résistance mécanique et chimique, une meilleure isolation thermique ou phonique... Ces performances remarquables sont à l'origine de solutions technologiques innovantes.

En général les matériaux composites sont formés d'un mélange de renfort avec une matrice. Les renforts sont présents pour donner aux matériaux composites leurs propriétés mécaniques, par l'adaptation de la densité du renfort, son orientation ou sa nature (fibre de verre, de carbone...). D'autre part, la matrice répartit les charges entre les fibres et assure la flexibilité des matériaux composites, clé principale dans la mise en forme. D'ailleurs, il existe deux types de matrices, les matrices thermoplastiques et les matrices thermodurcissables. Les matrices thermoplastiques ont des propriétés physiques réversibles en fonction de la température. En d'autres termes, on peut les faire fondre en augmentant la température. Au contraire, les matrices thermodurcissables comme leur nom l'indique, durcissent en cours de cuisson, et gagnent des propriétés chimiques irréversibles.

Les principales caractéristiques des matériaux composites sont donc obtenues grâce à la combinaison des caractéristiques des différents composants. Cependant cette structure non homogène est à l'origine de plusieurs problèmes de mise en forme non rencontrés avec les matériaux homogènes. On pourra citer les contraintes résiduelles même en très faible épaisseur (Msallem *et al.*, 2010a,b), le délaminage (Gornet *et al.*, 2000)... Ces problèmes sont loin d'être résolus, et varient d'un procédé de formage à l'autre. Dans cette thèse, nous nous intéressons à la bonne modélisation des champs de température dans ces composites, comme point de départ afin de mieux aborder les couplages existants.

1.1.1 Les contraintes résiduelles

Bien que ces matériaux semblent être des solutions industrielles pour plusieurs cas intraitables, beaucoup de problèmes sont rencontrés lors de leur fabrication. Principalement, les contraintes résiduelles qui sont inévitables dans les matériaux composites (Msallem, 2008). Ces contraintes omniprésentes dans les matériaux composites peuvent avoir des effets significatifs sur les performances mécaniques de la structure provoquant des délaminages, fissures (Low, 1990; Sirivedin *et al.*, 2000) ou la distorsion de pièces (Wissnom *et al.*, 2006; Svanberg et Holmberg, 2004; Albert et Fernlund, 2002). Pour ces raisons il est nécessaire de pouvoir modéliser ces contraintes afin d'améliorer les procédés de formage.



Dans le cas des matériaux thermodurcissables, les contraintes résiduelles sont présentes dans la pièce pendant la cuisson et après la phase de solidification (Msallem, 2008). Ces contraintes sont classées en trois catégories ou niveaux, en fonction de leurs origines (Favre, 1988; Manson et Barbero, 1992).

Niveau global ou de la pièce

Au niveau global, la présence de gradients de température et de vitesse de réticulation non uniforme (comme le montre la figure 1.1) sont à l'origine de nouvelles contraintes résiduelles (Bailleul *et al.*, 1994, 1996a). La figure 1.2 illustre un exemple de contraintes générées de ce fait. Ces contraintes sont accentuées par l'augmentation de l'épaisseur et la faible conductivité thermique de la matrice (Boyard *et al.*, 2007). En général lors de la réaction chimique de cuisson, la température au centre de la pièce est plus importante qu'à la surface. De même pendant le refroidissement, le centre se refroidit plus lentement que la surface. Ces phénomènes ont été longtemps étudiés afin de modéliser la répartition des contraintes résiduelles suivant l'épaisseur (Bogetti et Gillespie, 1992; Chapman *et al.*, 1990).

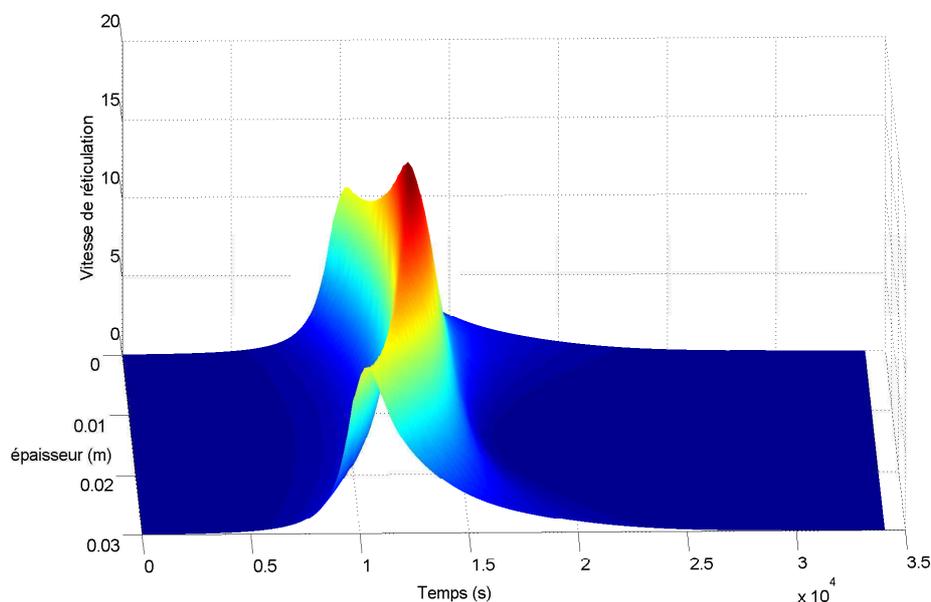


figure 1.1 – Exemple de la cinétique de réticulation selon le modèle de Bailleul

La figure 1.1 montre la cinétique de réticulation de la résine selon le modèle de Bailleul. On remarque que la réaction s'enclenche sur les bords, ce qui chauffe les bords énormément avant la transmission de cette chaleur au centre de la pièce. Plus tard, cette réaction s'enclenche

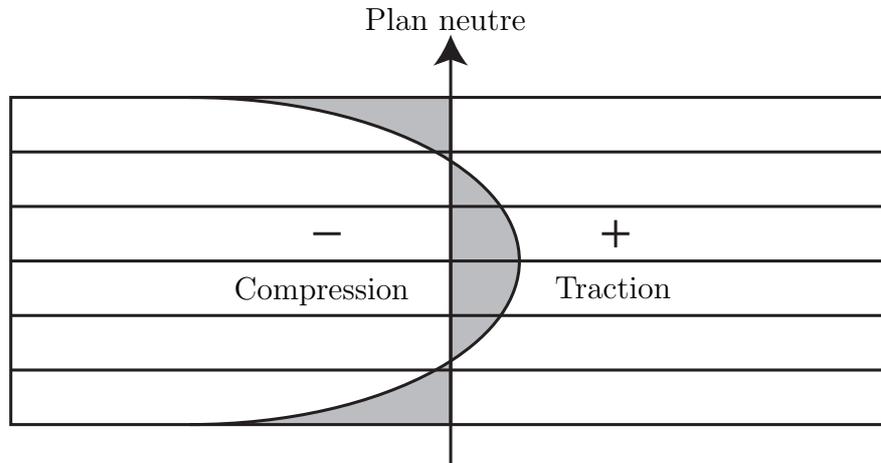


figure 1.2 – Exemple de contraintes générées au niveau global

de façon plus violente au centre de la pièce, alors que les bords commencent à se refroidir. Cette histoire thermique non homogène engendre des contraintes résiduelles de compression sur les bords et de traction au centre de la pièce, comme illustré à la figure 1.2.

Niveau des plis

Les contraintes résiduelles sont aussi induites par la continuité du déplacement entre les plis. Ainsi, la différence d'orientation des plis, des coefficients de dilatation thermique et de retrait chimique inter plis influencent énormément la formation des contraintes résiduelles (Msallem, 2008). La figure 1.3 illustre un exemple de contraintes générées par le placement des plis.

Niveau des constituants

Le renfort et la matrice n'ont pas forcément les mêmes coefficients de dilatation thermique. Evidemment ceci est un paramètre prépondérant pour la formation de contraintes résiduelles. Un autre facteur très important est le retrait volumique de la matrice, surtout dans le cas des matrices thermodurcissables. Ce phénomène est à l'origine de la fissuration au jeune âge des matrices (Cortas, 2012). Ce genre de contraintes résiduelles est sensible aux proportions et volume de fibres. On démontre que les macro-fibres provoquent davantage de fissurations que les microfibrilles (Banthia *et al.*, 1993). Les contraintes résiduelles générées au niveau des constituants dépendent aussi de la nature de la matrice utilisées. La figure 1.4 montre les fissures dans une matrice sédimentaire dues à ces contraintes résiduelles (Cortas, 2012).

Les théories classiques des plaques réduisent le cas 3D à un cas 2D en introduisant des hypothèses de comportements prédéfinies dans l'épaisseur. Vis à vis des problèmes rencontrés dans

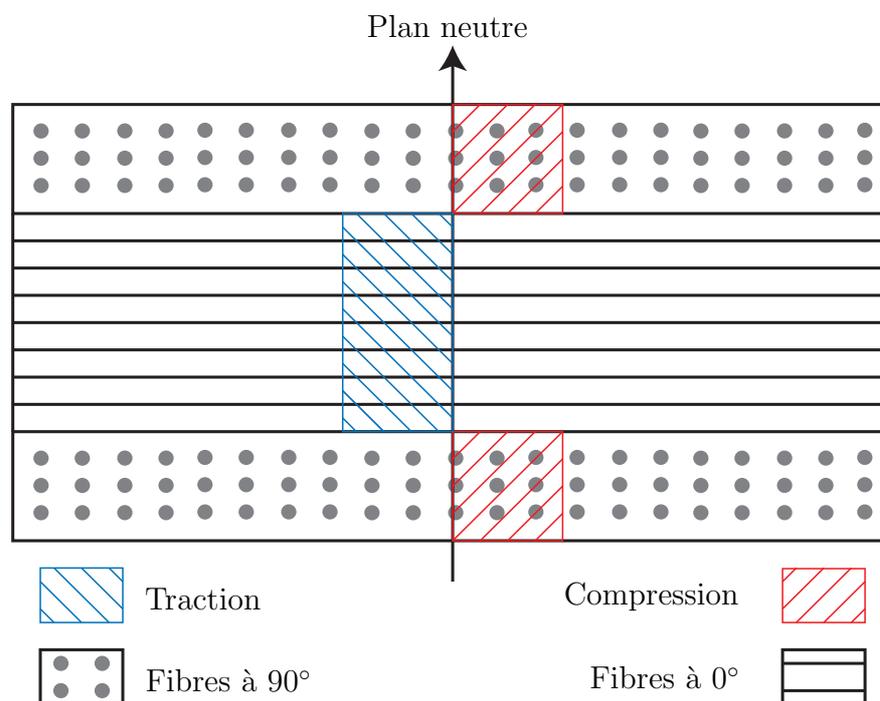


figure 1.3 – Exemples de contraintes résiduelles générées au niveau des plis

les matériaux composites, ces hypothèses de comportements ne sont plus évidentes ni intuitives. Par suite, une simulation 3D est souvent nécessaire (Bognet *et al.*, 2011; Chinesta *et al.*, 2010c). D'autre part, cette simulation 3D s'avère très couteuse en terme de temps de calcul si on utilise des techniques classiques de calcul (Marcin *et al.*, 2009). Ainsi, on devra avoir recours à des méthodes de calcul avancées telles que les méthodes de réduction de modèle.

1.1.2 Procédés de mise en forme

Les procédés de mise en forme des composites sont très nombreux. Dans ce qui suit on énoncera deux procédés particuliers de fabrication étudiés dans le cadre de cette thèse. Ces procédés sont particulièrement intéressants du fait de la difficulté inhérente à leur simulation.

La pultrusion

La pultrusion est un procédé de production continue de matériaux composites à section constante (Chen X, 2010). Dans ce procédé, les fibres sont imbibées de résine puis traversent un four où la résine réticulera pour former le composite. A la sortie, ces composites sont coupés à la longueur désirée. Le procédé est illustré dans la figure 1.5. Traditionnellement, les ingénieurs s'appuyaient sur l'expérience et le savoir faire pour déterminer les paramètres



figure 1.4 – Fissuration due aux contraintes résiduelles générées par les constituants

de ce procédé (Martin, 2006). Cependant, vu l'intérêt croissant dans l'industrie pour cette technique, une optimisation avancée doit être réalisée (Jacob, 2006).

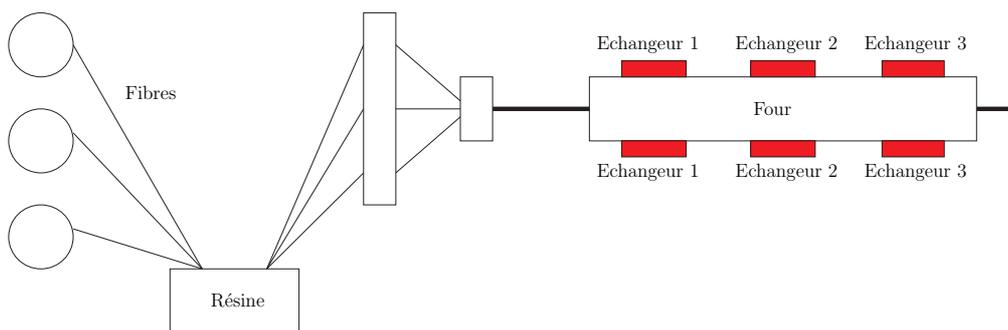


figure 1.5 – Schéma du procédé de pultrusion

Le drapage assisté

Le drapage assisté par robot est un autre procédé de fabrication de matériaux composites par dépose de bandes. Le dispositif est formé d'une tête de robot qui vient déposer des plis

au fur et à mesure de l'avancement du procédé de formage. Au cours de la dépose, une source de chaleur chauffe la matrice pour permettre l'adhésion du nouveau pli. Finalement, un rouleau exerce une pression sur l'empilement afin de réaliser la consolidation du nouveau pli (Barasinski *et al.*, 2012; Ghnatios *et al.*, 2011a). Le dispositif est illustré à la figure 1.6.

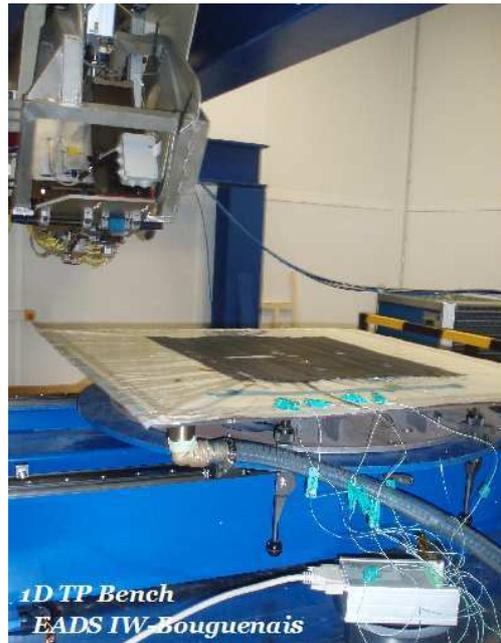


figure 1.6 – Figure d'un dispositif de drapage assisté par robot

La simulation du drapage assisté par robot est intraitable utilisant les techniques de calcul classiques. D'une part le traitement de la vitesse d'avancement de la dépose nécessite un maillage espace-temps extrêmement fin (Donea et Huerta, 2003; Scovazzi, 2007). D'autre part la résistance thermique de contact, générée entre les bandes, induites par un contact imparfait des plis entre eux, exige un maillage fin dans l'épaisseur. De ce fait, les théories classiques de loi de comportement, ou le calcul par éléments finis, ne sont pas adaptées pour simuler efficacement ce procédé.

1.1.3 Contrôle non destructif et défauts de surfaces

De nos jours les tests non destructifs occupent un large espace dans les agendas des chercheurs. De plus la demande industrielle du perfectionnement de ce genre de tests est en perpétuelle augmentation. Un des tests les plus populaires est la thermographie infrarouge stimulée, vu son potentiel d'automatisation important (Maillard *et al.*, 2008). La mesure par

thermographie infrarouge a été introduite à la fin des années quatre-vingts. Elle permet une mesure globale et sans contact de la température des objets présents dans la scène thermique analysée (Pajani, 1989). Ceci est bien adapté aux cas où les gradients de température sont très élevés et les coefficients de dilatation thermique sont faibles. En revanche il faudra une excitation extérieure pour les autres applications, d'où le nom thermographie infrarouge "stimulée" (Maillard *et al.*, 2009). Le procédé est illustré dans la figure 1.7.

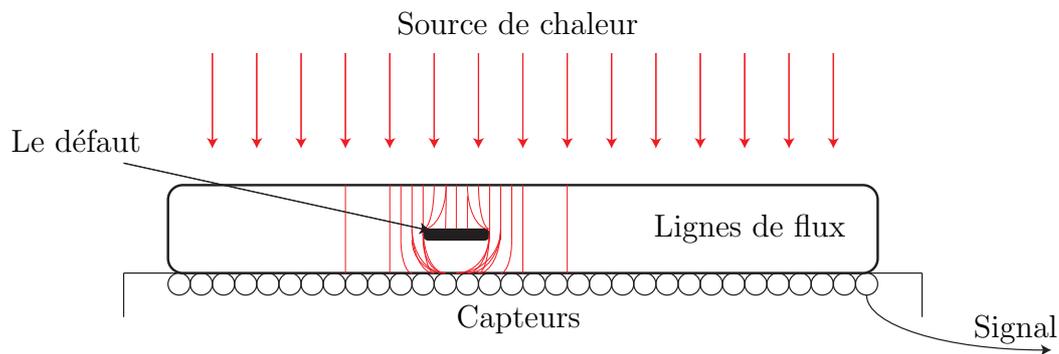


figure 1.7 – Schéma du procédé de contrôle non destructif

Le contrôle non destructif utilisant la méthode de thermographie infrarouge consiste d'abord à capter le comportement thermique des pièces par la thermographie, pour ensuite réaliser des étapes d'inversion afin d'établir le diagnostic des défauts. Dans les dernières années, la résolution de problèmes inverses a donc été appliquée à l'identification des défauts de surfaces ou des résistances thermiques de contact... Plusieurs méthodes ont été testées pour réaliser les étapes d'inversion, analytiques comme les méthodes intégrales (Maillet *et al.*, 1996), discrètes (Niliot et Gallet, 1998), réduction de modèle (Grinzato *et al.*, 1995) et bien d'autres (Krapez *et al.*, 1994).

L'emploi des techniques analytiques basées sur des transformations d'intégrales en temps et en espace permet de calculer rapidement des champs de température dans des géométries simples. Cependant, l'utilisation de cette méthode dans des modèles bidimensionnels ou tridimensionnels nécessite une régularisation de la troncature associée à l'utilisation d'une telle méthode (Maillet *et al.*, 1996).

Dans la modélisation numérique, ce problème d'inversion de données est connu pour être mal posé (Niliot et Gallet, 1998). En effet, la solution obtenue est très sensible aux incertitudes des mesures expérimentales. Plusieurs méthodes de traitement des problèmes inverses ont été proposées. Les traitements se basent souvent sur la régularisation de la solution, ou le filtrage

des données d'entrées.

De nos jours, les caméras infrarouges matricielles promettent de populariser ce type de contrôle dans l'industrie. Par contre, le coût principal de ce contrôle non-destructif réside principalement dans la difficulté de la mise en œuvre des étapes d'inversion pour établir le diagnostic. Ainsi, une idée prometteuse serait de pouvoir calculer toutes les solutions possibles pour tout défaut de surface, et ceci à priori. Par suite le problème inverse ne fera plus appel à des résolutions du problème direct, mais à des particularisations de la solution pré calculée. Classiquement, ceci induit énormément de résolutions de problèmes directs, et une utilisation énorme de mémoire afin de stocker les solutions pré calculées. Par suite, l'utilisation de méthodes de calcul classiques s'avère impossible. Une solution envisagée est donc l'utilisation des méthodes de réduction de modèle, ce qui sera présenté dans la section 1.2 (Chinesta *et al.*, 2009a).

1.2 Méthodes de réduction de modèle

L'objectif principal de la réduction de modèle est le calcul de solutions de problèmes que l'on trouve en sciences et ingénierie avec toutes leurs complexités géométriques et leurs comportements thermomécaniques. Plusieurs d'entre eux n'ont jamais été résolus auparavant à cause de la complexité numérique engendrée par tels problèmes. Ces modèles nécessitent d'être résolus très vite, dans certains cas en temps réel, et si possible, ce qui semble antinomique, en ne faisant appel qu'à des moyens de calcul très légers. Les techniques de simulation classiques comme nous venons de le voir dans la section 1.1 ne sont pas capables, du moins à elles seules, de répondre à ce cahier de charges.

Les modèles d'ordre réduit sont des modèles de substitution, qui remplacent donc les modèles de référence, dont les inconnues doivent vérifier des équations de bilan physique. Ces modèles tendent généralement à réduire le temps de calcul, sans nécessairement réduire la complexité mathématique du modèle, cependant, parfois ils l'amplifient. Souvent, le temps de calcul est réduit au prix de l'exactitude de la solution. En général les modèles d'ordre réduit doivent préserver (Amsallem et Farhat, 2011) :

- La facilité du contrôle.
- La possibilité d'application à un grand nombre de systèmes.
- La tendance à minimiser l'erreur avec le modèle de référence si on augmente le temps de calcul.



- Les propriétés du modèle de référence.

Ainsi, il existe à présent trois grandes familles de technique de réduction de modèle. Les trois se basent sur la représentation séparée des tenseurs. Dans ce qui suit nous illustrons brièvement les caractères et propriétés fondamentales de chaque méthode.

1.2.1 Proper Orthogonal Decomposition

La POD ou “Proper Orthogonal Decomposition”, connue aussi sous le nom de “Karhunen Loève Expansion” ou KLE (Vorechovsky, 2008; Loeve, 1945) est à la base un développement spectral d’un tenseur par rapport aux valeurs propres. En pratique, la POD est une technique de réduction de modèle qui génère un sous-espace linéaire optimal de dimension D à partir d’un tenseur de dimension supérieure à D . De plus, la POD peut générer le produit de fonctions affines qui génère au mieux le tenseur, et ceci en considérant des sous-espaces 1D. Par suite, on peut écrire un tenseur \mathbf{V} comme étant la somme de produits de fonctions des coordonnées de ce dernier, par exemple :

$$\mathbf{V}(\mathbf{x}, t) \approx \sum_{i=1}^{i=N} \alpha_i \mathbf{X}_i(\mathbf{x}) T_i(t) \quad (1.1)$$

Avec \mathbf{V} le tenseur, \mathbf{X}_i et T_i les fonctions des coordonnées \mathbf{x} et t respectivement, α_i étant le poids ou la valeur propre correspondant à ce produit de fonction. On réfère dans la suite de cette thèse à un produit de fonction dans la somme par “un mode”. Evidemment un des intérêts que porte cette notation est la réduction de la quantité de mémoire utilisée pour stocker le tenseur (Kolda et Bader, 2009).

Dans un espace 2D, la séparation d’un espace discret se fait par la séparation aux valeurs singulières SVD. Cela donne la meilleure approximation utilisant la norme L_2 . Ainsi en considérant \mathbf{x}_k et \mathbf{t}_l deux maillages d’un espace unidimensionnel et du temps on peut écrire :

$$\mathbf{V}(\mathbf{x}_k, \mathbf{t}_l) = \sum_{i=1}^{i=N} \alpha_i \mathbf{X}_i(\mathbf{x}_k) \mathbf{T}_i(\mathbf{t}_l) \quad (1.2)$$

Les techniques de résolution d’équations aux dérivées partielles basées sur la POD sont relativement anciennes. Elles ont été largement utilisées dans différentes applications. En effet, elles sont basées sur l’extraction des modes les plus significatifs associés à la solution d’un problème. Ces modes sont ensuite utilisés comme fonctions de base dans la résolution d’autres problèmes “similaires” (Liberge et Hamdouni, 2010; Allery *et al.*, 2005) (différents



par les conditions initiales ou aux limites, les paramètres des matériaux, les paramètres définissant la géométrie...) ou encore pour résoudre le même problème qui avait servi à extraire les fonctions de la base réduite mais dans des intervalles temporels bien plus large que celui dans lequel la résolution avait permis l'extraction des fonctions (Atwell et King, 2001).

Pour illustrer la résolution par la POD, considérons l'équation de la chaleur unidimensionnelle suivante :

$$\frac{\partial \mathcal{U}}{\partial t} - \frac{\partial}{\partial x} \left(K \frac{\partial \mathcal{U}}{\partial x} \right) = f \quad (1.3)$$

Où K est la conductivité, U la température, f une source de chaleur, x la variable d'espace et t le temps. Pour simplifier les notations, K est considérée pour l'instant constante. La forme discrète de l'équation (1.3) devient :

$$\frac{\mathcal{U}_{i+1} - \mathcal{U}_i}{dt} - K \mathbf{D} \mathcal{U}_i = f \quad (1.4)$$

Où \mathbf{D} est la matrice de rigidité. Supposons connaître une solution U^\dagger de l'équation (1.3) et que l'on cherche à résoudre cette même équation pour des conditions limites différentes ou des intervalles d'espace et/ou de temps différents. On agit d'abord par une décomposition aux valeurs propres (ou singulières) λ_i et on va tronquer la solution afin de garder les N modes significatifs. La troncature est réalisée quand $\lambda_N/\lambda_1 < \epsilon$. Ainsi la solution U^\dagger s'écrit :

$$\mathcal{U}^\dagger(x,t) \approx \sum_{i=1}^{i=N} \alpha_i X_i(x) T_i(t) \quad (1.5)$$

Avec N suffisamment grand pour satisfaire une condition de convergence donnée. Ou matriciellement :

$$\mathcal{U}^\dagger(x,t) \approx \begin{bmatrix} \vdots & \vdots & & \\ X_1 & X_2 & \cdots & \\ \vdots & \vdots & & \end{bmatrix} \times \begin{bmatrix} \alpha_1 & 0 & & \\ 0 & \alpha_2 & & \\ & & \ddots & \end{bmatrix} \times \begin{bmatrix} \cdots & T_1 & \cdots \\ \cdots & T_2 & \cdots \\ \vdots & & \end{bmatrix} = \mathbf{X} \mathbf{\Lambda} \mathbf{T} \quad (1.6)$$

Par suite, on considère pour le nouveau problème les mêmes X_i ou \mathbf{X} , mais on cherche les T_i ou \mathbf{T} . Ainsi la formulation approchée du problème (1.3) devient :

$$\mathbf{X} \mathbf{\Lambda} \frac{\partial \mathbf{T}}{\partial t} - K (\mathbf{D} \mathbf{X}) \mathbf{\Lambda} \mathbf{T} \approx f \quad (1.7)$$

■

L'équation (1.7) donne un système rectangulaire non inversible, par suite on agit par une méthode de Galerkin où le résidu est orthogonalisé par rapport à \mathbf{X} . Ainsi le système devient :

$$\mathbf{X}^T \mathbf{X} \Lambda \frac{\partial \mathbf{T}}{\partial t} - K (\mathbf{X}^T \mathbf{D} \mathbf{X}) \Lambda \mathbf{T} \approx \mathbf{X}^T f \quad (1.8)$$

Ainsi, résoudre l'équation (1.8) revient à résoudre un système ODE dont la taille est égale au nombre de vecteurs de temps T_i à calculer (ici noté N). Bien que la POD réduise considérablement le temps de calcul, la principale difficulté de cette démarche est précisément l'impossibilité de quantifier l'impact de l'emploi des bases réduites dans des conditions différentes de celles auxquelles la base était associée (Chinesta *et al.*, 2012). La réduction de modèle basée sur l'emploi de la POD a été largement utilisée (Bernardi et Maday, 1997). Des travaux récents sur la POD réalisés par Ryckelynck (2005) permettent plus ou moins de s'affranchir des difficultés rencontrées avec la résolution par POD. Cette technique fut appelée "a priori" et puis on proposa une hyper-réduction basée sur une pondération incomplète de la forme faible du modèle, ceci dans le cadre d'une formulation de type Petrov-Galerkin (Ryckelynck et Benziane, 2010).

1.2.2 Reduced Basis

La RB ou "Reduced Basis" est une deuxième grande famille de techniques. Elle est basée sur la construction et l'emploi de bases réduites semblables à celles de la POD, en utilisant des techniques "intelligentes" pour choisir la base réduite. Apparemment l'idée qui a lancé cette méthode a été introduite par Nagy (1977) pour l'analyse des treillis articulés (Porsching et Lee, 1987). Par suite, la méthode "Reduced Basis" fut présentée dans les années 80 par les travaux de Noor et Peters (1980). Elle est fondée de nos jours sur la connaissance d'estimateurs d'erreurs "a priori". Ces derniers permettent dans un premier temps de construire la base réduite d'approximation et d'autre part de valider les calculs en proposant des bornes assez proches des véritables erreurs (Patera et Ronquist, 2007). Par ailleurs, les techniques d'interpolations empiriques ont permis à la RB le traitement de problèmes fortement non-linéaires. Enfin, le couplage de cette méthode avec les techniques de décomposition de domaine a permis de traiter des problèmes qui apparaissaient impossible à simuler (Maday et Ronquist, 2002). Des développements récents de cette technique assez ancienne ont été réalisés (Patera et Ronquist, 2007; Huynh *et al.*, 2012). De nos jours, cette méthode comprend deux étapes :

- *Première étape* : C'est l'étape de pré calcul. Un algorithme "Glouton" ou "Greedy" en anglais, combiné avec des estimateurs "A priori" est utilisé pour explorer "off-line" l'es-



pace paramétrique multidimensionnel construisant une base réduite.

- *Deuxième étape* : C’est l’étape du modèle en temps réel. La base réduite pré calculée sera employée comme une boîte noire pour une résolution “on-line” des modèles. C’est ce qu’on appelle aussi “The Blackbox Approach” (Machiels *et al.*, 2000).

Dans la première version de la RB introduite par Nagy (1977), elle avait servi à calculer la déformation non linéaire comme une combinaison des modes propres de la solution linéaire, d’une structure soumise à un chargement \mathbf{R} .

La RB proprement dit a été développée ultérieurement par les travaux de Noor et Peters (1980); Fink et Rheinboldt (1983); Porsching (1985). De nos jours, des estimateurs d’erreurs à postériori existent maintenant pour évaluer l’erreur engendrée par cette résolution superposant les modes propres d’une ou de plusieurs solutions (Machiels *et al.*, 2000). Enfin, les résultats théoriques sont confirmés par des simulations numériques tout au long de ces dernières années.

1.2.3 Proper Generalized Decomposition

La PGD ou “Proper Generalized Decomposition” est une technique qui reçoit de plus en plus d’attention de la communauté scientifique (Bungartz et M.Griebel, 2004). Elle combine la représentation séparée à un algorithme qui construit cette dernière progressivement (Chinesta *et al.*, 2009b). Elle comporte une différence majeure avec les deux précédentes qui est le fait de ne pas imposer une base de construction de la solution “à priori”, ni “à postériori”. Par contre, cette base est créée “à la volée” à fur et à mesure de la résolution.

La PGD fut historiquement introduite par Pierre Ladeveze dans les années 80, sous le nom d’approximation radiale, comme un des ingrédients principaux de la technique non-incrémentale-non-linéaire ou “LATIN”(Ladeveze, 1999; Boucard et Ladeveze, 1999). Cependant la PGD généralise l’approximation radiale, qui n’utilise que la séparation espace-temps. Cette approximation séparée en espace et en temps permettant d’écrire :

$$U(\mathbf{x},t) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) T_i(t) \quad (1.9)$$

Dans les années 2000 ce type de représentation séparée a expérimenté deux importantes évolutions. D’une part, elle fut appliquée à la séparation d’une coordonnée stochastique et



une deuxième déterministe dans les travaux de [Nouy \(2007\)](#). D'autre part, elle fut appliquée à la résolution de problèmes multidimensionnels impliquant des coordonnées de type divers : physiques, configurationnelles, paramètres matériaux, conditions initiales, conditions limites ([Ammar et al., 2006](#); [Ghnatios et al., 2011b](#))... Ainsi, si le modèle implique l'espace, le temps et P paramètres p^1, \dots, p^P la forme séparée s'écrit :

$$U(\mathbf{x}, t, p^1, \dots, p^P) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) T_i(t) \prod_{j=1}^{j=P} F_i^j(p^j) \quad (1.10)$$

Pour illustrer la construction de cette solution avec des conditions initiales et limites de Dirichlet homogènes, considérons à nouveau le cas de l'équation unidimensionnelle transitoire de la chaleur (1.3). On cherche à résoudre ce problème dans un domaine Ω en considérant la conductivité k comme paramètre du problème et f une source de chaleur uniforme constante. La solution thermique s'écrit donc comme étant :

$$U(x, t, k) \approx \sum_{i=1}^{i=N} X_i(x) T_i(t) K_i(k) \quad (1.11)$$

où $(x, t, k) \in [\Omega \times I \times \mathfrak{S}]$. Dans ce qui suit, on suppose que les n premiers vecteurs de la base de construction de la solution sont connus et qu'on cherche le mode $n + 1$. Ainsi on a :

$$U_n = \sum_{i=1}^{i=n} X_i(x) T_i(t) K_i(k) \quad (1.12)$$

Par suite on peut écrire que :

$$U_{n+1} = \sum_{i=1}^{i=n} X_i(x) T_i(t) K_i(k) + R(x) S(t) W(k) \quad (1.13)$$

Le calcul de R , S et W est fortement non linéaire, et est résolu par un algorithme itératif de point fixe. Ainsi, on suppose S et W connues pour chercher R . A partir de R on met S à jour puis W . On continue jusqu'à convergence de l'algorithme ([Chinesta et al., 2009b](#)). En effet, considérons la formulation intégrale de l'équation (1.3) :

$$\int_{\Omega \times I \times \mathfrak{S}} U^* \left(\frac{\partial U}{\partial t} - k \Delta U - f \right) dx dt dk = 0 \quad (1.14)$$

1. Calcul de $R(x)$ à partir de $S(t)$ et $W(k)$:



Puisque S et W sont connues on peut écrire :

$$U^*(x,t,k) = R^*(x) S(t) W(k) \quad (1.15)$$

En remplaçant les équations (1.13) et (1.15) dans la formulation intégrale donnée par l'équation (1.14) on obtient :

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{S}} R^* \cdot S \cdot W \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) dx dt dk = \\ & = - \int_{\Omega \times I \times \mathfrak{S}} R^* \cdot S \cdot W \left(\sum_{i=1}^{i=n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{i=n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f \right) dx dt dk \end{aligned} \quad (1.16)$$

En intégrant l'équation (1.16) sur $I \times \mathfrak{S}$ et en considérant les notations suivantes :

$$\left[\begin{array}{lll} w_1 = \int_{\mathfrak{S}} W^2 dk & s_1 = \int_I S^2 dt & r_1 = \int_{\Omega} R^2 d\mathbf{x} \\ w_2 = \int_{\mathfrak{S}} kW^2 dk & s_2 = \int_I S \cdot \frac{dS}{dt} dt & r_2 = \int_{\Omega} R \cdot \Delta R d\mathbf{x} \\ w_3 = \int_{\mathfrak{S}} W dk & s_3 = \int_I S dt & r_3 = \int_{\Omega} R d\mathbf{x} \\ w_4^i = \int_{\mathfrak{S}} W \cdot K_i dk & s_4^i = \int_I S \cdot \frac{dT_i}{dt} dt & r_4^i = \int_{\Omega} R \cdot \Delta X_i d\mathbf{x} \\ w_5^i = \int_{\mathfrak{S}} kW \cdot K_i dk & s_5^i = \int_I S \cdot T_i dt & r_5^i = \int_{\Omega} R \cdot X_i d\mathbf{x} \end{array} \right], \quad (1.17)$$

l'équation (1.16) se réduit à :

$$\begin{aligned} & \int_{\Omega} R^* \cdot (w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R) d\mathbf{x} = \\ & = - \int_{\Omega} R^* \cdot \left(\sum_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) d\mathbf{x} \end{aligned} \quad (1.18)$$

Ce qui constitue la formulation intégrale d'un problème elliptique stationnaire 1D qu'on peut résoudre pour trouver $R(x)$ en utilisant n'importe quelle technique de discrétisation classique après intégration par parties (Chinesta *et al.*, 2010a, 2009b). Une autre possibilité consiste à revenir à la formulation forte du problème :

$$w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R = - \left(\sum_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) \quad (1.19)$$

2. Calculer $S(t)$ à partir de $R(x)$ et $W(k)$:



Dans le cas présent, on peut écrire :

$$U^*(x,t,k) = R(x) S^*(t) W(k) \quad (1.20)$$

Par suite la formulation intégrale (1.14) devient :

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{S}} S^* \cdot R \cdot W \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) dx dt dk = \\ & = - \int_{\Omega \times I \times \mathfrak{S}} S^* \cdot R \cdot W \left(\sum_{i=1}^{i=n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{i=n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f \right) dx dt dk \end{aligned} \quad (1.21)$$

En tenant compte des notations déjà introduites dans l'équation (1.17) et après intégration sur $\Omega \times \mathfrak{S}$, on obtient :

$$\begin{aligned} & \int_I S^* \left(w_1 \cdot r_1 \cdot \frac{\partial S}{\partial t} - w_2 \cdot r_2 \cdot S \right) dt = \\ & = - \int_I S^* \left(\sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{\partial T_i}{\partial t} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) dt \end{aligned} \quad (1.22)$$

L'équation (1.22) représente la forme faible d'un problème d'évolution. Nous résolvons l'équation (1.22) par n'importe quelle méthode de stabilisation (Donea et Huerta, 2003). La forme forte de ce même problème est donnée par :

$$w_1 \cdot r_1 \cdot \frac{\partial S}{\partial t} - w_2 \cdot r_2 \cdot S = - \left(\sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) \quad (1.23)$$

3. Calculer $W(k)$ à partir de $R(x)$ et $S(t)$:

Dans cette partie de l'algorithme, la fonction test s'écrit comme étant :

$$U^*(x,t,k) = R(x) S(t) W^*(k) \quad (1.24)$$

En remplaçant cette fonction test dans l'équation (1.14) qui constitue la forme intégrale du problème on obtient après intégration sur $\Omega \times I$:

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{S}} W^* \cdot R \cdot S \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) dx dt dk = \\ & = - \int_{\Omega \times I \times \mathfrak{S}} W^* \cdot R \cdot S \left(\sum_{i=1}^{i=n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{i=n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f \right) dx dt dk \end{aligned} \quad (1.25)$$

■

De plus, en considérant les notations de l'équation (1.17) on obtient :

$$\begin{aligned} \int_{\mathfrak{S}} W^* (r_1 \cdot s_2 \cdot W - r_2 \cdot s_1 \cdot k \cdot W) dk = \\ = - \int_{\mathfrak{S}} W^* \left(\sum_{i=1}^{i=n} r_5^i \cdot s_4^i \cdot K_i - \sum_{i=1}^{i=n} r_4^i \cdot s_5^i \cdot k \cdot K_i - r_3 \cdot s_3 \cdot f \right) dk \end{aligned} \quad (1.26)$$

L'équation (1.26) représente une simple équation algébrique sans opérateurs différentiels. Par suite, on peut conclure que l'introduction de ce paramètre supplémentaire au modèle n'influe pas considérablement le temps de calcul, mais peut nécessiter plus de termes dans la somme. La formulation forte de ce problème s'écrit donc :

$$(r_1 \cdot s_2 - r_2 \cdot s_1 \cdot k) \cdot W = - \left(\sum_{i=1}^{i=n} (r_5^i \cdot s_4^i - r_4^i \cdot s_5^i \cdot k) \cdot K_i - r_3 \cdot s_3 \cdot f \right) \quad (1.27)$$

Ainsi de suite, on répète la résolution jusqu'à convergence. Ceci constitue l'algorithme le plus basique pour construire une solution par la méthode PGD. Il existe d'autres formulations de la PGD plus robustes (Pruliere *et al.*, 2010a). Une formulation tensorielle plus générale de la PGD est détaillée dans la section 2.1.

La PGD à été testée sur plusieurs applications comme les équations intégrales (Bonithon *et al.*, 2011), problèmes inverses (Ghnatios *et al.*, 2011d,c), la multiphysique, les problèmes thermiques (Ghnatios *et al.*, 2010, 2011e), rhéologie (Chinesta *et al.*, 2011a)... Les travaux récents sont repris dans Chinesta *et al.* (2011b). Mais cette technique reste considérée comme jeune par rapport aux "POD" et "Reduced Basis" décrites dans les sections 1.2.1 et 1.2.2.

1.2.4 Comparaison des différentes techniques de réduction de modèles

D'abord il faut commencer en disant qu'il n'y a pas de technique optimale, tous les modèles sont faux mais certains sont utiles (Box, 1979). Par suite nous allons comparer les modèles suivant certains critères de comparaison, pour pouvoir adapter le choix de la technique par rapport aux problèmes rencontrés, selon le cas traité.

Un des critères majeurs dans le choix de la méthode de réduction de modèle est la capacité de manipuler la multidimensionnalité. Les problèmes multidimensionnels contiennent évidemment un grand nombre de paramètres, sans dérivées explicites de la solution par rapport à ces paramètres. Les paramètres peuvent être intérieurs au problème (conductivité, raideur...) ou



non (paramètres géométriques, conditions aux limites...). Les trois méthodes sont capables de manipuler un grand nombre de paramètres. De plus, la PGD montre quelques capacités supérieures vis à vis des problèmes linéaires. Les trois méthodes sont aussi capables de manipuler des paramètres stochastiques. Dans ce cas, la POD est connue sous le nom de “Karhunen Loève Expansion”.

Par contre, quand on passe à la multi-dimensionnalité physique, espace ND... La PGD reste supérieure. En effet pour la séparation de l’espace physique, quelques travaux ont été récemment publiés en utilisant la RB (Perotto, 2010). Par contre cette séparation est au cœur de la PGD. De plus, en utilisant la PGD nous sommes capables de séparer certaines géométries 3D, plaques, coques ou géométries dégénérées (Gonzalez *et al.*, 2009).

Un autre critère de comparaison est la nécessité de calculer “a priori” une solution, afin de pouvoir l’utiliser dans un calcul “on-line”. Les trois méthodes exploitent cette approche. La PGD pré calcule des solutions mais d’une façon un peu différente. En effet, la RB et la POD calculent ce qu’on appelle “snapshots” afin d’utiliser une méthode de construction adaptative qui dépend du code ou problème à implémenter. Cependant, la PGD calcule la solution paramétrique elle même, pour agir par des particularisations “on-line”.

A propos de la capacité de manipuler les non-linéarités, quelques travaux ont été réalisés en utilisant la PGD (travaux au sein de l’équipe de Pierre Ladevèze). Quelques travaux existent aussi côté RB pour résoudre des problèmes non linéaires (Safjan, 1988). On peut dire que le travail est en cours pour adapter ces deux premières techniques aux problèmes non linéaires en grande dimension. Cependant, la POD est plus mature que les autres méthodes.

D’autre part, la manipulation du temps reste un peu fastidieuse. Les trois méthodes peuvent employer des techniques incrémentales pour résoudre des problèmes transitoires. Toutefois, la PGD est la seule pour le moment capable d’utiliser des techniques non incrémentales (Ladeveze *et al.*, 2010; Ammar *et al.*, 2010b; Bonithon *et al.*, 2011; Chinesta *et al.*, 2011a).

Un autre critère de comparaison est l’intrusivité, ou la capacité d’introduire la méthode comme une boîte noire dans un code du commerce pour réaliser des simulations. Il est évident que la PGD est la plus intrusive des trois.

Finalement, le tableau 1.1 résume cette comparaison. On utilise les notations ‘+’ pour bon, ‘-’ pour mauvais et ‘?’ pour une réponse non évidente. Ce tableau est tiré de la discussion



finale du workshop “POD, PGD, Reduced basis” organisé à Paris en 2011 ([Chinesta, 2011](#)).

Critère	RB	POD	PGD
Multidimensionnalité (physique)	--	--	++
Multidimensionnalité (paramétrique)	++	++	++
Séparation de l'espace physique	?	+	++
Calcul à priori	Possible	Possible	Oui
Off-line/On-line	+	+	++
Calcul Non linéaire	-	+	-
Caractère Intrusif	-	-	--
Calcul probabiliste	?	++	++
Aspect général	+	++	+
Calcul non incrémental	Non	Non	Possible
Méthode certifiée	++	++	+

tableau 1.1 – *La comparaison des méthodes de réduction de modèle*

1.3 Positionnement du travail

La présentation des problématiques de fabrication et de contrôle de matériaux composites dans la section 1.1 motive une simulation “riche”, vu la complexité de ces matériaux et de leurs procédés de mise en forme. De plus, la simulation 3D de ces matériaux est quasiment impossible avec des méthodes classiques vu la nécessité d’un maillage trop fin ([Bognet et al., 2012](#); [Chinesta et al., 2010c](#)). Ainsi, on motive l’utilisation des méthodes de réduction de modèle. La thèse portera sur le développement de la PGD, ses applications et les résultats obtenus dans des domaines différents. Nous présentons une approche matricielle pour la construction de solutions paramétriques en utilisant cette méthode. Par la suite, une approche originale couplera les solutions paramétriques multidimensionnelles générées à des méthodes d’optimisation pour résoudre “on-line” des problèmes inverses.



2

La PGD, formulation tensorielle et autres formulations spécifiques

2.1	Formulation tensorielle de la PGD	25
2.1.1	Forme classique de la représentation	26
2.1.2	Conditionnement du système et minimisation du résidu	28
2.1.3	Améliorations apportées	30
2.2	Cas particuliers et formulations différentes	32
2.2.1	Contrôle non destructif par thermographie infrarouge	32
2.2.2	DPGD ou “Discontinuous Proper Generalized Decomposition”	39
2.3	Conclusion	43

Dans ce chapitre, une étude détaillée de la PGD est menée. Une formulation tensorielle basique de la PGD sera présentée dans la section 2.1. Cette formulation est efficace pour la résolution des problèmes elliptiques paramétrés dans la plupart des cas. Cependant, il existe toujours des cas particuliers où cette formulation n’est pas applicable ou doit-être adaptée. Nous présentons quelques uns de ces cas-exceptions dans le cadre d’applications avancées dans la section 2.2.

2.1 Formulation tensorielle de la PGD

La PGD a bien évolué durant les dernières années, ainsi que la représentation utilisée. Nous commençons donc par montrer le cas le plus basique de la construction d’une solution par la PGD suivant un algorithme général et efficace. Dans un deuxième temps nous représentons quelques améliorations portées sur cette construction.

2.1.1 Forme classique de la représentation

La PGD sera utilisée pour la résolution de plusieurs problèmes dans cette thèse. Il a fallu trouver, alors, une formulation générale capable de traiter la plupart des problèmes. Pour illustrer cette formulation, considérons le problème modèle suivant :

$$\mathcal{L}U = \mathcal{B}, \quad (2.1)$$

ainsi la forme intégrale des résidus pondérés du problème est écrite :

$$\int_{\Omega} \mathcal{U}^* \mathcal{L}U = \int_{\Omega} \mathcal{U}^* \mathcal{B}, \quad (2.2)$$

où U est une fonction scalaire de $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D)$. Le problème (2.1) est défini donc dans le domaine $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D) \in \Omega =]-L, +L[^l$. \mathcal{L} est un opérateur symétrique défini positif de l'équation aux dérivées partielles (2.1) et \mathcal{B} contient les termes du second membre. Le choix des opérateurs symétriques dans cette section sera justifié ultérieurement. Par simplicité et sans perte de généralité, nous considérons le problème soumis à des conditions de Dirichlet homogènes aux bords. Comme nous avons déjà précisé dans la section 1.2.3, la solution du problème (2.1) s'écrit sous la forme :

$$U(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D) \approx \sum_{i=1}^{i=N} \prod_{j=1}^{j=D} \mathbf{F}_{ij}(\mathbf{x}_j) \quad (2.3)$$

On assume d'abord dans la PGD que les n premiers produits de fonctions \mathbf{F}_{ij} (n premiers "modes") sont connus et on cherche à enrichir par un nouveau mode. Ainsi la fonction U_{n+1} s'écrit comme étant :

$$U_{n+1} = \sum_{i=1}^{i=n} \prod_{j=1}^{j=D} \mathbf{F}_{ij}(\mathbf{x}_j) + \prod_{j=1}^{j=D} \mathbf{R}_j(\mathbf{x}_j) \quad (2.4)$$

De plus, on écrit la fonction test comme suit :

$$U^* = \mathbf{R}_1^* \otimes \mathbf{R}_2 \otimes \dots \otimes \mathbf{R}_D + \dots + \mathbf{R}_1 \otimes \mathbf{R}_2 \otimes \dots \otimes \mathbf{R}_D^* \quad (2.5)$$

Afin de rester dans un cadre, supposons que la projection des opérateurs \mathcal{L} et de \mathcal{B} sur les maillages des différentes coordonnées peuvent s'écrire aussi sous formes séparées, notons \mathcal{A}



et \mathcal{B} , donc :

$$\begin{cases} \mathcal{A} = \sum_{i=1}^{i=n_A} \mathbf{A}_1^i \otimes \mathbf{A}_2^i \otimes \cdots \otimes \mathbf{A}_D^i \\ \mathcal{B} = \sum_{i=1}^{i=n_B} \mathbf{B}_1^i \otimes \mathbf{B}_2^i \otimes \cdots \otimes \mathbf{B}_D^i \end{cases} \quad (2.6)$$

Dans l'équation (2.6), n_A représente le nombre d'opérateurs qui existent dans l'équation aux dérivées partielles, n_B est le nombre de modes qui constituent le second membre. La construction de ces tenseurs provient directement de la formulation faible du problème à résoudre. Plus de détails sur cette construction sont représentés dans l'annexe A. Un tel choix de la représentation peut-être justifié et reste valable dans la plupart des cas (Ammar *et al.*, 2006, 2007). Cependant, il reste toujours des exceptions où nous n'arriverions pas à utiliser cette forme (voir la section 2.2).

Ensuite, on emploie un algorithme de point fixe et directions alternées afin de résoudre le problème. Ainsi, considérons qu'on cherche la fonction R_j . On peut écrire la fonction test sous la forme suivante :

$$\mathcal{U}^* = \mathbf{R}_1 \otimes \mathbf{R}_2 \otimes \cdots \otimes \mathbf{R}_j^* \otimes \cdots \otimes \mathbf{R}_D \quad (2.7)$$

On remplace les notations introduites par les équations (2.4), (2.6) et (2.7) dans l'équation (2.2). Par suite, l'intégrale de chaque terme est calculée suivant l'opérateur correspondant. On obtient ainsi la formulation discrète du problème :

$$\begin{aligned} & \sum_{i=1}^{i=n_A} \sum_{k=1}^{k=n} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{F}_{k1} \times \cdots \times (\mathbf{R}_j^*)^T \mathbf{A}_j^i \mathbf{F}_{kj} \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{F}_{kD} + \\ & + \sum_{i=1}^{i=n_A} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times (\mathbf{R}_j^*)^T \mathbf{A}_j^i \mathbf{R}_j \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{R}_D = \\ & = \sum_{i=1}^{i=n_B} (\mathbf{R}_1)^T \mathbf{B}_1^i \times \cdots \times (\mathbf{R}_j^*)^T \mathbf{B}_j^i \times \cdots \times (\mathbf{R}_D)^T \mathbf{B}_D^i \end{aligned} \quad (2.8)$$

Pour simplifier le reste des notations, on introduit la forme suivante :

$$\sum_{i=1}^{i=n_C} \prod_{j=1}^{j=D} \mathbf{C}_j^i = \sum_{i=1}^{i=n_B} \prod_{j=1}^{j=D} \mathbf{B}_j^i + \sum_{i=1}^{i=n_A} \sum_{k=1}^{k=n} \prod_{j=1}^{j=D} \mathbf{A}_j^i \mathbf{F}_{kj} \quad (2.9)$$

■

En remplaçant la notation (2.9) dans l'équation (2.8) on obtient :

$$\begin{aligned} \sum_{i=1}^{i=n_A} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times (\mathbf{R}_j^*)^T \mathbf{A}_j^i \mathbf{R}_j \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{R}_D = \\ = \sum_{i=1}^{i=n_C} (\mathbf{R}_1)^T \mathbf{C}_1^i \times \cdots \times (\mathbf{R}_j^*)^T \mathbf{C}_j^i \times \cdots \times (\mathbf{R}_D)^T \mathbf{C}_D^i \end{aligned} \quad (2.10)$$

En supposant que l'équation (2.10) reste valable pour tout \mathbf{R}_j^* on pourra écrire :

$$\left(\sum_{i=1}^{i=n_A} \left(\prod_{k=1; k \neq j}^{k=D} \mathbf{R}_k^T \mathbf{A}_k^i \mathbf{R}_k \right) \mathbf{A}_j^i \right) \mathbf{R}_j = \sum_{i=1}^{i=n_C} \left(\prod_{k=1; k \neq j}^{k=D} \mathbf{R}_k^T \mathbf{C}_k^i \right) \mathbf{C}_j^i, \quad (2.11)$$

Ce qui donne un système de même dimension que \mathbf{R}_j à résoudre. Cette construction de la PGD est basée sur la formulation de Galerkin. L'expérience montre que c'est un algorithme efficace dans le cas des problèmes elliptiques.

2.1.2 Conditionnement du système et minimisation du résidu

Dans la section 2.1.1 une hypothèse de symétrie sur la forme de l'opérateur \mathcal{L} a été utilisée. Dans le cas des opérateurs non symétriques, aucune convergence vers la solution réelle n'est assurée. Toutefois, une autre possibilité de formuler la PGD existe et ceci par la minimisation de la norme du résidu. Cette méthode peut-être interprétée comme un moyen de préconditionnement. De ce point de vue, la minimisation déjà mentionnée devient cohérente avec le préconditionnement des algorithmes de descente de gradient ($A^T A x = A^T b$) (Glowinski, 1980; der Vorst et Dekker, 1988). En effet, en utilisant les notations déjà mentionnées dans la section 2.1.1, on pourra écrire le résidu de l'équation (2.1) comme étant :

$$Res = \sum_{i=1}^{i=n_A} \mathbf{A}_1^i \mathbf{R}_1 \otimes \cdots \otimes \mathbf{A}_D^i \mathbf{R}_D - \sum_{i=1}^{i=n_C} \mathbf{C}_1^i \otimes \cdots \otimes \mathbf{C}_D^i \quad (2.12)$$

Par suite la norme du résidu sera notée comme :

$$\begin{aligned} \|Res\|^2 = \sum_{i=1}^{i=n_A} \sum_{j=1}^{j=n_A} (\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{A}_1^j \mathbf{R}_1 \rangle \times \cdots \times \langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{A}_D^j \mathbf{R}_D \rangle) - \\ - 2 \sum_{i=1}^{i=n_A} \sum_{j=1}^{j=n_C} (\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{C}_1^j \rangle \times \cdots \times \langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{C}_D^j \rangle) + \\ + \sum_{i=1}^{i=n_C} \sum_{j=1}^{j=n_C} (\langle \mathbf{C}_1^i, \mathbf{C}_1^j \rangle \times \cdots \times \langle \mathbf{C}_D^i, \mathbf{C}_D^j \rangle) \end{aligned} \quad (2.13)$$

Où $\langle \cdot, \cdot \rangle$ est un produit scalaire et $\|\cdot\|$ est la norme associée. La fonction coût à minimiser



pour calculer la fonction \mathbf{R}_j s'écrit donc :

$$\frac{\partial}{\partial \mathbf{R}_j} \langle Res, Res \rangle = 0 \quad (2.14)$$

Si on note Res_i comme étant le résidu après l'enrichissement par le $i^{\text{ème}}$ mode, un critère de convergence acceptable peut-être noté :

$$\frac{Res_i}{Res_1} < \epsilon \quad (2.15)$$

Par suite la résolution du système peut être réalisée de la même façon que décrite dans la section 2.1.1. L'équation (2.15) introduit un critère de convergence qui se base sur la réduction du résidu de l'équation. D'autres critères de convergences plus fiables, qui se basent sur les quantités d'intérêts calculées "à postériori", existent aussi (Ammar *et al.*, 2010a). D'ailleurs, durant cette thèse nous avons adapté l'estimateur d'erreur de Zienkiewicz et Zhu (1987) a la PGD. Plus de détails sont présentés dans l'annexe B.

L'importance de la minimisation du résidu et du conditionnement du système réside dans la convergence monotone de la norme du résidu. Cela est donc utilisé lorsque la formulation de Galerkin converge trop lentement ou ne converge pas, et notamment lorsqu'on a des opérateurs non symétriques dans la formulation variationnelle. En revanche, cette méthode de résolution engendre quelques inconvénients. Principalement, la convergence devient beaucoup plus lente en comparant avec la convergence, quand elle existe, de l'approche décrite dans la section 2.1.1. Le tableau 2.1 compare la convergence des deux approches de calcul PGD, selon le critère donné par l'équation (2.15) pour un problème de Poisson 3D stationnaire donné par l'équation (2.16) en séparant (x,y) et z . D'autre part, le temps de calcul d'une itération en utilisant la minimisation du résidu est plus important que le temps d'une itération utilisant la formulation de Galerkin classique. Pour le même problème formulé ci-dessous par l'équation (2.16) le temps de calcul de 20 modes en inversion de Galerkin est de 6.2 secondes, alors qu'avec la minimisation du résidu il atteint 7.2 secondes. Dans des problèmes plus compliqués la différence est beaucoup plus flagrante.

Problème de comparaison :

Soit $X \in \Omega$ tell que :

$$\begin{cases} \Delta U = 50 \\ U = 0 \text{ sur } \partial\Omega \end{cases} \quad (2.16)$$

■

Avec Ω montré dans les figures 2.1 et 2.2 comme étant un maillage sur $(x,y) \times$ un maillage sur z . La solution est affiché par la figure 2.3.

Après l'ajout de	Galerkin	Minimisation du résidu
5 modes	0.0075239	0.0081543
10 modes	1.4295×10^{-6}	0.0011552
15 modes	7.8432×10^{-8}	0.00018127
20 modes	$1.9622e \times 10^{-8}$	5.6807×10^{-5}

tableau 2.1 – Les valeurs de $\frac{Res_i}{Res_1}$ pour les deux méthodes de calcul PGD

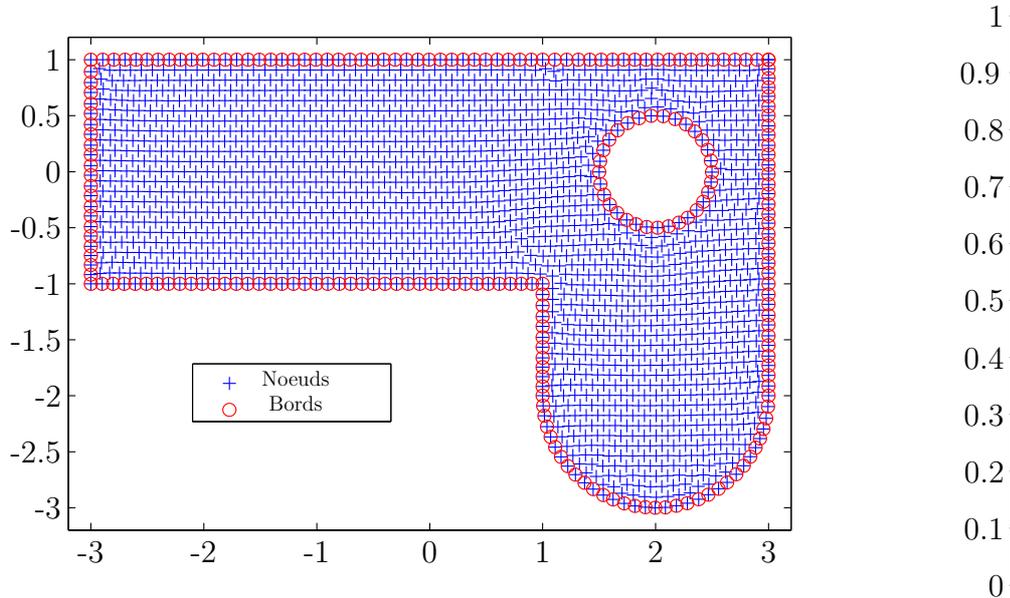


figure 2.1 – Les nœuds du domaine Ω en (x,y) , 1735 nœuds

figure 2.2 – Maillage dans l'épaisseur, 100 nœuds

2.1.3 Améliorations apportées

Deux améliorations similaires, mais différentes, ont été apportées à cet algorithme. Ces deux améliorations reposent principalement sur le principe de reconvergence des modes après avoir ajouté de nouveaux modes, donc de nouvelles informations. Ecrivons d'abord la solution sous la forme suivante :

$$\mathcal{U}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D) = \sum_{k=1}^{k=N} \alpha_k \prod_{j=1}^{j=D} \mathbf{F}_{kj}(\mathbf{x}_j), \quad (2.17)$$

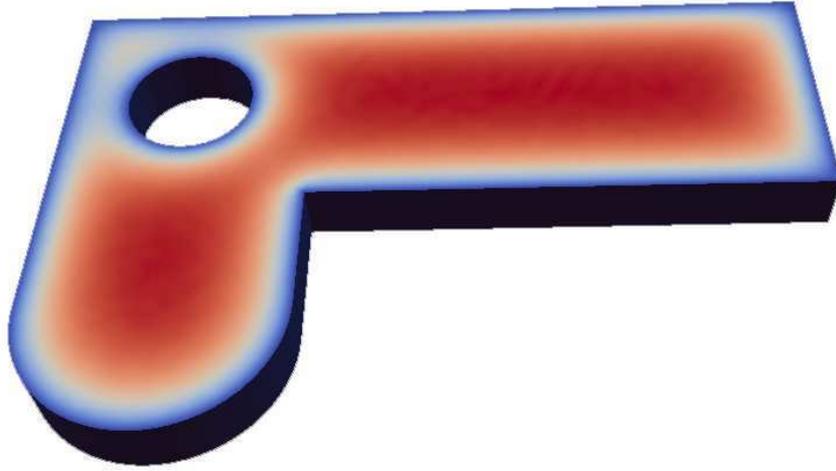


figure 2.3 – Coupe de la solution 3D du problème-exemple

où les α_k sont les “poids” des modes définis par le produit des \mathbf{F}_{kj} , qui sont maintenant normalisés. Une étape supplémentaire vient s’intercaler entre deux enrichissements successifs, et correspond à la reconvergence des α_i . La fonction test s’écrit maintenant :

$$\mathcal{U}^*(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D) = \sum_{k=1}^{k=N} \alpha_k^* \prod_{j=1}^{j=D} \mathbf{F}_{kj}(\mathbf{x}_j), \quad (2.18)$$

En remplaçant dans la formulation faible de l’équation (2.1) et en suivant la même démarche que dans la section 2.1.1 on obtient un système à résoudre :

$$\mathcal{K}\alpha = \mathcal{F}, \quad (2.19)$$

Avec α le vecteur des α_k à calculer. \mathcal{K} et \mathcal{F} définit comme suit :

$$\begin{cases} \mathcal{K}_{kl} = \sum_{i=1}^{i=n_A} \prod_{j=1}^{j=D} \mathbf{F}_{kj}^T \mathbf{A}_j^i \mathbf{F}_{lj} \\ \mathcal{F}_k = \sum_{i=1}^{i=n_B} \prod_{j=1}^{j=D} \mathbf{F}_{kj}^T \mathbf{B}_j^i \end{cases} \quad (2.20)$$

Ainsi on pourra déterminer une nouvelle pondération des modes calculées, ce qui améliore la convergence de l’algorithme dans quelques cas.

D’autre part, la deuxième amélioration consiste à recalculer jusqu’à convergence des modes déjà calculés. Nous avons appelé cette technique la “mise à jour” des modes calculés. Ceci



améliore le nombre de modes nécessaire à la convergence de la solution considérablement, surtout pour un problème de très haute dimensionnalité. Par exemple, pour un problème paramétrique à dimension 18, on a besoin de calculer plus de 1600 modes dans un calcul classique sans “retour en arrière” ou “mise à jour” des anciens modes, cependant la solution était toujours bruitée. En revanche, quand on utilise la “mise à jour”, la solution obtenue est meilleure après avoir calculé 137 modes seulement. Cependant le temps de calcul n’est pas amélioré de façon considérable. Pour plus de détails sur cet exemple se référer à la section 5.2.2 de cette thèse.

La figure 2.4 illustre l’algorithme final utilisé durant le calcul PGD. En raison de simplification des notations, on va omettre la pondération α_k des notations dans les problèmes suivants. Cependant, sauf précision différente, ce format de pondération a été utilisé dans tout le reste de la thèse.

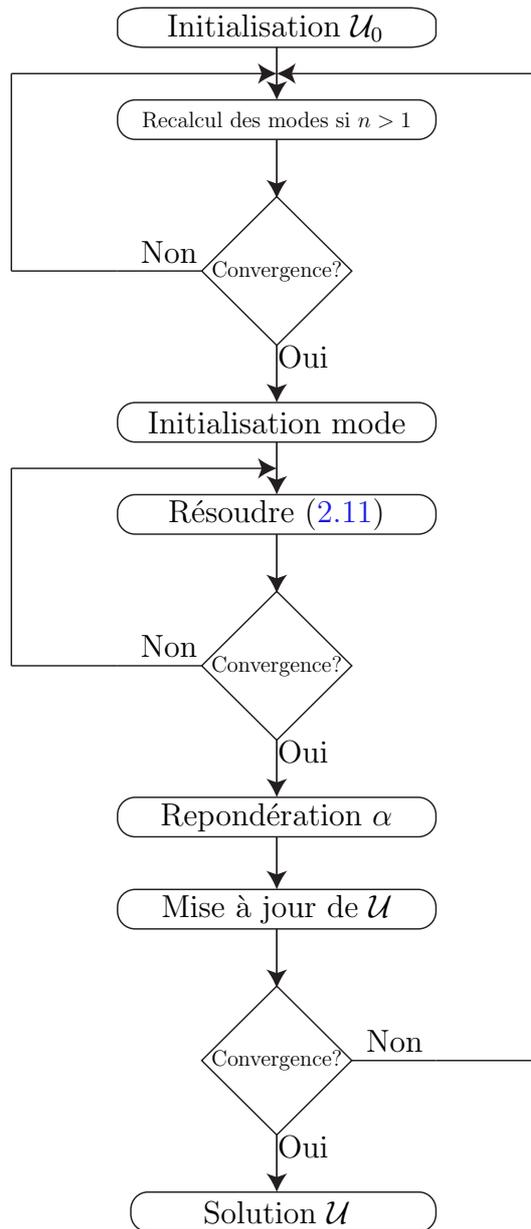
2.2 Cas particuliers et formulations différentes

Dans la section 2.1.1 une hypothèse a été émise sur la séparabilité de l’opérateur \mathcal{L} , d’où la forme séparée de \mathcal{A} donnée par l’équation (2.6). En revanche, nous ne pouvons pas toujours trouver une forme séparée des opérateurs. Cependant, on pourra démontrer que la forme séparée du tenseur \mathcal{B} existe toujours (Messiter et Shamash, 1985). Le cas où la formulation déjà présentée dans la section 2.1 n’est pas applicable sera illustré dans une application de contrôle non destructif et par une introduction aux résolutions dans des domaines contenant des fissures. La méthode de calcul dans les domaines englobant des fissures sera nommée la DPGD ou “Discontinuous Proper Generalized Decomposition”. Dans ce dernier cas le modèle est séparable mais engendre des calculs d’intégrales sur différents maillages.

2.2.1 Contrôle non destructif par thermographie infrarouge

Dans cette section nous présentons l’aspect numérique du contrôle non destructif, en utilisant la PGD. En effet, après la prise d’une image thermique du matériau, la recherche d’une solution numérique identique est entamée. Avec des méthodes classiques, ce problème est traité en changeant la position du défaut à chaque itération du calcul. Cependant en utilisant la PGD nous résolvons le problème en modélisant la position du défaut comme une coordonnée supplémentaire du problème. Ainsi, une solution générale est calculée une fois par pièce. Par suite le contrôle non destructif devient un problème d’identification, rapide à résoudre.



figure 2.4 – Algorithme de calcul PGD, n étant le nombre de modes calculés

Exemple numérique

Considérons d'abord l'équation de la chaleur suivante :

$$-\nabla(\mathbf{K}\nabla U) = Q \quad (2.21)$$

Où \mathbf{K} est la conductivité, U la température et Q le terme source. Par soucis de simplicité et sans perte de généralité, considérons Q uniforme. Le problème a été résolu sur une pièce 1D et 2D. Pour simplifier les notations, on considère les équations dans un domaine 1D, $\Omega = [0; L]$. Le calcul est donc réalisé sur la pièce qui constitue un domaine Ω , avec des conditions limites homogènes. Le domaine contient un défaut caractérisé par une conductivité différente du reste de la pièce. Notons la position du défaut y . Par suite, on cherche la solution U sous la forme :

$$U(x,y) \approx \sum_{i=1}^{i=N} \alpha_i \mathbf{X}_i(x) \times \mathbf{Y}_i(y) \quad (2.22)$$

La formulation du problème en 1D, pour $\Omega_x = [0,L] \times \Omega_y = [0,L]$ devient :

$$\begin{cases} \frac{\partial}{\partial x} (\mathbf{K} \frac{\partial U}{\partial x}) = -Q; \\ U(0,y) = 0; \\ U(L,y) = 0; \\ \mathbf{K} = K_1 - K_2 \times \xi(x,y) \end{cases} \quad (2.23)$$

Avec K_1 la conductivité de la pièce, K_2 la différence de conductivité entre la pièce et le défaut. Notons l la largeur du défaut à détecter. Cette largeur peut être aussi modélisée comme étant coordonnée supplémentaire. On obtient donc $\xi(x,y)$ définit par :

$$\begin{cases} \xi(x,y) = 1 \text{ pour } x \in [y; y + l] \\ \xi(x,y) = 0 \text{ pour } x \in [0; y[\cup]y + l; L] \end{cases} \quad (2.24)$$

A noter que dans ce cas, \mathbf{K} est défini par un échelon dans le domaine (x,y) , par suite très difficilement séparable en produits de fonctions. Donc la formulation définie dans la section 2.1 est inapplicable. On revient ainsi à la section 1.2.3 pour réaliser le traitement du problème. La fonction test s'écrit donc comme étant :

$$U^*(x,y) = \mathbf{r}^*(x) \cdot \mathbf{s}(y) + \mathbf{r}(x) \cdot \mathbf{s}^*(y) \quad (2.25)$$



La formulation intégrale du problème (2.23) s'écrit donc :

$$\int_{\Omega} U^* \cdot \left(-\frac{\partial}{\partial x} \left(\mathbf{K} \frac{\partial U}{\partial x} \right) - Q \right) dx dy \quad (2.26)$$

Par suite en remplaçant la fonction test de l'équation (2.25), dans la formulation (2.26) du problème, et en suivant la même démarche décrite dans la partie 1.2.3, on obtient l'algorithme suivant :

1. On suppose que $\mathbf{s}(y)$ est connue et donc $\mathbf{s}^* = 0$. De plus on pose :

$$\begin{cases} \alpha(x) = \int_{\Omega_y} \mathbf{K} \cdot \mathbf{s} \cdot dy \\ \gamma(x) = \sum_{i=1}^{i=n} \left(\frac{\partial^2 \mathbf{X}_i}{\partial x^2} \int_{\Omega_y} (\mathbf{K} \cdot \mathbf{s} \cdot \mathbf{Y}_i) dy \right) \\ q_1 = \int_{\Omega_y} \mathbf{s} \cdot Q \cdot dy \end{cases} \quad (2.27)$$

En remplaçant ces notations dans l'équation (2.26), on obtient une première équation différentielle à résoudre. La formulation forte s'écrit sous la forme suivante :

$$-\alpha \frac{\partial^2 \mathbf{r}}{\partial x^2} = q_1 + \gamma \quad (2.28)$$

A noter que les termes α et γ sont des fonctions de x , par suite l'intégration pour le calcul de ces termes doit-être faite pour point $x \in \Omega_x$. Ainsi le temps de formulation de la PGD pour ce problème sera augmenté par le besoin de calculer plusieurs intégrales.

2. On utilise la valeur de \mathbf{r} retrouvée par la résolution de l'équation (2.28) afin de mettre à jour \mathbf{s} . Maintenant \mathbf{r} est connu et donc $\mathbf{r}^* = 0$. On note :

$$\begin{cases} \beta(y) = \int_{\Omega_x} \mathbf{K} \cdot \mathbf{r} \cdot \frac{\partial^2 \mathbf{r}}{\partial x^2} \cdot dx \\ \delta(y) = \sum_{i=1}^{i=n} \left(\mathbf{Y}_i \int_{\Omega_x} \left(\mathbf{K} \cdot \mathbf{r} \cdot \frac{\partial^2 \mathbf{X}_i}{\partial x^2} \right) dx \right) \\ q_2 = \int_{\Omega_x} \mathbf{r} \cdot Q \cdot dx \end{cases} \quad (2.29)$$

Finalement, on obtient une deuxième formulation forte à résoudre donnée par :

$$-\beta \cdot \mathbf{s} = \delta + q_2 \quad (2.30)$$



Même remarque que précédemment, les termes β et δ doivent être calculés pour chaque point $y \in \Omega_y$.

La figure 2.5 montre la solution PGD dans un domaine Ω unidimensionnel, alors que la figure 2.6 montre cette solution calculée par la méthode des éléments finis. L'erreur relative entre les deux solutions est montrée dans la figure 2.7, l'erreur relative maximale est inférieure à 1%. La solution PGD montrée a été obtenue après le calcul de 97 modes. Les paramètres de la résolution de ce problème sont donnés par le tableau 2.2. Les valeurs sont adimensionnelles. A noter que dans ce cas on utilise une formulation sans repondération ou remise à jour des modes.

Remarque: L'utilisation d'un échelon dans la fonction $\xi(x,y)$ rend le calcul difficile à converger. En effet, le fait de lisser l'échelon par une fonction de Gauss réduit le nombre de modes nécessaires à la convergence de 97 à 54. $\xi(x,y)$ est écrit donc :

$$\begin{cases} \xi(x,y) = 5 - 4e^{-\frac{(x-y)^2}{0.01}} & \text{pour } x \in [y; y+l] \\ \xi(x,y) = 0 & \text{pour } x \in [0; y[\cup]y+l; L] \end{cases} \quad (2.31)$$

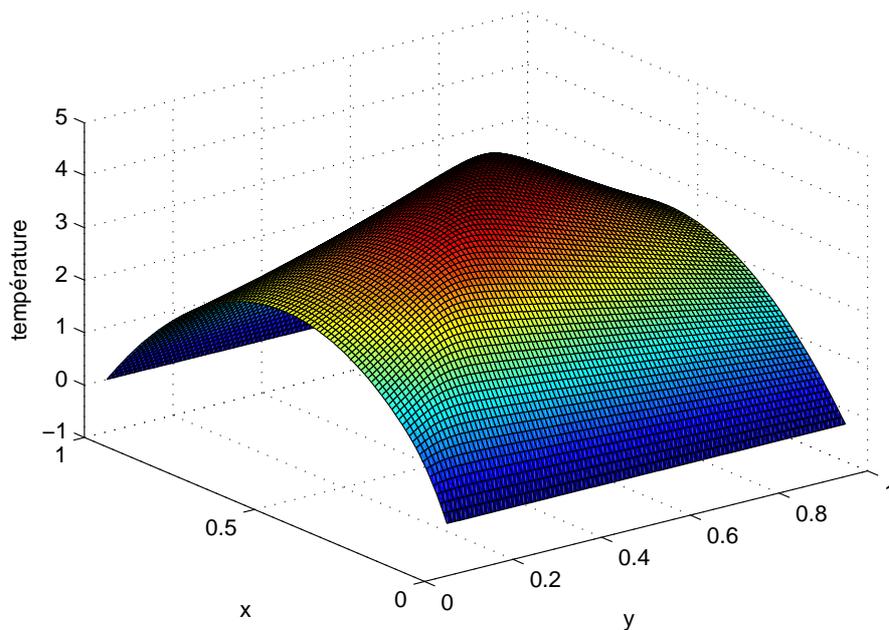


figure 2.5 – Solution PGD dans un domaine 1D avec la position du défaut en coordonnée

Paramètre	Valeur
Q	100
L	1
\mathbf{K}_1	5
\mathbf{K}_2	4

tableau 2.2 – Les valeurs utilisées lors de la résolution du problème posé

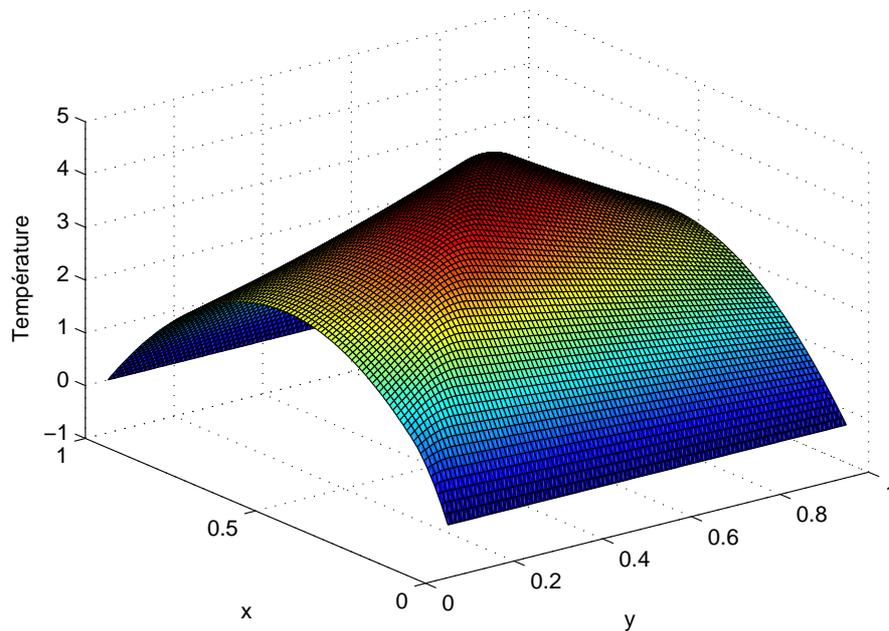


figure 2.6 – Solution éléments finis pour toute position du défaut

Par la même démarche, le problème 2D a été résolu pour le même domaine montré dans la figure 2.1. Ainsi, une combinaison donnée des coordonnées de la position du défaut donne la solution montrée par la figure 2.8. Les mêmes valeurs des paramètres du tableau 2.2 ont été utilisées dans le problème 2D.

Calcul inverse de la position du défaut

Afin de réaliser le traitement des données et déduire la position du défaut, on utilise une méthode de calcul inverse. Nous avons utilisé une fonction coût \mathcal{J} basée sur les moindres



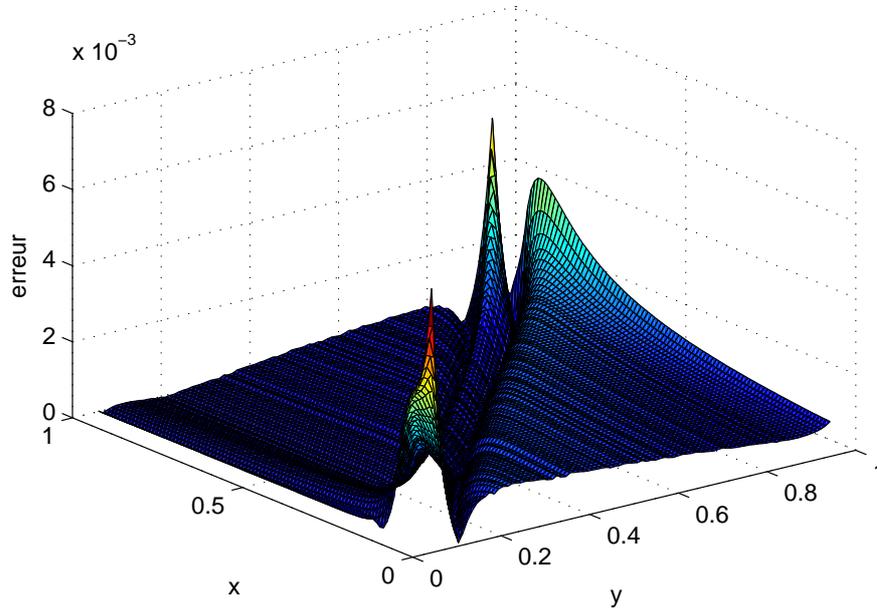


figure 2.7 – Erreur relative entre la solution éléments finis et la solution PGD

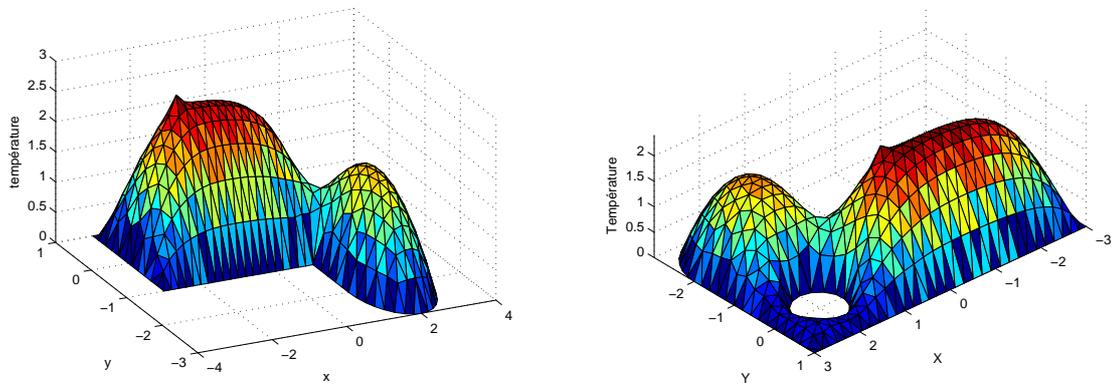


figure 2.8 – Solution PGD en 2D pour deux positions possibles du défaut

carrées donnée par :

$$\begin{cases} y_{opt} = \operatorname{argmin} \mathcal{J}(y) \\ \mathcal{J}(y) = \int_{\Omega} (U(x,y) - U^{mes}(x))^2 dx \end{cases} \quad (2.32)$$

La minimisation a été réalisée par la méthode dite “*SIMPLEX Method*” préintégré dans MATLAB. Cette méthode est efficace, mais lente par rapport aux algorithmes à descente de gradient. En effet, cette méthode calcule les gradients numériquement ce qui augmente le

temps de calcul (Lagarias *et al.*, 1998; Nocedal et Wright, 1999). D'autres méthodes d'optimisation seront présentées ultérieurement dans la thèse. Le résultat d'un calcul inverse est donné par la figure 2.9. Le défaut fait la taille de dix nœuds du maillage. La position du défaut détecté coïncide en moyenne avec la vraie position, même si la solution diffère légèrement. Cette différence remarquée dans la solution est due à la convergence de la PGD.

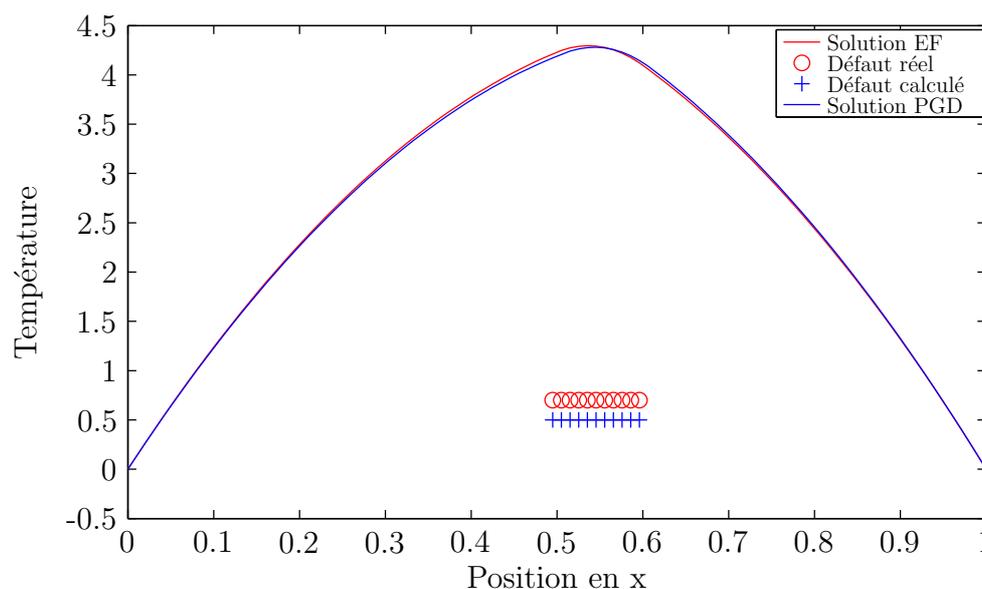


figure 2.9 – Détection d'un défaut par la méthode proposée

2.2.2 DPGD ou “Discontinuous Proper Generalized Decomposition”

La discontinuité dans la DPGD fait référence aux calculs de domaines englobant une discontinuité de la solution, comme une fissure par exemple. De nos jours, les fissures sont traitées efficacement par la méthode connue sous le nom de X-FEM ou méthode des éléments finis étendus (Dreau *et al.*, 2010). Cette méthode utilise un enrichissement de la méthode des éléments finis au niveau des éléments traversés par les fissures. L'enrichissement est réalisé par une fonction échelon ou “Heaviside”. En effet, cet enrichissement découple les propriétés des deux parties de l'élément (Moes *et al.*, 1999). Les fonctions de forme par élément fissuré s'écrivent donc sous la forme :

$$\mathcal{U} = u_1 N_1 + u_2 N_2 + a_1 N_1 H(x) + a_2 N_2 H(x) \quad (2.33)$$

\mathcal{U} étant la variable recherchée, u_1 et u_2 les valeurs aux nœuds 1 et 2, N_1 et N_2 les fonctions de forme des éléments finis classiques. Les termes ajoutés sont $H(x)$ qui est la Heaviside



exprimant la discontinuité des champs calculés au niveau de la fissure, a_1 et a_2 donnent l'amplitude de la discontinuité. A noter que nous devons enrichir par deux Heavysides puisqu'on a une discontinuité dans la valeur du champ calculé, mais aussi une discontinuité au niveau des gradients (Moes, 2011).

Dans la PGD, on s'inspire de la notation (2.33) pour calculer le champ de température autour de la fissure. Par contre, dans le cas de la PGD, nous n'enrichissons pas un élément particulier, mais on ajoute un mode discontinu au début du calcul, qui définira la position de la fissure. Pour illustrer la démarche, considérons l'équation de Poisson suivante :

$$\Delta \mathcal{U} = f, \quad (2.34)$$

dans un domaine bidimensionnel $\Omega = [-L, L]^2$, avec la fissure étant situé à $y = 0$ et $x \in [0, L]$. Les conditions limites sont les suivantes :

$$\begin{cases} \mathcal{U}(-L, y) = 0 \\ \mathcal{U}(L, y) = G(y) \\ \frac{\partial \mathcal{U}}{\partial y}(x, -L) = \frac{\partial \mathcal{U}}{\partial y}(x, L) = 0 \end{cases} \quad (2.35)$$

De plus, on considère un flux normal nul au niveau de la fissure. Ainsi, aucun échange thermique n'est possible d'une part et d'autre de la fissure. D'autre part, $G(y)$ est définie comme étant :

$$\begin{cases} G(y) = 0 \text{ si } y \in [-L, 0] \\ G(y) = 1 \text{ si } y \in]0, L] \end{cases} \quad (2.36)$$

Pour résoudre ce problème en utilisant la PGD, on commence par considérer un premier mode connu, qui satisfait les conditions limites. Le premier mode définit aussi la position de la discontinuité de température. Dans ce cas, nous cherchons donc la solution \mathcal{U} , en séparant les coordonnées de l'espace Ω , comme étant :

$$\mathcal{U} = X_0(x) \cdot Y_0(y) + \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \quad (2.37)$$

Avec $X_0(x) \cdot Y_0(y)$ le mode qui exprime la position de la fissure. Pour l'exemple présenté, on



pourra utiliser :

$$\begin{cases} X_0(x) = 0 \text{ si } x \leq 0 \\ X_0(x) = x \text{ si } x > 0 \\ Y_0(y) = G(y) \end{cases} \quad (2.38)$$

Afin de bien modéliser la discontinuité au niveau de la fissure, Y_0 n'est pas mise à jour durant le calcul PGD. Cependant, il faudra calculer les dérivées de Y_0 indépendamment de part et d'autre de la fissure. En effet, il faudra s'assurer de bien conserver la discontinuité des dérivées. Pour bien calculer l'amplitude de la discontinuité, il faudra mettre à jour le vecteur $X_0(x)$ lors de chaque itération. Ainsi, quand on cherche à enrichir par un mode de plus, la fonction test devient :

$$\mathcal{U}^* = R(x) \cdot S^*(y) + R^*(x) \cdot S(y) + X_0^*(x) \cdot Y_0(y), \quad (2.39)$$

alors, la fonction \mathcal{U} s'écrit :

$$\mathcal{U}_{i+1} = X_0(x) \cdot Y_0(y) + \sum_{i=1}^{i=n} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y) \quad (2.40)$$

A tour de rôle, on commence par considérer R et X_0 connues pour calculer S , puis S et X_0 connues pour calculer R , et enfin R et S connues pour calculer X_0 . Considérons le premier cas où on cherche à calculer S . La fonction test devient :

$$\mathcal{U}^* = R(x) \cdot S^*(y) \quad (2.41)$$

En remplaçant dans l'équation différentielle (2.34) on obtient la formulation intégrale suivante :

$$\begin{aligned} \int_{\Omega} S^* \cdot R \left(S \cdot \frac{\partial^2 R}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \cdot R \right) d\Omega = \\ = - \int_{\Omega} S^* \cdot R \left(\sum_{i=1}^{i=n} X_i \cdot \frac{\partial^2 Y_i}{\partial y^2} + \sum_{i=1}^{i=n} \frac{\partial^2 X_i}{\partial x^2} \cdot Y_i - f \right) d\Omega - \int_{\Omega} S^* \cdot R \cdot X_0 \cdot Y_0 d\Omega \end{aligned} \quad (2.42)$$

Si nous considérons maintenant S connue et nous cherchons R , la fonction test s'écrit comme étant :

$$\mathcal{U}^* = R^*(x) \cdot S(y) \quad (2.43)$$

■

Par suite, la formulation intégrale du problème devient :

$$\begin{aligned} \int_{\Omega} S \cdot R^* \left(S \cdot \frac{\partial^2 R}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \cdot R \right) d\Omega = \\ = - \int_{\Omega} S \cdot R^* \left(\sum_{i=1}^{i=n} X_i \cdot \frac{\partial^2 Y_i}{\partial y^2} + \sum_{i=1}^{i=n} \frac{\partial^2 X_i}{\partial x^2} \cdot Y_i - f \right) d\Omega - \int_{\Omega} S \cdot R^* \cdot X_0 \cdot Y_0 d\Omega \end{aligned} \quad (2.44)$$

Finalement, en mettant X_0 à jour la fonction test s'écrit :

$$\mathcal{U}^* = X_0^*(x) \cdot Y_0(y), \quad (2.45)$$

donc la formulation intégrale du problème devient :

$$\begin{aligned} \int_{\Omega} X_0^* \cdot Y_0 \cdot X_0 \cdot Y_0 d\Omega = \int_{\Omega} X_0^* \cdot Y_0 \left(S \cdot \frac{\partial^2 R}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \cdot R \right) d\Omega - \\ - \int_{\Omega} X_0^* \cdot Y_0 \left(\sum_{i=1}^{i=n} X_i \cdot \frac{\partial^2 Y_i}{\partial y^2} + \sum_{i=1}^{i=n} \frac{\partial^2 X_i}{\partial x^2} \cdot Y_i - f \right) d\Omega \end{aligned} \quad (2.46)$$

La résolution des équations (2.42), (2.44) et (2.46) se fait comme dans la partie 1.2.3, en tenant compte de la discontinuité de Y_0 . Cependant, si on voulait passer à une formulation tensorielle, il faudrait calculer les opérateurs d'intégration des domaines continus et discontinus, ce qui multiplie le nombre d'opérateurs à construire par trois. En effet, voilà quelques exemples d'intégrations numériques :

$$\begin{cases} \int_y Y_i \cdot S = Y_i^T \mathbf{M}_{cc} S \\ \int_y Y_i \cdot Y_0 = Y_i^T \mathbf{M}_{cd} Y_0 \\ \int_y Y_0 \cdot Y_0 = Y_0^T \mathbf{M}_{dd} Y_0 \end{cases} \quad (2.47)$$

La matrice \mathbf{M} étant la matrice de masse, les lettres c et d signifient respectivement *continue* et *discontinue*. A noter que $\mathbf{M}_{cd} = \mathbf{M}_{dc}^T$ et que la matrice \mathbf{M}_{dd} apparaît lors de la mise à jour du terme $X_0(x)$. La construction des matrices continues et discontinues se fait par un propre assemblage des matrices des domaines situés de part et d'autre de la fissure. Les matrices *continue/continue* sont les matrices classiques des éléments finis. Pour une matrice *discontinue/discontinue*, si a_{ij} et b_i étant les termes de la matrice de masse de part et d'autre



de la discontinuité, on peut à titre d'exemple, écrire :

$$\mathbf{M}_{dd} = \begin{pmatrix} a_{11} & \cdots & a_{1p} & & & \\ \vdots & & \vdots & & & 0 \\ a_{o1} & \cdots & a_{op} & & & \\ & & & b_{11} & \cdots & b_{1r} \\ & 0 & & \vdots & & \vdots \\ & & & b_{q1} & \cdots & b_{qr} \end{pmatrix} \quad (2.48)$$

Alors que pour une matrice *continue/discontinue* on peut écrire :

$$\mathbf{M}_{cd} = \begin{pmatrix} a_{11} & \cdots & a_{1p} & & & \\ \vdots & & \vdots & & & 0 \\ a_{o1} & \cdots & a_{op} & b_{11} & \cdots & b_{1r} \\ & & & \vdots & & \vdots \\ & 0 & & b_{q1} & \cdots & b_{qr} \end{pmatrix} \quad (2.49)$$

En transposant la matrice \mathbf{M}_{cd} on pourra obtenir une matrice \mathbf{M}_{dc} . La figure 2.10 montre la solution du problème (2.34) posé dans cette section. La solution est donnée par la DPGD pour $L = 1$ et $f = 0$. La figure 2.11 montre la solution du même problème pour $L = 1$ et $f = 10$.

A noter que X_0 n'apparaît pas dans un produit avec R ni avec S . Nous pouvons alors envisager la mise à jour de X_0 avec R ou S en même temps, ou avec les deux. Cette approche n'est pas testée mais pourrait aboutir à une convergence plus rapide.

Pour l'instant, l'algorithme utilisé sépare les coordonnées spatiales x et y , ce qui est simple dans le cas d'une fissure suivant l'une des directions des axes du repère utilisé. Cependant, dans le cas d'une fissure oblique, il vaut mieux ne pas séparer les coordonnées spatiales. L'implémentation de la X-FEM dans le cadre de la PGD est possible (Niroomandi *et al.*, 2012).

2.3 Conclusion

La forme tensorielle de la PGD est une forme efficace pour l'implémentation de la PGD. Durant la suite de cette thèse, la PGD sera utilisée pour adresser la solution à plusieurs pro-



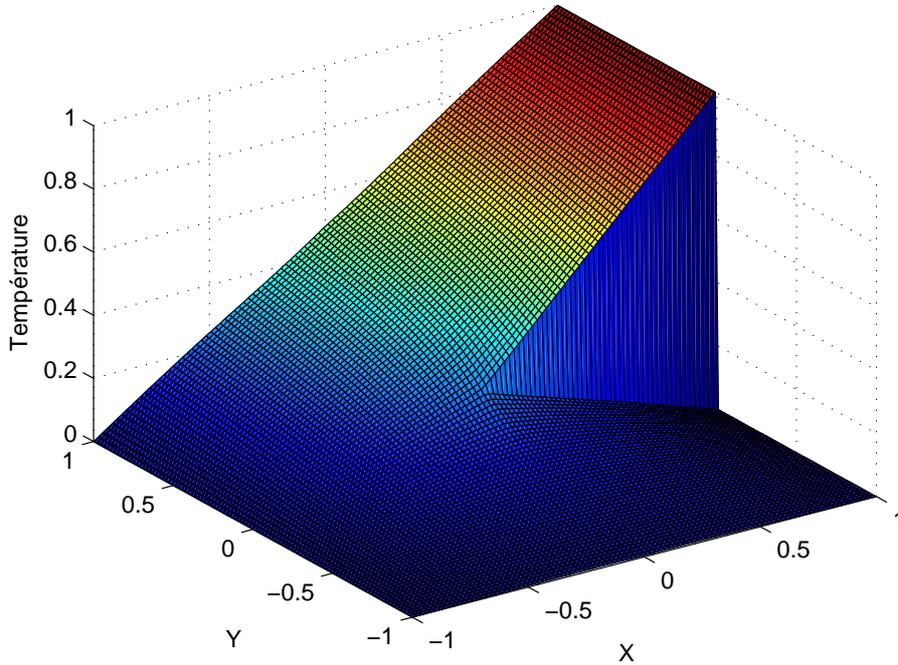


figure 2.10 – Solution donnée par la DPGD pour $f = 0$ et $L = 1$

blèmes multidimensionnels, avec des multidimensionnalités physiques et/ou paramétriques. On utilisera dans tout ce qui suit des algorithmes de résolution présentés dans ce chapitre. La formulation tensorielle décrite dans la section 2.1 adresse de façon efficace la solution d’une grande gamme de problèmes. Le temps de résolution est considérablement inférieur au temps classiquement prévu pour la résolution de tels problèmes.

D’autre part, la section 2.2 présente une façon d’adresser la solution de problèmes présentant une discontinuité des paramètres, utilisant la PGD. Une faible modification de l’algorithme de base est nécessaire afin d’englober la “communication” de fonctions discontinues avec d’autres continues. Toutefois, ces changements n’augmentent sensiblement pas la complexité ni le temps du calcul. Pour la résolution des problèmes englobant du transport, il faudra prévoir une stabilisation de l’équation à priori. Une telle démarche sera présentée dans le chapitre 4.

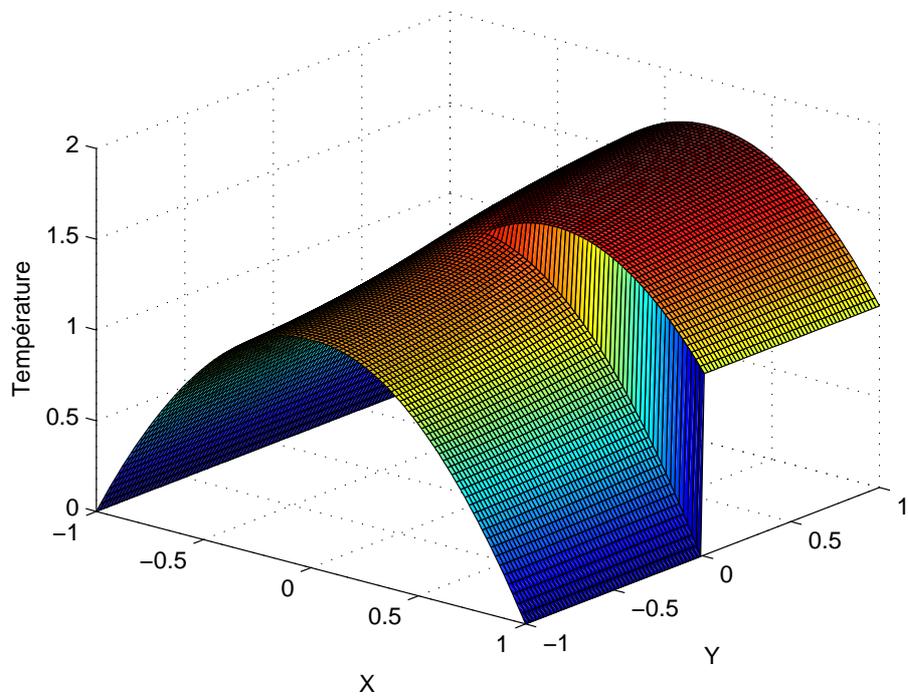


figure 2.11 – Solution donnée par la DPGD pour $f = 10$ et $L = 1$

3

La PGD dans la simulation de la pultrusion

3.1	Introduction	47
3.2	Optimisation de la pultrusion	48
3.3	Mariage de la PGD avec la DDDAS	79
3.4	Conclusion	108

Le but du travail développé dans ce chapitre est de proposer une méthodologie de modélisation, optimisation et contrôle de procédés industriels. Le procédé choisi dans ce chapitre est la pultrusion. Cependant, la méthode proposée peut très bien s'étendre à d'autres procédés.

Généralement, l'optimisation implique la présence d'une fonction coût, ainsi que la recherche de paramètres optimaux. De plus, le contrôle est un choix des paramètres optimisant une fonction coût, prenant en considération des variations dans le système. Afin d'assurer un bon asservissement du système contrôlé, le calcul doit s'effectuer rapidement, si possible en temps réel.

3.1 Introduction

Traditionnellement, la simulation se basait sur des entrées statiques, ce qui signifie que les valeurs des entrées ne changent pas durant la simulation. Cependant un nouveau paradigme de simulation connue sous le nom de la DDDAS ou "Dynamic Data-Driven application Systems" a vu le jour. Il consiste à incorporer des liens de retour en temps réel des systèmes simulés, pour mettre à jour les informations dans la simulation (authors, 2006; Ghnatios *et al.*, 2012). Cette technique de calcul possède un potentiel d'impact majeur dans les domaines de fabrication, commerce, transport, management, médecine et bien d'autres (Darema, 1994).

L'application la plus importante de la DDDAS est l'optimisation et le contrôle des procédés. Dans ce cas, nous avons besoin d'un temps de calcul énorme, beaucoup plus grand que le temps disponible pour achever l'asservissement. Dans ce qui suit, nous proposons d'abord une technique de calcul de modèles paramétriques. Un tel modèle possède une grande capacité d'accélération des problèmes d'optimisation. Dans un deuxième temps, un modèle de calcul en temps réel est montré. Une fois que l'optimisation en temps réel est possible, le contrôle de systèmes devient possible. Nous présenterons comment la combinaison de la PGD avec un schéma de calcul "Off-line/On-line" rend la DDDAS applicable dans le cadre d'une application industrielle bien répandue, la pultrusion.

3.2 Optimisation de la pultrusion

L'optimisation des procédés inclut le choix des paramètres optimaux. Classiquement, les stratégies d'optimisation tendent à définir un choix de paramètres, pour ensuite résoudre les équations du problème direct qui en découle. Plus tard, une fonction coût appropriée est définie pour rechercher l'optimalité. Tant que les paramètres optimaux ne sont pas encore atteints, le choix de la combinaison des paramètres est mis à jour suivant une stratégie adaptée, et une résolution du problème direct est entamée. De plus, cette stratégie devrait être suffisamment sophistiquée afin d'échapper aux pièges des minimums locaux. Evidemment, ceci influe dramatiquement le temps de calcul et d'optimisation.

Dans ce qui suit, nous proposons une méthode de modélisation numérique des problèmes qui surmonte ces difficultés. Ce scénario est par exemple rencontré dans les procédés composites où un flux de matière traverse un four. Un exemple intuitif de ce genre de procédé est la pultrusion. Nous montrons donc, d'un point de vue méthodologique, comment optimiser le procédé en ayant recours à la résolution du problème direct plus qu'une seule fois, en introduisant des paramètres procédés comme coordonnées. Evidemment, cela augmente la dimensionnalité du problème. Toutefois, la malédiction de la multidimensionnalité peut-être surmontée, grâce à l'utilisation de la PGD [Chinesta *et al.* \(2010b\)](#). Il est donc facile de remarquer les capacités offertes par un modèle pareil. Beaucoup de stratégies d'optimisation sont compatibles avec la résolution PGD. Dans ce qui suit nous proposons une méthode d'optimisation basée sur les moindres carrés mobiles ou "MLS". Cette stratégie utilise une reconstruction locale de la fonction au voisinage d'un point afin de réaliser une optimisation globale d'une fonction coût ([Breitkopf *et al.*, 2005](#)). Une telle stratégie est capable d'échapper aux pièges des optimums locaux et de trouver ainsi l'optimum global ([Breitkopf *et al.*, 2002](#)).



Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion

Ch. Ghnatios^{*,a}, F. Chinesta^b, E. Cueto^c, A. Leygue^a, A. Poitou^a, P. Breilkopf^d, P. Villon^d

^a*GEM: UMR CNRS-Centrale de Nantes-Université de Nantes
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France*

^b*EADS Corporate Foundation International Chair
GEM: UMR CNRS-Centrale de Nantes-Université de Nantes
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France*

^c*Aragón Institute of Engineering Research (I3A),
Universidad de Zaragoza,*

María de Luna, 3, E-50018 Zaragoza, Spain.

^d*UTC-Roberval, UMR 6253, F-60200 Compiègne, France*

Abstract

Optimization of manufacturing processes involves the optimal choice of many process parameters. Usual strategies proceed by defining a trial choice of those parameters and then solving the resulting model. Then, an appropriate cost function is evaluated and its optimality checked. While the optimum is not reached, the process parameters should be updated by using an appropriate optimization procedure, and then the model must be solved again for the updated process parameters. Thus, a direct numerical solution is needed for each choice of the process parameters, with the subsequent impact on the computing time. In this work we propose a methodological approach to the efficient numerical modeling and optimization of thermal processes taking

[☆]This work has been partially supported by the Spanish Ministry of Science and Innovation, through grant number CICYT-DPI2011-27778-C02-01.

*Corresponding author

Email addresses: chady.ghnatios@ec-nantes.fr (Ch. Ghnatios), Francisco.Chinesta@ec-nantes.fr (F. Chinesta), ecueto@unizar.es (E. Cueto), Adrien.Leygue@ec-nantes.fr (A. Leygue), Arnaud.Poitou@ec-nantes.fr (A. Poitou), piotr.breilkopf@utc.fr (P. Breilkopf), pierre.villon@utc.fr (P. Villon)



place in a die. This scenario is usually encountered in polymer and composites processing where material flows inside a die equipped with different heating devices. An example of such kind of processes concerns the pultrusion of composites. The main aim of this work is to describe an original approach for modeling and then optimizing the thermal process by solving only once the thermal model, and then, optimizing the process without the necessity of performing new solutions of the thermal model. For this purpose we introduce the temperatures of the heaters as extra-coordinates in the thermal model. The solution of the resulting multidimensional heat equation gives the temperature field for any choice of the temperature prescribed in the heaters. The curse of dimensionality is circumvented by invoking the Proper Generalized Decomposition —PGD— introduced in our former works but never until now used in the framework of process optimization.

Key words: Proper Generalized Decomposition; Separated representations; Model reduction; Optimization; Pultrusion process.

1. Introduction

The main aim of this work is proposing an original methodological approach to efficient numerical modeling and optimization. More specifically we are focusing on thermal processes taking place in a die, although the methodology proposed in this paper could easily be extended to many other processes.

In general optimization implies the definition of a cost function and the search of the optimum process parameters (e.g. temperatures of the heaters, temperature of the material coming to the die, flow rate, ... in the case of materials flowing into a heated die) defining the minimum of that cost function. The process starts by choosing a tentative set of process parameters. Then the process is simulated by discretizing the equations defining the model of the process. The solution of the model is the most costly step of the optimization procedure. As soon as that solution is available, the cost function can be evaluated and its optimality checked. If the chosen parameters do not define a minimum (at least local) of the cost function, the process parameters should be updated and the solution recomputed. The procedure continues until reaching the minimum of the cost function. Obviously, nothing ensures that such minimum is global, so more sophisticated procedures exist in order to explore the domain defined by the parameters and escape



from local minimums traps. The parameter updating is carried out in a way ensuring the reduction of the cost function. Many techniques update the model parameters in order to move along the cost function gradient. However, for identifying the direction of the gradient one should compute not only the fields involved in the model but also the derivatives of such fields with respect to the different process parameters. The evaluation of these derivatives is not in general an easy task. Conceptually, one could imagine that by perturbing slightly only one of the parameters involved in the process optimization and then solving the resulting model, one could estimate, using a finite difference formula, the derivative of the cost function with respect to the perturbed parameter. By perturbing sequentially all the parameters we could have access to the derivatives of the cost function with respect to all the process parameters (that is, the sensibilities) that define the cost function gradient, on which the new trial set of parameters should be chosen. There are many strategies for updating the set of process parameters and the interested reader can find most of them in the books focusing on optimization procedures.

Our interest here is not the discussion on particular optimization strategies, but pointing out that standard optimization strategies need numerous direct solutions of the problem that represents the process, one solution for each tentative choice of the process parameters. The solution of such models is a tricky task that demands important computational resources and usually implies extremely large computing times.

In this paper we propose a radically different approach, to the authors' knowledge never explored. The approach here proposed considers the unknown process parameters as new coordinates of the model. In fact, coordinates, or space dimensions, represent the (non-necessarily physical) locations at which the solution is to be represented. Thus, strictly speaking, one could compute the solution of the problem for any value of the unknown parameters (in a bounded interval). This converts those unknown parameters in new dimensions of the space in which the model is defined. This idea seems exciting but it involves a major difficulty.

This strategy faces a challenging problem if the number of parameters of the model increases. It is well known that the number of degrees of freedom for a mesh-based discretization technique (say, finite element, finite difference, ...) increases exponentially with the number of dimensions. Thus, for a hypercubic domain, the number of degrees of freedom scales with the number of nodes along each spatial direction to the power of the number



of dimensions. For instance, in 2D and considering 100 nodes along each direction with a single degree of freedom per node, the resulting number of degrees of freedom becomes 100^2 . In 3D, the number of degrees of freedom rises to 100^3 and so on. This exponential increase of the number of degrees of freedom can be literally out of reach for nowadays computers even if the number of dimensions increases only moderately. This phenomenon is known as *curse of dimensionality*.

Of course, to efficiently deal with this problem a strategy different of mesh-based discretization methods should be employed. Although efficient techniques exist for moderate number of spatial dimensions, such as sparse grid methods [4], they fail when the dimensionality increases. Here, we consider the use of Proper Generalized Decompositions (PGD) [1] [2] [7] [8]. PGD techniques construct an approximation of the solution by means of a sequence of products of separable functions. These functions are determined “on the fly”, as the method proceeds, with no initial assumption on their form.

The PGD method, while it can be considered as a model reduction technique (and hence its name, a generalization of the Proper Orthogonal Decomposition —POD—) can deal very efficiently with highly multi-dimensional problems, since only a sequence of low-dimensional problems is solved. Details of the technique are provided in Section 2 of this paper.

Once an efficient strategy of dealing with high-dimensional solutions has been defined, the numerical solution of problems with unknown data becomes straightforward. As mentioned before, the strategy here proposed consists of considering the unknown parameters as new coordinates of the model. Thus, the solution is computed only once and it allows to have access to the unknown field, as well as to the explicit expression of its derivatives, for any possible choice of the model parameters by a simple particularization of the parametric solution, that is, by a simple postprocessing.

As can be readily noticed, the potential of the technique for inverse identification, optimization, etc. seems to be huge. In this paper we propose a methodological approach in this direction. In what follows we are considering a thermal model of a material moving through a die equipped with some heating devices on the die walls. We could consider as process parameters the temperatures prescribed in the different heaters, the flow rate, the temperature of the material coming into the die, etc. For the sake of simplicity in what follows we are restricting the parametric space to the heating devices temperatures. The choice of the cost function depends on each particular



process. There are many choices and because in this work we are more interested in proposing and illustrating a new methodological modeling and optimization approach than in analyzing deeply a particular process, we will restrict our analysis to a simplified model of pultrusion processes. Pultrusion is a continuous process to produce constant cross-sectional profile composites. During this process, fiber reinforcements are saturated with resin, which are then pulled through a heated die. The resin gradually cures inside the die while generating heat. At the exit, pullers draw the composite out and a traveling cut-off saw cuts it at the designed length. We consider the process conditions described in [5] and sketched in Fig. 1.

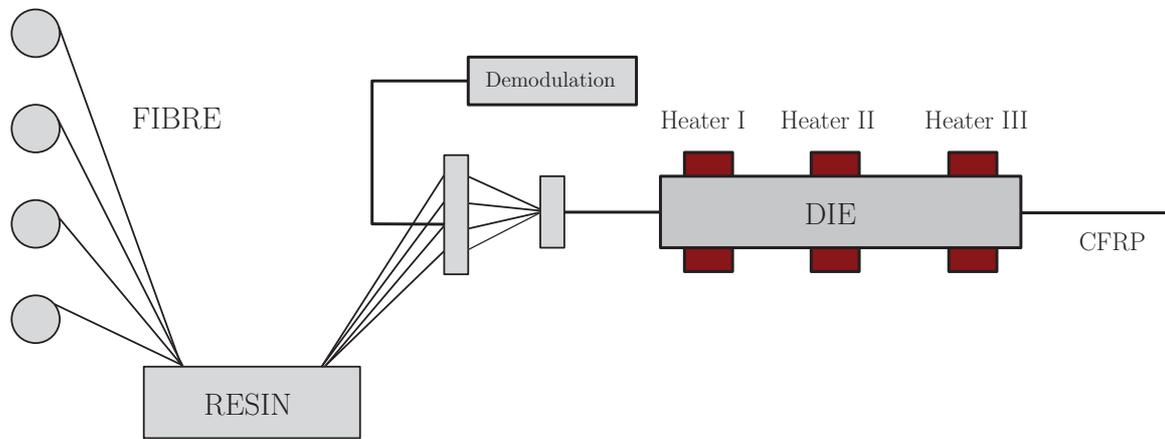


Figure 1: Pultrusion process.

For decades, engineers have relied on experience to define optimal parameters for pultrusion processes, pushed by the popularization of this technique in industry [10] [12]. Nowadays, this process has been extended to thermoplastic resins as well as to reactive systems, in which the monomers polymerize inside the die, an efficient route for considering high viscosity thermoplastic resins. In these scenarios new efficient optimization procedures are urgently needed. Standard optimization approaches and others based on the use of genetic algorithms have been recently proposed and applied (see for example [5] and the references therein). However the efficiency of those approaches seems to be limited to a reduced number of process parameters because one must solve a thermal model for each choice of the process parameters, and it is well known that when the dimension of the parametric space increases the exploration of the space defined by the process parameters be-

comes more and more arduous, needing for numerous, sometimes excessive, solutions of the model governing the process.

If we consider the thermal model related to the pultrusion process sketched in figure 1, whose parametric space reduces to the temperatures prescribed at the three heating devices, θ_1 , θ_2 and θ_3 , we could summarize traditional optimization procedures as follows:

- *Until reaching a minimum of the cost function $\mathcal{C}(\theta_1, \theta_2, \theta_3)$ proceed by:*
 1. Computing the temperature field related to the trial choice of the process parameters, i.e. $u(\mathbf{x}; \theta_1, \theta_2, \theta_3)$.
 2. Computing the cost function $\mathcal{C}(\theta_1, \theta_2, \theta_3)$ from the just calculated thermal field.
 3. Checking the optimality. While the optimum is not reached, update the process parameters by using an appropriate strategy and comeback to step 1 for another solution of the thermal model for the process parameters just updated.

In the approach that we propose in this work the procedure is substantially different. It proceeds as follows:

- Compute the thermal field for any possible choice of the process parameters: $u(\mathbf{x}, \theta_1, \theta_2, \theta_3)$ (here the heaters temperatures play the same role that the space coordinates), the problem becoming multidimensional.
- *Until reaching a minimum of the cost function $\mathcal{C}(\theta_1, \theta_2, \theta_3)$ proceed by:*
 1. Particularizing the parametric solution to the considered values of the process parameters.
 2. Computing the cost function $\mathcal{C}(\theta_1, \theta_2, \theta_3)$ from the just calculated thermal field.
 3. Checking the optimality. While the optimum is not reached, update the process parameters by using an appropriate strategy and comeback to step 1 for another particularization of the parametric solution.

Thus, in our proposal the thermal model is solved only once and then it is particularized for any choice of the process parameters. The price to pay is the necessity of solving a multidimensional thermal model that now has



as coordinates the physical space \mathbf{x} and all the process parameters, i.e. the three heaters temperatures in the example addressed here.

Obviously, the solution of the resulting multidimensional model is a tricky task if one consider a standard mesh based discretization strategy because the number of degrees of freedom increases exponentially with the dimensionality of the model. To circumvent this serious difficulty, also known as *curse of dimensionality*, we consider a separated representation of the temperature field in the PGD framework, in which the temperature reads:

$$u(\mathbf{x}, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(\mathbf{x}) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (1)$$

To build up such separated representation we only need to compute the functions defined in the space domain $F_i(\mathbf{x})$ and the one-dimensional functions $\Theta_{ji}(\theta_j)$, $j = 1, 2, 3$, defined in the intervals in which the heaters temperatures can evolve.

In section 2 we revisit the main ideas involved in the construction of such separated representation. In section 3 we describe the procedure to include the boundary conditions as extra-coordinates. In section 4 a simplified pultrusion model is addressed. Then, in section 5 we consider one possible optimization strategy based on the combination of the already computed parametric solution with a minimization strategy based on the use of moving least squares on the response surface. Due to the methodological purposes of this paper a simple cost function will be considered, enforcing a constant temperature of the flowing material on die outlet. Despite the simple scenario here addressed the reader can appreciate the huge potentialities that such approach represents.

2. Illustrating the solution of multidimensional parametric models by using the PGD

Imagine for example that you are interested in solving the heat equation but that you do not know the material thermal conductivity, because it has a stochastic nature or simply because prior to solve the thermal model you should measure it. You have three possibilities: (i) you wait to know the conductivity before solving the heat equation (a conservative solution!); (ii) you solve the equation for many values of the conductivity (a sort of Monte Carlo) and then the work is done (a sort of brute force approach!); or (iii)



you solve the heat equation only once for any value of the conductivity (the cleverest alternative!). Obviously the third alternative is the most exciting one. To compute this “magic” solution it suffices to introduce the conductivity as an extra coordinate, playing the same role than the standard space and time coordinates, even if the model does not involve derivatives with respect to this extra-coordinate. This procedure runs very well, and can be extended for introducing many other extra-coordinates: the source term, initial condition, etc. It is easy to understand that after performing this type of calculations, a posteriori inverse identification or optimization can be easily handled, but we will come back to these potential applications later.

In what follows, we illustrate the construction of the Proper Generalized Decomposition [11] [1] [2] [7] [14] of the solution by considering the following simple parametric heat transfer equation:

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0 \quad (2)$$

where $(\mathbf{x}, t, k) \in \Omega \times I \times \mathfrak{S}$. $\Omega \subset \mathbb{R}^d$ is the physical space in which the model is defined, I is the time interval in which the solution is found and finally \mathfrak{S} represents the interval in which the conductivity could take its value.

For the sake of simplicity the source term is assumed to be constant, i.e. $f = \text{cte}$. Because the conductivity is considered unknown, it is assumed to be a new coordinate defined in the interval \mathfrak{S} . Thus, instead of solving the thermal model for different values of the conductivity parameter we are introducing it as a new coordinate. Therefore, we solve a more general problem, but obviously the price to pay is the increase of the model dimensionality. However, as the complexity of PGD scales only linearly (and not exponentially) with the space dimension, the consideration of the conductivity as a new coordinate still allows one to obtain a fast and cheap solution. Thus, in this case, the solution of equation (2) is searched under the form:

$$u(\mathbf{x}, t, k) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) \quad (3)$$

To start the PGD algorithm, we assume that the approximation at iteration n is already known:

$$u^n(\mathbf{x}, t, k) = \sum_{i=1}^{i=n} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) \quad (4)$$



and that at the present iteration we look for the next functional product $X_{n+1}(\mathbf{x}) \cdot T_{n+1}(t) \cdot K_{n+1}(k)$ that for alleviating the notation will be denoted by $R(\mathbf{x}) \cdot S(t) \cdot W(k)$. Before solving the resulting non linear model related to the calculation of these three functions, a model linearization is performed. The simplest choice consists in using an alternating-directions fixed-point algorithm. First of all, we proceed by assuming $S(t)$ and $W(k)$ given at the previous iteration of the non-linear solver and then computing $R(\mathbf{x})$. From the just updated $R(\mathbf{x})$ and the previously used $W(k)$ we can update $S(t)$. Finally from the just computed $R(\mathbf{x})$ and $S(t)$ we update $W(k)$. The procedure continues until reaching convergence. The converged functions $R(\mathbf{x})$, $S(t)$ and $W(k)$ allow us to define all the needed functions: $X_{n+1}(\mathbf{x}) = R(\mathbf{x})$, $T_{n+1}(t) = S(t)$ and $K_{n+1}(k) = W(k)$. The explicit form of each one of the just referred steps is widely described in the appendix.

Remark 1. The construction of each term in Eq. (3) needs a certain number of iterations because of the non-linearity of the problem related to the approximation given by Eq. (4). Denoting by m_i the number of iterations needed for computing the i -th sum in Eq. (3), let $m = \sum_{i=1}^{i=N} m_i$ be the total number of iterations involved in the construction of the separated approximation, Eq. (3). It is easy to note that the solution procedure involves the solution of m dD ($d = 1, 2, 3$) problems related to the construction of the space functions $X_i(\mathbf{x})$, $i = 1, \dots, N$; m $1D$ ordinary differential equations related to the construction of functions $T_i(t)$ and, finally, m linear systems related to the definition of functions $K_i(k)$. In general m_i rarely exceeds ten. On the other hand the number N of sums needed to approximate the solution of a given problem depends on the solution regularity itself, but all the experiments carried out so far reveal that this number ranges from few tens to slightly more than one hundred. Thus, we can conclude that the complexity of the solution procedure is of some tens of solutions defined in the dD space (the cost related to the one dimensional time problems being negligible with respect to the one related to the space problems [6]). On the contrary, if we follow a classical approach, we should solve a dD problem at each time step and for each value of the parameter k . In usual applications the complexity can easily reach millions of space solutions. In [8] the authors proved that the CPU time savings by applying the PGD can be of several orders of magnitude.

Remark 2. Note also that another possibility exists consisting in the separation of the three-dimensional physical space into a sequence of one-dimensional

ones:

$$u(x, y, z, t, k) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \cdot T_i(t) \cdot K_i(k) \quad (5)$$

This possibility further reduces the complexity mentioned before, allowing to solve models with some 10^9 degrees of freedom in few minutes in a laptop. Techniques to cope in this framework with non-paralelepipedic domains in three or more dimensions have been analyzed in [9], in which a domain penalization strategy was proposed.

3. Parametric boundary conditions

Very often, in the optimization of an industrial process, it is necessary to solve the problem for different boundary conditions. Boundary conditions do not behave as any other parameter in the PGD, and therefore deserve some additional comments. In general, it is needed to perform a change of variable to introduce the boundary condition into the differential equation and then define it as an extra coordinate. To illustrate this procedure we consider, for the sake of simplicity and without any loss of generality, the following simple problem:

$$-\Delta u = f \text{ in } \Omega \quad (6)$$

subjected to the boundary conditions:

$$u = g \neq 0 \text{ on } \Gamma \equiv \partial\Omega \quad (7)$$

Let us assume that we are able to find a function ψ , continuous in the closure of Ω , $\bar{\Omega}$, such that $-\Delta\psi \in L_2(\Omega)$ verifying Eq. (7). Then, the solution of the problem given by Eqs. (6)-(7) can be obtained straightforwardly by

$$u = \psi + z, \quad (8)$$

where we thus face a problem in the z variable

$$-\Delta z = f + \Delta\psi \text{ in } \Omega \quad (9)$$

$$z = 0 \text{ on } \Gamma \quad (10)$$

easily solvable by the PGD method presented before. This procedure was deeply analyzed in [9].



The introduction of the value of the boundary condition (g in Eq. (7)) as an extra-coordinate generates an extra dimension in the problem, which means dramatically increasing the computation cost of the problem solution in classical mesh-based numerical methods for discretizing partial differential equations.

3.1. Illustrating the main ideas from a numerical example

To fix the ideas, let us start by considering the one dimensional differential equation that follows:

$$-\frac{\partial^2 u}{\partial x^2} = 1 \quad (11)$$

to be solved in the domain $x \in \Omega = (0, L = 10)$ and for the following boundary conditions:

$$\begin{cases} u(0) = 0 \\ u(L) = y \end{cases} \quad (12)$$

where y is a parameter taking values within the interval α . To prescribe non-homogeneous boundary conditions in the framework of the PGD, it is a common practice, as just argued, to perform a change of variable to transform the non-homogeneous boundary conditions into homogeneous ones [9]. To this end, often simple procedures suffice. For instance, the following function $g(x, y)$ satisfies all the prescribed boundary conditions (12):

$$g(x, y) = \frac{x \cdot y}{L} \quad (13)$$

Since, as we have seen in Section 2, the problem solution $u(x, y)$ can be written in the separated form:

$$u(x, y) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \quad (14)$$

we can introduce the function verifying the essential boundary conditions, $g(x, y)$, as the first mode of the solution $X_1(x) \cdot Y_1(y)$. In other words,

$$X_1(x) \cdot Y_1(y) = \frac{x \cdot y}{L} \quad (15)$$



and

$$u(x, y) \approx X_1(x) \cdot Y_1(y) + \sum_{i=2}^{i=N} X_i(x) \cdot Y_i(y) \quad (16)$$

Now we proceed as shown in Section 2. The trial function after n iterations reads:

$$u(x, y) = X_1(x) \cdot Y_1(y) + \sum_{i=2}^{i=n} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y) \quad (17)$$

By replacing Eq. (17) into the weak form of Eq. (11), and repeating the same procedure used in Section 2 and the appendix we have:

- Computing $R(x)$ from $S(y)$. In this case the test function is written as:

$$u^*(x, y) = R^*(x) \cdot S(y) \quad (18)$$

and the resulting weak form reads:

$$\int_{\Omega} \int_{\alpha} \left(R^* \cdot S \cdot Y_1 \cdot \frac{d^2 X_1}{dx^2} + R^* \cdot S \cdot \left(\sum_{i=2}^{i=n} \frac{d^2 X_i}{dx^2} \cdot Y_i \right) + R^* \cdot S \cdot \frac{d^2 R}{dx^2} \cdot S + R^* \cdot S \right) dy \cdot dx = 0 \quad (19)$$

As functions involving the parametric coordinate y are assumed known in the present step, integrals in the parametric domain α can be calculated, leading to a problem defined in the space domain Ω whose solution is precisely the searched unknown function $R(x)$.

- Computing $S(y)$ from $R(x)$. In this case the test function is:

$$u^*(x, y) = R(x) \cdot S^*(y) \quad (20)$$

and the weak function now reads:

$$\int_{\Omega} \int_{\alpha} \left(S^* \cdot R \cdot Y_1 \cdot \frac{d^2 X_1}{dx^2} + S^* \cdot R \cdot \left(\sum_{i=2}^{i=n} \frac{d^2 X_i}{dx^2} \cdot Y_i \right) + S^* \cdot R \cdot \frac{d^2 R}{dx^2} \cdot S + S^* \cdot R \right) dy \cdot dx = 0 \quad (21)$$



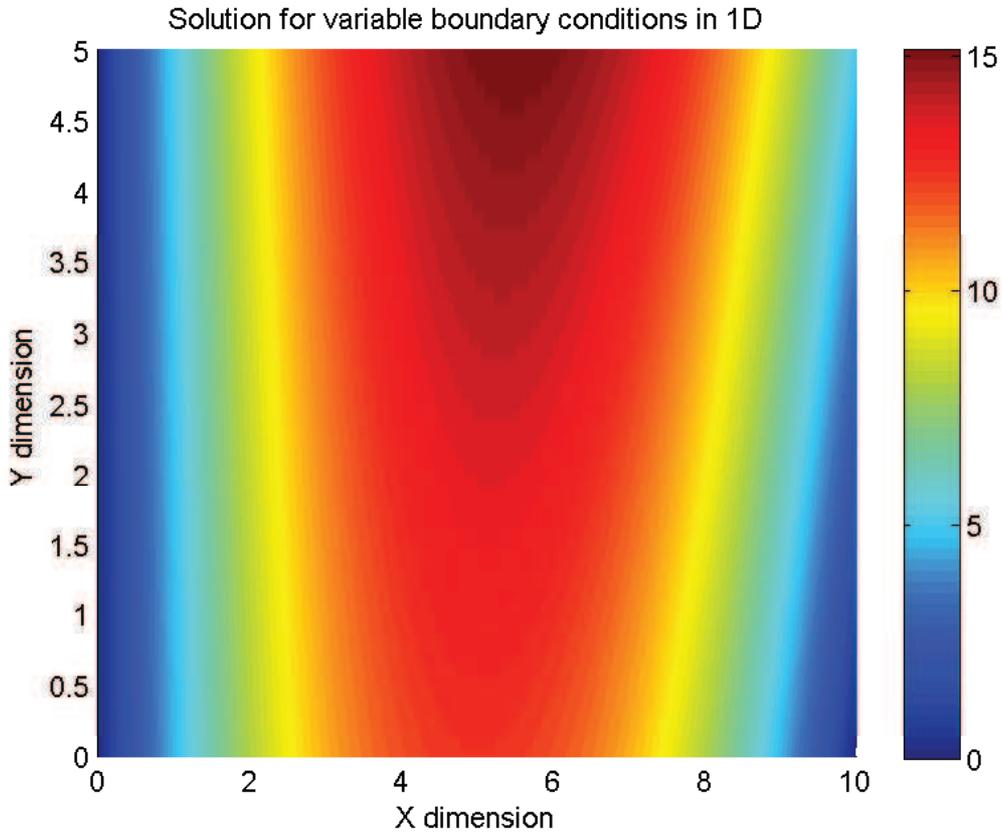


Figure 2: Solution of the parametric Eq. (11). The solution $u(x)$ is depicted for any possible boundary condition $y \in [0, 5]$.

As functions involving the space coordinate x are assumed known in the present step, integrals in the domain Ω can be calculated, leading to a problem defined in the parametric domain α whose solution is precisely the searched unknown function $S(y)$.

Functions R and S vanish in the boundary of the domain. That is mandatory, because the boundary conditions are already satisfied by the first mode. The new modes improve the solution inside the domain. The reconstructed solution is depicted in Fig. 2 for any value of $y \in [0, 5]$.

3.2. Boundary conditions as parameters

Once the main ingredient of the method has been introduced, we are generalizing the procedure for treating more complex models. For the sake



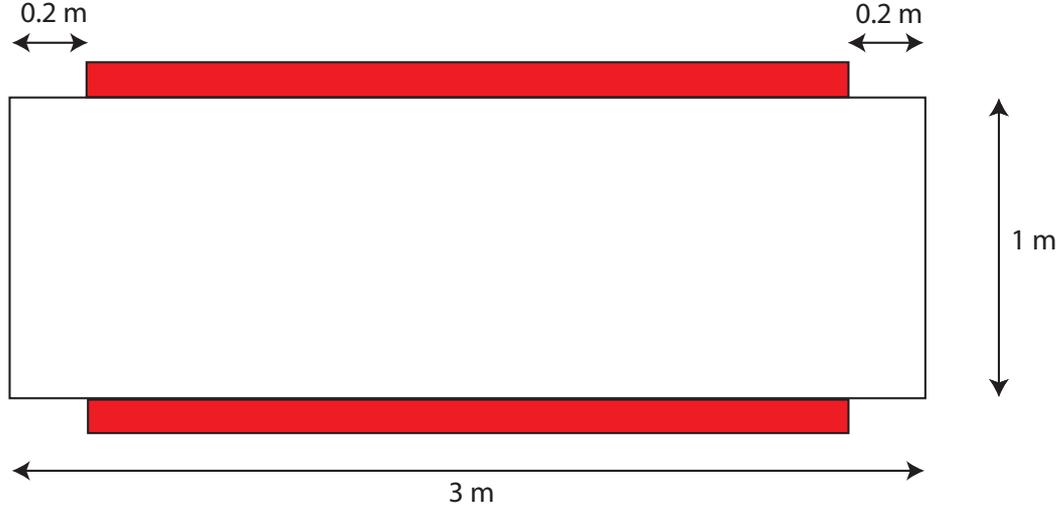


Figure 3: Domain geometry and boundary conditions.

of simplicity, we describe the solution of a two dimensional model defined in $\Omega = (0, L_1 = 3) \times (0, L_2 = 1)$. Let us consider the basic steady-state heat transfer equation:

$$-K \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = q \quad (22)$$

On the domain boundary $\Gamma \equiv \partial\Omega$ either the temperature or the heat flux are enforced, the first one on the part of the boundary Γ_D and the second one on the remaining part $\Gamma_N = \Gamma - \Gamma_D$. Figure 3 shows a sketch of the domain geometry and the boundary conditions.

We assume the prescribed temperatures given by:

$$\begin{cases} u(0, y) = U_0 \\ u(\epsilon \leq x \leq L_1 - \epsilon, y = 0) = \theta \\ u(\epsilon \leq x \leq L_1 - \epsilon, y = L_2) = \theta \end{cases} \quad (23)$$

where $\epsilon = 0.2m$, $U_0 = 300K$ is a constant value and θ any value defined in a certain interval α .

A null heat flux is enforced in the remaining part of the domain boundary, i.e. $\nabla u \cdot \mathbf{n}|_{\Gamma_N} = 0$, where \mathbf{n} represents the outward unit vector defined on the domain boundary.

Traditionally, this type of problems is solved by considering different values of that parameter and solving the resulting model for each one of these values, with the consequent impact on the computing time. Here we propose to introduce this parameter θ as a new extra-coordinate, and then solving only once the resulting multi-dimensional model. The solution $u(x, y, \theta)$ allows one to have access to the value of the field u at each space position (x, y) and for each value of the parameter θ in the interval α in which it is defined ($\theta \in (300, 500)$ in this example).

Prior to solve the problem it is necessary to find a function satisfying the essential boundary conditions, in order to perform the change of variable illustrated in the previous section. For arbitrarily-shaped domains a somewhat more sophisticated technique was introduced by the authors in [9] and will not be reproduced here. The interested reader can refer to this reference.

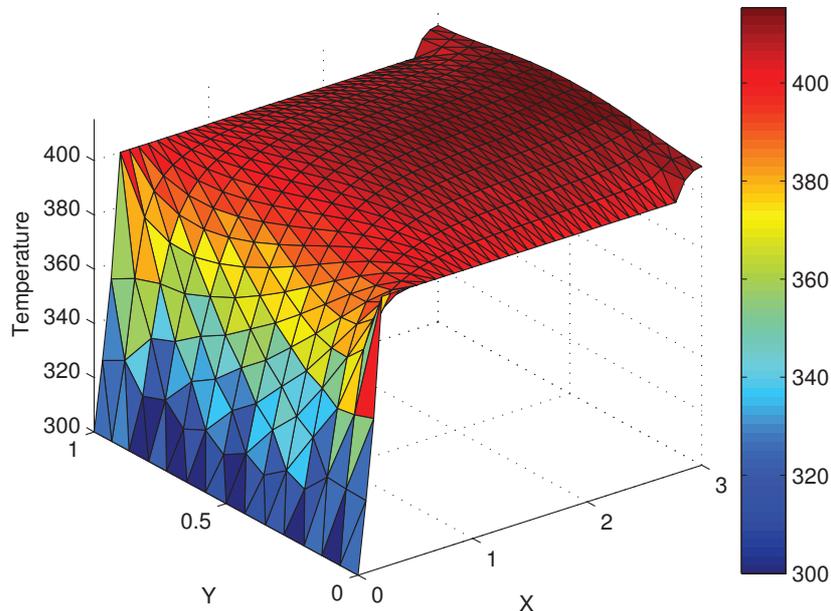
Therefore, the solution is assumed to be of the form:

$$u(x, y, \theta) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_i(\theta) \quad (24)$$

In the present case, it is difficult to find a single function satisfying all boundary conditions, unlike the previous example. Two initial modes have been considered in the separated representation of the solution. The first mode fixes the value of the solution at $x = 0$ by prescribing the nodal values at that location, and vanishing elsewhere. The second one fixes the boundary conditions at $y = 0$ and $y = L_2$, also vanishing at the remaining nodes of the domain. This *ad hoc* procedure is not always easy to implement. For more sophisticated cases, a special type of transfinite interpolation was introduced in [9].

After applying the change of variable related to the imposition of essential boundary conditions, we solve Eq. (22) with homogeneous Dirichlet boundary conditions. The solution is represented in Fig. 4 for $\theta = 400K$.

Comparing this solution to the standard finite element one, which is computed for the precise boundary conditions mentioned before to avoid the increase of the dimensionality, a very small difference is found, lower than 10^{-12} using the L_2 -norm. For more details on the accuracy of the technique in general settings, we address the reader to [7].

Figure 4: Temperature field for $\theta = 400K$.

4. Parametric thermal model of a heated die in pultrusion processes

In modeling the pultrusion process, as sketched in Fig. 1, we consider the thermal process within the die as modeled by the following two-dimensional convection-diffusion equation:

$$\rho \cdot C \left(\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} \right) = k \Delta u + q \quad (25)$$

where k is the thermal conductivity, q is the internal heat generated by the resin curing reaction, ρ is the density, C is the specific heat and v is the extruded profile speed. The material flowing inside the die is in contact with the die wall. Thus, conduction is the only heat transfer mechanism taking place.

The die is equipped with three heaters as depicted in Fig. 5 whose temperatures constitute the process parameters to be optimized. For the sake of simplicity we consider constant profile velocity v and inlet temperature U_0 , all of them assumed known. The curing kinetics was coupled with the

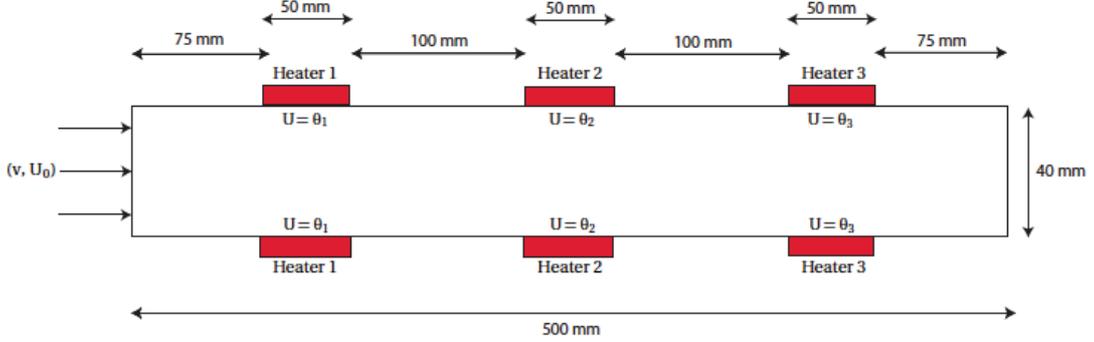


Figure 5: Domain geometry and boundary conditions.

thermal field as described in [15].

Thus, the temperature field u depends on five different coordinates, the two space coordinates (x, y) and the three temperatures prescribed in three regions on the die wall. In fact $u = u(x, y, \theta_1, \theta_2, \theta_3)$, where θ_1 , θ_2 and θ_3 are the temperature of the heaters I, II and III, depicted in Fig. 5 respectively. The separated representation of u reads:

$$u(x, y, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (26)$$

Temperatures are assumed prescribed as Dirichlet boundary conditions on Γ_D whereas a null flux is prescribed on the complementary part $\Gamma_N = \Gamma - \Gamma_D$, i.e. $\nabla u \cdot \mathbf{n}|_{\Gamma_N} = 0$.

The prescribed temperatures write:

$$\begin{cases} u(x=0, y, \theta_1, \theta_2, \theta_3) = U_0 \\ u(x \in L_1, y=0 \text{ or } y=h, \theta_1, \theta_2, \theta_3) = \theta_1 \\ u(x \in L_2, y=0 \text{ or } y=h, \theta_1, \theta_2, \theta_3) = \theta_2 \\ u(x \in L_3, y=0 \text{ or } y=h, \theta_1, \theta_2, \theta_3) = \theta_3 \end{cases} \quad (27)$$

where L_1 , L_2 and L_3 are the intervals of the x -coordinate related to heaters I, II and III respectively. $L = 0.5m$ is the length of the die and $h = 0.04m$ its width. The parameters θ_1 , θ_2 and θ_3 take values in the intervals \mathcal{I}_1 , \mathcal{I}_2 and \mathcal{I}_3 respectively.

To solve this problem, we need to define some functional products in order to introduce the boundary conditions (27) into the partial differential

equation (four modes suffices for this end) and then proceed by applying the Proper Generalized Decomposition as described in section 2 and the appendix.

5. Optimization strategy

In this section we consider the parametric solution already computed:

$$u(x, y, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (28)$$

The objective of the optimization procedure consists of the determination of the process parameters θ_1 , θ_2 and θ_3 in order to minimize an appropriate cost function depending on the considered physics. In this work we are only interested in describing the main ingredients involved in the proposed optimization strategy, and for this reason, in what follows, we consider a cost function without any particular physical significance.

The cost function is defined from the difference between the temperature on the outflow boundary $x = L$, $u(x = L, y; \theta_1, \theta_2, \theta_3)$, and the desired one $\bar{u}(y)$. Thus, the cost function writes:

$$\mathcal{C}(\theta_1, \theta_2, \theta_3) = \frac{1}{2} \int_0^h (u(x = L, y; \theta_1, \theta_2, \theta_3) - \bar{u}(y))^2 dy \quad (29)$$

We look for a uniform profile of temperatures on the outlet. This kind of condition is quite difficult to fulfill due to the heat conduction mechanism as well as the internal heat generation. We consider this cost function only to prove the robustness of the proposed optimization approach.

The parametric domain is defined by $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 \times \mathcal{I}_3$ in which the optimal solution is sought. The optimal solution constitutes a minimum of the cost function (29). We denote points in the parametric domain by $\boldsymbol{\theta}$ with components $(\theta_1, \theta_2, \theta_3)$.

The algorithm starts by considering an arbitrary point within \mathcal{I} . Then, the gradient and Hessian defined by the cost function are computed in order to apply a Newton strategy. For computing both, we need to define an appropriate approximation of the cost function in the vicinity of the evaluation point. By defining a quadratic approximation we could compute the gradient and the Hessian, and because the approximation is quadratic the Newton algorithm is an appropriate choice for performing the minimization.



Let $\boldsymbol{\theta}^0 \in \mathcal{I}$ be the starting point. We consider a small paralepipedic volume \mathcal{P}^0 in the parametric space centered at that point in which the cost function will be approximated using a polynomial approximation of a certain order, and whose edges lengths l_1 , l_2 and l_3 , in the directions of the axes θ_1 , θ_2 and θ_3 respectively, depend on the desired order of the approximation as well as on the mesh used for discretizing the intervals \mathcal{I}_1 , \mathcal{I}_2 and \mathcal{I}_3 in which θ_1 , θ_2 and θ_3 , respectively, are defined. Then, a number of points $\boldsymbol{\theta}_i^0$, $i = 1, \dots, n_{LH}$, are considered within \mathcal{P}^0 according to the LATIN hypercube technique.

The cost function \mathcal{C}_i^0 , $i = 1, \dots, n_{LH}$ is computed at those points $\boldsymbol{\theta}_i^0$, $i = 1, \dots, n_{LH}$. Now, a quadratic approximation of $\mathcal{C}(\theta_1, \theta_2, \theta_3)$ could be defined in \mathcal{P}^0 , $\mathcal{C}^h(\boldsymbol{\theta} \in \mathcal{P}^0)$, and then both the gradient and the Hessian computed.

Obtaining a smooth enough, quadratic approximation of a field given by scattered data is not always easy. For this purpose we make use of a centered moving least square (MLS) strategy that we revisit in the next section.

5.1. The Moving Least Squares approximation revisited

In this section we revisit the Moving Least Squares —MLS— approximation introduced by Nayroles et al. [13] that will be then used for defining the optimization strategy.

We look for a local approximation of a generic function $u(\mathbf{x})$ at point \mathbf{x} , $u^h(\mathbf{x})$, based on the nodal values u_i of the function $u(\mathbf{x})$ at a limited number n of points \mathbf{x}_i close to \mathbf{x} . Thus, the unknown field is approximated in the vicinity of \mathbf{x} by:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{p}(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) \quad (30)$$

with $\mathbf{p}(\mathbf{x})$ a polynomial basis, i.e. $\mathbf{p}^T(\mathbf{x}) = [1, x_1, x_2, x_1 \cdot x_2]$ and $\mathbf{p}^T(\mathbf{x}) = [1, x_1, x_2, x_1 \cdot x_2, x_1^2, x_2^2]$ for a bilinear and quadratic basis, respectively, in 2D, and $\mathbf{a}(\mathbf{x})$ a vector of unknown coefficients. In order to determine $\mathbf{a}(\mathbf{x})$, we define the functional $J(\mathbf{a})$ that must be minimized with respect to $\mathbf{a}(\mathbf{x})$:

$$J(\mathbf{a}) = \frac{1}{2} \sum_{i=1}^n w_i(\mathbf{x}) (\mathbf{p}(\mathbf{x}_i) \cdot \mathbf{a}(\mathbf{x}) - u_i)^2 \quad (31)$$

where u_i are the nodal unknowns associated with the neighboring nodes \mathbf{x}_i of point \mathbf{x} and $w_i(\mathbf{x})$ is a weighting function whose value decreases as the distance between \mathbf{x}_i and \mathbf{x} increases. The weight function $w_i(\mathbf{x})$ vanishes at

a finite distance from \mathbf{x}_i , called radius of influence. In general it is positive, takes its maximum value (usually unit) at position \mathbf{x}_i and decreases continuously over the domain of influence. There are many choices according to the problem to be solved. The reader can refer to [3] for more details on the properties of this function and the most used ones. The minimization of $J(\mathbf{a})$ with respect to the unknown coefficient $a_j(\mathbf{x})$ leads to:

$$\frac{\partial J(\mathbf{a})}{\partial a_j(\mathbf{x})} = \sum_{k=1}^n a_k \left[\sum_{i=1}^n w_i(\mathbf{x}) p_j(\mathbf{x}_i) p_k(\mathbf{x}_i) \right] - \sum_{i=1}^n w_i(\mathbf{x}) p_j(\mathbf{x}_i) u_i = 0 \quad (32)$$

which leads to the linear system:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u} \quad (33)$$

where matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are defined by:

$$A_{jk}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \cdot p_j(\mathbf{x}_i) \cdot p_k(\mathbf{x}_i) \quad (34)$$

$$B_{ij}(\mathbf{x}) = w_i(\mathbf{x}) \cdot p_j(\mathbf{x}_i) \quad (35)$$

Substituting $\mathbf{a}(\mathbf{x})$ in Eq. (30), results in:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \quad (36)$$

The difference between the diffuse finite element and the element free Galerkin schemes comes from the evaluation of the shape function derivatives. In the first scheme only the term $\mathbf{p}(\mathbf{x})$ in Eq. (30) is derived

$$\frac{\delta u^h(\mathbf{x})}{\delta x_i} = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_i} \cdot \mathbf{a}(\mathbf{x}), \quad i = 1, 2 \quad (37)$$

whereas all terms depending on \mathbf{x} are derived in the element free Galerkin approach, according to:

$$\frac{\partial u^h(\mathbf{x})}{\partial x_i} = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_i} \cdot \mathbf{a}(\mathbf{x}) + \mathbf{p}(\mathbf{x}) \cdot \frac{\partial \mathbf{a}(\mathbf{x})}{\partial x_i}, \quad i = 1, 2. \quad (38)$$



5.2. Centered Moving Least Squares

We define the polynomial basis \mathbf{q} centered at the evaluation point \mathbf{x} . For a node \mathbf{x}_i we have

$$\mathbf{q}^T(\mathbf{x}^i - \mathbf{x}) = [1, (x_1^i - x_1), (x_2^i - x_2), (x_1^i - x_1) \cdot (x_2^i - x_2), \dots] \quad (39)$$

For the sake of clarity, and without loss of generality, from now on we consider the bilinear basis $\mathbf{q}^T = [1, (x_1^i - x_1), (x_2^i - x_2), (x_1^i - x_1) \cdot (x_2^i - x_2)]$.

It is easy to verify the relationship

$$\mathbf{Q}(\mathbf{x}) \cdot \mathbf{p}(\mathbf{x}_i) = \mathbf{q}(\mathbf{x}^i - \mathbf{x}) \quad (40)$$

with the matrix $\mathbf{Q}(\mathbf{x})$ given by:

$$\mathbf{Q}(\mathbf{x}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -x_1 & 1 & 0 & 0 \\ -x_2 & 0 & 1 & 0 \\ x_1 \cdot x_2 & -x_2 & -x_1 & 1 \end{pmatrix} \quad (41)$$

Because matrix $\mathbf{Q}(\mathbf{x})$ is nonsingular (it corresponds to a change of basis in a polynomial vector space), we can write:

$$\mathbf{p}(\mathbf{x}_i) = \mathbf{Q}^{-1}(\mathbf{x})\mathbf{q}(\mathbf{x}^i - \mathbf{x}) \quad (42)$$

from which the approximation $u(\mathbf{x}_i)$ reads:

$$u(\mathbf{x}_i) = \mathbf{p}(\mathbf{x}_i) \cdot \mathbf{a}(\mathbf{x}) = \mathbf{q}^T(\mathbf{x}^i - \mathbf{x})\mathbf{Q}^{-T}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{q}^T(\mathbf{x}^i - \mathbf{x}) \boldsymbol{\alpha}(\mathbf{x}) \quad (43)$$

where

$$\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{Q}^{-T}(\mathbf{x})\mathbf{a}(\mathbf{x}) \quad (44)$$

with

$$\mathbf{Q}^{-T}(\mathbf{x}) = \begin{pmatrix} 1 & x_1 & x_2 & x_1 \cdot x_2 \\ 0 & 1 & 0 & x_2 \\ 0 & 0 & 1 & x_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (45)$$

Now, defining the functional

$$J(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=1}^n w_i(\mathbf{x}) (\mathbf{q}(\mathbf{x}_i - \mathbf{x}) \cdot \boldsymbol{\alpha}(\mathbf{x}) - u_i)^2 \quad (46)$$



and proceeding as described in the previous section, we can compute $\boldsymbol{\alpha}(\mathbf{x})$ from

$$\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \quad (47)$$

where

$$A_{jk}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \cdot q_j(\mathbf{x}_i - \mathbf{x}) \cdot q_k(\mathbf{x}_i - \mathbf{x}) \quad (48)$$

$$B_{ij}(\mathbf{x}) = w_i(\mathbf{x}) \cdot q_j(\mathbf{x}_i - \mathbf{x}) \quad (49)$$

Now, by computing the matrix-vector product (44) we obtain:

$$\begin{cases} \alpha_1 = a_1 + a_2 \cdot x_1 + a_3 \cdot x_2 + a_4 \cdot x_1 \cdot x_2 \\ \alpha_2 = a_2 + a_4 \cdot x_2 \\ \alpha_3 = a_3 + a_4 \cdot x_1 \\ \alpha_4 = a_4 \end{cases} \quad (50)$$

that corresponds to

$$\begin{cases} \alpha_1 = \mathbf{p}(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) \equiv u^h(\mathbf{x}) \\ \alpha_2 = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_1} \cdot \mathbf{a}(\mathbf{x}) \equiv \frac{\delta u^h(\mathbf{x})}{\delta x_1} \\ \alpha_3 = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_2} \cdot \mathbf{a}(\mathbf{x}) \equiv \frac{\delta u^h(\mathbf{x})}{\delta x_2} \\ \alpha_4 = \frac{\partial^2 \mathbf{p}(\mathbf{x})}{\partial x_1 \partial x_2} \cdot \mathbf{a}(\mathbf{x}) \equiv \frac{\delta^2 u^h(\mathbf{x})}{\delta x_1 \delta x_2} \end{cases} \quad (51)$$

Thus, by employing a centered basis, we have a direct access to the different diffuse derivatives that could be used within an optimization algorithm for computing the gradient vector and the Hessian matrix.

5.3. Numerical example

In order to illustrate the above presented procedure, we consider the geometry shown in figure 5 and the following material and process parameters: $v = 0.26$, $\rho = 1560$, $C = 1700$ and $k = 3.7 \cdot 10^{-7}$, all these values expressed in the metric system.

First, proceeding as indicated in the previous sections, we compute the solution for all possible boundary conditions:

$$u(x, y, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (52)$$



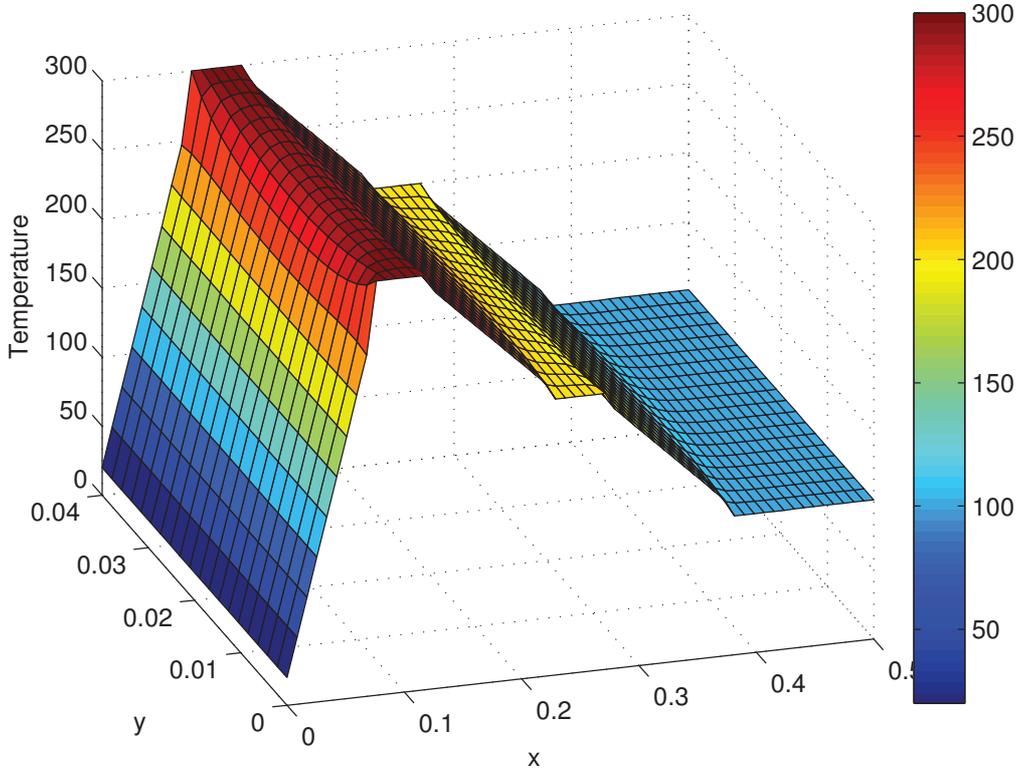


Figure 6: Multidimensional solution particularized for the heaters temperatures $\theta_1 = 300^\circ\text{C}$, $\theta_2 = 200^\circ\text{C}$ and $\theta_3 = 100^\circ\text{C}$.

This parametric solution can then be particularized for any choice of the temperatures of the heaters. One of the possible solutions related to a particular choice of the heaters temperatures is depicted in Figure 6.

If we compare the solution represented in Figure 6 to the one obtained by the finite element method for the same values of the heaters temperatures, the difference (still using the L_2 -norm) is found to be lower than 10^{-5} for N as little as 30 modes.

Now, we are considering the optimization process with respect the cost function given by Eq. (29), where the target temperature at the outflow is fixed to $\bar{u} = 150^\circ\text{C}$.

The temperature distribution in the whole die for these optimized values

is shown in Fig. 7, for the optimal temperatures of the three heaters:

$$\begin{aligned}\theta_1^{\text{opt}} &= 211.9^\circ\text{C} \\ \theta_2^{\text{opt}} &= 89.3^\circ\text{C} \\ \theta_3^{\text{opt}} &= 150.2^\circ\text{C}\end{aligned}\tag{53}$$

The value of the associated cost function when the three optimal temperatures were considered was of $3.1 \cdot 10^{-7}$.

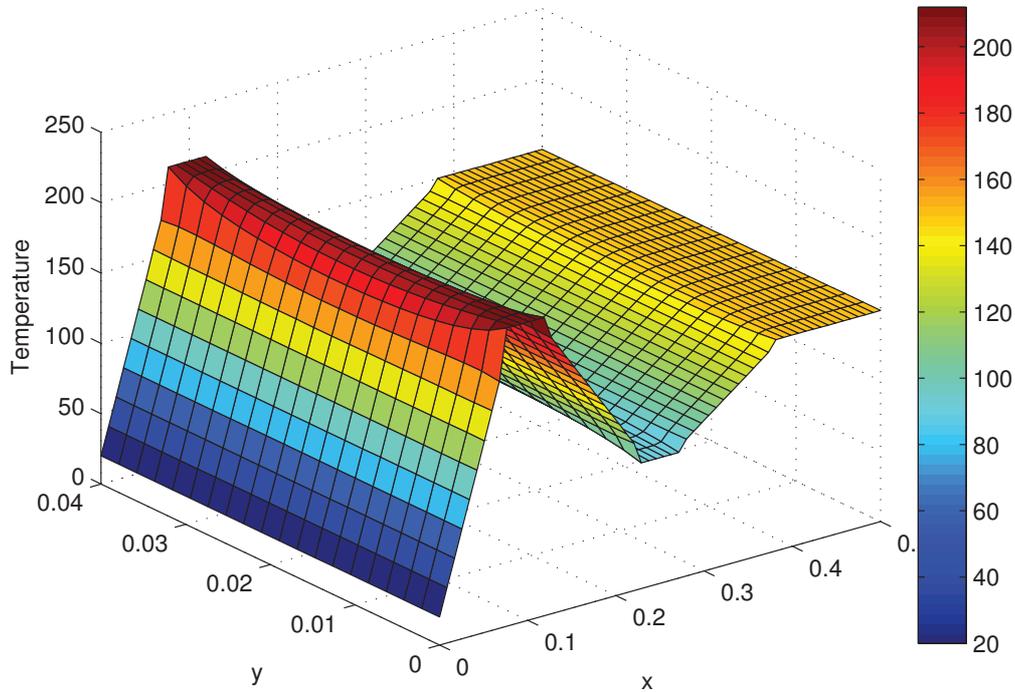


Figure 7: Final, optimized distribution of temperatures.

Remark 3. Note that the values taken in this academic example do not correspond to any physically meaningful ones. The obtained temperatures could have an impact on the resin viscosity and therefore in its thermal conductivity. We are here mostly interested in introducing the optimization strategy.

The technique here proposed could allow impressive computing efficiencies, however it is too early for quantifying the reachable computing time savings. The present technique only needs one solution of a multidimensional model, that thanks to the PGD features can be performed easily and very fast. As soon as this parametric solution is known we can imagine a panoply of optimization problems, involving different cost functions and numerical strategies, performed on-line without the necessity of solving the model again. The optimization described above was carried out by using Matlab and a standard laptop in 0.0015 seconds.

6. Conclusion

In this paper we have presented the abilities of the Proper Generalized Decomposition for solving a multidimensional problem in which parameters involved in optimization procedures are introduced as extra-coordinates in the model. Using the PGD the solution complexity scales linearly with the dimension of the space. The method seems to be able to deal with optimization problems whose deterministic solutions were a dream until now.

Advanced modeling of thermal processes taking place in a heated die, as encountered in pultrusion processes, is achieved by introducing the heating devices temperatures as extra-coordinates. This fact allows by solving once the resulting multidimensional model, to have access to the solution for any choice of the process parameters. For circumventing the resulting curse of dimensionality a PGD based discretization was applied, whose computational complexity scales linearly with the dimension of the space, instead of the exponential growing characteristic of mesh-based discretization strategies.

The potentialities of such procedure seem extraordinary, but at present it is too early to conclude on its advantages and its limitations. The coupling of parametric off-line solutions with most optimization strategies could be suitable because we only need to particularize the pre-computed parametric solution for any choice of the process parameters. Thus, optimization can be performed on-line and by using very light computing platforms, as for example smartphones.



A. Alternating directions separated representation constructor

A.1. Computing $R(\mathbf{x})$ from $S(t)$ and $W(k)$

We consider the global weak form of equation (2):

$$\int_{\Omega \times I \times \mathfrak{S}} u^* \left(\frac{\partial u}{\partial t} - k \Delta u - f \right) d\mathbf{x} dt dk = 0 \quad (54)$$

where the trial and test functions write respectively:

$$u(\mathbf{x}, t, k) = \sum_{i=1}^{i=n} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) + R(\mathbf{x}) \cdot S(t) \cdot W(k) \quad (55)$$

and, assuming S and W known from the previous iteration,

$$u^*(\mathbf{x}, t, k) = R^*(\mathbf{x}) \cdot S(t) \cdot W(k) \quad (56)$$

Introducing (55) and (56) into (54) it results:

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{S}} R^* \cdot S \cdot W \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) d\mathbf{x} dt dk = \\ & = - \int_{\Omega \times I \times \mathfrak{S}} R^* \cdot S \cdot W \cdot \mathcal{R}^n d\mathbf{x} dt dk \end{aligned} \quad (57)$$

where \mathcal{R}^n defines the residual at iteration n that reads:

$$\mathcal{R}^n = \sum_{i=1}^{i=n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{i=n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f \quad (58)$$

Once all functions involving time and conductivity have been determined, we can integrate Eq. (57) along its respective domains $I \times \mathfrak{S}$, and by taking into account the following notations:

$$\left[\begin{array}{ccc} w_1 = \int_{\mathfrak{S}} W^2 dk & s_1 = \int_I S^2 dt & r_1 = \int_{\Omega} R^2 d\mathbf{x} \\ w_2 = \int_{\mathfrak{S}} kW^2 dk & s_2 = \int_I S \cdot \frac{dS}{dt} dt & r_2 = \int_{\Omega} R \cdot \Delta R d\mathbf{x} \\ w_3 = \int_{\mathfrak{S}} W dk & s_3 = \int_I S dt & r_3 = \int_{\Omega} R d\mathbf{x} \\ w_4^i = \int_{\mathfrak{S}} W \cdot K_i dk & s_4^i = \int_I S \cdot \frac{dT_i}{dt} dt & r_4^i = \int_{\Omega} R \cdot \Delta X_i d\mathbf{x} \\ w_5^i = \int_{\mathfrak{S}} kW \cdot K_i dk & s_5^i = \int_I S \cdot T_i dt & r_5^i = \int_{\Omega} R \cdot X_i d\mathbf{x} \end{array} \right], \quad (59)$$



Eq. (57) is reduced to:

$$\begin{aligned} & \int_{\Omega} R^* \cdot (w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R) \, d\mathbf{x} = \\ & = - \int_{\Omega} R^* \cdot \left(\sum_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) \, d\mathbf{x} \end{aligned} \quad (60)$$

Eq. (60) defines an elliptic steady-state boundary value problem that can be solved by using any discretization technique operating on the weak form of the problem (finite elements, finite volumes, ...). Another possibility consists in coming back to the strong form of Eq. (60):

$$\begin{aligned} & w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R = \\ & = - \left(\sum_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) \end{aligned} \quad (61)$$

that could be solved by using any classical collocation technique (finite differences, SPH, ...).

A.2. Computing $S(t)$ from $R(\mathbf{x})$ and $W(k)$:

In the present case the test function is written as:

$$u^*(\mathbf{x}, t, k) = S^*(t) \cdot R(\mathbf{x}) \cdot W(k) \quad (62)$$

Now, the weak form becomes:

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{S}} S^* \cdot R \cdot W \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) \, d\mathbf{x} \, dt \, dk = \\ & = - \int_{\Omega \times I \times \mathfrak{S}} S^* \cdot R \cdot W \cdot \mathcal{R}^n \, d\mathbf{x} \, dt \, dk \end{aligned} \quad (63)$$

Integrating in the space $\Omega \times \mathfrak{S}$ and by taking into account the notation (59) gives:

$$\begin{aligned} & \int_I S^* \cdot \left(w_1 \cdot r_1 \cdot \frac{dS}{dt} - w_2 \cdot r_2 \cdot S \right) \, dt = \\ & = - \int_I S^* \cdot \left(\sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) \, dt \end{aligned} \quad (64)$$

Eq. (64) represents the weak form of the ODE defining the time evolution of the field S that can be solved by using any stabilized discretization technique (SU, Discontinuous Galerkin, ...). The strong form of Eq. (64) reads:

$$w_1 \cdot r_1 \cdot \frac{dS}{dt} - w_2 \cdot r_2 \cdot S =$$

$$= - \left(\sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) \quad (65)$$

Eq. (65) can be solved by using backward finite differences, or higher order Runge-Kutta schemes, among many other possibilities.

Computing $W(k)$ from $R(\mathbf{x})$ and $S(t)$:

In this part of the algorithm, the test function is written as:

$$u^*(\mathbf{x}, t, k) = W^*(k) \cdot R(\mathbf{x}) \cdot S(t) \quad (66)$$

Now, the weak form becomes:

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{S}} W^* \cdot R \cdot S \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) d\mathbf{x} dt dk = \\ & = - \int_{\Omega \times I \times \mathfrak{S}} W^* \cdot R \cdot S \cdot \mathcal{R}^n d\mathbf{x} dt dk \end{aligned} \quad (67)$$

Integrating the weak form, Eq. (67), in $\Omega \times I$ and considering the notations given by Eq. (59) leads to:

$$\begin{aligned} & \int_{\mathfrak{S}} W^* \cdot (r_1 \cdot s_2 \cdot W - r_2 \cdot s_1 \cdot k \cdot W) dk = \\ & = - \int_{\mathfrak{S}} W^* \cdot \left(\sum_{i=1}^{i=n} r_5^i \cdot s_4^i \cdot K_i - \sum_{i=1}^{i=n} r_4^i \cdot s_5^i \cdot k \cdot K_i - r_3 \cdot s_3 \cdot f \right) dk \end{aligned} \quad (68)$$

Equation (68) does not involve any differential operator. The strong form of Eq. (68) is:

$$(r_1 \cdot s_2 - r_2 \cdot s_1 \cdot k) \cdot W = - \left(\sum_{i=1}^{i=n} (r_5^i \cdot s_4^i - r_4^i \cdot s_5^i \cdot k) \cdot K_i - r_3 \cdot s_3 \cdot f \right) \quad (69)$$

Therefore, Equation (69) represents an algebraic equation. Despite the introduction of parameters as additional model coordinates, the change of computational cost will not be noticed. In fact, the original equation does not contain derivatives with respect to those parameters, so obviously solving this equation will not affect the total computation time in a dramatic way.

Note that other strategies for constructing the separated representation are available in the literature, leading to sometimes more robust and faster convergence [6] [14].



References

- [1] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. *J. Non Newtonian Fluid Mech.*, 139:153–176, 2006.
- [2] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A New Family Of Solvers For Some Classes Of Multidimensional Partial Differential Equations Encountered In Kinetic Theory Modelling Of Complex Fluids. Part II: Transient Simulation Using Space-Time Separated Representations. *J. Non Newtonian Fluid Mech.*, 144:98–121, 2007.
- [3] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1998.
- [4] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:1–123, 2004.
- [5] X. Chen, H. Xie, and H. Chen. Optimisation for CFRP Pultrusion Process Based on Genetic Algorithm-Neural Network. *International Journal of Material Forming*, pages 1–11, In press.
- [6] F. Chinesta, A. Ammar, and E. Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering*, in press, September 2009.
- [7] F. Chinesta, A. Ammar, and E. Cueto. Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering*, in press, 2010.
- [8] F. Chinesta, A. Ammar, and E. Cueto. Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering*, In press, 2010.
- [9] D. Gonzalez, A. Ammar, F. Chinesta, and E. Cueto. Recent advances on the use of separated representations. *International Journal of Numerical Methods in Engineering*, 85(5):637–659, 2010.



- [10] A. Jacob. Globalisation of the pultrusion industry. *Reinforced Plastics*, 50(5):38–41, 2006.
- [11] P. Ladeveze. *Nonlinear Computational Structural Mechanics*. Springer, N.Y., 1999.
- [12] J. Martin. Pultruded composites compete with traditional construction materials. *Reinforced Plastics*, 50(5):20–27, 2006.
- [13] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
- [14] A. Nouy. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1603–1626, 2010.
- [15] E. Pruliere, J. Ferec, F. Chinesta, and A. Ammar. An efficient reduced simulation of residual stresses in composites forming processes. *International Journal of Material Forming*, 3(2):1339–1350, 2010.



3.3 Mariage de la PGD avec la DDDAS

Traditionnellement, la simulation se basait sur des entrées statiques afin de réaliser les calculs. Cependant, un nouveau paradigme de calcul se base sur le dialogue entre la simulation et l'observateur du système, et est connu sous le nom de la DDDAS. Dans ce qui suit, nous proposons une approche d'optimisation et de contrôle en temps réel, basée sur la PGD. En effet, on calcule une solution paramétrique, puis on utilisera l'algorithme de Levenberg-Marquardt pour réaliser l'optimisation et surveiller le contrôle (Marquardt, 1963). Le calcul est séparé en deux étapes, "off-line" et "on-line". Le travail est résumé par le schéma de la figure 3.3.

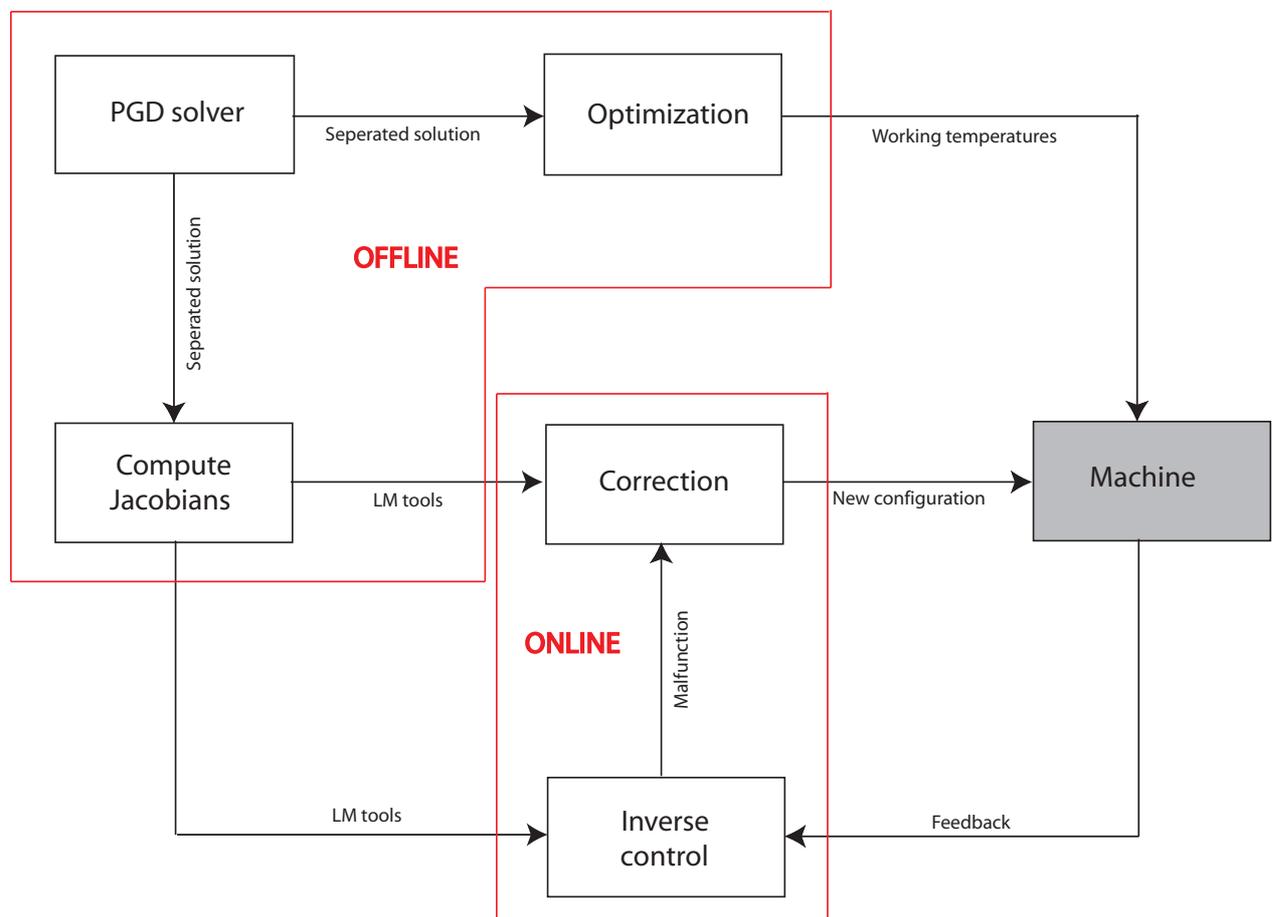


figure 3.1 – Schéma global illustrant le travail

Proper Generalized Decomposition Based Dynamic Data-Driven Control of Thermal Processes[★]

Ch. Ghnatios, F. Masson

*EADS Corporate Foundation International Chair
GEM, UMR CNRS-Centrale Nantes
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France*

A. Huerta

*Laboratori de Calcul Numeric
Universitat Politècnica de Catalunya
Jordi Girona, 1 Campus Nord, C2, E-08034 Barcelona, Spain
antonio.huerta@upc.es*

A. Leygue

*EADS Corporate Foundation International Chair
GEM, UMR CNRS-Centrale Nantes
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France*

E. Cueto

*Aragon Institute of Engineering Research
Universidad de Zaragoza
María de Luna, 7, E-50018 Zaragoza, Spain
ecueto@unizar.es*

F. Chinesta

*EADS Corporate Foundation International Chair
GEM, UMR CNRS-Centrale Nantes
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France
Francisco.Chinesta@ec-nantes.fr*

Abstract

Dynamic Data-Driven Application Systems —DDDAS— appear as a new paradigm in the field of applied sciences and engineering, and in particular in simulation-based engineering sciences. By DDDAS we mean a set of techniques that allow to



link simulation tools with measurement devices for real-time control of systems and processes. In this paper a novel simulation technique is developed with an eye towards its employ in the field of DDDAS. The main novelty of this technique relies in the consideration of parameters of the model as new dimensions in the parametric space. Such models often live in highly multidimensional spaces suffering the so-called curse of dimensionality. To avoid this problem related to mesh-based techniques, in this work an approach based upon the Proper Generalized Decomposition —PGD— is developed, which is able to circumvent the redoubtable curse of dimensionality. The approach thus developed is composed by a marriage of DDDAS concepts and a combination of PGD “off-line” computations, linked to “on-line” post-processing. In this work we explore some possibilities in the context of process control, malfunctioning identification and system reconfiguration in real time, showing the potentialities of the technique in real engineering contexts.

Keywords: Proper Generalized Decomposition; Dynamic Data-Driven Application Systems; Control; Real time; Parametric Models; Thermal processes

1 Introduction: Dynamic Data-Driven Application Systems (DDDAS)

Traditionally, Simulation-based Engineering Sciences (SBES) relied on the use of static data inputs to perform the simulations. These data could be parameters of the model(s) or boundary conditions, outputs at different time instants, etc., traditionally obtained through experiments. The word static is intended here to mean that these data could not be modified during the simulation.

A new paradigm in the field of Applied Sciences and Engineering has emerged in the last decade. Dynamic Data-Driven Application Systems (DDDAS) constitute nowadays one of the most challenging applications of SBES. By DDDAS we mean a set of techniques that allow the linkage of simulation tools with measurement devices for real-time control of simulations and applications. As defined by the U.S. National Science Foundation, “DDDAS entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process” [37].

The term Dynamic Data-Driven Application System was coined by F. Darema in a NSF workshop on the topic in 2000 [36]. The document that initially put forth this initiative stated that DDDAS constitute “application simulations that can dynamically accept and respond to ‘online’ field data and mea-

* This work has been partially supported by the Spanish Ministry of Science and Innovation, through grant number CICYT-DPI2011-27778-C02-01/02.



surements and/or control such measurements. This synergistic and symbiotic feedback control loop among applications, simulations, and measurements is a novel technical direction that can open new domains in the capabilities of simulations with a high potential pay-off, and create applications with new and enhanced capabilities. It has the potential to transform the way science and engineering are done, and induces a major beneficial impact in the way many functions in our society are conducted, such as manufacturing, commerce, transportation, hazard prediction/management, and medicine, to name a few” [14].

The importance of DDDAS in the forthcoming decades can be noticed from the NSF Blue Ribbon Panel on SBES report [33], that in 2006 included DDDAS as one of the five core issues or challenges in the field for the next decade (together with multiscale simulation, model validation and verification, handling large data and visualization). This panel concluded that “Dynamic data-driven application systems will rewrite the book on the validation and verification of computer predictions” and that “research is needed to effectively use and integrate data-intensive computing systems, ubiquitous sensors and high-resolution detectors, imaging devices, and other data-gathering storage and distribution devices, and to develop methodologies and theoretical frameworks for their integration into simulation systems” [33]. Moreover, the NSF believes that “... The DDDAS community needs to reach a critical mass both in terms of numbers of investigators, and in terms of the depth, breadth and maturity of constituent technologies...” [37].

A DDDAS includes different constituent blocks:

- (1) A set of (possibly) heterogeneous simulation models.
- (2) A system to handle data obtained from both static and dynamic sources.
- (3) Algorithms to efficiently predict system behaviour by solving the models under the restrictions set by the data.
- (4) Software infrastructure to integrate the data, model predictions, control algorithms, etc.

Almost a decade after the establishment of the concept, the importance of the challenge is better appreciated. As can be noticed, it deals with very different and transversal disciplines: from simulation techniques, numerical issues, control, modelling, software engineering, data management and telecommunications, among others. The three different blocks of interactions concern: (i) the one between human systems and the simulation, (ii) the simulation interaction with the physical system and (iii) the simulation and the hardware/data infrastructure. Physical systems operate at very different time scales: from 10^{-20} Hz for cosmological systems to 10^{20} Hz for problems at the atomic scales. Humans, however, can be considered as a system operating at rates from 3 Hz to 500 Hz in haptic devices for instance to transmit realistic touch



sensations. A crucial aspect of DDDAS is that of real-time simulation. This means that the simulations must run at the same time (or faster) than data are collected. While this is not always true (as in weather forecasting, for instance, where collected data are usually incorporated to the simulations after long time periods), most applications require different forms of real-time simulations. In haptic surgery simulators, for instance, the simulation result, i.e., forces acting on the surgical tool, must be translated to the peripheral device at a rate of 500 Hz, which is the frequency of the free hand oscillation. In other applications, such as some manufacturing processes, the time scales are much bigger, and therefore real-time simulations can last for seconds or minutes.

As can be noticed from the introduction above, DDDAS can revolutionize the way in which simulation will be done in the next decades. No longer a single run of a simulation will be considered as a way of validating a design on the basis of a static data set [33].

While research on DDDAS should involve applications, mathematical and statistical algorithms, measurement systems, and computer systems software methods, see for instance [16] [17] [21] [29] [28], our work focuses on the development of mathematical and statistical algorithms for the simulation within the framework of such a system. In brief, we intend to incorporate a new generation of simulation techniques into the field, allowing to perform faster simulations, able to cope with uncertainty, multiscale phenomena, inverse problems and many other features that will be discussed. This new generation of simulation techniques has received the name of Proper Generalized Decomposition—PGD—and has received an increasing level of attention by the SBES community. PGD was initially introduced for addressing multidimensional models encountered in science and engineering (see [1,2] and the references therein) and was then extended to address general computational mechanics models [10]. We are revisiting the motivation and the key ideas of such technique in the next sections.

1.1 When the solution of many direct problems is needed

An important issue encountered in DDDAS, related to process control and optimization, inverse analysis, etc., lies in the necessity of solving many direct problems. Thus, for example, process optimization implies the definition of a cost function and the search of optimum process parameters, which minimize the cost function. In most engineering optimization problems the solution of the model is the most expensive step. Real-time computations with zero-order optimization techniques can not be envisioned except for very particular cases. The computation of sensitivity matrices and adjoint approaches also hampers fast computations. Moreover, global minima are only ensured under severe



conditions, which are not (or cannot be) verified in problems of engineering interest. There are many strategies for updating the set of design parameters and the interested reader can find most of them in books focusing on optimization procedures. Our interest here is not the discussion on optimization strategies, but pointing out that standard optimization strategies need numerous direct solutions of the problem that represents the process, one solution for each tentative choice of the process parameters, plus those required for sensitivity.

As we discussed in the previous paragraphs, the solution of the model is a tricky task that demands important computational resources and usually implies extremely large computing times. Usual optimization procedures are inapplicable under real-time constraints because they need numerous solutions. The same issues are encountered when dealing with inverse analysis in which material or process parameters are expected to be identified from numerical simulation, by looking for the unknown parameters such that the computed fields agree in minute with the ones measured experimentally. However, some previous references exist on the treatment of problems that require extensive solution procedures for different parameter values. The interested reader can consult, for instance [20] [6] [7].

1.2 Towards generalized parametric modelling

One possibility for solving many problems very fast consists of using some kind of model order reduction based on the use of reduced bases [18][34]. In these works authors proved the capabilities of performing real time simulation even using light-computing devices, as smartphones for example. The tricky point in such approaches is the construction of such reduced bases and the way of adapting them when the system explores regions far from the ones considered in the construction of the reduced model. Solutions to this issue exist and others are been developed to fulfil with real time requirements.

Multidimensionality offers an alternative getaway to avoid too many direct solutions. In our opinion it could represent a new paradigm in computational mechanics. For the sake of clarity, the use of multidimensional modelling in an academic physical problem is illustrated and motivated.

Imagine for example that we are interested in solving the heat equation but the material's thermal conductivity is not known, because it has a stochastic nature or simply because prior to solve the thermal model it is necessary to measure it experimentally. Three possibilities arise: (i) wait to know the conductivity before solving the heat equation (a conservative solution); (ii) solve the equation for many values of the conductivity (a sort of Monte Carlo



method); or (iii) solve the heat equation only once for any value of the conductivity.

Obviously the third alternative is the most appealing one. To compute this quite general solution it suffices to introduce the conductivity as an extra independent coordinate, taking values in a certain interval and playing a similar role as standard space and time coordinates. Note, that there is no need to have derivatives involving this extra-coordinate. Thus, by solving only once the resulting multidimensional thermal model, the most general solution is computed; that is, a solution that produces at each physical point and instant the value of the temperature for any value of the thermal conductivity.

This procedure works very well, as will be proven later. Note that it can be extended to introduce many other extra-coordinates: source term, initial conditions, boundary conditions and even the domain geometry. Thus, moving loads in structural mechanics, geometrical parameters in shape optimization, material parameters in material characterization, boundary conditions in inverse analysis or process optimization, etc., can be treated as extra-coordinates to compute *off-line* multidimensional parametric solutions that could then be used *on-line*, running in real time. These general solutions computed off-line could be introduced in very light computing devices, as for example smartphones, opening an unimaginable field of applications that Figure 1 caricatures. This methodology constitutes in our opinion a new paradigm of real-time simulation.

The procedure outlined above introduces a major difficulty. If unknown parameters are considered as new dimensions of the problem, we will rapidly attain a high-dimensional model. Problems defined in high dimensional spaces are well known in the literature for the so-called *curse of dimensionality* [24]. This curse is related to the exponential growth of the number of degrees of freedom if such a multidimensional model is discretized by standard mesh-based discretization techniques.

To illustrate the technique proposed here to circumvent this curse of dimensionality, consider a problem defined in a space of dimension d for the unknown field $u(x_1, \dots, x_d)$. Here, the coordinates x_i denote any coordinate (scalar or vectorial) related to physical space, time or any model parameter. We thus seek a solution for each $(x_1, \dots, x_d) \in \Omega_1 \times \dots \times \Omega_d$ in separated forms:

$$u(x_1, \dots, x_d) \approx \sum_{i=1}^N F_i^1(x_1) \cdots F_i^d(x_d). \quad (1)$$

The Proper Generalized Decomposition (PGD) (the interested reader can find a complete state of the art with all up to now published papers in [13]) consists in the construction of a separated representation of the model solution from N functional products involving each a number d of functions $F_i^j(x_j)$, $j =$



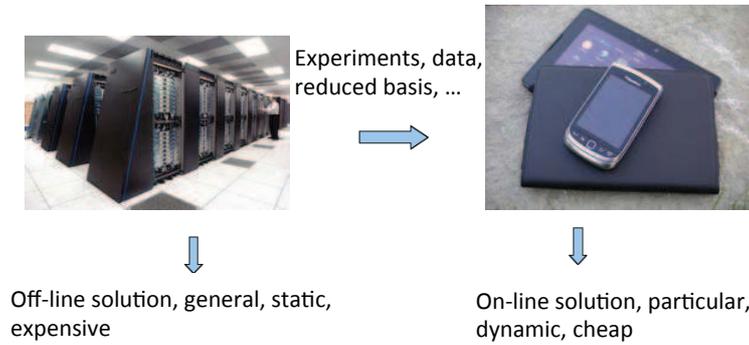


Figure 1. “Off-line” solution of a general enough parametric model and “on-line” particularization of such a general solution in a particular context. Source: <http://es.wikipedia.org/wiki/Archivo:UPM-CeSViMa-SupercomputadorMagerit.jpg>

$1, \dots, d$, that are unknown *a priori*. This approximation is constructed by successive enrichment up to a prescribed accuracy, whereby each functional product is determined in sequence. At a particular enrichment step $n + 1$, the functions $F_i^j(x_j)$ are known for $i \leq n$ from the previous steps, and one must compute the new product involving the d unknown functions $F_{n+1}^j(x_j)$, $j = 1, \dots, d$. This is achieved by invoking the weak form of the problem under consideration. The resulting formulation is non-linear, which implies that iterations are needed at each enrichment step. By using a fixed-point alternating-directions strategy a low-dimensional problem can thus be defined in Ω_j for each of the d functions $F_{n+1}^j(x_j)$.

If M degrees of freedom are used to discretize each coordinate, the total number of PGD unknowns is $N \cdot M \cdot d$ instead of the M^d degrees of freedom involved in standard mesh-based discretizations. Moreover, all numerical experiments carried out to date with the PGD show that the number of terms N required to obtain an accurate solution is not a function of the problem dimension d , but rather depends on the regularity of the solution, very much like Singular Value Decomposition methods. The PGD thus avoids the exponential complexity with respect to the problem dimension.

In many applications studied to date, N is found to be as small as a few tens. Moreover, the approximation converges towards the solution associated with the complete tensor product of the approximation bases considered for each Ω_j . Thus, we can be confident on the generality of the separated representation (1). When an exact solution of a particular problem can be represented with enough accuracy by a reduced number of functional products, the PGD approximation is suitable. If the solution is a strictly non-separable function, the PGD solver proceeds to enrich the approximation until including all the elements of the functional space, i.e. the M^d functions involved in the full tensor product of the approximation bases considered for each Ω_j .

The foundations and some recent advances in the proper generalized decomposition based discretization strategies can be found in [1]-[12], [19]-[32], [35] and the references therein.

1.3 *Paper outline*

Section 2 illustrates the construction of a separated, PGD, representation of the solution of a parametric boundary valued problem. The off-line solution construction procedure is analysed and thoroughly described. Then, Section 3 simulates a breakdown scenario that implies (i) identifying the malfunctioning device and (ii) the process reconfiguration. Both tasks should be performed as fast as possible and using the lightest possible computational resources (e.g. a smartphone, for instance) for real-time decision making. This is particularly crucial when the device is operated by personnel without a technical background. These on-line calculations on light computing platforms are described in Section 4.

2 Proper Generalized Decomposition for a parametric model of a material flowing in a heated die

In this section the main ideas related to casting the model into a multidimensional framework, followed by process optimization, are introduced. For the sake of clarity in what follows we consider the thermal model related to a material flowing into a heated die. Despite the apparent simplicity, the strategy here described can be extended to address more complex scenarios.

The 2D thermal process is sketched in Figure 2. The material flows with a velocity \mathbf{v} inside a die Ω of length L and width H . The temperature of the material at the die entrance is u_0 . The die is equipped with two heating devices of lengths L_1 and L_2 respectively, whose temperatures θ_1 and θ_2 respectively,



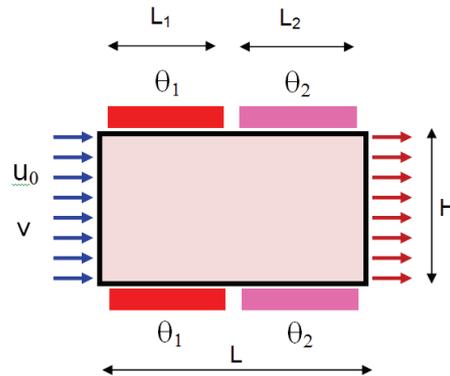


Figure 2. Thermal process consisting of two heating devices located on the die walls where the temperature is enforced to the values θ_1 and θ_2 respectively.

can range within an interval $[\theta_{min}, \theta_{max}]$.

The steady state temperature field $u(\mathbf{x})$ in any point of the die $\mathbf{x} = (x, y) \in \Omega \subset \mathbb{R}^2$ can be obtained from the solution of the 2D heat transfer equation that involves advection and diffusion mechanics as well as a possible source term Q . The velocity field is everywhere unidirectional, i.e. $\mathbf{v}^T = (0, v)$. The steady-state heat transfer equation thus reduces to:

$$\rho c \left(v \frac{\partial u}{\partial x} \right) = k \Delta u + Q, \quad (2)$$

where k is the thermal conductivity, ρ is the density and c is the specific heat.

2.1 Building-up the parametric solution in the framework of a multidimensional model

The die is equipped with two heating devices as depicted in Figure 2 whose temperatures constitute the process parameters to be optimized and, eventually, controlled. For the sake of simplicity the internal heat generation Q is assumed constant, as well as the velocity v and the inlet temperature u_0 .

Different values of prescribed temperatures at both heating devices can be considered. The resulting 2D heat transfer equation can be then solved. As

noted earlier, optimization or inverse identification will require many direct solutions or, as named in the introduction, static data computations. Obviously, when the number of the process parameters involved in the model is increased, standard approaches fail to compute optimal solutions in a reasonable time. Thus, for a large number of process parameters, real-time computations are precluded and, moreover, performing “on-line” optimization or inverse analysis is a challenging issue.

The method proposed here consists on introducing both process parameters, i.e. temperatures of the heating devices, θ_1 and θ_2 , as extra coordinates.

Remark 1 *If some of these desired extra-coordinates are fields depending on other coordinates instead independent parameters (temperatures of the heating devices evolving in time, source term evolving in time and/or space...) prior to introduce them as extra-coordinates one should parametrize such evolutions in an appropriate manner, and finally introduce the coefficients involved in those parametrizations as extra-coordinates as considered in [25].*

Other parameters such as ρ , c , v , k , \dots , can be set as extra-coordinates as well. The temperature field can be thus computed at each point and for any possible value of the temperatures θ_1 and θ_2 . As soon as this multidimensional solution $u(x, y, \theta_1, \theta_2)$ is available, it is possible to particularize it for any value of the process parameters without the necessity of further executions of the code. Thus, optimization procedures can proceed from the only knowledge of an “off-line” pre-computed parametric solution.

To circumvent the curse of dimensionality related to the high dimensional space in which the temperature field $u(x, y, \theta_1, \theta_2)$ is defined—which we retain to be four-dimensional for the ease of exposition—we consider a separated representation of that field:

$$u(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^N F_i(x, y) \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) \quad (3)$$

where all the functions involved in such separated representation are computed by applying the Proper Generalized Decomposition technique, described below.

Remark 2 *Because the geometrical simplicity of Ω that can be written as $\Omega = \Omega_x \times \Omega_y$, we could consider a fully separated representation of the unknown field that now writes:*

$$u(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^N X_i(x) \cdot Y_i(y) \cdot \Theta_i^1(\theta_1) \cdot \Theta_i^2(\theta_2) \quad (4)$$

This fully separated representation can be also applied in complex domains as



proved in [19], however when the physical domain is complex the most natural representation is the one given in Eq. (3).

The prescribed essential boundary conditions write:

$$\begin{cases} u(x = 0, y, \theta_1, \theta_2) = u_0, \\ u(x \in I_1, y = 0 \text{ or } y = H, \theta_1, \theta_2) = \theta_1, \\ u(x \in I_2, y = 0 \text{ or } y = H, \theta_1, \theta_2) = \theta_2, \\ \frac{\partial u}{\partial x}(x = L, y, \theta_1, \theta_2) = 0, \end{cases} \quad (5)$$

where I_1 and I_2 are the intervals of the x coordinate where the heating devices of length L_1 and L_2 respectively are defined. A null heat flux is assumed in the remaining part of the domain boundary. Thus, the temperature field u depends on four different coordinates, the two space coordinates (x, y) and the two temperatures prescribed in both regions on the die wall. Parameters θ_1 and θ_2 now take values in the intervals \mathcal{I}_1 and \mathcal{I}_2 respectively.

The case of essential (Dirichlet) boundary conditions as parameters of the model deserves some comments. Non-homogeneous essential boundary conditions in PGD methods are usually treated by means of a simple change of variable

$$u = \psi + z, \quad (6)$$

where ψ is a function verifying essential boundary conditions. This leads to a problem in the z variable with homogeneous boundary conditions. Efficient construction of ψ functions in the framework of PGD approximations has been deeply analysed in [19]. We refer the interested reader to this reference for further details. This function ψ can, for the problems addressed here, be expressed in separated form

$$\psi(x, y, \theta_1, \theta_2) = \sum_{i=1}^3 F_i(x, y) \Theta_i^1(\theta_1) \Theta_i^2(\theta_2). \quad (7)$$

where each functional product is used to impose initial conditions and the two non-homogeneous essential boundary conditions, one for each heater position. These functions are depicted in Fig. 3

The resulting PGD approximation reads

$$u(x, y, \theta_1, \theta_2) \approx \psi(x, y, \theta_1, \theta_2) + \sum_{i=4}^N F_i(x, y) \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) \quad (8)$$

Functions F_i , Θ_i^1 and Θ_i^2 , $i = 4, \dots, N$, are determined by solving a sequence of non-linear problems with homogeneous boundary conditions, as described below. Assuming that the first n functional products of Eq. (8) have already



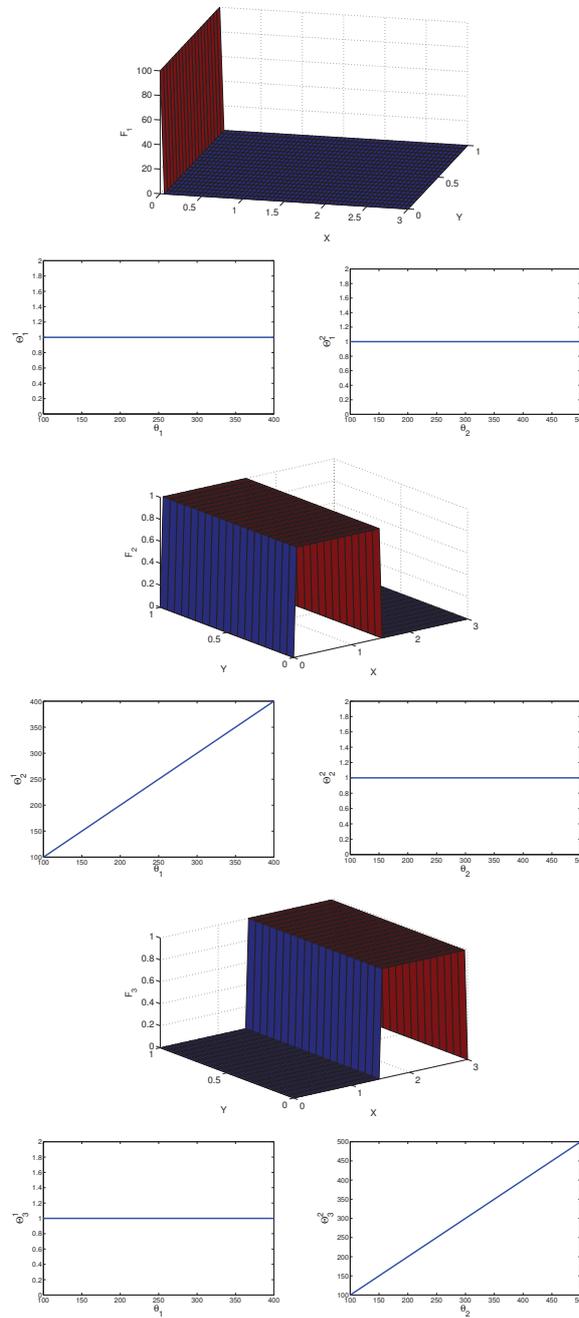


Figure 3. Functions $F_i(x, y)$, $\Theta_i^1(\theta_1)$ and $\Theta_i^2(\theta_2)$ used to approximate initial and boundary conditions.

been computed, we look for the $n + 1$ term:

$$u^{n+1}(x, y, \theta_1, \theta_2) = \sum_{i=1}^n F_i(x, y) \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) + R(x, y) S(\theta_1) T(\theta_2) \quad (9)$$

or, equivalently,

$$u^{n+1}(x, y, \theta_1, \theta_2) = u^n(x, y, \theta_1, \theta_2) + R(x, y) S(\theta_1) T(\theta_2). \quad (10)$$

Note that in the weak form associated to (2) the unknown functions R , S , and T are determined and the test function are

$$u^*(x, y, \theta_1, \theta_2) = R^*(x, y) S(\theta_1) T(\theta_2) + R(x, y) S^*(\theta_1) T(\theta_2) + R(x, y) S(\theta_1) T^*(\theta_2). \quad (11)$$

This approach allows us to determine the unknown functions $R(x, y)$, $S(\theta_1)$ and $T(\theta_2)$ in an alternating directions fixed-point algorithm. In fact, we proceed by determining sequentially each one of these functions, as described below, until reaching convergence. For more details we refer the interested reader to [10].

2.1.1 Determining $R(x, y)$ assuming $S(\theta_1)$ and $T(\theta_2)$ known

First, the weak form associated to equation (2) and boundary conditions defined in (5) must be determined. It reads

$$\int_{\Omega \times \mathcal{I}_1 \times \mathcal{I}_2} \left(u^* \rho c \left(v \frac{\partial u}{\partial x} \right) + \nabla u^* k \nabla u \right) d\Omega d\theta_1 d\theta_2 = \int_{\Omega \times \mathcal{I}_1 \times \mathcal{I}_2} u^* Q d\Omega d\theta_1 d\theta_2, \quad (12)$$

where the integration by parts, to reduce the order of derivation, has only been used for the spatial domain, Ω . Then, since an alternating direction approach is used to solve this problem, see the definition of the test function in (11), and inasmuch as S and T are assumed known, (11) reduces to:

$$u^*(x, y, \theta_1, \theta_2) = R^*(x, y) S(\theta_1) T(\theta_2)$$

For the sake of clarity, in what follows functional dependencies of each function are omitted. Approximation u^{n+1} , see (10), is substituted in (12). Thus, only R , S , and T must be determined. Now, as noted earlier, S and T are assumed known, and, consequently the unknown is R and the test function, defined above, induces the following problem: find R for all R^* (in the appropriate functional spaces) such that

$$\int_{\Omega \times \mathcal{I}_1 \times \mathcal{I}_2} S^2 T^2 \left(\rho c v R^* \frac{\partial R}{\partial x} + k \nabla R^* \nabla R \right) d\Omega d\theta_1 d\theta_2 = r(R^* S T, u^n), \quad (13)$$

where $r(\cdot, \cdot)$ the weak residual, see (12), evaluated for $u^* = R^* S T$ at iteration n , i.e. $u = u^n$,

$$r(u^*, u) = \int_{\Omega \times \mathcal{I}_1 \times \mathcal{I}_2} \left(u^* \rho c \left(v \frac{\partial u}{\partial x} \right) + \nabla u^* k \nabla u - u^* Q \right) d\Omega d\theta_1 d\theta_2.$$



Since functions involving the parametric coordinates θ_1 and θ_2 are assumed known in the present step, integrals in the parametric domain $\mathcal{I}_1 \times \mathcal{I}_2$ can be calculated, leading to a problem defined in the two-dimensional space domain Ω whose solution is precisely the searched unknown function $R(x, y)$. This domain can be discretized, of course, by standard finite elements. Note however, that the resulting problem is a convection-diffusion equation along the x coordinate, which eventually may need some form of stabilization.

2.1.2 Determining $S(\theta_1)$ assuming $R(x, y)$ and $T(\theta_2)$ known

In this case, the test function reads

$$u^*(x, y, \theta_1, \theta_2) = R(x, y) S^*(\theta_1) T(\theta_2),$$

and the weak problem, see (12), becomes: find S for all S^* (in the appropriate functional spaces) such that

$$\int_{\Omega \times \mathcal{I}_1 \times \mathcal{I}_2} T^2 \left(\rho c v R \frac{\partial R}{\partial x} + k \nabla R \nabla R \right) S^* S \, d\Omega d\theta_1 d\theta_2 = r(R S^* T, u^n), \quad (14)$$

Again, integrations involving the domain $\Omega \times \mathcal{I}_2$ can be performed, leading to a one-dimensional algebraic problem defined in \mathcal{I}_1 , because the model does not contain derivatives with respect to the extra-coordinate θ_1 . Its solution allows computing the function $S(\theta_1)$.

The calculation of function $T(\theta_2)$ can be performed in a similar way.

Remark 3 *In the present case we assumed a linear model, however many problems encountered in science and engineering are in fact non-linear. For example the heat equation here addressed becomes non linear as soon as dependences of thermal parameters on the temperature field are taken into account.*

In the non-linear case additional linearisation techniques are required. Application of standard linearisation strategies within the PGD framework were addressed in [3] [35]

A better way to address strongly non-linear models consists in using the LATIN technique [22]. However its extension to the case of multidimensional parametric modelling is not straightforward.

2.2 “Off-line” optimization procedure

Optimization procedures look for optimal parameters minimizing an appropriate single or multi objective cost function (sometimes subjected to many



constraints). In this work we consider a simple scenario, in which the cost function only involves the coldest thermal history of an imaginary material particle traversing the die, it is expressed as:

$$\mathcal{C}(\theta_1, \theta_2) = \frac{1}{2} \left(\int_0^L u\left(x, \frac{H}{2}, \theta_1, \theta_2\right) dx - \beta \right)^2, \quad (15)$$

where β denotes the optimal value of the thermal history able to ensure a certain material transformation. Values lower than β imply that the material has not received the necessary amount of heat, whereas values higher than β imply an unnecessary extra-heating.

Now, optimal process parameters θ_1^{opt} and θ_2^{opt} must be calculated by minimizing the cost function. There exist many techniques for such minimization. The interested reader can refer to any book on optimization. Many of them proceed by evaluating the gradient of the cost function and then moving on that direction. The gradient computation involves the necessity of performing first derivatives of the cost function with respect to the process parameters. Other techniques involve the calculation of second derivatives. To this end, one should calculate the derivatives of the problem solution with respect to the optimization parameters.

It is important to note that separated representations of the process parameters drastically simplifies this task because as the solution depends explicitly on the parameters its derivation is straightforward, namely,

$$\frac{\partial u}{\partial \theta_1}(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^N F_i(x, y) \frac{\partial \Theta_i^1}{\partial \theta_1}(\theta_1) \Theta_i^2(\theta_2),$$

and

$$\frac{\partial u}{\partial \theta_2}(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^N F_i(x, y) \Theta_i^1(\theta_1) \frac{\partial \Theta_i^2}{\partial \theta_2}(\theta_2).$$

Note that second derivatives are also similarly obtained. The calculation of the solution derivatives is a tricky point when proceeding from standard discretization techniques because the parametric dependency of the solution is, in general, not explicit.

Moreover, the separated rank- N representation of the solution, see (3), further simplifies the expression of the cost function. That is, substituting (3) in (15), induces

$$\mathcal{C}(\theta_1, \theta_2) = \frac{1}{2} \left(\sum_{i=1}^N \alpha_i \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) - \beta \right)^2 \quad (16)$$

where $\alpha_i = \int_0^L F_i(x, \frac{H}{2}) dx$, and the different derivatives of the cost function



becomes:

$$\begin{cases} \frac{\partial \mathcal{C}}{\partial \theta_1}(\theta_1, \theta_2) = \left(\sum_{i=1}^N \alpha_i \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) - \beta \right) \left(\sum_{i=1}^N \alpha_i \frac{\partial \Theta_i^1}{\partial \theta_1}(\theta_1) \Theta_i^2(\theta_2) \right), \\ \frac{\partial \mathcal{C}}{\partial \theta_2}(\theta_1, \theta_2) = \left(\sum_{i=1}^N \alpha_i \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) - \beta \right) \left(\sum_{i=1}^N \alpha_i \Theta_i^1(\theta_1) \frac{\partial \Theta_i^2}{\partial \theta_2}(\theta_2) \right). \end{cases}$$

In the simulations carried out in what follows, the minimization of the cost function was performed by using a Levenberg-Marquardt algorithm, see [15] for further details.

2.3 Defining the process control

Once the optimal parameters θ_1^{opt} and θ_2^{opt} are determined, the general solution, see (3), can be evaluated for those optimal values of the process parameters, namely

$$u(x, y, \theta_1^{\text{opt}}, \theta_2^{\text{opt}}) \approx \sum_{i=1}^N F_i(x, y) \Theta_i^1(\theta_1^{\text{opt}}) \Theta_i^2(\theta_2^{\text{opt}})$$

to obtain the temperature field everywhere in the domain Ω .

For the sake of clarity, in what follows, we illustrate the dynamic data driven thermal model through a numerical example without physical relevance. We consider the thermal model described above, see Figure 2, with the following values of the different parameters (units are omitted because all are defined in the metric system): $\rho = 1$, $c = 1$, $k = 1$, $Q = 50$, $v = 1$, $u_0 = 100$, $L = 3$, and $H = 1$. The first heating device acts in the interval $I_1 = [0.2, 1.4]$ whereas the second one is defined by $I_2 = [1.6, 2.8]$, both having the same length $L_1 = L_2 = 1.2$.

To solve the parametric model one needs to approximate the functions $F_i(\mathbf{x})$ involved in the solution's separated representation (3), as well as functions $\Theta_i^1(\theta_1)$ and $\Theta_i^2(\theta_2)$. Space functions are approximated by using a finite element mesh composed of 4-nodes quadrilateral elements, on a uniform nodal distribution composed of 60×20 nodes in the x and y direction respectively. Functions depending on the process parameters θ_1 and θ_2 are approximated by two uniform 1D linear finite meshes (300 nodes each) over the interval of variation of these parameters ($\mathcal{I}_1 = [100, 400]$ and $\mathcal{I}_2 = [100, 500]$).

The parametric solution $u(\mathbf{x}, \theta_1, \theta_2)$ is computed by using the Proper Generalized Decomposition strategy illustrated in Section 2.1. This solution implies 42 functional products in the sum, that is $N = 42$ in Eq. (3). From this general solution we compute the optimal process parameters θ_1^{opt} and θ_2^{opt} with respect

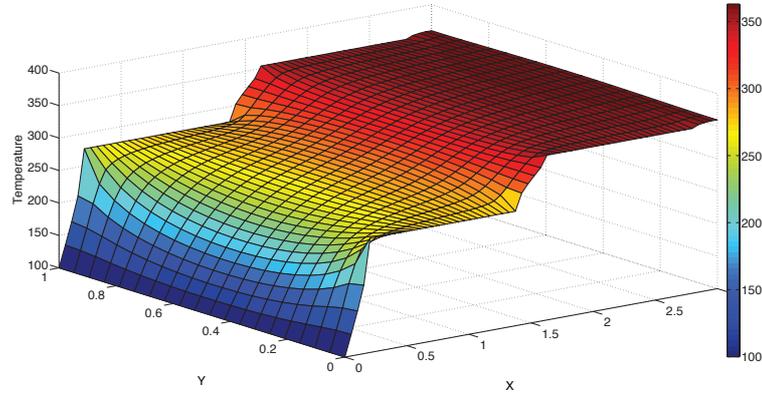


Figure 4. Multidimensional solution particularized for the optimal temperature of both heating devices: $u(x, y, \theta_1^{\text{opt}} = 275.8, \theta_2^{\text{opt}} = 353.4)$

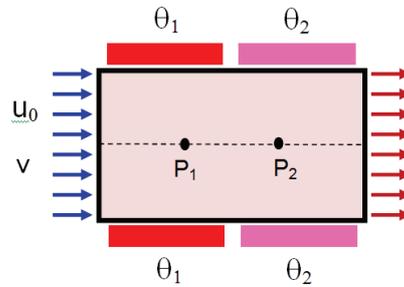


Figure 5. Location of the thermocouples \mathbf{P}_1 and \mathbf{P}_2 whose measurements serve for controlling the process, identify malfunctioning devices and reconfigure the system after a breakdown.

to the cost function introduced in (15) where $\beta = 897$. The convergence of the Levenberg-Marquardt algorithm is reached in only 4 iterations, being the optimal values $\theta_1^{\text{opt}} = 275.8$ and $\theta_2^{\text{opt}} = 353.4$. Figure 4 depicts the resulting temperature field related to the optimal process parameters.

Remark 4 *Up to now, the convergence criterion that we use concerns the norm of the residuals. We have considered in our former works more sophisticated error estimators, as the one based on quantities of interest [4].*

Error estimators based on quantities of interest are preferred because they are more adapted to the outputs considered in control strategies. However, at present we have not extended these estimators to the multidimensional parametric modelling. Thus, prior to consider any parametric solution computed off-line in on-line procedures we should proceed to verification and validation.

In order to control the process we could imagine two thermocouples located at two points on the axis of symmetry, namely $\mathbf{P}_1 = (1, 0.5)$ and $\mathbf{P}_2 = (2, 0.5)$, see also Figure 5. Obviously, for negligible validation and verification errors, the “on-line” measurements \tilde{u}_1 and \tilde{u}_2 will give values coincident, or almost, with the predictions of our model. These predictions are easily computed from the separate representation of the solution, i.e.

$$\begin{aligned} u^{\text{opt}}(\mathbf{P}_1) &= u(1, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{opt}}) = 286.6 \\ u^{\text{opt}}(\mathbf{P}_2) &= u(2, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{opt}}) = 352. \end{aligned}$$

Under these circumstances the process can be considered working in optimal conditions, the one related to the optimal temperatures of both heating devices.

However, if the heating devices do not run optimally the thermocouple measures \tilde{u}_1 and \tilde{u}_2 will differ from the predicted optimal conditions $u^{\text{opt}}(\mathbf{P}_1)$ and $u^{\text{opt}}(\mathbf{P}_2)$. Under these circumstances we could infer a breakdown in the system. Thus, checking the state of the system seems quite easy, however, the most important point is the identification of the malfunction and the reconfiguration of the system taking into account the real state of the system. These questions are addressed in the next section.

3 Simulating a breakdown scenario

In this section we simulate a breakdown scenario, whose solution requires both: (i) identifying the device malfunction and (ii) the process reconfiguration. Both tasks should be performed as fast as possible and using the lightest computational resources (e.g. a smartphone) in order to be able to do real-time decision making independently of the expertise of the personnel in charge of the process.

3.1 “On-line” inverse analysis

A possible scenario of a breakdown is a malfunction of the second heating device. For illustration purposes, we will assume that it only applies a fraction



of the desired temperature. That is, the device is only able to prescribe a temperature of $\theta_2^{\text{brk}} = 0.4\theta_2^{\text{opt}} = 141.4$, instead of the optimal one (the slight gap is due to the fact of considering few iterations in order to fulfill real time requirements). Under these circumstances, both thermocouples will indicate temperatures at \mathbf{P}_1 and \mathbf{P}_2 equal to

$$\begin{aligned} u^{\text{brk}}(\mathbf{P}_1) &= u(1, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{brk}}) = 254 \\ u^{\text{brk}}(\mathbf{P}_2) &= u(2, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{brk}}) = 165. \end{aligned} \quad (17)$$

These values are obtained from the representation of the general solution, see (3), because validation and verification errors are assumed negligible.

To reproduce the practical scenario, it is assumed that a decision must be taken with the only information of the thermocouple temperatures, namely (17). That is, it is not known which device is malfunctioning and, of course, that the real prescribed temperature is θ_2^{brk} . In fact, the only available data is (17) which clearly indicates that $u^{\text{opt}}(\mathbf{P}_1) \neq u^{\text{brk}}(\mathbf{P}_1)$ and $u^{\text{opt}}(\mathbf{P}_2) \neq u^{\text{brk}}(\mathbf{P}_2)$. Under these circumstances, the system should be reconfigured to ensure that the process continues working.

The first step is to determine which are the actual working temperatures. Thus, an inverse analysis is required. The following step, see the next subsection, will be to reconfigure the process in order to impose the desired thermal history, see (15). Obviously, both steps require a swift resolution if real-time decision are required. That is “on-line” computations are designed.

The “on-line” inverse analysis minimizes the following least-squares problem

$$\tilde{\mathcal{C}}(\theta_1, \theta_2) = \frac{1}{2} \sum_{i=1}^2 \left(u^{\text{brk}}(\mathbf{P}_i) - u(x_i, 0.5, \theta_1, \theta_2) \right)^2 \quad (18)$$

where x_i for $i = 1, 2$ are the coordinates of the points at which the thermocouples are located. The Levenberg-Marquardt algorithm, see [15], reaches convergence after three iterations and the estimated temperatures for both heating devices are $\theta_1^{\text{est}} = 261$ and $\theta_2^{\text{est}} = 146$, agree with the considered scenario ($\theta_1^{\text{opt}} = 275.8$, $\theta_2^{\text{brk}} = 141.4$). The inverse identification runs very fast and it only involves slight calculations, so it could be performed “on-line” on a very light computing devices, such as a smartphone.

3.2 “On-line” process reconfiguration

Finally, the process is reconfigured to impose the desired thermal history, see (15). Obviously, there are many possibilities and strategies. Here, since the second heating device is the one not giving the desired optimal heating. The



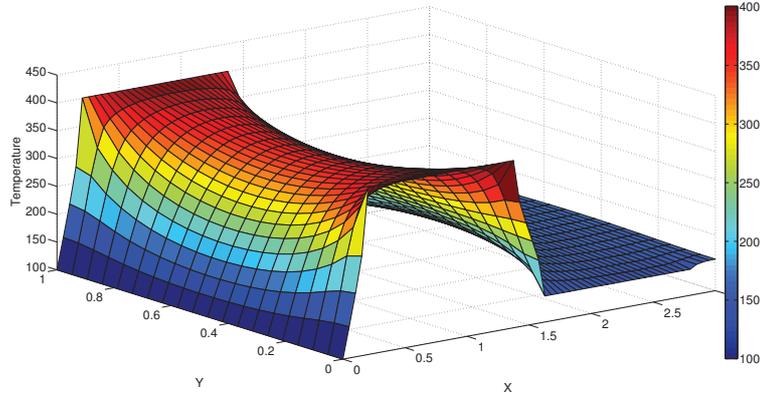


Figure 6. Multidimensional solution particularized for the optimal temperature after reconfiguring the system: $u(x, y, \theta_1^*, \theta_2^{\text{est}})$.

action consists in keeping the second heating device in its present state, i.e. $\theta_2^{\text{est}} = 146$ and looking for the optimal value of θ_1^* minimizing the cost function (15) for a fixed and known $\theta_2 = \theta_2^{\text{est}} = 146$, namely

$$\mathcal{C}(\theta_1) = \frac{1}{2} \left(\int_0^L u\left(x, \frac{H}{2}, \theta_1, \theta_2^{\text{est}}\right) dx - \beta \right)^2.$$

In three iterations of the Levenberg-Marquardt algorithm convergence is attained to the value $\theta_1^* = 400$. Figure 6 depicts the resulting temperature field related to the new optimized process parameters $\theta_1^* = 400$ and $\theta_2^{\text{est}} = 146$.

4 Parametric solution post-processing by using light computing platforms

As soon as the parametric solution $u(x, y, \theta_1, \theta_2)$ has been computed only once and off-line, the subsequent processes, i.e. optimization, control and system reconfiguration, only involve very light calculations that could be performed on-line and using mobile computing platforms, like smartphones.

To illustrate this capability, we assume that the parametric solution previously considered has already been computed and that it is available in a separated form. Now this solution can be introduced in a smartphone that will perform all the on-line calculations described in the previous sections. Only the most significative modes of the separated representation are retained in order to speed-up the computations. In our applications we considered a Nokia platform, with 256 MB of RAM, 16 GB of internal memory, a 680 MHz ARM 11 CPU and with Symbian3 as operating system.





Figure 7. Optimal temperatures of both heating devices.

Figures 7 to 13 illustrate all the scenarios analyzed in the previous sections. Figure 7 shows the output for the optimal values of both heating devices $\theta_1^{\text{opt}} = 276.82$ and $\theta_2^{\text{opt}} = 354.97$. They are slightly different from the ones previously computed because as just argued the parametric solution introduced into the smartphone has been restricted to the most significant modes of the off-line parametric solution to alleviate the on-line calculations performed by the smartphone.

The temperatures at the thermocouple positions $\mathbf{P}_1 = (1, 0.5)$ and $\mathbf{P}_2 = (2, 0.5)$ are then computed for the optimal temperatures of both heating devices. Figure 8 shows both temperatures $u^{\text{opt}}(\mathbf{P}_1) = 283.43$ and $u^{\text{opt}}(\mathbf{P}_2) = 353.12$.

A failure scenario is then simulated by considering that the temperatures at positions \mathbf{P}_1 and \mathbf{P}_2 are not the optimal ones. In fact we consider the scenario defined by $u^{\text{brk}}(\mathbf{P}_1) = 283.43$ and $u^{\text{brk}}(\mathbf{P}_2) = 224.60$ illustrated in Figure 9. The system identifies the malfunction and displays an alert message as shown in Figure 10. The system then identifies the real temperatures of both heating devices, by applying the inverse strategy previously discussed. The real temperatures of both heating devices are identified as $\theta_1^{\text{est}} = 297.98$ and $\theta_2^{\text{est}} = 205.97$ instead of the optimal ones $\theta_1^{\text{opt}} = 276.82$ and $\theta_2^{\text{opt}} = 354.97$ as depicted in Figure 11.

We can decide to reconfigure the system by choosing to change the temperature of one (or eventually both) heating devices. In our case we decide to calculate the new temperature of the first heating device able to ensure optimal process conditions. For this purpose we select the first heater as





Figure 8. Optimal temperatures at positions \mathbf{P}_1 and \mathbf{P}_2 when both heaters work at the optimal conditions.



Figure 9. Simulating a failure by considering a temperature at position \mathbf{P}_2 different of the optimal one.

shown in Figure 11. The system then recomputes on-line the optimal temperature of the first heating device: $\theta_1^* = 474.61$ as shown in Figure 12. The new temperature field corresponding to the new operational conditions $u(x, y, \theta_1^* = 474.61, \theta_2^{\text{est}} = 205.97)$ is illustrated in Figure 13.



Figure 10. Malfunction alert.



Figure 11. Identification of the real process parameters, i.e. the real temperatures of both heating devices.

5 Conclusions

This work presents a first attempt of applying dynamic data driven simulation for controlling industrial processes whose modelling involves complex linear or non-linear partial differential equations. For the sake of simplicity we addressed in the present work the linear case, the non-linear one constituting a work in progress.

The procedure that we propose combines heavy “off-line” calculations for solv-

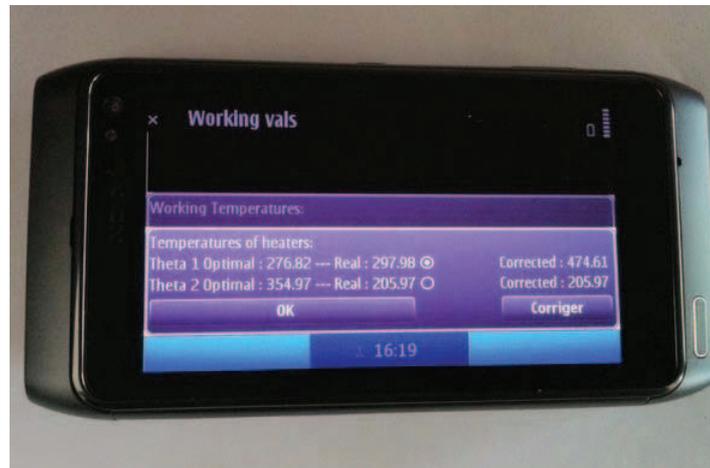


Figure 12. System reconfiguration by adjusting the temperature of the first heating device.

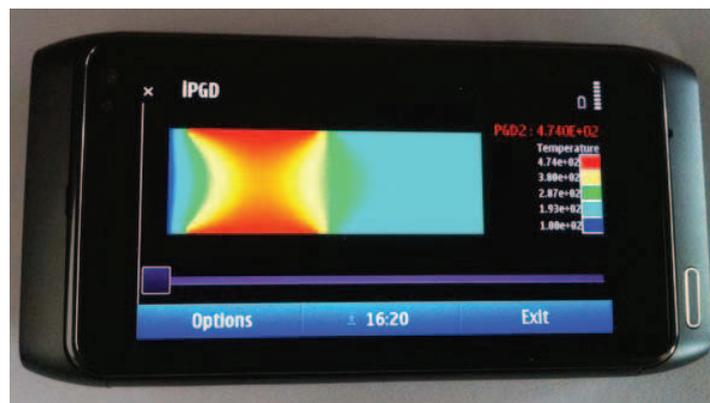


Figure 13. New optimal temperature field related to the reconfigured heating system.

ing the partial differential model associated to the industrial process by introducing all the sources of variability or process parameters as extra-coordinates. Thus, the resulting model becomes multidimensional making the use of the well experienced mesh-based discretization techniques impossible. However, the use of separated representations within the proper generalized decomposition framework allows us to circumvent the curse of dimensionality that multidimensional models suffer.

As soon as this parametric solution is available, one could proceed to optimize the process, still “off-line”, by calculating the optimal process parameters in order to minimize an appropriate cost function.

However, the system’s response faced to a breakdown should be computed “on-line” and as fast as possible. Separated representation of the solution built-up by applying the PGD-based solver allows computing explicitly the derivatives of the solution with respect to the process parameters, making

the fast calculation of minimization strategies possible. Thus, malfunctioning devices can be identified “on-line” and the systems reconfigured by making some light calculations that we could perform using for example a simple smartphone.

This paper constitutes a first attempt of solving complex models by combining “off-line” and “on-line” computations in the framework of proper generalized decompositions, in scenarios needing real time responses. In our opinion the possibility of computing parametric solutions that are then used by simply post-processing opens an unimaginable number of potential applications. For instance, uncertainty in the measurements (that has not been considered here), and that constitutes an essential ingredient in DDDAS, could eventually be efficiently treated in the PGD framework by considering uncertain parameters as additional dimensions. Worst-case scenarios could be computed in a similar way than the here proposed optimisation procedures. This constitutes our current effort of research and will be published elsewhere.

In the numerical, academic, examples here addressed “off-line” calculations needed two minutes of computing time (using matlab, a standard laptop and a non optimized simulation code) whereas all the “on-line” calculations were performed in 0.0015 seconds. The examples here addressed are too simple to be conclusive, but at least, they prove the pertinence of the proposed approach.

References

- [1] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, *J. Non-Newtonian Fluid Mech.*, **139**, 153-176, 2006.
- [2] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: transient simulation using space-time separated representations, *J. Non-Newtonian Fluid Mech.*, **144**, 98-121, 2007.
- [3] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, F. Chinesta, Non-incremental strategies based on separated representations: Applications in computational rheology, *Communications in Mathematical Sciences*, **8/3**, 671-695, 2010.
- [4] A. Ammar, F. Chinesta, P. Diez, A. Huerta. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199 1872-1880, 2010.



- [5] A. Ammar, M. Normandin, F. Chinesta, Solving parametric complex fluids models in rheometric flows, *Journal of Non-Newtonian Fluid Mechanics*, **165**, 23–24, 1588-1601, 2010.
- [6] P.-A. Boucard, P. Ladeveze. Une application de la méthode LATIN au calcul multirésolution de structures non linéaires. *Revue Européenne des Eléments Finis* (in french). Vol 8. Pages 903-920. 1999.
- [7] P.-A. Boucard, S. Buytet, P.-A. Guidault. A multiscale strategy for structural optimization. *Int Jal for Numerical Methods in Engineering*. Vol 78. Num 1. Pages 101-126. 2009.
- [8] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, *Computer Methods in Applied Mechanics and Engineering*, Volume 199, Issues 25-28, 15 May 2010, Pages 1872-1880, ISSN 0045-7825, DOI: 10.1016/j.cma.2010.02.012.
- [9] F. Chinesta, A. Ammar, P. Joyot, The nanometric and micrometric scales of the structure and mechanics of materials revisited: An introduction to the challenges of fully deterministic numerical descriptions, *International Journal for Multiscale Computational Engineering*, **6/3**, 191-213, 2008.
- [10] F. Chinesta, A. Ammar, E. Cueto, Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models, *Archives of Computational Methods in Engineering*, **17**, 4, 327–350, 2010.
- [11] F. Chinesta, A. Ammar, E. Cueto, Proper Generalized Decomposition of multiscale models, *International Journal of Numerical Methods in Engineering*, In press.
- [12] F. Chinesta, A. Ammar, E. Cueto, On the use of Proper Generalized Decompositions for solving the multidimensional Chemical Master Equation, *European Journal of Computational Mechanics*, **19**, 53-64, 2010.
- [13] F. Chinesta, P. Ladeveze, E. Cueto. A Short Review on Model Order Reduction based on Proper Generalized Decomposition. *Archives of Computational Methods in Engineering*, in press, 2011. DOI:10.1007/s11831-011-9064-7)
- [14] F. Darema, Engineering/Scientific and Commercial applications: differences, similarities, and future evolution, *Proceedings of the Second Hellenic European Conference on Mathematics and Informatics. HERMIS*, vol. 1, pp. 367-374, 1994.
- [15] J.E. Dennis Jr. and R.B.Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, *Classics in Applied Mathematics*, **16**, Corrected reprint of the 1983 original, Society for Industrial and Applied Mathematics (SIAM), PA, 1996.
- [16] John Michopoulos, Panagiota Tsompanopoulou, Elias N. Houstis, Charbel Farhat, Michel Lesoinne, John R. Rice, Anupam Joshi, On a data-driven environment for multiphysics applications. *Future Generation Comp. Syst.* 21(6): 953-968 (2005)



- [17] C. Farhat, J. G. Michopoulos, F. K. Chang, L. J. Guibas and A. J. Lew, Towards a Dynamic Data Driven System for Structural and Material Health Monitoring, Lecture Notes in Computer Science, ed. V. N. Alexandrov, G. D. van Albada, P. M.A. Slood, and J. Dongarra, Springer-Verlag, Vol. 3993, pp. 456-464 (2006).
- [18] D. Amsallem, J. Cortial and C. Farhat, Toward Real-Time CFD-Based Aeroelastic Computations Using a Database of Reduced-Order Information, AIAA Journal, Vol. 48, pp. 2029-2037 (2010)
- [19] D. Gonzalez, A. Ammar, F. Chinesta, E. Cueto, Recent advances in the use of separated representations, International Journal for Numerical Methods in Engineering, 81(5), 637-659, 2010.
- [20] C. Heyberger, P.-A. Boucard, D. Neron. Multiparametric analysis within the Proper Generalized Decomposition framework. Computational Mechanics. 2011. Available online DOI 10.1007/s00466-011-0646-x.
- [21] Vipul Hingne, Anupam Joshi, Elias Houstis, John Michopoulos, "On the Grid and Sensor Networks," Grid Computing, IEEE/ACM International Workshop on, p. 166, Fourth International Workshop on Grid Computing, 2003.
- [22] P. Ladeveze, Nonlinear computational structural mechanics, Springer, NY, 1999.
- [23] P. Ladeveze, J.-C. Passieux, D. Neron, The LATIN multiscale computational method and the proper generalized decomposition, Computer Methods in Applied Mechanics and Engineering, **199**, 21–22, 1287-1296, 2010.
- [24] R. B. Laughlin and David Pines. The theory of everything. Proceedings of the National Academy of Sciences, 97(1), 28-31, 2000.
- [25] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Jacquemin, F. Chinesta. Routes for Efficient Computational Homogenization of Non-Linear Materials Using the Proper Generalized Decomposition. Archives of Computational Methods in Engineering, 17/4, 373-391, 2010.
- [26] C. Le Bris, T. Lelièvre, Y. Maday, Results and Questions on a Nonlinear Approximation Approach for Solving High-dimensional Partial Differential Equations, Constructive Approximation, DOI 10.1007/s00365-009-9071-1.
- [27] G.M. Leonenko, T.N. Phillips, On the solution of the Fokker-Planck equation using a high-order reduced basis approximation. Computer Methods in Applied Mechanics and Engineering, **199**,1–4, 58-168 , 2009.
- [28] John G. Michopoulos, Panayota Tsompanopoulou, Elias N. Houstis, Anupam Joshi, Sasikanth Avancha, and Haiping Zhang, Towards Agent-Based Grid-Enabled and Sensor-Driven Fire Dynamics Simulation Harnessed Over Bluetooth and Wi-Fi Devices. ASME Conf. Proc. 2005, 111 (2005), DOI:10.1115/DETC2005-84770
- [29] John Michopoulos, Panagiota Tsompanopoulou, Elias N. Houstis, Anupam Joshi: Agent-Based Simulation of Data-Driven Fire Propagation Dynamics. International Conference on Computational Science 2004: 732-739.



- [30] A. Nouy, P. Ladeveze, Multiscale computational strategy with time and space homogenization: A radial-type approximation technique for solving microproblems, *International Journal of Multiscale Computational Engineering*, **170/2**, 2004.
- [31] A. Nouy, Recent developments in spectral stochastic methods for the solution of stochastic partial differential equations, *Archives of Computational Methods in Engineering*, **16/3**, 251-285, 2009.
- [32] A. Nouy, Proper Generalized Decompositions and separated representations for the numerical solution of high-dimensional stochastic problems. *Archives of Computational Methods in Engineering*, **17**, 4, 403-434, 2010.
- [33] J.T. Oden, T. Belytschko, J. Fish, T.J.R. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz and S.Yip, *Simulation-based Engineering Science: Revolutionizing Engineering Science through simulation*. NSF Blue Ribbon Panel on SBES, 2006.
- [34] JL Eftang, DJ Knezevic, and AT Patera, An hp Certified Reduced Basis Method for Parametrized Parabolic Partial Differential Equations. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):395-422, 2011.
- [35] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of parametric models by using the Proper Generalized Decomposition, *Mathematics and Computer Simulation*, **81**, 4, 791-810, 2010.
- [36] DDDAS Workshop 2000 Final Report. Arlington, VA, U.S.A. Technical report, National Science Foundation, 2000.
- [37] DDDAS Workshop 2006 Final Report. Arlington, VA, U.S.A. Technical report, National Science Foundation, 2006.



3.4 Conclusion

Le travail réalisé porte sur une modélisation innovante dans le domaine des méthodes numériques. L'utilisation de la PGD pour la résolution des modèles multidimensionnels a permis le calcul de solutions paramétriques "off-line". Plus tard, des optimisations "on-line" sont réalisées, pour permettre le contrôle en temps réel de procédés industriels. La méthode est appliquée sur la pultrusion à titre d'exemple. A noter que grâce à la formulation de la PGD englobant les paramètres comme coordonnées du problème, le calcul des sensibilités de la solution par rapport aux paramètres devient direct. Par la suite les matrices de gradients ou Hessiennes peuvent être calculées "off-line" afin d'accélérer le calcul "on-line". Une application pour téléphone portable et tablette a été codée pour illustrer la vitesse du calcul. De plus, une marque déposée, la "IPGD®" a vu le jour.

Cependant, plus de travail devra être réalisé. Le bruit des capteurs et du système n'est pas pris en compte. Souvent les valeurs données par les capteurs ne correspondent à aucune combinaison des paramètres. Ceci est dû au manque d'informations sur les perturbations, même minimales, dans le système. D'ailleurs les problèmes de calcul inverse sont connus pour être sensibles aux moindres perturbations. Toutefois, le travail proposé pourra être étendu dans de telles situations.



4

Modèle thermique paramétrique de la dépose de bandes thermoplastiques

4.1	Simulation des plaques stratifiées en 3D	110
4.1.1	Un problème modèle	110
4.1.2	Ressemblances et différences avec la théorie classique	112
4.1.3	Pour aller plus loin	113
4.2	Simulation des plaques stratifiées paramétrées	117
4.3	Simulation du drapage assisté en 3D	119
4.3.1	Présentation du modèle	119
4.3.2	Simulation des résistances thermiques	121
4.3.3	Stabilisation de la convection	123
4.3.4	Les résultats obtenus	125
4.4	Simulation paramétrique de la dépose de bandes	127
4.4.1	Le traitement des paramètres	127
4.4.2	Le résultat obtenu: une solution 8D	128
4.4.3	Le post-traitement des solutions 8D	130
4.5	Conclusion	132

La modélisation avancée des procédés de mise en forme des composites est confrontée à plusieurs verrous. Malgré les avancées récentes de la mécanique numérique et des stratégies de discrétisation, certains problèmes restent irrésolvables. En effet, les matériaux composites sont hétérogènes, anisotropes et multi-échelles. Dans ce chapitre on utilise la PGD pour simuler le comportement thermique des structures composites en 3D. De plus, on résoudra des problèmes 3D paramétrés, allant à des dimensions supérieures à huit. De telles résolutions sont simplement inimaginables en utilisant les méthodes de discrétisation classiques.

4.1 Simulation des plaques stratifiées en 3D

Souvent, des lois de comportement sont définies pour décrire le comportement des plaques stratifiées dans l'épaisseur (Thionnet et Renard, 2001). Cependant, ce comportement homogénéisé n'est pas toujours facile à déterminer. Toutefois, le calcul 3D utilisant les éléments finis est long et lourd à réaliser, mais indispensable pour la bonne modélisation de la pièce. C'est là que la PGD s'impose comme alternative aux éléments finis classiques. En effet, comme on a déjà dit, la PGD réalise dans des géométries plaques ou coques, des calculs 3D avec un temps de calcul caractérisant des simulations 2D.

4.1.1 Un problème modèle

Dans cette section, nous calculons le champ de température en 3D dans une plaque stratifiée. La différence avec l'approche classique ainsi que quelques limites de ces théories sont montrées. Afin d'illustrer l'approche PGD utilisée, considérons le problème modèle suivant :

$$\frac{\partial \mathcal{U}}{\partial t} - \nabla \cdot (\mathbf{K} \nabla \mathcal{U}) = 0 \quad (4.1)$$

où \mathbf{K} est le tenseur de conductivité thermique, \mathcal{U} est le champ de température 3D considéré. On cherche à résoudre cette équation dans une plaque stratifiée où les fibres sont placées suivant les directions $45^\circ / -45^\circ$ par rapport à l'axe des x . Les dimensions de la plaque sont $L \times l \times e$. Les conditions aux limites sont les suivantes :

$$\begin{cases} \mathcal{U}(0, y, z, t) = \tau_1 \\ \mathcal{U}(L, y, z, t) = \tau_2 \\ \nabla \mathcal{U} \cdot \mathbf{n} = 0 \text{ ailleurs} \end{cases} \quad (4.2)$$

τ_1 et τ_2 sont deux températures connues. Le tenseur de conductivité \mathbf{K}^j , pour une plaque formée seulement du pli j est écrit comme étant :

$$\mathbf{K}^j = \begin{bmatrix} K_{xx}^j & K_{xy}^j & 0 \\ K_{yx}^j & K_{yy}^j & 0 \\ 0 & 0 & K_{zz}^j \end{bmatrix} \quad (4.3)$$

D'après Loix (2005), les paramètres K_{xx} , K_{xy} , K_{yx} , K_{yy} et K_{zz} s'écrivent sous la forme



suivante :

$$\begin{cases} K_{xx}^j = \cos^2(\theta_j) \times (K_{//} - K_{\perp}) + K_{\perp} \\ K_{xy}^j = K_{yx}^j = \cos(\theta_j) \times \sin(\theta_j) \times (K_{//} - K_{\perp}) \\ K_{yy}^j = \sin^2(\theta_j) \times (K_{//} - K_{\perp}) + K_{\perp} \\ K_{zz}^j = K_{\perp} \end{cases} \quad (4.4)$$

$K_{//}$ étant la conductivité dans la direction des fibres, alors que K_{\perp} la conductivité dans la direction perpendiculaire aux fibres. θ_j l'orientation des fibres par rapport à l'axe des x pour le pli j . Par suite, en généralisant pour tout un stratifié formé de P plis, le tenseur de conductivité devient un assemblage dans la direction de l'épaisseur de tous les tenseurs élémentaires :

$$\mathbf{K} = \sum_{j=1}^{j=P} Z^j(z) \mathbf{K}^j = \sum_{j=1}^{j=P} Z^j(z) \begin{bmatrix} K_{xx}^j & K_{xy}^j & 0 \\ K_{yx}^j & K_{yy}^j & 0 \\ 0 & 0 & K_{zz}^j \end{bmatrix} \quad (4.5)$$

Où P est le nombre de plis et :

$$Z^j(z) = \begin{cases} 1 & \text{si } z \in \text{pli } j \\ 0 & \text{autrement} \end{cases} \quad (4.6)$$

On remarque ainsi que les termes du tenseur de conductivité ne sont fonctions que de l'épaisseur z . De plus, le fait d'écrire le tenseur de conductivité sous cette forme conserve toutes les propriétés 3D du stratifié. D'autre part, nous pouvons maintenant écrire l'opérateur $\nabla(\mathbf{K}\nabla\mathcal{U})$ sous une forme séparée, en fonction de $\Omega \times \mathcal{Z}$. $\Omega = (x; y)$ est un domaine 2D, $\mathcal{Z} = (z)$ un domaine 1D. Ainsi, nous utilisons la formulation de la PGD décrite dans la section 2.1.

Le calcul est réalisé en séparant l'espace physique en $\Omega \times \mathcal{Z} \times \mathcal{T}$, avec \mathcal{T} représentant l'espace 1D du temps. On cherche la solution du problème (4.1) sous la forme suivante :

$$\mathcal{U} \approx \sum_{i=1}^{i=N} \mathbf{F}_i(x, y) \times Z_i(z) \times T_i(t) \quad (4.7)$$

La formulation intégrale de résidus pondérés du problème (4.1) est donc :

$$\int_{\Omega \times \mathcal{Z} \times \mathcal{T}} \mathcal{U}^* \left(\frac{\partial \mathcal{U}}{\partial t} - \nabla(\mathbf{K}\nabla\mathcal{U}) \right) d\Omega d\mathcal{Z} d\mathcal{T} = 0 \quad (4.8)$$

■

Ensuite, en développant l'équation (4.8) intégrant par partie et tenant compte des conditions limites définies dans l'équation (4.2), on obtient :

$$\int_{\Omega \times \mathcal{Z} \times \mathcal{T}} \mathcal{U}^* \left(\frac{\partial \mathcal{U}}{\partial t} \right) d\Omega d\mathcal{Z} d\mathcal{T} - \int_{\Omega \times \mathcal{Z} \times \mathcal{T}} \left(K_{xx} \frac{\partial \mathcal{U}}{\partial x} \frac{\partial \mathcal{U}^*}{\partial x} + K_{xy} \frac{\partial \mathcal{U}}{\partial x} \frac{\partial \mathcal{U}^*}{\partial y} + K_{yx} \frac{\partial \mathcal{U}}{\partial y} \frac{\partial \mathcal{U}^*}{\partial x} + K_{yy} \frac{\partial \mathcal{U}}{\partial y} \frac{\partial \mathcal{U}^*}{\partial y} + K_{zz} \frac{\partial \mathcal{U}}{\partial z} \frac{\partial \mathcal{U}^*}{\partial z} \right) d\Omega d\mathcal{Z} d\mathcal{T} = 0 \quad (4.9)$$

A partir d'où on procède comme décrit dans la section 2.1.

4.1.2 Ressemblances et différences avec la théorie classique

Dans cette section, la solution de l'équation de la chaleur (4.1) dans une plaque stratifiée formée de deux plis, où les fibres sont placées suivant les directions $45^\circ / -45^\circ$ avec l'axe des x , est montrée. Plus tard, une comparaison avec la théorie moyennant les propriétés thermiques dans l'épaisseur, sera réalisée. Les dimensions de la plaque sont $L = 6 \text{ m}$ pour la longueur, $l = 3 \text{ m}$ pour la largeur et $e = 0.01 \text{ m}$ pour l'épaisseur. Les conditions limites sont données par l'équation (4.2) avec $\tau_1 = 10^\circ C$ et $\tau_2 = 0^\circ C$. La solution est donnée par la figure 4.1 pour l'instant $t = 1 \text{ s}$ et par la figure 4.2 pour l'instant $t = 10 \text{ s}$. A noter que les solutions sont montrées pour les plaques 3D, mais comme l'épaisseur est minime par rapport au reste de la pièce, la face perçue est donc la face supérieure de la plaque. La conductivité $K_{//}$ est considérée dix fois plus grande que la conductivité K_{\perp} , on prend ainsi $K_{//} = 5 \text{ W/m.K}$ et $K_{\perp} = 0.5 \text{ W/m.K}$. La solution PGD est obtenue en 23 produits de fonctions et 5.03 secondes de calcul pour un maillage de 55300 ddl.

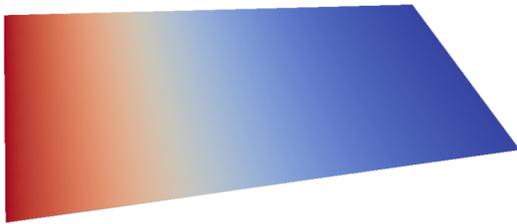


figure 4.1 – *Solution du problème à $t = 1 \text{ s}$*

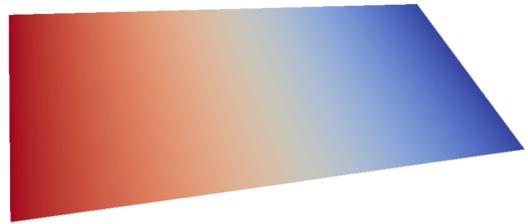


figure 4.2 – *Solution du problème à $t = 10 \text{ s}$*

Les champs de température obtenus sont conformes avec les résultats prévus par la théorie d'homogénéisation des lois de comportements (Kaminski, 2003; Benazzouk *et al.*, 2008; Nowacki et Zarka, 2008) qui proposent une moyenne de conductivité dans l'épaisseur. Par contre, ceci n'est pas le cas pour les flux dans la pièce. En effet, on sait que le flux thermique

(notons FL) s'écrit comme étant :

$$FL = - \begin{pmatrix} K_{xx} & K_{xy} & 0 \\ K_{yx} & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{pmatrix} \begin{pmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{\partial U}{\partial z} \end{pmatrix} = - \begin{pmatrix} K_{xx} \frac{\partial U}{\partial x} + K_{xy} \frac{\partial U}{\partial y} \\ K_{yx} \frac{\partial U}{\partial x} + K_{yy} \frac{\partial U}{\partial y} \\ K_{zz} \frac{\partial U}{\partial z} \end{pmatrix} \quad (4.10)$$

La figure 4.3 représente le flux de température dans la plaque, après $t = 1$ s. On remarque que les flux suivent la direction des fibres. Les théories classiques approximent un flux suivant l'axe des x en calculant la moyenne sur l'épaisseur, ce qui est très loin de la réalité 3D. Les flux à $t = 10$ s sont illustrés par la figure 4.4. Après un certain temps, le régime permanent s'est installé. Les flux sont devenus uniformes dans la plaque, mais ils sont loin des prédictions avec une conductivité moyenne dans l'épaisseur.

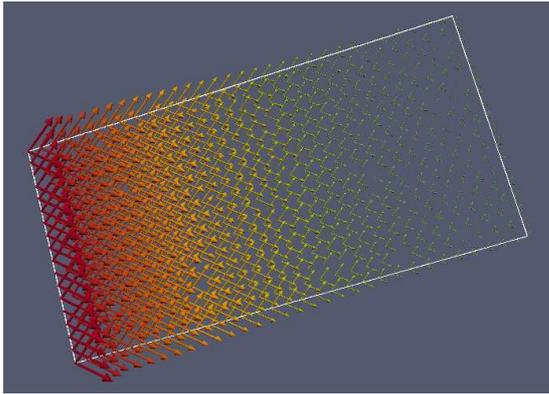


figure 4.3 – Flux dans la plaque pour $t = 1$ s

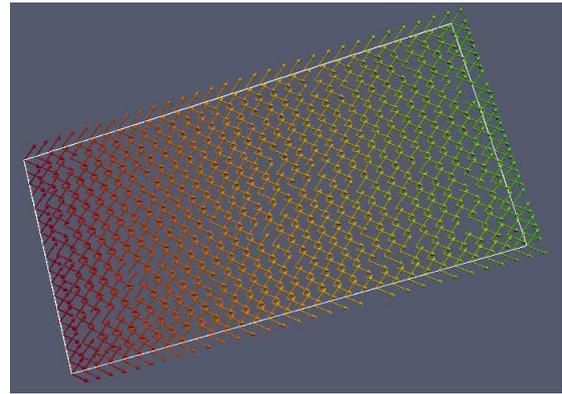


figure 4.4 – Flux dans la plaque pour $t = 10$ s

Si l'on suit une seule ligne de flux, on remarque que ce flux atteint le bord adiabatique de la pièce en suivant une direction donnée. Par suite, ce dernier rebondit sur le bord pour passer à l'autre pli (figure 4.4). Ensuite, la ligne de flux se poursuit dans la direction orthogonale. Ce mécanisme est illustré par la figure 4.5. Ce comportement n'aurait pas été prévu facilement sans une simulation 3D des stratifiées composites. D'autres travaux ont été réalisés dans l'équipe sur l'imprégnation et la loi de Darcy en utilisant la PGD (Chinesta *et al.*, 2011a). Ces travaux ont abouti à des résultats similaires.

4.1.3 Pour aller plus loin

Dans la section 4.1.1, une plaque formée de deux plis a été étudiée. Cependant des simulations de pièces plus complexes ont été réalisées. La figure 4.6 montre le champ de température à $t = 1$ s dans une plaque 3D formée d'un empilement de quatre empilements $0^\circ/45^\circ/90^\circ/-45^\circ$.



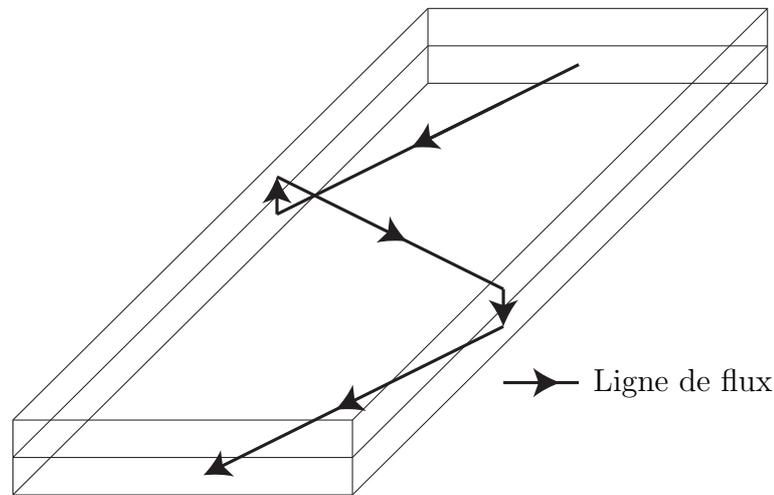
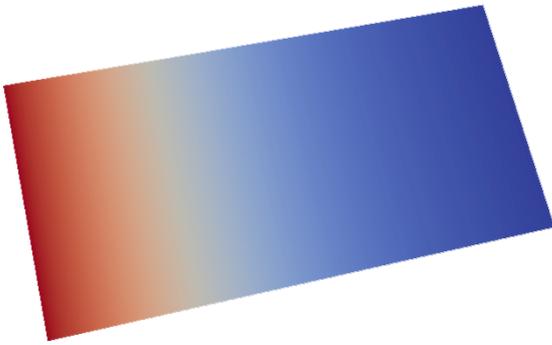
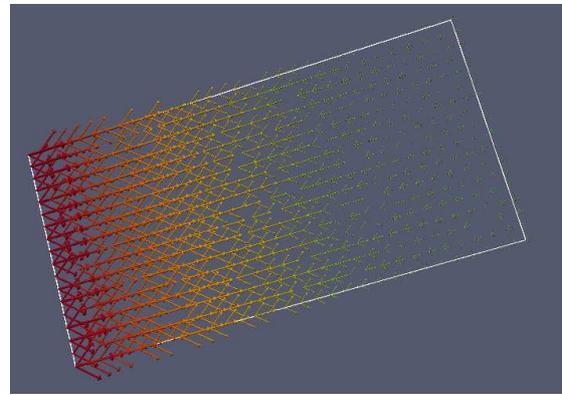


figure 4.5 – Mécanisme de “transport” de la chaleur dans une plaque stratifiée

La figure 4.7 montre le flux dans cette même plaque. Les mêmes valeurs des paramètres utilisés dans la section 4.1.2 ont été reprises lors de cette simulation. La solution est obtenue en 33 modes et 10.19 secondes de calcul.

figure 4.6 – Champ de température dans la plaque formée de quatre périodes à $t = 1$ sfigure 4.7 – Flux dans la plaque formée de quatre périodes à $t = 1$ s

De plus, une pièce plus complexe a été simulée. La figure 4.8 montre le maillage de la plaque. Les conditions limites sont $\tau_1 = 10^\circ$ pour $x = 0$ et $\tau_2 = 0^\circ C$ pour le reste des nœuds soumis à des conditions de Dirichlet. Le reste du bord est soumis à des conditions de flux normal nul. D’abord on commence par simuler une plaque à deux plis, avec des fibres placées sur les directions $45^\circ / -45^\circ$ par rapport à l’axe des x . La solution est obtenue en 28 modes et 5.88 secondes de calcul. La figure 4.9 montre le champ de température dans la plaque. L’amplitude des gradients de température est donnée par la figure 4.10. Finalement, le flux est illustré dans la figure 4.11. A noter que les flux suivent les directions des fibres, mais ils s’amplifient

aussi aux voisinages des obstacles circulaires, et aux points anguleux. L'effet de bord observé dans les simulations 3D n'est pas prévu dans le cadre des théories homogénéisées 2D.

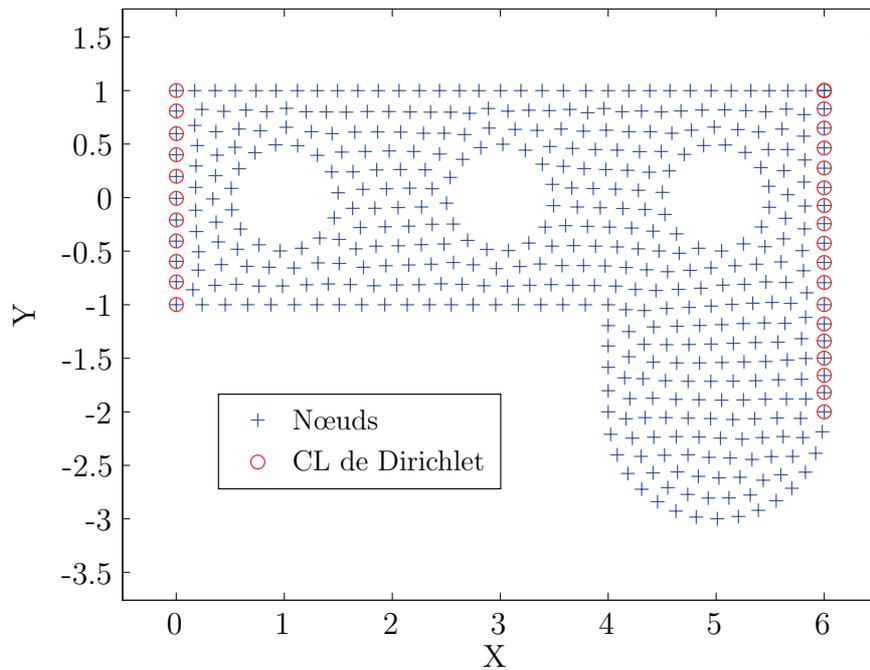


figure 4.8 – Maillage de la plaque dans le domaine Ω 2D

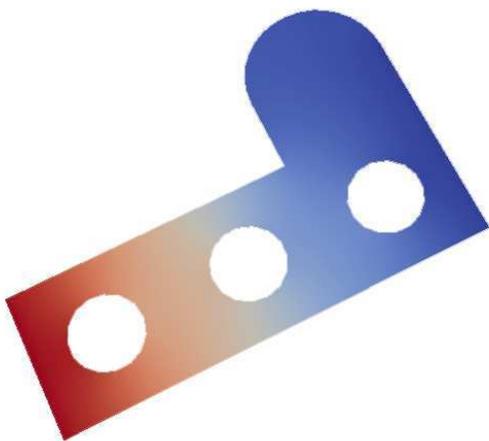


figure 4.9 – Champ de température dans la plaque formée de deux plis à $t = 10$ s

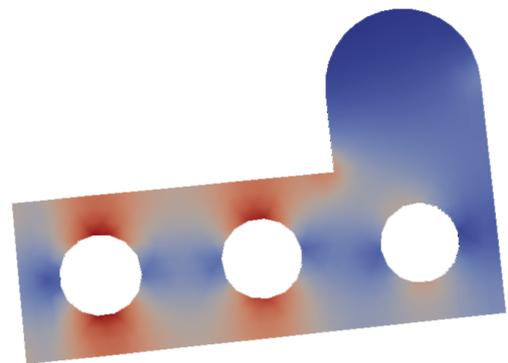


figure 4.10 – Carte des gradients dans la plaque formée de deux plis $t = 10$ s

Enfin, cette même géométrie (figure 4.8) a été simulée aussi pour l'empilement défini par la suite $[0^\circ/45^\circ/90^\circ/-45^\circ] \times 4$. La solution est obtenue en 46 modes et 17.12 secondes



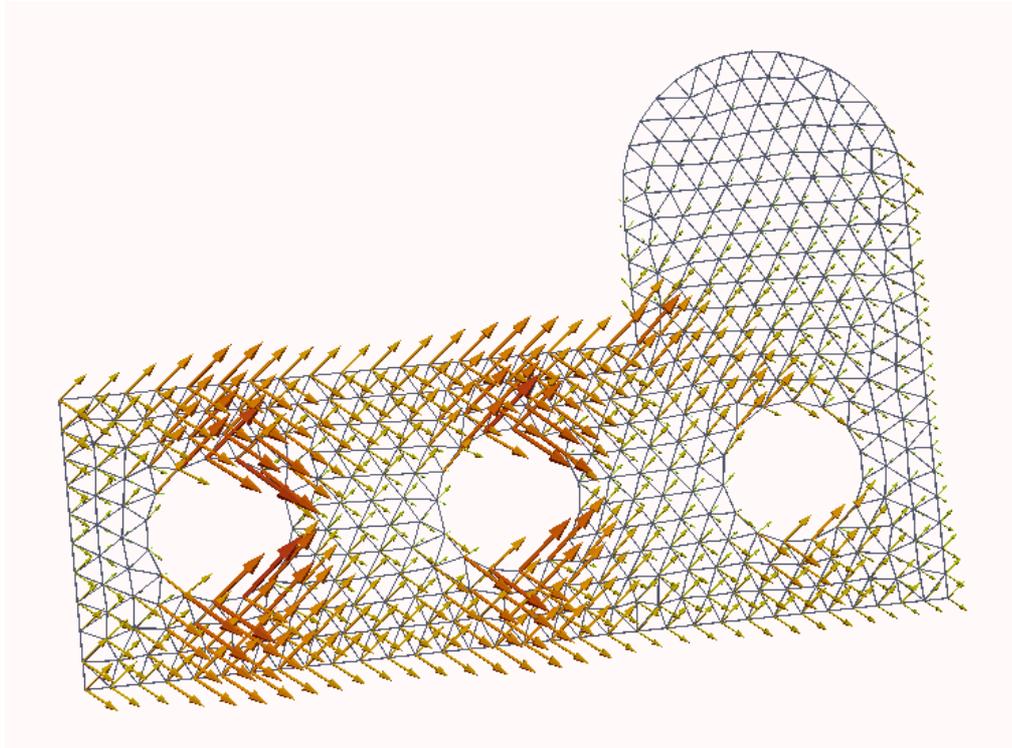


figure 4.11 – Carte des flux dans la plaque formée de deux plis $t = 10$ s

de calcul. Les champs de température, de gradient thermique et de flux sont donnés par les figures 4.12, 4.13 et 4.14 respectivement.

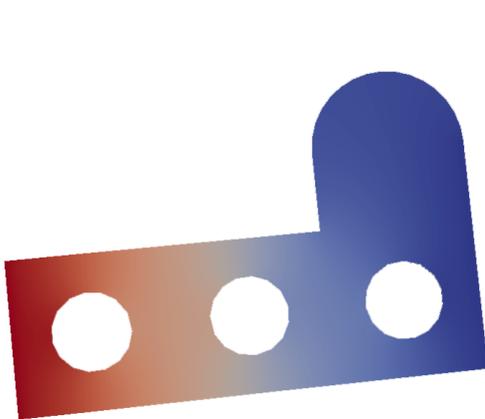


figure 4.12 – Champ de température pour un empilement $[0^\circ/45^\circ/90^\circ/-45^\circ] \times 4$

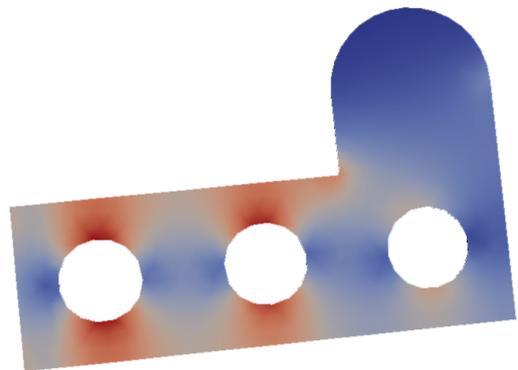


figure 4.13 – Carte des gradients pour un empilement $[0^\circ/45^\circ/90^\circ/-45^\circ] \times 4$ à $t = 10$ s

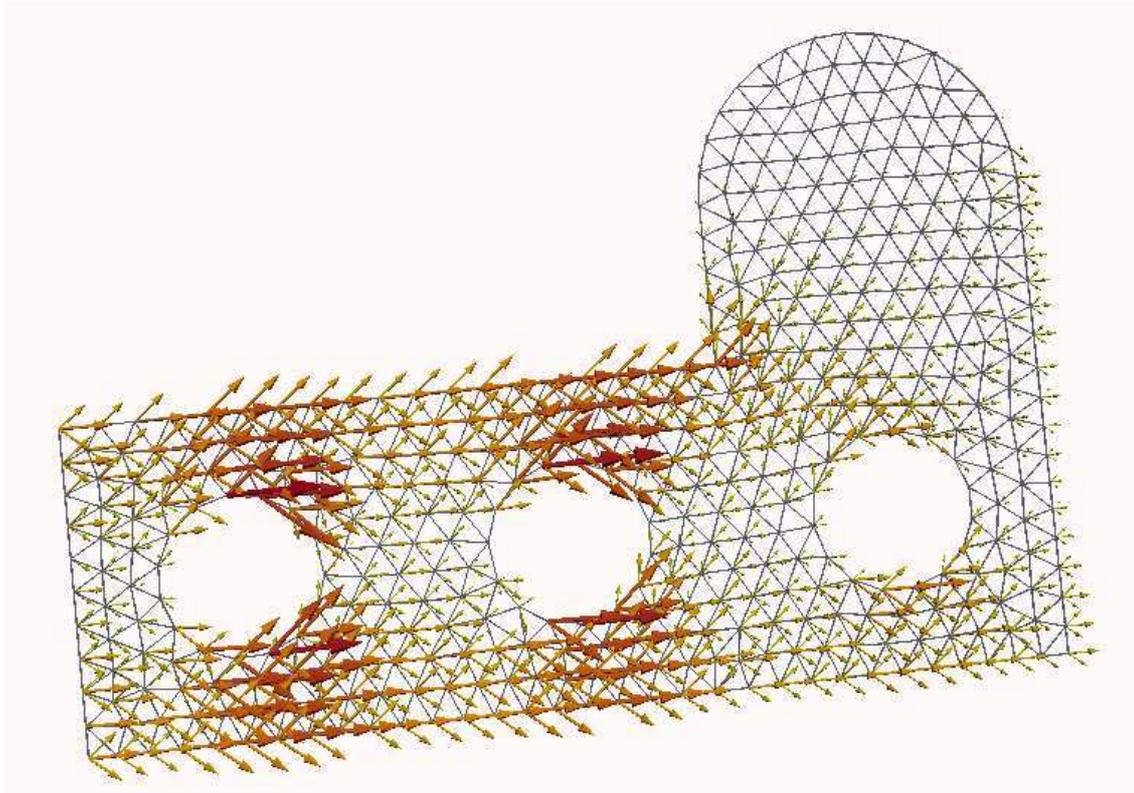


figure 4.14 – Les flux dans la plaque d'empilement $[0^\circ/45^\circ/90^\circ/-45^\circ] \times 4$ à $t = 10$ s

4.2 Simulation des plaques stratifiées paramétrées

Le potentiel de la PGD réside dans sa capacité à résoudre des modèles multidimensionnels. Ainsi, nous avons considéré la simulation d'une plaque composite stratifiée paramétrée. Pour ce modèle, le paramétrage de l'orientation des plis dans une plaque a été réalisé. Afin d'atteindre notre but, une première étape de mise en équation du problème est nécessaire. Considérons une nouvelle fois l'équation (4.1). Dans cette section, le domaine de simulation est une plaque de dimension $((L,l) \times e) \in \Omega \times \mathcal{Z}$, formée de deux plis. La résolution de l'équation de la chaleur en régime transitoire est réalisée sur un domaine temporel $\mathcal{T} = [0,10]$ s.

De plus, dans cette section on résout pour toute orientation possible des fibres. Ainsi, deux coordonnées sont ajoutées, θ_1 et θ_2 , qui correspondent aux orientations des fibres dans le premier et le second pli respectivement. Ainsi on cherche la solution de l'équation par la PGD sous la forme :

$$\mathcal{U} \approx \sum_{i=1}^{i=N} \mathbf{F}_i(x,y) \cdot Z_i(z) \cdot T_i(t) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \quad (4.11)$$

■

Les nouvelles dimensions apparaissent dans le tenseur de conductivité \mathbf{K} . Ainsi en tenant compte de l'équation (4.4), on pourra définir à nouveau un tenseur de conductivité fonction de l'épaisseur z , mais aussi des orientations θ_1 et θ_2 . Par suite on écrit :

$$\mathbf{K} = \sum_{j=1}^{j=P} Z^j(z) \begin{bmatrix} \cos^2(\theta_j) (K_{//} - K_{\perp}) + K_{\perp} & \cos(\theta_j) \sin(\theta_j) (K_{//} - K_{\perp}) & 0 \\ \cos(\theta_j) \sin(\theta_j) (K_{//} - K_{\perp}) & \sin^2(\theta_j) (K_{//} - K_{\perp}) + K_{\perp} & 0 \\ 0 & 0 & K_{\perp} \end{bmatrix} \quad (4.12)$$

P étant le nombre de plis paramétrés, dans cette section $P = 2$. Cette notation influe directement sur la construction des opérateurs. En effet, si on reprend la formulation variationnelle présentée dans l'équation (4.9), on remarque que K_{xx} , K_{xy} et K_{yy} sont fonctions de z , θ_1 et θ_2 .

La figure 4.15 montre les flux pour différentes orientations des plis. La plaque est formée de deux plis d'épaisseur 0.5 mm chacun, de dimension 6×3 m. Le calcul est achevé au bout de 1085.17 secondes avec 30 produits de fonctions utilisant l'algorithme 2.4, avec la minimisation du résidu. Ces résultats sont obtenus en utilisant la mise à jour des modes à chaque itération.

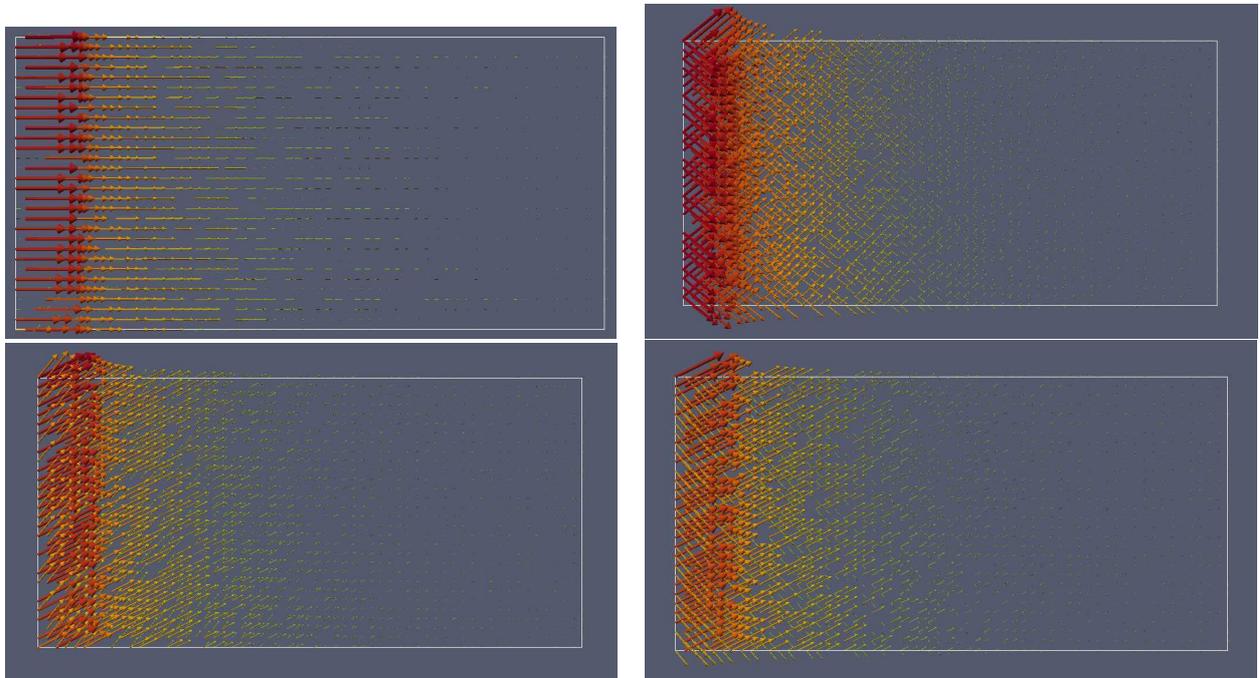


figure 4.15 – Flux dans la plaque à $0^\circ/90^\circ$, $45^\circ/-45^\circ$, $30^\circ/60^\circ$ et $30^\circ/120^\circ$ à $t = 1s$

4.3 Simulation du drapage assisté en 3D

La simulation du drapage assisté par robot représente un défi du point de vue de sa simulation. En effet, la richesse de la physique intra et inter-plis incite à l'utilisation d'une simulation 3D. Cette dernière nécessite un maillage fin dans la direction de l'épaisseur, capable de reproduire le champ riche de température. En utilisant la PGD, l'épaisseur est séparée des coordonnées planes. Alors, la possibilité de raffiner le maillage sans amplifier dramatiquement le temps de calcul, est envisageable. Dans un premier temps, on réalise le calcul pour des valeurs de paramètres fixées. Plus tard, le calcul est répété en dimension supérieure, en ajoutant les paramètres comme coordonnées supplémentaires.

4.3.1 Présentation du modèle

Dans cette section nous présentons un modèle thermique du drapage assisté par robot. Dans le procédé, le robot se déplace pour déposer les plis à une vitesse v . La figure 4.16 illustre le procédé du drapage, à dimensions nettement inférieures. Pour simuler ce modèle, considérons qu'un pli déposé possède une largeur de 10mm , et qu'on cherche à simuler une plaque de $2 \times 1\text{m}$, avec 50 couches. Supposons qu'on réalise une simulation incrémentale avec un déplacement de un centimètre de la source à chaque itération. Si une simulation est achevée au bout d'une minute, la simulation de ce modèle 3D s'achèverait au bout de 19 ans de calcul!

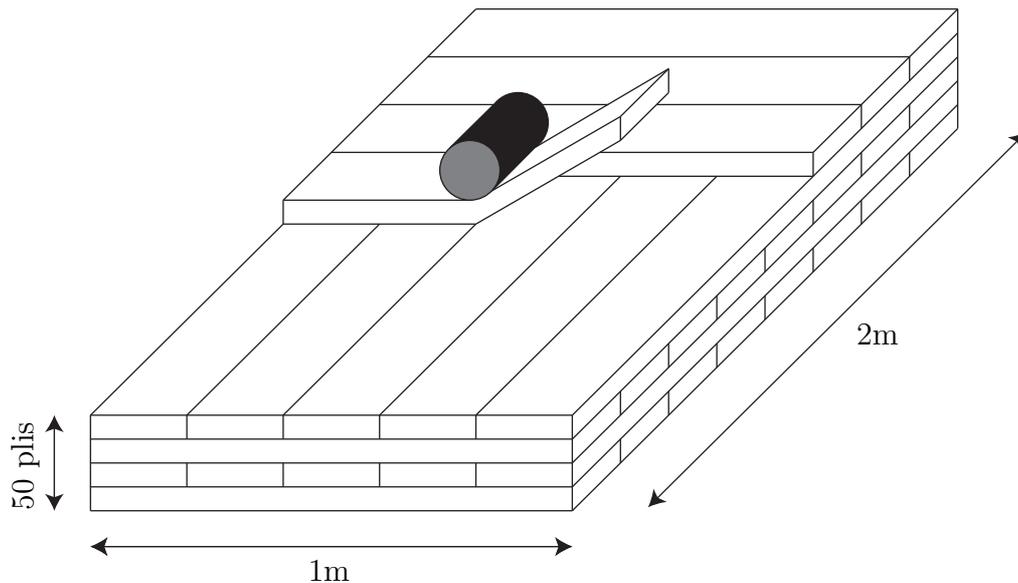


figure 4.16 – *Illustration du drapage assisté par robot, modèle réel*

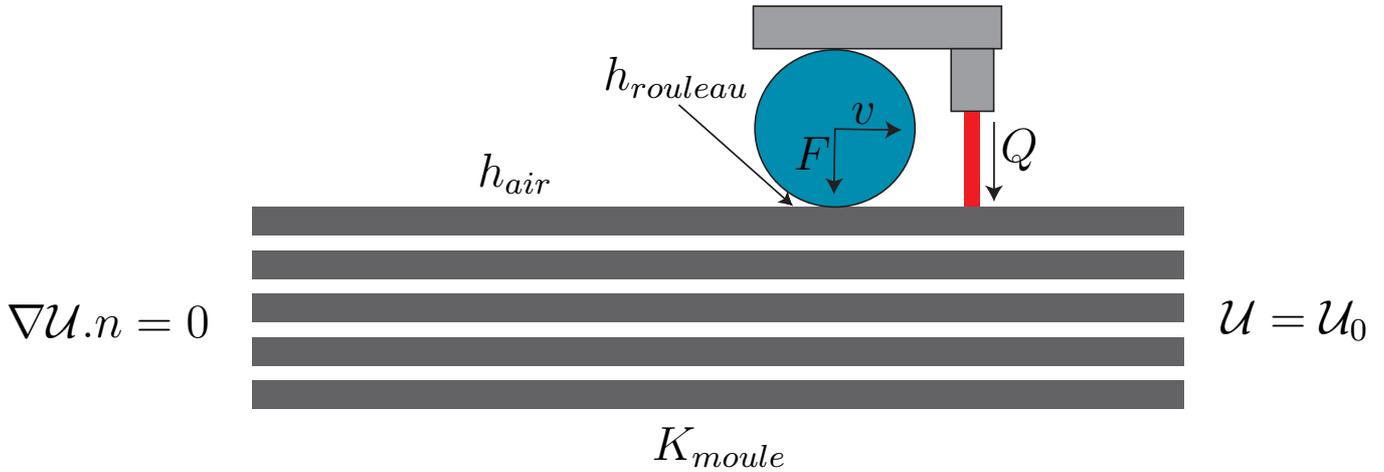


figure 4.17 – Schéma du modèle local du procédé de dépose de bandes

Cependant, ce modèle est quasi-statique si on se place dans le référentiel de la tête du robot et loin des effets de bord. La figure 4.17 illustre le modèle local autour de la tête du robot. Ainsi, on réalisera une simulation en considérant la matière mobile et la source fixe. Considérons donc l'équation suivante d'advection-diffusion de la chaleur :

$$-\rho C_p v \frac{\partial \mathcal{U}}{\partial x} - \nabla (\mathbf{K} \nabla \mathcal{U}) = 0 \quad (4.13)$$

On note ρ la masse volumique de la pièce et C_p sa capacité calorifique. Le calcul est réalisé dans un domaine $\Omega \times \mathcal{Z}$, avec Ω défini par les coordonnées dans le plan, $\Omega = [0, L] \times [0, l]$. D'autre part, \mathcal{Z} est un domaine 1D relatif à l'épaisseur avec $\mathcal{Z} = [0, e]$. Les conditions aux limites sont les suivantes :

$$\begin{cases} \nabla \mathcal{U}(0, y, z) \cdot n = 0 \\ \mathcal{U}(L, y, z) = \mathcal{U}_0 \\ \nabla \mathcal{U}(x, 0, z) \cdot n = 0 \\ \nabla \mathcal{U}(x, l, z) \cdot n = 0 \\ -\mathbf{K} \nabla \mathcal{U}(x, y, e) \cdot n = \mathbf{h}_a (\mathcal{U} - \mathcal{U}_a) \\ -\mathbf{K} \nabla \mathcal{U}(x, y, 0) \cdot n = \mathbf{h}_m (\mathcal{U} - \mathcal{U}_m) \end{cases} \quad (4.14)$$

\mathcal{U}_0 est la température de la pièce avant le passage de la tête du robot, \mathcal{U}_a la température de l'air et \mathcal{U}_m la température du moule. De plus, \mathbf{h}_a est le coefficient de convection entre l'air et la pièce, alors que \mathbf{h}_m le coefficient de conduction entre cette dernière et le moule.

Afin de modéliser les résistances thermiques de contact, on suppose que la distance entre les



deux plis est suffisamment petite pour que le transfert de chaleur se fasse par conduction. Ainsi, la loi d'échange de température entre deux plis s'écrit :

$$\begin{cases} -\mathbf{K}^- \nabla \mathcal{U}^- = \mathbf{h} (\mathcal{U}^- - \mathcal{U}^+) \\ -\mathbf{K}^+ \nabla \mathcal{U}^+ = \mathbf{h} (\mathcal{U}^- - \mathcal{U}^+), \end{cases} \quad (4.15)$$

avec h étant la conductance thermique inter-plis. La section 4.3.2 explique en détail la méthode utilisée.

4.3.2 Simulation des résistances thermiques

Afin de tester l'approche utilisée pour la simulation des résistances thermiques, on commence par simuler un cas simple. Dans notre exemple, nous considérons un domaine unidimensionnel $\omega = [0,2]$ et la résistance $r = 1$ placée au milieu du domaine ω . L'équation de la chaleur s'écrit :

$$\nabla (K \nabla \mathcal{U}) = 0 \quad (4.16)$$

\mathcal{U} étant la température et K la conductivité thermique, avec :

$$\begin{cases} K(x) = 1 \text{ si } x \in [0; 1] \\ K(x) = 2 \text{ si } x \in [1; 2] \end{cases} \quad (4.17)$$

De plus les conditions limites sont :

$$\begin{cases} \mathcal{U}(0) = 50^\circ\text{C} \\ \mathcal{U}(1) = 100^\circ\text{C} \end{cases} \quad (4.18)$$

Pour simuler l'existence de la résistance r , on utilise un nœud double à la position $x = 1$. En outre, la matrice \mathbf{M} qui exprime l'opérateur en question intègre la relation d'échange thermique :

$$\begin{cases} -K^- \nabla \mathcal{U}^- = \frac{1}{r} (\mathcal{U}^- - \mathcal{U}^+) \\ -K^+ \nabla \mathcal{U}^+ = \frac{1}{r} (\mathcal{U}^- - \mathcal{U}^+), \end{cases} \quad (4.19)$$



pourra plus être amélioré par les passages suivants. Cela est illustré par la figure 4.19, x_1 étant l'abscisse du centre du rouleau.

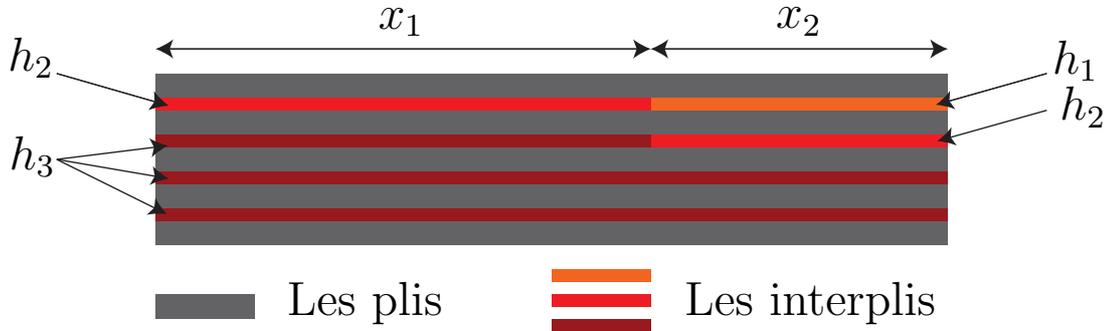


figure 4.19 – Modélisation des résistances de contact

4.3.3 Stabilisation de la convection

L'équation (4.13) est une équation de convection-diffusion de la chaleur. Ce genre d'équation pose de très fortes limitations à l'utilisation des techniques de discrétisation adaptées aux problèmes elliptiques (Bird *et al.*, 1960; Bejan, 2004). En effet, si on considère une discrétisation de Galerkin par éléments finis de l'équation de convection-diffusion en 1D autour d'un nœud i , on obtient :

$$(1 - P_e)(\mathcal{U}_{i-1} - \mathcal{U}_i) = (1 + P_e)(\mathcal{U}_{i-1} - \mathcal{U}_i), \quad (4.21)$$

où \mathcal{U} étant les températures. En notant K la conductivité et p la taille des éléments de discrétisation, P_e est le nombre de Peclet défini par :

$$P_e = \frac{v \cdot p}{2K} \quad (4.22)$$

De l'analyse de l'équation (4.21), il est facile de constater que pour des valeurs du nombre de Peclet supérieures à l'unité, on obtient des oscillations de la solution, dues au changement du signe de la dérivée à chaque nœud. On peut démontrer que cette oscillation est engendrée par une diffusion négative introduite par la discrétisation (Donea et Huerta, 2003). Dans la PGD, dans sa formulation Galerkin, ce problème d'oscillation apparaît également. D'ailleurs, quand nous avons tenté de résoudre cette équation sans stabilisation, le résultat a montré des oscillations énormes (voir la figure 4.20).

Généralement, un raffinement suffisant du maillage peut éliminer les oscillations. Cependant, ce choix n'est pas efficace, car il engendre une augmentation du temps de calcul, mais aussi du



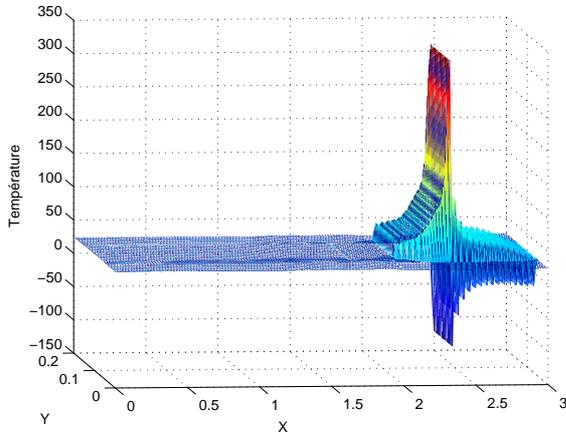


figure 4.20 – Solution à la surface supérieure de la pièce, sans stabilisation

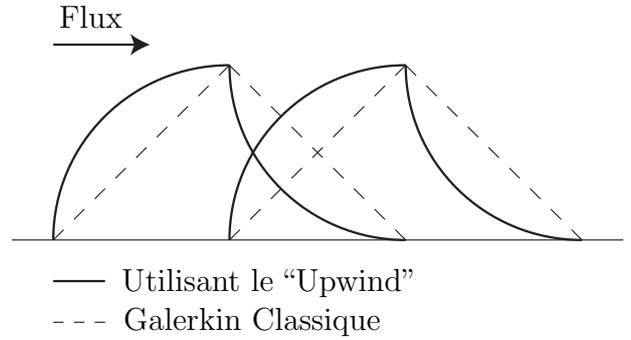


figure 4.21 – Fonctions de forme utilisant la méthode “Upwind”

temps nécessaire à l’évaluation de la solution précalculée. D’autre part, plusieurs méthodes de stabilisation ont été proposées tout au long des années (Donea et Huerta, 2003). Les méthodes classiques se basent sur la sur-diffusion pour compenser la diffusion négative. Nous pouvons citer la formulation dite “Upwind” qui utilise une formulation décentrée en espace vers l’amont pour les dérivées premières. Ainsi les fonctions de forme éléments finis auront une allure similaire à celle de la figure 4.21. On pourra démontrer qu’une telle formulation introduit un effet de sur-diffusion, qui élimine les oscillations (Donea et Huerta, 2003).

Une autre formulation équivalente est dite “Galerkin et sur-diffusion”. Cette dernière consiste à écrire l’équation de convection-diffusion de façon “modifiée”, ceci en ajoutant une diffusion artificielle. En 1D on peut écrire :

$$\left(-K - \frac{\vartheta v p}{2}\right) \frac{\partial^2 \mathcal{U}}{\partial x^2} + v \frac{\partial \mathcal{U}}{\partial x} = 0 \quad (4.23)$$

On démontre dans Donea et Huerta (2003) que le paramètre optimal de sur-diffusion ϑ_{opt} est donné par :

$$\vartheta_{opt} = \coth P_e - 2 \quad (4.24)$$

Enfin, une des méthodes les plus utilisées actuellement pour la stabilisation des équations de convection-diffusion est la SUPG (Donea et Huerta, 2003; Scovazzi, 2007). Elle consiste, comme toutes les autres méthodes de stabilisation récentes, à ajouter un terme de stabilisation à la formulation intégrale de l’équation aux dérivées partielles. Ainsi, la formulation intégrale



de l'équation du drapage (4.13) devient :

$$\int_{\Omega \times \mathcal{Z}} \left(\rho C_p \mathcal{U}^* \left(v \frac{\partial \mathcal{U}}{\partial x} \right) - \mathcal{U}^* \nabla (\mathbf{K} \nabla \mathcal{U}) \right) d\Omega d\mathcal{Z} + \int_{\Omega \times \mathcal{Z}} \mathcal{P}(\mathcal{U}^*) \tau R(\mathcal{U}) d\Omega d\mathcal{Z} = 0 \quad (4.25)$$

Dans le terme de stabilisation ajouté $\mathcal{P}(\mathcal{U}^*) \tau R(\mathcal{U})$ on retrouve $R(\mathcal{U})$ qui est le résidu de l'équation (4.13), τ une pondération de la stabilisation. Enfin, le terme $\mathcal{P}(\mathcal{U}^*)$ est un opérateur de la fonction test \mathcal{U}^* . La SUPG définit cet opérateur comme étant :

$$\mathcal{P}(\mathcal{U}^*) = v \nabla \mathcal{U}^* \quad (4.26)$$

A noter que pour une discrétisation 1D uniforme et linéaire, il existe une pondération τ qui donne la solution exacte du problème de convection-diffusion (Donea et Huerta, 2003). Cette pondération est donnée par :

$$\tau = \frac{p}{2v} \left(\coth(P_e) - \frac{1}{P_e} \right) \quad (4.27)$$

4.3.4 Les résultats obtenus

Afin de simuler la source de chaleur, on approxime l'intensité du "flux" du laser à l'aide d'une fonction de Gauss dans la direction "x". Dans la direction "y" on utilise un échelon. La simulation du procédé de drapage a été réalisée pour les valeurs des paramètres donnés par le tableau 4.1. Les champs de température sont simulés dans les pièces formées de 1 à 10 plis. Les orientations des fibres dans les plis sont 45/0/ - 45/ - 45/0/45/90/90/45/0 en allant du premier pli déposé au dernier pli respectivement.

ρ	1560Kg/m ³	C_p	1700J/Kg°C
Q	1000W	\mathcal{U}_0	25°C
\mathcal{U}_a	25°C	\mathcal{U}_m	25°C
\mathbf{h}_{air}	25/ρC _p	$\mathbf{h}_{rouleau}$	1000/ρC _p
\mathbf{h}_m	1500/ρC _p	\mathbf{h}_1	1000/ρC _p
\mathbf{h}_2	(0.08×10 ⁶)/ρC _p	\mathbf{h}_3	(0.5×10 ⁶)/ρC _p
L	3m	l	0.2m
e	nb × 1.35 × 10 ⁻⁴ m	nb	Nombre de plis
v	0.08m/s	$K_{//}$	5/ρC _p
K_{\perp}	0.5/ρC _p	Largeur source	12.5 × 10 ⁻³ m

tableau 4.1 – Valeurs des paramètres utilisés pour le drapage



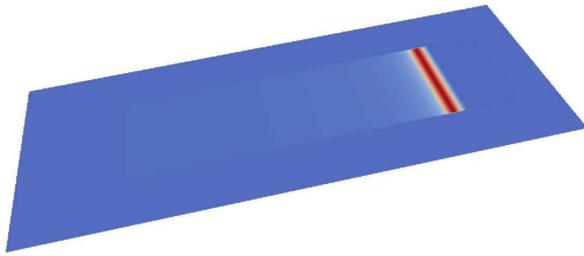


figure 4.22 – *La pièce simulée en 3D*

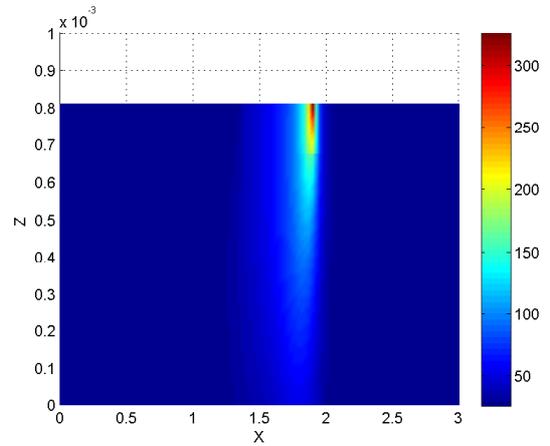


figure 4.23 – *Coupe au milieu de la pièce drapée montrant les champs de température*

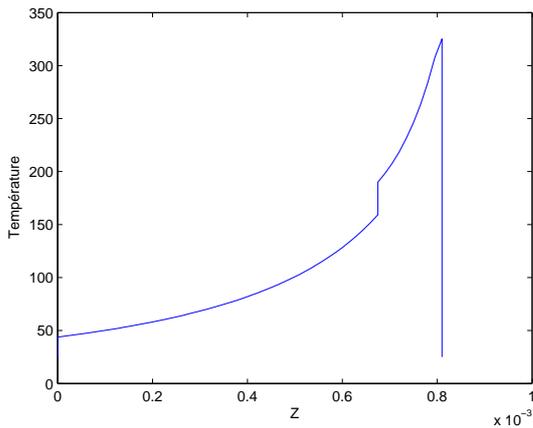


figure 4.24 – *Champs de température suivant l'épaisseur sous le jet laser*

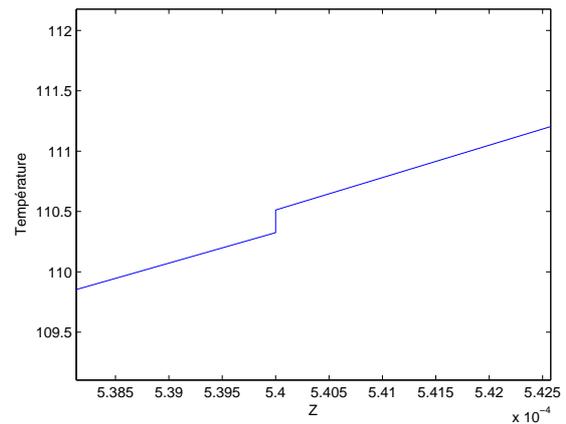


figure 4.25 – *Zoom sur une résistance interplis*

La figure 4.22 montre la pièce en 3D formée de 6 plis, alors que la figure 4.23 montre le champ de température dans une coupe verticale au milieu de cette pièce. On remarque des discontinuités du champ de température au niveau des interfaces entre les plis. La figure 4.24 illustre ces discontinuités clairement. En effet, la figure 4.24 montre le champ de température suivant l'épaisseur, pour une section située juste au-dessous de la source de chaleur. La figure 4.25 illustre un zoom sur un inter-pli dans la partie inférieure de la pièce.

Utilisant la PGD, on obtient la solution avec une grande précision dans l'épaisseur. De plus, le calcul s'achève au bout de 98,58 secondes et 16 produits de fonctions sachant qu'on utilise 150×50 noeuds pour discrétiser le domaine Ω et 102 noeuds pour discrétiser l'épaisseur \mathcal{Z} .



4.4 Simulation paramétrique de la dépose de bandes

Afin d'exploiter les capacités offertes par la PGD, nous avons résolu le problème de la dépose de bandes paramétré. En effet, les conductances inter-plis h_1 , h_2 et h_3 , la source de chaleur Q et la vitesse de dépose v ont été ajoutées comme coordonnées supplémentaires du problème. Ceci nécessite une résolution du problème 8D. Evidemment une telle solution n'aurait jamais été possible en utilisant des méthodes à maillage classiques. Dans ce qui suit, on résume d'abord le traitement utilisé pour chacun de ces paramètres afin de l'ajouter comme coordonnée supplémentaire. Dans un deuxième temps, les résultats obtenus seront présentés.

4.4.1 Le traitement des paramètres

D'abord, on cherche à écrire les opérateurs de l'équation aux dérivées partielles (4.13) sous forme séparée. Puisque cette équation est linéaire, elle s'écrit sous la forme :

$$\mathcal{L}U = \mathcal{B} \quad (4.28)$$

On choisit d'écrire l'opérateur \mathcal{L} sous la forme suivante :

$$\mathcal{L} = \sum_{i=1}^{i=nb+1} \prod_{j=1}^{j=D} \mathbf{R}_j^i + \sum_{i=2}^{i=2} \prod_{j=1}^{j=D} \mathbf{C}_j^i + \sum_{i=5}^{i=5} \prod_{j=1}^{j=D} \mathbf{A}_j^i + \prod_{j=1}^{j=D} \mathbf{S}\mathbf{V}_j + \prod_{j=1}^{j=D} \mathbf{S}\mathbf{T}_j \quad (4.29)$$

\mathbf{R}_j^i étant les opérateurs manipulant les résistances thermiques, avec nb étant le nombre de plis, D le nombre de *domains* PGD qui est égal à 7 dans notre cas ($6 \times 1D + 1 \times 2D$). Le $nb+1$ est dû aux $nb-1$ inter-plis existant auquel viennent s'ajouter aux deux premiers inter-plis qui sont décomposés en deux résistances (clef voir figure 4.19). \mathbf{C}_j^i sont les opérateurs relatifs aux conditions mixtes, dues à la convection avec l'air et le rouleau d'une part, et le moule d'autre part. \mathbf{A}_j^i sont les opérateurs relatifs au laplacien. Finalement, $\mathbf{S}\mathbf{V}$ est le terme de convection et $\mathbf{S}\mathbf{T}$ est le terme de stabilisation de l'équation d'advection-convection.

La paramétrisation des résistances thermiques inter-plis intervient dans la construction des opérateurs \mathbf{R}_j^i . Ainsi on écrit une résistance thermique \mathbf{h}_o comme étant :

$$\mathbf{h}_o = F_o(x,y) \times Z_o(z) \times h_o \quad (4.30)$$

h_o étant la valeur de la résistance, $Z_o(z)$ étant une fonction indicatrice qui est égale à 1 sur l'inter-pli correspondant à \mathbf{h}_o et zéro ailleurs. $F_o(x,y)$ est égale à un vecteur identité sauf dans le cas des deux inter-plis supérieurs où elle définit la position de la résistance dans le



plan (x,y) . Cette écriture est suffisante pour pouvoir construire les opérateurs \mathbf{R}_j^i sous formes séparées.

Le traitement de la source de chaleur variable doit apparaître dans le second membre. D'ailleurs, c'est le même traitement utilisé pour paramétrer des conditions limites de Neumann. Par suite on écrit le second membre \mathcal{B} sous forme séparée. Dans notre cas, ceci est possible en utilisant un seul produit de fonctions :

$$\mathcal{B} = \prod_{j=1}^{j=D} \mathbf{B}_j \quad (4.31)$$

Enfin, la vitesse v apparaît dans les deux termes \mathbf{SV} et \mathbf{ST} . Pour le terme de convection \mathbf{SV} , aucun souci n'est présent, le terme s'écrit sous forme séparée facilement. Par contre pour le terme de stabilisation, la vitesse apparaît dans le dénominateur, dans une cotangente hyperbolique... Alors, puisque l'écriture sous forme séparée de ce terme est compliquée, nous avons choisi de traiter la vitesse comme une constante égale à la vitesse maximale dans l'opérateur \mathbf{ST} . Ce choix implique une “*sur – stabilisation*” pour les vitesses inférieures. Par suite la marge de variation de la vitesse est relativement petite.

4.4.2 Le résultat obtenu : une solution 8D

Le calcul a été réalisé pour un nombre de plis allant de 1 à 10. Les fibres sont orientées dans l'ordre 45/0/−45/−45/0/45/90/90/45/0, respectivement du pli le plus bas au plus haut. Par suite on obtient 10 solutions, chacune en 8D. Quelques particularisations de la solution sont présentées dans les figures 4.26, 4.27, 4.28 et 4.29. Les valeurs maximales et minimales des paramètres sont données par les intervalles définis préalablement avant d'aborder le calcul. Pour une diminution du résidu d'un facteur de 100, les temps de convergence sont donnés dans le tableau 4.2.

D'après le tableau 4.2 on pourra dire que le temps de calcul est très faible relativement à la taille du calcul réalisé. De plus la solution utilise une quantité de mémoire acceptable.

D'autre part, la solution paramétrique a permis de bien illustrer l'effet de la vitesse. Plus on augmente la vitesse de dépose, plus on diminue le pic de température. La valeur maximale de la température en fonction de la vitesse, pour une combinaison donnée des paramètres (nombre de plis égal à 8, $Q = 1000W$, $h_1 = \frac{500}{\rho C_p}$, $h_2 = \frac{4 \cdot 10^4}{\rho C_p}$ et $h_3 = \frac{25 \cdot 10^4}{\rho C_p}$) est donnée par la figure 4.30. On remarque que la courbe de température maximale évolue quasi linéairement



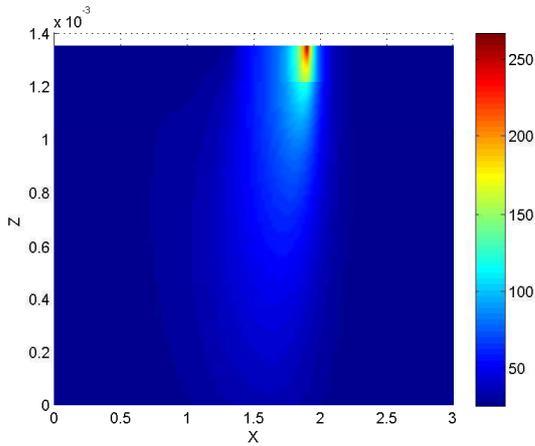


figure 4.26 – La pièce simulée en 3D à vitesse et résistance maximale

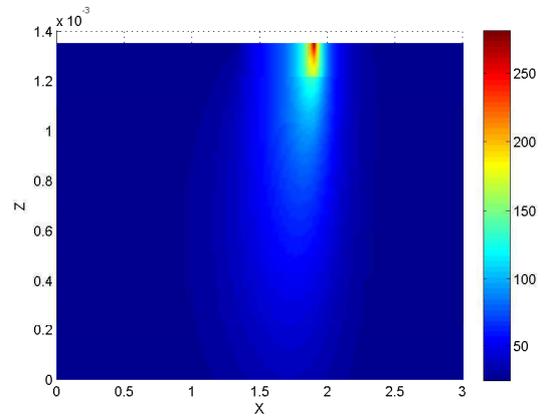


figure 4.27 – La pièce simulée en 3D à vitesse minimale et résistance maximale

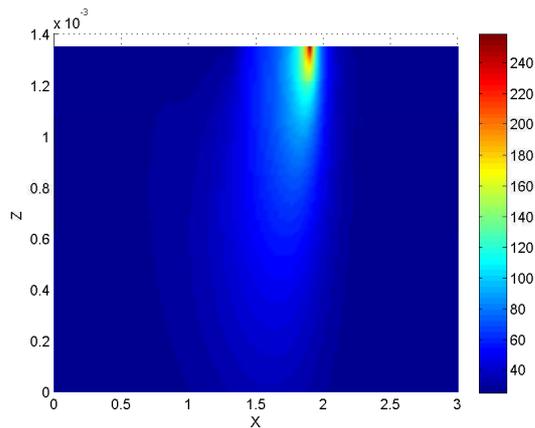


figure 4.28 – La pièce simulée en 3D à vitesse maximale et résistance minimale

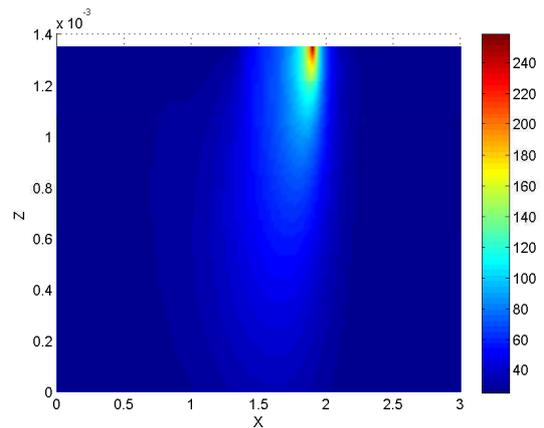


figure 4.29 – La pièce simulée en 3D à vitesse et résistance minimale

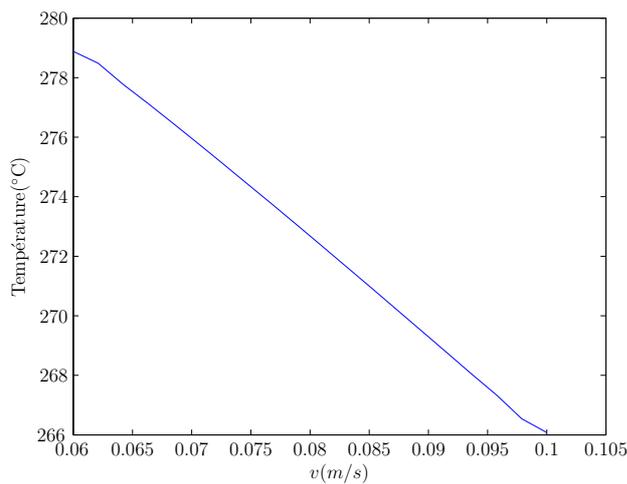
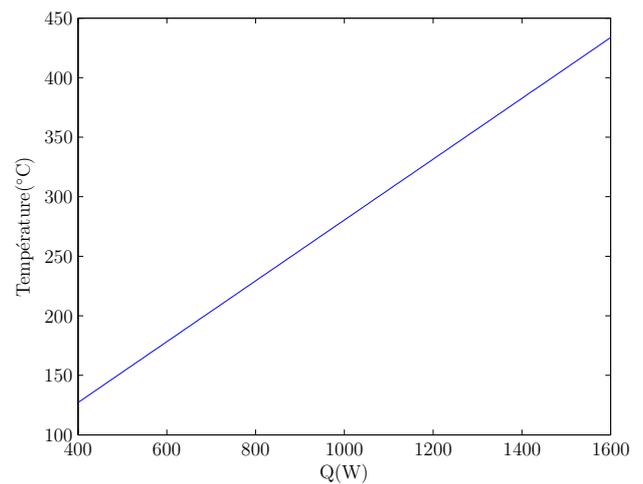
en fonction de la vitesse, par contre ceci n'est pas vrai pour tout point de la pièce.

De plus, la source de chaleur influe linéairement sur la température maximale. Ceci est illustré par la figure 4.31 pour les mêmes paramètres précédents, avec une vitesse $v = 0.08m/s$.

Enfin, l'effet de la conductance inter-plis sur la pièce n'est en revanche pas linéaire du tout. Cependant, cet effet est plutôt logarithmique. On pourra dire que l'influence de la résistivité est donc exponentielle. La figure 4.32 illustre les effets des conductances h_1 et h_2 sur la température maximale de la pièce. Evidemment h_1 influe beaucoup plus sur cette dernière.



Nombre de plis	Temps de calcul (s)	Nombre de modes
1 pli	13.38	6
2 plis	90.20	12
3 plis	189.18	19
4 plis	145.14	18
5 plis	184.37	20
6 plis	274.98	24
7 plis	276.76	25
8 plis	307.93	27
9 plis	346.47	29
10 plis	424.40	27

tableau 4.2 – *Calcul nécessaire pour converger le problème de drapage paramétré*figure 4.30 – *L'effet de la vitesse sur la température maximale de la pièce*figure 4.31 – *L'effet de la source de chaleur sur la température de la pièce*

4.4.3 Le post-traitement des solutions 8D

Les résultats de calcul ont été implémentés sur une tablette afin de faciliter la visualisation rapide des solutions. La figure 4.33 illustre quelques images de l'application créée sur cette tablette.

Cependant, vis à vis des industriels, le champ de température n'est pas le but final du calcul, mais le post traitement qui en découle. En effet, l'intérêt est de pouvoir calculer le coefficient de cicatrisation ou de dégradation.

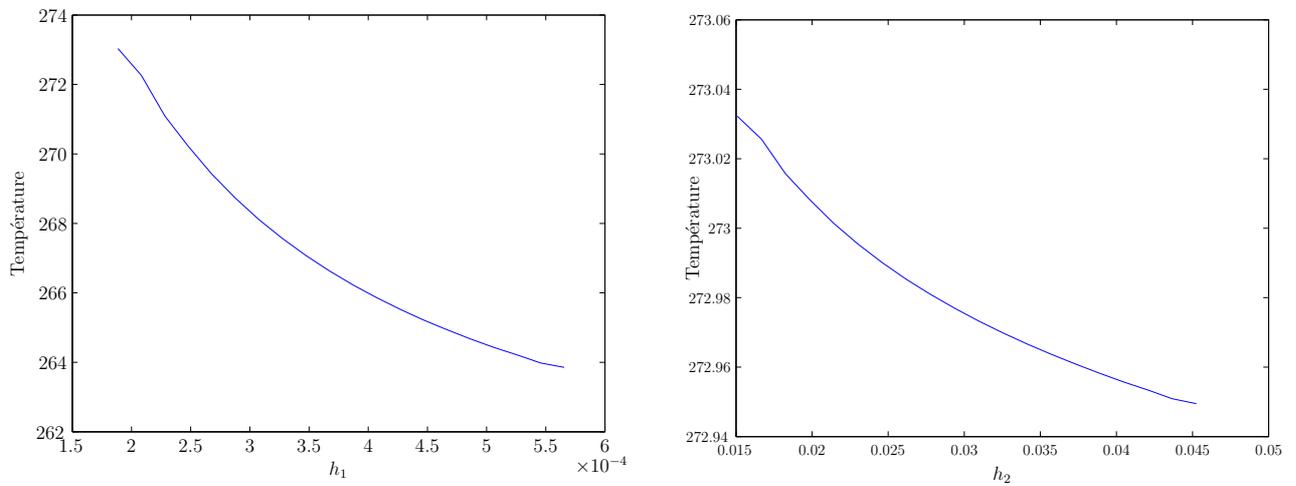


figure 4.32 – L’effet des résistances inter-plis h_1 à gauche et h_2 à droite

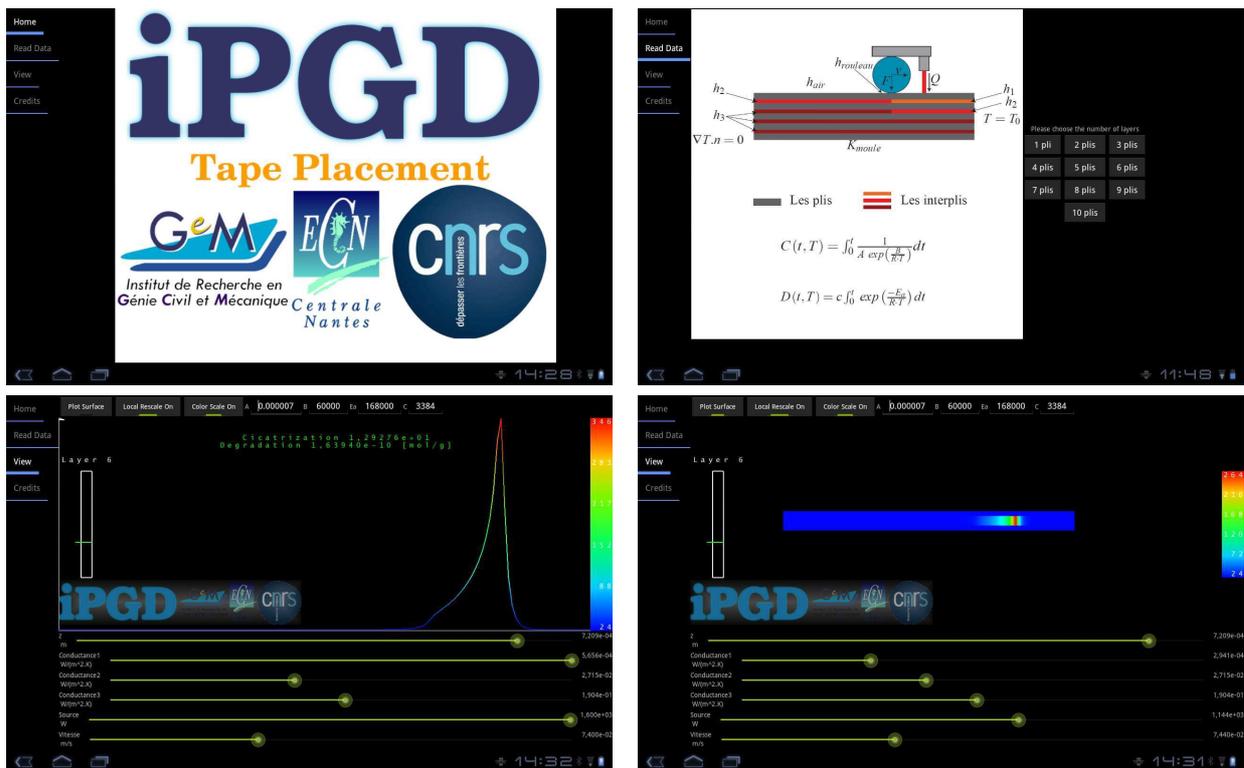


figure 4.33 – Une application tablette exploitant la solution 8D, développée par F. Bordeu

Plusieurs lois de cicatrisation ont été établies. Dans ce travail, nous utilisons la loi établie par la thèse de [Nicodeau \(2005\)](#). Alors, en notant t le temps en secondes, le degré de cicatrisation



$D(t)$ est donné par la loi suivante :

$$D(t) = \int_0^t \frac{1}{t_r} dt \quad (4.32)$$

t_r est le temps de reptation, qui dépend de l'histoire thermique $\mathcal{U}(t)$. Ce dernier est donné par la loi d'Arrhenius dépendante de la température (Kostopoulos *et al.*, 1979) :

$$t_r = A \exp\left(\frac{B}{R\mathcal{U}}\right) \quad (4.33)$$

On note que R est la constante des gaz parfaits, A et B deux constantes tel que, pour le PEEK, on a :

$$\begin{cases} A = 7 \cdot 10^{-6} SI \\ B = 60 KJ/mol \\ R = 8.32 SI \end{cases} \quad (4.34)$$

D'autre part la cicatrisation ne s'enclenche qu'à partir de la température de fusion de la matrice, et s'arrête à la température de cristallisation. Par la suite, l'intégrale (4.32) est calculée entre ces deux valeurs seulement. Finalement, une pièce est considérée cicatrisée si $D(t) \geq 1$.

D'autre part, la dégradation est définie par Nicodeau (2005) comme étant :

$$Rr(t) = C \int_0^{t_1} \exp\left(\frac{-E_a}{R \cdot \mathcal{U}}\right) dt \quad (4.35)$$

R est la constante des gaz parfaits, $C = 3387 \text{ mol}/(\text{g.s})$ et $E_a = 1.68 \times 10^5 \text{ J/mol}$. Une pièce est considérée dégradée si $Rr(t) \geq 10^{-7}$.

Par la suite, les intégrales (4.32) et (4.35) furent aussi implémentées dans l'application sur la tablette afin de pouvoir calculer facilement ces paramètres qui traduisent la qualité du procédé.

4.5 Conclusion

Dans ce chapitre nous avons présenté la possibilité d'utiliser la PGD afin d'obtenir des simulations 3D des composites stratifiés, incluant parfois de nombreuses extra-coordonnées. Les calculs réalisés sont rapides et faciles à stocker et exploiter. L'histoire thermique permet le



calcul des contraintes résiduelles et ensuite la déformée de la pièce, comme le montre la figure 4.34. Ce couplage de la thermique avec la mécanique est réalisée dans la thèse de B. Bognet.

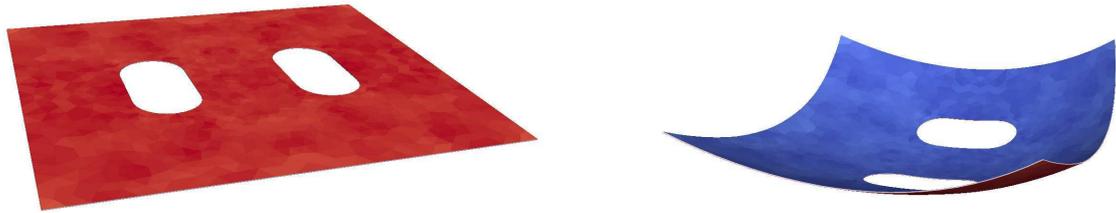


figure 4.34 – *La déformation d'une plaque due à un champ de température*

Ensuite, utilisant les calculs paramétrés réalisés, on pourra construire une enveloppe de déformation par exemple (Bognet *et al.*, 2012). La figure 4.35 montre l'enveloppe d'une pièce déformée sous champ de température suivant avec une orientation des fibres allant de -20° à $+20^\circ$. En pratique, ceci est intéressant lors des calculs d'encombrements.

D'autre part, les résultats montrés pour la simulation de dépose de bandes sont intéressants et méritent un développement plus approfondi. La simulation peut être considérée comme achevée du point de vue thermique. Cependant, un calcul mécanique approprié est nécessaire pour traduire les connaissances et calculer les contraintes résiduelles engendrées. Cela aboutira à l'amélioration des connaissances actuelles de ce procédé.

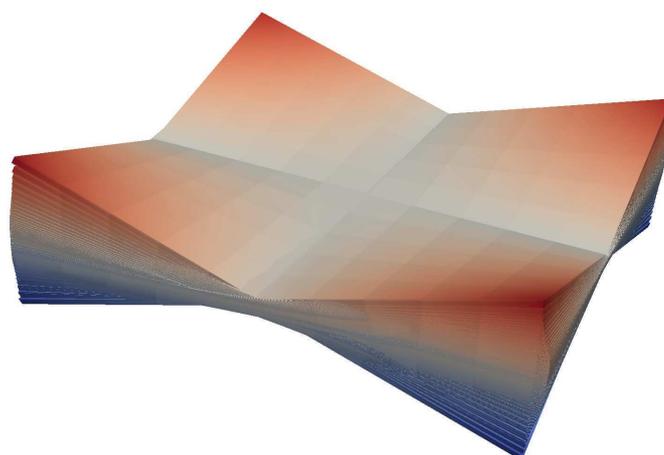


figure 4.35 – *Enveloppe d'une pièce déformée par l'élasticité thermique*

5

La PGD dans l'analyse inverse Off-line/On-line

5.1	Présentation d'un algorithme original	135
5.2	Traitement du problème de Cauchy	136
5.2.1	Problème de Cauchy 1D transitoire	138
5.2.2	Généralisation pour un modèle 2D	142
5.2.3	Discussion et perspectives	147
5.3	Résolution par la PGD dans des géométries complexes	148
5.3.1	Extension de domaine	149
5.3.2	Modélisation paramétrique de la géométrie	153
5.4	Conclusion	156

Comme en témoignent les chapitres précédents, le développement de la PGD ou “Proper Generalized Decomposition” a ouvert de nouvelles possibilités de résolution de modèles hautement multidimensionnels. En particulier, on peut calculer dans une étape dite *off-line* la solution d'un problème multidimensionnel et l'utiliser ensuite dans une étape *on-line* pour la résolution d'un problème inverse dans l'espace multidimensionnel. Un exemple a été présenté au chapitre 3 de cette thèse. Dans ce chapitre, nous allons approfondir le traitement de cette approche. Ainsi, l'approche *off-line/on-line* proposée ici permet de ne résoudre qu'une seule fois un problème multidimensionnel. Par suite, la résolution du problème inverse ne demande que l'évaluation de la solution pré-calculée. Cette évaluation peut-être donc envisagée en temps réel et sur des plateformes de calcul légères.

5.1 Présentation d'un algorithme original

Le traitement des problèmes inverses par recalage est difficile (Grédiac, 2011). En effet, chaque itération de l'algorithme correspond à une résolution du problème direct. Le nombre de résolutions peut augmenter encore si l'on effectue le calcul des sensibilités numériquement.

D'ailleurs, c'est le cas de la plupart des algorithmes de descente de gradient. Ces nombreuses résolutions constituent souvent l'essentiel du coût calcul total et peuvent constituer un facteur limitant. Plusieurs méthodes alternatives ont été développées par les chercheurs dans le domaine des problèmes inverses afin de remédier à ce problème. A titre d'exemple, on pourra citer l'utilisation des problèmes adjoints pour l'évaluation des gradients (Bonnet, 1995) et le calcul de champs virtuels (Toussaint *et al.*, 2006).

Dans cette section, on propose une méthode originale de traitement des problèmes inverses. En effet, comme la plus grande difficulté rencontrée dans les algorithmes classiques (cf. figure 5.1) réside dans la résolution des problèmes directs, nous proposons de résoudre le problème direct paramétrique off-line. L'algorithme proposé est présenté dans la figure 5.2, notons que \mathbf{d} est le vecteur contenant les paramètres du système.

En utilisant l'algorithme proposé, non seulement nous avons extrait les calculs directs de la boucle de résolution du problème inverse, mais aussi on a accès directement à tous les gradients par rapport aux paramètres. Par suite, la résolution s'accélère et le temps de calcul diminue énormément. Parfois, l'analyse en temps réel est possible. Quelques exemples de résolution sont présentés dans ce qui suit.

5.2 Traitement du problème de Cauchy

Le développement rapide des méthodes de traitement d'images renforce l'intérêt pour les problèmes aux conditions redondantes sur une partie du bord (Andrieux et Baranger, 2008). D'autre part, en essayant de résoudre des problèmes de détection de fissures en se basant sur l'écart à la réciprocité, on a besoin de connaître les données sur tout le bord. Ceci cependant est impossible dans plusieurs cas pratiques, comme dans le cas des échangeurs dans les centrales nucléaires par exemple (Andrieux et Abda, 1996). C'est ainsi, et pour beaucoup d'autres raisons, que les problèmes de complétion de données, connus classiquement sous le nom de problème de Cauchy, sont revenus au premier plan (Lesnic *et al.*, 1997). Le problème est formulé comme la recherche de conditions limites manquantes sur une partie du domaine à partir de conditions limites redondantes sur une autre partie du domaine. Les applications ne se limitent pas au traitement d'image ni à l'identification de conditions aux limites inconnues, mais s'étendent aussi aux problèmes inverses géométriques, à l'estimation de zones de contact et beaucoup d'autres (Andrieux et Baranger, 2008).

La résolution d'un problème inverse par une méthode itérative requiert traditionnellement



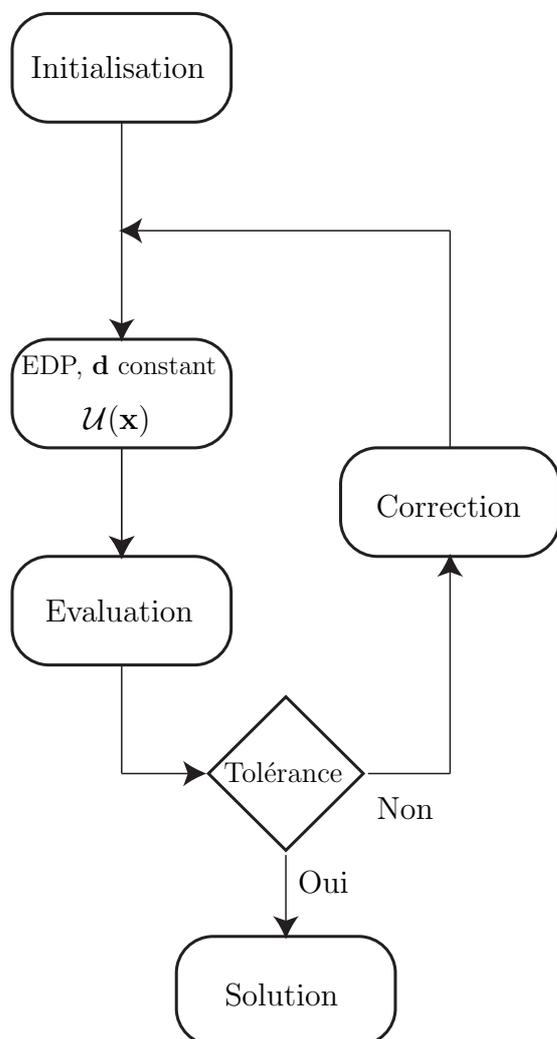


figure 5.1 – Le recalage classique, \mathbf{d} étant les paramètres

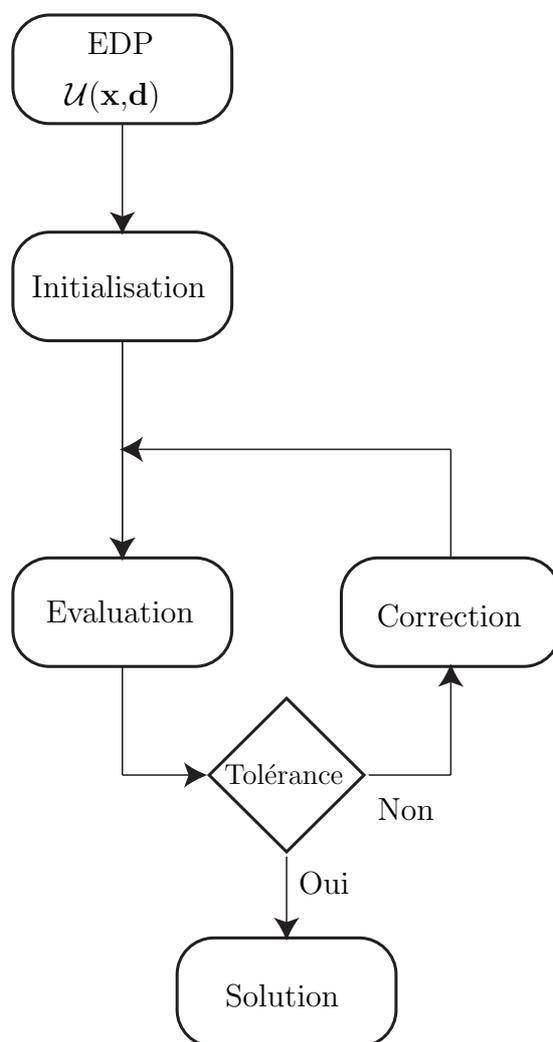


figure 5.2 – L'idée proposée

la résolution de nombreux problèmes directs. Dans cette section, on utilise la PGD afin de calculer “off-line” une solution pour toutes les conditions aux limites possibles sur la frontière du domaine. Grâce aux caractéristiques de la PGD, la prise en compte de ces dimensions supplémentaires dans le modèle se fait à un coût calcul tout à fait raisonnable (Chinesta *et al.*, 2009b). Ensuite, on utilise la solution ainsi calculée dans la résolution d’un problème inverse dans lequel on souhaite déterminer les conditions aux limites inconnues à partir de conditions aux limites redondantes. Cette étape “on-line” se fait à l’aide d’une descente de gradient qui à chaque itération ne nécessite que l’évaluation de la solution calculée par PGD et pas la résolution d’un nouveau problème direct. En particulier, on abordera le cas d’un problème thermique en régime transitoire 1D, puis celui d’un problème stationnaire 2D.



5.2.1 Problème de Cauchy 1D transitoire

Dans cette section, le modèle traité est le problème de la chaleur en régime transitoire unidimensionnel. L'équation du modèle est donnée par :

$$\frac{\partial \mathcal{U}}{\partial t} - \frac{\partial}{\partial x} \left(K \frac{\partial \mathcal{U}}{\partial x} \right) = Q, \quad (5.1)$$

où \mathcal{U} est la température, $t \in [0,1]$ le temps et $x \in [0,1]$ l'espace physique unidimensionnel. Q est une source de chaleur uniforme sur tout le domaine. De plus, on suppose connaître la température et le flux pour $x = 0$, et on cherche à retrouver la température et/ou le flux de l'autre côté du domaine.

Résolution paramétrique du problème

Afin de pouvoir se servir de l'algorithme montré dans la figure 5.2, une solution paramétrique du problème doit-être construite. Par conséquent, la solution \mathcal{U} pour toute histoire de température pour $x = 1$ et $t \in [0,1]$ doit-être calculée d'abord. Ainsi, les conditions limites se résument dans l'équation :

$$\begin{cases} \mathcal{U}(0,t) = a(t) \\ \mathcal{U}(1,t) = f(t) \\ \mathcal{U}(x,0) = 0 \end{cases} \quad (5.2)$$

Où a est la valeur de la température en $x = 0$ considérée connue. Par simplicité et sans perte de généralité, considérons $a = 0$ dans cette section. Par contre, $f(t)$ est une fonction du temps. Pour résoudre le problème, $f(t)$ est approximée linéairement par des droites sur un maillage grossier du temps. L'approximation utilisée est montrée par la figure 5.3.

Il est clair, d'après l'approximation réalisée dans la figure 5.3, que l'histoire de la température n'est donc fonction que des températures $\theta_1, \theta_2, \theta_3$ et θ_4 sur les points du maillage grossier t_1, t_2, t_3 et t_4 . Ainsi, en ajoutant ces quatre valeurs comme coordonnées supplémentaires du problème, et en résolvant l'équation (5.1) pour toutes les valeurs de ces coordonnées, on obtient la solution pour toute histoire thermique en $x = 1$, représentée par la paramétrisation illustrée dans la figure 5.3. Par la suite, on cherche la solution \mathcal{U} sous la forme :

$$\mathcal{U}(x,t,\theta_1,\theta_2,\theta_3,\theta_4) \approx \sum_{i=1}^{i=N} X_i(x) \cdot T_i(t) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \cdot \Theta_{4i}(\theta_4) \quad (5.3)$$



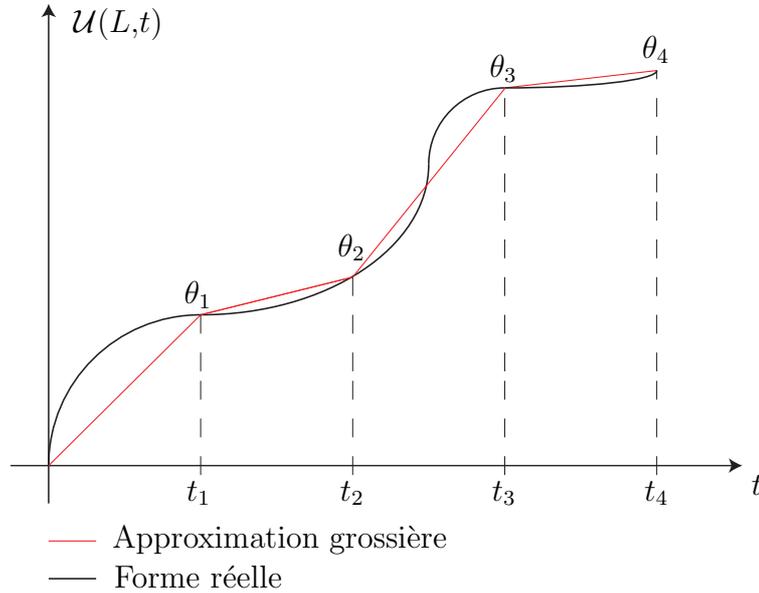


figure 5.3 – Approximation de l'histoire thermique

Pour introduire les nouvelles coordonnées dans la résolution de l'équation différentielle, on procède d'abord par un "relèvement" des conditions aux limites. Ceci se traduit par l'écriture des fonctions de forme du maillage grossier du temps sous forme de produit séparée, de la même forme que \mathcal{U} dans l'équation (5.3). Plus clairement, prenons les deux points (t_i, θ_i) et (t_{i+1}, θ_{i+1}) . Analytiquement, l'équation y_i de la droite entre ces deux points s'écrit :

$$y_i = \left[\left[\frac{\theta_{i+1} - \theta_i}{t_{i+1} - t_i} (t - t_i) + \theta_i \right] \times \chi_i \right] \quad (5.4)$$

Où χ_i est une fonction égale à 1 sur $[t_i; t_{i+1}]$ et nulle ailleurs. Enfin, la fonction de l'histoire du temps y est la somme de toutes les y_i :

$$y = \sum_{i=1}^{i=n} y_i \quad (5.5)$$

Une fois y écrite sous la même forme séparée, on va injecter ses "modes" dans le calcul de \mathcal{U} comme étant les premiers termes de la somme (5.3). Puis, on réalisera la résolution de l'équation (5.1) pour des conditions limites homogènes, puisque les conditions aux limites sont déjà satisfaites par les modes de y . La somme restera donc fidèle aux conditions limites désirées.

La résolution PGD par l'algorithme donné dans la chapitre 2 prend 7.81 secondes seulement.



La solution est donnée par 25 modes pour une réduction du résidu égale à 8.9×10^{-5} . La figure 5.4 représente la solution du problème avec $\theta_1 = 50^\circ C$, $\theta_2 = 40^\circ C$, $\theta_3 = 60^\circ C$, $\theta_4 = 70^\circ C$, ceci en considérant $K = 1$ et $Q = 50$. Pour ces mêmes valeurs, la figure 5.5 montre le flux de la température sur $x = 0$.

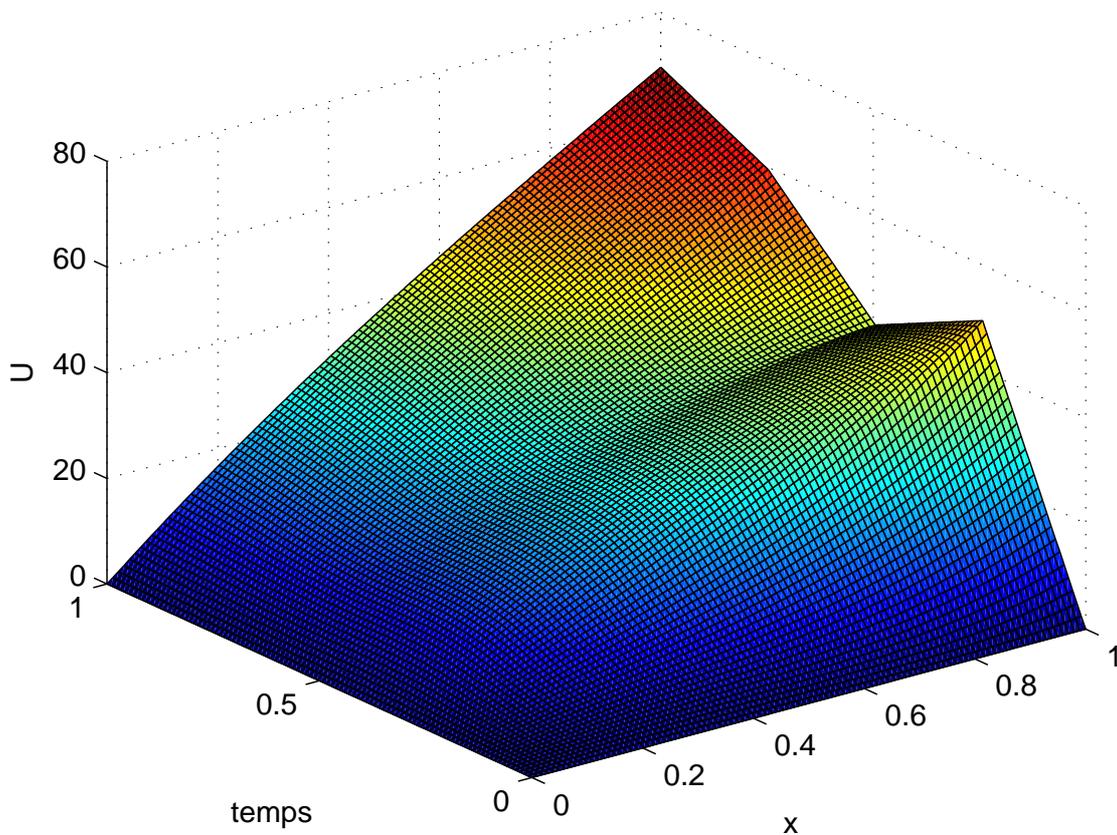


figure 5.4 – Solution pour $\theta_1 = 50$, $\theta_2 = 40$, $\theta_3 = 60$ et $\theta_4 = 70$

Calcul inverse

Ayant la solution \mathcal{U} paramétrée pour toute histoire de température définie par $(\theta_1, \theta_2, \theta_3, \theta_4)$, l'identification de l'histoire qui donne le gradient de température approprié en $x = 0$ est entamée. Ceci est réalisable sur des plateformes mobiles “on-line” grâce au faible temps et volume de calcul nécessaire à la convergence de l'algorithme. Le problème consiste donc à



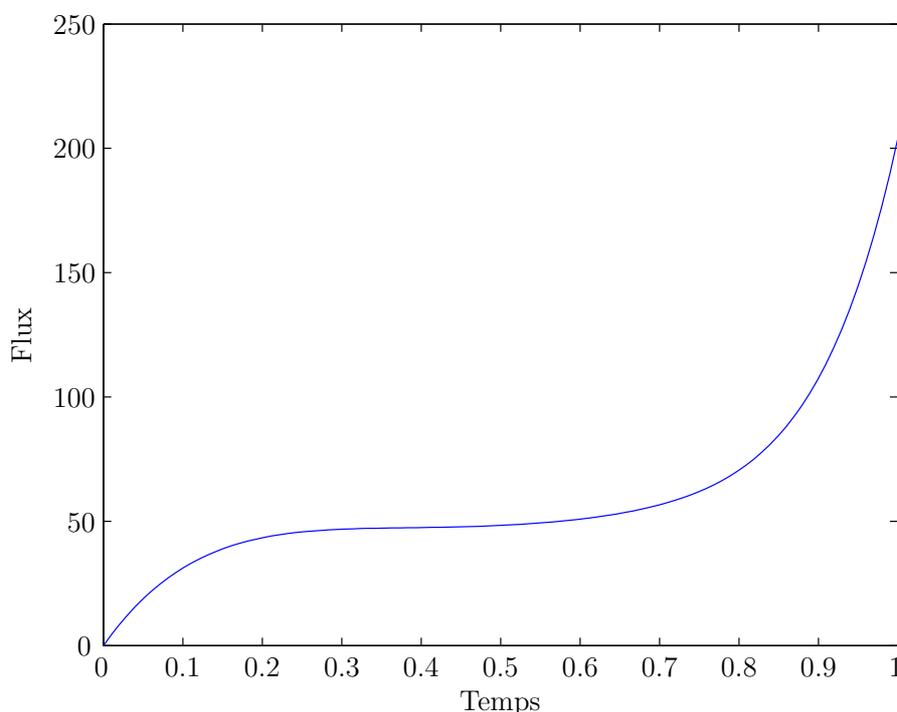


figure 5.5 – Flux en $x = 0$ pour $\theta_1 = 50$, $\theta_2 = 40$, $\theta_3 = 60$ et $\theta_4 = 70$

obtenir le minimum des fonctions coût suivantes :

$$\left\{ \begin{array}{l} C_1(\theta_1, \theta_2, \theta_3, \theta_4) = \frac{1}{2} \left[\frac{\partial \mathcal{U}(0, t_1, \theta_1, \theta_2, \theta_3, \theta_4)}{\partial x} - \frac{\partial U^{desire}(t_1)}{\partial x} \right]^2 dt \\ C_2(\theta_1, \theta_2, \theta_3, \theta_4) = \frac{1}{2} \left[\frac{\partial \mathcal{U}(0, t_2, \theta_1, \theta_2, \theta_3, \theta_4)}{\partial x} - \frac{\partial U^{desire}(t_2)}{\partial x} \right]^2 dt \\ \vdots \\ C_j(\theta_1, \theta_2, \theta_3, \theta_4) = \frac{1}{2} \left[\frac{\partial \mathcal{U}(0, t_j, \theta_1, \theta_2, \theta_3, \theta_4)}{\partial x} - \frac{\partial U^{desire}(t_j)}{\partial x} \right]^2 dt \\ \vdots \\ C_m(\theta_1, \theta_2, \theta_3, \theta_4) = \frac{1}{2} \left[\frac{\partial \mathcal{U}(0, t_m, \theta_1, \theta_2, \theta_3, \theta_4)}{\partial x} - \frac{\partial U^{desire}(t_m)}{\partial x} \right]^2 dt \end{array} \right. \quad (5.6)$$

Les t_j sont les points du maillage fin en temps, $m = 100$ étant le nombre total de points dans notre exemple. De plus, $\frac{\partial U^{desire}(t_m)}{\partial x}$ est le “flux-consigne” du problème. Enfin, on applique une stratégie de descente appropriée pour obtenir l’optimum de ces fonctions coût. Pour le cas considéré, un algorithme de Newton a été utilisé. La figure 5.6 illustre cet exemple de résolution par calcul inverse en régime transitoire. La courbe en rouge continue est la courbe du “flux-consigne”, alors que les courbes bleues tiretées sont les solutions aux itérations



successives. Le critère de convergence est :

$$e = \frac{1}{2} \int_0^1 \left[\frac{\partial \mathcal{U}(0,t,\theta_1,\theta_2,\theta_3,\theta_4)}{\partial x} - \frac{\partial U^{desire}(t)}{\partial x} \right]^2 dt < 10^{-2} \quad (5.7)$$

La convergence s'achève en cinq itérations, où on obtient $e = 1.9 \times 10^{-3}$. Le calcul aurait duré 0.4 secondes en utilisant MATLAB et un simple ordinateur portable à deux cœurs. Le “*champ-consigne*” est donné par la figure 5.7, alors que le champ de température retrouvé est donné par la figure 5.8.

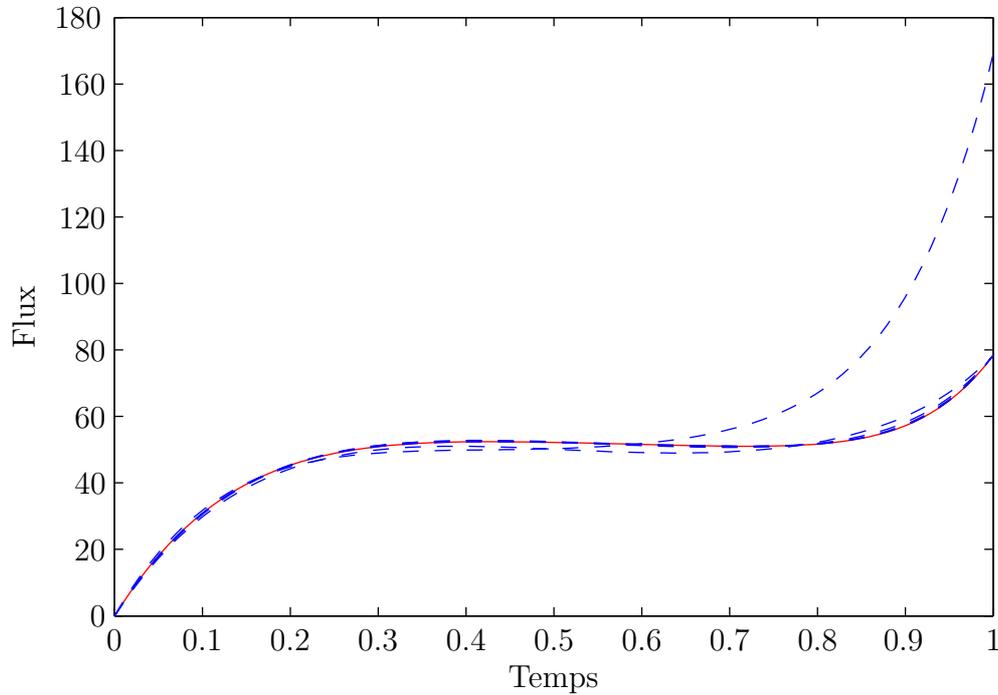


figure 5.6 – Le gradient désiré est en rouge, en bleu tireté les itérations successives

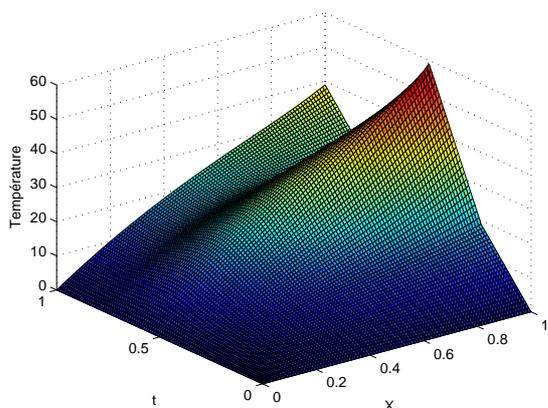
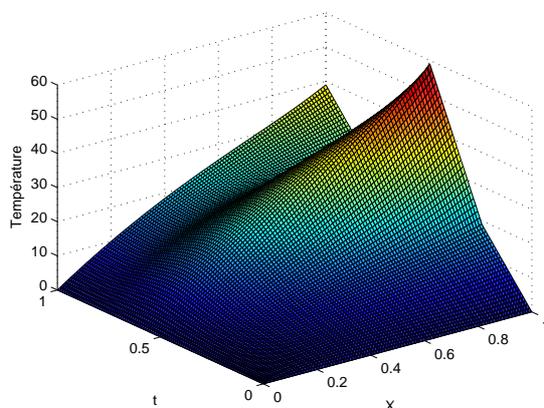
5.2.2 Généralisation pour un modèle 2D

Considérons l'équation de la chaleur en régime stationnaire dans un espace bidimensionnel Ω avec un terme source arbitraire Q :

$$-K\Delta \mathcal{U} = Q \quad (5.8)$$

\mathcal{U} étant la température et K la conductivité thermique. On considère un problème étendu pour lequel les conditions limites du problème de transfert de chaleur sont des coordonnées



figure 5.7 – *Champ de température “consigne” du problème*figure 5.8 – *Solution par calcul inverse du problème*

supplémentaires. On paramètre donc ces conditions aux limites à l’aide d’un maillage grossier sur la frontière du domaine, notons M ce maillage. On cherche la solution sous la forme suivante :

$$\mathcal{U}(x,y,\mathbf{d}) \approx \sum_{i=1}^{i=N} \left(F_i(x,y) \times \prod_{j=1}^{j=m} \mathbf{D}_{ij}(\mathbf{d}_j) \right), \quad (5.9)$$

où \mathbf{d} est le vecteur de dimension m des valeurs nodales paramétrisant les conditions limites sur le maillage grossier de la frontière et (x,y) est le vecteur des coordonnées spatiales. On choisit comme exemple le problème dans un domaine carré et de considérer une approximation linéaire par morceau sur un maillage M , avec $m = 16$ valeurs nodales pour paramétrer les conditions limites sur la frontière du domaine. On va donc déterminer à l’aide de la PGD la fonction $\mathcal{U}(x,y,d_1, \dots, d_{16})$, c’est à dire résoudre un problème en dimension 18! La solution d’un tel problème avec une méthode classique d’éléments finis est simplement impensable, et il ne faut que quelques heures de calcul sur un simple PC pour obtenir la solution à l’aide de la PGD. Le maillage spatial est illustré dans la figure 5.9.

Les conditions aux limites se résument donc par :

$$\begin{cases} \mathcal{U}(x_i,y_i,\mathbf{d}) = d_i, i = 1, \dots, m \\ \frac{\partial \mathcal{U}}{\partial n} = 0 \text{ ailleurs} \end{cases} \quad (5.10)$$

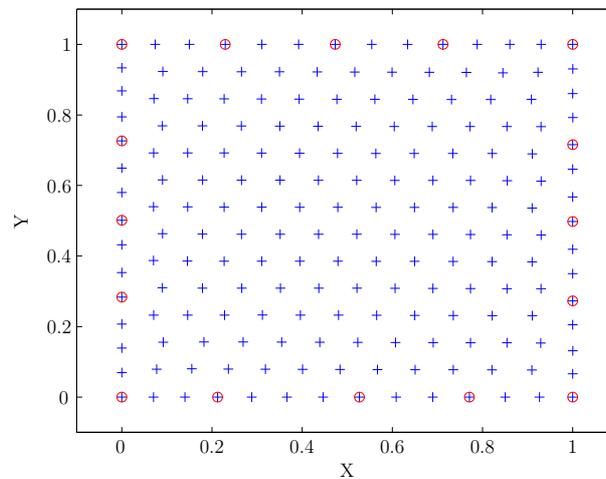


figure 5.9 – Maillages de résolution, les cercles rouges indiquent le maillage grossier M

Résolution avec la PGD

Suivant l'approche PGD, on cherche la solution du problème (5.8) sous la forme d'une somme finie de produits de fonctions d'une seule coordonnée chacune. Pour la résolution, on agit d'abord par le "relèvement" des conditions aux limites. Si $f(x, y, \mathbf{d})$ vérifie les conditions aux limites exprimées dans l'équation (5.10), il suffit de la séparer et appliquer le changement de variable :

$$V = \mathcal{U} - f, \quad (5.11)$$

on voit qu'il faut simplement résoudre le problème suivant :

$$-k\Delta V = Q + k \cdot \Delta f, \quad (5.12)$$

La figure 5.10 représente la solution du problème pour un choix arbitraire des paramètres \mathbf{d} . La convergence s'achève au bout de 137 modes et quelques heures de calcul en utilisant l'algorithme 2.4, sans la minimisation du résidu. Cependant, si on n'utilise pas la repondération des modes, la convergence ne s'achève même pas au bout de 1600 enrichissements. Ceci montre que l'algorithme utilisé n'est pas toujours optimal, sachant que la solution est écrite en 17 modes seulement utilisant les propriétés de la linéarité du problème.



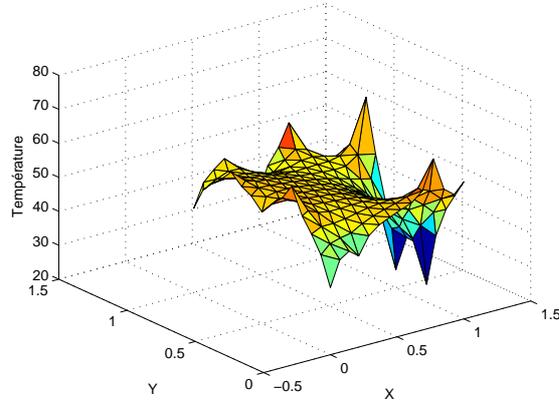


figure 5.10 – Particularisation de la solution pour un choix donné de \mathbf{d}

Analyse inverse on-line

Dans le cas 1D, le problème était plus simple. En effet, le problème de Cauchy en 2D est connu pour être mal posé (Cimetiere *et al.*, 2001). Par conséquent, pour le problème inverse montré dans cette section, on utilise la méthode dite de régularisation évanescence pour formuler une fonction coût et obtenir un problème d'optimisation bien posé (Cimetiere *et al.*, 2000). En pratique, on ajoute à la fonction coût un terme faisant intervenir la différence de température sur le bord à l'itération présente, et à l'itération précédente. Ainsi la fonction coût j à minimiser à l'itération $k + 1$ est :

$$\begin{aligned}
 j(\mathbf{d}^{k+1}) = & \int_{\Gamma_d} \frac{1}{2} (\mathcal{U}(\mathbf{d}^{k+1}, x, y) - \mathcal{U}_{mesure}(x, y))^2 d\gamma(x, y) + \\
 & \int_{\Gamma_d} \frac{1}{2} (\nabla \mathcal{U}(\mathbf{d}^{k+1}, x, y) \cdot \mathbf{n} - \mathcal{U}'_{mesure}(x, y))^2 d\gamma(x, y) + \\
 & \frac{c}{2} \int_{\Gamma} (\mathcal{U}(\mathbf{d}^{k+1}, x, y) - \mathcal{U}(\mathbf{d}^k, x, y))^2 d\gamma(x, y),
 \end{aligned} \tag{5.13}$$

où Γ_d est une partie de la frontière sur laquelle on connaît à la fois la valeur de \mathcal{U} et sa dérivée normale. Γ est l'intégralité de la frontière du domaine. On identifie \mathcal{U} sur Γ minimisant j . Pour cela, on a essayé deux méthodes de descente dans la direction du gradient.

1. L'algorithme de Levenberg-Marquardt (Marquardt, 1963; Levenberg, 1944). C'est un algorithme de descente de gradient pour la minimisation des moindres carrés non linéaires (Dennis et Schnabel, Corrected reprint of the 1983 original, 1996). Ainsi, on



écrit la fonction coût sous forme de :

$$j = \frac{1}{2} \sum_{i=1}^{i=n} (r_i^2). \quad (5.14)$$

La fonction j contient trois parties, et on cherche à minimiser les trois à la fois. Pour écrire j sous la forme donnée par l'équation (5.14), on va définir une nouvelle fonction J écrite comme suit :

$$\begin{aligned} J(\mathbf{d}^{k+1}) = & \frac{1}{2} \left(\int_{\Gamma_d} (\mathcal{U}(\mathbf{d}^{k+1}, x, y) - \mathcal{U}_{mesure}(x, y))^2 d\gamma(x, y) - 0 \right)^2 + \\ & \frac{1}{2} \left(\int_{\Gamma_d} (\nabla \mathcal{U}(\mathbf{d}^{k+1}, x, y) \cdot \mathbf{n} - \mathcal{U}'_{mesure}(x, y))^2 d\gamma(x, y) - 0 \right)^2 + \\ & \frac{1}{2} \left(c \int_{\Gamma} (\mathcal{U}(\mathbf{d}^{k+1}, x, y) - \mathcal{U}(\mathbf{d}^k, x, y))^2 d\gamma(x, y) - 0 \right)^2 \end{aligned} \quad (5.15)$$

La matrice jacobienne s'écrit alors :

$$\mathbf{J} = \begin{bmatrix} \frac{\partial r_1}{\partial d_1} & \frac{\partial r_1}{\partial d_2} & \cdots & \frac{\partial r_1}{\partial d_m} \\ \frac{\partial r_2}{\partial d_1} & \frac{\partial r_2}{\partial d_2} & \cdots & \frac{\partial r_2}{\partial d_m} \\ \frac{\partial r_3}{\partial d_1} & \frac{\partial r_3}{\partial d_2} & \cdots & \frac{\partial r_3}{\partial d_m} \end{bmatrix} \quad (5.16)$$

On écrit de plus le gradient et une évaluation de la matrice Hessienne respectivement sous les formes suivantes (Marquardt, 1963) :

$$\nabla J = \mathbf{J}^T \mathbf{r}, \quad (5.17)$$

$$\nabla^2 J = \mathbf{J}^T \mathbf{J} + \sum_{j=1}^m \mathbf{r}_j \nabla^2 \mathbf{r}_j. \quad (5.18)$$

On note que les r_i ainsi que leurs dérivées sont évalués directement à l'aide de la solution PGD calculée a priori. Le problème itératif de minimisation peut donc s'écrire comme suit :

$$\mathbf{d}^{k+1} = \mathbf{d}^k - [\mathbf{J}^T \mathbf{J} + \lambda I]^{-1} \times \mathbf{J}^T \mathbf{r}. \quad (5.19)$$

2. L'algorithme de Newton quadratique. Bien qu'il soit plus lent en général, il donne une convergence plus rapide en nombre d'itérations dans le cas ici traité. Il consiste à écrire l'approximation de Taylor, au voisinage d'un point, de la dérivée de la fonction coût j ,



la fonction de descente sera donnée par :

$$\mathbf{d}^{n+1} = \mathbf{d}^n - \left((\nabla^2 j(\mathbf{d}^n))^{-1} \times \nabla j(\mathbf{d}^n) \right) \quad (5.20)$$

En pratique, on observe une convergence très rapide de la méthode vers un minimum de la fonction coût. Un exemple de solution est illustré dans la figure 5.11.

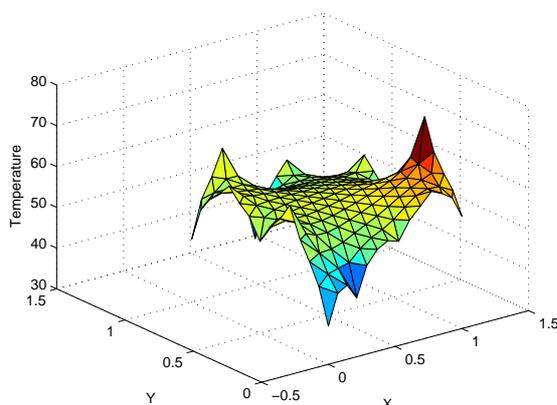


figure 5.11 – *Solution reconstruite après identification des conditions aux limites*

5.2.3 Discussion et perspectives

Dans cette section nous avons proposé une nouvelle approche pour la résolution des problèmes avec des conditions aux limites redondantes sur une partie de la frontière du domaine et en l'absence de toute condition dans la partie complémentaire. Pour parvenir à la résolution, nous proposons de résoudre le modèle off-line pour toute condition aux limites. Puis, dans l'étape on-line, il n'y a plus qu'à déterminer les conditions inconnues à l'aide de celles redondantes. Cette approche permet l'identification inverse rapide (on-line) et avec des moyens de calcul très légers.

Cependant, les applications montrées restent basiques. En effet, les problèmes inverses sont généralement mal posés et sont très sensibles aux erreurs de mesure (Han et Reinhardt, 1997), alors que les applications montrées ici considèrent les mesures parfaites. Des tests sur la méthode PGD maniant l'utilisation des méthodes de stabilisation des bruits de mesure doivent être effectués pour valider l'algorithme proposé.

5.3 Résolution par la PGD dans des géométries complexes

La PGD se base principalement sur la séparation des coordonnées impliquées dans un modèle. Dans ce sens, nous séparons des coordonnées d'espace, de temps, ou d'autres paramètres. Ces paramètres peuvent être des chargements, des conditions initiales, des paramètres matériaux ou même géométriques.

Dans cette section, l'analyse de la construction de solutions définies dans des géométries complexes est réalisée. Une analyse similaire a déjà été traitée par [Gonzalez *et al.* \(2009\)](#). Dans ce qui suit, deux stratégies différentes sont testées. La première consiste à immerger le domaine ω complexe dans un autre rectangulaire Ω (cf. figure 5.12), donc adaptée à une représentation séparée. A noter que dans ce cas les conditions limites sur le bord du grand domaine Γ sont inconnues, et sont recherchées à partir du bord de ω , notons ce dernier γ . La deuxième approche consiste à considérer des paramètres géométriques définissant le domaine comme des coordonnées du problème. Ainsi, nous résolvons le modèle pour toutes valeurs de ces paramètres. Le résultat sera donc un problème multidimensionnel dans lequel on calculera la solution pour toute forme géométrique définie par la même paramétrisation. Les points forts et faibles de chacune de ces deux approches seront montrés.

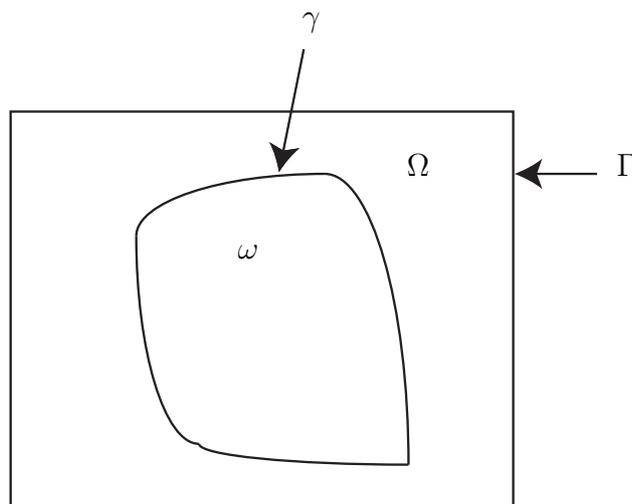


figure 5.12 – Illustration de l'extension de domaine

5.3.1 Extension de domaine

Considérons d'abord un modèle défini dans un domaine complexe ω . Nous étendons ensuite ce dernier à un autre domaine hexaédrique, notons le par Ω . Les bords de ω et Ω sont nommés respectivement γ et Γ .

Considérons le problème défini par :

$$\begin{cases} \Delta u = 0 & \text{dans } \omega \\ u = u_d & \text{sur } \gamma \end{cases} \quad (5.21)$$

Une équation plus générale sera écrite comme suit:

$$Lu = 0 \text{ dans } \omega \quad (5.22)$$

avec $u(\mathbf{x} \in \gamma_1) = f$ et $\frac{\partial u}{\partial n}(\mathbf{x} \in \gamma_2) = g$. Dans l'équation (5.22), on note L un opérateur elliptique (à coefficients constants). γ_1 et γ_2 forment une partition de γ , tandis que f et g sont deux fonctions connues, suffisamment régulières. La solution de l'équation (5.21), respectivement (5.22), est la restriction sur ω d'une fonction U définie sur Ω , qui vérifie la relation $\Delta U = 0$, respectivement $LU = 0$. Cette considération exclue les fonctions admettant des singularités au voisinage de γ . A remarquer que cette hypothèse n'est pas très restrictive du fait que les solutions singulières au voisinage du bord ne sont pas nombreuses.

Considérons H un espace de Hilbert défini par : $H(\Omega) = \{U : \Omega \rightarrow \mathbb{R} / \Delta U = 0 \text{ in } \Omega\}$, $u = U|_{\omega}$ est solution de l'équation (5.21) si et seulement si :

$$\left\{ U = \operatorname{argmin}_{V \in H(\Omega)} \{ \|V - u_d\|_{\gamma}^2 \} \right. \quad (5.23)$$

On note $\|\cdot\|$ une norme associée à $H(\gamma)$, qui est un espace d'Hilbert muni du produit scalaire $(\cdot|\cdot)_{\gamma}$. Deux exemples possibles de $H(\gamma)$ sont :

$$H(\gamma) = L^2(\gamma), (u|v) = \int_{\gamma} uvd\gamma \quad (5.24)$$

$$H(\gamma) = H^1(\gamma), (u|v) = \int_{\gamma} uvd\gamma + \int_{\gamma} \frac{\partial u}{\partial s} \frac{\partial v}{\partial s} d\gamma \quad (5.25)$$

Le problème défini par l'équation (5.21) est bien posé. La solution de ce dernier définit la valeur de u sur γ , mais aussi la valeur de ses dérivées normales. Par la suite, le calcul de U dans



Ω (et donc ω) définit un problème de Cauchy, donc un problème mal posé (Han et Reinhardt, 1997). Afin de résoudre ce problème, on utilise la régularisation évanescence (Cimetiere *et al.*, 2005). Cette régularisation procède par itération jusqu'à convergence de la fonctionnelle suivante :

$$U^{K+1} = \operatorname{argmin}_{V \in H(\Omega)} \{ \|V - u_d\|_{\gamma}^2 + c \|V - U^K\|_{\Gamma}^2 \}, \quad (5.26)$$

avec c une constante positive suffisamment petite et $U^0 = 0$.

Calcul d'une solution paramétrique

Afin de construire une solution générale paramétrique, on commence par considérer M_n nœuds sur Γ , ayant les coordonnées \mathbf{x}_r , $r = 1, \dots, M_n$, capable de décrire $U|_{\Gamma}$ précisément. De plus, nous calculons les solutions $U^r(\mathbf{x})$ du problème :

$$\begin{cases} \Delta U = 0 \\ U^r = \delta_{rs}, \forall r, s = 1, \dots, M_n \end{cases} \quad (5.27)$$

Etant Ω rectangulaire, nous pouvons séparer les coordonnées spatiales en utilisant la PGD, ainsi la solution s'écrit sous la forme suivante :

$$U^r(x, y, z) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \quad (5.28)$$

Considérant la linéarité du problème posé, nous écrivons la solution paramétrique du problème $\Delta U = 0$ comme étant :

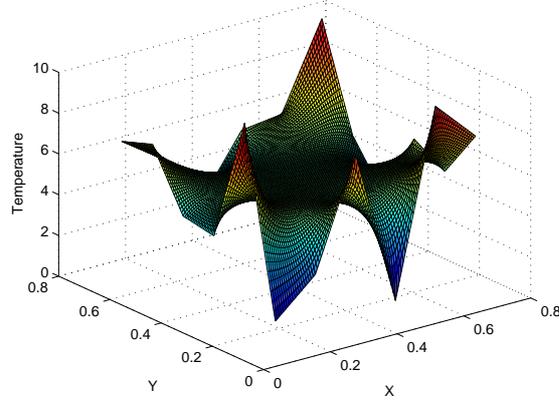
$$U(\mathbf{x}) = \sum_{i=1}^{M_n} \alpha_i \cdot U^i(\mathbf{x}) \quad (5.29)$$

Un exemple de combinaison de α_i est donné par la figure 5.13. Les coefficients α_i avec $i = 1, \dots, M_n$ sont calculés de façon à satisfaire l'équation (5.26). Un algorithme de minimisation est donc nécessaire afin d'identifier la bonne solution.

Identification de la solution

Afin de calculer les coefficients α_i optimaux, nous considérons d'abord m_n nœuds sur γ , de coordonnées \mathbf{x}_s , $s = 1, \dots, m_n$, permettant de décrire précisément u_d . De plus, nous



figure 5.13 – Exemples de solution pour différents α_i

définissons les résidus r_1^s et r_2^t à l'itération k , relatifs à la solution U^{k+1} et définis comme :

$$\begin{cases} r_1^s = (U^{k+1}(\mathbf{x}_s) - u_d(\mathbf{x}_s)) & s = 1, \dots, m_n \\ r_2^t = \sqrt{c} \cdot (U^{k+1}(\mathbf{X}_t) - U^k(\mathbf{X}_t)) & t = 1, \dots, M_n \end{cases} \quad (5.30)$$

Ce qui donne une fonction coût $C(\alpha_1, \dots, \alpha_{M_n})$ à minimiser qui s'écrit :

$$C(\alpha_1, \dots, \alpha_{M_n}) = \sum_{s=1}^{m_n} (r_1^s)^2 + \sum_{t=1}^{M_n} (r_2^t)^2 \quad (5.31)$$

La minimisation est par suite réalisée par l'algorithme de Levenberg - Marquardt. A noter que les composantes du jacobien \mathbf{J} sont maintenant simples à exprimer :

$$\begin{cases} \frac{\partial r_1^s}{\partial \alpha_i} = U^i(\mathbf{x}_s) \text{ pour } s = 1, \dots, m_n \\ \frac{\partial r_2^t}{\partial \alpha_i} = \sqrt{c} \cdot \delta_{ti} \cdot U^i(\mathbf{x}_t) \text{ pour } t = 1, \dots, M_n \end{cases} \quad (5.32)$$

Finalement, une itération de l'algorithme de descente s'écrit :

$$\mathbf{U}^{k+1} = \mathbf{U}^k - [\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}]^{-1} \cdot \mathbf{J}^T \mathbf{r}. \quad (5.33)$$

λ est un paramètre qui évolue dans l'algorithme afin d'améliorer la convergence et \mathbf{r} le vecteur résidu.



Résultats numériques

Afin de tester l'algorithme proposé, nous avons abordé les essais numériques en calculant la solution de l'équation (5.21) dans ω , et les nœuds utilisés pour la représentation de $\gamma \equiv \partial\omega$ détaillés dans la figure 5.14. La figure 5.15 compare les courbes u_d et $U|_\gamma$. L'erreur relative entre les deux courbes utilisant la norme L_2 est inférieure à 1% après trois itérations (17 ms).

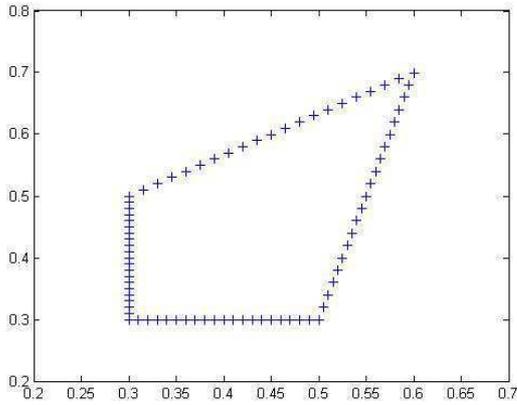


figure 5.14 – Le maillage de γ

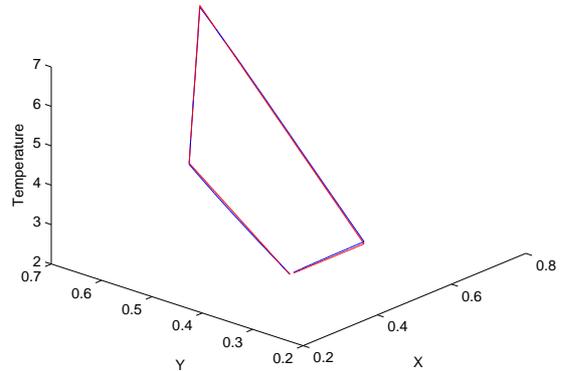


figure 5.15 – u_d (en rouge) contre $U|_\gamma$ (en bleu)

Un domaine ω plus complexe est illustré par la figure 5.16. De plus, les courbes u_d et $U|_\gamma$ sont comparées dans la figure 5.17. Dans ce cas, l'erreur relative est de l'ordre de 1% après 3 itérations (21 ms). Le calcul pour les deux cas a été réalisé sur un PC portable à deux cœurs (2.13 GHz par cœur).

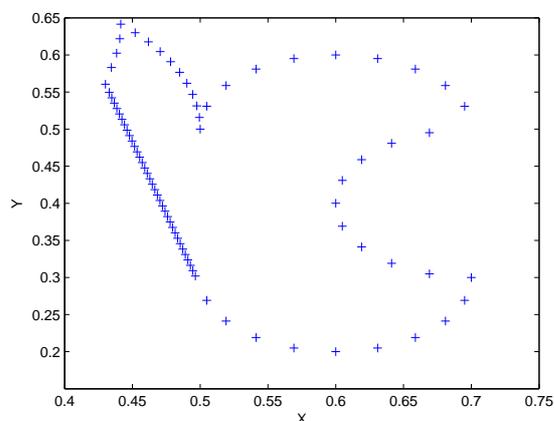
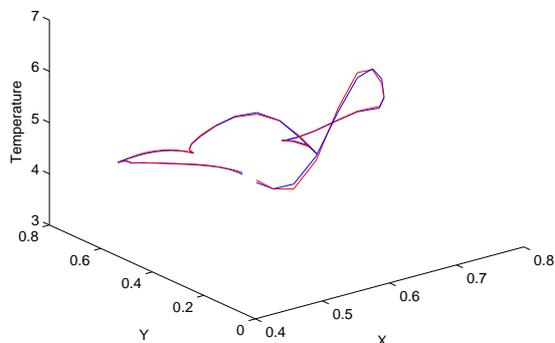
Ainsi, nous concluons que l'algorithme proposé fonctionne bien pour les conditions limites de Dirichlet. Alors nous avons appliqué la même stratégie pour un problème plus complexe, comportant des conditions de bords essentielles (Dirichlet) ainsi que des conditions naturelles (Neumann). Considérons d'abord une partition de γ , $\gamma = \gamma_1 \cup \gamma_2$. La figure 5.18 illustre la partition utilisée. Dans cette dernière, les nœuds encerclés en rouge sont les conditions essentielles de Dirichlet, sur γ_2 . Tandis que le reste du bord est noté γ_1 , avec les conditions aux bords définies par :

$$\frac{\partial u}{\partial n} = h(u - u_{ext}), \quad (5.34)$$

avec u_{ext} la température ambiante considérée constante. On sait aussi que le champ u vérifie :

$$\Delta u = 0 \quad (5.35)$$



figure 5.16 – Un deuxième exemple : Maillage de γ figure 5.17 – Un deuxième exemple : u_d (en rouge) contre $U|_{\gamma}$ (en bleu)

Dans ce cas, le problème régularisé s'écrit comme étant :

$$\begin{aligned}
 U^{k+1} = \operatorname{argmin}_{V \in H} \{ & \|V - u_d\|_{\gamma_2}^2 + \\
 & + \|\frac{\partial V}{\partial n} - h(V - u_{ext})\|_{\gamma_1}^2 + \\
 & + c\|V - U^k\|_{\Gamma}^2 \}
 \end{aligned} \tag{5.36}$$

La solution obtenue est donnée par la figure 5.19. L'erreur obtenue par la norme L_2 , calculée sur le bord, est inférieure à 1%.

5.3.2 Modélisation paramétrique de la géométrie

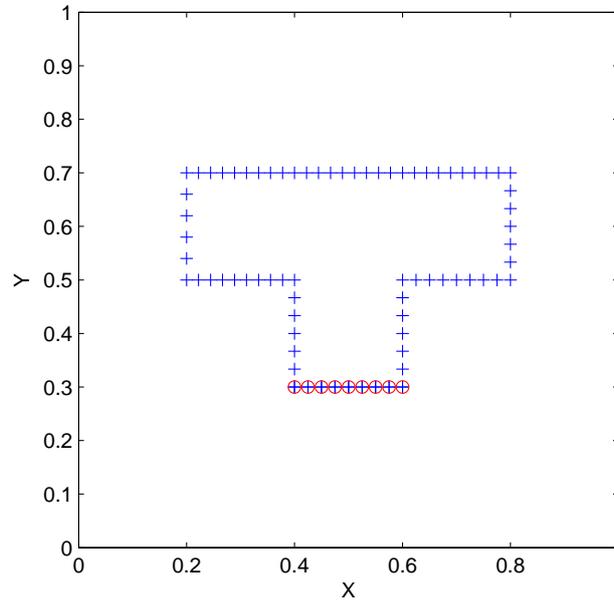
On revient au problème modèle défini dans ω , en considérant à présent un terme source :

$$-\Delta u = 1 \tag{5.37}$$

Considérons le mapping permettant de passer des coordonnées réelles (x,y) au domaine de référence (r,s) . A partir du domaine original ω on obtient ainsi un domaine hexaédrique η . Dans ce dernier, la séparation des coordonnées spatiales devient triviale (voir la figure 5.20). La transformation de domaine, ou le mapping, s'écrit alors :

$$\begin{cases} x = r + \lambda_1 \cdot r \cdot s \\ y = s + \lambda_2 \cdot r \cdot s \end{cases} \tag{5.38}$$



figure 5.18 – Partition de γ

Par suite, le jacobien de la transformation s'écrit :

$$\mathbf{J} = \begin{pmatrix} 1 + \lambda_1 \cdot s & \lambda_2 \cdot s \\ \lambda_1 \cdot r & 1 + \lambda_2 \cdot r \end{pmatrix} \quad (5.39)$$

Cependant, ce qui nous intéresse est l'inverse de ce jacobien \mathbf{J} , afin de pouvoir transformer les équations du problème (5.37) du domaine ω au domaine η . Alors, l'inverse du jacobien \mathbf{J}^{-1} s'écrit comme étant :

$$\mathbf{J}^{-1} = \frac{1}{r \cdot \lambda_2 + s \cdot \lambda_1 + 1} \begin{pmatrix} 1 + \lambda_2 \cdot r & -\lambda_2 \cdot s \\ -\lambda_1 \cdot r & 1 + \lambda_1 \cdot s \end{pmatrix} \quad (5.40)$$

Toutefois, nous cherchons à résoudre un problème multidimensionnel et trouver une solution paramétrique $u(r,s,\lambda_1,\lambda_2)$. Ainsi, une représentation séparée est obligatoire, comme nous l'avons expliqué déjà, afin d'éviter la malédiction de la multidimensionnalité. Ainsi, nous cherchons u sous la forme suivante :

$$u(r,s,\lambda_1,\lambda_2) \approx \sum_{i=1}^{i=N} R_i(r) \cdot S_i(s) \cdot \Lambda_i^1(\lambda_1) \cdot \Lambda_i^2(\lambda_2) \quad (5.41)$$

En revanche, afin de pouvoir calculer u , nous devons écrire l'inverse du jacobien \mathbf{J}^{-1} sous



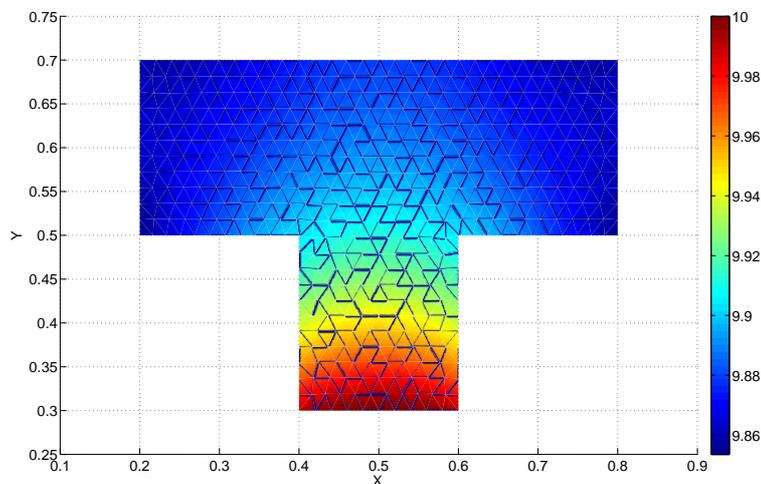
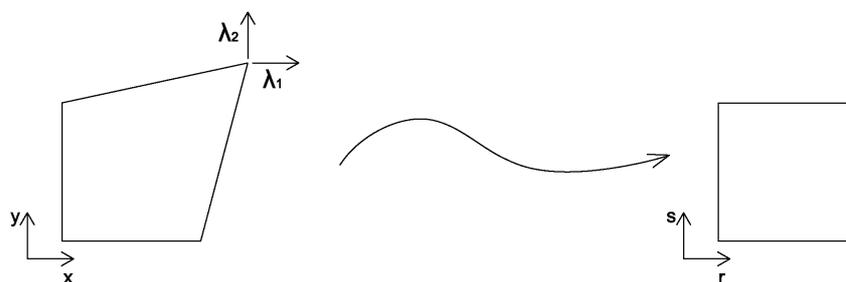
figure 5.19 – La solution calculée avec $u_d = 10$ et $u_{ext} = 0$.

figure 5.20 – Mapping du domaine paramétrique

forme séparée. Une possibilité est d'utiliser la "High Order SVD" (Kolda et Bader, 2009). Cependant, à haute dimensionnalité ce travail devient fastidieux. Une autre possibilité serait d'écrire le jacobien sous la forme suivante :

$$\mathbf{J} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} \lambda_1 \cdot s & \lambda_2 \cdot s \\ \lambda_1 \cdot r & \lambda_2 \cdot r \end{pmatrix} = \mathbf{I} + \mathbf{D} \quad (5.42)$$

Ensuite, nous développons \mathbf{J}^{-1} au premier ordre :

$$\mathbf{J}^{-1} \approx \mathbf{Q} = \mathbf{I} - \mathbf{D} \quad (5.43)$$



La forme faible de l'équation (5.37) devient :

$$\int_{\eta} (\nabla_{rs} u^*)^T \cdot \mathbf{Q}^T \cdot \mathbf{Q} \cdot \nabla_{rs} u \cdot \det(\mathbf{J}) \, d\eta = \int_{\eta} 1 \cdot \det(\mathbf{J}) \, d\eta \quad (5.44)$$

Cette écriture prend en considération la transformation $\omega \rightarrow \eta$. Après développement, l'équation (5.44) devient :

$$\int_{\eta} \frac{\partial u^*}{\partial r} \frac{\partial u}{\partial r} M_{11} + \frac{\partial u^*}{\partial r} \frac{\partial u}{\partial s} M_{12} + \frac{\partial u^*}{\partial s} \frac{\partial u}{\partial r} M_{21} + \frac{\partial u^*}{\partial s} \frac{\partial u}{\partial s} M_{22} d\eta = \int_{\eta} u^* \cdot \det(\mathbf{J}) d\eta \quad (5.45)$$

La matrice \mathbf{M} est définie par :

$$\mathbf{M} = \mathbf{Q}^T \cdot \mathbf{Q} \cdot \det(\mathbf{J}) \quad (5.46)$$

Ainsi, les termes de la matrice \mathbf{M} sont donnés par :

$$\begin{cases} M_{11} = 1 + \lambda_2 r - \lambda_1 s - 2\lambda_1 \lambda_2 r s + 2\lambda_1^2 \lambda_2 r s^2 + 2\lambda_1^3 s^3 \\ M_{12} = M_{21} = -\lambda_2 r - \lambda_2^2 r^2 - \lambda_1 s + 2\lambda_1 \lambda_2^2 r^2 s - \lambda_1^2 s^2 + 2\lambda_1^2 \lambda_2 r s^2 \\ M_{22} = 1 - \lambda_2 r + 2\lambda_2^3 r^3 + \lambda_1 s - 2\lambda_1 \lambda_2 r s + 2\lambda_1 \lambda_2^2 r^2 s \end{cases} \quad (5.47)$$

Maintenant, nous appliquons l'algorithme PGD présenté dans le chapitre 2 de façon classique afin d'obtenir la représentation séparée de la solution paramétrique $u(r, s, \lambda_1, \lambda_2)$.

Exemple numérique

Dans ce qui suit, nous considérons un domaine carré η , avec λ_1 et $\lambda_2 \in [-0.5, 0.5]$. Les figures 5.21 et 5.22 montrent les solutions $u(r, s, \lambda_1 = -0.3, \lambda_2 = -0.3)$ et celle obtenue en utilisant les éléments finis classiques $u^{EF}(x, y)$ dans ω respectivement. Les figures 5.23 et 5.24 illustrent des résultats similaires pour $\lambda_1 = \lambda_2 = 0.25$. Les erreurs relatives engendrées par cette méthode, pour toutes les valeurs de λ_1 et λ_2 dans le domaine $[-0.5, 0.5]$ sont représentées par la figure 5.25.

5.4 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle approche pour la résolution des problèmes de complétion de données. Cette idée consiste à utiliser des solutions paramétriques précalculées pour résoudre "On-line" des problèmes inverses. Puisque les solutions précalculées sont paramétrées, nous avons souvent aussi accès aux dérivées par rapport aux paramètres.



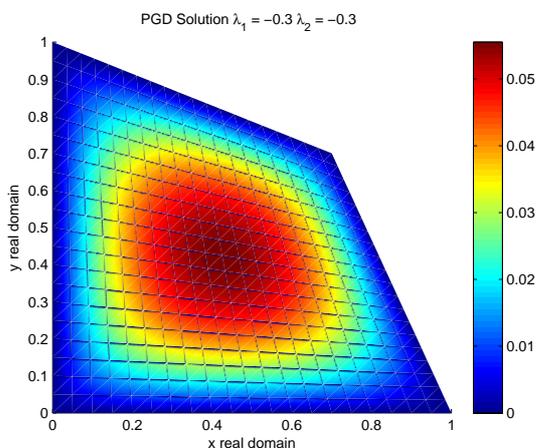


figure 5.21 – Solution PGD pour $\lambda_1 = \lambda_2 = -0.3$

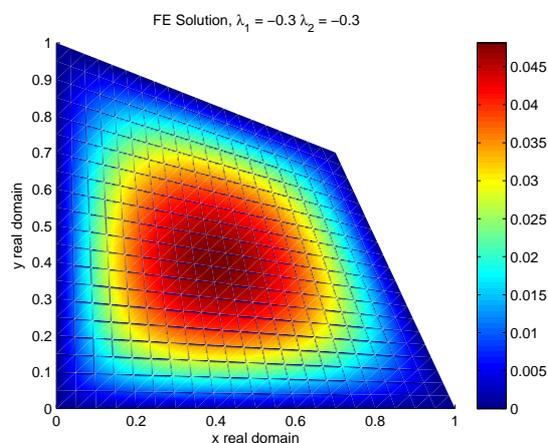


figure 5.22 – Solution éléments finis

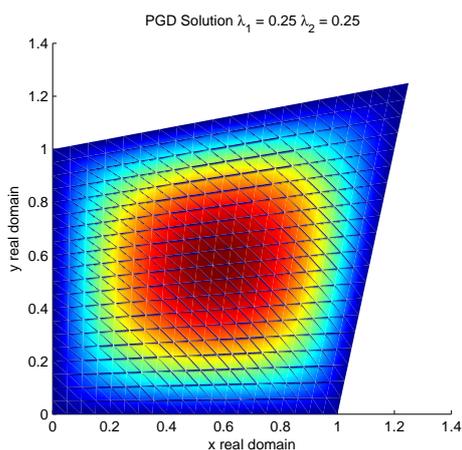


figure 5.23 – Solution PGD pour $\lambda_1 = \lambda_2 = 0.25$

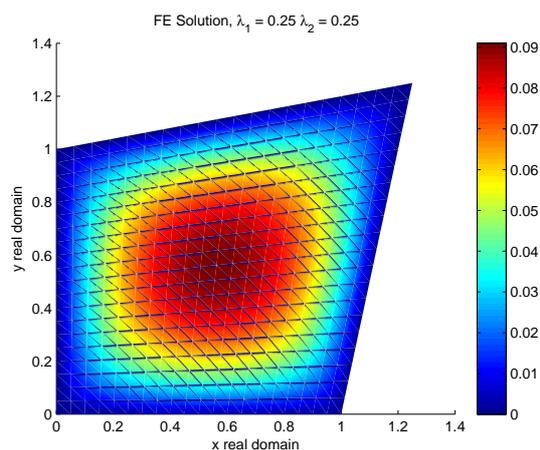


figure 5.24 – Solution éléments finis

Par conséquent, aucune résolution du problème directe ne doit-êtré effectuée “On – line”. Quelques applications ont été présentées, avec des exemples numériques assez prometteurs.

Cependant, les problèmes de complétion de données sont mal posés (Bui, 1993). Afin de régulariser les problèmes présentés, nous avons utilisé la régularisation évanescence.

De plus, nous avons présenté deux alternatives de calcul de solutions à représentation séparée dans des domaines complexes. La première technique consiste à étendre le domaine complexe à un autre domaine, lui hexaédrique. Ceci permet de calculer une solution à représentation séparée. Ensuite, les bonnes conditions sont retrouvées par la résolution d’un problème de Cauchy avec une régularisation évanescence. L’inconvénient principal de cette méthode est le



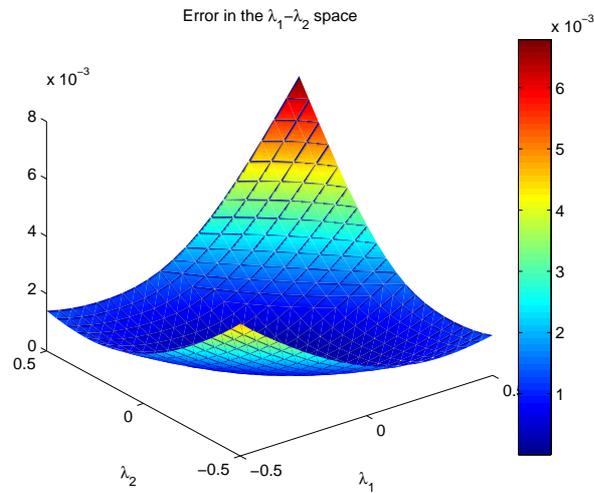


figure 5.25 – Erreur en norme L_2 pour λ_1 et $\lambda_2 \in [-0.5, 0.5]$

traitement des non-linéarités, ou des termes sources.

La deuxième méthode proposée consiste à introduire les paramètres géométriques comme des coordonnées supplémentaires du problème. Ceci permet d'obtenir, par un seul calcul, la solution pour n'importe quelle géométrie, obtenue à partir d'un choix des paramètres. L'inconvénient principal de cette approche reste la difficulté de la séparation de l'inverse des jacobiens. Dans la section 5.3.2 nous avons utilisé une approximation de l'inverse du jacobien qui permet d'avoir d'excellents résultats pour des petites "transformations" de la géométrie. Cependant, pour d'autres transformations, d'autres stratégies doivent être utilisées, comme des développements d'ordre supérieur.

6

Optimisation et contrôle pour des modèles non-linéaires

6.1	Modèle de Bailleul non-linéaire paramétrique	160
6.1.1	Mise en équation de la thermique de cuisson	160
6.1.2	Résolution paramétrique du modèle	161
6.1.3	Découplage	164
6.1.4	Post traitement de la solution multidimensionnelle	167
6.2	Linéarisation et contrôle d'un modèle non-linéaire	170
6.2.1	Problème modèle	171
6.2.2	Découplage	172
6.2.3	Implémentation numérique	174
6.2.4	Problèmes rencontrés	174
6.2.5	Collage des solutions à chaque Δt	178
6.3	Conclusion et perspectives	182

Les méthodes de réduction de modèles semblent très efficaces jusqu'à présent. Toutefois, face à la non-linéarité, le défi n'est toujours pas relevé complètement. Une étape de linéarisation doit être toujours réalisée "à priori", pour ensuite entamer le calcul. Dans la "reduced basis", on calcule des vecteurs sur des "Smart points" du domaine multidimensionnel. Ces solutions constitueront plus tard la base de calcul "On-line". Des algorithmes dit "Greedy" sont utilisés et réalisent des approximations très acceptables (Temlyakov et Zheltov, 2011). Dans la PGD, les non-linéarités sont traitées jusqu'à présent par une étape de linéarisation, de façon itérative, ou en réalisant des développements par la méthode asymptotique numérique (Pruliere *et al.*, 2010b). Cependant, l'amélioration en temps de calcul, qui est le but final de l'utilisation des méthodes de réduction de modèles, reste faible. De même, peu de modèles paramétriques non-linéaires ont été traités par la PGD avant cette thèse. Dans ce chapitre, nous proposons une nouvelle méthode de traitement de la non-linéarité dans l'équation de la chaleur en utilisant la PGD, tout en paramétrant la solution afin de réaliser de l'optimisation ou du contrôle.

6.1 Modèle de Bailleul non-linéaire paramétrique

Le modèle de Bailleul exprime la cinétique de réticulation d'une résine comme étant une fonction exponentielle de la température (Bailleul *et al.*, 1994). Ceci donne un modèle couplé non-linéaire. Ce modèle est utilisé pour simuler les champs de température dans les matrices composites durant la période de cuisson (Bailleul *et al.*, 1996a). La bonne simulation de la cuisson des composites est nécessaire afin de modéliser le bon comportement des pièces finies tenant compte des contraintes résiduelles. Quelques travaux couplant la thermique et la chimie de réticulation ont été déjà réalisés (Msallem, 2008). En effet, Rabearison *et al.* (2009) avaient pris en compte ce couplage dans la simulation, ce qui a abouti à un bon accord avec l'expérience. D'autres travaux de simulation ont été réalisés en utilisant ce modèle couplé (Li *et al.*, 2001; Zhu *et al.*, 2001; Cheung *et al.*, 2004). Ces travaux, par contre, n'envisagent aucun paramétrage de la solution, vu le temps de calcul et la mémoire nécessaire pour stocker la solution. Avec la PGD, nous calculerons la solution paramétrique de ce modèle non linéaire.

6.1.1 Mise en équation de la thermique de cuisson

L'équation de la chaleur en présence d'une réticulation de la résine s'écrit comme étant :

$$\rho \mathbf{C}_p(\mathcal{U}, \alpha) \frac{\partial \mathcal{U}}{\partial t} = \nabla [\mathbf{K}(\mathcal{U}, \alpha) \nabla \mathcal{U}] + \rho w H \frac{\partial \alpha}{\partial t}, \quad (6.1)$$

où t est le temps, \mathcal{U} la température, \mathbf{K} le tenseur de conductivité, \mathbf{C}_p la capacité calorifique, ρ la masse volumique du matériau, H l'enthalpie de la réaction, w la fraction en masse de résine et α le degré d'avancement de la réaction. L'équation est non-linéaire puisque \mathbf{K} et \mathbf{C}_p dépendent de \mathcal{U} et en plus de α .

D'autre part la vitesse de réticulation ou l'avancement de la réaction chimique est donnée par l'équation suivante :

$$\frac{\partial \alpha}{\partial t} = \mathcal{K}(\mathcal{U}) \sum_{i=0}^{i=7} a_i \alpha^i \quad (6.2)$$

Les coefficients a_i dépendent de la résine utilisée alors que \mathcal{K} est donné par une loi d'Arrhenius :

$$\mathcal{K}(\mathcal{U}) = \mathcal{K}_{ref} \exp \left(-b \left(\frac{U_{ref}}{\mathcal{U}} - 1 \right) \right) \quad (6.3)$$

Les coefficients b , U_{ref} et \mathcal{K}_{ref} sont des constantes qui dépendent du matériau utilisé. Ce



problème (non paramétrique) a déjà été résolu en utilisant la PGD (Pruliere *et al.*, 2010b). La technique de résolution utilise un algorithme itératif de point fixe, qui met à jour \mathbf{K} et \mathbf{C}_p à chaque itération pour résoudre ensuite un problème linéaire. \mathcal{K} a été représenté de façon séparée par un développement en série de Taylor.

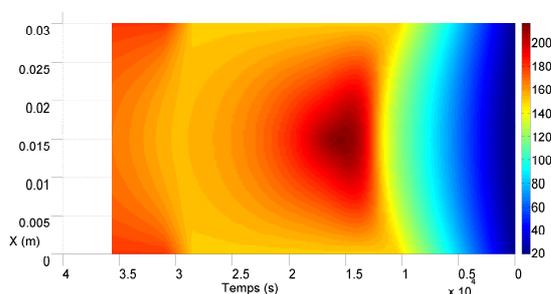


figure 6.1 – Température U de la résine pour le cycle optimal

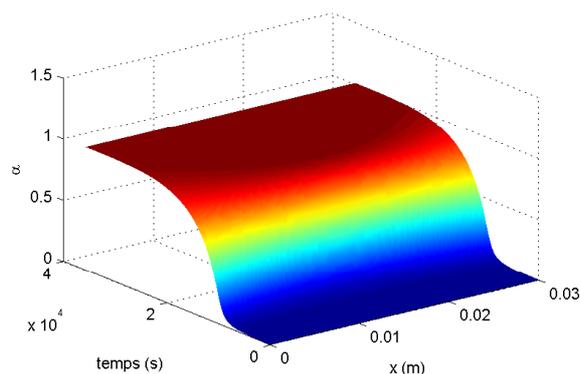


figure 6.2 – Réticulation α de la résine pour le cycle optimal

Le cycle optimal de cuisson donné par Hexel (2010) pour l'époxy MP21 consiste à monter la température de 0.5 à 1°C par minute jusqu'au premier palier $\tau_1 = 135^\circ C$. Ensuite après 180 minutes sur ce palier on monte la température de 0.5 à 1°C par minute jusqu'à $\tau_2 = 180^\circ C$ où on reste 120 minutes. La simulation correspondante est donnée par les figures 6.1 et 6.2. Ce cycle est montré à titre d'exemple. Les paramètres de ce cycle sont donnés par l'expérience et le savoir faire des ingénieurs.

6.1.2 Résolution paramétrique du modèle

La figure 6.3 illustre la cuisson du composite étudié. Par symétrie et loin des effets de bord, la modélisation de ce système se réduit à une étude unidimensionnelle dans l'espace. Par la suite, nous résolvons un problème dans un domaine physique 1D noté $\Omega = [0, L]$ unidimensionnel, transitoire et paramétrique. Le cycle de cuisson choisi est schématisé par la figure 6.4. Il est évident que l'optimisation de ce cycle revient à bien déterminer $\theta_1, \theta_2, \theta_3, \theta_4, \tau_1$ et τ_2 , paramètres qui seront introduites comme des extra-coordonnées du problème. L'idée est donc de résoudre le problème 8D (l'espace x , le temps t et les six paramètres) qui représentera tous les cycles de cuisson d'une résine donnée. Pour notre simulation nous utilisons les paramètres donnés dans le tableau 6.1.

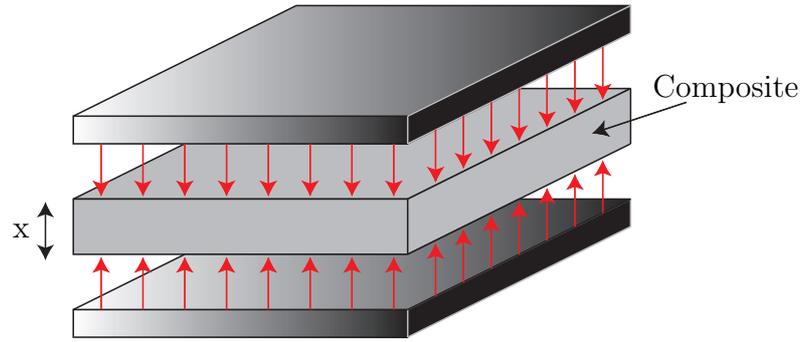


figure 6.3 – Illustration de la cuisson à optimiser

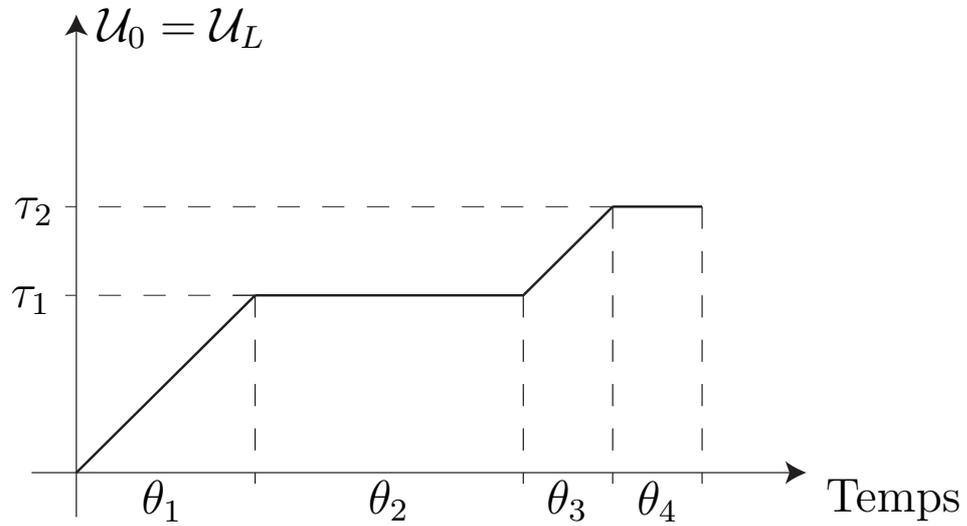


figure 6.4 – Schémas d'un cycle de cuisson pour un stratifié

$a_0 = 0.10560429$	$w = 0.7$
$a_1 = 2.4391767$	$H = 44.51 \text{ J/Kg}$
$a_2 = 44.4947643$	$a_K = 0.256 \text{ W.m}^{-1}.\text{°C}^{-1}$
$a_3 = -293.26562$	$b_K = -0.0012 \text{ W.m}^{-1}.\text{°C}^{-2}$
$a_4 = 724.82261$	$a_c = 1200 \text{ J.Kg}^{-1}.\text{°C}^{-1}$
$a_5 = -898.512237$	$b_c = 6.809 \text{ J.Kg}^{-1}.\text{°C}^{-2}$
$a_6 = 559.603379$	$\mathcal{K}_{ref} = 0.0001614 \text{ s}^{-1}$
$a_7 = -139.683785$	$b = 12.1198$
$\rho = 1560 \text{ Kg/m}^3$	$U_{ref} = 402.63 \text{ K}$

tableau 6.1 – Les paramètres du modèle de Bailleul utilisés dans la simulation

Pour réaliser le calcul avec la PGD, on procède d'abord par une étape d'homogénéisation pour pouvoir définir les coefficients \mathbf{K} et \mathbf{C}_p . D'abord on peut démontrer sous certaines hypothèses qu'il est possible d'établir une loi de mélange pour déterminer la capacité thermique en

fonction de la température (Guyonvarch *et al.*, 1994). On considère dans ce qui suit une évolution linéaire du coefficient de conductivité \mathbf{K} et de la capacité calorifique \mathbf{C}_p en fonction de la température \mathcal{U} (Bailleul *et al.*, 1996b; Msallem, 2008) :

$$\begin{cases} \mathbf{K} = b_K \mathcal{U} + a_K \\ \mathbf{C}_p = b_c \mathcal{U} + a_c \end{cases} \quad (6.4)$$

Les coefficients a_K , b_K , a_c et b_c varient en fonction de la réticulation de la résine. Par exemple, on observe une variation de 30% lors de la cuisson de l'époxy (Bailleul *et al.*, 1996a). Pour résoudre ce problème avec la PGD, on va considérer que la température varie peu dans l'épaisseur de la plaque. Par la suite, on pourra calculer la conductivité \mathbf{K} et la capacité calorifique \mathbf{C}_p en fonction des conditions limites \mathcal{U}_0 et \mathcal{U}_L montrées dans la figure 6.4, et ceci *a priori*. Par contre, dans ce problème on agit en paramétrant les conditions aux limites. Ces dernières sont illustrées par la figure 6.4. Ainsi le problème revient à chercher \mathcal{U} sous la forme :

$$\mathcal{U}(x, t, \theta_1, \theta_2, \theta_3, \theta_4, \tau_1, \tau_2) = \sum_{i=1}^{i=N} X_i(x) T_i(t) \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) \Theta_i^3(\theta_3) \Theta_i^4(\theta_4) \Pi_i^1(\tau_1) \Pi_i^2(\tau_2), \quad (6.5)$$

tel que \mathcal{U} vérifie les équations (6.1), (6.2), (6.3) et (6.4). Les paramètres sont de deux natures différentes et subissent par la suite deux traitements différents.

1. Paramétrisation des intervalles de temps :

Pour considérer les intervalles de temps comme paramètres supplémentaires, on agit par un changement de variables géométriques (Leygue et Verron, 2010). Notons l'intervalle de définition de t , $\Omega_t = [0; \theta_1 + \theta_2 + \theta_3 + \theta_4]$. On définit aussi $s \in \Omega_s = [0; 4]$ tel que :

$$\begin{cases} \Omega_s = \mathcal{T}^{-1}(\Omega_t) \\ \mathcal{T} = \begin{cases} s \times \theta_1 & \text{si } s \in [0, 1] \\ (s - i + 1) \times \theta_i + \sum_{j=1}^{j=i-1} \theta_j & \text{sinon} \end{cases} \end{cases} \quad (6.6)$$

Ceci aboutit donc à définir Ω_s^i tel que $\Omega_s^i = \mathcal{T}^{-1}(\Omega_t^i) = [(i - 1); i]$ et on obtient dans Ω_s^i le jacobien de la transformation géométrique \mathcal{T} de Ω_s à Ω_t tel que :

$$\frac{\partial t}{\partial s} = \theta_i \quad \forall i \quad (6.7)$$

■

2. Paramétrisation des paliers :

Suivant l'approche PGD, on cherche dans le cas présent la solution du problème sous la forme d'une somme finie de produits de fonctions, d'une seule coordonnée chacune. \mathcal{U} s'écrira sous la forme illustrée dans l'équation (6.5). Pour rentrer les conditions limites on écrit :

$$\mathcal{U} \approx \mathcal{F} + \sum_{i=n+1}^{i=N} X_i(x) T_i(t) \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) \Theta_i^3(\theta_3) \Theta_i^4(\theta_4) \Pi_i^1(\tau_1) \Pi_i^2(\tau_2) \quad (6.8)$$

où \mathcal{F} vérifie les conditions aux limites. Ceci se traduit par l'écriture de la fonction $\mathcal{F}(x, t, \theta_1, \theta_2, \theta_3, \theta_4, \tau_1, \tau_2)$ sous une forme séparée. La seule contrainte pour définir \mathcal{F} est de satisfaire les conditions aux limites (cf. figure 6.4).

La combinaison des deux paramétrisations, et le remplacement de \mathcal{U} par son approximation dans les équations (6.1), (6.2), (6.3) et (6.4) aboutit à un nouveau problème à résoudre. Le couplage des équations introduit les paramètres dans des équations non-linéaires, notamment l'équation (6.2). Une approche PGD classique est incapable de gérer de façon efficace la résolution de ce problème car le nombre de termes générés par les puissances de α explose.

6.1.3 Découplage

Comme on vient d'avancer, une fois tous les paramètres remplacés dans l'équation (6.1), le problème devient très complexe à résoudre. En effet la cinétique de la réaction ainsi que le dégagement de chaleur sont fortement non-linéaires. A noter que puisque la conductivité et la capacité calorifique sont maintenant déterminées par les conditions limites, le traitement thermique est linéaire. Par la suite, on agit donc en résolvant d'abord le problème homogène, puis en calculant une solution particulière du problème complet, afin de séparer la difficulté due à la multidimensionnalité de celles dues à la non linéarité. Ceci revient donc à résoudre les deux problèmes suivants :

$$\left\{ \begin{array}{l} \rho \mathbf{C}_p(\mathcal{F}) \frac{\partial \mathcal{U}_h}{\partial t} - \nabla [\mathbf{K}(\mathcal{F}) \nabla \mathcal{U}_h] = 0 \\ \text{Vérifiant les conditions initiales et limites} \end{array} \right. \quad (6.9)$$

$$\left\{ \begin{array}{l} \rho \mathbf{C}_p(\mathcal{F}) \frac{\partial \mathcal{U}_p}{\partial t} = \nabla [\mathbf{K}(\mathcal{F}) \nabla \mathcal{U}_p] + \rho w H \frac{\partial \alpha}{\partial t} \\ \text{Avec conditions initiales et limites homogènes} \end{array} \right. \quad (6.10)$$

■

La solution du problème est donc donnée par la somme des deux solutions : $\mathcal{U} = \mathcal{U}_h + \mathcal{U}_p$.

6.1.3.1 Le problème homogène

En remplaçant le second membre de l'équation (6.1) par zéro, on obtient la formulation homogène du problème à résoudre dans Ω_s :

$$\int \mathcal{U}^* \left(\rho \mathbf{C}_p \frac{\partial \mathcal{U}_h}{\partial s} \cdot \frac{\partial s}{\partial t} - \frac{\partial}{\partial x} \left(\mathbf{K} \frac{\partial \mathcal{U}_h}{\partial x} \right) \right) = 0 \quad (6.11)$$

Par la suite on cherche la solution dans Ω_s comme une somme de produits de fonctions, ou un tenseur de Kruskal (Kolda et Bader, 2009). Après un relèvement des conditions limites, on remarque qu'il faut simplement résoudre le problème défini dans Ω_s tel que :

$$\int \mathcal{V}^* \left(\rho \mathbf{C}_p \frac{\partial \mathcal{V}}{\partial s} \cdot \frac{\partial s}{\partial t} - \frac{\partial}{\partial x} \left(\mathbf{K} \frac{\partial \mathcal{V}}{\partial x} \right) \right) = \int \mathcal{V}^* \mathcal{R}(\mathcal{F}) \text{ avec } \mathcal{U} = \mathcal{V} + \mathcal{F} \quad (6.12)$$

Où \mathcal{R} est le résultat de l'introduction de \mathcal{F} dans l'équation homogène définie dans (6.11). En résolvant l'équation (6.12) pour des conditions aux limites de type Dirichlet homogènes, on obtient la solution du problème paramétrique sans second membre. Le problème homogène est résolu en 4 minutes sur un ordinateur portable à deux cœurs de 2.13 Ghz chacun. Une particularisation de la solution est montrée dans la figure 6.5.

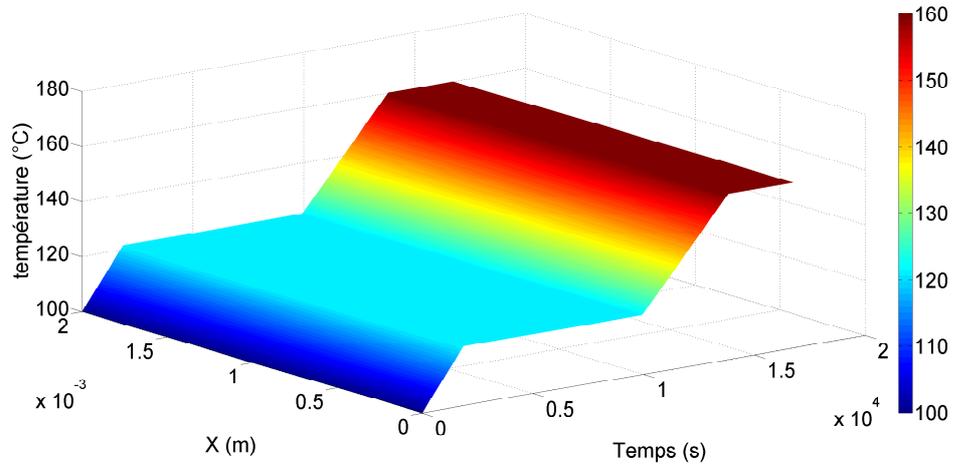


figure 6.5 – Une particularisation de la solution homogène paramétrique

6.1.3.2 Solution particulière

Afin de calculer la solution particulière de l'équation complète "on-line", nous avons envisagé d'abord d'utiliser la POD. Ainsi on précalcule "off-line" quelques solutions par des méthodes classiques pour extraire les vecteurs de la base réduite par une décomposition aux valeurs singulières (Horn et Johnson, 1985). Ainsi, on obtient une équation pour les coefficients de la base réduite :

$$\mathbf{\Gamma}^T [(\rho\mathbf{M} - \Delta t\mathbf{A}) \mathbf{\Gamma}\Lambda_{i+1}] = \mathbf{\Gamma}^T [\rho\mathbf{M}U_i + \rho\omega H(\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i)], \quad (6.13)$$

où $\mathbf{\Gamma}$ est la matrice des vecteurs d'espace et Λ le vecteur avec les coefficients dépendants du temps. Ainsi, on obtient un système de m équations à m inconnues, avec m le nombre de vecteurs dans la base réduite. Ce système remplace donc le système original de taille égale au nombre de degrés de libertés en x . Pour le système étudié m est égal à 3. Pour le maillage utilisé (101 nœuds en espace x et 401 nœuds en temps t), l'erreur relative de cette méthode par rapport aux différences finies est de 0.565%, par contre le temps de calcul est de 2.14 secondes contre 2.74 secondes pour les différences finies, pour le même maillage. Le gain en temps n'est dû qu'à l'utilisation de la POD, puisque à chaque pas de temps on devra reconstruire la solution homogène et résoudre l'équation (6.2). Le gain obtenu est donc insignifiant et ne justifie pas l'utilisation de la PGD, au moins quand l'espace physique est unidimensionnel.

6.1.3.3 Calcul on-line de la solution particulière

Par la suite nous avons changé de méthode afin de calculer off-line la solution particulière aussi. Ainsi, on réalise des "snapshots" pour une large bande de valeurs, afin d'interpoler localement plus tard. Ainsi 10 000 snapshots ont été effectuées pour différentes combinaisons des paramètres $\theta_1, \theta_2, \theta_3, \theta_4, \tau_1$ et τ_2 qui servent à calculer les paramètres de l'équation complète, d'où on obtient la solution particulière. Nous stockons ces 10 000 snapshots sous une forme séparée pour économiser de l'espace mémoire (Bader et Kolda, 2007). Dans l'étape "on-line", on interpole linéairement et localement entre les snapshots. Une évaluation de la solution homogène prend 26.5 ms, alors que l'interpolation de la solution particulière 712 ms, contre 3.59 secondes pour la solution par différences finies pour le même maillage. La figure 6.6 montre la solution retrouvée si on tombe sur le snapshot précalculé tandis que la figure 6.7 illustre la solution différences finies pour les mêmes valeurs des paramètres. La figure 6.8 représente la solution retrouvée pour des paramètres qui ne coïncident avec aucun des snapshots, et 6.9 montre la même solution différences finies. Les figures 6.10 et 6.11 montrent



les erreurs relatives pour les deux cas respectivement. L'erreur montrée par la figure 6.10 est due seulement à l'hypothèse de considérer la conductivité et la capacité calorifique fonction de \mathcal{F} , par suite fonction du temps seulement et non de l'épaisseur.

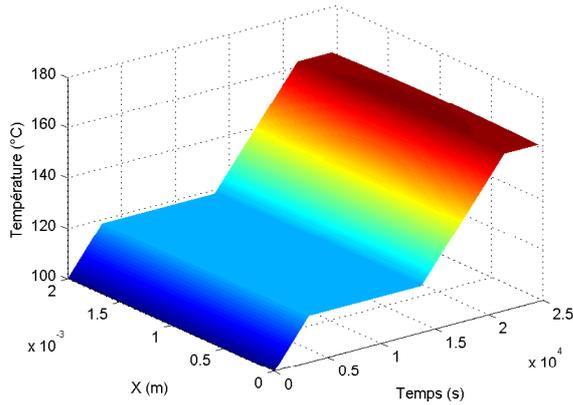


figure 6.6 – La solution $\mathcal{U}_h + \mathcal{U}_p$ sur un snapshot

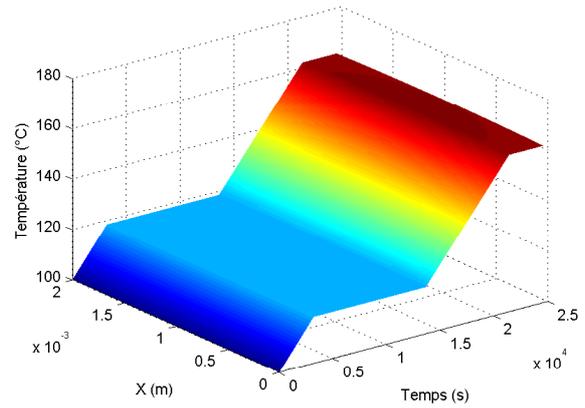


figure 6.7 – La solution différences finies sur un snapshot

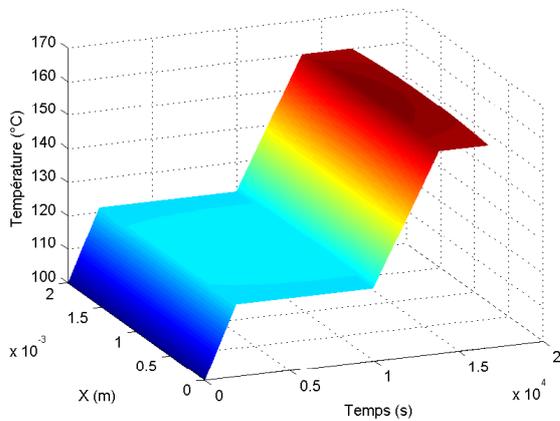


figure 6.8 – La solution $\mathcal{U}_h + \mathcal{U}_p$ non coïncidant avec un snapshot

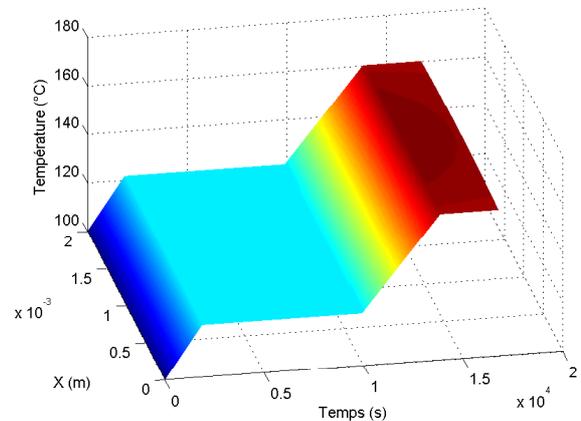


figure 6.9 – La solution différences finies loin du snapshot

6.1.4 Post traitement de la solution multidimensionnelle

Une fois la solution multidimensionnelle trouvée, l'optimisation du procédé de cuisson est entamée. Dans cette section, nous cherchons à minimiser le temps total du cycle défini par :

$$j_1 = \theta_1 + \theta_2 + \theta_3 + \theta_4 \quad (6.14)$$



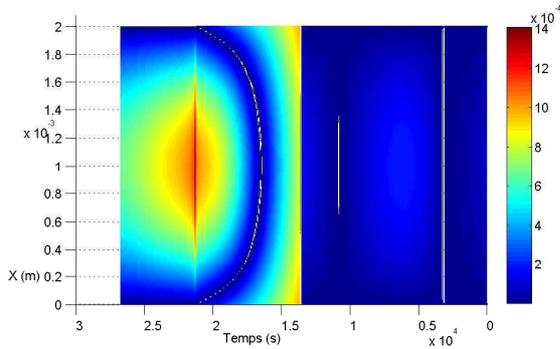


figure 6.10 – *Erreur relative entre notre méthode et les différences finies sur un snapshot*

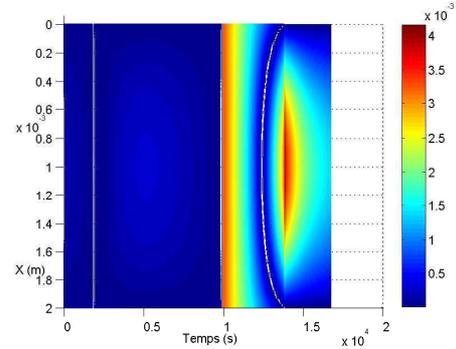


figure 6.11 – *Erreur relative entre notre méthode et les différences finies loin du snapshot*

En même temps, nous minimisons la différence de température entre le bord et le milieu de la pièce. Ceci se traduit par la minimisation d'une seconde fonction coût J_2 définie par :

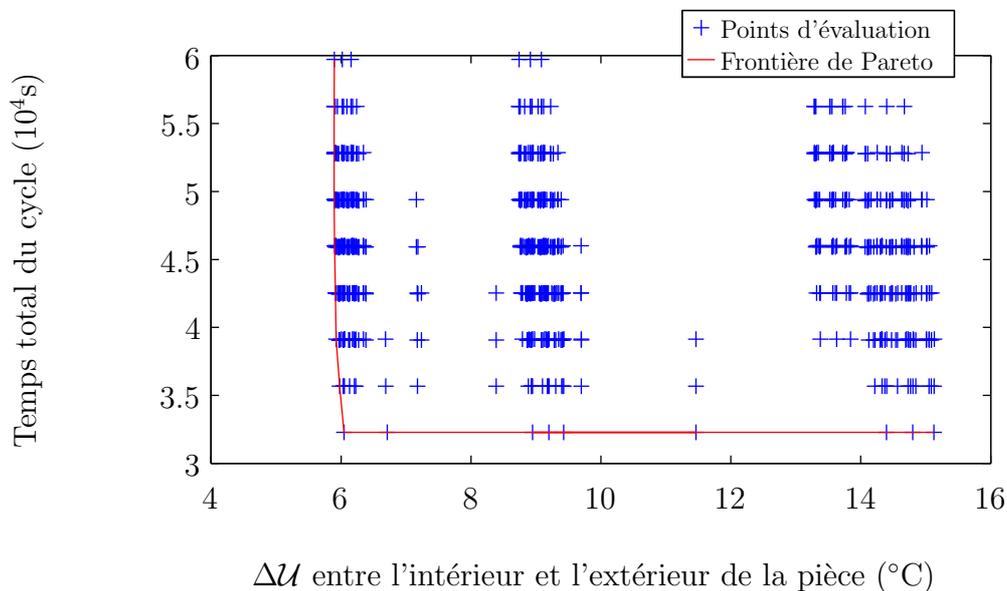
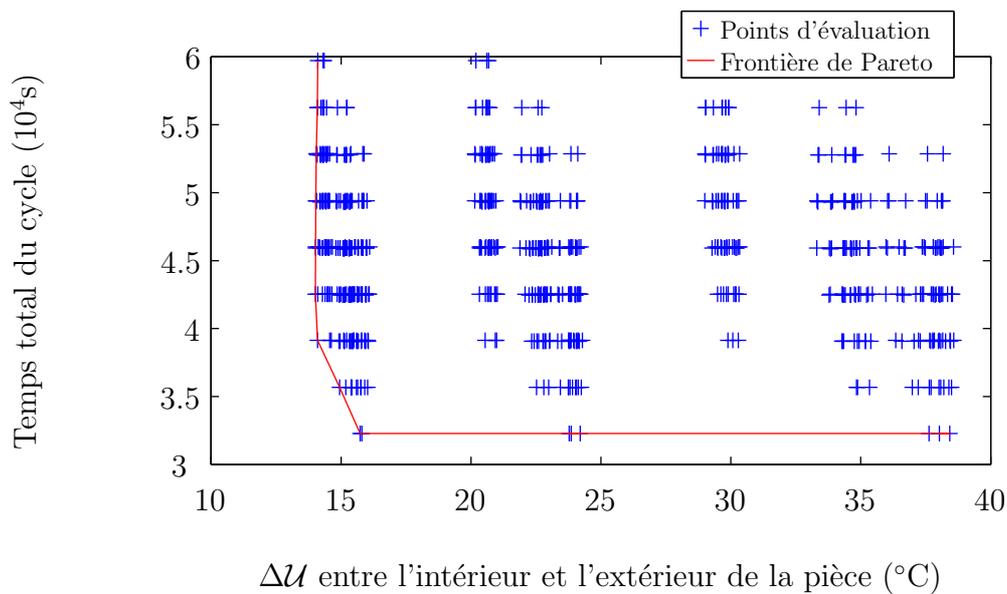
$$j_2 = \Delta\mathcal{U} = \sqrt{(\mathcal{U}(0,t) - \mathcal{U}(L/2,t))^2} \quad (6.15)$$

Ce problème consiste donc en une minimisation à multiobjectifs. Par conséquent, la solution n'est pas unique. Elle dépend de la pondération de chacune des deux fonctions coût. Cependant, en utilisant la PGD, nous pouvons tracer la frontière de Pareto, ensemble de tous les points non dominés. Par la suite, une fois face à cette courbe, nous pourrions choisir plus facilement le point optimal. La figure 6.12 montre la frontière de Pareto pour une plaque de 10mm d'épaisseur, tandis que la figure 6.13 montre la frontière de Pareto pour une courbe de 20mm d'épaisseur.

De plus, si nous cherchons à optimiser un four contenant deux plaques d'épaisseurs différentes, le problème devient multiobjectif ayant 3 fonctions coût :

$$\begin{cases} j_1 = \theta_1 + \theta_2 + \theta_3 + \theta_4 \\ j_2 = \Delta\mathcal{U}_1 = \sqrt{(\mathcal{U}_1(0,t) - \mathcal{U}_1(L_1/2,t))^2} \\ j_3 = \Delta\mathcal{U}_2 = \sqrt{(\mathcal{U}_2(0,t) - \mathcal{U}_2(L_2/2,t))^2} \end{cases} \quad (6.16)$$

La figure 6.14 illustre une surface de Pareto de tous les points non dominés, avec $L_1 = 10mm$ et $L_2 = 20mm$. La création d'une telle surface en utilisant notre méthode ne nécessite que quelques particularisations. En revanche, retrouver une telle surface par les méthodes classiques nécessite beaucoup de résolutions directes de l'équation aux dérivées partielles.

figure 6.12 – *Frontière de Pareto pour une plaque de 10mm d'épaisseur*figure 6.13 – *Frontière de Pareto pour une plaque de 20mm d'épaisseur*

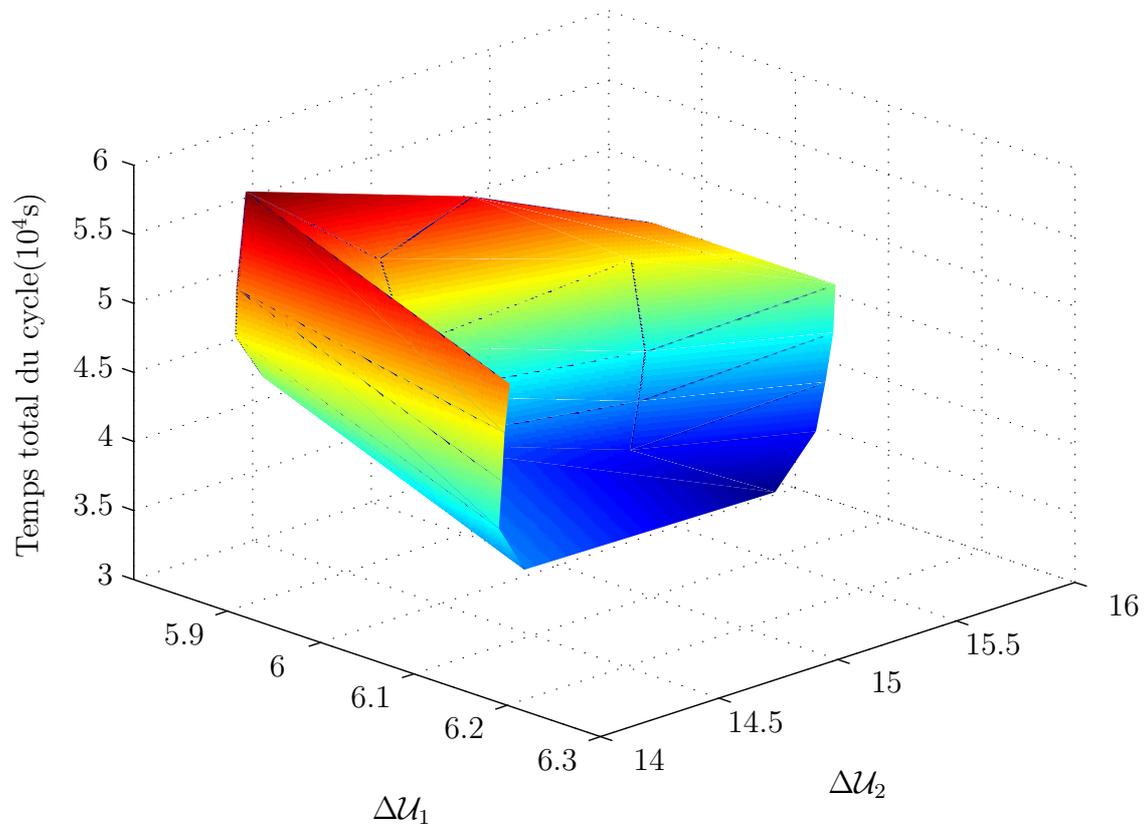


figure 6.14 – Surface de Pareto pour deux plaques ayant 10mm et 20mm d'épaisseur

6.2 Linéarisation et contrôle d'un modèle non-linéaire

Dans cette section on utilisera le principe de la DDDAS pour contrôler un procédé à partir de la génération de chaleur thermique interne. On entend par le principe de la DDDAS l'introduction de données supplémentaires à la simulation tous les Δt secondes. A la suite de l'introduction de ces nouvelles données, on utilisera un calcul inverse pour déterminer la chaleur générée et évaluer le procédé. Evidemment le critère de validation de ce modèle serait de calculer plus rapidement une itération du problème inverse par PGD que par les méthodes classiques. Donc, la vitesse de particularisation de la solution paramétrique calculée par PGD devrait être supérieure à la vitesse de résolution éléments finis ou différences finies. De plus, l'idée est de réaliser le contrôle d'un modèle non linéaire, à l'aide d'une solution paramétrée précalculée. Dans ce qui suit, un problème modèle est présenté, ainsi que la méthode de résolution et ses résultats.

6.2.1 Problème modèle

Considérons un domaine 1D $\Omega = [0, l]$ m, avec $x \in \Omega$. L'équation du modèle à résoudre s'écrit donc :

$$\frac{\partial U}{\partial t} = (K(U)\Delta U) + \mathcal{S}(a,b,c) \quad (6.17)$$

U étant le champ de température, t le temps avec $t \in \Gamma$, $K(U)$ la conductivité qui dépend de U . On note aussi le terme de génération de chaleur \mathcal{S} qu'on va paramétrer en fonction de a , b et c . Ces trois paramètres sont la valeur de la fonction pour trois intervalles de temps (voir figure 6.15), ou pour 3 intervalles d'espace (voir figure 6.16). On suppose avoir l'accès à la température à tous les Δt . On définit donc un maillage grossier du temps, comme le montre la figure 6.17. On cherchera donc à résoudre off-line le problème pour toute condition initiale et condition limite de Dirichlet :

$$\begin{cases} U(0,t) = \theta_1 \\ U(l,t) = \theta_2 \\ U(x,0) = G(x) \end{cases} \quad (6.18)$$

Avec $G(x)$ un polynôme de Newton paramétré par les valeurs des trois quartiles du domaine, θ_3 , θ_4 et θ_5 respectivement, comme le montre la figure 6.18.

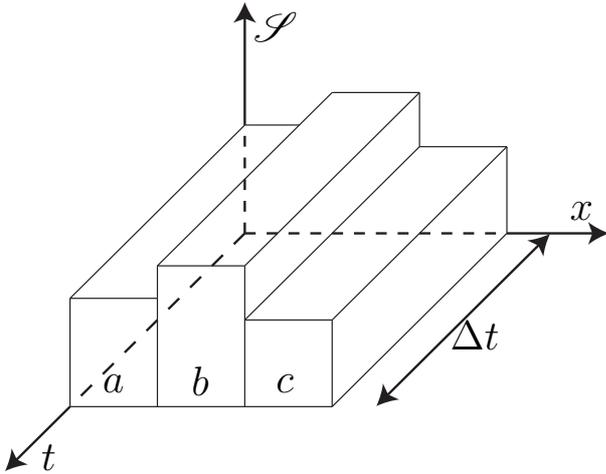


figure 6.15 – Décomposition de la source de chaleur dans un intervalle Δt

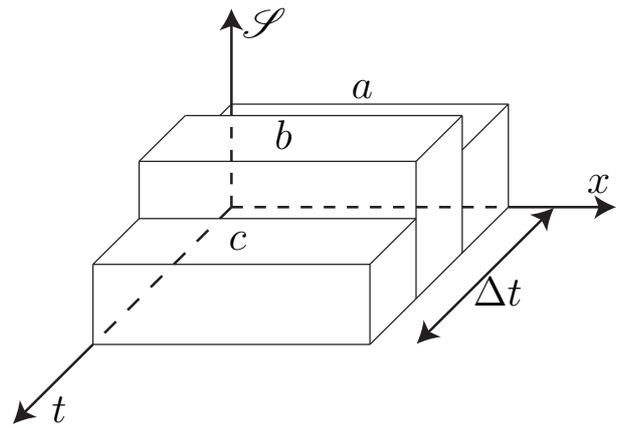
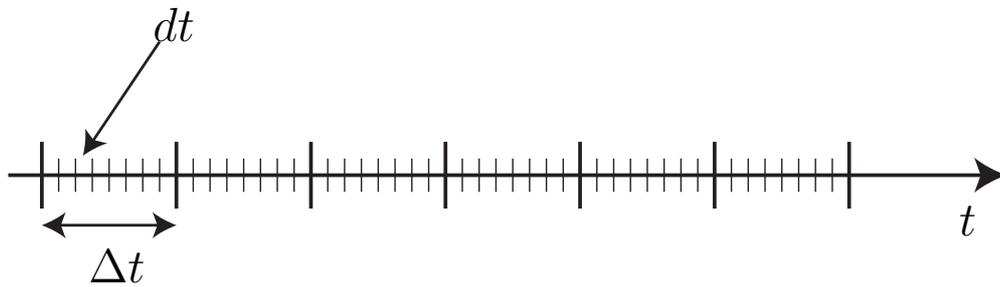
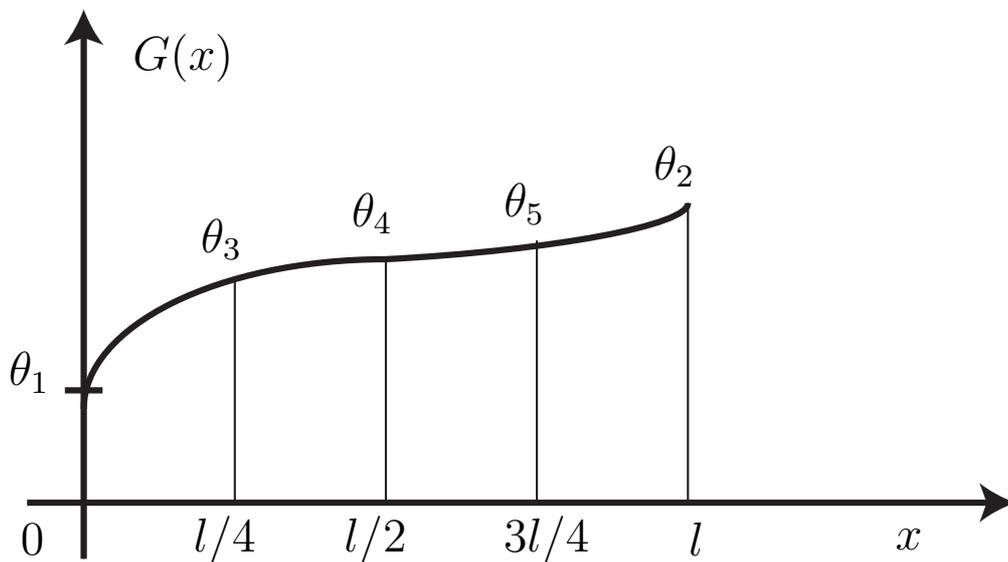


figure 6.16 – Paramétrisation de la fonction \mathcal{S} en x , constante sur un pas de temps Δt

Par la suite, si on résout l'équation (6.17) dans un intervalle Δt , on pourra identifier la valeur de la source de chaleur dans cet intervalle en mesurant la température en trois points, ce qui



figure 6.17 – Maillage du temps : Δt est le pas grossierfigure 6.18 – La condition initiale, paramétré par les valeurs de θ_i

donne trois équations à trois inconnues pour faire l'identification. On pourra aussi calculer la source pour atteindre ces trois températures “consigne” afin de réaliser le contrôle.

6.2.2 Découplage

Les difficultés du problème défini dans la section 6.2.1 sont principalement la multidimensionnalité et la non linéarité de l'équation (6.17). D'abord on cherche à découpler ces deux difficultés.

Hypothèse 1 : Sachant que la conductivité K peut être approximée comme une fonction li-



néaire de la température (Msallem, 2008), on posera K comme étant :

$$K(U) = \chi + \varphi U = \chi + \varphi U(x, t = 0) = D(x) \quad (6.19)$$

et écrivant : $D(x) = m + h(x)$ où m est la moyenne de $D(x)$, on obtient :

$$\frac{\partial (U^\dagger + U^*)}{\partial t} = [m + h(x)] \frac{\partial^2 (U^\dagger + U^*)}{\partial x^2} + \mathcal{S}(a, b, c), \quad (6.20)$$

avec $U = U^\dagger + U^*$ où U^\dagger provient de la solution d'une équation ne contenant pas $h(x)$ et U^* la correction de cette dernière, ce qui peut se réécrire tenant compte de la linéarité :

$$\frac{\partial U^\dagger}{\partial t} = m \frac{\partial^2 U^\dagger}{\partial x^2} + \mathcal{S}(a, b, c) \quad (6.21)$$

$$\frac{\partial U^*}{\partial t} = m \frac{\partial^2 U^*}{\partial x^2} + h(x) \frac{\partial^2 U^\dagger}{\partial x^2} + h(x) \frac{\partial^2 U^*}{\partial x^2} \quad (6.22)$$

Séparant la solution de l'équation (6.21) en une homogène et une autre particulière $U^\dagger = U^h + U^{p1}$, l'équation (6.22) donne :

$$\frac{\partial U^*}{\partial t} = m \frac{\partial^2 U^*}{\partial x^2} + h(x) \frac{\partial^2 (U^h + U^{p1}(a, b, c))}{\partial x^2} + h(x) \frac{\partial^2 U^*}{\partial x^2} \quad (6.23)$$

En conséquence, le problème final à résoudre pourra être décomposé sous forme de trois problèmes élémentaires plus simples :

1. Un problème homogène linéaire, avec une condition initiale paramétrée :

$$\begin{cases} \frac{\partial U^h}{\partial t} = m \frac{\partial^2 U^h}{\partial x^2} \\ U^h(x, 0) = G(x) \end{cases} \quad (6.24)$$

A noter que m doit-être aussi ajouté comme paramètre supplémentaire du problème puisque m est la moyenne de $D(x)$, qui varie dans chaque sous-intervalle de temps, en fonction de U_0 .

2. Une solution particulière, avec une source de chaleur paramétrée :

$$\begin{cases} \frac{\partial U^{p1}}{\partial t} = m \frac{\partial^2 U^{p1}}{\partial x^2} + \mathcal{S}(a, b, c) \\ U^{p1}(x, 0) = 0 \end{cases} \quad (6.25)$$

■

De même que pour le problème homogène, m est un paramètre du problème.

3. Une deuxième solution particulière qui contient les termes où $h(x)$ apparaît, assez simple pour être résolu *on – line* :

$$\begin{cases} \frac{\partial U^{p2}}{\partial t} = m \frac{\partial^2 U^{p2}}{\partial x^2} + h(x) \frac{\partial^2 (U^h + U^{p1}(a,b,c))}{\partial x^2} + h(x) \frac{\partial^2 U^{p2}}{\partial x^2} \\ U^{p2}(x,0) = 0 \end{cases} \quad (6.26)$$

Les équations (6.24) et (6.25) sont résolues facilement avec la PGD (Ghnatios *et al.*, 2011b; Chinesta *et al.*, 2010a). La résolution de l'équation (6.26) se fait *on – line*, après avoir particularisé tous les paramètres du modèle.

6.2.3 Implémentation numérique

Pour la résolution du problème modèle (6.17), on utilise un domaine $\Omega = [0,4]$ et $\Gamma = [0,1]$. Pour le problème homogène de l'équation (6.24), on utilise la PGD pour obtenir une solution sous la forme :

$$U^h(x,t,\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,m) \approx \sum_{i=1}^{i=N} X_i(x)T_i(t)\Theta_{1i}(\theta_1)\Theta_{2i}(\theta_2)\Theta_{3i}(\theta_3)\Theta_{4i}(\theta_4)\Theta_{5i}(\theta_5)M_i(m) \quad (6.27)$$

Pour l'équation (6.25) on cherche la solution par PGD sous la forme de :

$$U^{p1}(x,t,a,b,c,m) \approx \sum_{i=1}^{i=N} X_i(x)T_i(t)A_i(a)B_i(b)C_i(c)M_i(m) \quad (6.28)$$

Enfin, pour l'équation (6.26), on utilise les différences finies avec un algorithme implicite :

$$\frac{U_{i+1}^{p2} - U_i^{p2}}{dt} = m \mathcal{D}_2 U_{i+1}^{p2} + \mathcal{D}_{h2} (U^h + U^{p1}) + \mathcal{D}_{h2} U^{p2} \quad (6.29)$$

Où \mathcal{D}_2 et \mathcal{D}_{h2} proviennent de la discrétisation des dérivées secondes. Les détails de l'algorithme seront donnés plus loin dans la figure 6.26.

6.2.4 Problèmes rencontrés

6.2.4.1 Solution homogène difficile à converger

La solution homogène était d'abord calculée avec 30 modes. Cette solution est définie dans un espace de grande dimension et nécessite une symétrisation vu la présence de l'opérateur



temporel non symétrique. La PGD converge en utilisant la technique de minimisation du résidu. La solution obtenue avec 30 modes en fixant $\theta_i = 80^\circ\text{C}$ et $m = 0.5$ est donnée par la figure 6.19. Ceci engendre une erreur de l'ordre de 8% suivant la norme L_2 alors qu'elle passe à 0.2% avec 300 modes (Plus tard ce résultat a été comprimé en utilisant une séparation aux valeurs singulières SVD). Les solutions PGD avec 30 modes et différences finies sont montrées dans la figure 6.20, alors que l'erreur par rapport aux différences finies est donnée dans la figure 6.21. La solution PGD après le calcul de 300 modes est donnée dans la figure 6.22.

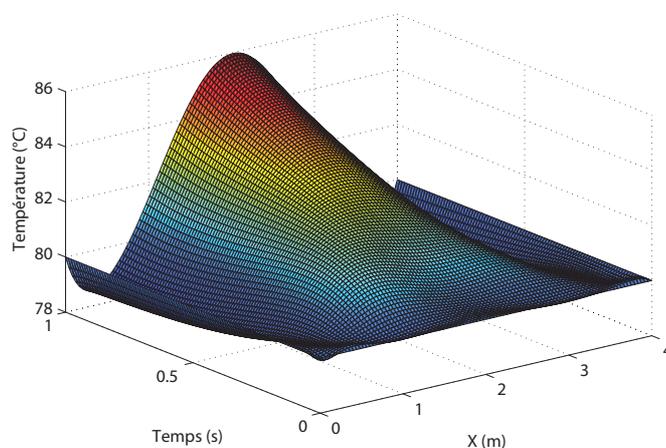


figure 6.19 – Solution homogène non convergée après 30 modes

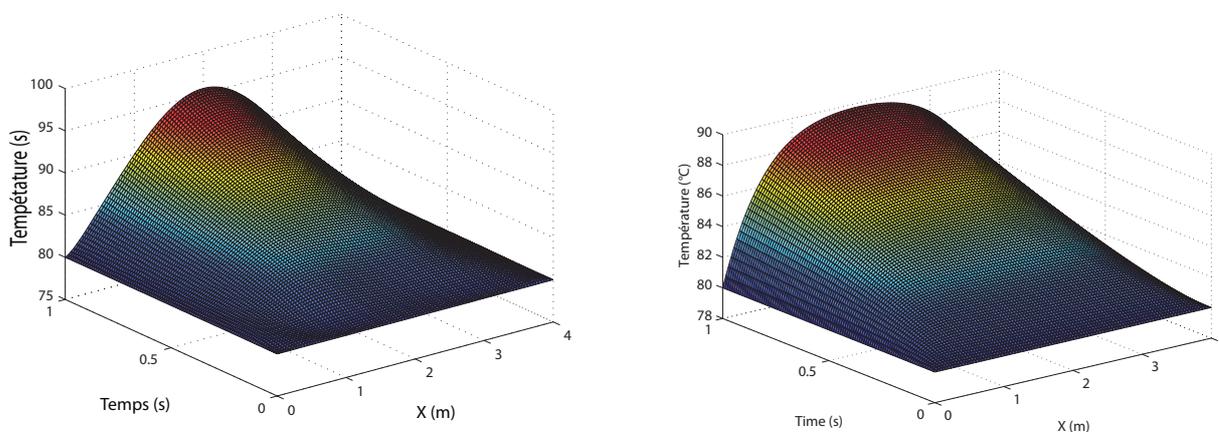


figure 6.20 – Solution non convergée à gauche et la solution de référence à droite

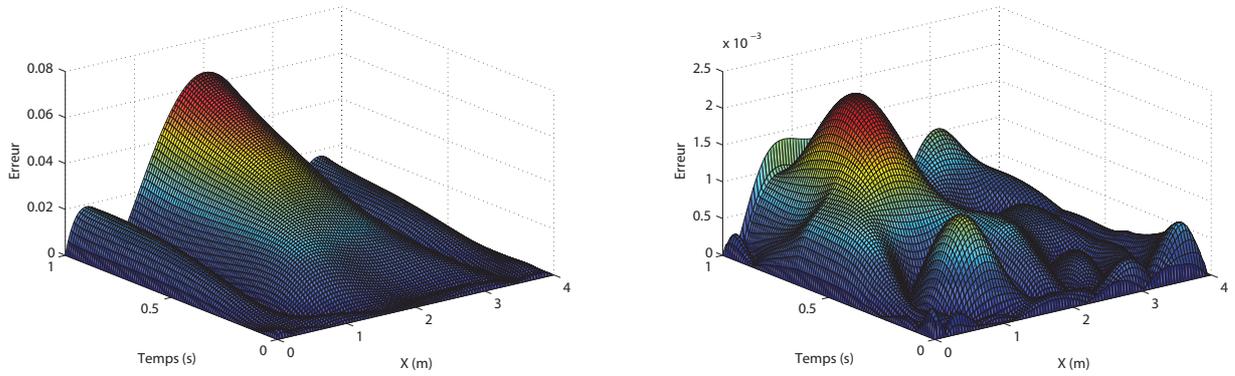


figure 6.21 – L'erreur en norme L_2 en utilisant $N = 30$ à gauche et $N = 300$ à droite

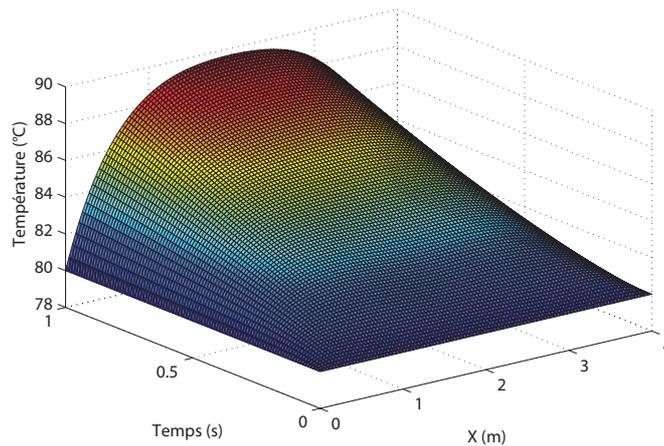


figure 6.22 – Solution linéaire convergée avec $U_0 = 80^\circ C$

6.2.4.2 Paramétrisation de la condition initiale

L'approximation polynomiale de la condition initiale nécessite l'écriture de cette dernière sous la forme :

$$U_0(x) = \sum_{i=1}^{i=5} \theta_i \phi_i(x) \quad (6.30)$$

Où les ϕ_i sont les fonctions d'approximation. Les polynômes de Newton sont définis sur tout l'ensemble des x . Etant donné un ensemble de $nb + 1$ points $(x_0, y_0), (x_1, y_1), \dots, (x_{nb}, y_{nb})$ un



interpolant est défini par :

$$\mathcal{N} = \sum_{j=0}^{j=nb} d_j \mathcal{F}_j(x) \quad (6.31)$$

Avec $\mathcal{F}_j = \prod_{i=0}^{j-1} (x - x_i)$ et $d_j = [y_0, \dots, y_j]$ une différence divisée définie par :

$$\begin{cases} [y_0] = y_0 \\ [y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} \\ \vdots \\ [y_0, \dots, y_j] = \frac{[y_j, \dots, y_1] - [y_{j-1}, \dots, y_0]}{x_j - x_0} \end{cases} \quad (6.32)$$

Ainsi en développement on peut obtenir les ϕ_j comme étant :

$$\phi_j(x) = \frac{1}{\prod_{k=1}^{j-1} \left(\sum_{k=1}^{j-1} \Delta i \right)} \left(\mathcal{F}_j + \sum_{e=j+1}^{e=5} \frac{\mathcal{F}_e (-1)^{e+j}}{\prod_{n=1}^{j+1} \left(\sum_{m=j}^{n+j+1} \Delta m \right)} \right) \quad (6.33)$$

Ceci avec $\Delta i = x_i - x_{i-1}$. Dans notre cas, c'est la distance entre deux points du maillage grossier, où on mesure la température.

6.2.4.3 Le calcul on-line U^{p2}

Si d'abord on néglige U^{p1} dans le terme $h(x) \left(\frac{\partial^2 (U^h + U^{p1})}{\partial x^2} \right)$ de l'équation (6.26), on obtient une erreur de 2.5% en norme L_2 . Par contre, si on ne néglige pas ce dernier, on obtient une erreur égale à 0.39%. La figure 6.23 montre l'erreur en négligeant U^{p1} , en choisissant pour la condition initiale: $\theta_1 = \theta_2 = 100^\circ\text{C}$, $\theta_3 = \theta_5 = 80^\circ\text{C}$ et $\theta_4 = 20^\circ\text{C}$. La solution après avoir considéré U^{p1} est donnée dans la figure 6.24. L'erreur par rapport aux différences finies avec la norme L_2 est donnée dans la figure 6.25.

Pour tous les calculs réalisés, le temps de calcul *on - line* varie entre 20 et 30 ms. A noter que puisque U^{p1} est calculé *on - line*, les coefficients a , b et c doivent être prédéfinis a priori. Ainsi on devra réaliser une étape d'estimation de ces derniers *on - line*, nommons "étape prédicteur", avant de réaliser l'identification finale, nommons "étape correcteur" (clef l'algorithme montré dans la figure 6.26).



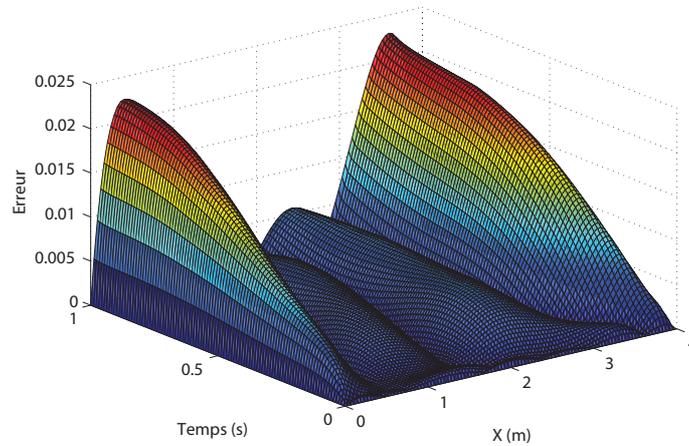
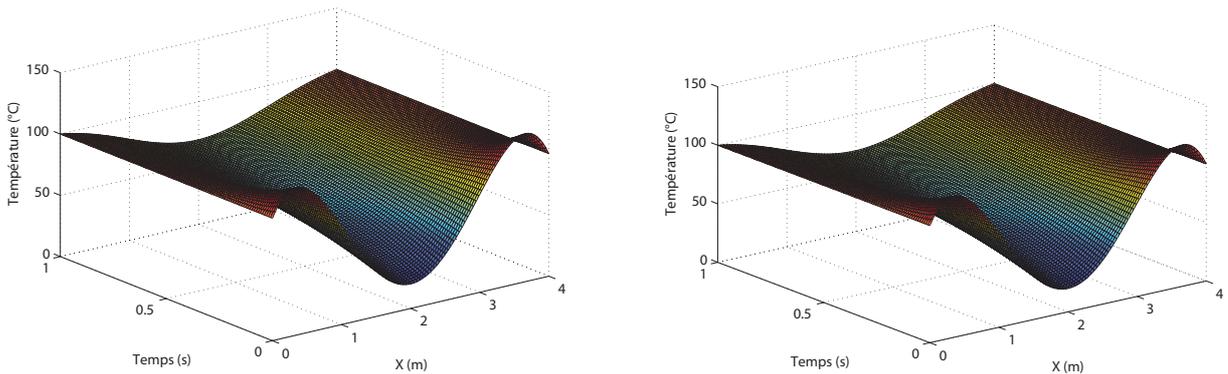
figure 6.23 – L'erreur en norme L_2 en négligeant U^{p1} 

figure 6.24 – Les solutions en PGD à gauche et en différences finies à droite

6.2.5 Collage des solutions à chaque Δt

Pour le recollage des sous-domaines de temps Δt , la lecture de la condition initiale pour trouver θ_3 , θ_4 et θ_5 est directe, à partir des capteurs. Par contre, pour le choix de la fonction de génération de chaleur, pour atteindre la consigne \tilde{U}^1, \tilde{U}^2 et \tilde{U}^3 , on a besoin d'utiliser un algorithme de minimisation d'une fonction coût fortement non linéaire :

$$J(a,b,c) = \frac{1}{2} \left(U(x_1, \Delta t, a, b, c) - \tilde{U}^1 \right)^2 + \frac{1}{2} \left(U(x_2, \Delta t, a, b, c) - \tilde{U}^2 \right)^2 + \frac{1}{2} \left(U(x_3, \Delta t, a, b, c) - \tilde{U}^3 \right)^2, \quad (6.34)$$

x_1 , x_2 et x_3 étant les abscisses des trois points de mesure des températures \tilde{U}^1 , \tilde{U}^2 et \tilde{U}^3 . Pour

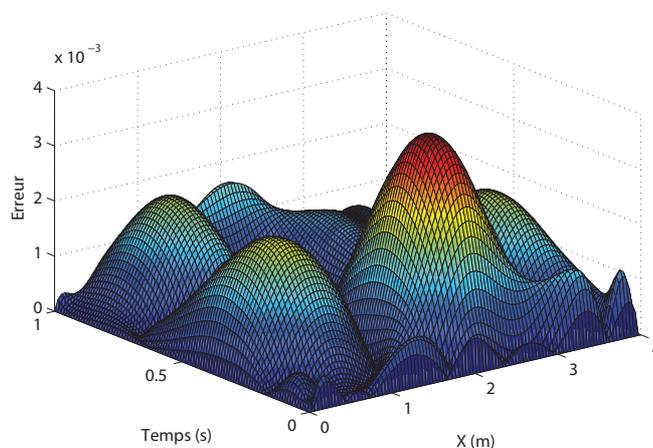


figure 6.25 – L'erreur en norme L_2 en tenant compte de U^{p1}

pouvoir évaluer la fonction U^{p1} , nécessaire pour calculer U^{p2} , on a besoin des valeurs de a , b et c . Donc, on procède d'abord par une première minimisation pour calculer (a^*, b^*, c^*) , et ceci en négligeant U^{p2} . Ensuite, on utilise (a^*, b^*, c^*) pour évaluer U^{p1} afin de calculer U^{p2} lors d'une deuxième minimisation, ce qui donne les valeurs optimales recherchées $(a^{opt}, b^{opt}, c^{opt})$. Ainsi l'algorithme utilisé, illustré dans la figure 6.26, devient :

- Avec les équations (6.24) et (6.25) on minimise (6.34) une première fois pour calculer une prédiction de (a, b, c) , que l'on note (a^*, b^*, c^*) .
- Evaluer $U^{p1}(a^*, b^*, c^*)$.
- Résoudre l'équation (6.26) en utilisant $U^{p1}(a^*, b^*, c^*)$ pour obtenir U^{p2} .
- Avec les équations (6.24), (6.25) et (6.26) minimiser à nouveau l'équation (6.34) pour obtenir les bonnes valeurs de a , b et c .

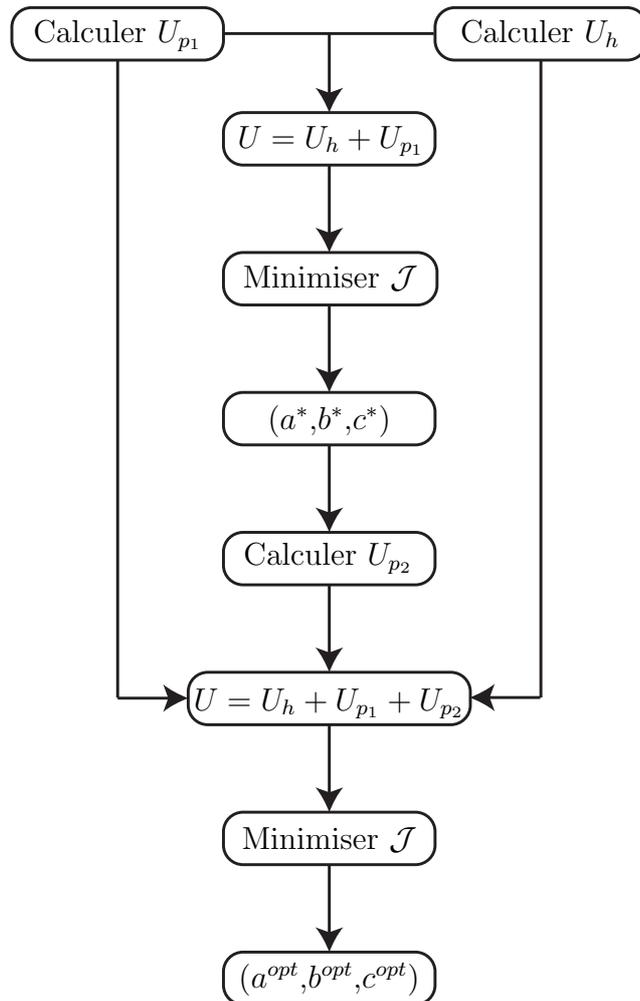
La minimisation est accomplie par la technique de Levenberg-Marquardt, sans stabilisation.

On pose donc :

$$\begin{aligned} f^k &= [a, b, c]^k \\ r_j &= \left(U(x_j, \Delta t, f_1^k, f_2^k, f_3^k) - \tilde{U}^j \right) \end{aligned} \quad (6.35)$$

Où x_j sont les abscisses des points de mesures de la température \tilde{U}^j . f^k étant le triplet (a, b, c)



figure 6.26 – *Algorithme de recollage des sous domaines de temps Δt*

à l'itération k . De plus on calcul la matrice \mathcal{J} comme étant :

$$\mathcal{J} = \begin{bmatrix} \frac{\partial r_j}{\partial f_i} \end{bmatrix} \begin{matrix} i = 1,2,3 \\ j = 1,2,3 \end{matrix} \quad (6.36)$$

Pour calculer les termes de la matrice \mathcal{J} nous évaluons la fonction r_i deux fois pour chaque terme de \mathcal{J} afin de faire des différences finies, par exemple :

$$\mathcal{J}_{23} = \frac{\left(U(x_3, \Delta t, f_1^k, f_2^k, f_3^k) - \tilde{U}^3 \right) - \left(U(x_3, \Delta t, f_1^k, f_2^k + \epsilon, f_3^k) - \tilde{U}^3 \right)}{\epsilon} \quad (6.37)$$

où ϵ étant un pas suffisamment petit. Enfin une itération correspond à :

$$f^{k+1} = f^k - [\mathcal{J}^T \mathcal{J} + \lambda \text{Diag}(\mathcal{J})] \times \mathcal{J}^T r \quad (6.38)$$

λ étant un paramètre qui accélère la convergence. L'erreur par rapport à la solution différences finies est de 2% après 2 itérations (voir figure 6.31). La convergence est acceptable (moins de 0.4% d'erreur relative) au bout de 3 à 4 itérations. Dans la première version de l'algorithme, le temps de recollage à chaque Δt était de l'ordre de 7 secondes. Ce temps énorme est dû à la multitude de fonctions non optimisées appelées durant la minimisation. Cet algorithme a été amélioré pour réduire le temps de calcul. Ci dessous les détails du calcul et les améliorations apportées.

Algorithme de base pour $\Delta t = 1s$, $dt = 0.01s$ et $N = 300$: La reconstruction de la solution homogène dure 330 ms alors que la particulière U^{p1} 125 ms. D'autre part, le calcul de la solution particulière U^{p2} coûte 27 à 30 ms. On remarque donc que le coût d'appel d'une fonction est de presque 0.5 s, alors que la résolution différences finies pour la même taille du maillage dure 126 ms. Comme on a besoin d'appeler 5 fois la fonction par itération à cause de la minimisation de \mathcal{J} , soit à 2.5 secondes par itération, et 3 itérations pour coller un sous-domaine représente 7 secondes environ. Par la suite, il a fallu optimiser l'algorithme en réduisant le nombre de modes dans les solutions PGD. Par conséquent, nous avons utilisé le "back span" ou retour en arrière pour "compresser" les produits de fonctions déjà calculés (voir la section 2.1.3), ce qui réduit considérablement le nombre de modes. En effet, pour la solution homogène on passe de 300 à 18 modes, et pour la solution particulière on passe de 150 à 16 modes.

Algorithme amélioré $\Delta t = 1s$, $dt = 0.01s$ et $N = 83$: Le temps requis pour une évaluation



se situe entre 24 et 28 ms quelque soit le maillage en x (testé pour des valeurs allant de 10 à 1000 points). Le temps de résolution en différences finies explose en fonction du nombre de points utilisés (voir figure 6.27). Le gain réalisé au niveau de la PGD est dû principalement au compactage de la solution U^h et U^{p1} réalisé et à l'amélioration de quelques boucles d'évaluation des fonctions pré-calculées.

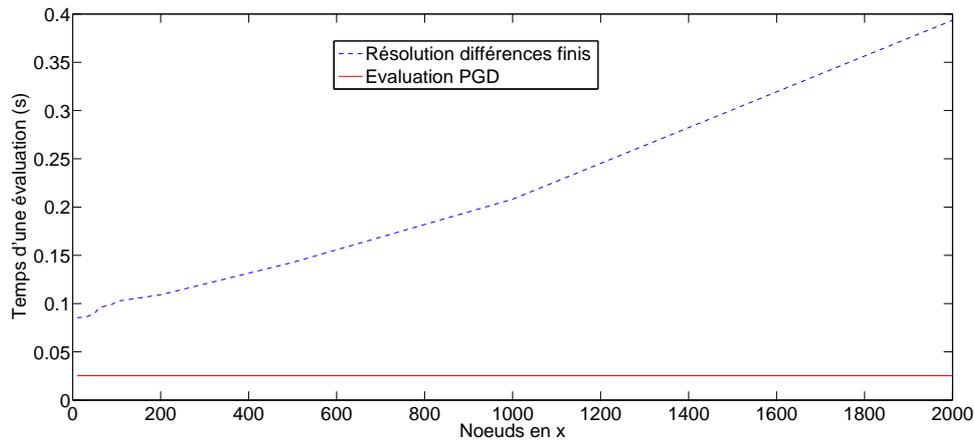


figure 6.27 – Comparaison des temps de calcul entre l'évaluation PGD et différences finies

Nous avons testé le recollage de trois sous domaines Δt . La figure 6.28 superpose le temps de calcul de la solution donnée par les différences finies (en bleu pointillé) et celui nécessaire à l'évaluation de la solution PGD (en rouge continue) pour un domaine recollé, la source \mathcal{S} étant paramétrée en espace. L'erreur relative calculée point par point en norme L_2 est montré dans la figure 6.29. La figure 6.30 montre la solution pour une source \mathcal{S} paramétrée en temps, et l'erreur relative en norme L_2 est donnée dans la figure 6.31.

A noter que les valeurs $(a^{opt}, b^{opt}, c^{opt})$ obtenues pour la fonction source \mathcal{S} sont légèrement différentes des valeurs imposées lors de la conception de la solution de référence en différences finies. L'algorithme converge vers les valeurs exactes (a^{ex}, b^{ex}, c^{ex}) en augmentant le nombre d'itérations.

6.3 Conclusion et perspectives

Dans la section 6.1 de ce chapitre nous avons appliqué la PGD à un problème thermique non linéaire couplé paramétré pour optimiser un cycle de cuisson. Le calcul d'une solution



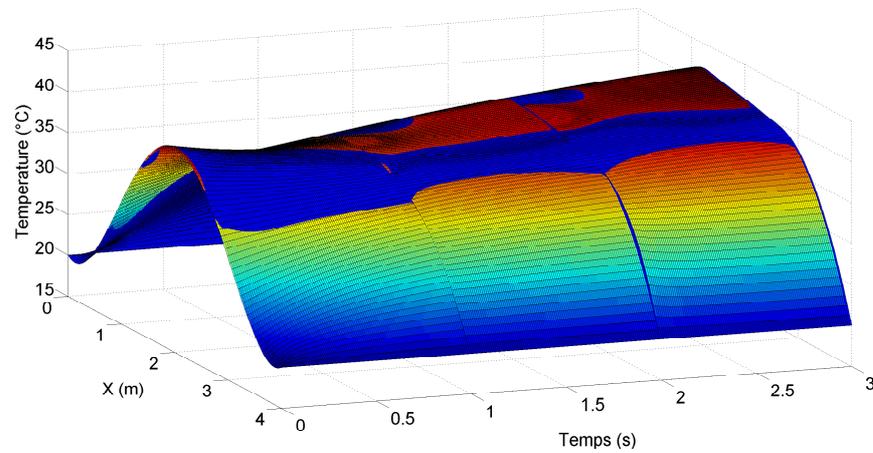


figure 6.28 – *Solution finale après recollage pour un terme source paramétré en espace x*

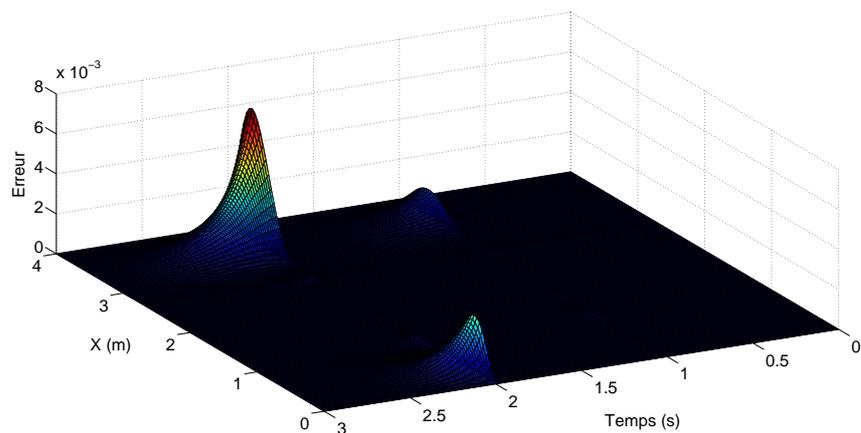


figure 6.29 – *Erreur en norme L_2 de la solution recollée par rapport à la vraie solution*

particulière “on-line” s’avère très handicapant, par la suite une méthode s’inspirant des autres approches de réduction de modèles a été mis en œuvre pour précalculer les solutions particulières. Dix mille snapshots ont été réalisés et une interpolation locale est effectuée. Ainsi une particularisation et une interpolation sont réalisées “on-line” puis additionnées pour obtenir la solution.

Autrement, le modèle de Bailleul a été utilisé pour la simulation de la cuisson de la résine. Il existe d’autres modèles très utilisés comme celui de [S. Sourour \(1976\)](#) ([Ryan et Dutta, 1979](#)). Même si ce modèle est relativement ancien, il reste bien d’actualité ([Nakouzi et al., 2011](#)).



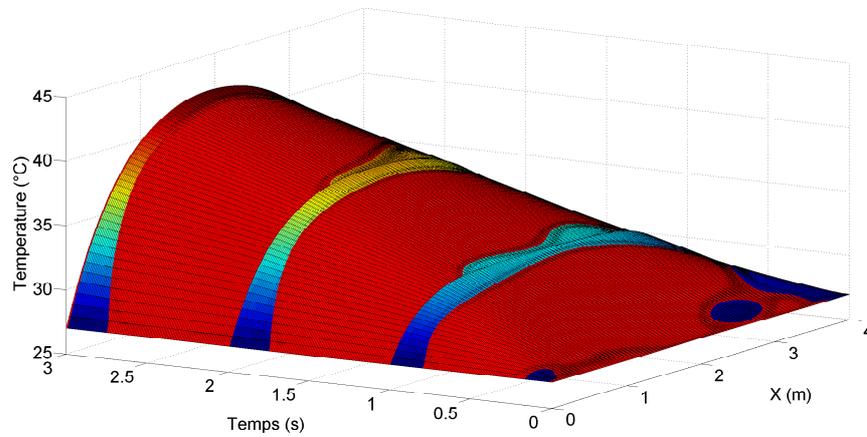


figure 6.30 – *Solution différences finies en rouge et solution après recollage en couleur*

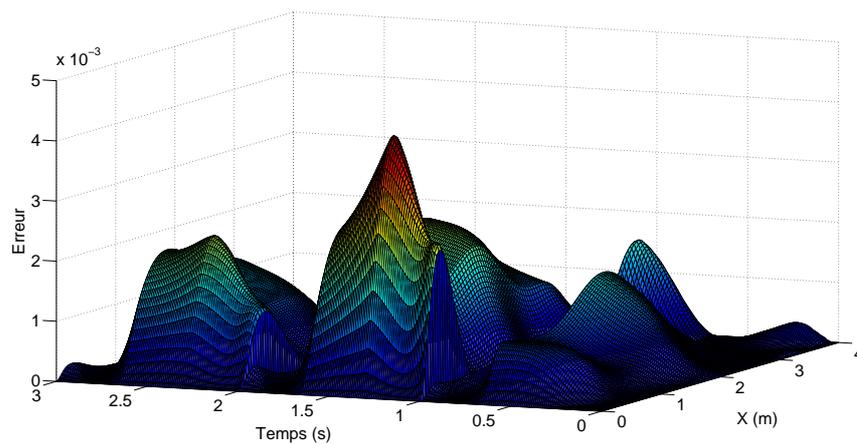


figure 6.31 – *L'erreur en norme L_2 après recollage de trois sous-domaines*

On pourra envisager une simulation en utilisant ce modèle et puis la comparer aux résultats du modèle de Bailleul.

D'autre part, le travail présenté dans la section 6.2 porte sur une application DDDAS non-linéaire “on-line” en utilisant la PGD. Des essais numériques valident l'approche proposée. A noter que maintenant les erreurs présentes dans la solution sont principalement dues à la paramétrisation de la solution initiale (voir figure 6.29). En effet les polynômes de Newton utilisés ne semblent pas être le choix idéal pour représenter la solution. De plus, si on rajoute encore des points comme paramètres dans la condition initiale, tout en conservant la même

interpolation polynomiale, l'erreur s'amplifie. Ceci est dû à la tendance de ce polynôme à osciller entre tous les "nœuds" considérés.

Afin d'améliorer l'algorithme, on pourra utiliser une meilleure approximation de la condition initiale, et ceci en considérant les fonctions de base de l'opérateur lui même. Par contre, cela induit une difficulté majeure, qui est celle de pouvoir paramétrer la condition initiale en fonction de la température sur des points à coordonnées données. L'utilisation des bases réduites POD est envisagée dans des travaux en cours. Cela aboutira à une bonne approximation de la condition initiale et à une réduction de l'erreur relative.





Conclusion générale et perspectives

Conclusion générale

Les matériaux composites présentent des défis de simulation mécanique. En effet, la structure complexe de ces matériaux exige une simulation 3D. D'autre part, l'apparition des méthodes de réduction de modèles offre une solution à ce problème. Ces méthodes se basent sur une représentation séparée de l'espace afin de réduire le coût du calcul nécessaire à la résolution.

Dans cette thèse, on s'est focalisé sur la PGD afin de résoudre des problèmes multidimensionnels. Différentes formulations de la PGD ont été présentées. Dans un premier temps, une formulation tensorielle générale a été montrée. Cette dernière est utilisée pour formuler la solution de problèmes thermiques paramétriques. Cette formulation générale est adaptée à la résolution de la plupart des problèmes rencontrés. D'autre part, le critère de convergence utilisé est la réduction du résidu. De plus, d'autres formulations adaptant la PGD à des cas spécifiques ont été réalisées, comme dans le cadre du contrôle non-destructif par exemple, un domaine qui acquiert de plus en plus d'attention des industriels. Ensuite, une première démarche de résolution d'équations différentielles dans des domaines contenant des discontinuités de propriétés telles que les délaminations a été présentée.

Dans un deuxième temps, une méthodologie de calcul paramétrique utilisant la PGD a été présentée. Des calculs de conditions limites paramétrées ont été réalisés dans le cadre de la pultrusion. Ces calculs présentent l'intérêt d'être très généraux. Par la suite, des particularisations donnent la solution pour les conditions limites envisagées. Les particularisations étant rapides, le calcul en temps réel devient possible. Par la suite, un premier travail visant l'appli-

cation de la DDDAS dans le contrôle de procédés industriels modélisés par des équations aux dérivées partielles a été présenté. De plus, la procédure proposée combine des calculs “off-line” avec d’autres “on-line”. La partie “off-line” génère des solutions paramétrées sous une forme tensorielle séparée, utilisant la PGD. Grâce à cette dernière, la solution est facile à stocker dans des plateformes de calcul légères, tel que les “smartphones”. De même, toutes les sensibilités par rapport aux paramètres peuvent être calculées explicitement “off-line”. D’autre part, dans la partie “on-line”, des particularisations de la solution sont réalisées. De plus, l’optimisation, mais aussi le contrôle, sont réalisées “on-line”. Les paramètres du procédé sont entrés par l’utilisateur dans le smartphone qui donne, en sortie, la reconfiguration nécessaire afin de conserver un fonctionnement optimal du procédé. Pour l’optimisation et le contrôle, l’algorithme de Levenberg-Marquardt a été utilisé. Par la suite, un brevet d’invention et une marque enregistrée, la “iPGD®”, ont été déposés.

Ensuite, le défi de la simulation 3D des matériaux composites a été relevé. En commençant par la simulation des stratifiés composites à direction des fibres paramétrée, un intérêt de la simulation 3D est bien présenté. De plus, une simulation des stratifiés composites paramétriques est développée dans la thèse. D’autre part, la simulation de la dépose de bande paramétrée est présentée en détail. La simulation de ce procédé présente un problème d’oscillation numérique classique du schéma d’intégration centré, rencontré dans les problèmes de convection-diffusion, qui nécessite une stabilisation. Cette dernière est réalisée par la méthode dite “SUPG”. D’autre part, le résultat de cette simulation est implémenté sur une tablette. Cette dernière présente aussi la possibilité de post-traitement des résultats, en calculant les coefficients de cicatrisation et dégradation de la matrice d’après les lois développées dans [Nicodeau \(2005\)](#).

Plus tard, l’utilisation de la PGD dans les calculs inverses est présentée. Une solution paramétrique est d’abord calculée par la PGD. Ensuite, dans l’étape d’identification, on ne réalise que des particularisations de la solution. Des problèmes de Cauchy 1D et 2D ont été résolus. Par la suite, une application sur les problèmes de Cauchy a été montrée. Cette application s’inscrit dans le cadre de l’extension de domaines. En effet, des domaines à forme complexe sont immergés dans d’autres hexaédriques ou rectangulaires. Ces derniers sont traités facilement par la séparation de variables. Une autre méthode de traitement de domaines à forme complexe est présentée. Cette dernière consiste à réaliser un “mapping” paramétré du domaine complexe vers un autre carré. Finalement, une comparaison montrant les avantages et les inconvénients de chacune de ces méthodes a été montrée.



Ensuite, un problème non linéaire a été traité. Ce dernier est un problème thermique de cuisson d'une plaque composite. Le modèle de Bailleul a été utilisé pour modéliser la cinétique de réticulation. Lors de la résolution, une solution PGD est calculée "off-line", et une base réduite POD a été construite pour calculer une autre solution "on-line". Une application d'optimisation d'un four de cuisson en découle. L'optimisation étant multiobjectifs, la solution n'est pas unique. Par la suite, une frontière de Pareto est retrouvée après quelques milliers de particularisation de la solution. Pour le cas d'un four ayant deux pièces d'épaisseur différente à l'intérieur, une surface de Pareto a été identifiée. Finalement, un autre modèle de contrôle s'inscrivant dans le cadre de la DDDAS a été présenté. Ce modèle montre la supériorité de la PGD face aux techniques classiques où une évaluation de la solution PGD est beaucoup plus rapide que le calcul par différences finies.

Perspectives

La PGD est une méthode de calcul prometteuse en simulation mécanique. Des solutions simplement inimaginables par des techniques classiques sont facilement construites par la PGD. De plus, la représentation séparée facilite le stockage des informations et économise amplement l'espace mémoire. Cependant, le défi n'est toujours pas gagné. La résolution paramétrique non-linéaire reste peu évidente. Les travaux continuent dans cette direction au sein de l'équipe.

D'autre part, l'estimation d'erreur et le remaillage par les méthodes classiques ne s'appliquent pas forcément sur les espaces séparés. Ainsi, de l'énergie devra être investie dans cette direction afin de pouvoir retrouver un maillage optimal adapté aux domaines de calcul et aux méthodes de résolution.

Finalement, l'algorithme de calcul des modes par la PGD est bon en tant que tel. En revanche cet algorithme pourra être amélioré afin d'obtenir une solution contenant le nombre optimal de modes nécessaires à sa représentation.





Références

- Albert, G. et Fernlund, G., 2002, “Sring-in and Warpage of Angled Composite Laminates,” *Composites Science and Technology* 62, pp. 1895–1912.
- Allery, C., Beghein, C., et Hamdouni, A., 2005, “Applying Proper Orthogonal Decomposition to the Computation of Particle Dispersion in a Two-Dimensional Ventilated Cavity,” *Communications in Nonlinear Science an Numerical Simulation* 10, pp. 907–920.
- Ammar, A., Chinesta, F., Diez, P., *et al.*, 2010a, “An Error Estimator for Separated Representation of Highly Multidimensional Models,” *Computer Methods in Applied Mechanics and Engineering* 199, pp. 1872–1880.
- Ammar, A., Mokdad, B., Chinesta, F., *et al.*, 2006, “A New Family Of Solvers For Some Classes Of Multidimensional Partial Differential Equations Encountered In Kinetic Theory Modelling Of Complex Fluids.” *J. Non Newtonian Fluid Mech.* 139, pp. 153–176.
- Ammar, A., Mokdad, B., Chinesta, F., *et al.*, 2007, “A New Family Of Solvers For Some Classes Of Multidimensional Partial Differential Equations Encountered In Kinetic Theory Modelling Of Complex Fluids. PartII : Transient Simulation Using Space-Time Seperated Representations.” *Journal of Non Newtonian Fluid Mechanics* 144, no. 2-3, pp. 98–121.
- Ammar, A., Normandin, M., Daim, F., *et al.*, 2010b, “Non-Incremental Strategies Based on Separated Representations: Applications in Computational Rheology,” *Communications in Mathematical Sciences* 8/3, pp. 671–695.
- Amsallem, D. et Farhat, C., 2011, “Projection-Based Reduced-Order Modeling,” *Stanford University Reduced Order modelling Course*.
- Andrieux, S. et Abda, A. B., 1996, “Identification of planar cracks by complete overdetermined data: inversion formulae,” *Inverse Problems* 12, pp. 553–563.

- Andrieux, S. et Baranger, T. N., 2008, “Energy Method For Cauchy Problems Of Evolutions Equations,” *Journal of Physics* 135, pp. 135–142.
- Ashby, M. F. et Jones, D. R., 1991, *Matériaux, microstructure et mise en oeuvre*, Dunod.
- Atwell, J. A. et King, B. B., 2001, “Proper Orthogonal Decomposition for Reduced Basis Feedback Controllers for Parabolic Equations,” *Mathematical and Computer Modelling* 33, pp. 1–19.
- authors, V., 2006, “DDDAS Workshop 2006,” Tech. rep., National Science Foundation, Arlington, VA, U.S.A.
- Bader, B. W. et Kolda, T. G., 2007, “Efficient MATLAB Computations With Sparse and factored tensors.” *Society for Industrial and Applied Mathematics Journal on Scientific Computing* 30, no. 1, pp. 205–231.
- Bailleul, J., Guyonvarch, G., Garnier, B., *et al.*, 1996a, “Identification des propriétés thermiques de composites fibres de verre / résines therm durcissables. Application à l’optimisation des procédés de moulage,” *Revue Générale de Thermique* 35, pp. 65–77.
- Bailleul, J., Jarny, Y., Delaunay, D., *et al.*, 1994, “Characterization of Heat Source in Materials During Transformation. Experimental Validation With Rubber and Epoxy Resins,” *Taylor Francis Ed.* 7, pp. 203–208.
- Bailleul, J. L., Delaunay, D., et Jarny, Y., 1996b, “Determination of Temperature Variable Properties of Composite Materials: Methodology and Experimental Results,” *Journal of Reinforced Plastics and Composites* 15, pp. 480–496.
- Banthia, N., Azzabi, M., et Pigeon, M., 1993, “Restrained Shrinkage Cracking in fiber Reinforced Cementitious Composites,” *Materials and Structures* 26, no. 161, pp. 405–413.
- Barasinski, A., Ghnatios, C., Bognet, B., *et al.*, 2012, “In Situ Consolidation for Thermoplastic Tape Placement Process is Not Obvious,” *Groupement de recherche GDR : Mise en oeuvre des composites et propriétés induites*.
- Bejan, A., 2004, *Convection Heat Transfer*, Wiley.
- Benazzouk, A., Douzane, O., Mezreb, K., *et al.*, 2008, “Thermal Conductivity of Cement Composites Containig Rubber Waste Particles : Expérimental Study and Modelling,” *Construction and Building Materials and Structures* 22, no. 4, pp. 573–579.
- Bernardi, C. et Maday, Y., 1997, “Spectral Methods,” *Handbook of Numerical Analysis* 5, pp. 209–485.
- Berreur, L., de Maillard, B., et Nösperger, S., 2002, “L’industrie française des matériaux



composites,” Tech. rep., Nodal Consultants.

Bird, R. B., Stewart, W. E., et Lightfoot, E. N., 1960, *Transport Phenomena*, Wiley.

Bogetti, T. A. et Gillespie, J. W., 1992, “Process-induced stress and deformation in thick-section thermoset composite laminates,” *Journal of Composite Materials* 26, no. 5, pp. 626–659.

Bognet, B., Bordeu, F., Chinesta, F., *et al.*, 2012, “Advanced Simulation of Models Defined in Plate Geometries : 3D Solutions with 2D Computational Complexity,” *Computer Methods in Applied Mechanics and Engineering* 201-204, no. 1, pp. 1–12.

Bognet, B., Ghnatios, C., Barasinski, A., *et al.*, 2011, “Composites Manufacturing Processes - Towards an advanced simulation,” *Revue des Composites et des Matériaux Avancés* 21, pp. 23–32.

Bonithon, G., Joyot, P., Chinesta, F., *et al.*, 2011, “Non-Incremental Boundary Element Discretization of Parabolic Models Based on the Use of the Proper Generalized Decompositions,” *Engineering Analysis with Boundary Elements* 35, no. 1, pp. 2–17.

Bonnet, M., 1995, “BIE and Material Differentiation Applied to the Formulation of Obstacle Inverse Problems,” *Engineering Analysis with Boundary Elements* 15, pp. 121–136.

Boucard, P.-A. et Ladeveze, P., 1999, “Une application de la méthode LATIN au calcul multirésolution de structures non linéaires,” *Revue Européenne des éléments finis* 8, pp. 903–920.

Box, G. E. P., 1979, “Robustness in the Strategy of Scientific Model Building,” *Robustness in Statistics: Proceedings of a Workshop*.

Boyard, N., Millischer, A., Sobotka, V., *et al.*, 2007, “Behaviour of a Moulded Composite Part : Modelling of Dilatometric Curve (Constant Pressure) or Pressure (Constant Volume) with Temperature and Conversion Degree Gradients,” *Composites Science and Technology* 27, no. 6, pp. 943–954.

Breitkopf, P., Naceur, H., Racineux, A., *et al.*, 2005, “Moving Least Squares Response Surface Approximation Formulation And Metal Forming Applications,” *Computers And Structures* 83, pp. 1411–1428.

Breitkopf, P., Rassineux, A., et Villon, P., 2002, “An Introduction To Moving Least Squares Meshfree Methods,” *Meshfree Computational Mechanics* 11, pp. 825–867.

Bui, H. D., 1993, *Introduction aux problèmes inverses en mécanique des matériaux*, De Boeck Université.



- Bungartz, H.-J. et M.Griebel, 2004, “Sparse Grids,” *Acta Numerica* 13, pp. 1–123.
- Chapman, T. J. ., Gillespie, J. W., Pipe, R. B., *et al.*, 1990, “Prediction of process-induced residual-stress in thermoplastic composites,” *Journal of Composite Materials* 24, no. 6, pp. 616–643.
- Chen X, Xie H, C. H., 2010, “Optimisation for CFRP pultrusion process based on 927 genetic algorithm-neutral network,” *International Journal of Material Forming* 3, pp. 1391–1399.
- Cheung, A., Yu, Y., et Pochiraju, K., 2004, “Three Dimensional Finite Element Simulation of Curing of Polymer Composites,” *Finite Elements in Analysis and Design* 40, pp. 895–912.
- Chinesta, F., 2011, “Reduced basis, POD, PGD Closing Discussion,” *Reduction Techniques: a Breakthrough in Computational Engineering?*.
- Chinesta, F., Ammar, A., et Cueto, E., 2009a, “On The Use Of Model Reduction In Material And Processes Simulation: A Way For Realizing Simulation Dreams,” Tech. rep., EADS Corporate Foundation Chaire, GeM CNRS - Centrale Nantes.
- Chinesta, F., Ammar, A., et Cueto, E., 2009b, “Recent Advances And New Challenges In the Use Of The Proper Generalized Decomposition For Solving Multidimensional Models,” *Archives of Computational Methods in Engineering* 17, no. 4, pp. 327–350.
- Chinesta, F., Ammar, A., et Cueto, E., 2010a, “Proper Generalized Decomposition of Multiscale Models,” *International Journal of Numerical Methods in Engineering* 83, no. 8-9, pp. 1114–1132.
- Chinesta, F., Ammar, A., et Cueto, E., 2010b, “Recent Advances in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models,” *Archives of Computational Methods in Engineering* 17, no. 4, pp. 327–350.
- Chinesta, F., Ammar, A., Leygue, A., *et al.*, 2011a, “An Overview of the Proper Generalized Decomposition With Applications in Computational Rheology,” *Journal of Non-Newtonian Fluid Mechanics* 166, no. 11, pp. 578–592.
- Chinesta, F., Ladeveze, P., et Cueto, E., 2011b, “A Short Review on Model Order Reduction Based on Proper Generalized Decomposition,” *Archive of Computational Methods in Engineering* 18, pp. 395–404.
- Chinesta, F., Ladeveze, P., et Maday, Y., 2012, “Advanced Order Model Reduction in Engineering and Sciences,” Tech. rep., GDR CNRS - Computational Structural Mechanics Association.



-
- Chinesta, F., Poitou, A., Leygue, A., *et al.*, 2010c, “Modélisation Avancée de la Mise en Forme de Composites,” *Matériaux 2010*, Palais des congrès, Nantes.
- Cimetiere, A., Delvare, F., Jaoua, M., *et al.*, 2001, “Solution of the Cauchy Problem Using Iterated Tikhonov Regularization,” *Inverse Problems* 17, pp. 553–570.
- Cimetiere, A., Delvare, F., et Pons, F., 2000, “Une Méthode Inverse à Régularisation Evanescence,” *Mechanics of Solids and Structures* 328 (2b), pp. 639–644.
- Cimetiere, A., Delvare, F., et Pons, F., 2005, “Une méthode inverse d’ordre un pour les problèmes de complétion de données,” *C. R. Mécanique* 333, pp. 123–126.
- Cortas, R., 2012, *Nouvelle approche expérimentale pour la maîtrise de la fissuration du béton jeune : Influence de la nature et de la saturation des granulats*, Ph.D. Thesis, Ecole Centrale Nantes.
- Darema, F., 1994, “Engineering, Scientific and Commercial Applications : Differences, Similarities and Future Evolution,” *Proceedings of the Second Hellenic European Conference on Mathematics and Informatics, HERMIS* 1, pp. 367–374.
- Dennis, J. J. et Schnabel, R., Corrected reprint of the 1983 original, 1996, *Numerical Methods For Unconstrained Optimization and Nonlinear Equations*, vol. 16, Society for Industrial and Applied Mathematics (SIAM).
- der Vorst, H. V. et Dekker, K., 1988, “Conjugate Gradient Type Method and Preconditioning,” *Journal of Computational and Applied Mathematics* 24, pp. 73–87.
- Donea, J. et Huerta, A., 2003, *Finite element method for flow problem*, Wiley.
- Dreau, K., Chevaugeon, N., et Moes, N., 2010, “Studied X-FEM Enrichment to Handle Material Interfaces with Higher Order Finite Element,” *Computer Methods in Applied Mechanics and Engineering* 199, pp. 1922–1936.
- Favre, J. P., 1988, “Residual Thermal Stresses in Fiber Reinforced Composite Materials - A Review,” *Journal of Mechanical Behavior of Materials* 1, no. 1-4, pp. 37–53.
- Fink, J. et Rheinboldt, W., 1983, “On the Error Behavior of Reduced Basis Technique for Nonlinear Finite Element Approximations,” *ZAMM* 63, pp. 21–28.
- Ghnatios, C., Chinesta, F., Barasinski, A., *et al.*, 2011a, “Modélisation et simulation 3D paramétrique des champs de températures dans des plaques composites,” *MECAMAT 2011*.
- Ghnatios, C., Chinesta, F., Cueto, E., *et al.*, 2010, “Optimizing Composites Forming Processes by Applying the Proper Generalized Decomposition,” *AIP Conference Proceedings* 1315, pp. 197–202.



- Ghnatios, C., Chinesta, F., Cueto, E., *et al.*, 2011b, “Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion,” *Composite Part A: Applied Science and Manufacturing* 42, pp. 1169–1178.
- Ghnatios, C., Leygue, A., et Chinesta, F., 2011c, “Calcul inverse « on-line » basé sur des solutions « off-line » PGD paramétriques,” *Méthodes numériques & méthodologies avancées pour les problèmes inverses en mécanique*, INSA Lyon.
- Ghnatios, C., Leygue, A., Chinesta, F., *et al.*, 2011d, “Stratégies avancées d’optimisation et analyse inverse basées sur l’utilisation de la PGD,” *Vingtième congrès français de la mécanique*.
- Ghnatios, C., Masson, F., Huerta, A., *et al.*, 2012, “Proper Generalized Decomposition Based Dynamic Data-Driven of Thermal Processes,” *Computer Methods in Applied Mechanics and Engineering* 213-216, pp. 29–41.
- Ghnatios, C., Masson, F., Leygue, A., *et al.*, 2011e, “Proper Generalized Decomposition Based Dynamic Data-Driven Control of Materiel Forming Processes,” *AIP Conference Proceedings* 1353, pp. 1163–1168.
- Glowinski, R., 1980, “An Efficient Preconditioning Scheme for Iterative Numerical Solutions of Partial Differential Equations,” *Applied Mathematical Modelling* 4, no. 3, pp. 187–192.
- Gonzalez, D., Ammar, A., Chinesta, F., *et al.*, 2009, “Recent Advances on the Use of Separated Representations,” *International Journal for Numerical Methods in Engineering* 81, pp. 637–659.
- Gornet, L., Leveque, D., et Perret, L., 2000, “Modélisation, identification et simulations éléments finis des phénomènes de délaminage dans les structures composites stratifiées,” *Mécanique et Industries* 1(3), pp. 267–276.
- Grédiac, M., 2011, “Identification de paramètres constitutifs à partir de mesures de champs avec la méthode des champs virtuels,” *Méthodes numériques et méthodologies avancées pour les problèmes inverses en mécanique*.
- Grinzato, E., Marinetti, S., Bison, P. G., *et al.*, 1995, “Application of Neutral Network to Thermographic Data Reduction for Non-Destructive Evaluation,” *Revue Générale de Thermique* 397, pp. 17–27.
- Guyonvarch, G., Recouvreur, M., Soufflet, B., *et al.*, 1994, “Détermination de la capacité thermique, du diagramme de phase et de la cinétique de réticulation d’une résine polyester par calorimétrie différentielle.” *Colloque SFT*.



-
- Han, H. et Reinhardt, H.-J., 1997, "Some Stability Estimates for Cauchy Problems for Elliptic Equations," *Inverse Ill-Posed Problems* 5, pp. 437–454.
- Hexel, 2010, "Epoxy Matrix Product Data," Tech. rep., HexPly.
- Horn, R. A. et Johnson, C. R., 1985, *Matrix Analysis*, Cambridge University Press, Cambridge, UK.
- Huynh, D., Knezevic, D., et Patera, A., 2012, "Certified Reduced Basis Model Validation : A Frequentistic Uncertainty Framework," *Computer Methods in Applied Mechanics and Engineering* 201-204, pp. 13–24.
- Jacob, A., 2006, "Globalisation Of The Pultrusion Industry," *Reinforced Plastics* 50(5), pp. 38–41.
- Kaminski, M., 2003, "Homogenization of Transient Heat Transfer Problems for Some Composite Materials," *International Journal of Engineering Science* 41, no. 1, pp. 1–29.
- Kolda, T. G. et Bader, B. W., 2009, "Tensor Decompositions and Applications," *Society for Industrial and Applied Mathematics* 51, no. 3, pp. 455–500.
- Kostopoulos, D., Beniere, F., et Reddy, K., 1979, "Examen de la loi d'Arrhenius de la diffusion des solides," *Journal of Physics and Chemistry of Solids* 40, pp. 357–367.
- Krapez, J. C., Balageas, D., Deom, A., et al., 1994, "Early Detection by Stimulated Infrared Thermography. Comparison With Ultrasonics and Holo/Shearography. In Advances in Signal Processing for Non-Destructive Evaluation Materials." *NATO ASI Series* 262, pp. 303–321.
- Ladeveze, P., 1985a, "New Algorithms: Mechanical Framework and Developments," Tech. Rep. 57, LMT-Cachan.
- Ladeveze, P., 1985b, "On a Family of Algorithms For Structural Mechanics," *Compte Rendu Academie des Sciences Paris* 300/2, pp. 41–44.
- Ladeveze, P., 1996, *Mecanique non-lineaire des structures*, Hermes, Paris.
- Ladeveze, P., 1999, *Nonlinear Computational Structural Mechanics*, Mechanical Engineering Series, Springer NY.
- Ladeveze, P., Passieux, J.-C., et Neron, D., 2010, "The LATIN Multiscale Computational Method and the Proper Generalized Decomposition," *Computer Methods in Applied Mechanics and Engineering* 199, pp. 1287–1296.
- Lagarias, J. C., Reeds, J. A., Wright, M. H., et al., 1998, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal of Optimization* 9, no. 1, pp. 112–147.



- Lesnic, D., Elliot, L., et Ingham, D., 1997, “An Alternative Boundary Element Method for Solving Numerically the Cauchy Problem for the Laplace Equation,” *Engineering Anal. Bound. Elem.* 20, pp. 123–133.
- Levenberg, K., 1944, “A Method for the Solution of Certain Problems in Least Squares,” *Quarterly of Applied Mathematics* 2, pp. 164–168.
- Leygue, A. et Verron, E., 2010, “A First Step Towards The Use Of Proper Generalized Decomposition Method For Structural Optimisation,” *Archives of Computational Methods in Engineering* 17, no. 4, pp. 465–472.
- Li, M., Zhu, Q., Geubelle, P., *et al.*, 2001, “Optimal Curing of Thermoset Matrix Composites: Thermochemical Considerations,” *Polymer Composites* 22, pp. 118–131.
- Liberge, E. et Hamdouni, A., 2010, “Reduced Order Modelling Method via Proper Orthogonal Decomposition (POD) For Flow Around And Oscillating Cylinder,” *Journal of Fluid and Structures* 26, pp. 292–311.
- Loeve, M., 1945, *Fonctions aleatoires du second ordre*, Compte rendu de l’academie des sciences.
- Loix, F., 2005, *Modelling of transport phenomena in porous media*, Ph.D. Thesis, Universitas Catholica Lovaniensis.
- Low, I. M., 1990, “Effect of Residual Stresses on the Failure Micromechanisms in Toughened Epoxy Systems,” *Journal of Material Science* 25, no. 4, pp. 2144–2148.
- Machiels, L., Maday, Y., Oliveira, I. B., *et al.*, 2000, “Output Bounds For Reduced-Basis Approximations of Symetric Positive Definite Eigenvalue Problems,” *Comptes Rendus de L’Académie des Sciences - Series 1* 331(2), pp. 153–158.
- Maday, Y. et Ronquist, E. M., 2002, “A Reduced Basis Element Method,” *Comptes Rendus Mathématique* 335(2), pp. 195–200.
- Maillard, S., Cadith, J., Legros, G., *et al.*, 2009, “Thermographie et matériaux métalliques,” *Congrès Thermogram’ 2009*, pp. 41–52.
- Maillard, S., Cadith, J., Walaszek, H., *et al.*, 2008, “La thermographie infrarouge stimulée, une nouvelle technique de contrôle sur les lignes de production?” *COFREND*, pp. 93–103.
- Maillet, D., Bastsale, J. C., Bendala, A., *et al.*, 1996, “Méthodes intégrales et contrôle non destructif par thermographie infrarouge stimulée,” *Revue Générale de Thermique* 35, pp. 5–13.
- Manson, J. A. E. et Barbero, E. J., 1992, “Process Simulated Laminate: A Methodo-



- logy to Internal Stress Characterization in Advanced Composite Materials,” *Journal of Composite Materials* 26, no. 3, pp. 405–431.
- Marcin, L., Maire, J.-F., et Carrère, N., 2009, “Strength Analysis of 3D Woven Composite Structures With High Stress Gradient,” *JNC 16 Toulouse*, Toulouse France.
- Marquardt, D., 1963, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *SIAM Journal on Applied Mathematics* 11, pp. 431–441.
- Martin, J., 2006, “Pultruded Composites Compete With Traditional Construction Materials.” *Reinforced Plastics* 50(5), pp. 20–27.
- Messiter, M. et Shamash, Y., 1985, “Product and Sum Separable Functions,” *IEEE Transactions on Automatic control* 30, no. 7, pp. 694–697.
- Moes, N., 2011, “The Extended Finite Element Method,” *Computational Mechanics: Seminars and Lectures Week* (E. D. Spiga, ed.).
- Moes, N., Dolbow, J., et Belytschko, T., 1999, “A Finite Element Method for Crack Grows Without Remeshing,” *International Journal for Numerical Methods in Engineering* 46, pp. 131–150.
- Msallem, Y. A., 2008, *Caractérisation thermique et mécanique d’un matériau composite aéronautique pendant le procédé d’élaboration. Contribution à l’estimation des contraintes résiduelles*, Ph.D. Thesis, Ecole Centrale de Nantes.
- Msallem, Y. A., Jacquemin, F., Boyard, N., et al., 2010a, “Material Characterization and Residual Stresses Simulation During the Manufacturing Process of Epoxy Matrix Composite,” *Composites Part A: Applied Science and Manufacturing* 41/1, pp. 108–115.
- Msallem, Y. A., Jacquemin, F., et Poitou, A., 2010b, “Residual Stress Formation During the Manufacturing Process of Epoxy Matrix Composite,” *International Journal of Material Forming* 3/22, pp. 1363–1372.
- Nagy, D. A., 1977, “Model Representation of Geometrically Nonlinear Behavior by the Finite Element Method,” *Computers and Structures* 10, pp. 683–688.
- Nakouzi, S., Pancrace, J., Schmidt, F., et al., 2011, “Simulation of Infrared Composite Curing Process,” *Advanced Engineering Materials* 13, pp. 604–608.
- Nicodeau, C., 2005, *Modélisation du soudage en continu de composites à matrice thermoplastique*, Ph.D. Thesis, ENSAM Paris.
- Niliot, C. L. et Gallet, P., 1998, “Infrared Thermography Applied to the Resolution of Inverse Heat Conduction Problems: Recovery of Heat Line Sources and Boundary



- Conditions,” *Revue Générale de Thermique* 37, pp. 629–643.
- Niroomandi, S., Alfaro, I., Gonzalez, D., *et al.*, 2012, “Real Time Simulation of Surgery by Reduced Order Modelling and X-FEM Techniques,” *Numerical Methods in Biomedical Engineering* 28, no. 5, pp. 574–588.
- Nocedal, J. et Wright, S. J., 1999, *Numerical Optimization*, Springer.
- Noor, A. K. et Peters, J. M., 1980, “Reduced Basis Technique for Nonlinear Analysis of Structures,” *AIAA Journal* 18, pp. 455–462.
- Nouy, A., 2007, “A Generalized Spectral Decomposition Technique to Solve a Class of Linear Stochastic Partial Differential Equations,” *Computer Methods in Applied Mechanics and Engineering* 196, pp. 4521–4537.
- Nowacki, W. K. et Zarka, J., 2008, “On temperature in thermoelastoplasticity,” *Comptes Rendus Mécanique* 336, pp. 210–233.
- Pajani, D., 1989, *Mesure par thermographie infrarouge*, ISBN 2-9504171-0-8.
- Patera, A. T. et Ronquist, E. M., 2007, “Reduced basis approximation and a posteriori error estimation for a Boltzmann model,” *Computer Methods in Applied Mechanics and Engineering* 196, pp. 2925–2942.
- Perotto, S., 2010, “Hierarchical Local Model Reduction for Elliptic Problems : A Domain Decomposition Approach,” *Multiscale Model Simulation* 8, no. 4, pp. 1102–1127.
- Porsching, T., 1985, “Estimation of the Error in Reduced Basis Method Solution of Nonlinear Equations,” *Mathematical and Computer Modelling* 45(172), pp. 487–496.
- Porsching, T. et Lee, M. L., 1987, “The Reduced Basis Method for Initial Value Problems,” *Society for Industrial and Applied Mathematics Journal of Numerical Analysis* 24, no. 24, pp. 1277–1287.
- Pruliere, E., Chinesta, F., et Ammar, A., 2010a, “On the Deterministic Solution of Multidimensional Parametric Models Using the Proper Generalized Decomposition,” *Mathematics And Computer Simulation* 81, pp. 791–810.
- Pruliere, E., Ferec, J., Chinesta, F., *et al.*, 2010b, “An Efficient Reduced Simulation of Residual Stresses in Composite Forming Processes,” *International Journal of Material Forming* 3, pp. 1339–1350.
- Rabearison, N., Jochum, C., et Grandidier, J. C., 2009, “A FEM Coupling Model for Properties Prediction During the Curing of an Epoxy Matrix,” *Computational Materials Science* 45, pp. 715–724.
- Ryan, M. et Dutta, A., 1979, “Kinetics of Epoxy cure : Rapid Technique for Kinetic



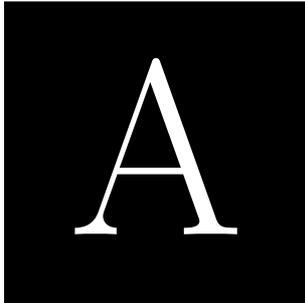
- Parameter Estimation,” *Polymer* 20, no. 2, pp. 203–206.
- Ryckelynck, D., 2005, “A priori Hyper-Reduction Method: An Adaptive Approach,” *Journal of Computational Physics* 202, pp. 346–366.
- Ryckelynck, D. et Benziane, D. M., 2010, “Multi-level A Priori Hyper-Reduction of Mechanical Models Involving Internal Variables,” *Computer Methods in Applied Mechanics and Engineering* 199, pp. 1134–1142.
- S. Sourour, M. K., 1976, “Differential Scanning Calorimetry of Epoxy Cure: Isothermal Cure Kinetics,” *Thermochimica Acta* 14, pp. 41–59.
- Safjan, A., 1988, “Nonlinear Structural Analysis via Reduced Basis,” *Computers and Structures* 29, pp. 1055–1061.
- Scovazzi, G., 2007, “A discourse on Galilean invariance, SUPG stabilization, and the variational multiscale framework,” *Computer Methods in Applied Mechanics and Engineering* 196, pp. 1108–1132.
- Sirivedin, S., Fenner, D. N., Nath, R. B., *et al.*, 2000, “Matrix Crack Propagation Criteria for Model Sort Carbon Fiber/ Epoxy Composites,” *Computer Science and Technology* 60, pp. 2835–2847.
- Svanberg, J. M. et Holmberg, J. A., 2004, “Prediction of Shape Distortions Part II, Experimental Validation and Analysis of Boundary Conditions,” *Composites Part A: Applied Science and Manufacturing* 35, pp. 723–734.
- Temlyakov, V. N. et Zheltov, P., 2011, “On Performance of Greedy Algorithms,” *Journal of Approximation Theory* 163, no. 9, pp. 1134–1145.
- Thionnet, A. et Renard, J., 2001, “General Methodology to Predict Viscoelastic Behaviour of Composite Laminates,” *Mécanique & Industries* 2, no. 2, pp. 143–147.
- Toussaint, E., Grediac, M., et Pierron, F., 2006, “The Virtual Fields Method with Piecewise Virtual Fields,” *International Journal of Mechanical Sciences* 48, pp. 256–264.
- Vorechovsky, M., 2008, “Simulation Of Simply Cross Correlated Random Fields by Series Expansion Methods,” *Structural Safety* 30, pp. 337–363.
- Wissnom, M. R., Gigliotti, M., Ersoy, N., *et al.*, 2006, “Mechanisms Generating Residual Stresses and Distorsion During Manufacture of Polymer-Matrix Composite Structures,” *Composite Part A: Applied Science and Manufacturing* 37, pp. 522–529.
- Zhu, Q., Geubelle, P., Li, M., *et al.*, 2001, “Dimensional Accuracy of Thermoset Composites: Simulation of Process-Induced Residual stress,” *Journal of Composite Materials*



35, pp. 2171–2205.

Zienkiewicz, O. C. et Zhu, J. Z., 1987, “A Simple Error Estimator and Adaptative Procedure for Practicale Engineering Analysis,” *International Journal for Numerical Methods in Engineering* 24, pp. 337–357.





Exemple de construction des tenseurs élémentaires

A.1 Exemple d'enrichissement d'un système 2D	201
A.2 Problème multidimensionnel	203

A.1 Exemple d'enrichissement d'un système 2D

Considérons l'équation de Poisson suivante :

$$\Delta T = -f(x,y) \tag{A.1}$$

T étant une fonction de (x,y) . On résout donc l'équation (A.1) dans un domaine Ω 2D tel que $\Omega =]-L; L[\times]-L; L[$. La solution séparée s'écrit sous la forme :

$$T \approx \sum_{i=1}^{i=m} \alpha_i F_i(x) G_i(y) \tag{A.2}$$

Considérons la somme connue jusqu'au rang n et on cherche à déterminer le mode $n + 1$. On note N_j (respectivement M_k) les fonctions de formes de l'approximation éléments finis en x (respectivement y). On note que l'approximation éléments finis est donc 1D par domaine. Notons les vecteurs contenant les valeurs des fonctions N_j et M_k par \mathbf{N} et \mathbf{M} respectivement. Lors de l'enrichissement, on note la fonction test par T^* . Ainsi à l'étape d'enrichissement

$n + 1$ on obtient :

$$\begin{cases} T(x,y) = \sum_{i=1}^{i=n} \alpha_i F_i(x) G_i(y) + R(x) S(y) \\ T^*(x,y) = R^*(x) S(y) + R(x) S^*(y) \end{cases} \quad (\text{A.3})$$

Considérons de plus R , S , F_i et G_i des projections sur un maillage des vrais fonctions. Ainsi, les écritures matricielles des fonctions T et T^* ainsi que leurs gradients seront :

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \sum_{i=1}^{i=n} \alpha_i \begin{bmatrix} d\mathbf{N}^T F_i \mathbf{M}^T G_i \\ \mathbf{N}^T F_i d\mathbf{M}^T G_i \end{bmatrix} + \begin{bmatrix} \mathbf{M}^T S d\mathbf{N}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{N}^T R d\mathbf{M}^T \end{bmatrix} \times \begin{bmatrix} R \\ S \end{bmatrix} = \sum_{i=1}^{i=n} \alpha_i \mathbf{D}_i + \mathcal{E} \times \begin{bmatrix} R \\ S \end{bmatrix}, \quad (\text{A.4})$$

$$T^*(x,y) = \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \times \begin{bmatrix} S^T \mathbf{M} \mathbf{N} \\ R^T \mathbf{N} \mathbf{M} \end{bmatrix}, \quad (\text{A.5})$$

$$\begin{bmatrix} \frac{\partial T^*}{\partial x} & \frac{\partial T^*}{\partial y} \end{bmatrix} = \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \times \begin{bmatrix} S^T \mathbf{M} d\mathbf{N} & S^T d\mathbf{M} \mathbf{N} \\ R^T d\mathbf{N} \mathbf{M} & R^T \mathbf{N} d\mathbf{M} \end{bmatrix} = \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \times \mathcal{F}^T \quad (\text{A.6})$$

En supposant que le second membre $f(x,y)$ s'écrit sous forme séparée, $f = \sum_{h=1}^{h=l} a_h(x) b_h(y)$, l'écriture de la formulation faible devient :

$$\begin{aligned} \int_{\Omega} \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \sum_{i=1}^{i=n} \alpha_i \mathcal{F}^T \mathbf{D}_i d\Omega + \int_{\Omega} \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \mathcal{F}^T \mathcal{E} \begin{bmatrix} R \\ S \end{bmatrix} d\Omega = \\ = \sum_{h=1}^{h=l} \int_{\Omega} \begin{bmatrix} R^{*T} & S^{*T} \end{bmatrix} \begin{bmatrix} S^T \mathbf{M} b_h \mathbf{N} a_h \\ R^T \mathbf{N} a_h \mathbf{M} b_h \end{bmatrix} d\Omega \end{aligned} \quad (\text{A.7})$$

De plus, chaque intégrale de l'équation (A.7) peut s'écrire sous forme séparée comme suit :

$$\begin{aligned} \int_{\Omega} \mathcal{F}^T \mathbf{D}_i d\Omega = & \begin{bmatrix} \left(\int_{-L}^L d\mathbf{N} d\mathbf{N}^T dx \right) F_i S^T \left(\int_{-L}^L \mathbf{M} \mathbf{M}^T dy \right) G_i \\ R^T \left(\int_{-L}^L d\mathbf{N} d\mathbf{N}^T dx \right) F_i \left(\int_{-L}^L \mathbf{M} \mathbf{M}^T dy \right) G_i \end{bmatrix} + \\ & + \begin{bmatrix} \left(\int_{-L}^L \mathbf{N} \mathbf{N}^T dx \right) F_i S^T \left(\int_{-L}^L d\mathbf{M} d\mathbf{M}^T dy \right) G_i \\ R^T \left(\int_{-L}^L d\mathbf{N} d\mathbf{N}^T dx \right) F_i \left(\int_{-L}^L d\mathbf{M} d\mathbf{M}^T dy \right) G_i \end{bmatrix} \end{aligned} \quad (\text{A.8})$$

■

Les quatre “sous-matrices” du terme $\mathcal{F}^T \mathcal{E}$ s’écrivent aussi comme suit :

$$\begin{aligned}
\int_{\Omega} \mathcal{F}^T \mathcal{E}_{11} d\Omega &= \left(\int_{-L}^L d\mathbf{N} d\mathbf{N}^T dx \right) S^T \left(\int_{-L}^L \mathbf{M} \mathbf{M}^T dy \right) S, \\
\int_{\Omega} \mathcal{F}^T \mathcal{E}_{12} d\Omega &= \left(\int_{-L}^L \mathbf{N} \mathbf{N}^T dx \right) R S^T \left(\int_{-L}^L d\mathbf{M} d\mathbf{M}^T dy \right), \\
\int_{\Omega} \mathcal{F}^T \mathcal{E}_{21} d\Omega &= \left(\int_{-L}^L \mathbf{M} \mathbf{M}^T dy \right) S R^T \left(\int_{-L}^L d\mathbf{N} d\mathbf{N}^T dx \right), \\
\int_{\Omega} \mathcal{F}^T \mathcal{E}_{22} d\Omega &= R^T \left(\int_{-L}^L \mathbf{N} \mathbf{N}^T dx \right) R \left(\int_{-L}^L d\mathbf{M} d\mathbf{M}^T dy \right)
\end{aligned} \tag{A.9}$$

Finalement, le second membre de l’équation (A.7) s’écrit aussi sous forme séparée comme suit :

$$\mathcal{V}_2 = \sum_{h=1}^{h=l} \left[\begin{array}{c} S^T \left(\int_{-L}^L \mathbf{M} b_h dy \right) \left(\int_{-L}^L \mathbf{N} a_h dx \right) \\ R^T \left(\int_{-L}^L \mathbf{N} a_h dx \right) \left(\int_{-L}^L \mathbf{M} b_h dy \right) \end{array} \right] \tag{A.10}$$

Ainsi, en notant $\mathcal{V}_1 = \int_{\Omega} \mathcal{F}^T \mathbf{D}_i d\Omega$ et $\mathcal{W} = \int_{\Omega} \mathcal{F}^T \mathcal{E} d\Omega$, une forme matricielle compact de l’équation (A.7) peut s’écrire :

$$\mathcal{V}_1(R, S) + \mathcal{W}(R, S) \begin{bmatrix} R \\ S \end{bmatrix} = \mathcal{V}_2(R, S) \tag{A.11}$$

Cette écriture nous donne la forme matricielle engendrée pour résoudre un enrichissement de l’équation (A.1) utilisant la PGD. Cet exemple est donné pour un domaine bidimensionnel. La même méthodologie de travail peut s’étendre à un domaine multidimensionnel. Ce travail est résumé dans la section A.2 de cette annexe.

A.2 Problème multidimensionnel

Dans ce qui suit, on reprend le problème de Poisson, mais en multidimensionnel, de dimension z . Notons le domaine de résolution $\Omega =]-L; L[^z$. Par suite, par la même démarche, on retrouve



les matrices \mathcal{V}_1 , \mathcal{V}_2 et \mathcal{W} suivantes :

$$\begin{aligned} \mathcal{W} &= \begin{bmatrix} \chi_{11} & \chi_{12} & \cdots & & \\ \chi_{21} & \chi_{22} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \chi_{zz} \end{bmatrix}, \\ \mathcal{V}_1 &= \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_z \end{bmatrix}, \\ \mathcal{V}_2 &= \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_z \end{bmatrix} \end{aligned} \quad (\text{A.12})$$

Dans ces projections, on retrouve donc $\chi_{k_1 k_2}$ une matrice de dimensions (p_{k_1}, p_{k_2}) , λ_{k_1} et γ_{k_2} des vecteurs de dimension $(p_{k_1}, 1)$. De plus, on retrouve :

$$\begin{cases} \chi_{k_1 k_1} = \left(\int_{-L}^L d\mathbf{N}_{k_1} d\mathbf{N}_{k_1}^T dx_{k_1} \right) \prod_{\substack{g=1 \\ g \neq k_1}}^{g=z} R_g^T \left(\int_{-L}^L \mathbf{N}_g \mathbf{N}_g^T dx_g \right) R_g \\ \chi_{\substack{k_1 k_2 \\ k_1 \neq k_2}} = \left(\int_{-L}^L \mathbf{N}_{k_1} \mathbf{N}_{k_1}^T dx_{k_1} \right) R_{k_1} R_{k_2}^T \left(\int_{-L}^L d\mathbf{N}_{k_2} d\mathbf{N}_{k_2}^T dx_{k_2} \right) \times \prod_{\substack{g=1 \\ g \neq k_1 \\ g \neq k_2}}^{g=z} R_g^T \left(\int_{-L}^L \mathbf{N}_g \mathbf{N}_g^T dx_g \right) R_g \end{cases} \quad (\text{A.13})$$

$$\lambda_{k_1} = \sum_{i=1}^{i=n} \alpha_i \left(\begin{aligned} & \left(\int_{-L}^L d\mathbf{N}_{k_1} d\mathbf{N}_{k_1}^T dx_{k_1} \right) F_{k_1 i} \times \prod_{\substack{g=1 \\ g \neq k_1 \\ g \neq k_2}}^{g=z} R_g^T \left(\int_{-L}^L \mathbf{N}_g \mathbf{N}_g^T dx_g \right) F_{g i} + \\ & + \sum_{\substack{k_2=1 \\ k_2 \neq k_1}}^{k_2=z} \left(\left(\int_{-L}^L d\mathbf{N}_{k_1} d\mathbf{N}_{k_1}^T dx_{k_1} \right) F_{k_1 i} R_{k_2}^T \left(\int_{-L}^L d\mathbf{N}_{k_2} d\mathbf{N}_{k_2}^T dx_{k_2} \right) F_{k_2 i} \times \right. \\ & \quad \left. \times \prod_{\substack{g=1 \\ g \neq k_1 \\ g \neq k_2}}^{g=z} R_g^T \left(\int_{-L}^L \mathbf{N}_g \mathbf{N}_g^T dx_g \right) F_{g i} \right) \end{aligned} \right) \quad (\text{A.14})$$



$$\gamma_{k_1} = \sum_{h=1}^{h=l} \left(\left(\int_{-L}^L \mathbf{N}_{k_1} a_{k_1 h} dx_{k_1} \right) \times \prod_{g \neq k_1} g = z_{g=1} R_g^T \left(\int_{-L}^L \mathbf{N}_g a_{gh} dx_{k_1} \right) \right) \quad (\text{A.15})$$

Ainsi le calcul des fonctions d'enrichissement se fait successivement par inversion des lignes du système obtenu, une ligne à la fois.





B

Estimation d'erreur à postériori

B.1 Problème modèle	207
B.2 Estimation sur un problème bidimensionnel	210

B.1 Problème modèle

Dans cette section, on étudie la possibilité d'utiliser les estimateurs d'erreur donnés par la méthode de [Zienkiewicz et Zhu \(1987\)](#) sur un problème de convection-diffusion résolue par la PGD dans un domaine $\Omega = [-1; 1]^2$ bidimensionnel, en séparant les dimensions du plan x et y . La conductivité K est considérée constante sur le domaine Ω , paramètre du problème. L'équation différentielle du problème modèle est :

$$v \frac{\partial \mathcal{U}}{\partial x} - K \Delta \mathcal{U} = 0 \tag{B.1}$$

Les conditions limites du problème (B.1) sont :

$$\begin{cases} \mathcal{U}(-1, y, K) = 0 \\ \mathcal{U}(1, y, K) = 1 \\ \nabla \mathcal{U}(x, -1, K) \cdot \mathbf{n} = \nabla \mathcal{U}(x, 1, K) \cdot \mathbf{n} = 0 \end{cases} \tag{B.2}$$

La solution pour une particularisation de K est montrée par la figure [B.1](#). Notons par la suite la dérivée partielle correct de \mathcal{U} suivant x , $\frac{\partial \mathcal{U}}{\partial x}$, par σ . Cependant, une interpolation linéaire

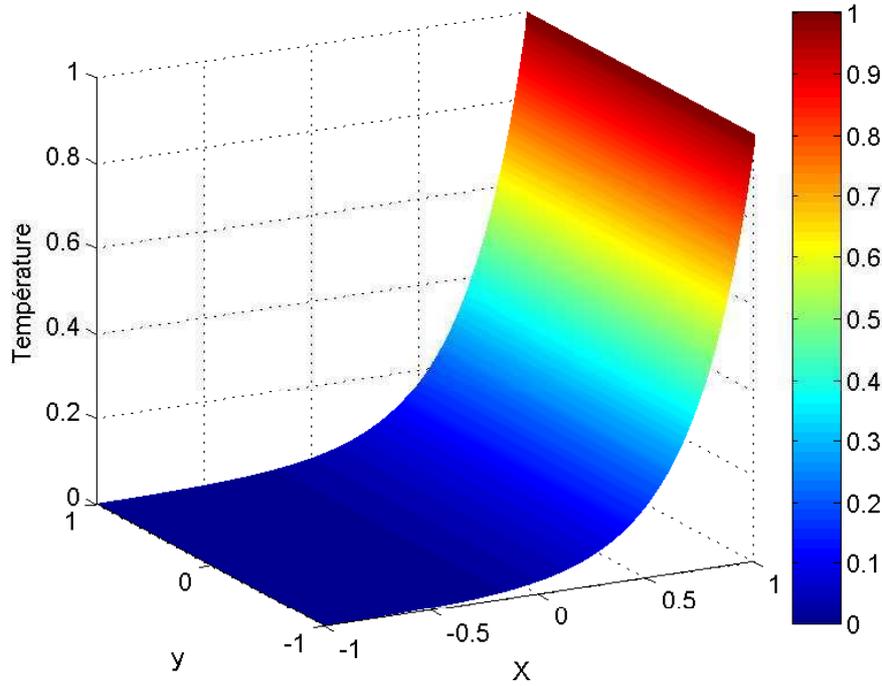


figure B.1 – Solution du problème modèle pour $K = 1$ et $v = 20$

par élément est utilisée pour \mathcal{U} , donc l'estimation donnée par le calcul éléments finis de σ , notons $\hat{\sigma}$, est donc constant par élément. Ce résultat de l'approximation éléments finis n'est pas réaliste. Par la suite, une meilleure estimation de σ , notons σ^\dagger est définie par les mêmes fonctions d'interpolation de \mathcal{U} . En d'autres termes, nous relient les milieux des discontinuités entre les éléments pour obtenir les σ^\dagger , comme le montre la figure B.2.

Normalement, l'erreur est donnée par :

$$\|e\| = \sqrt{\int_{\Omega} (\sigma - \hat{\sigma})^2 d\Omega} \quad (\text{B.3})$$

Cependant, comme σ est inconnue, σ^\dagger est utilisée pour calculer une estimation de l'erreur :

$$\|e_{es}\| = \sqrt{\int_{\Omega} (\sigma^\dagger - \hat{\sigma})^2 d\Omega} \quad (\text{B.4})$$



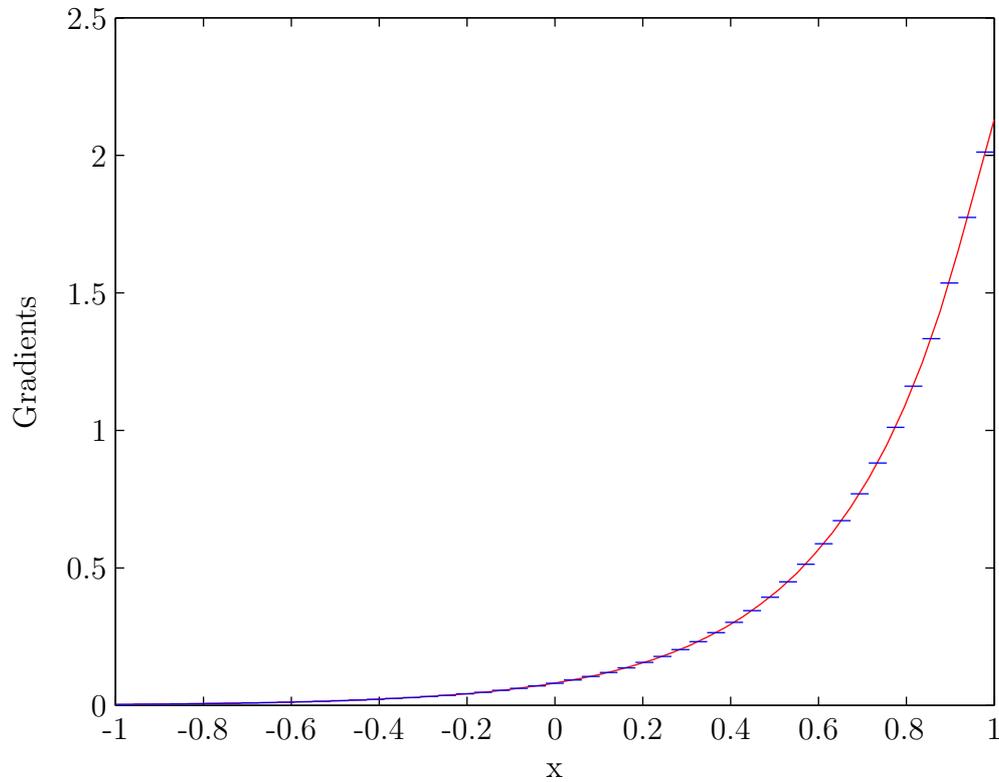


figure B.2 – $\hat{\sigma}$ en bleu et σ^\dagger en rouge

Finalement, l'erreur relative s'écrit :

$$\eta = \frac{\|e_{es}\|}{\sqrt{\hat{\sigma}^2 + \|e_{es}\|^2}} \quad (\text{B.5})$$

La stratégie de remaillage utilisée est simple, les éléments ayant $\eta > \epsilon$ sont divisés en deux. ϵ étant un seuil choisi selon le problème. Pour l'exemple montré, on choisi $\epsilon = 0.01$. La figure B.3 montre le gradient après un premier remaillage alors que la figure B.4 montre l'erreur η . La figure B.4 présente une cassure de la courbe au point de début du raffinement du maillage.

La convergence de la solution est achevée après six itérations de raffinement. La solution est montrée par la figure B.5, alors que les estimations du gradients par la figure B.6. La courbe montrée par la figure B.7 montre l'erreur en fonction du nombre des itérations.



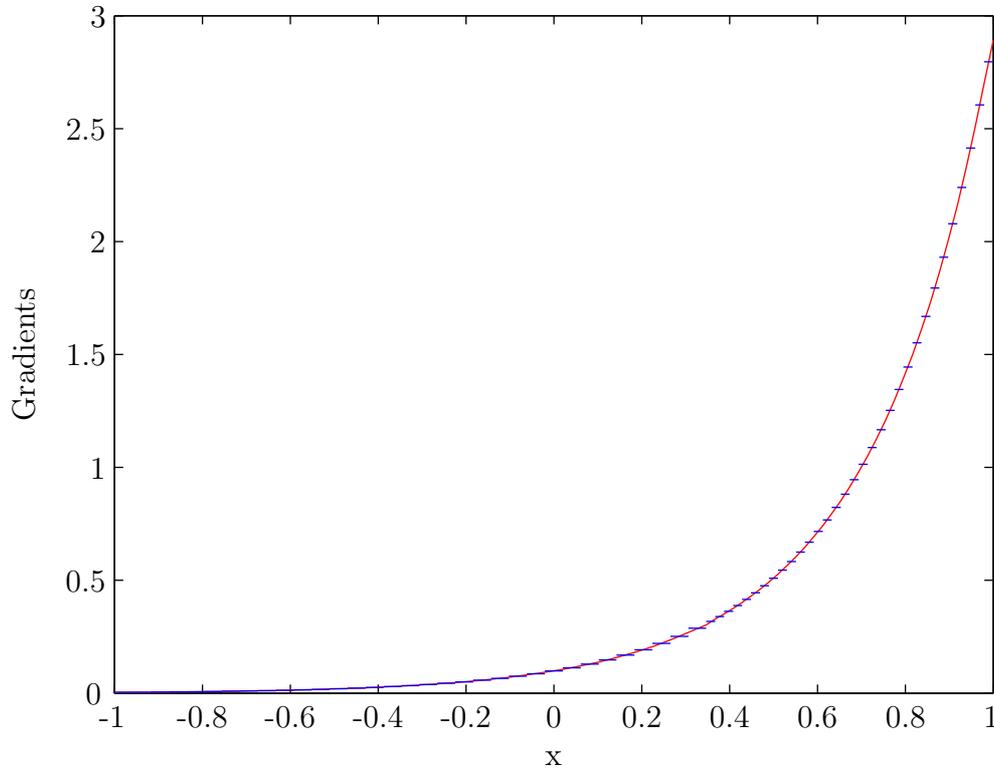


figure B.3 – $\hat{\sigma}$ en bleu et σ^\dagger en rouge après un premier raffinement

B.2 Estimation sur un problème bidimensionnel

Dans un deuxième temps, une estimation d'erreur indépendante dans les directions de x et y a été réalisée. Ainsi le problème modèle devient :

$$v_x \frac{\partial \mathcal{U}}{\partial x} + v_y \frac{\partial \mathcal{U}}{\partial y} - K \Delta \mathcal{U} = 0 \quad (\text{B.6})$$

Avec les conditions limites :

$$\begin{cases} \mathcal{U}(-1, y, K) = 0 \\ \mathcal{U}(1, y, K) = 2y \\ \mathcal{U}(x, -1, K) = 0 \\ \mathcal{U}(x, 1, K) = 2x \end{cases} \quad (\text{B.7})$$

La solution du problème est montrée dans la figure B.8. Cependant, l'estimation de l'erreur η est réalisée indépendamment suivant x et y . Ainsi on évalue l'erreur sur toutes les lignes de x et toutes les lignes de y . Par la suite, on raffine en utilisant l'erreur la plus grande.



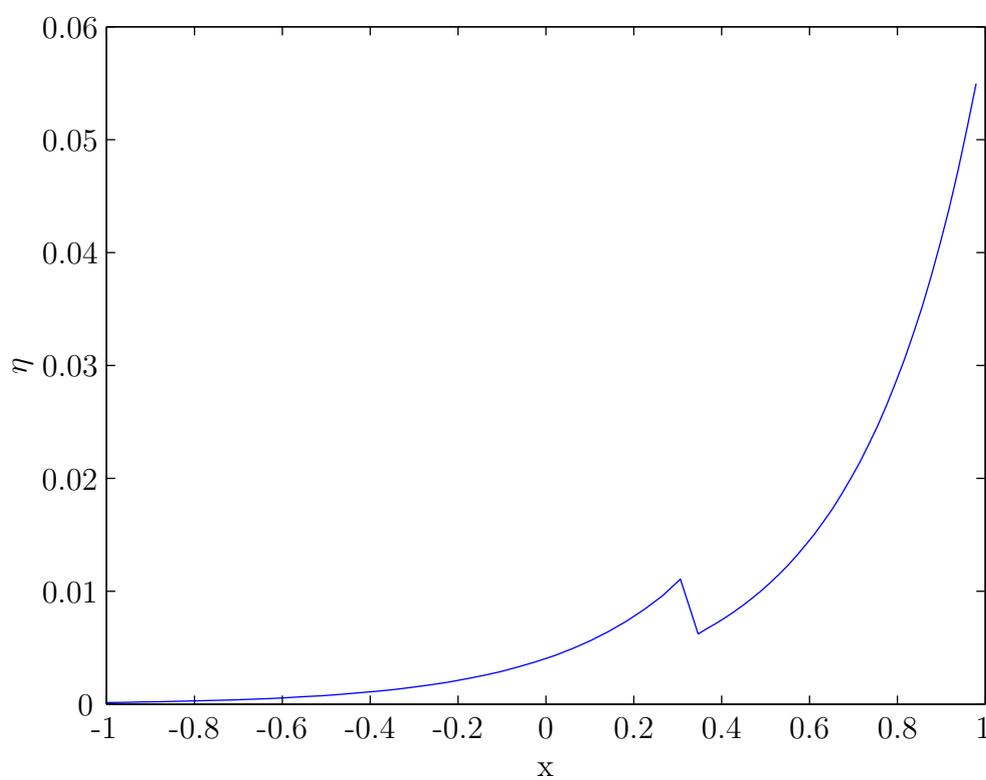


figure B.4 – η après un premier raffinement du maillage

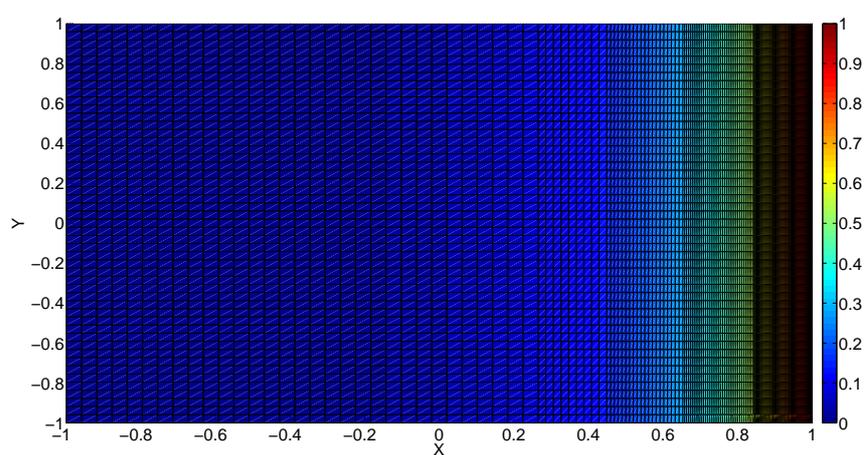


figure B.5 – U après convergence de l'algorithme de raffinement

Ainsi, nous sommes capables d'estimer une erreur fiable pour itérer sur le maillage sans avoir à reconstruire la surface. En utilisant $v_x = 10$ et $v_y = 10$ et en particulierisant $K = 1$, on obtient la solution montrée dans la figure B.9 après 6 itération de remaillage.



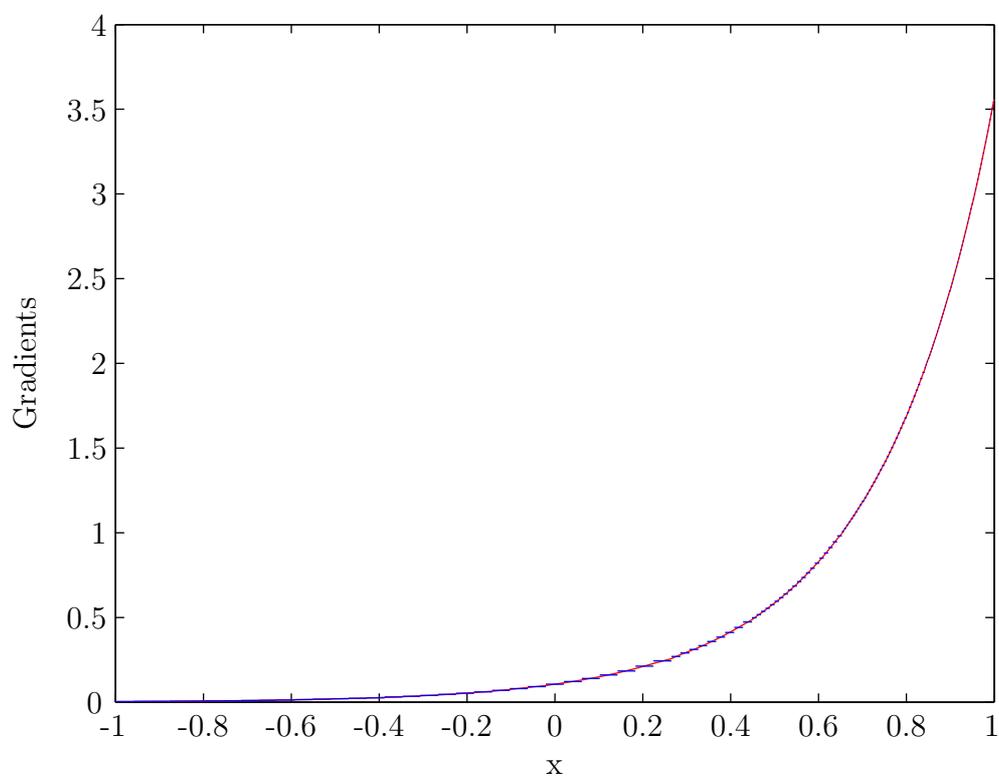


figure B.6 – $\hat{\sigma}$ en bleu et σ^\dagger en rouge après convergence du raffinement



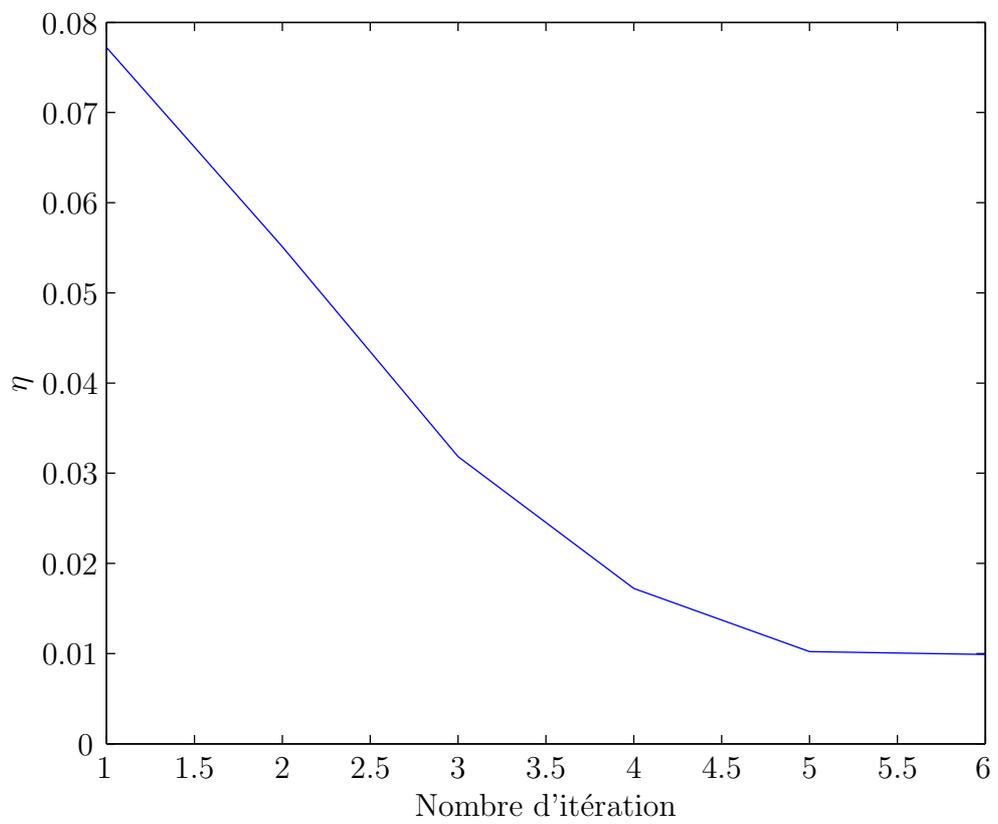


figure B.7 – η en fonction du nombre d'itération de raffinement



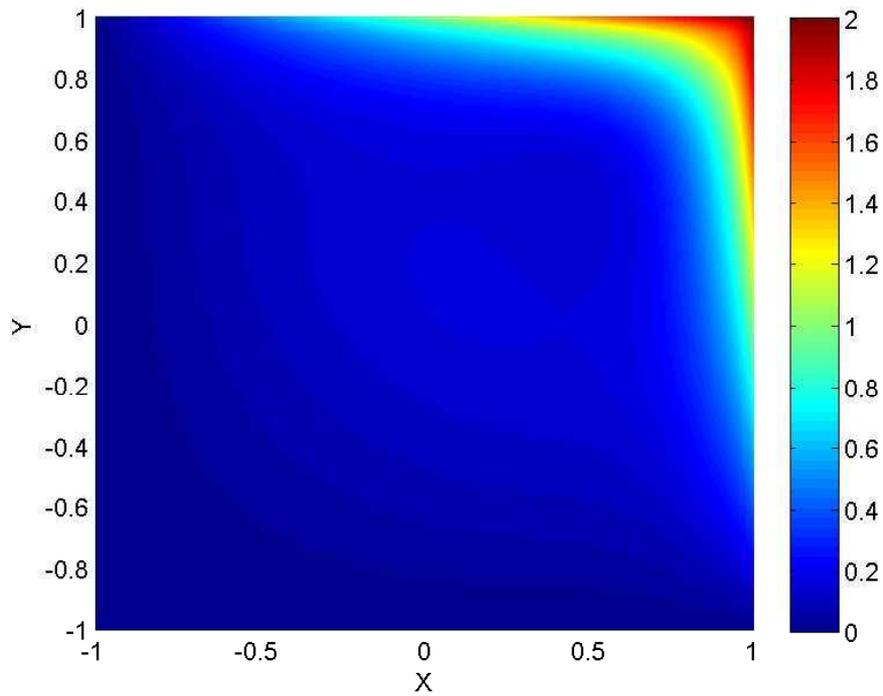


figure B.8 – La solution \mathcal{U} avec une convection suivant x et y

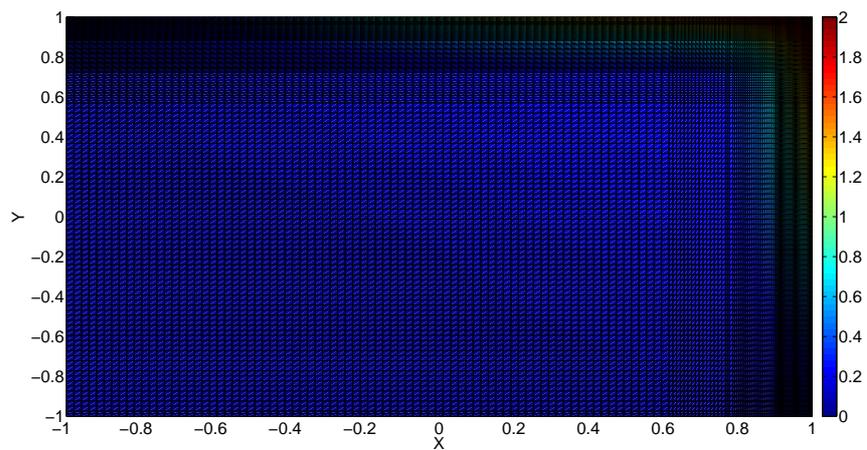


figure B.9 – \mathcal{U} après raffinement en x et en y

Résumé : La modélisation des procédés de mise en forme de composites est confrontée à de nombreux verrous scientifiques malgré les avancées récentes en matière de modélisation mécanique, analyse numérique, stratégies de discrétisation et capacité de calcul. En effet, la mise en forme de composites est confrontée à la nécessité de la prise en compte des comportements non-linéaires anisotropes et fortement couplés, définis dans des géométries très complexes. De plus, l'optimisation des procédés ainsi que l'identification par calcul inverse nécessite de multiples résolutions du problème direct. Dans ce contexte les techniques de réduction de modèles offrent de nouvelles possibilités, permettant d'accélérer les calculs de quelques ordres de magnitude, et même de résoudre des modèles jamais résolus jusqu'à présent.

La "Proper Generalized Decomposition" ou PGD est une des trois grandes familles des méthodes de réduction de modèles, susceptible de constituer un changement de paradigme en mécanique numérique. En effet, la PGD permet de résoudre des problèmes multidimensionnels résultants de l'introduction de paramètres physiques ou de conformation tout en évitant la malédiction de la dimensionnalité. Dans ce travail, on utilise la PGD pour adresser la solution de problèmes thermiques rencontrés lors de la mise en forme des composites. De plus, une approche de calcul "off-line/on-line" pour l'optimisation et le contrôle en temps réel est proposée. En effet, la PGD est utilisée pour calculer "off-line" des solutions paramétriques, exploitées ensuite "on-line" sur des plateformes de calcul légères (smartphones ou tablettes).

Mots-clés : PGD; Matériaux composites; Modèles paramétrés; Calcul Off-line/On-line; Optimisation; Contrôle.

Abstract : The modeling of composites manufacturing processes remains today a scientific challenging issue despite the impressive progress reached in mechanical modeling, numerical analysis, discretization techniques and computer science during the last decade. Indeed, composite manufacturing involves highly non-linear anisotropic behaviors and strongly coupled multiphysics defined in complex geometries. Moreover, optimization, inverse analysis and process control require the solutions of many direct problems, as fast and accurate as possible. In this context, reduced order models constitute an appealing simulation choice, accelerating the computations of several orders of magnitude, and even enabling the solution of models never solved until now.

The "Proper Generalized Decomposition" or PGD is one of the three main families of reduced order model techniques. PGD represents a new paradigm in computational mechanics. PGD can address the solution of multidimensional problems involving space, time and parameters as extra-coordinates, while circumventing the curse of dimensionality thanks to the separated representations that it involves. In this work we use the PGD to solve thermal problems encountered in composite forming processes. Moreover, an "offline/online" computational technique is proposed in order to optimize and control processes in real time. In fact, the PGD is used to compute parametric solutions "offline", while optimization techniques are performed "online" in order to identify optimal material, process or geometrical parameters. Furthermore, "Online" calculations can be performed on light computing devices like smartphones or tablets.

Keywords : PGD; Composite Materials; Parametric Models; Offline/Online Computation; Optimization; Control.

Discipline : Sciences de l'Ingénieur