



Quelques méthodes de réduction de modèle et calcul intensif pour les simulations numériques de grande échelle

Mémoire d'Habilitation à Diriger des Recherches par

FABIEN CASENAVE

Soutenance prévue le 8 Juillet 2020 devant le jury :

Rapporteurs

Florian De Vuyst
Mario Ohlberger
Gianluigi Rozza

professeur à l'Université de Technologie de Compiègne
professeur à l'Université de Münster
professeur à l'École Internationale Supérieure d'Études Avancées

Examineurs

Virginie Ehrlacher
Yvon Maday
Isabelle Panet
Christian Rey
David Ryckelynck

maître de conférences à l'École des Ponts ParisTech
professeur à l'Université Pierre-et-Marie-Curie
directeur de recherche à l'Institut de Physique du Globe de Paris
professeur à l'École Normale Supérieure Paris-Saclay et à Safran
professeur à l'École des Mines ParisTech

Université Pierre-et-Marie-Curie
Faculté de Mathématiques (UFR 929)

Table des matières

Activités scientifiques et académiques	i
Introduction	v
1 Simulation numérique de la propagation acoustique avec écoulement en milieu non borné	1
2 Simulation efficace du potentiel de gravitation généré par une source volumique quelconque	7
2.1 Un algorithme de maillage adapté au calcul des corrections topographiques	8
2.2 Couplage BEM/FEM pour les équations de gravitation	11
2.3 Calcul rapide de champs de gravitations par Fast Multipole Method	14
2.4 Une méthode de sommation rapide et adaptative pour noyaux invariants par translation	20
3 Travaux sur les méthodes de réduction de modèle physique	27
3.1 Application de la méthode des bases réduites à l'étude d'un problème aérothermique multi-échelles	27
3.2 Une méthode POD-EIM pour les problème de thermique transitoire non-linéaire avec une convection importante	29
3.3 Variantes de l'Empirical Interpolation Method : formulation symétrique, choix des normes et extension rectangulaire	38
3.4 Évaluation précise et efficace de la borne d'erreur a posteriori dans la méthode des bases réduites	41
3.5 Mise en œuvre des méthodes de réduction de modèle : non-intrusivité, problèmes non-linéaires de grande taille, indication d'erreur, variabilités non-paramétrées	45
3.5.1 Nonintrusive Empirical Interpolation Method : faible intrusivité pour la méthode des bases réduites	45
3.5.2 Approximation non-intrusive des limites d'algorithmes de puissances de matrices : application aux inverses et log-déterminants de matrices et à la réduction non-intrusive de modèles non-linéaires	49
3.5.3 Un environnement non-intrusif et parallèle en mémoire distribuée pour la réduction de modèle non-linéaire en mécanique des structures : application à l'extrapolation cyclique de matériaux élastoviscoplastiques	60
3.5.4 Un modèle réduit non-intrusif adaptatif basé sur un indicateur d'erreur, pour la mécanique des structures non-linéaire avec variabilités non-paramétrées	68
3.5.5 Un modèle réduit non-intrusif pour un problème de thermique transitoire non-linéaire avec variations non-paramétrées	75
3.6 POD-Galerkine stable pour l'équation de Navier-Stokes incompressible en régime turbulent avec variations géométriques non-paramétrées.	78
4 Perspectives	81
4.1 Amélioration des méthodes de réduction d'ordre de modèle	81
4.2 Implémentation générique et non-intrusive de la réduction d'ordre de modèle	81
4.3 Intelligence artificielle au service de la réduction d'ordre de modèle	

Activités scientifiques et académiques

Mis à jour le 18 Mai 2020.

Publications dans revues avec comité de relecture

Prépublications

- Pub.1 N. Akkari, F.C. and D. Ryckelynck, *A novel gappy reduced order method to capture non-parameterized geometrical variation in fluid dynamics problems*
- Pub.2 F.C., *A Fast Summation Method for translation invariant kernels*

Publications

- Pub.3 T. Daniel, F.C., N. Akkari and D. Ryckelynck, *Model order reduction assisted by deep neural networks (ROM-net)*, **Adv. Model. Simul. Eng. Sci.** 7(16), (2020)
- Pub.4 F.C., A. Gariah, C. Rey and F. Feyel, *A nonintrusive reduced order model for nonlinear transient thermal problems with nonparametrized variability*, **Adv. Model. Simul. Eng. Sci.**, 7(22) (2020)
- Pub.5 F.C., N. Akkari and D. Ryckelynck, *Reduced Order Modeling assisted by Convolutional Neural Network for thermal problems with nonparametrized geometrical variability*, accepted in **Adv. Intell. Syst.**
- Pub.6 N. Akkari, F.C., M.-E. Perrin and D. Ryckelynck, *Deep Convolutional Generative Adversarial Networks applied to 2D incompressible and unsteady fluid flows*, accepted in **Adv. Intell. Syst.**
- Pub.7 F.C., N. Akkari, F. Bordeu, C. Rey and D. Ryckelynck, *A Nonintrusive Distributed Reduced Order Modeling Framework for nonlinear structural mechanics – application to elastoviscoplastic computations*, accepted for publication in **Int. J. Numer. Meth. Eng.**
- Pub.8 N. Akkari, F.C. and V. Moureau, *Time Stable Reduced Order Modeling by an Enhanced Reduced Order Basis of the Turbulent and Incompressible 3D Navier-Stokes Equations*, **Math. Comput. Appl.** 24(2) (2019), 45 (19 p), **chosen for issue cover paper**
- Pub.9 F.C. and N. Akkari, *An error indicator-based adaptive reduced order model for nonlinear structural mechanics – application to high-pressure turbine blades*, **Math. Comput. Appl.** 24(2) (2019), 41 (28 p)
- Pub.10 F.C., N. Akkari, A. Charles and C. Rey, *Nonintrusive approximation of parametrized limits of matrix power algorithms – application to matrix inverses and log-determinants*, **M2AN Math. Model. Numer. Anal.** 53 (2019), 219-248
- Pub.11 A. Charles, F.C. and C. Glocker, *A catching-up algorithm for multibody dynamics with impacts and dry friction*, **Comput. Methods. Appl. Mech. Eng.** 334(1) (2018), 208-237
- Pub.12 F.C., A. Ern, and T. Lelièvre, *Variants of the Empirical Interpolation Method: Symmetric formulation, choice of norms and rectangular extension*, **Appl. Math. Lett.** 56 (2016), 23–28
- Pub.13 M. Van Camp, O. De Viron, G. Pajot-Métivier, F.C., A. Watlet, A. Dassargues and M. Vanclooster, *Direct measurement of evapotranspiration from a forest using a superconducting gravimeters*, **Geophys. Res. Lett.** 43(19) (2016), 10225–10231
- Pub.14 F.C., L. Métivier, Pajot-Métivier G. and Panet I., *Fast computation of general forward gravitation problems*, **J. Geodesy.** 90(7) (2016), 655–675
- Pub.15 F.C., A. Ern and T. Lelièvre, *A nonintrusive Reduced Basis Method applied to aeroacoustic simulations*, **Adv. Comput. Math.** 41(5) (2015), 961–986
- Pub.16 N. Balin, F.C., F. Dubois, E. Duceau, S. Duprey and I. Terrasse, *Boundary element and finite element coupling for aeroacoustics simulations*, **J. Comput. Phys.** 294(1) (2015), 274–296

-
- Pub.17 F.C., A. Ern and G. Sylvand, *Coupled BEM-FEM for the convected Helmholtz equation with non-uniform flow in a bounded domain*, **J. Comput. Phys.** 257-A (2014), 627-644
- Pub.18 F.C., A. Ern and T. Lelièvre, *Accurate and online-efficient evaluation of the a posteriori error bound in the reduced basis method*, **M2AN Math. Model. Numer. Anal.** 48 (2014), 207-229
- Pub.19 F.C., *Accurate a posteriori error evaluation in the reduced basis method*, **C. R. Math. Acad. Sci.** Paris 350 (2012), no. 9-10, 539-542
- Pub.20 F.C., M. Ghattassi and R. Joubaud, *A multiscale problem in thermal science*, CEMRACS'11 : Multiscale coupling of complex models in scientific computing, **ESAIM Proc.**, EDP Sci., Les Ulis 38 (2012), 202-219

Rapports de recherche

- Rap.1 F.C., *Nonintrusive nonlinear reduced order modeling by approximation of inverse tangent operators*, 2019
- Rap.2 F.C., *Stability of empirical interpolation methods for the reduction of nonlinear transient thermal problems with translation effects – application to single crystal casting processes*, 2019
- Rap.3 F.C., *Machine learning based model order reduction of nonlinear structural mechanics problems by meta modelisation of the behavior law solver*, 2019
- Rap.4 F.C., G. Pajot-Métivier, L. Métivier and I. Panet, *A volumic mesh for computing global Earth topographic corrections*, 2015
- Rap.5 F.C., A. Ern, T. Lelièvre and G. Sylvand, *A nonintrusive EIM to approximate linear systems with nonlinear parameter dependence*, 2013
- Rap.6 F.C., *Étude d'un modèle d'incertitudes non paramétriques*, 2012

Présentations

Sur invitation

- Exp.1 *Physical ROM : how to accomodate nonparametrized variability, nonintrusivity, performance and error indication for large scale industrial applications ?*, USNCCM15, 28/07-01/08/2019, Austin, Texas, USA
- Exp.2 *An error indicator-based adaptative reduced order model for nonlinear structural mechanics*, ADMOS2019, 27-29/05/2019, El Campello (Alicante), Spain
- Exp.3 *Towards nonintrusive large scale reduced order modeling of nonlinear structural mechanics at Safran*, Club Z-set users' day, 19/03/2019, ONERA Châtillon, France
- Exp.4 *Nonintrusive large scale ElastoViscoPlastic cycle extrapolation using a POD-Hyperreduced Order Model*, EUROMECH 597, 28-31/08/2018, Bad Herrenalb, Allemagne
- Exp.5 *Towards nonintrusive large scale reduced order modeling of nonlinear structural mechanics at Safran*, WCCM 2018, 22-27/07/2018, New-York, États-Unis d'Amérique
- Exp.6 *Some model reduction developments at Safran*, 2nd CSMA J. workshop, 14/03/2018, Gif/Yvette, France
- Exp.7 *Fast direct computation of general gravitation problems using Scalfnm*, séminaire de l'équipe HiePACS, INIRA Bordeaux – Sud-Ouest, 10/06/2015, Talence, France
- Exp.8 *The Fast Multipole Method and application to the direct computation of general gravitation problems*, séminaire de calcul scientifique du CERMICS, 16/04/2015, Champs-sur-Marne, France
- Exp.9 *Fast direct computation of gravitational topographic correction by accelerated numerical integration over tetrahedrons*, séminaire du laboratoire Géosciences Montpellier, 15/01/2015, Montpellier, France
- Exp.10 *Couplage éléments de frontière / éléments finis, réduction de modèle et méthodes de sommation rapide*, séminaire du Laboratoire de Recherche en Géodésie (LAREG), IGN, 05/06/2014, Paris, France
- Exp.11 *The Reduced Basis method for aeroacoustic simulations solved by integral equations*, Workshop numerical methods for high-dimensional problems, 16/04/2014, Champs-sur-Marne, France
- Exp.12 *Reduced order methods applied to aeroacoustic simulations solved by integral equations*, séminaire des doctorants du CERMICS, 29/11/2013, Champs-sur-Marne, France

-
- Exp.13 *Reduced order methods applied to aeroacoustic simulations solved by integral equations*, séminaire du laboratoire Jacques-Louis Lions, 04/10/2013, Paris, France
- Exp.14 *On the reduced basis method*, séminaire du laboratoire Analyse et Mathématiques Appliquées, 19/06/2012, Marne-La-Vallée, France
- Exp.15 *On the reduced basis method*, séminaire des doctorants du CERMICS, 07/06/2012, Champs-sur-Marne
- Exp.16 *Advanced methods in computational acoustics*, 6th Noise & Vibration Workshop, 09-10/11/2011, Toulouse
- Exp.17 *Domaine decomposition for convective acoustics in 3D*, journée de la chaire "Mathematical Modelling and Numerical Simulation" (patenariat Ecole Polytechnique et EADS), 13/10/2010, Suresnes, France

Sur candidature

- Exp.18 *Some model reduction developments at Safran*, Coupled Problems 2017, 12-14/06/2017, Rhodes, Grèce
- Exp.19 *A POD-EIM model reduction for a nonlinear thermal transient problem*, WCCM 2016, 24-29/07/2016, Seoul, Corée du Sud
- Exp.20 *Fast computation of general direct gravitation problems*, 26th IUGG General Assembly, 22/6-2/7/2015, Prague, République Tchèque
- Exp.21 *Reduced basis strategies for aeroacoustics simulations*, WAVES'13, 3-7/06/2013, Tunis, Tunisie
- Exp.22 *3D coupled resolution of acoustics waves between a potential flow and a uniform flow*, WAVES'11, 25-29/07/2011, Vancouver, Canada

Actes de conférence et poster

- Abs.1 F.C., A. Ern, T. Lelièvre et G. Sylvand, *Reduced basis strategies for aeroacoustics simulations*, 11th International Conference on Mathematical and Numerical Aspects of Waves (WAVES'13), 03-07/06/2013, Tunis, Tunisia
- Abs.2 N. Balin, F.C. et G. Sylvand, *3D coupled resolution of acoustics waves between a potential flow and a uniform flow*, 10th International Conference on Mathematical and Numerical Aspects of Waves (WAVES'11), 25-29/07/2011, Vancouver, Canada
- Pos.1 F.C., *Acurate and efficient evaluation of the a posteriori estimator in the reduced basis method*, 2nd International Workshop on Model Reduction for Parametrized Systems, 02-05/10/2012, Gunzburg, Allemagne

Enseignements

- Ens.1 École des Ponts ParisTech, *Calcul scientifique et Optimisation* (avec Alexandre Ern), 2011-2012
- Ens.2 École des Ponts ParisTech, *Calcul scientifique et Optimisation* (avec Alexandre Ern), 2012-2013
- Ens.3 École des Mines ParisTech, *Introduction au Calcul Scientifique* (avec Gabriel Stoltz), 2012-2013
- Ens.4 École des Ponts ParisTech, *Analyse et Calcul scientifique* (avec Gabriel Stoltz), 2017-2018

Animation de la recherche

- Ani.1 (à venir) Co-organisation (avec N. Akkari) du mini-symposium *Physical reduced order modelling and machine learning for engineering applications : get the best from both worlds*, à la conférence WCCM2020, 19-24/07/2020, Paris, France
- Ani.2 Co-organisation (avec N. Akkari et D. Ryckelynck) du mini-symposium *Data assimilation in Model Order Reduction techniques for Computational Mechanics*, à la conférence USNCCM15, 28/07-01/08/2019, Austin, Texas, États-Unis d'Amérique
- Ani.3 Co-organisation (avec N. Akkari et D. Ryckelynck) du mini-symposium *Industrial challenges : machine learning, large scale problems, nonlinear behaviors* à la conférence ADMOS 2019, 27-29/05/2019, El Campello (Alicante), Espagne

Encadrements scientifiques

Thèses

- PhD.1 (**candidat à trouver**) *Model order reduction for contact problems in structural dynamics : application to the dimensioning of aeronautical cabin components*; F.C., N. Akkari, D. Ryckelynck 2020–2023
- PhD.2 T. Daniel, *Apprentissage artificiel de modèles réduits physiques pour la prévision rapide de la durée de vie d'aubes de turbine avec quantification d'incertitudes*; F.C, N. Akkari, D. Ryckelynck, 2018–2021

Stages de Master2 Recherche

- M2R.1 M.-E. Perrin, *Generative adversarial network for turbulent fluid mechanics*, 2019 (encadré avec N. Akkari)
- M2R.2 A. El-Mizabi, *Application de la méthode de réduction POD-ECM à des cas elastoviscoplastiques dans un contexte HPC*, 2018 (encadrée avec A. Parret-Fréaud)
- M2R.3 J.-B. Berry, *Comparaison de méthodes de réduction de modèles appliquées à des calculs thermiques et calculs élastoviscoplastiques*, 2017
- M2R.4 M. Kogri, *Modélisation et calcul de durée de vie des aubes de turbines : vers des calculs plus rapides*, 2017 (encadré avec T. Rose)
- M2R.5 A.-A. Homman, *Étude et implémentation des algorithmes Pure Greedy Algorithm et Orthogonal Greedy Algorithm*, 2012 (encadré avec T. Lelièvre)

Projets de recherche étudiants

- Pro.1 Etude d'une méthode de sommation rapide et application à un calcul de correction de topographie globale du potentiel de gravitation (encadrement de 2 étudiants), 2014-2015
- Pro.2 Analyse numérique et optimisation : Méthode des bases réduites (avec Alexandre Ern), 2012-2013

Activités de rapporteur

1) ESAIM : Proceedings, 2) CR MECANIQUE, 3) SIAM Journal on Scientific Computing, 4) Advances in Computational Mathematics, 5) Journal of Scientific Computing, 6) Computers and Mathematics with Applications, 7) Mathematical and Computational Applications, 8) Journal of Computational Physics

	doctorat (2010-2013)	IGN (2013-2015)	Safran Tech (2015-présent)
Publications et rapports	Pub.15, Pub.16, Pub.17, Pub.18, Pub.19, Pub.20, Rap.5, Rap.6	Pub.2, Pub.12, Pub.13, Pub.14, Rap.4	Pub.1, Pub.3, Pub.4, Pub.5, Pub.6, Pub.7, Pub.8, Pub.9, Pub.10, Pub.11, Rap.1, Rap.2, Rap.3
Présentations	Exp.12, Exp.13, Exp.14, Exp.15, Exp.16, Exp.17, Exp.21, Exp.22, Pos.1	Exp.7, Exp.8, Exp.9, Exp.10, Exp.11, Exp.20	Exp.1, Exp.2, Exp.3, Exp.4, Exp.5, Exp.6, Exp.18, Exp.19
Encadrement scientifique	M2R.5, Pro.2	Pro.1	M2R.1, M2R.2, M2R.3, M2R.4, PhD.1, PhD.2
Enseignement, animation	Ens.1, Ens.2, Ens.3	–	Ens.4, Ani.1, Ani.2, Ani.3
Développements info.	bases réduites (python) Actipole (C, fortran)	topo mesher (python) ScalFMM (C++) Actipole (C, fortran)	Zset (C++) niROM (python) BasicTools (python)

Introduction

Dans ce manuscrit, nous présentons de façon didactique et synthétique les travaux et résultats de recherche dont l'auteur a la contribution principale, hormis la courte Section 3.6. En particulier, toutes les simulations numériques présentées ont été réalisées par l'auteur, hormis ceux de cette introduction et des Figures 1.5, 1.6 et 2.17. Les références vers les publications contenant les détails, commentaires et applications numériques non inclus dans ce manuscrit sont données dans le texte.

Dans cette introduction, nous commençons par donner les motivations applicatives des travaux, puis nous donnons quelques éléments sur les méthodes numériques qui seront utilisées de façon intensive dans le reste du manuscrit.

Motivations applicatives

Nous présentons les motivations des travaux de recherche par poste occupé.

Applications pour EADS-Innovation Works

La réduction du bruit en aéronautique constitue un enjeu industriel de première importance. En effet, face à l'augmentation substantielle du trafic aérien, les normes de certification acoustique sont fortement renforcées afin de protéger les riverains des nuisances sonores essentiellement occasionnées par les avions aux abords des aéroports. Pour pouvoir intégrer ces normes dès les phases d'avant-projet, il est nécessaire aux manufacturiers comme Airbus de se doter d'outils numériques efficaces en acoustique. Résoudre un problème acoustique direct n'est pas suffisant en pratique. Les choix industriels relèvent plutôt de questions comme : "comment optimiser le revêtement de l'intérieur d'un turboréacteur pour minimiser le bruit généré par l'avion dans certaines directions, ou quelle est la probabilité de respecter une norme environnementale de bruit si la valeur de la fréquence d'une source acoustique est mal connue?". Ces études d'optimisation et de propagation d'incertitudes requièrent, en général, l'évaluation intensive d'une fonction objectif, impliquant la résolution d'un système d'équations aux dérivées partielles paramétriques de très grande taille, pour une certaine valeur des paramètres.

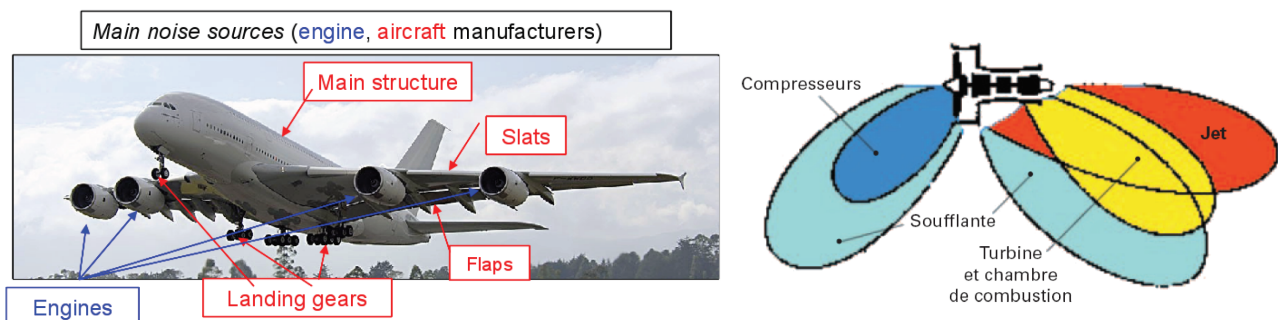


FIGURE 1 – Origine du bruit généré par un avion (gauche : image fournie par EADS, droite : image tirée de [76]).

Applications pour l'IGN

La gravité est l'accélération subie par un corps, due aux forces combinées de la Terre, de la Lune, du Soleil et des planètes environnantes. Pour un corps au repos sur la surface de la Terre, il faut aussi considérer les effets

centrifuges dus à la rotation de la Terre.

En géodésie et géophysique, les études ont recours à des données de gravitations issues de mesures expérimentales. En particulier, les missions spatiales scientifiques de gravimétrie spatiale GOCE [1] et GRACE [2] fournissent des données détaillées de la gravité terrestre. Les instruments contemporains mesurent les effets gravitationnels avec une précision de $10^{-11}g$ [110]. Les champs de gravitation sont étudiés par les géophysiciens et les compagnies d'exploitation minières et de gaz pour explorer les ressources souterraines [24]. Pour pouvoir étudier les structures souterraines à partir de ces mesures, le signal doit être nettoyé des effets mentionnés ci-dessus. La correction de la Lune, du Soleil et des autres planètes peut être réalisée avec une précision de $10^{-11}g$ grâce à la mécanique céleste [63, 115]. Les effets centrifuges et atmosphériques peuvent être également corrigés précisément à partir des données de rotation de la Terre et pression atmosphérique de surface. Enfin, les derniers effets à corriger sur le signal correspondent aux structures topographiques connues : après avoir modélisé les masses situées entre la surface de la Terre et l'ellipsoïde de référence, il faut calculer le potentiel de gravitation généré par ces masses.

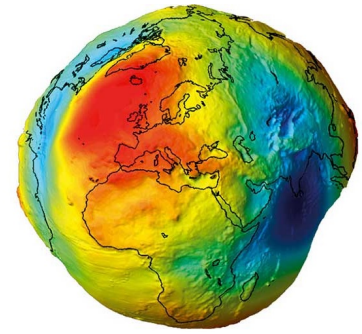


FIGURE 2 – Illustration d'un géoïde [42] : une surface équipotentielle de référence du champ de pesanteur terrestre

Applications pour Safran

Une course à la performance a lieu dans l'industrie aéronautique. Certaines pièces du moteur sont poussées à leur limite de résistance pour améliorer le plus possible le rendement propulsif. En particulier, les aubes de turbine haute pression, qui sont situées immédiatement après la chambre de combustion, subissent des chargements thermiques très importants, ou le compresseur haute pression, voir Figure 3. Des efforts sont menés pour protéger ces pièces de ruine par fluage thermique et fatigue à grand nombre de cycles [41, 87], en augmentant leur résistance par l'utilisation de canaux de refroidissement internes [10, 112], barrières thermiques [104] et superalliages avancés [31, 47]. La fabrication de ces aubes se fait par moulage et solidification dirigée, ce qui permet la sélection d'un cristal unique, améliorant encore la résistance et les propriétés mécaniques du matériau [73, 111].

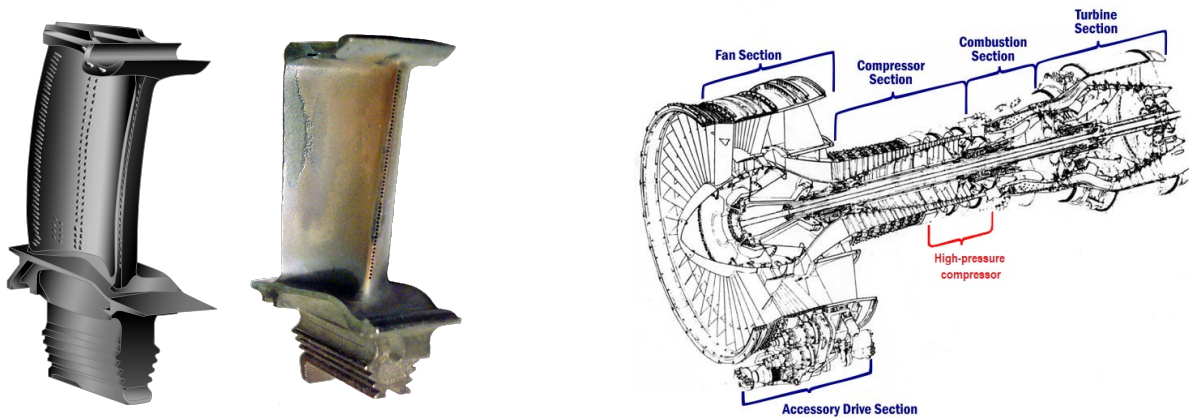


FIGURE 3 – Aube de turbine haute pression ; gauche : schéma [3] ; milieu : photographie d'un aube de turbine usagée d'un moteur Turbo-Union RR 199 [4] ; droite : coupe d'un moteur CF6-6 engine avec mise en évidence du compresseur haute pression [117].

Le superalliage liquide est contenu dans un moule à l'intérieur du four et est translaté hors du four par le bas, avec une vitesse contrôlée. Ainsi, le refroidissement commence par le bas de l'aube, où la croissance du cristal est contrôlé à partir d'une dendrite unique. Les paramètres du four doivent être optimisés pour la sélection d'un cristal unique : un angle élevé du front de solidification peut conduire à des points froids et l'apparition

de grains secondaires de solidification et, combiné avec un refroidissement trop lent, à l'apparition de freckles, voir Figure 4 pour des illustrations de ces défauts et la configuration du four.

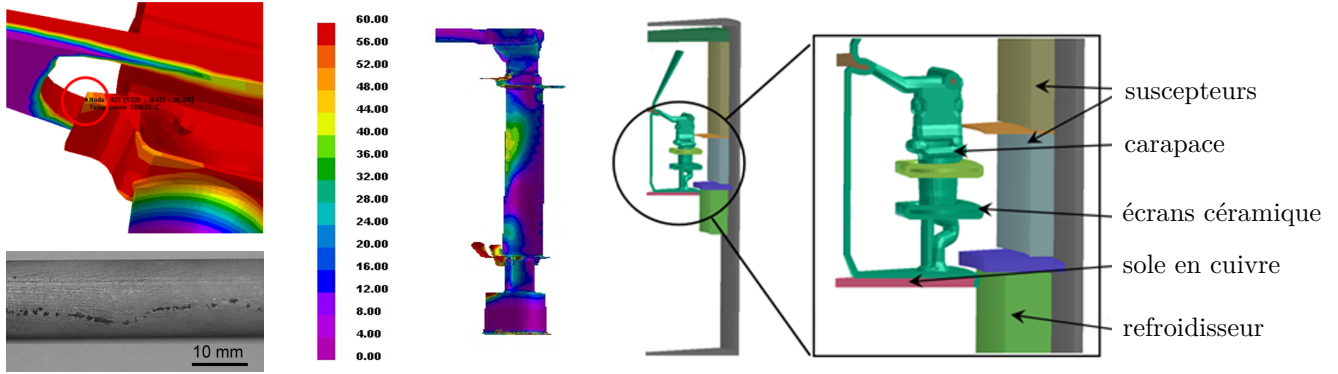


FIGURE 4 – Illustration de défauts. A gauche : haut : point froid ; bas : freckles ; au milieu : champ de température ($^{\circ}\text{C}$) conduisant à un front de solidification non-horizontale ; droite : configuration du four.

Les besoins calculatoires autour des aubes de turbine haute pression sont doubles ; (i) pour la fabrication : la recherche de paramètres optimaux du four (température du four, la vitesse de translation, la présence de masselottes et la position des écrans thermiques en céramique) conduit à la résolution numérique d'un grand nombre de simulations et (ii) la prédiction de la durée de vie de ces pièces est une tâche calculatoire très coûteuse : les maillages ont un nombre important d'éléments pour prendre en compte des petites structures, comme les canaux de refroidissement, les lois de comportement sont fortement non-linéaires et comportent un nombre important de variables d'état, et un grand nombre de cycles doit être simulé. En effet, la ruine peut venir d'effets structurels locaux, dont l'évolution ne peut pas être prédite sans simuler potentiellement des centaines de cycles. Pour les compresseurs haute pression, un enjeu important est d'éviter le pompage en service.

Définitions, méthodes principales et premiers éléments bibliographiques

Sans prétention d'exhaustivité, nous regroupons dans cette section quelques définitions, présentations et méthodes et éléments bibliographiques pertinents pour le reste du document.

Réduction de modèle – ROM

Soit un espace paramétrique \mathcal{P} , nous considérons le problème suivant :

$$E_{\mu} : \mathcal{P} \ni \mu \xrightarrow{E_{\mu}} u_{\mu}, \quad (1)$$

où μ est le paramètre et u_{μ} la solution.

La notion de réduction de modèle peut être interprétée de façon large, en regroupant toute méthode ou stratégie visant à accélérer la résolution (1). Par exemple, exploiter les symétries éventuelles du problème pour en diminuer la taille peut être considéré comme une réduction de modèle. Dans la communauté de l'apprentissage statistique, une problématique d'intérêt consiste à construire des formules de régression pour la fonction $\mu \mapsto Q(u_{\mu})$, où $Q(u_{\mu})$ est une quantité d'intérêt sur la solution de (1) (cette quantité pouvant être la solution elle-même). Ces régressions, parfois appelées méta-modèles, peuvent être considérées également comme une réduction de modèle. Considérons maintenant que le problème (1) est exprimé sous une forme faible et qu'il est résolu numériquement par une méthode de (Petrov-)Galerkine (une méthode de résolution numérique regroupant une très large classe de problèmes physiques rencontrés dans l'industrie) : trouver $u_{\mu} \in \mathcal{V}_h$ tel que

$$E_{\mu} : a(u_{\mu}, v) = b(v), \quad \forall v \in \mathcal{V}_h, \quad (2)$$

où a est une forme bilinéaire symétrique coercive sur $\mathcal{V} \times \mathcal{V}$ et b une forme linéaire sur \mathcal{V} , où \mathcal{V} est un espace de Hilbert et $\mathcal{V}_h \subset \mathcal{V}$ est une approximation conforme de \mathcal{V} de dimension finie. L'équation (2) est la forme la plus simple possible, nous traiterons dans ce document des problèmes linéaires et non-linéaires plus complexes. Lorsque la méthode des éléments finis est utilisée, les fonctions-test v dans (2) sont remplacées par les fonctions

de base éléments finis $\{\varphi_i\}_{1 \leq i \leq N}$, N pouvant être de l'ordre de 10^7 dans certains problèmes industriels, ce qui conduit à un système d'équations linéaires de taille N . Une classe de méthodes de réduction de modèle dites par projection consistent à remplacer dans la méthode de Galerkin, la base éléments finie $\{\varphi_i\}_{1 \leq i \leq N}$, bien adaptée pour représenter n'importe quelle fonction de \mathcal{V}_h , par une base dite d'ordre réduit $\{\Psi_i\}_{1 \leq i \leq n}$, $n \ll N$, bien adaptée pour représenter les solutions de (2). Le problème réduit s'écrit

$$\hat{E}_\mu : \quad a(\hat{u}_\mu, \hat{v}) = b(\hat{v}), \quad \forall \hat{v} \in \hat{\mathcal{V}}_h, \quad (3)$$

où $\hat{\mathcal{V}}_h = \text{Vect}(\Psi_i)_{1 \leq i \leq n}$ (en général inclus dans \mathcal{V}_h). La construction de la base d'ordre réduit nécessite de résoudre dans un premier temps le problème (2) pour en extraire les composantes principales : nous parlons de réduction de modèle a posteriori. Dans la suite du document, nous appelons "réduction de modèle" la classe de méthode de réduction de modèle a posteriori par projection.

Nous pouvons maintenant introduire quelques premières notions. Le problème E_μ est appelé problème haute fidélité, ou problème de confiance. La précision du problème réduit \hat{E}_μ est évaluée en calculant l'erreur commise par rapport à la solution de référence u_μ du problème E_μ . Un modèle réduit est dit certifié lorsque que cette erreur est garantie plus faible qu'une erreur fixée à l'avance. La réduction de modèle se fait en deux étapes : une première étape dite "offline" d'apprentissage du modèle E_μ , où la solution u_μ est calculée pour certaines valeurs du paramètre μ et où la base d'ordre réduit $\{\Psi_i\}_{1 \leq i \leq n}$ est construite ; les solutions u_μ à notre disposition sont appelées snapshots. Dans cette étape, les opérations en complexité algorithmique dépendante de N sont autorisées. Une deuxième étape dite "online" correspond à l'étape d'exploitation du modèle réduit : le problème réduit \hat{E}_μ est construit et résolu pour potentiellement n'importe quelle valeur de $\mu \in \mathcal{P}$. Dans cette étape, toutes les opérations doivent être en complexité algorithmique indépendante de N : on dit alors que le modèle réduit est (online)-efficace. Sauf mention du contraire, nous définissons speedup comme le ratio entre le temps de résolution du problème haute-fidélité E_μ et celui du problème réduit \hat{E}_μ . En fonction de la dépendance paramétrique du problème E_μ , des étapes d'approximation supplémentaires peuvent être nécessaires, nous renvoyons à la Section 3.5.3 et à la Publication Pub.7 pour une discussion détaillée. En général, la réductibilité du problème E_μ est définie par la capacité de la méthode de réduction considérée à construire une base d'ordre réduit de cardinal suffisamment petit. De façon plus pragmatique, nous considérons un modèle comme réductible si les étapes d'approximation et d'exploitation complète (c'est-à-dire l'ensemble des résolutions des problèmes réduits \hat{E}_μ pour toutes les valeurs paramétriques μ requises) seront résolues en temps de retour significativement plus court que l'exploitation correspondante du modèle haute fidélité, et avec le niveau de précision requis. Enfin, nous introduisons la notion d'intrusivité : cette notion peut varier d'un auteur à l'autre, ici, nous considérons un modèle réduit non-intrusif s'il peut être construit et exécuté tel que le problème haute-fidélité E_μ est résolu avec un code commercial, c'est-à-dire sans possibilité de modifier le code source associé. Les enjeux de certification, d'intrusivité et d'efficacité sont centraux aux travaux en réduction de modèle présentés dans ce document, avec comme souci de contenir les coûts de la phase offline à des niveaux raisonnables (et non pas de considérer le budget illimité dans cette phase), comme nous l'illustrerons sur des applications industrielles.

Comme méthode de réduction de modèle a posteriori par projection, nous pouvons citer la Proper Orthogonal Decomposition (POD) [105] et la méthode des bases réduites (BR) [80]. La différence entre les deux méthodes réside dans le choix des valeurs paramétriques μ pour lesquelles les snapshots sont générés : avec la POD, les choix sont en général heuristiques et a priori, alors qu'avec les BR, les valeurs μ sont choisies de façon gloutonne de sorte à maximiser un estimateur d'erreur a posteriori. Par construction, les BR sont certifiées de façon efficace, mais cette contrainte d'estimation a posteriori rend difficile, pour le moment, son applicabilité à des problèmes industriels complexes. Nous mentionnons une classe de réduction de modèle a priori appelée Proper Generalized Decomposition (PGD) [37, 38], où tout l'effort est concentré dans la phase offline : la solution est exprimée comme fonction de l'espace et des paramètres, et une approximation est construite de façon itérative sous forme de somme de produits tensoriels de rang 1. La phase online correspond simplement à l'évaluation de cette fonction. Les caractéristiques recherchées pour la réduction de modèle dans l'industrie sont illustrées dans la Figure 5.

Empirical Interpolation Method – EIM

Considérons une fonction scalaire $K(x, y)$ définie sur $\mathcal{D}_x \times \mathcal{D}_y$ et supposée à valeurs réelles, où $\mathcal{D}_x \subset \mathbb{R}^{d_x}$ et $\mathcal{D}_y \subset \mathbb{R}^{d_y}$, $d_x, d_y \in \mathbb{N}^*$, sont bornées. Introduite dans [17], l'EIM est une procédure offline-online cherchant à déterminer une représentation en variables séparées approchée de la fonction K . Pendant l'étape offline, qui peut avoir un coût élevé (en terme de temps CPU), de l'information est acquise sur le comportement de cette

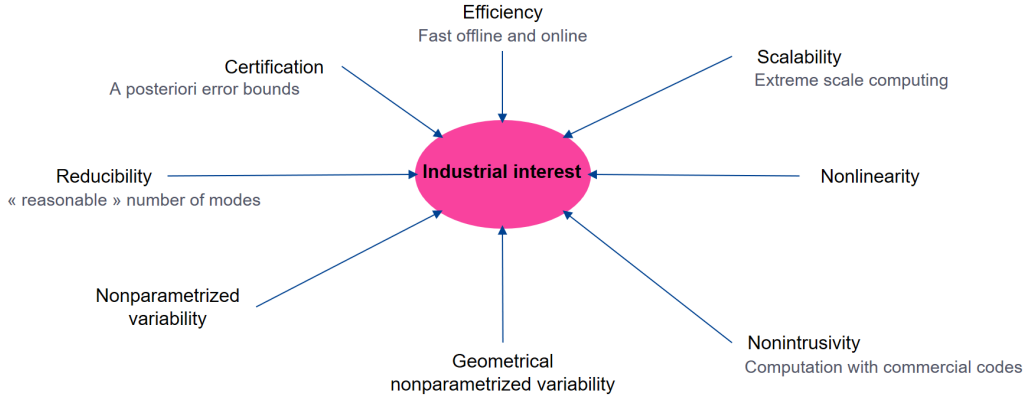


FIGURE 5 – Caractéristiques recherchées pour la réduction de modèle dans l’industrie.

fonction : c’est l’étape d’apprentissage. Pendant la phase online, la représentation en variables séparées peut être obtenue avec un coût faible : c’est l’étape d’exploitation. Soit $d \in \mathbb{N}^*$ tel que l’hypothèse suivante soit vérifiée

(H) La dimension de $\text{Span}(K(x, \cdot))$ est plus grande que d .
 $x \in \mathcal{D}_x$

L’approximation EIM s’écrit :

$$K(x, y) \approx (I_d K)(x, y) := \sum_{m=1}^d \lambda_m(x) q_m(y), \quad (4)$$

où $\lambda_m(x)$ est tel que

$$\sum_{m=1}^d B_{l,m} \lambda_m(x) = K(x, y_l), \quad \forall 1 \leq l \leq d. \quad (5)$$

Les fonctions $q_m(\cdot)$ et la matrice $B \in \mathbb{R}^{d \times d}$, qui est triangulaire avec des termes diagonaux égaux à 1, sont construites par l’Algorithme 1, où $\delta_d = \text{Id} - I_d$ et $\|\cdot\|_{\mathcal{D}_y}$ est la norme $L^\infty(\mathcal{D}_y)$. En pratique, le argmax apparaissant dans l’Algorithm 1 est cherché dans des sous-ensembles finis de \mathcal{D}_x et \mathcal{D}_y , notés respectivement $\mathcal{D}_{x,\text{trial}}$ et $\mathcal{D}_{y,\text{trial}}$, et appelés ensembles d’apprentissage. Remarquons que l’Algorithme 1 construit aussi l’ensemble de points $\{y_l\}_{1 \leq l \leq d}$ dans \mathcal{D}_y , qui est utilisé dans (5), et un ensemble de points $\{x_l\}_{1 \leq l \leq d}$ dans \mathcal{D}_x . Ce sont des points d’interpolation pour l’approximation de K , voir Propriété 0.1. Naturellement, l’approximation (4) peut être évaluée dans $(\mathcal{D}_x \times \mathcal{D}_y) \setminus (\mathcal{D}_{x,\text{trial}} \times \mathcal{D}_{y,\text{trial}})$. D’après (H), les fonctions $\{K(x_l, \cdot)\}_{1 \leq l \leq d}$ sont linéairement indépendantes (sinon, $(\delta_k K)(\mu_{k+1}, x_{k+1}) = 0$ pour certains k dans l’Algorithme 1).

Algorithm 1 Étape offline de EIM

1. Choisir $d > 1$
 2. Initialiser $k := 1$
 3. Calculer $x_1 := \underset{x \in \mathcal{D}_x}{\text{argmax}} \|K(x, \cdot)\|_{\mathcal{D}_y}$
 4. Calculer $y_1 := \underset{y \in \mathcal{D}_y}{\text{argmax}} |K(x_1, y)|$ [Premier point d’interpolation]
 5. Définir $q_1(\cdot) := \frac{K(x_1, \cdot)}{K(x_1, y_1)}$ [Première fonction de base]
 6. Définir $B_{1,1} := 1$ [Initialisation de la matrice B]
 7. **while** $k < d$ **do**
 8. Calculer $x_{k+1} := \underset{x \in \mathcal{D}_x}{\text{argmax}} \|(\delta_k K)(x, \cdot)\|_{\mathcal{D}_y}$
 9. Calculer $y_{k+1} := \underset{y \in \mathcal{D}_y}{\text{argmax}} |(\delta_k K)(x_{k+1}, y)|$ [$(k+1)$ -ème point d’interpolation]
 10. Définir $q_{k+1}(\cdot) := \frac{(\delta_k K)(x_{k+1}, \cdot)}{(\delta_k K)(x_{k+1}, y_{k+1})}$ [$(k+1)$ -ème fonction de base]
 11. Définir $B_{k+1,m} := q_m(y_{k+1})$, for all $1 \leq m \leq k+1$ [Incrémement de la matrice B]
 12. $k \leftarrow k+1$ [Incrémement de la taille de la décomposition]
 13. **end while**
-

Nous avons montré dans la Publication [Pub.15](#) que l'approximation (4) peut être écrite de la forme

$$(I_d K)(x, y) = \sum_{l,m=1}^d \Delta_{l,m} K(x, y_l) K(x_m, y), \quad (6)$$

où $\Delta = B^{-t} \Gamma^{-1}$, et où la matrice Γ est triangulaire supérieure et est construite récursivement dans la boucle k de l'Algorithme 1 de la façon suivante :

— $k = 1$:

$$\Gamma_{1,1} = K(x_1, y_1),$$

— $k \rightarrow k + 1$:

$$\begin{aligned} \Gamma_{k+1,k+1} &= (\delta_k K)(x_{k+1}, y_{k+1}), \\ \Gamma_{m,k+1} &= 0, & \forall 1 \leq m \leq k, \\ \Gamma_{k+1,m} &= \alpha_m, & \forall 1 \leq m \leq k, \end{aligned}$$

où le vecteur α est tel que $\sum_{m=1}^k B_{l,m} \alpha_m = K(x_{k+1}, y_l)$, pour tout $1 \leq l \leq k$.

D'après [83, Theorem 1], la représentation en variables séparées (4) construite par l'Algorithme 1 est bien définie. Nous rappelons le résultats suivant :

Propriété 0.1 (Propriété d'interpolation [83, Lemma 1]). *Pour tout $1 \leq m \leq d$,*

$$\begin{cases} (I_d K)(x, y_m) = K(x, y_m), & \text{pour tout } x \in \mathcal{D}_x, \\ (I_d K)(x_m, y) = K(x_m, y), & \text{pour tout } y \in \mathcal{D}_y. \end{cases}$$

Un aspect important de la procédure est que l'erreur faite par l'approximation EIM sur les ensembles d'apprentissage $\mathcal{D}_{x,\text{trial}} \times \mathcal{D}_{y,\text{trial}}$, notée ϵ_k et définie par $\epsilon_k := \max_{y \in \mathcal{D}_y} |(\delta_k K)(x_{k+1}, y)|$, est monitorée pendant la procédure gloutonne. En pratique, un critère d'arrêt est utilisé sur ϵ_k au lieu de fixer un nombre de points d'interpolation d a priori. Ainsi, en fonction de la régularité de la fonction K , l'EIM va sélectionner automatiquement le nombre de points d'interpolation pour certifier l'erreur sur $\mathcal{D}_{x,\text{trial}} \times \mathcal{D}_{y,\text{trial}}$. Notons que l'erreur n'est pas certifiée sur $\mathcal{D}_x \times \mathcal{D}_y$. Une variation appelée DEIM [35] (Discrete Empirical Interpolation Method) a été proposée pour le cas particulier où une des variables est discrète (par exemple, un numéro de composante d'un vecteur).

Fast Multipole Method – FMM

Introduite dans [59], la FMM est une méthode de sommation rapide adaptée à certains types de problèmes. Considérons un ensemble de points d'observations $\{x_i\}_{1 \leq i \leq M}$, un ensemble de points source $\{y_j\}_{1 \leq j \leq N}$ de \mathbb{R}^3 , auxquels sont attachées des données scalaires $\{\sigma_j\}_{1 \leq j \leq N}$ appelées potentiels, et un noyau $\mathbb{R}^3 \times \mathbb{R}^3 \ni (x, y) \mapsto K(x, y) \in \mathbb{R}$, pour lequel la dépendance en x et y n'est pas séparable (i.e. il n'existe pas de fonctions g_k et h_k , $1 \leq k \leq d$ telles que $K(x, y) = \sum_{k=1}^d g_k(x) h_k(y)$), et ayant possiblement une singularité en $x = y$. Pour certains noyaux K , la somme

$$f(x_i) := \sum_{j=1}^M \sigma_j K(x_i, y_j), \quad 1 \leq i \leq N, \quad (7)$$

de complexité algorithmique $O(NM)$ peut être approchée par une procédure de complexité asymptotique inférieure par FMM, en $O(\bar{N})$ ou $O(\bar{N} \log(\bar{N}))$, où $\bar{N} = \max(N, M)$.

Dans la suite, nous présentons la méthode en $1D$, mais elle est directement extensible en $3D$ – les simulations numériques présentées dans ce manuscrit sont en $3D$. Le domaine de calcul $\Omega = (-\frac{L}{2}, \frac{L}{2})$ est découpé en intervalles égaux de façon récursive sous la forme d'un arbre binaire avec $\kappa + 1$ niveaux. Le niveau 0 est la racine de l'arbre : il n'y a qu'un seul intervalle égal à Ω , le niveau 1 contient le partitionnement de Ω en deux intervalles de longueur $\frac{L}{2}$, et les niveaux suivants sont obtenus en partitionnant de la même façon chaque intervalle du niveau précédent, jusqu'au niveau κ , le sommet de l'arbre. La hauteur de l'arbre correspond au nombre de niveaux. Au niveau k , le domaine contient 2^k intervalles de longueur $2l_\kappa$, avec $l_\kappa := 2^{-\kappa-1}L$. Pour généraliser aux dimensions supérieures, les intervalles sont appelés "boîtes" dans la suite (un exemple d'octree de hauteur 1 en $3D$ est illustré dans la partie gauche de la Figure 7).

Considérons une boîte I^k au niveau k . Nous appelons voisinage $\mathcal{N}(I^k)$ d'une boîte I^k l'ensemble contenant I^k et les boîtes adjacentes à I^k au niveau k . L'ensemble des autres boîtes du niveau k , $\mathcal{F}(I^k)$, sont les boîtes en interaction lointaine par rapport à I^k . Une boîte J^{k+1} est un enfant de I^k si elle est au niveau $k+1$ et est incluse dans I^k . De même, le parent de I^k , noté $\mathcal{P}(I^k)$, est la boîte du niveau $k-1$ contenant I^k . Deux boîtes sont dites bien séparées si elles sont en interaction lointaine. La liste d'interaction de I^k , notée $\mathcal{I}(I^k)$, est l'ensemble des boîtes qui sont les enfants des boîtes dans le voisinage du parent de la boîte I^k et qui sont bien séparées de I^k au niveau k , voir Figure 6.

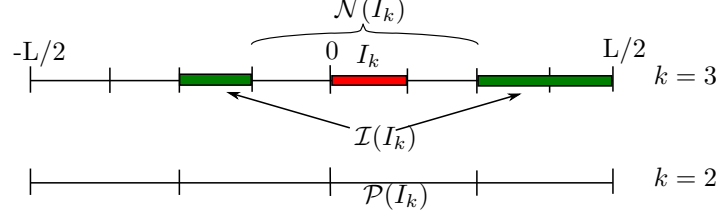


FIGURE 6 – Représentation du voisinage et de la liste d'interaction de la boîte I^k

L'algorithme FMM repose sur une approximation de rang faible du comportement du champ lointain du noyau de la forme :

$$K(x, y) \approx \sum_{l,m=1}^d A_{l,m} g_l(x) h_m(y), x \in I, y \in J, \quad (8)$$

where I and J sont des boîtes bien séparées.

Remarque 0.1 (Dépendances des quantités aux boîtes). *Pour que l'approximation de rang-faible (8) soit compatible avec une FMM multiniveau, $g_l(x)$ doit être indépendant de la boîte J et $h_m(y)$ doit être indépendant de la boîte I .*

De telles approximations de rang faible du comportement du champ lointain du noyau peuvent être analytiques, comme l'interpolation de Chebyshev [54, 62]) ou des développements multipolaires. Des FMM pour noyaux génériques ont été développés, conduisant à des procédures indépendantes du noyau, voir [86, 121] ou encore la Black-Box FMM [53] une implémentation reposant sur l'interpolation de Chebyshev. Pour d'autres procédures indépendantes du noyau, nous citons [45, 46]. L'Adaptive Cross Approximation (ACA) [19, 20] construit des approximations directement depuis les évaluations du noyau sur les points sources et d'observation $K(\bar{x}_i, \bar{y}_j)$, en sélectionnant certaines colonnes et lignes de la matrice $(K(\bar{x}_i, \bar{y}_j))_{i,j}$. L'approximation d'une matrice par extraction de certaines de ses lignes et colonnes a été étudié dans [56]. Avec les méthodes ACA, l'approximation de rang faible (8) est telle que g_l dépend de J et h_m dépend I , ce qui conduit à un stockage de complexité $dN \log(N)$. Avec les FMM multiniveaux et les H2-matrices, g_l et h_m sont indépendants de respectivement J et I , ce qui permet des factorisations supplémentaires et l'utilisation de bases imbriquées pour abaisser la complexité du stockage à dN . Un ACA amélioré exploitant des bases imbriquées a été proposé dans [21].

L'approximation de rang faible (8) permet la factorisation du calcul de quantités clés et la propagation de l'information à travers une structure d'arbre, résultant en l'optimisation de la complexité algorithmique du calcul de la quantité d'intérêt (7). Nous donnons quelques éléments sur les flux d'information dans la structure d'arbre dans la Figure 7.

Plan du manuscrit

Nous commençons par le Chapitre 1, où une méthode de couplage éléments de frontière / éléments finis volumiques (BEM/FEM) est introduite pour la simulation numérique de la propagation acoustique en milieu non borné, avec comme application le bruit rayonné par un turbo-réacteur. Nous montrons que cette méthode est bien posée. Dans le Chapitre 2, nous présentons les travaux autour de la simulation du potentiel de gravitation et ses dérivées première et seconde, générés par une source volumique quelconque de matière. Un couplage BEM/FEM est testé, mais la méthode la plus prometteuse est basée sur la Fast Multipole Method (FMM). Enfin, dans ce chapitre est également proposée une nouvelle méthode FMM pour noyaux invariants par translation.

Dans le Chapitre 3, le plus conséquent, sont regroupés les travaux sur la réduction de modèle. La première section est une application de la méthode des bases réduites avec l'état de l'art de l'époque, la deuxième section investigate des problèmes de réductibilité et de stabilité en présence de phénomènes de translation importants, la troisième section introduit une nouvelle formulation pour EIM, la quatrième section propose un nouvel estimateur a posteriori pour la méthode des bases réduites, plus robuste aux erreurs d'arrondis machine que l'estimateur classique et enfin la cinquième section regroupe les contributions visant à réduire la contrainte d'intrusivité pour les méthodes de réduction de modèle par projection. D'abord, un nouvel EIM non-intrusif est introduit, puis une méthode originale pour approcher des limites d'algorithmes de puissance de façon non-intrusive est proposée (avec des premiers éléments théoriques sur la convergence de l'approximation proposée) et appliquée à des solutions de problèmes linéaires et non-linéaires. Le chapitre se poursuit ensuite avec la mise en place d'un environnement logiciel capable de construire des modèles réduits non-intrusifs (c'est-à-dire avec des snapshot générés par des codes commerciaux), entièrement parallèles en mémoire distribuée, pour la mécanique des structures non-linéaire, avec des applications sur des problèmes plus coûteux que ceux résolus actuellement dans l'industrie. Ensuite, un nouvel indicateur d'erreur efficace pour les problèmes nonlinéaires en mécanique des structures, implémenté dans la librairie parallèle nonintrusive décrite précédemment. Enfin, le chapitre se termine avec l'utilisation de cette librairie nonintrusive à la réduction d'un problème de thermique transitoire nonlinéaire avec variabilité nonparamétrée, pour le design d'un compresseur haute-pression.

Le Chapitre 1 et les Sections 3.1, 3.4 et 3.5.1 correspondent à la période de doctorat. Ces travaux, plus anciens, sont résumés de façon très concise : ils ont été présentés en détail dans le manuscrit de thèse. Par exemple, nous n'avons pas présenté les méthodes d'équations intégrales dans cette introduction (dans le manuscrit de thèse, elle sont introduites dans la Section 1.3 et elles sont l'objet de la Partie I). Le Chapitre 2 et la Section 3.3 correspondent à la période de chercheur à l'IGN, et le reste du manuscrit, à savoir les Sections 3.2, 3.5.2-3.5.5, et 3.6 correspondent à la période de chercheur chez Safran.

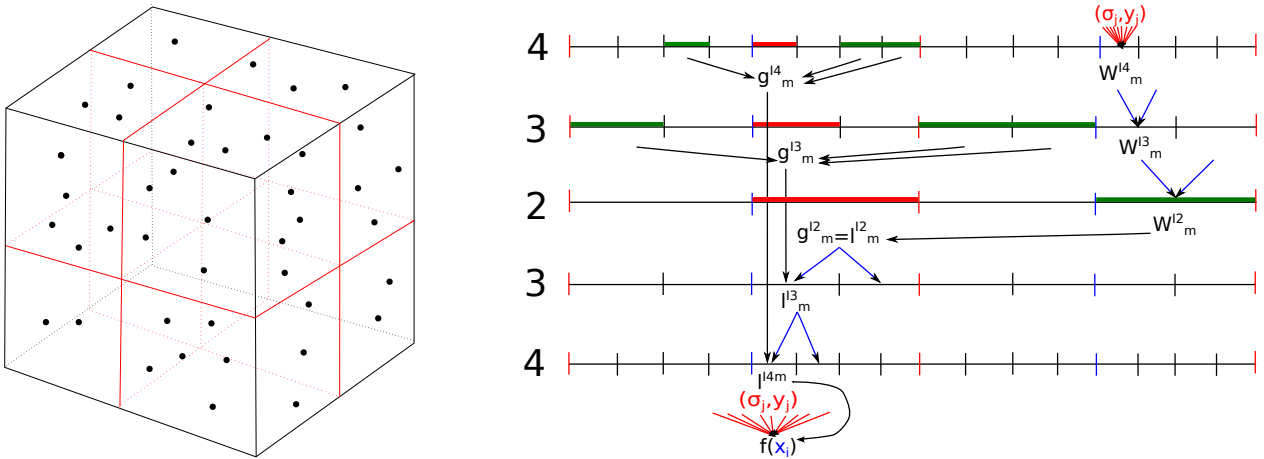


FIGURE 7 – Gauche : exemple d'arbre de hauteur 1 en 3D : la racine (niveau 0) de l'arbre est représentée en noir et contient le domaine complet ; en rouge sont représentées les boîtes du niveau 1 (8 boîtes : 2 par direction). Droite : flux d'information dans un arbre en 1D, l'algorithme étant exécuté de haut en bas. D'abord, les vecteurs W (intermédiaires de calcul) sont calculés dans chaque boîte à partir des informations locales σ_j et y_j (étape P2M : particule to multipole). Ensuite, dans la passe upward, les vecteurs locaux W sont assemblés à chaque niveau (la taille et la localisation des boîtes changent d'un niveau à l'autre dans l'arbre) en utilisant les vecteurs W du niveau précédent (étape M2M : multipole to multipole). À chaque niveau, le vecteur g combine les vecteurs W de la liste d'interaction (en vert) de la boîte considérée (en rouge) (étape M2L : multipole to local). La passe downward consiste à calculer les vecteurs l , en utilisant les vecteurs g du niveau considéré et les vecteurs W du niveau précédent (étape L2L : local to local). Enfin, au dernier niveau, le vecteur l correspond à une composante de l'interaction lointaine, à laquelle une étape de complexité quadratique locale est ajoutée pour obtenir une approximation de $f(x_i)$ (étape P2P : particule to particule).

Chapitre 1

Simulation numérique de la propagation acoustique avec écoulement en milieu non borné

Cette section correspond aux Publications [Pub.16](#) et [Pub.17](#) et aux Exposés [Exp.10](#) à [Exp.13](#), [Exp.16](#), [Exp.17](#), [Exp.21](#) et [Exp.22](#).

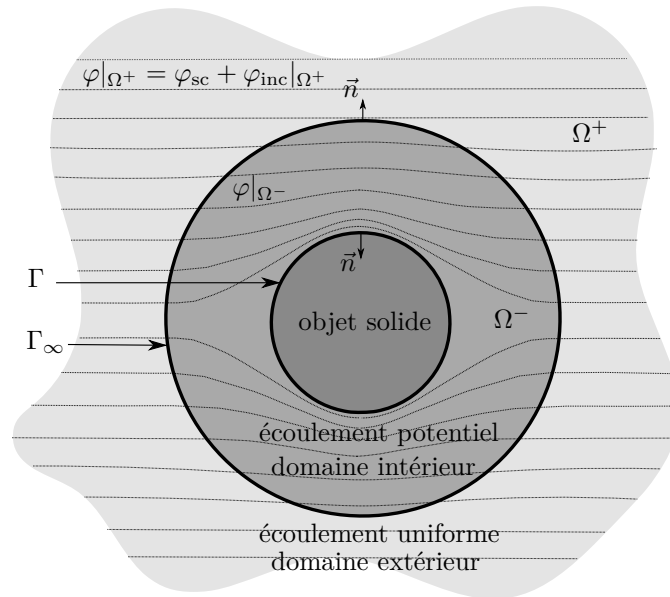


FIGURE 1.1 – Géométrie du cas test pour le problème aéroacoustique dans un écoulement potentiel dans un domaine borné Ω^- et uniforme à l'extérieur de ce domaine

La Figure 1.1 présente la géométrie du cas test. Le domaine intérieur, correspondant à la zone près de l'objet diffractant où l'écoulement est non uniforme, est noté Ω^- . Dans le domaine extérieur, Ω^+ , l'écoulement est supposé uniforme. Le domaine complet de propagation, noté $\Omega \subset \mathbb{R}^3$, est tel que $\Omega := \Omega^+ \cup \Omega^- \cup \Gamma_\infty = \mathbb{R}^3 \setminus \{\text{objet solide}\}$, où $\Gamma_\infty := \partial\Omega^+ \cap \partial\Omega^-$ est la frontière entre les domaines intérieur et extérieur. La surface Γ_∞ est supposée Lipschitz. Une telle hypothèse est suffisamment générale pour inclure les surfaces polyédriques que nous obtenons lorsque nous utilisons un maillage par éléments finis dans Ω^- . La surface de l'objet solide diffractant, $\partial\Omega^- \setminus \Gamma_\infty$, est notée Γ et est supposée Lipschitz.

L'équation fondamentale sur laquelle nous nous appuyons dans cette partie est l'équation d'Helmholtz convectée. Dans le chapitre 1.2 du manuscrit de thèse, nous détaillons les hypothèses et les calculs permettant de dériver cette équation à partir des équations de Navier-Stokes. D'abord, nous supposons que, dans les

conditions normales de température et de pression, l'air peut être considéré comme un gaz parfait. Ensuite, nous supposons que dans l'approximation acoustique, les termes dus au tenseur des contraintes et au flux de chaleur sont négligeables, et que les phénomènes acoustiques instationnaires sont une perturbation au premier ordre d'un écoulement moyen stationnaire. Ensuite, en supposant l'écoulement irrotationnel et homentropique et les perturbation harmoniques, nous obtenons l'équation d'Helmholtz convectée

$$\rho \left(k^2 \varphi + ik \vec{M} \cdot \vec{\nabla} \varphi \right) + \vec{\nabla} \cdot \left[\rho \left(\vec{\nabla} \varphi - \left(\vec{M} \cdot \vec{\nabla} \varphi \right) \vec{M} + ik \varphi \vec{M} \right) \right] = g, \quad (1.1)$$

où ρ est la densité de l'air, $k = \frac{\omega}{c}$ est le nombre d'onde, avec ω la fréquence de la source et c la vitesse du son (au repos), $\vec{M} = \frac{\vec{v}_m}{c}$ est la vitesse moyenne renormalisée, avec \vec{v}_m la vitesse de l'écoulement moyen, et g est le terme source : un monopole acoustique harmonique de pulsation ω , localisé à $x_s \in \Omega^+$ et d'amplitude A_s , tel que $g := A_s \delta_{x_s} \cos(\omega t)$, où δ représente la distribution de Dirac. L'inconnue φ , le potentiel acoustique, est telle que $\varphi' = \text{Re}(\varphi e^{-i\omega t})$, où $\vec{v}_t - \vec{v}_m = \vec{v} = \vec{\nabla} \varphi'$, avec \vec{v}_t la vitesse totale et \vec{v} la vitesse acoustique. La pression acoustique est notée p . Dans le cas où l'écoulement moyen stationnaire est nul, ρ et k sont uniformes et $\vec{M} = \vec{0}$, conduisant à l'équation d'Helmholtz classique $\Delta \varphi + k^2 \varphi = g$. Dans nos notations, l'exposant ∞ renvoie vers des quantités propres à l'écoulement uniforme dans Ω^+ , tandis que l'exposant 0 renvoie vers des quantités dépendant de l'écoulement non uniforme dans Ω^- , telles que $\rho_{|\Omega^-} = \rho_0(\vec{x})$, $\rho_{|\Omega^+} \equiv \rho_\infty$, $k_{|\Omega^-} = k_0(\vec{x})$, $k_{|\Omega^+} \equiv k_\infty$, $c_{|\Omega^-} = c_0(\vec{x})$, $c_{|\Omega^+} \equiv c_\infty$, $\vec{M}_{|\Omega^-} = \vec{M}_0(\vec{x})$, $\vec{M}_{|\Omega^+} \equiv \vec{M}_\infty$. L'écoulement est continu à travers Γ_∞ et tangent à Γ . Ainsi, ρ , k et \vec{M} sont continus à travers Γ_∞ , et $\vec{M} \cdot \vec{n} = 0$ sur Γ , \vec{n} est la normale sortante sur Γ . La condition aux limites sur Γ est une condition de Neumann homogène : $\vec{\nabla} \varphi \cdot \vec{n} = 0$ et la condition aux limites à l'infini est une condition de radiation de Sommerfeld. Les quantités physiques sont associées à des quantités complexes avec la convention suivante sur, par exemple, la pression acoustique : $p \leftrightarrow \text{Re}(p \exp(-i\omega t))$. Dans ce qui suit, nous considérons toujours les quantités complexes. Par ailleurs, le produit hermitien de deux vecteurs $\vec{U}, \vec{W} \in \mathbb{C}^3$ est noté $\vec{U} \cdot \vec{W} := \sum_{i=1}^3 \bar{U}_i W_i$, où $\bar{\cdot}$ représente la conjugaison complexe, et la norme euclidienne associées dans \mathbb{C}^3 est notée $\|\cdot\|$.

Une transformation de Prandtl–Glauert, basée sur la vitesse renormalisée \vec{M}_∞ , est appliquées au domaine complet de propagation. Elle consiste au changement de variables spatiales et temporelles suivant

$$\begin{cases} \vec{x}' = \gamma_\infty \left(\hat{M}_\infty \cdot \vec{x} \right) \hat{M}_\infty + \left(\vec{x} - \left(\hat{M}_\infty \cdot \vec{x} \right) \hat{M}_\infty \right) & \vec{x} \in \Omega, \\ t' = t - \frac{\gamma_\infty^2}{c_\infty} \vec{M}_\infty \cdot \vec{x} & t \in \mathbb{R}, \end{cases} \quad (1.2)$$

où $\gamma_\infty := \frac{1}{\sqrt{1-M_\infty^2}}$ et $\hat{M}_\infty := M_\infty^{-1} \vec{M}_\infty$ avec $M_\infty := |\vec{M}_\infty|$. La transformation spatiale correspond à une dilatation le long de \hat{M}_∞ d'amplitude γ_∞ , la composante orthogonale à \hat{M}_∞ restant inchangée. Dans ce qui suit, nous supposons que $M_\infty < 1$, de sorte que la transformation de Prandtl–Glauert est un C^∞ -difféomorphisme de $\Omega \times \mathbb{R}$ dans $\Omega' \times \mathbb{R}$, où Ω' représente le domaine de propagation transformé. Soit f tel que $\varphi(\vec{x}) = f(\vec{x}') \exp\left(-ik_\infty \gamma_\infty \left(\vec{M}_\infty \cdot \vec{x}' \right)\right)$, $\vec{x}' \in \Omega'$; f_{inc} et f_{sc} sont définis à partir de φ_{inc} et φ_{sc} de la même façon, de sorte que f_{inc} est connu analytiquement et défini dans \mathbb{R}^3 . Soit $\varsigma(\vec{x}') := \rho_\infty^{-1} \exp\left(ik_\infty \gamma_\infty \left(\vec{M}_\infty \cdot \vec{x}' \right)\right) g(\vec{x}')$, $\vec{x}' \in \Omega'$. Dans ce qui suit, nous considérons systématiquement la géométrie, les inconnues et les opérateurs transformés, sauf lorsque spécifié autrement. Par souci de concision, les primes sont omis. Pour appliquer la transformation de Prandtl–Glauert à une équation aux dérivées partielles dans le domaine fréquentiel, il faut d'abord modifier les opérateurs différentiels de la façon suivante :

$$\vec{\nabla} u = \mathcal{N} \vec{\nabla}' u, \quad \vec{\nabla} \cdot \vec{U} = \vec{\nabla}' \cdot \mathcal{N} \vec{U}, \quad (1.3)$$

pour une fonction scalaire u et une fonction vectorielle \vec{U} . Ici, $\mathcal{N} = I + C_\infty \vec{M}_\infty \vec{M}_\infty^T$ avec $C_\infty = \frac{\gamma_\infty - 1}{M_\infty^2}$ et $\gamma_\infty = \frac{1}{\sqrt{1-M_\infty^2}}$, et $\vec{\nabla}'$ désigne le gradient par rapport aux variables transformées \vec{x}' . De plus, on vérifie facilement que

$$\mathcal{N} \vec{M} = \vec{M} + C_\infty P \vec{M}_\infty, \quad \mathcal{N} \vec{M}_\infty = \gamma_\infty \vec{M}_\infty, \quad \mathcal{N} \vec{M} \cdot \vec{M}_\infty = \mathcal{N} \vec{M}_\infty \cdot \vec{M} = \gamma_\infty P, \quad (1.4)$$

où $P = \vec{M} \cdot \vec{M}_\infty$. Après modification des opérateurs différentiels, il faut changer la fonction inconnue par $\varphi(\vec{x}) = \alpha(\vec{x}') f(\vec{x}')$, où $\alpha(\vec{x}') := \exp\left(-ik_\infty \gamma_\infty \left(\vec{M}_\infty \cdot \vec{x}' \right)\right)$.

Après transformation de Prandtl–Glauert, l'équation (1.1) devient

$$rk^2\beta f + irk\vec{V} \cdot \vec{\nabla}' f + \vec{\nabla}' \cdot (irkf\vec{V} + r\Xi\vec{\nabla}' f) = \varsigma, \quad (1.5)$$

où $r = \frac{\rho}{\rho_\infty}$, $\beta = (1 + qP)^2 - q^2M_\infty^2$, $\vec{V} = (1 + qP)\mathcal{N}\vec{M} - q\gamma_\infty\vec{M}_\infty$, $q = \gamma_\infty^2 \frac{k_\infty}{k}$, $\Xi = \mathcal{N}(I - \vec{M}\vec{M}^T)\mathcal{N}$, et $\varsigma = \rho_\infty^{-1}\alpha^{-1}g$. La condition aux limites sur Γ devient $(irkf\vec{V} + r\Xi\vec{\nabla}' f) \cdot \vec{n}' = 0$.

Nous montrons que le problème et les conditions aux limites transformées sont équivalents au problème de transmission suivant, où l'opérateur de trace $\gamma_{n,\Gamma}^-$ de Ω^- sur Γ est utilisé pour formuler la condition aux limites sur l'objet solide :

$$rk^2\beta f^- + irk\vec{V} \cdot \vec{\nabla} f^- + \vec{\nabla} \cdot (irkf\vec{V} + r\Xi\vec{\nabla} f)^- = 0 \quad \text{in } \Omega^-, \quad (1.6a)$$

$$\Delta f_{sc} + \hat{k}_\infty^2 f_{sc} = 0 \quad \text{in } \Omega^+, \quad (1.6b)$$

$$\gamma_{n,\Gamma}^- (irkf\vec{V} + r\Xi\vec{\nabla} f)^- = 0 \quad \text{on } \Gamma, \quad (1.6c)$$

$$\gamma_0^+ f^+ - \gamma_0^- f^- = 0 \quad \text{on } \Gamma_\infty, \quad (1.6d)$$

$$\gamma_1^+ f^+ - \gamma_1^- f^- = 0 \quad \text{on } \Gamma_\infty, \quad (1.6e)$$

$$\lim_{r \rightarrow +\infty} r \left(\frac{\partial(f^+ - f_{inc}^+)}{\partial r} - i\hat{k}_\infty(f^+ - f_{inc}^+) \right) = 0. \quad (1.6f)$$

Nous dérivons maintenant une formulation variationnelle. Soit X , représentant la surface Γ ou Γ_∞ . Le produit scalaire dans $L^2(X)$, $\langle \cdot, \cdot \rangle_{L^2(X), L^2(X)} : L^2(X) \times L^2(X) \rightarrow \mathbb{C}$, est défini par

$$\langle \lambda, \mu \rangle_{L^2(X), L^2(X)} := \int_X \bar{\lambda}(\vec{y}) \mu(\vec{y}) ds(\vec{y}). \quad (1.7)$$

Ce produit scalaire peut être étendu en produit de dualité sur $H^{-\frac{1}{2}}(X) \times H^{\frac{1}{2}}(X)$, noté $\langle \cdot, \cdot \rangle_{H^{-\frac{1}{2}}(X), H^{\frac{1}{2}}(X)}$. Nous définissons maintenant le produit

$$(\lambda, \mu)_X := \begin{cases} \langle \lambda, \mu \rangle_{H^{-\frac{1}{2}}(X), H^{\frac{1}{2}}(X)} & \text{if } \lambda \in H^{-\frac{1}{2}}(X), \quad \mu \in H^{\frac{1}{2}}(X), \\ \overline{\langle \mu, \lambda \rangle_{H^{-\frac{1}{2}}(X), H^{\frac{1}{2}}(X)}} & \text{if } \lambda \in H^{\frac{1}{2}}(X), \quad \mu \in H^{-\frac{1}{2}}(X). \end{cases} \quad (1.8)$$

La formulation variationnelle du problème est : trouver $(\Phi, \lambda, p) \in \mathbb{H}$ tel que $\forall (\Phi^t, \lambda^t, p^t) \in \mathbb{H}$,

$$\mathcal{V}(\Phi, \Phi^t) + (N(\gamma_0^- \Phi), \gamma_0^- \Phi^t)_{\Gamma_\infty} + \left(\left(\tilde{D} - \frac{1}{2}I \right) (\lambda), \gamma_0^- \Phi^t \right)_{\Gamma_\infty} = (\gamma_1 f_{inc}, \gamma_0^- \Phi^t)_{\Gamma_\infty}, \quad (1.9a)$$

$$\left(\left(D - \frac{1}{2}I \right) (\gamma_0^- \Phi), \lambda^t \right)_{\Gamma_\infty} - (S(\lambda), \lambda^t)_{\Gamma_\infty} + i\bar{\eta} (p, \lambda^t)_{\Gamma_\infty} = -(\gamma_0 f_{inc}, \lambda^t)_{\Gamma_\infty}, \quad (1.9b)$$

$$(N(\gamma_0^- \Phi), p^t)_{\Gamma_\infty} + \left(\left(\tilde{D} + \frac{1}{2}I \right) (\lambda), p^t \right)_{\Gamma_\infty} - \delta_{\Gamma_\infty}(p, p^t) = (\gamma_1 f_{inc}, p^t)_{\Gamma_\infty}, \quad (1.9c)$$

avec l'espace produit $\mathbb{H} := H^1(\Omega^-) \times H^{-\frac{1}{2}}(\Gamma_\infty) \times H^1(\Gamma_\infty)$ et le produit scalaire $((\Phi, \lambda, p), (\Phi^t, \lambda^t, p^t))_{\mathbb{H}} := (\Phi, \Phi^t)_{H^1(\Omega^-)} + (\lambda, \lambda^t)_{H^{-\frac{1}{2}}(\Gamma_\infty)} + (p, p^t)_{H^1(\Gamma_\infty)}$, où

$$\delta_{\Gamma_\infty}(p, q) := \left(\vec{\nabla}_{\Gamma_\infty} p, \vec{\nabla}_{\Gamma_\infty} q \right)_{\Gamma_\infty} + (p, q)_{\Gamma_\infty}, \quad (1.10)$$

où S , D , \tilde{D} et N sont respectivement les opérateurs intégraux de simple couche, double couche, transposé de l'opérateur double couche et hypersingulier et où γ_0^- est la trace intérieure de Dirichlet sur Γ_∞ , et γ_0 et γ_1 représentent les traces de Dirichlet et Neumann sur Γ_∞ , lorsque cette trace est continue à travers Γ_∞ . Le paramètre complexe η , tel que $\text{Re}(\eta) \neq 0$, est le paramètre de couplage de l'équation intégrale à champs combinés utilisée pour stabiliser l'opérateur de Dirichlet-to-Neumann sous-jacent. De nombreux détails sur ce couplage et les propriétés des opérateurs intégraux et des opérateurs de trace sont donnés dans le manuscrit de thèse, notamment dans la section 3.2.

Nous montrons que si f est solution de (1.6), alors $(f^-, \gamma_1 f, 0)$ est solution de (1.9). Réciproquement, si (Φ, λ, p) est solution de (1.9), alors $\mathcal{R}(\Phi, \lambda)$ est solution de (1.6) et $p = 0$, où $\mathcal{R} : \mathcal{H} \rightarrow H_{\text{loc}}^1(\Omega \setminus \Gamma_\infty)$ est tel que $\mathcal{R}(\Phi, \lambda)|_{\Omega^-} := \Phi$ et $\mathcal{R}(\Phi, \lambda)|_{\Omega^+} := (-\mathcal{S}(\lambda) + \mathcal{D}(\gamma_0^- \Phi) + f_{\text{inc}})|_{\Omega^+}$, où \mathcal{S} et \mathcal{D} sont les potentiels de simple couche et de double couche. Nous montrons aussi que (1.9) est bien posé à toutes les fréquences de la source. La preuve repose sur l'alternative de Fredholm : (i) l'unicité est montrée grâce au lemme de Rellich et une propriété de continuation unique forte, avec "régularité faible" des coefficients, (ii) la forme sesquilinéaire dans (1.9) est continue et (iii) cette forme sesquilinéaire peut être décomposée en une partie coercive (faisant intervenir les opérateurs intégraux à fréquence nulle) et une perturbation compacte. Les preuves complètes sont disponibles dans la section 3.7 du manuscrit de thèse.

L'utilisation de Combined Fields Integral Equations (CFIE) est bien connue pour assurer l'unicité des problèmes intégraux à toutes les fréquences. La formulation proposée (1.9) repose justement sur un opérateur Dirichlet-to-Neumann défini pour toutes les fréquences de la source par CFIE. Cet opérateur peut être interprété comme un cas particulier du formalisme générique proposé dans [69] sous la forme transformation d'opérateurs de trace, dans le cadre de problème de transmission de Helmholtz. Notre formulation (1.9) est une extension de [69] au problème d'Helmholtz convecté avec un écoulement non-uniforme dans une région bornée de l'espace, pour un cas particulier de transformation d'opérateur de trace dans ce formalisme.

Nous nous intéressons maintenant à l'approximation en dimension finie de l'équation (1.9). Soit \mathcal{M} un maillage tétraédral régulier de Ω^- . Le maillage \mathcal{F}_∞ de Γ_∞ est composé des faces libres de \mathcal{M} . Soient $h_\mathcal{M} > 0$ le diamètre de maille, $V_\mathcal{M}^1$ l'espace des polynômes continus affines par morceau sur \mathcal{M} , $S_\mathcal{M}^0$ l'espace des polynômes constants par morceaux sur \mathcal{F}_∞ , et $S_\mathcal{M}^1$ l'espace des polynômes continus affines par morceau sur \mathcal{F}_∞ . Soit $\mathbb{H}_\mathcal{M} := V_\mathcal{M}^1 \times S_\mathcal{M}^0 \times S_\mathcal{M}^1$. La discrétisation de (1.9) est : trouver $(\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M}) \in \mathbb{H}_\mathcal{M}$ tel que, $\forall (\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t) \in \mathbb{H}_\mathcal{M}$,

$$a((\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M}), (\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t)) = b(\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t), \quad (1.11)$$

avec a et b directement déduits de (1.9). Dans ce qui suit, nous notons $A \lesssim B$ l'inégalité $A \leq cB$ avec c une constante positive indépendante du diamètre de maille et des solutions discrète et continue. Nous montrons la propriété d'approximation suivante : $\forall (\Phi, \lambda, p) \in H^2(\Omega^-) \times H^{\frac{1}{2}}(\Gamma_\infty) \times H^2(\Gamma_\infty)$,

$$\inf_{(\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M}) \in \mathbb{H}_\mathcal{M}} \|(\Phi, \lambda, p) - (\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M})\|_{\mathbb{H}} \lesssim h_\mathcal{M} \left(\|\Phi\|_{H^2(\Omega^-)} + \|\lambda\|_{H^{\frac{1}{2}}(\Gamma_\infty)} + \|p\|_{H^2(\Gamma_\infty)} \right). \quad (1.12)$$

Soient $(\theta_i)_{1 \leq i \leq p}$, $(\psi_i)_{1 \leq i \leq q}$ et $(\xi_i)_{1 \leq i \leq r}$, les bases éléments finis pour $V_\mathcal{M}^1$, $S_\mathcal{M}^0$ et $S_\mathcal{M}^1$ respectivement. La décomposition de, par exemple, $p_\mathcal{M} \in S_\mathcal{M}^1$ sur la base $(\xi_i)_{1 \leq i \leq r}$ est écrite de la forme $p_\mathcal{M} = \sum_{i=1}^r p_{\mathcal{M},i} \xi_i$. Ces fonctions de bases sont à valeurs réelles. Soient

$$u_\mathcal{M} = \begin{pmatrix} (\Phi_{\mathcal{M}i})_{1 \leq i \leq p} \\ (\lambda_{\mathcal{M}i})_{1 \leq i \leq q} \\ (p_{\mathcal{M}i})_{1 \leq i \leq r} \end{pmatrix}, \quad B = \begin{pmatrix} (\gamma_1 f_{\text{inc}}, \gamma_0^- \theta_i)_{\Gamma_\infty} & 1 \leq i \leq p \\ -(\gamma_0 f_{\text{inc}}, \psi_i)_{\Gamma_\infty} & 1 \leq i \leq q \\ (\gamma_1 f_{\text{inc}}, \xi_i)_{\Gamma_\infty} & 1 \leq i \leq r \end{pmatrix}, \quad (1.13)$$

$$A = \left(\begin{array}{c|c|c} \mathcal{V}(\theta_j, \theta_i) + (N(\gamma_0^- \theta_j), \gamma_0^- \theta_i)_{\Gamma_\infty} & ((\tilde{D} - \frac{1}{2}I)(\psi_j), \gamma_0^- \theta_i)_{\Gamma_\infty} & 0 \\ \hline ((D - \frac{1}{2}I)(\gamma_0^- \theta_j), \psi_i)_{\Gamma_\infty} & -(S(\psi_j), \psi_i)_{\Gamma_\infty} & i\bar{\eta}(\xi_j, \psi_i)_{\Gamma_\infty} \\ \hline (N(\gamma_0^- \theta_j), \xi_i)_{\Gamma_\infty} & ((D - \frac{1}{2}I)(\psi_j), \xi_i)_{\Gamma_\infty} & -\delta_{\Gamma_\infty}(\xi_j, \xi_i)_{\Gamma_\infty} \end{array} \right), \quad (1.14)$$

où, dans la définition de A , i désigne les lignes et j , les colonnes. Le système linéaire résultant de (1.11) est

$$Au_\mathcal{M} = B. \quad (1.15)$$

Nous montrons maintenant l'inf-sup stabilité de (1.11). Pour toutes les fréquences de la source, si $h_\mathcal{M}$ est suffisamment petit, alors pour tout $(\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M}) \in \mathbb{H}_\mathcal{M}$,

$$\sup_{(0,0,0) \neq (\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t) \in \mathbb{H}_\mathcal{M}} \frac{|a^{\text{stab}}((\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M}), (\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t))|}{\|(\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t)\|_{\mathbb{H}}} \gtrsim \|(\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M})\|_{\mathbb{H}}. \quad (1.16)$$

Nous pouvons alors en déduire l'estimation d'erreur optimale suivante. Pour toutes les fréquences de la source, si $h_\mathcal{M}$ est suffisamment petit, le problème discret (1.11) a une unique solution $(\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M}) \in \mathbb{H}_\mathcal{M}$, et

$$\|(\Phi, \lambda, p) - (\Phi_\mathcal{M}, \lambda_\mathcal{M}, p_\mathcal{M})\|_{\mathbb{H}} \lesssim \inf_{(\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t) \in \mathbb{H}_\mathcal{M}} \|(\Phi, \lambda, p) - (\Phi_\mathcal{M}^t, \lambda_\mathcal{M}^t, p_\mathcal{M}^t)\|_{\mathbb{H}}, \quad (1.17)$$

où (Φ, λ, p) est l'unique solution de (1.9).

Dans le manuscrit de thèse, nous présentons également une autre formulation variationnelle, qui est sensible à des fréquences de résonance, pour lesquelles elle admet une infinité de solutions. Cette formulation est appelée instable, alors que la formulation (1.9) est appelée stable du fait qu'elle est bien posée à toutes les fréquences. La formulation instable correspondait à une première étape dans le processus de recherche et conduit à des systèmes linéaires de plus petite taille que la formulation (1.9).

Les deux formulations ont été implémentées dans le code d'éléments de frontière d'Airbus appelé ACTIPOLE. Ce logiciel peut traiter des géométries générales tri-dimensionnelles. Le solveur itératif est un block-GMRES sans restart, valide pour les systèmes linéaires non-symétriques. Ce solveur est optimisé pour prendre en compte les spécificités de chaque bloc de la matrice A (1.14). Les blocs pleins sont accélérés par un produit rapide utilisant la fast multipole method. Des techniques de parallélisation "out-of-core" sont utilisées. Le préconditionneur utilise une combinaison d'inverse approché creux et de factorisation par le solveur direct MUMPS.

De nombreuses simulations ont été réalisées et sont présentées dans le manuscrit de thèse. Dans un premier exemple, nous comparons les formulations stables et instables pour justifier que le surcoût de calcul de la formulation stable est nécessaire. On considère un ellipsoïde de révolution dont l'axe de symétrie est dirigé le long de l'axe z . Cet objet est inclus dans une boule. La frontière extérieure de cette boule, après discrétisation, est la surface Γ_∞ . Un écoulement potentiel est calculé autour de l'ellipsoïde et dans la boule, de sorte que l'écoulement est uniforme à l'extérieur de la boule, de nombre de Mach 0.3 et dirigé le long de l'axe z , voir Figure 1.2. Un monopole acoustique est localisé en amont de la boule, sur l'axe z . Nous considérons 4 maillages différents, Mesh 1 à Mesh 4, du plus fin au plus grossier.

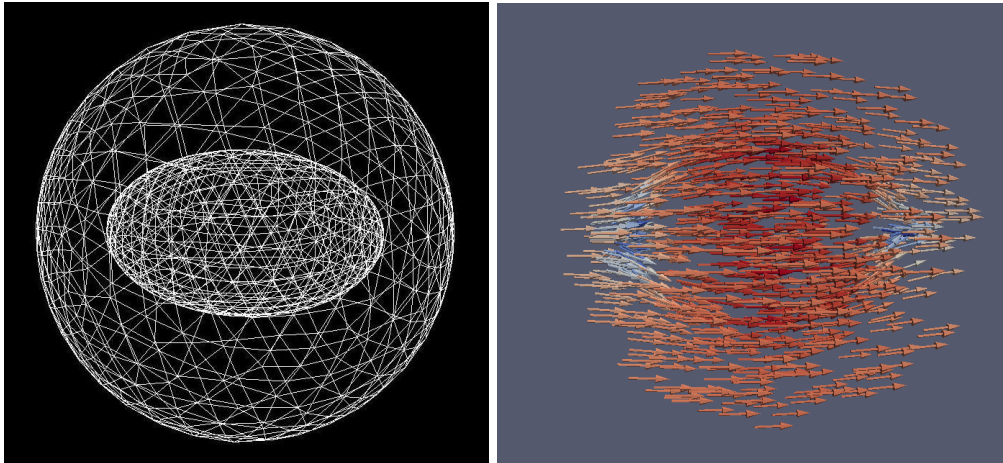


FIGURE 1.2 – Gauche : représentation de Mesh 1, droite : écoulement potentiel autour de l'ellipsoïde.

Dans la Figure 1.3, nous comparons le conditionnement des matrices obtenues avec les formulations stables et instables. Le conditionnement des matrices obtenues avec la formulation stable est insensible aux fréquences de résonance, alors que pour la formulation instable, le conditionnement explose d'autant plus que le maillage est fin. La formule de Weyl indique que le nombre de fréquences de résonance inférieures à f croît comme $f^{\frac{3}{2}}$, indiquant que le recours à la formulation stabilisée est d'autant plus nécessaire que les fréquences étudiées sont élevées. La divergence du conditionnement a des conséquences désastreuses sur la qualité du résultat calculé. Dans la Figure 1.4, nous remarquons que le champ de pression diffracté calculé par la formulation instable est dominé par l'amplification des erreurs d'approximation, conséquence du mauvais conditionnement de la matrice associée.

Nous présentons maintenant un cas-test industriel, simulé par Nolwenn Balin à Airbus. Il consiste en un modèle de turboréacteur simplifié, avec des surfaces modales (surfaces sur lesquelles la source acoustique est définie et décomposée sur une base modale) orthogonales à \vec{e}_z , voir la Figure 1.5 gauche. L'écoulement lointain est défini par $\vec{M}_\infty = -0.3\vec{e}_z$ et l'écoulement imposé au niveau des surfaces modales est défini par $\vec{M}_M = -0.42\vec{e}_z$. L'écoulement potentiel dans le domaine intérieur est calculé par un algorithme de point fixe, voir la Figure 1.5 droite. Le diamètre de maille moyen est 83 mm, correspondant à 11.8×10^6 tétraèdres et 1.2×10^6 degrés de liberté. Nous considérons une source modale sur les pâles amont de fréquence 200 Hz. Le calcul, accéléré par la FMM, prend 1h30 sur 160 cœurs. Sur la Figure 1.6 sont représentés les champs de pression acoustiques correspondant aux deux premiers modes de la source.

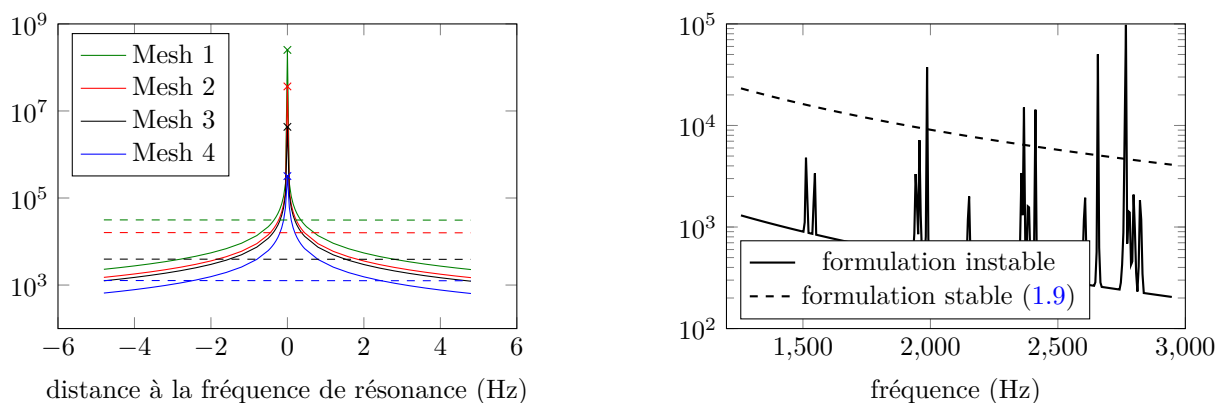


FIGURE 1.3 – Conditionnement de la matrice pour la formulation instable (traits continus) et la formulation stable (1.9) (pointillés). Gauche : représentations centrées autour d’une résonance, pour les quatre maillages, droite : large bande de fréquence avec Mesh 2.

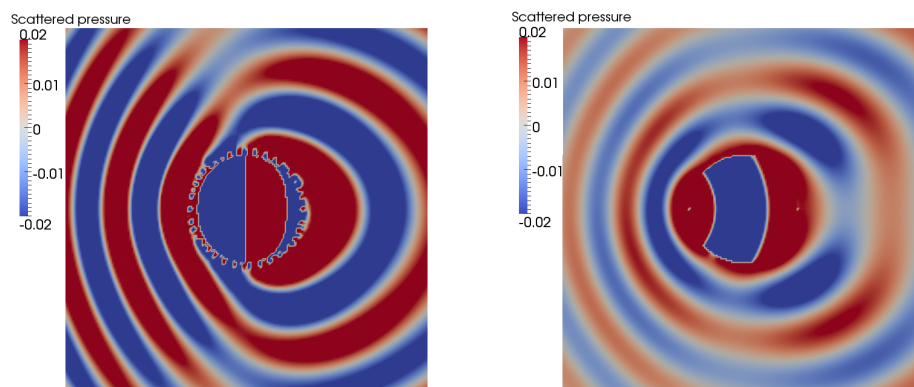


FIGURE 1.4 – Mesh 1, à 1509.849 Hz, une fréquence de résonance. Champ de pression diffracté; gauche : formulation instable, droite : formulation stable (1.9).

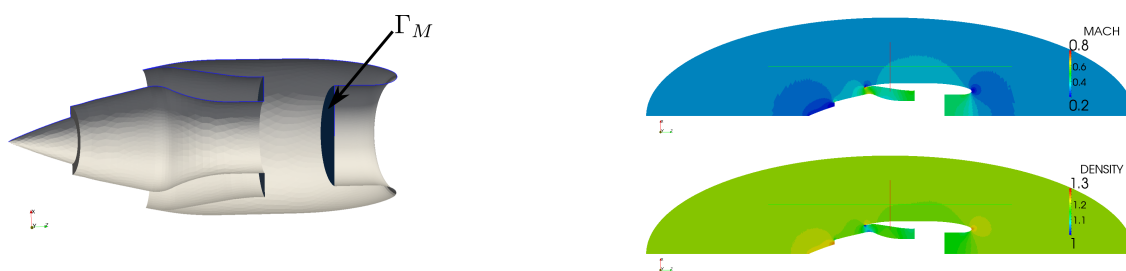


FIGURE 1.5 – Gauche : géométrie du turboréacteur; droite : écoulement potentiel dans le domaine intérieur (haut : nombre de Mach, bas : densité, calcul par N. Balin.)

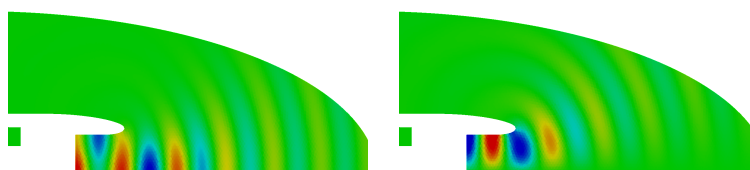


FIGURE 1.6 – Pression acoustique correspondant aux deux premiers modes de la source, calcul par N. Balin.

Chapitre 2

Simulation efficace du potentiel de gravitation généré par une source volumique quelconque

Une partie des recherches conduites au Laboratoire de Recherche en Géodésie (LAREG) de l'IGN ont pour but de répondre aux besoins d'amélioration des références géodésiques, géométriques et altimétriques sur Terre. Avec l'amélioration et le développement exceptionnel des méthodes de mesures ces deux dernières décennies (positionnement par GPS, gravimétrie spatiale, etc), le traitement et l'interprétation des données de géodésie globale demandent de plus en plus de modélisations analytiques et numériques, que ce soit pour la combinaison de données de sources hétérogènes ou pour la correction, par exemple, des mouvements d'origine géophysique des stations de géodésie spatiale. Dans ce cadre, nous pouvons distinguer deux enjeux qui demandent un effort de modélisation particulier :

- La détermination du champ de pesanteur terrestre à haute résolution est un problème intrinsèquement sous déterminé à haute résolution en raison de la faible densité ou qualité des observations disponibles au sol, mais très fortement contraint à basse résolution depuis la disponibilité d'observations de gravimétrie spatiale (missions GRACE [2] puis GOCE [1]). Le LAREG conduit des recherches visant à l'inversion d'une représentation du potentiel de pesanteur (en fonction de la position et du temps) à partir d'observations hétérogènes de gravimétrie, à la fois en orbite et à la surface de la Terre sur des réseaux de mesures complexes. La modélisation précise de la topographie sur le champ de pesanteur est dans ce contexte un problème majeur.
- La composante solide de la Terre se déforme constamment sous l'action de nombreuses sollicitations "externes", tels que les surcharges climatiques engendrées par les enveloppes fluides de surface (océan, atmosphère, hydrologie continentale, glace), les marées luni-solaires, ou encore les séismes. Cela se traduit par le mouvement continu des stations de mesure de géodésie spatiale et par des variations de pesanteur, qui entachent la précision et la stabilité de la détermination des référentiels de positionnement à la fois géographiques (le repère international de référence terrestre dont le LAREG est en charge) et verticaux (la surface équipotentielle de gravité que l'on appelle "géoïde"). Nous parlons de déformations gravito-visco-élastiques de la Terre.

Mes travaux réalisés à l'IGN s'inscrivent dans ce contexte de correction de champs de gravitation d'effets connus. Dans la Section 2.1 un algorithme est proposé pour générer un maillage tétraédrique de façon efficace, modélisant la différence entre la Terre et l'ellipsoïde de référence WGS84, localement ou globalement sur la Terre entière. Dans la Section 2.3, nous proposons une méthode, utilisant la FMM, pour calculer de façon efficace le champs de gravitation (et ses dérivées première et seconde) d'une densité de matière quelconque et nous l'appliquons à la correction des effets topographiques (localement ou globalement sur la Terre) et des effets d'infiltration souterraine d'eau. Enfin, nous proposons dans la Section 2.4 une nouvelle méthode de sommation rapide pour noyaux invariants par translation.

2.1 Un algorithme de maillage adapté au calcul des corrections topographiques

Cette section correspond au Rapport [Rap.4](#).

Les méthodes classiques de correction topographique consistent à projeter des données sur des bases de fonctions comme les harmoniques sphériques, les splines ou les ondelettes [103], qui sont calculées à des points d'un réseau particulier, qui devient de plus en plus dense pour tirer profit de données avec une résolution grandissante. Les temps de calculs deviennent très importants et un besoin est exprimé pour des algorithmes rapides capables d'analyser des données très volumineuses. Par ailleurs, avec les méthodes classiques, la précision des données est altérée lors de leur projection sur la base de fonctions considérée. Par ailleurs, certaines bases, comme les harmoniques sphériques, sont globales (pas de raffinement local possible) et ces bases sont toutes restreintes à des géométries sphériques et elliptiques. Les données sont alors exploitées sous forme de densité surfacique sur ces géométries imposées.

Dans cette section, nous proposons un algorithme pour générer des maillages tétraédriques à partir des données topographiques ETOPO1 [9] : l'information d'altitude du terrain est utilisée directement, sans projection ni interpolation. Pour ce faire, un maillage triangulaire est d'abord proposé, à partir d'une sélection uniforme à la surface du globe des positions associées aux données ETOPO1. Ensuite, la différence entre la surface de la Terre et l'ellipsoïde de référence est ensuite maillée en tétraèdre à partir de ce maillage surfacique. Une attention particulière est accordée au traitement des côtes (l'intersection entre la surface de la Terre et cet ellipsoïde de référence).

Un algorithme de construction rapide d'un maillage triangulaire à la surface d'une sphère

La création d'un maillage homogène à la surface d'une sphère est un problème connu et traité dans la littérature. Parmi les solutions existantes, [18] propose une subdivision itérative des 20 faces d'un icosaèdre régulier, [98] propose la "cubed sphere", la projection de maillages cartésiens depuis les faces d'un cube inscrit dans la sphère et [57] propose une subdivision régulière (en terme d'aire des éléments de la partition) de la surface de la sphère (HEALPix). Nous proposons une méthode alternative simple pour mailler la surface de la sphère avec des triangles, calculée en complexité linéaire par rapport au nombre de nœuds du maillage.

Notons respectivement θ et ϕ la latitude et la longitude. Nous montrons que choisir $\lfloor \frac{\sqrt{N}\pi}{2} \rfloor + 1$ valeurs équidistantes pour la latitude (i.e. $\theta_i = -\frac{\pi}{2} + 2i\sqrt{\frac{\pi}{N}}$, $0 \leq i \leq \lfloor \frac{\sqrt{N}\pi}{2} \rfloor$) et $\lfloor \sqrt{N}\pi \cos(\theta_i) \rfloor$ points équidistants sur le cercle de latitude correspondante conduit à un ensemble de points répartis de façon quasi uniforme à la surface du globe, où $\lfloor \cdot \rfloor$ est la partie entière. En effet, la densité de point à la surface de la sphère est $\frac{\cos(\theta)}{\lfloor \sqrt{N}\pi \cos(\theta) \rfloor \lfloor \frac{\sqrt{N}\pi}{2} \rfloor}$, qui est approximativement indépendante de la position du point. Nous montrons également que le nombre de points ainsi choisis est N (dans la limite $N \gg 1$).

Une fois les points déterminés, la construction du maillage triangulaire repose sur une heuristique simple, illustrée sur la Figure 2.1. Les triangles sont construits récursivement latitude par latitude : en fonction de la position du point rouge par rapport aux 3 points verts successifs de latitude supérieure illustrés sur la figure, nous distinguons 4 cas différents conduisant à une mise à jour différent du maillage. En pratique dans nos applications, ces 4 cas de figure ont été rencontrés, le 2ème étant le plus rare. Le point vert entouré devient le prochain premier point du triplé de points verts de la latitude supérieure.

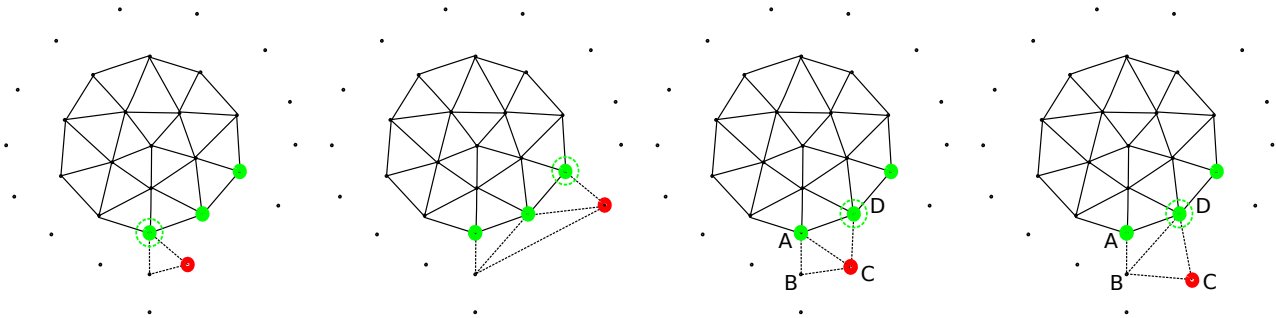


FIGURE 2.1 – Illustration des 4 cas possibles sur la construction des premiers triangles de la 3ème valeur de latitude. La différence entre les cas 3 et 4 réside dans les longueurs relatives des segments [AC] et [BD]

Par construction, la complexité algorithmique de la procédure de maillage est linéaire en N .

Caractéristiques et comparaison avec des algorithmes existants

Nous appelons “géographique” le maillage constitué des quadrilatères (projetés sur la sphère) définis par des valeurs régulières de latitude et longitude. Dans la Table 2.1, nous comparons notre algorithme avec d’autres algorithmes classiques, selon les critères suivants : l’homogénéité de la distribution de points à la surface de la sphère, l’égalité de la surface des éléments (en pratique, le bruit des instruments de mesure est intégré sur les surfaces des éléments lorsque nous représentons des champs acquis expérimentalement). Les distributions à isolatitudes permettent des traitements rapides par harmoniques sphériques. Pour des raisons pratiques ou algorithmiques, il peut être intéressant de pouvoir choisir finement le nombre de points sur la sphère. Dans notre algorithme, nous pouvons choisir n’importe quel nombre de points sur l’équateur, ce qui conduit à une certaine souplesse dans le nombre total de points N , qui n’est pas disponible dans les processus de construction de icosaèdre et HEALPix.

	notre algorithme	géographique	icosaèdre [18]	cubed sphere [98]	HEALPix [57]
distribution homogène	oui	non	oui	oui	oui
éléments d’aire égale	approché	non	approché	approché	exact
distribution isolatitude	oui	oui	non	non	oui
complexité linéaire	oui	oui	oui	oui	oui
densité quelconque	oui	oui	non	oui	non

TABLE 2.1 – Comparaison de quelques algorithmes pour calculer un maillage à la surface d’une sphère

La Figure 2.2 représente l’évolution linéaire temps de construction du maillage par rapport au nombre de points du maillage. La construction d’un maillage avec 1 million de points prend environ 1 minute.

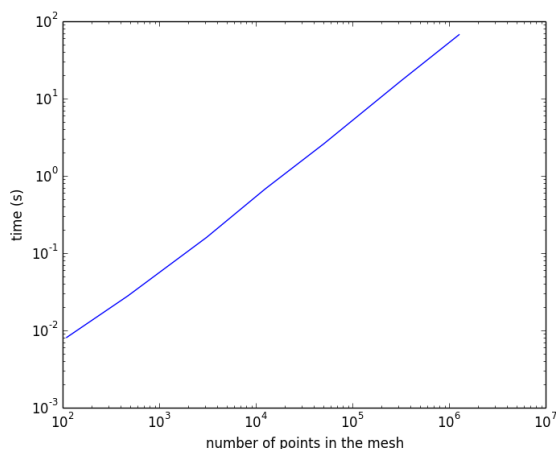


FIGURE 2.2 – Temps d’exécution en fonction du nombre de points dans le maillage.

Pour évaluer la qualité des maillages créés par notre algorithme, nous appliquons l’algorithme suivant, avec comme condition initiale le maillage obtenu par notre algorithme : nous modifions la position des points négativement le long du gradient de la 1-énergie du système de points définie par $\sum_{1 \leq i < j \leq N} \|x_i - x_j\|^{-1}$, en utilisant un schéma de Gauss-Seidel adaptatif, sans modifier la connectivité des triangles (cela revient à chercher une approximation des points de 1-Fekete sur la sphère). Notons que cette procédure étant de complexité quadratique, elle ne peut être considérée pour des données volumineuses. Ensuite, nous améliorons la qualité du maillage en considérant chaque paire de triangles et éventuellement en intervertissant l’arête partagée, pour qu’elle soit la plus petite possible, selon la Figure 2.3. En comparant le maillage créé par notre algorithme et son amélioration décrite ci-dessus, nous remarquons que l’écart-type sur la longueur des arêtes passe de 0.111 à 0.0662 (les points de Fekete cherchent les écarts de longueurs d’arêtes), mais que l’écart-type sur la surface des triangles passe de 0.0132 à 0.0244. Par ailleurs, la propriété d’isolatitude est perdue.

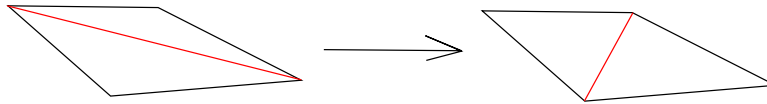


FIGURE 2.3 – Cas où l’arête partagée entre deux triangles est modifiée pour diminuer sa longueur

Ainsi, nous remarquons que notre algorithme, basé sur une heuristique simple, jouit de propriétés intéressantes de la Table 2.1, sans trop s’écarter d’une approximation de la solution optimale des points de 1-Fekete.

Un maillage volumique pour calculer les corrections topographiques

Nous disposons maintenant de données ETOPO1 d’altitudes sur un maillage géographique. Pour créer un maillage sans modifier les données, nous n’avons pas la liberté de choisir des points sur la surface de la Terre, mais nous choisissons de ne garder qu’un nombre de points donné par latitude ($\lfloor \sqrt{N\pi} \cos(\theta_i) \rfloor$), de sorte à garder une densité homogène à la surface de la sphère, voir Figure 2.4.

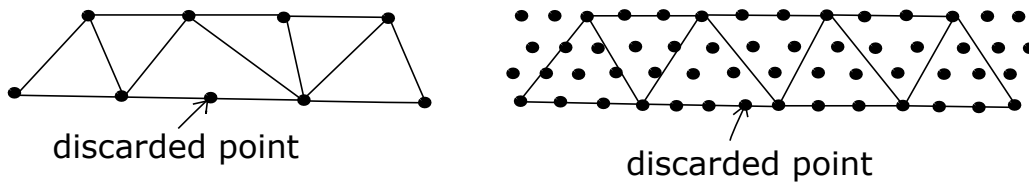


FIGURE 2.4 – Irrégularités locales introduites par la procédure ; gauche : tous les points sont gardés à l’équateur, droite : 1 points sur 3 est gardé à l’équateur

Nous expliquons maintenant comment un maillage de la différence entre la surface de la Terre et ellipsoïde de référence WGS84 est obtenu. Le maillage triangulaire obtenu est projeté d’une part sur la surface de la Terre (avec les informations d’altitude) et d’autre part sur l’ellipsoïde de référence. L’idée est de construire des tétraèdres entre ces deux surfaces : la Figure 2.5 montre les deux cas de figure avec et sans intersection (présence de côte maritime) sur la gauche, et comment les prismes à base triangulaire obtenus sont ensuite resubdivisés en tétraèdre sur la droite. Pour obtenir un maillage uniforme, des subdivisions verticales sont ajoutées dans les zones d’altitudes importantes.

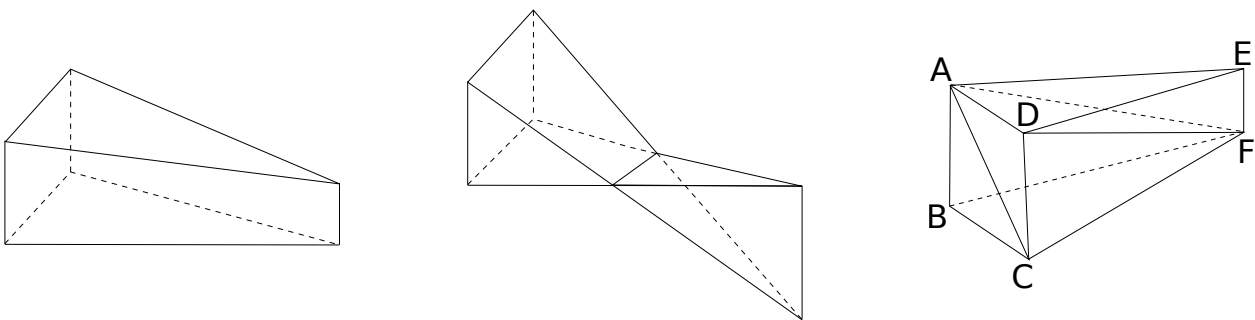


FIGURE 2.5 – Procédure de création de tétraèdres à partir du maillage triangulaire de surface ; sur la droite est indiqué comment le prisme à base triangulaire ADEFBC est subdivisé en les tétraèdres AEFD, ABCF et ACFD

L’algorithme a été appliqué aux données ETOPO1 à différentes résolutions, voir Table 2.2 pour une description des maillages obtenus et les temps d’exécution sur un processeur AMD Opteron 6276@2.3GHz. L’algorithme est implémenté en python, et pytables [7] est utilisé pour manipuler des données importantes dépassant la taille de la mémoire vive : en format hiérarchique, les formats de points, triangles et tétraèdres ont une taille respective de 9 GB, 3.4 GB et 14 GB pour la résolution à 1 minute.

Un avantage intéressant de cet algorithme est la possibilité de créer des maillages de régions locales, de nombreuses études de corrections topographiques étant réalisées pour des régions limitées. Dans la Figure 2.6

	résolution 1 min		résolution 3 min		résolution 10 min	
	temps. exec.	nombre	temps. exec.	nombre	temps. exec.	nombre
Points	5h09m07s	148'522'898	49m35s	16'491'636	25m35s	1'482'684
Mail. triang.	3h32m23s	297'002'590	24m10s	32'968'870	02m22s	2'961'046
Mail. tétra.	5h34m55s	891'845'305	49m06s	99'169'384	05m09s	8'933'982

TABLE 2.2 – Temps d'exécution et nombre de points, triangles et tétraèdres des maillages construits

sont représentées des maillages volumiques obtenus avec notre algorithme, où, pour des raisons de représentation, les altitudes ont été multipliées par 5 sur la péninsule indienne, et par 10 sur la France.

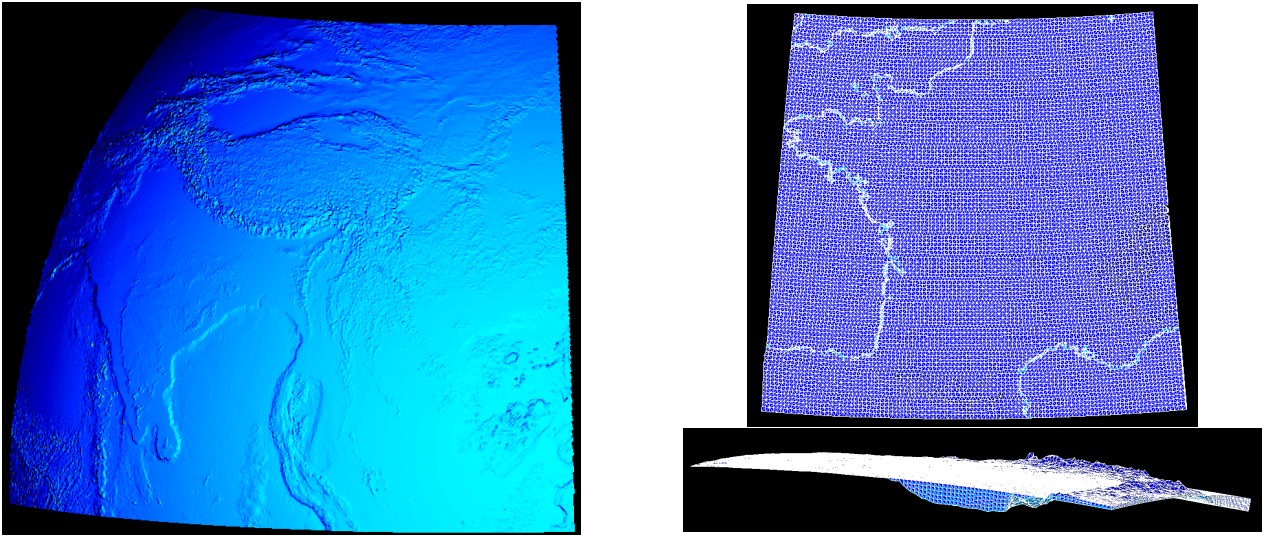


FIGURE 2.6 – Gauche : maillage topographique pour la région de latitude $0^\circ \leq \theta \leq 50^\circ$ et longitude $60^\circ \leq \phi \leq 120^\circ$, avec une résolution de 10 min : droite maillage grossier de la France (région de latitude $42^\circ \leq \theta \leq 51^\circ$ et longitude $-5^\circ \leq \phi \leq 8^\circ$) pour identifier les tétraèdres, notamment les traitements aux côtes. Les effets topographiques sont respectivement amplifiés par des facteurs 5 et 10.

2.2 Couplage BEM/FEM pour les équations de gravitation

Considérons le problème de calcul du potentiel de gravitation suivant :

$$\left\{ \begin{array}{ll} \Delta\Phi = 1 & \text{dans la Terre : } \Omega^- \\ \Delta\Phi = 0 & \text{extérieur de la Terre : } \Omega^+ \\ [\gamma_0\Phi]_\Gamma = 0 & \text{continuité du potentiel} \\ [\gamma_1\Phi]_\Gamma = 0 & \text{continuité de la dérivée normale du potentiel} \\ \Phi(|x|) \rightarrow 0 & \text{décroissance à l'infini} \\ |x| \rightarrow \infty & \end{array} \right. \quad (2.1)$$

où $[\gamma_0\Phi]_\Gamma$ et $[\gamma_1\Phi]_\Gamma$ représentent le saut des traces de Dirichlet et de Neumann à travers $\Gamma := \partial\Omega^-$, la surface de la Terre.

A l'instar de la méthode proposée dans le Chapitre 1, nous proposons une méthode de couplage fort pour les inconnues surfaciques et volumiques. La formulation variationnelle obtenue est : Trouver $(\Phi, \lambda) \in H^1(\Omega^-) \times$

$H^{-\frac{1}{2}}(\Gamma)$ tel que pour tout $(\Phi^t, \lambda^t) \in H^1(\Omega^-) \times H^{-\frac{1}{2}}(\Gamma)$,

$$\begin{cases} \iint_{\Omega^-} \vec{\nabla} \Phi \cdot \vec{\nabla} \Phi^t + (N(\gamma_0^- \Phi), \gamma_0^- \Phi^t)_\Gamma + \left(\left(\tilde{D} - \frac{1}{2} I \right) (\lambda), \gamma_0^- \Phi^t \right)_\Gamma = - \int_{\Omega^-} \Phi^t, \\ \left(\left(D - \frac{1}{2} I \right) (\gamma_0^- \Phi), \lambda^t \right)_\Gamma - (S(\lambda), \lambda^t)_\Gamma = 0, \end{cases} \quad (2.2)$$

où nous rappelons que S , D , \tilde{D} et N sont respectivement les opérateurs intégraux de simple couche, double couche, transposé de l'opérateur double couche et hypersingulier associés au noyau de Laplace et où γ_0^- est la trace intérieure de Dirichlet sur Γ et $(\cdot, \cdot)_\Gamma$ est l'extension du produit scalaire dans $L^2(\Gamma)$ au produit de dualité $H^{-\frac{1}{2}}(\Gamma) \times H^{\frac{1}{2}}(\Gamma)$. Une approximation en dimension finie de (2.2) est introduite de la même façon que pour (1.11). Une approximation de la solution de (2.1) est ensuite obtenue par :

$$\begin{cases} \Phi & \text{dans } \Omega^- \cup \Gamma \\ (-S(\lambda) + \mathcal{D}(\gamma_0^- \Phi))|_{\Omega^+} & \text{dans } \Omega^+, \end{cases} \quad (2.3)$$

où \mathcal{S} and \mathcal{D} sont les potentiels de simple couche et de double couche associés au noyau de Laplace.

Dans le code industriel ACTIPOLE, une routine a été ajoutée pour prendre en compte un second membre non nul pour les inconnues volumiques et les routines d'assemblages des opérateurs intégraux ont été modifiées pour traiter le cas du noyau de Laplace. Ces développements ont été testés sur les applications numériques suivantes :

1. boule :

Pour une boule de rayon 1, la solution analytique est radiale, telle que $\Phi(r) = -0.5 + \frac{r^2}{6}$ pour $r \leq 1$ et $\Phi(r) = -\frac{1}{3r}$ pour $r > 1$. Une comparaison entre une solution calculée par ACTIPOLE et la solution analytique est proposée dans la Figure 3.17. Nous remarquons un écart satisfaisant, étant donné le niveau de discrétisation du maillage.

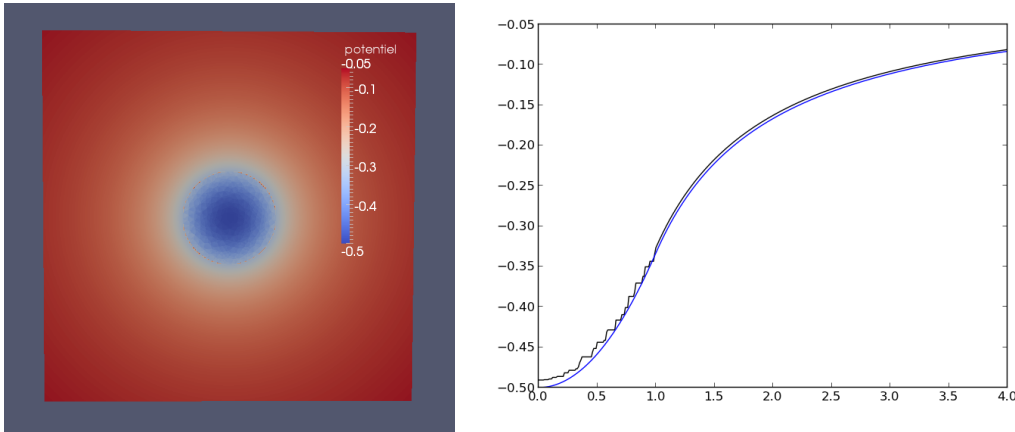


FIGURE 2.7 – Gauche : potentiel de gravitation dans le cas de la boule ; droite : coupe du potentiel de gravitation le long d'un rayon de la boule - noir : solution calculée, bleu : solution analytique.

2. modèle simplifié d'un massif montagneux

Dans cette application, nous considérons un modèle simplifié de massif montagneux, voir Figure 2.8. Nous traçons le potentiel de gravitation le long d'une ligne qui pénètre et ressort du massif deux fois. La solution est régulière à l'extérieur du massif, par la régularisation du post-traitement dans le domaine extérieur, mais moins régulière à l'intérieur, où la qualité de discrétisation du maillage a plus d'influence, voir (2.3).

3. correction topographique

En utilisant des maillages (relativement grossiers) générés par la stratégie présentée dans la Section 2.1, nous obtenons les résultats de la Figure 2.9.

L'outil développé offre une nouvelle technique du calcul du modèle du champ de gravitation avec les avantages suivants par rapport aux méthodes classiques en géosciences (par ex. les développements en harmoniques

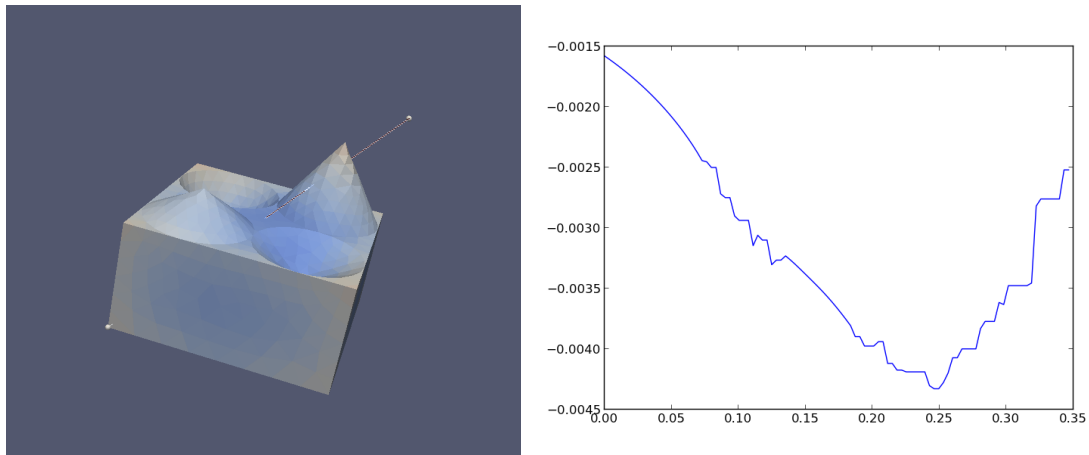


FIGURE 2.8 – Gauche : potentiel de gravitation sur un massif montagneux et ligne de coupe ; droite : potentiel de gravitation le long de la ligne de coupe.

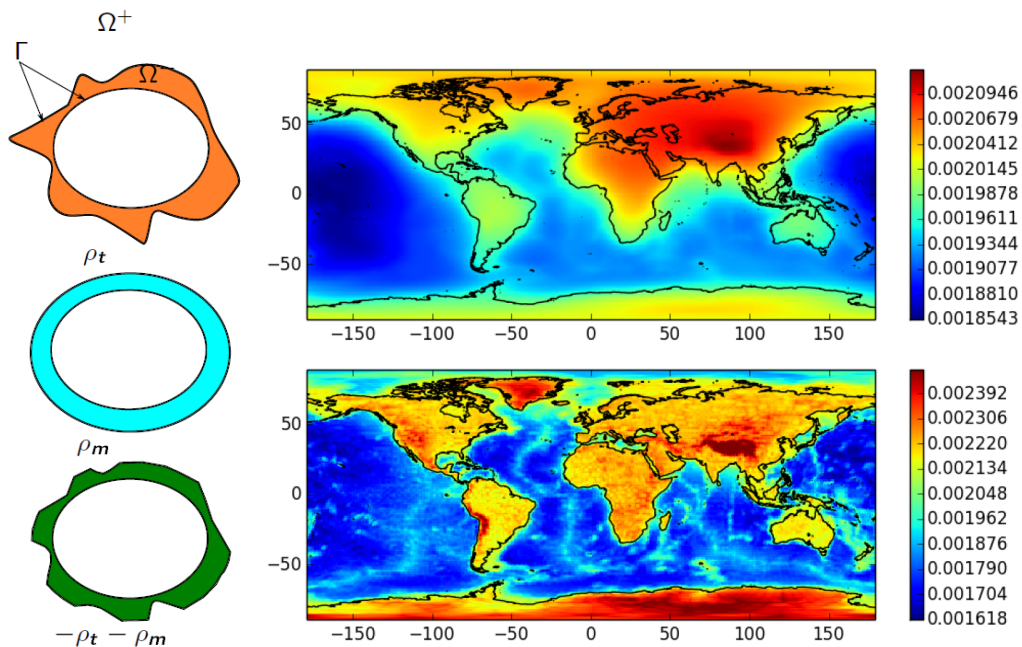


FIGURE 2.9 – Gauche : les trois maillages considérés, avec les densités (la correction topographique est la somme des calculs sur ces trois géométries) ; droite : potentiel de gravitation et sa dérivée normale, à 10 km d'altitude

sphériques) : (i) géométrie quelconque, non limitée à la sphère ou l'ellipsoïde de révolution, (ii) possibilité d'augmenter localement la résolution sans impacter fortement le coût de calcul, (iii) ajouter des informations de mesures de sources diverses sur des maillages hétérogènes, tout en prenant en compte la topographie.

Le développement de ce type de modèle peut accompagner la formulation des problèmes inverses (au sens très général) en sismologie et en gravimétrie, c'est-à-dire pour la détermination de la structure interne de la Terre en paramètres élastiques, viscosité et densité. Dans ce cadre, ces travaux pourraient contribuer aux recherches sur la dynamique globale de la Terre solide et sur l'évolution climatique.

Une perspective de ce travail serait de rendre la procédure compatible avec les solveurs performants d'ACTI-POLE (Mumps pour la condensation du problème interne et la FMM pour le problème surfacique) et la tester sur des cas de grande échelle. Une autre perspective consiste à développer ces approches de couplage éléments de frontière / éléments finis pour les équations gravito-visco-élastiques (couplage entre l'équation de Helmholtz à basses fréquences et l'équation de Poisson, pour des rhéologies visco-élastiques) : sans les constantes physiques

et dans son état le plus simple, le système à résoudre s'écrit

$$\left\{ \begin{array}{ll} \Delta \vec{u} + \vec{\nabla} \Phi = \vec{1} & \text{dans la Terre : } \Omega^- \\ \Delta \Phi + \vec{\nabla} \cdot \vec{u} = 0 & \text{dans la Terre : } \Omega^- \\ \Delta \Phi = 0 & \text{exterieur de la Terre : } \Omega^+ \\ [\gamma_0 \Phi]_\Gamma = 0 \\ [\gamma_1 \Phi]_\Gamma = \vec{n} \cdot \vec{u}^-|_\Gamma \\ \vec{n} \cdot \vec{\nabla} u_i^-|_\Gamma = 0 & 1 \leq i \leq 3 \\ \Phi(|x|) \rightarrow 0 & |x| \rightarrow \infty \end{array} \right. \quad (2.4)$$

où u est le déplacement, et qui admet comme formulation variationnelle :

Trouver $(u, \Phi, \lambda) \in H^1(\Omega^-)^3 \times H^1(\Omega^-) \times H^{-\frac{1}{2}}(\Gamma)$ tel que pour tout $(u^t, \Phi^t, \lambda^t) \in H^1(\Omega^-)^3 \times H^1(\Omega^-) \times H^{-\frac{1}{2}}(\Gamma)$,

$$\left\{ \begin{array}{l} -\sum_{i=0}^3 \int_{\Omega^-} \vec{\nabla} u_i \cdot \vec{\nabla} u_i^t + \int_{\Omega^-} \vec{\nabla} \Phi \cdot \vec{u}^t = \int_{\Omega^-} \vec{1} \cdot \vec{u}^t, \\ \int_{\Omega^-} \left(\vec{\nabla} \Phi \cdot \vec{\nabla} \Phi^t + \vec{u} \cdot \vec{\nabla} \Phi^t \right) + (N(\gamma_0^- \Phi), \gamma_0^- \Phi^t)_\Gamma + \left(\left(\tilde{D} - \frac{1}{2} I \right) (\lambda), \gamma_0^- \Phi^t \right)_\Gamma = 0, \\ \left(\left(D - \frac{1}{2} I \right) (\gamma_0^- \Phi), \lambda^t \right)_\Gamma - (S(\lambda), \lambda^t)_\Gamma = 0. \end{array} \right. \quad (2.5)$$

2.3 Calcul rapide de champs de gravitations par Fast Multipole Method

Cette section correspond aux Publications [Pub.13](#) et [Pub.14](#) et aux Exposés [Exp.7](#) à [Exp.9](#) et [Exp.20](#).

Comme évoqué en introduction du manuscrit, de nombreuses applications en géosciences nécessitent de nettoyer les champs de gravitation mesurés. Dans la Section 2.1, nous avons présenté un algorithme pour obtenir un maillage tétraédrique pour une densité quelconque de matière. Une fois ce maillage obtenu, et avec une densité associée à chaque tétraèdre, il nous faut une méthode efficace de calcul du champs de gravitation, et selon les applications, de ses dérivées première et seconde.

Dans cette section, nous présentons une procédure numérique efficace pour le calcul des corrections de champs de gravitation, reposant sur une méthode d'intégration accélérée par FMM, et avec des applications au calcul des corrections topographiques, mais aussi une correction due aux masses d'eau infiltrées en milieu forestier.

Le potentiel de gravitation généré par une densité de masse ρ est donné par l'intégrale de Newton

$$\phi(x) := G \int_{y \in \mathcal{V}} \frac{\rho(y)}{\|x - y\|} dV(y), \quad (2.6)$$

où G est la constante gravitationnelle et dV est l'élément de volume. Les dérivées première et seconde de ϕ s'écrivent respectivement

$$g_i(x) := \frac{\partial \phi}{\partial x_i}(x) = G \int_{y \in \mathcal{V}} \rho(y) \frac{y_i - x_i}{\|x - y\|^3} dV(y), \quad 1 \leq i \leq 3, \quad (2.7)$$

et

$$T_{i,j}(x) := \frac{\partial^2 \phi}{\partial x_i \partial x_j}(x) = G \int_{y \in \mathcal{V}} \rho(y) \left(\frac{3(x_i - y_i)(x_j - y_j)}{\|x - y\|^5} - \frac{\delta_{i,j}}{\|x - y\|^3} \right) dV(y), \quad 1 \leq i, j \leq 3, \quad (2.8)$$

où $\delta_{i,j}$ est le symbole de Kronecker. De nombreuses méthodes ont été proposées pour calculer ces expressions, notamment des méthodes d'intégration analytiques sur des polyèdres [94], des transformations de Fourier rapides [23] et des méthodes utilisant des tesseroïdes [61] (restreintes à des géométries sphériques et ellipsoïdales). Nous renvoyons vers [Pub.14](#) pour une liste étendue de références.

Il n'existe pas de méthode analytique pour calculer (2.7) pour des distributions de masse quelconque. Dans un premier temps, la distribution de masse est approchée par un maillage tétraédrique comme présenté dans la Section 2.1, et l'équation (2.7) devient

$$g_i(x) = G \sum_{k=1}^N \int_{y \in \mathcal{V}_k} \rho(y) \frac{y_i - x_i}{\|x - y\|^3} dV(y), \quad 1 \leq i \leq 3, \quad (2.9)$$

où $\{\mathcal{V}_k\}_{1 \leq k \leq N}$ désigne l'ensemble des tétraèdres du maillage (des relations équivalentes sont obtenues pour (2.6) et (2.8)). Une fois exprimées sur un maillage tétraédrique, ces intégrales peuvent être calculées analytiquement ; nous considérons en particulier les formules [94, Équation (31) et (51)] respectivement pour le calcul du potentiel de gravitation et de ses dérivées première et seconde. La formule analytique pour calculer \vec{g} est appelée ici formule de Okabe. Une autre façon de calculer ces intégrales est d'utiliser des schémas d'intégration numérique appelés formules de quadrature. Le calcul de \vec{g} du à l'influence du tétraèdre \mathcal{V}_k est approché par

$$g_i(x)_{\mathcal{V}_k} \approx G \sum_{s=1}^S \omega_{\mathcal{V}_k, s} \rho(y_{\mathcal{V}_k, s}) \frac{(y_{\mathcal{V}_k, s})_i - x_i}{\|x - y_{\mathcal{V}_k, s}\|^3} v(\mathcal{V}_k), \quad 1 \leq i \leq 3, \quad (2.10)$$

où S est l'ordre de la quadrature, et $\{\omega_{\mathcal{V}_k, s}\}_{1 \leq s \leq S}$, $\{y_{\mathcal{V}_k, s}\}_{1 \leq s \leq S}$ sont les poids et points de quadrature associés au tétraèdre \mathcal{V}_k , et $v(\mathcal{V}_k)$ est le volume de \mathcal{V}_k .

En terme de précision, les formules de Okabe et par quadrature ont un comportement différent : la formule de Okabe est précise proche des sources (tétraèdres) et manque de précision loin des sources ([77, Chapitre 5] : accumulation des erreurs d'arrondi machine), tandis que les quadratures ont un comportement opposé (du à la singularité pour $x = y_{\mathcal{V}_k, s}$ dans (2.10)). Nous évaluons sur la Figure 2.10 la différence relative suivante sur un ensemble de points d'observation à diverses distances de tétraèdres :

$$\mathcal{E}_r := \frac{\sqrt{\sum_i \|\vec{g}_q(x_{i,r}) - \vec{g}_O(x_{i,r})\|_2^2}}{\sqrt{\sum_i \|\vec{g}_O(x_{i,r})\|_2^2}}, \quad (2.11)$$

où $\|\cdot\|_2$ est la norme euclidienne dans \mathbb{R}^3 , et $\vec{g}_O(x_{i,r})$ et $\vec{g}_q(x_{i,r})$ correspondent à des évaluations de \vec{g} avec respectivement les formules d'Okabe et de quadrature.

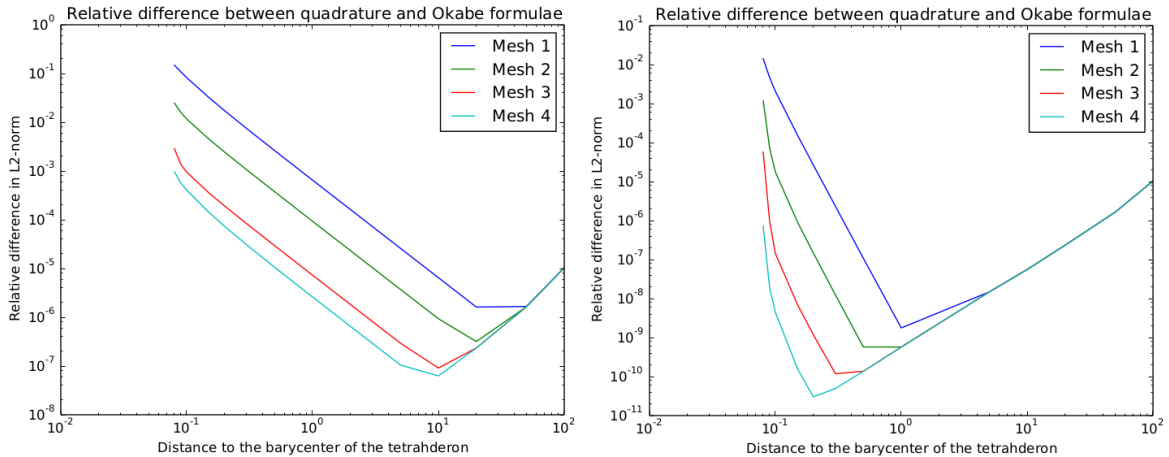


FIGURE 2.10 – \mathcal{E}_r (2.11) pour diverses valeurs de r/c (ratio d'aspect entre la distance au barycentre du tétraèdre et l'arête moyenne) ; gauche : avec une quadrature à 1 point, droite : avec un quadrature à 15 points

L'idée est de considérer, pour chaque point d'observation, deux ensembles de tétraèdres : ceux en interaction proche, pour lesquels les effets gravitationnels seront calculés par la formule de Okabe, et ceux en interaction lointaine, pour lesquels les formules de quadrature seront utilisées de sorte que chaque formule soit appliquée dans le régime où elle est précise. Enfin, les formules de quadrature seront accélérées par Fast Multipole Method (FMM), rendant la complexité algorithmique de la procédure complète linéaire par rapport au nombre de points d'observation et aux nombre de tétraèdres, rendant possibles les calculs de très grande échelle.

Couplage formule analytique de Okabe et quadratures accélérées par FMM

Dans l'algorithme FMM, la notion d'interactions proches et lointaines au sens de l'arbre a été introduite : un couple tétraèdre et point d'observation sont considérés en interaction proche s'ils sont dans la même boîte ou des boîtes adjacentes, et en interaction lointaine sinon. Ainsi, les interactions lointaines peuvent être accélérées par FMM car de la forme (7), alors que nous pouvons utiliser l'algorithme de notre choix pour les interactions proches : dans notre cas la formule Okabe. A données fixées (maillage tétraédrique de la densité de matière et ensemble de points d'observations), le ratio d'aspect doit être choisi de sorte que chacune des formules soit utilisée dans le régime où elle est précise, et la distance séparant ainsi les interactions proches et lointaines va directement déterminer la hauteur de l'arbre. D'autres paramètres sont également à choisir : le nombre de nœuds de Chebyshev par dimension dans l'approximation lointaine du noyau et le nombre de points de quadratures. Une campagne numérique a été réalisée pour déterminer les paramètres donnant l'algorithme le plus rapide garantissant une précision relative de 2×10^{-4} sur le résultat du calcul, voir Figure 2.11. Les paramètres choisis

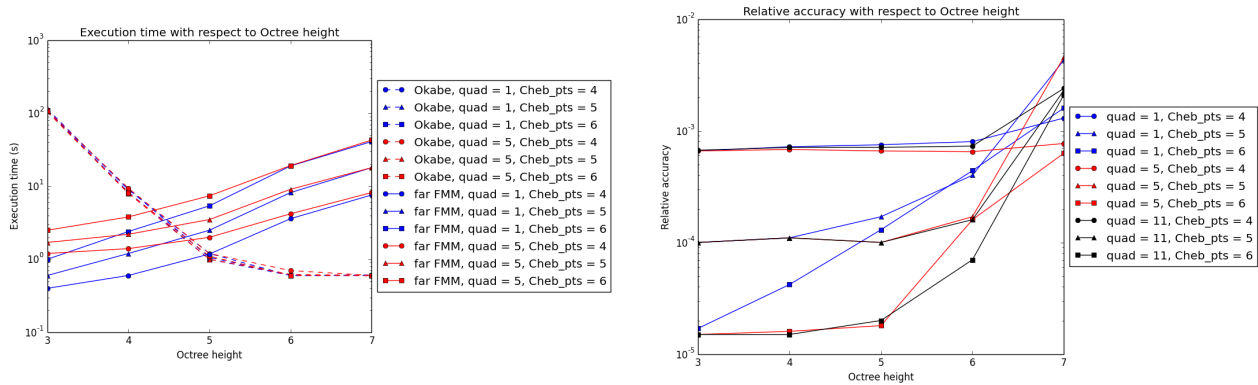


FIGURE 2.11 – Gauche : temps de calcul des interaction proches (Okabe) et lointaines (far FMM) en fonction de la hauteur de l'arbre pour 1 et 5 points de quadrature, et 4, 5 et 6 nœuds de Chebyshev par dimension ; droite : précision relative en fonction de la hauteur de l'arbre pour 1, 5 et 11 points de quadrature et 4, 5 et 6 nœuds de Chebyshev par dimension. Le calcul correspond à 1,000 points d'observation autour d'un cube discrétisé en 13,712 tétraèdres, dont la solution analytique est connue.

sont : une quadrature à 1 point, 5 nœuds de Chebyshev et un ratio d'aspect de 2.

Considérons la Figure 2.12 : en comparant la solution analytique connue de la boule avec notre procédure pour des jeux de paramètres conduisant à une précision de 10^{-4} (dans l'étude présentée dans la Figure 2.11), nous vérifions que les 4 maillages proposés conduisent à une précision relative de respectivement 10^{-1} , 2×10^{-2} , 3×10^{-3} et 10^{-3} . Ainsi, il n'apparaît pas utile de chercher une précision relative meilleure que 2×10^{-4} pour notre procédure, dans la mesure où l'approximation de la boule par des tétraèdres conduit à une erreur de modèle supérieure, même dans le cas du 4ème maillage précis de la Figure 2.12.

Pour évaluer les performances de notre procédure et son extensibilité, nous considérons 6 maillages différents d'un cube, de 10^3 à 10^7 tétraèdres, sur 32 cœurs répartis sur 2 processeurs AMD Opteron 6276@2.3 GHz, voir Figure 2.13. Dans chaque cas, le nombre de points d'observation est égal au nombre de tétraèdres. Le cas non accéléré correspond à la formule d'Okabe (extrapolée), pour laquelle nous savons que la précision est mauvaise pour les interactions lointaines. La complexité algorithmique de notre procédure est bien linéaire à partir de 10^5 tétraèdres. Le calcul à 10 millions de tétraèdres et points d'observation prend 47 minutes, alors que le calcul non accéléré prendrait plus de 3 ans. Un calcul à 100 millions de tétraèdres et 2 millions de points d'observation a été réalisé sur ce cluster. Les calculs plus gros sont possibles à condition d'augmenter les caractéristiques matérielles (plus de mémoire vive ou plus de nœuds de calcul pour partager les exigences en mémoire).

Applications numériques

De nombreuses applications numériques sont présentées dans Pub.13 et Pub.14 : des corrections topographiques dans les régions de l'Himalaya, de la région parisienne, de la France et de la Terre entière à l'altitude de mesure du satellite GOCE. Pour illustrer la capacité supplémentaire par rapport aux outils classiques en géodésie à traiter des géométries non sphériques ni ellipsoïdales, la dérivée première du champ de gravitation autour de la comète 67P/Churyumov–Gerasimenko est montrée, ainsi que les effets gravitationnels de masses

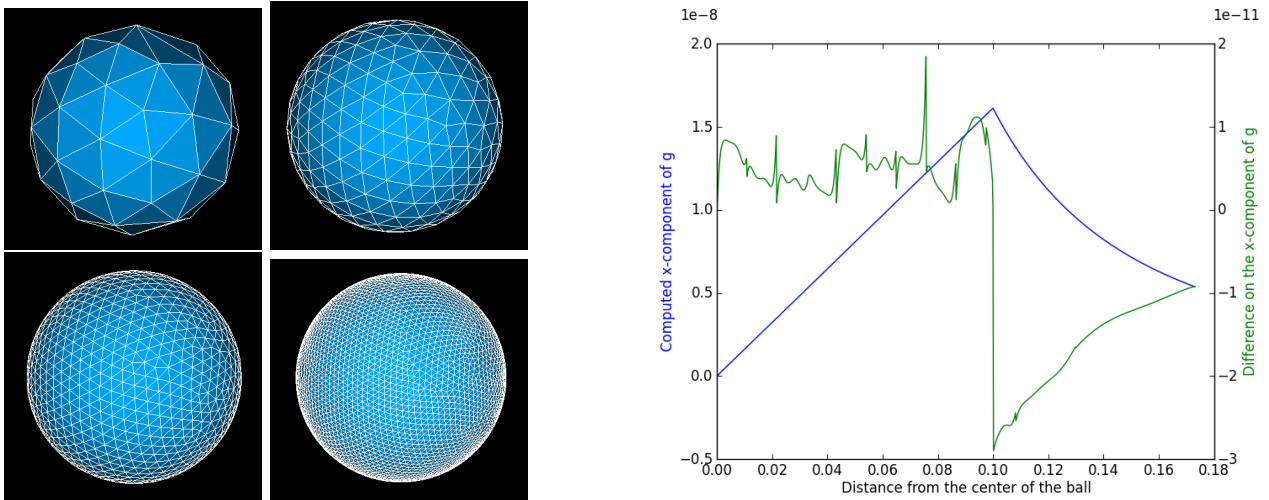


FIGURE 2.12 – Gauche : 4 maillages de précision croissante de la boule ; droite : solution analytique et différence entre la solution analytique et calculée par notre procédure pour la composante x de \vec{g} le long d'un rayon du maillage le plus précis de la boule, pour un ratio d'aspect de 2, 5 nœuds de Chebyshev par dimension et 15 points de quadrature.

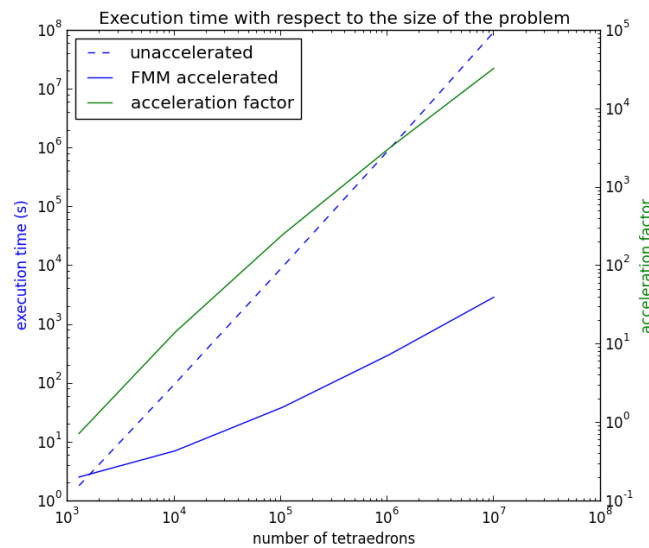


FIGURE 2.13 – Temps d'exécution pour les calculs non-accélérés et accélérés par FMM et facteur d'accélération (sur 32 cœurs)

d'eau infiltrées en milieu forestier. Par souci de concision du présent document, nous ne montrons ici que les résultats sur l'Himalaya et sur la Terre entière.

Le maillage utilisé pour le calcul autour de l'Himalaya a été présenté dans la Figure 2.6, qui correspond à la différence entre la surface de la Terre et l'ellipsoïde de référence WGS84 dans cette région. Ce maillage est composé de 1.2×10^7 tétraèdres, et le calcul est réalisé pour 10^6 points d'observation, localisé 1 cm au dessus des points topographiques. L'arbre a une hauteur de 11 et 1,327,186 feuilles. Le calcul prend moins de 20 minutes pour les trois composantes de \vec{g} sur 16 cœurs (AMD Opteron 6276@2.3 GHz). La correction topographique dans cette région est représentée sur la Figure 2.14

La Figure 2.15 montre indépendamment les interactions proches et lointaines de la correction topographique sur l'Himalaya, au sens de notre procédure. Nous remarquons que les deux composantes contiennent un signal physique, et que les interactions lointaines ne peuvent être simplement ignorées, car dans notre cas, le ratio d'aspect a la valeur faible de 2 : un point situé à plus de 2 fois l'arête moyenne du barycentre d'un tétraèdre ne

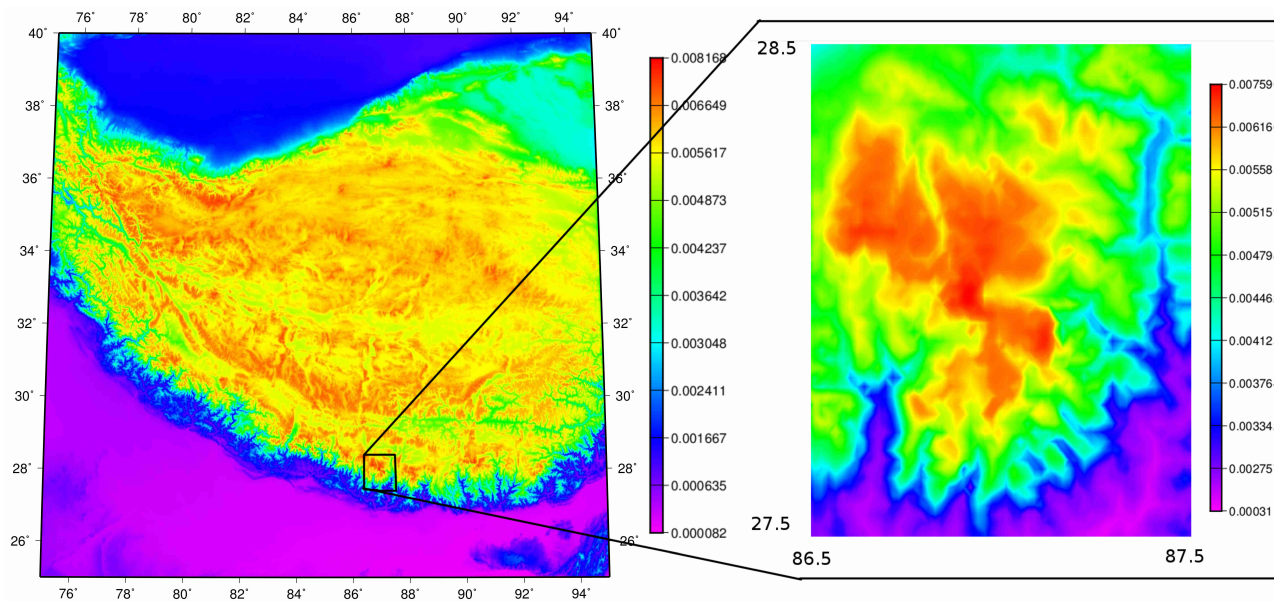


FIGURE 2.14 – Correction topographique sur l’Himalaya, avec un zoom sur le mont Everest : composante normale de \vec{g} générée par la différence entre la Terre et l’ellipsoïde de référence WGS84 (en ms^{-2})

peut ignorer ce tétraèdre.

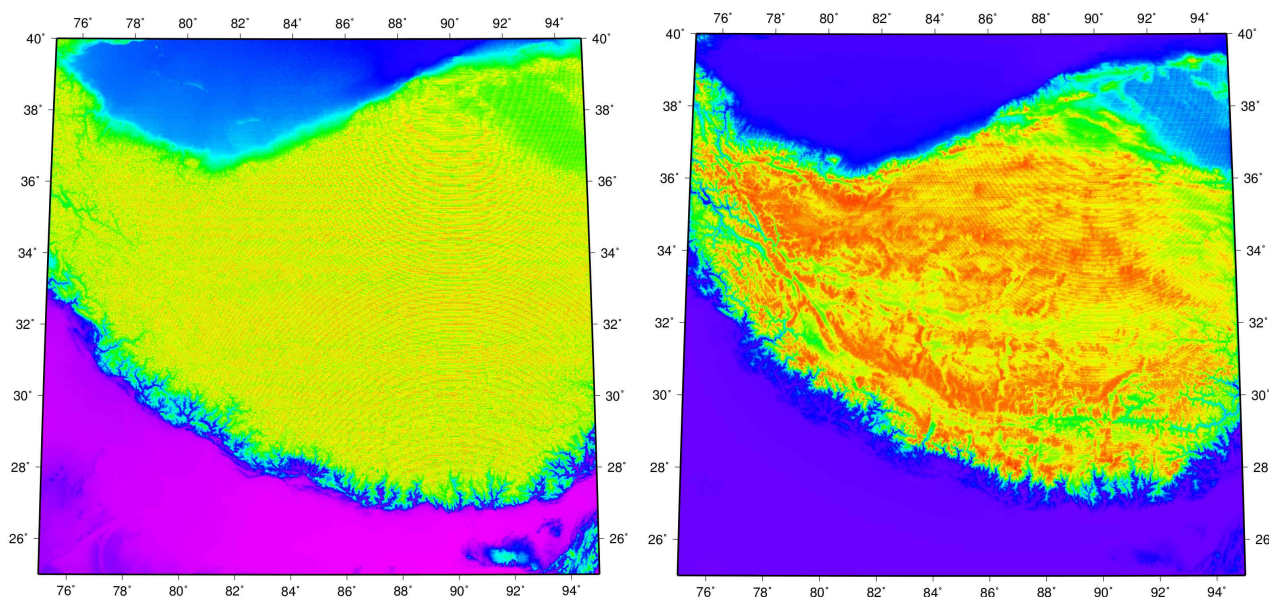


FIGURE 2.15 – Correction topographique sur l’Himalaya ; gauche : interactions proches ; droite : interactions lointaines

Le maillage utilisé pour le calcul de la correction topographique sur la Terre entière est composé de 10^8 tétraèdres, 3.3×10^7 triangles et 1.6×10^7 sommets, et le calcul est réalisé pour 2×10^6 points d’observation. L’arbre a une hauteur de 10, et le calcul prend 6m 30s pour ϕ et chaque composante de \vec{g} et T sur 16 cœurs (AMD Opteron 6276@2.3 GHz). La correction topographique est représentée sur la Figure 2.16 pour ϕ et les composantes de \vec{g} et T à une altitude de 250km, et pour g_r et T_{rr} à une altitude de 20km.

Pour montrer un intérêt supplémentaire de notre procédure par rapport aux procédures classiques en géodésie, nous représentons l’anomalie de gravitation d’air libre, calculée avec le logiciel SHTOOLS [116], sur

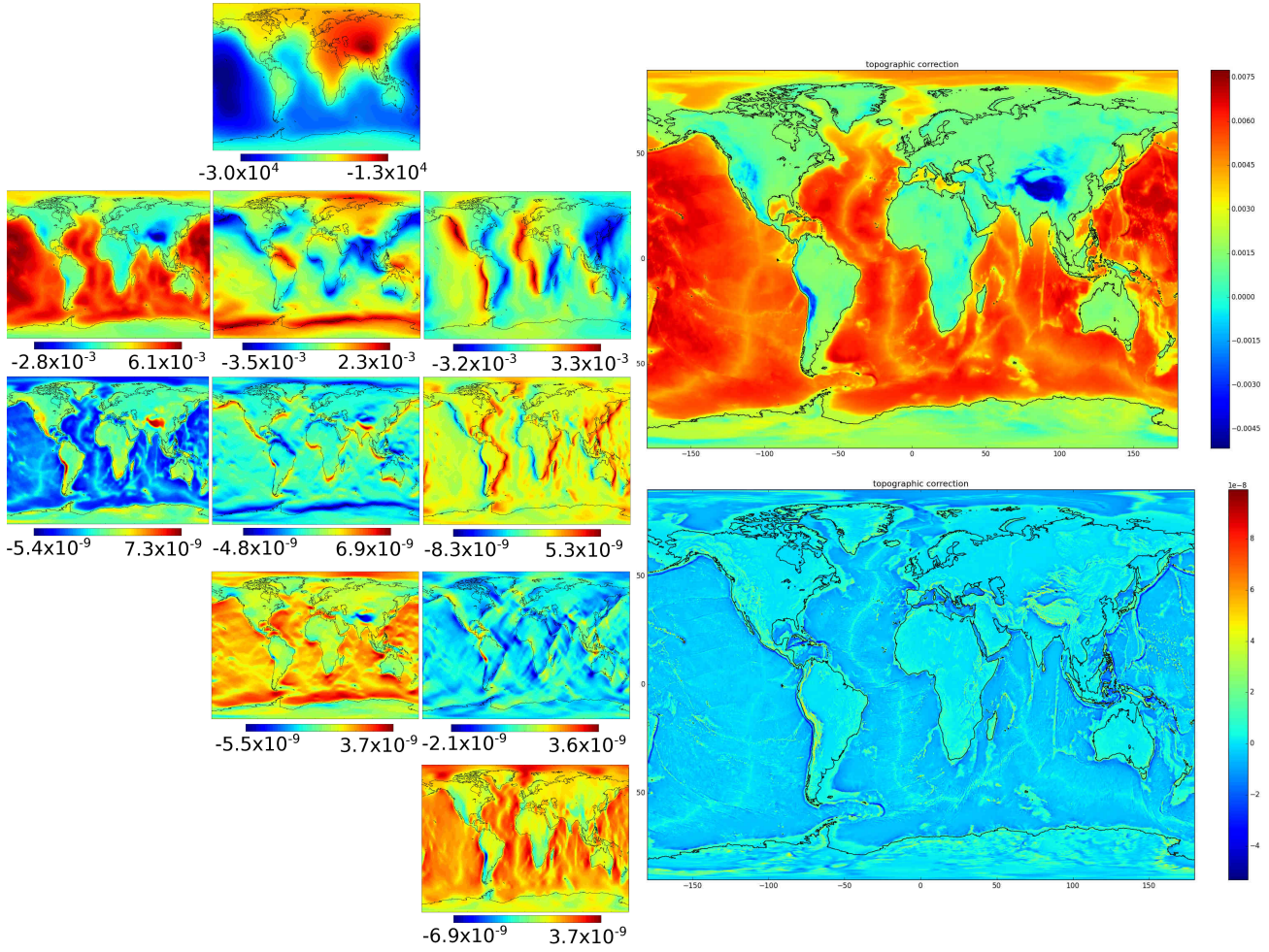


FIGURE 2.16 – Correction topographique sur la Terre entière ; gauche (à 250km) 1.1 : ϕ (en m^2s^{-2}), 1.2 : g_r, g_θ, g_ϕ (en ms^{-2}), 1.3 : $T_{rr}, T_{r\theta}, T_{r\phi}$, 1.4 : $T_{\theta\theta}, T_{\theta\phi}$, 1.5 : $T_{\phi\phi}$ (en s^{-2}) ; droite (à 20km) 1.1 g_r (en ms^{-2}), 1.2 : T_{rr} (en s^{-2}).

la Figure 2.17, sur une base d'harmoniques sphériques de degré 1500 : un phénomène de Gibbs est observé, ce qui dégrade très fortement la précision de la représentation du champ. Pour représenter des structures plus petites, il faudrait encore augmenter l'ordre des harmoniques sphériques, ce qui amplifiera encore le phénomène de Gibbs. En revanche, notre procédure peut manipuler des données de très grande échelle sans sacrifier la qualité des résultats, puisqu'elle a été construite pour une précision relative de 2×10^{-4} , qui est déjà plus faible que l'erreur de modèle du maillage.

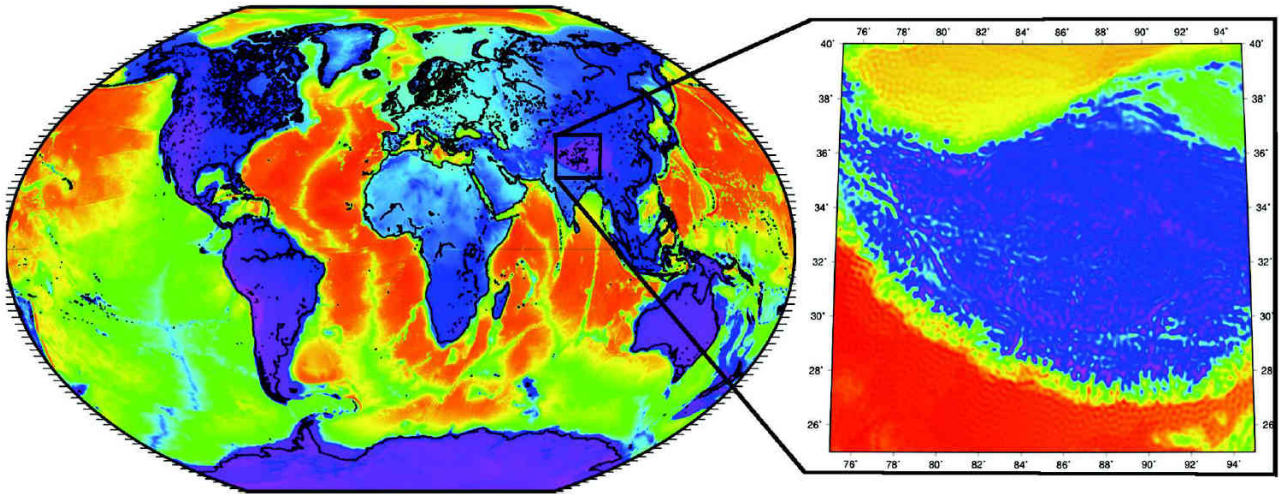


FIGURE 2.17 – Illustration du phénomène de Gibbs avec une représentation de l’anomalie d’air libre sur une expansion en harmoniques sphériques de degré 1500 sur l’ellipsoïde de référence WGS84, calcul par L. Métivier

2.4 Une méthode de sommation rapide et adaptative pour noyaux invariants par translation

Cette section correspond à la Publication [Pub.2](#) et aux Exposés [Exp.7](#) et [Exp.8](#)

Dans cette section, nous combinons l’Empirical Interpolation Method et la Fast Multipole Method pour proposer une méthode de sommation rapide pour noyaux invariants par translation. L’originalité repose sur l’approximation des interactions lointaines, qui se fait non pas sur une base d’approximation choisie a priori, mais adaptée à chaque nouveau noyau considéré. L’EIM n’est pas utilisé dans sa version classique, mais une version rapide (cependant sous-optimale en terme d’approximation) telle que dans chaque étape de l’algorithme glouton, les données sont parcourues de façon parcimonieuse. Pour les noyaux inhomogènes, où le problème d’approximation est différent d’un niveau à l’autre dans l’arbre, l’EIM va générer des approximations avec un nombre adapté de points d’interpolation, ce qui conduit à des gains importants par rapport aux méthodes classiques.

Une approximation compatible avec la FMM

Nous présentons la méthode en $1D$, mais elle est facilement extensible en $3D$ – les simulations numériques présentées sont en $3D$. Considérons un noyau $K(x, y)$, $(x, y) \in \Omega \times \Omega$, où $\Omega = (-\frac{L}{2}, \frac{L}{2})$ est le domaine complet où la FMM sera calculée. Le domaine Ω est partitionné en 2^κ intervalles, $\kappa \in \mathbb{N}^*$, de longueur $2l_\kappa$, avec $l_\kappa := 2^{-\kappa-1}L$. Ce noyau est supposé invariant par translation : $K(x, y) = K(x - a, y - a)$, $\forall a \in \mathbb{R}$. Considérons deux intervalles I et J , bien séparés dans le sens où $\min_{(x,y) \in I \times J} |x - y| \geq 2l_\kappa$. Pour généraliser aux dimensions supérieures, I et J sont appelées “boîtes” dans la suite ; c_I et c_J sont les centres des boîtes I et J . L’approximation EIM du noyau sur $I \times J$ s’écrit :

$$K(x, y) \approx \sum_{l,m=1}^d \Delta_{l,m}^{I,J} K(x, y_l^J) K(x_m^I, y), \quad x \in I, \quad y \in J, \quad (2.12)$$

où les exposants I et J indiquent les boîtes dont dépendent les quantités considérées. Par exemple, $x_m^I \in I$ et $y_l^J \in J$. Selon la Remarque [0.1](#), cette approximation EIM du noyau n’est pas compatible avec une FMM multinationale.

L’idée est d’utiliser l’invariance par translation du noyau et d’étendre le domaine de définition d’une des deux variables du noyau dans l’approximation des interactions lointaines. Considérons $I_0 := \{x - c_j, x \in \hat{I}\}$ pour toute boîte \hat{J} dans le partitionnement de Ω et toute boîte \hat{I} bien séparée de \hat{J} , et définissons J_0 la boîte de longueur $2l_\kappa$ localisée au centre de Ω (notons que J_0 n’est pas une boîte du partitionnement de Ω). Dans notre cas, $J_0 = (-l_\kappa, l_\kappa)$ et $I_0 = (-L + l_\kappa, -3l_\kappa) \cup (3l_\kappa, L - l_\kappa)$, voir Figure [2.18](#). Considérons un EIM sur $I_0 \times J_0$ et

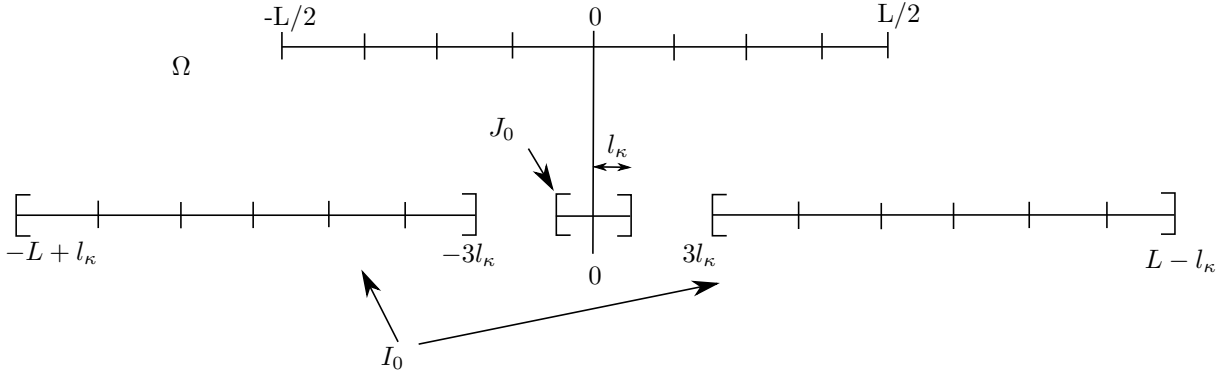


FIGURE 2.18 – Représentation du partitionnement de Ω en 1D pour $\kappa = 3$ et des boîtes correspondantes I_0 et J_0 .

un sur $J_0 \times I_0$, en utilisant l'invariance par translation de K , on peut montrer la relation suivante

$$K(x, y) \approx \sum_{l'=1}^d K(x - c_I, y_{l'}^{I_0}) \sum_{m'=1}^d \Delta_{l', m'}^{J_0, I_0} \sum_{l=1}^d K(x_{m'}^{J_0}, y_l^{J_0} + c_J - c_I) \sum_{m=1}^d \Delta_{l, m}^{I_0, J_0} K(x_m^{I_0}, y - c_J), x \in I, y \in J. \quad (2.13)$$

où les dépendances des quantités en certains domaines sont à nouveau indiquées par des exposants. Pour les noyaux symétriques, une seule approximation EIM est utilisée. En particulier, le terme dépendant de x ($K(x - c_I, y_{l'}^{I_0})$) est indépendant de la boîte J , et le terme dépendant de y ($K(x_m^{I_0}, y - c_J)$) est indépendant de la boîte I . La formule (2.13) fait maintenant intervenir deux approximations EIM et quatre sommes, mais est maintenant compatible avec une FMM multiniveaux.

Tous les détails sur les manipulations algébriques sont donnés dans la Publication Pub.2, où est également présentée, pour des raisons pédagogiques, une première FMM mononiveaux. Ici, nous donnons simplement les formules utilisées dans notre FMM multiniveaux, sans détailler les calculs. Une attention particulière doit être portée sur les domaines de définitions des approximations EIM disponibles à chaque niveau de l'arbre.

Considérons un ensemble de N points d'observation $\{\bar{x}_i\}_{1 \leq i \leq N}$, N points source $\{\bar{y}_j\}_{1 \leq j \leq N}$ dans Ω , et N potentiels $\{\sigma_j\}_{1 \leq j \leq N}$, avec $N \gg 1$. Notre FMM multiniveaux doit aboutir à une approximation en complexité algorithmique meilleure que $O(N^2)$ de la sommation suivante :

$$f(\bar{x}_i) = \sum_{j=1}^N \sigma_j K(\bar{x}_i, \bar{y}_j), \quad 1 \leq i \leq N. \quad (2.14)$$

La FMM cherche à compresser les interactions lointaines notées

$$f_{\mathcal{F}}(\bar{x}_i) = \sum_{J \in \mathcal{F}(I)} \sum_{\bar{y}_j \in J} \sigma_j K(\bar{x}_i, \bar{y}_j), \quad \text{pour tout boîte } I \text{ et tout } \bar{x}_i \in I. \quad (2.15)$$

Les notations et concepts autour de la FMM ont été introduits dans l'introduction.

Une FMM multiniveaux

Les domaines introduits au niveau k de l'arbre sont $I_0^k = (-L + l_k, -3l_k) \cup (3l_k, L - l_k)$, and $J_0^k = (-l_k, l_k)$, et un indice k est ajouté aux quantités dépendant du niveau de l'arbre. L'exercice consiste à déterminer des formules de récurrences pour propager l'information du sommet à la racine (passe upward), puis de la racine au sommet (passe downward).

Nous appelons EIFMM la procédure proposée de FMM multiniveaux reposant sur une approximation des interactions lointaines par EIM. EIFMM est présenté dans l'Algorithme 2, où les étapes 3 et 6 sont des étapes supplémentaires par rapport aux FMM multiniveaux classiques, conséquences de notre approximation originale (2.13) de rang faible des interactions lointaines.

La complexité algorithmique de EIFMM est donnée dans la Table 2.3. Pour que l'étape 4 de l'Algorithme 2 et le calcul des interaction proches soit de même complexité algorithmique par rapport à N , nous devons imposer $(2^D)^\kappa \propto N^2(2^D)^{-\kappa}$, conduisant à une complexité globale en $O(N)$. Le choix pour la hauteur de l'arbre

Algorithm 2 EIFMM multiniveau

1. (M2M début) Calculer $W_m^{I^\kappa} = \sum_{\bar{y}_j \in I^\kappa} \sigma_j K(x_m^{I_0^\kappa}, \bar{y}_j - c_{I^\kappa})$, pour toute boîte I^κ au niveau κ et tout $1 \leq m \leq d^\kappa$
 2. (M2M) Calculer par récurrence $W_m^{I^k} = \sum_{p,q=1}^{d^{k+1}} \Delta_{p,q}^{J_0^{k+1}, J_0^{k+1}} \sum_{J^{k+1} \in \mathcal{C}(I^k)} K(x_m^{I_0^k} + c_{I^k} - c_{J^{k+1}}, y_p^{J_0^{k+1}}) W_q^{J^{k+1}}$, pour toute boîte I^k aux niveaux $\kappa - 1 \geq k \geq 0$ et tout $1 \leq m \leq d^{k+1}$
 3. (Étape sup. upward) Calculer $\hat{W}_l^{I^k} = \sum_{m=1}^{d^k} \Delta_{l,m}^{J_0^k, J_0^k} W_m^{I^k}$, pour toute boîte I^k à tous les niveaux $0 \leq k \leq \kappa$ et tout $1 \leq l \leq d^k$
 4. (M2L) Calculer $g_{m'}^{I^k} = \sum_{J^k \in \mathcal{I}(I^k)} \sum_{l=1}^{d^k} K(x_{m'}^{J_0^k}, y_l^{J_0^k} + c_{J^k} - c_{I^k}) \hat{W}_l^{J^k}$, pour tout boîte I^k à tous les niveaux $0 \leq k \leq \kappa$ et tout $1 \leq m' \leq d^k$.
 5. (L2L) Fixer $l_{m'}^{I^0} = g_{m'}^{I^0}$ et calculer par récurrence $l_{m'}^{I^k} = g_{m'}^{I^k} + \sum_{p',q'=1}^{d^{k-1}} \Delta_{p',q'}^{J_0^{k-1}, J_0^{k-1}} K(x_{m'}^{J_0^k} + c_{I^k} - c_{\mathcal{P}(I^k)}, y_{p'}^{I_0^{k-1}}) l_{q'}^{\mathcal{P}(I^k)}$ pour toute boîte I^k aux niveaux $1 \leq k \leq \kappa$ et tout $1 \leq m' \leq d^{k-1}$
 6. (Étape sup. downward) Calculer $\hat{l}_{l'}^{I^\kappa} := \sum_{m'=1}^{d^\kappa} \Delta_{l',m'}^{J_0^\kappa, I_0^\kappa} l_{m'}^{I^\kappa}$ pour toute boîte I^κ au niveau κ et tout $1 \leq l' \leq d^\kappa$
 7. (L2L fin) Calculer $f_{\mathcal{F}, \text{EIFMM}}(\bar{x}_i) = \sum_{l'=1}^{d^\kappa} K(\bar{x}_i - c_{I^\kappa}, y_{l'}^{I_0^\kappa}) \hat{l}_{l'}^{I^\kappa}$, pour toute boîte I^κ et tout $x_i \in I^\kappa$ au niveau κ
-

Step number	Complexity
1	dN
2	$d^3(2^D)^\kappa$
3	$d^2(2^D)^\kappa$
4	$d^2 n_I (2^D)^\kappa$
5	$d^3(2^D)^\kappa$
6	$d^2(2^D)^\kappa$
7	dN
$f_{\mathcal{N}}$	$N^2(2^D)^{-\kappa}$

TABLE 2.3 – Complexité algorithmique des étapes de l’Algorithme 2 (pour le niveau le plus cher) et du calcul des interactions proches; n_I est le nombre maximum d’éléments dans la liste d’interaction d’une boîte (4 en 1D, 40 en 2D et 316 en 3D).

minimisant le coût total est $\kappa \propto \log(N)$ et un nombre constant de points par boîte au sommet de l’arbre. Grâce aux formules de récurrence dans la FMM multiniveaux, l’étape 4 de l’Algorithme 2 fait intervenir une somme sur les boîtes J^k dans la liste d’interaction de I^k , dont le nombre d’éléments est indépendant de N , alors que dans une FMM mononiveau, cette somme se fait sur les boîtes bien séparées de I^k , dont le nombre d’éléments dépend de I^k . La complexité algorithmique des étapes 2, 4 et 5 est réduite en précalculant certains termes.

Optimisation de la complexité algorithmique

Dans leur implémentation pratique, des étapes de compression supplémentaires sont classiquement appliquées pour réduire au maximum les temps d’exécution. Nous présentons d’abord des étapes de précalcul adaptées aux spécificités de notre algorithme, puis nous évoquons simplement des compressions plus classiques pour la passe de transfert (M2L). Nous avons implémenté les optimisations présentées ici.

Précalculs pour les étapes M2M et L2L

Considérons l’étape 2 de l’Algorithme 2 au niveau k . En définissant la matrice $K_{m,p}^{I^k, J^{k+1}} = K(x_m^{I_0^k} + c_{I^k} - c_{J^{k+1}}, y_p^{J_0^{k+1}})$, $1 \leq m \leq d^k$, $1 \leq p \leq d^{k+1}$, nous remarquons que $W_m^{I^k} = \sum_{J^{k+1} \in \mathcal{C}(I^k)} K^{I^k, J^{k+1}} \Delta_{0,0}^{I_0^{k+1}, J_0^{k+1}} W^{J^{k+1}}$. La matrice $K_{M2M}^{I^k, J^{k+1}} := K^{I^k, J^{k+1}} \Delta_{0,0}^{I_0^{k+1}, J_0^{k+1}}$ peut être précalculée par $\left((\Delta_{0,0}^{I_0^{k+1}, J_0^{k+1}})^t (K^{I^k, J^{k+1}})^t \right)^t$, où chaque produit matrice-vecteur pour chaque colonne de $(K^{I^k, J^{k+1}})^t$ est calculé en résolvant des systèmes linéaires faisant intervenir les matrices B et Γ des EIM correspondants (voir l’introduction) pour préserver la précision numérique (au lieu de précalculer une matrice inverse). Notons que $K_{M2M}^{I^k, J^{k+1}}$ ne dépend que des positions relatives de I^k and J^{k+1} , et comme J^{k+1} est un enfant de I^k , il y a seulement 2^D opérateurs $K_{M2M}^{I^k, J^{k+1}}$ différents, où D est la dimension de l’espace contenant les particules. Ces opérateurs sont notés K_{M2M}^i , $1 \leq i \leq 2^D$, et l’étape 2 de

l'Algorithme 2 au niveau k est réduite à

$$W_m^{I^k} = \sum_{i=1}^{2^D} K_{M2M}^i W_q^{J_i^{k+1}}, \quad (2.16)$$

où J_i^{k+1} désigne le $i^{\text{ème}}$ enfant de I^k . Cette étape est maintenant de complexité algorithmique $d^2 2^D (2^D)^k$ au niveau k , et de complexité dominante $d^2 2^D (2^D)^{\kappa-1} = d^2 (2^D)^\kappa$. La même optimisation est faite pour l'étape 5 de l'Algorithme 2, réduisant sa complexité algorithmique à $d^2 (2^D)^\kappa$.

Optimisation de l'étape M2L

Cette étape est connue comme étant la plus coûteuse, et de nombreux efforts ont été consacrés pour réduire son temps d'exécution sans dégrader la qualité du résultat. Nous utilisons ici la procédure SARcmp proposée dans [89], en l'adaptant à notre algorithme.

Considérons un niveau k , une boîte I^k à ce niveau, et une boîte J^k dans la liste d'interaction de I^k . Le vecteur $\delta := c_{J^k} - c_{I^k}$ dépend seulement des positions relatives de boîtes I^k et J^k . Ainsi, il y a seulement $n_I = 7^D - 3^D$ matrices (appelées M2L) $K_{i,j}^\delta := K(x_i^{J_i^k}, y_j^{J_j^k} + \delta)$ différentes par niveau. Supposons que les algorithmes EIM ont été exécutés pour une erreur relative ϵ sur l'approximation lointaine du noyau, conduisant à la sélection de d_k points d'interpolation. Les matrices M2L sont organisées dans une grande matrice $K_{\text{fat}} := [K^{\delta_1} K^{\delta_2} \dots K^{\delta_{n_I}}]$, où $K_{\text{fat}} \in \mathbb{R}^{d_k \times n_I d_k}$. Deux matrices rectangulaires U, V sont calculées par un fully pivoted ACA, de sorte que $|UV - K_{\text{fat}}|_F \leq \epsilon |K_{\text{fat}}|_F$, où $|\cdot|_F$ est la norme de Frobenius. Ensuite, deux décompositions QR sont calculées, telles que $U = Q_U R_U$ et $(V)^t = Q_V R_V$, et une SVD de $R_U (R_V)^t$ est également calculée. Notons s_i , $1 \leq i \leq d_k$, les valeurs singulières de $R_U (R_V)^t$ en ordre décroissant, et \hat{U}, \hat{V} les matrices unitaires telles que $R_U (R_V)^t = \hat{U} \text{diag}(s) \hat{V}$. Les deux filtres suivants sont appliqués : nous gardons les premières j valeurs singulières telles que (i) $\frac{s_j}{s_0} \leq \epsilon \leq \frac{s_{j-1}}{s_0}$, (ii) $\frac{\sum_{i=1}^j s_i}{\sum_{i=1}^{d_k} s_i} \geq 1 - \epsilon \geq \frac{\sum_{i=1}^{j-1} s_i}{\sum_{i=1}^{d_k} s_i}$. Notons r_k le nombre de valeurs singulières retenues,

et \hat{U}_{r_k} et \hat{V}_{r_k} les r_k premières colonnes de respectivement \hat{U} et \hat{V} . En suivant [53], nous montrons que les étapes M2L peuvent être réduites à des produits matrice-vecteur de taille $r_k \times r_k$ (au lieu de taille $d_k \times d_k$) faisant intervenir les matrices $C^\delta := \hat{U}_{r_k}^T K^\delta \hat{U}_{r_k}$, et quelques précalculs de plus faible complexité algorithmique.

Ensuite, la même compression est appliquée localement sur chaque opérateur M2L compressé C^δ . Cette fois-ci, le fully pivoted ACA est appliqué sur chaque C^δ , conduisant comme précédemment à des matrices U and V . Les décompositions QR, SVD et filtres sont appliqués de la même façon, sélectionnant les valeurs singulières s_k . Les matrices rectangulaires \tilde{U}_{s_k} et \tilde{V}_{s_k} sont construites, ainsi que la matrice diagonale S_{s_k} , dont les entrées diagonales sont les racines carrées des valeurs singulières sélectionnées. Enfin, nous construisons $\tilde{U}_{s_k} = Q_U \tilde{U}_{s_k} S_{s_k} \in \mathbb{R}^{r_k \times s_k}$ and $\tilde{V}_{s_k} = S \tilde{V}_{s_k} Q_V^t \in \mathbb{R}^{s_k \times r_k}$, de sorte que C^δ est approché par $\tilde{U}_{s_k} \tilde{V}_{s_k}$. Ainsi, les produits matrice-vecteur de taille $r_k \times r_k$ faisant intervenir les matrices C^δ sont remplacés par deux produits matrice-vecteur de taille $r_k \times s_k$ faisant intervenir les matrices \tilde{U}_{s_k} et \tilde{V}_{s_k} .

Applications numériques en 3D

L'algorithme EIFMM a été implémenté en C++, en utilisant la librairie open source scalfmm, voir [40]. Dans notre implémentation, les étapes de précalcul EIM et les optimisations SARcmp sont codés en C++ en utilisant BLAS et LAPACK, et stockés en fichiers binaires. Ces précalculs sont ensuite lus par la librairie scalfmm modifiée pour nos besoins.

Nous comparons notre algorithme et son implémentation avec la Black-Box FMM [53], dans sa version symétrique optimisée [89] disponible dans scalfmm. A notre connaissance, il s'agissait de l'implémentation la plus récente au moment de cette étude. Notons que dans notre procédure, l'EIM va automatiquement sélectionner une approximation des interactions lointaines avec un contrôle de l'erreur. Comme Black-Box FMM ne dispose pas d'un contrôle a priori de l'erreur, nous déterminons par dichotomie, pour chaque comparaison numérique, le nombre de points d'interpolation de Chebyshev optimal au niveau de précision requis. En faisant ainsi, nous nous assurons que les tests avec Black-Box FMM ne sont pas artificiellement ralentis avec une valeur trop grande du nombre de points d'interpolation. L'intérêt principal de notre approche étant une approximation des interactions lointaines de façon adaptative dans chaque niveau de l'arbre, nous illustrons notre approche sur des noyaux inhomogènes.

Cas tests à 1 million de points

Considérons des ensembles de points dans $\Omega = (-0.5, 0.5)^3$, pris (i) aléatoirement selon une loi uniforme dans Ω , (ii) aléatoirement selon une loi uniforme sur une sphère de rayons 0.5 et (iii) par transformation de (ii) en un ellipsoïde, voir Figure 2.19.

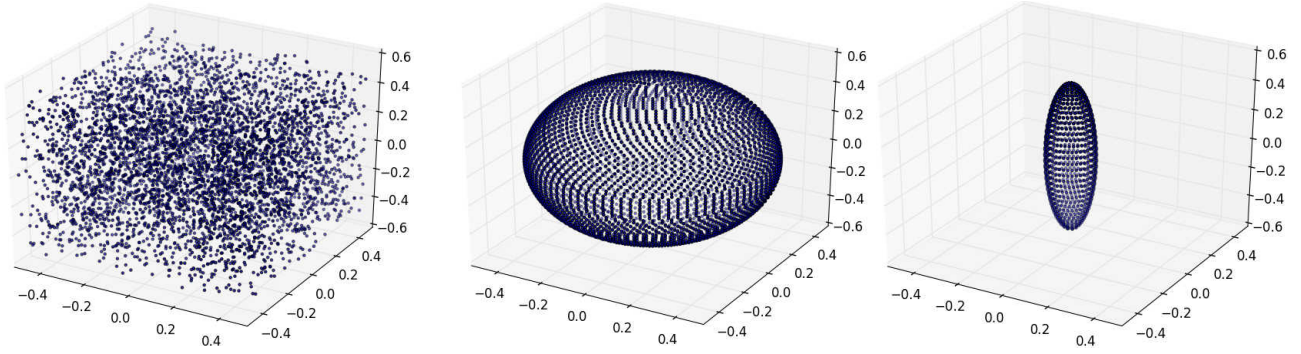


FIGURE 2.19 – Ensembles de points considérés dans l’application numérique; de gauche à droite : aléatoire (5000 points), sphère (10100 points), et ellipsoïde (1944 points). Ces valeurs sont choisies pour des raisons de représentation et des ensembles plus grands seront considérés dans l’application numérique.

Considérons les 3 cas-tests présentés dans la Figure 2.19 : 10^6 sont pris aléatoirement dans le cube unitaire avec un arbre de hauteur 6 (cas appelé ”cube“), 992,046 points à la surface d’une sphère et d’un ellipsoïde, avec respectivement des arbres de hauteur 8 et 9 (cas appelés ”sphère“ et ”ellipsoïde“). Les calculs sont réalisés en séquentiel sur un processeur Intel(R) Core(TM) i5-m560@2.67 GHz et 4 GB de mémoire vive.

Les Figures 2.20, 2.21 et 2.22 présentent les temps d’exécution pour les interactions lointaines accélérées par FMM en fonction de l’erreur relative sur la somme, respectivement pour les noyaux $\frac{\cos 20r}{r}$, e^{-r^2} et $\sqrt{r^2 + 1}$, pour les 3 cas-tests (cube, sphère et ellipsoïde). Grâce à une approximation faisant intervenir moins de points d’interpolation, EIFMM utilise moins de mémoire que Black-Box FMM, et dans les figures, les points manquants correspondent à des cas nécessitant plus de mémoire que disponible avec notre matériel, pour lesquels EIFMM a pu être utilisé mais pas Black-Box FMM. Pour les trois noyaux considérés, en particulier pour les deux derniers, EIFMM est plus rapide, grâce à la sélection optimale et automatique du nombre optimal de points d’interpolation à chaque niveau dans l’arbre. Par exemple, pour le noyau e^{-r^2} sur l’ellipsoïde et une erreur relative de 10^{-6} , la Black-Box FMM a besoin de 6 points d’interpolation par dimension (216 en tout), alors que EIFMM requiert respectivement 167, 74, 36, 21, 13, 10, 9, et 4 points d’interpolation aux niveaux 2 à 9 de l’arbre. Au fur et à mesure que nous montons dans les niveaux de l’arbre, le problème d’interpolation devient plus facile et EIM sélectionne de moins en moins de points d’interpolation pour une approximation de la même qualité.

Du fait de la décroissance rapide du noyau e^{-r^2} , d’autres méthodes dédiées ont été proposées, comme la fast Gauss transform [60]. Dans les cas considérés ici, la plus grande valeur possible pour r est $\sqrt{3}$ dans le cube (de la Figure 2.19), où l’évaluation du noyau est toujours à 5% de la valeur maximum à $r = 0$ pour le noyau e^{-r^2} , et la capacité de notre procédure à traiter les noyaux inhomogènes peut être correctement illustrée.

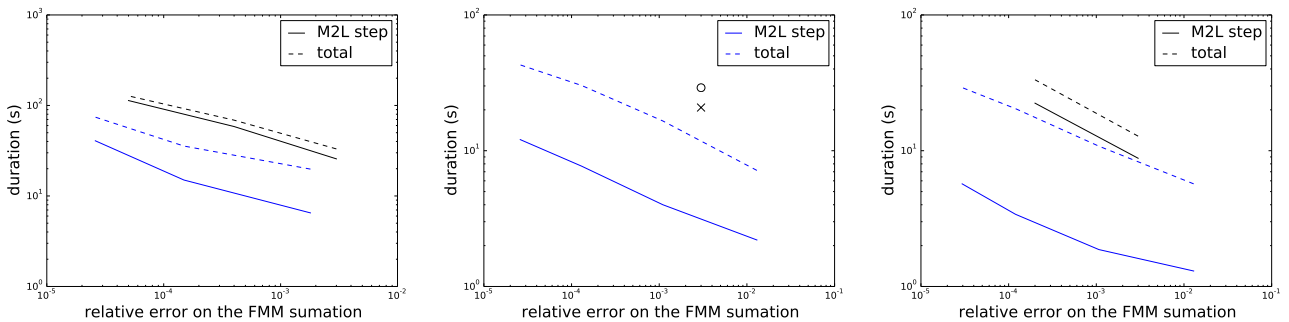


FIGURE 2.20 – Temps d’exécution pour les interactions lointaines accélérées par FMM en fonction de l’erreur relative sur la somme, pour le noyau $\frac{\cos 20r}{r}$; noir : Black-Box FMM symétrique, bleu : EIFMM. De gauche à droite : cas tests cube, sphère et ellipsoïde.

La Table 2.4 les temps de précalcul et les temps d’exécution pour les noyaux $\frac{\cos(20r)}{r}$, e^{-r^2} et $\sqrt{r^2 + 1}$ sur le cas-test du cube. Remarquons que, pour être sûrs de ne pas désavantager Black-Box FMM, nous comparons avec

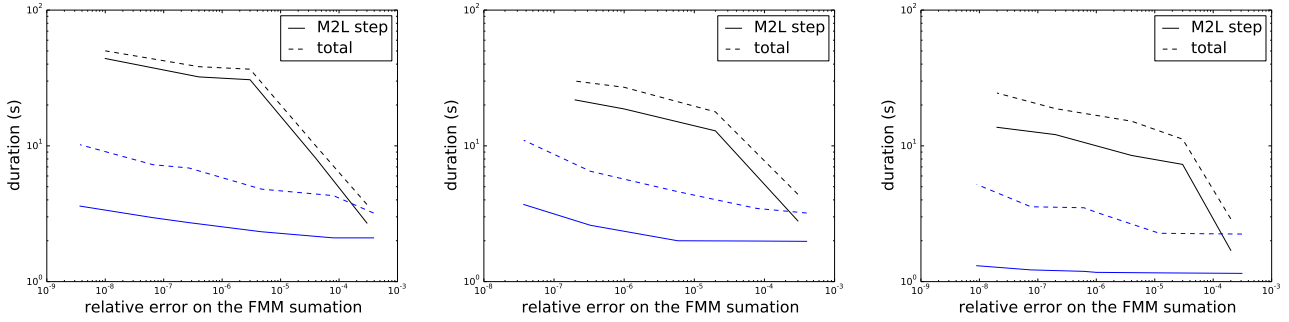


FIGURE 2.21 – Temps d’exécution pour les interactions lointaines accélérées par FMM en fonction de l’erreur relative sur la somme, pour le noyau e^{-r^2} ; noir : Black-Box FMM symétrique, bleu : EIFMM. De gauche à droite : cas tests cube, sphère et ellipsoïde.

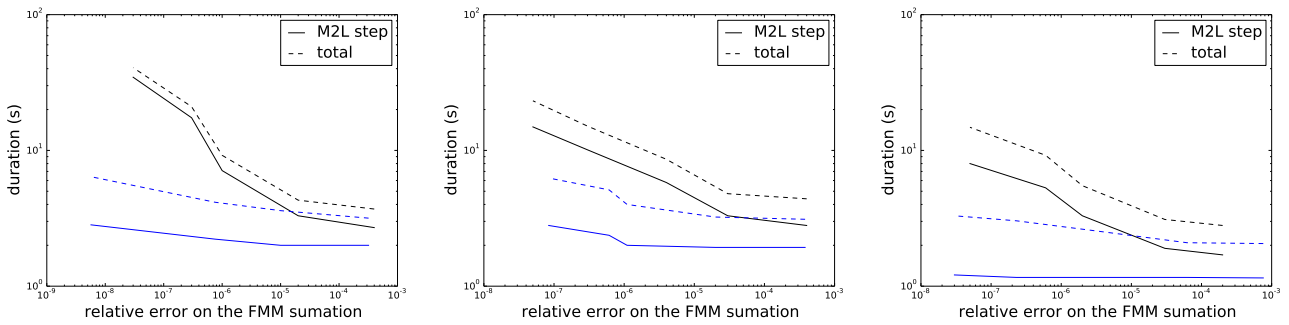


FIGURE 2.22 – Temps d’exécution pour les interactions lointaines accélérées par FMM en fonction de l’erreur relative sur la somme, pour le noyau $\sqrt{r^2 + 1}$; noir : Black-Box FMM symétrique, bleu : EIFMM. De gauche à droite : cas tests cube, sphère et ellipsoïde.

des précisions relatives de EIFMM légèrement meilleures. Dans les cas considérés, EIFMM devient avantageux (i.e. les temps de précalcul de EIFMM sont compensés par des interactions lointaines plus rapides) si l’on a 2 à 18 sommes à réaliser. Notons que d’un calcul à l’autre, la position des points et la valeur des potentiels peut changer : seul l’arbre doit rester identique.

Pour les cas-tests avec plus de points, l’étape M2L devient très coûteuse, et l’avantage de EIFMM peut apparaître dès le calcul de la première somme.

Cas-test à 100 millions de points

Nous considérons 10^8 points pris aléatoirement dans le cube unitaire dans un arbre à 9 niveaux, avec le noyau e^{-r^2} . Les calculs sont réalisés en séquentiel sur un processeur Intel(R) Xeon(R) E5-4620 v2@2.6 GHz et 128 GB de mémoire vive.

La Table 2.5 présente la précision relative et les temps d’exécution pour toutes les étapes du calcul avec EIFMM et Black-Box FMM. Avec une précision meilleure, EIFMM est plus rapide que Black-Box FMM, incluant les temps de précalculs (augmenter la précision du calcul avec Black-Box FMM nécessiterait plus que 128 GB de mémoire vive). Notons que pour ce cas-test faisant intervenir un nombre important de points N , les temps de précalculs d’EIFMM sont négligeables : ces temps ont une complexité algorithmique linéaire par rapport au nombre de niveaux dans l’arbre, et donc logarithmique par rapport à N .

noyau	$\frac{\cos(20r)}{r}$		e^{-r^2}		$\sqrt{r^2 + 1}$	
	EIFMM	BBFMM	EIFMM	BBFMM	EIFMM	BBFMM
précision relative	2×10^{-4}	4×10^{-4}	2×10^{-6}	3×10^{-6}	1×10^{-5}	2×10^{-5}
temps de précalcul EIM (s)	73	–	30	–	10	–
temps de compression (s)	177	–	33	–	1.1	–
temps de construction de l'arbre (s)	3		3		3	
temps de calcul des inter. lointaines (s)	52	69	5.4	37	3.5	4.3
temps de calcul des inter. proches (s)	32		30		16	
nombre d'appels minimum à EIFMM	18		2		14	

TABLE 2.4 – Temps de précalculs et de calcul pour les noyaux $\frac{\cos(20r)}{r}$, e^{-r^2} et $\sqrt{r^2 + 1}$ sur le cas-test du cube. La dernière ligne contient le nombre de sommes FMM minimum à réaliser pour être plus rapide que la Black-Box FMM (BBFMM), prenant en compte tous les temps de précalcul.

algorithme	EIFMM	Black-Box FMM
précision relative	7×10^{-5}	6×10^{-4}
temps de précalcul EIM	22.8 s	–
temps de compression	1.3 s	–
temps de compression de l'arbre	9 min 10 s	
P2M (1)	1 min 13 s	2 min 17 s
M2M (2-3)	31 s	49 s
M2L (4)	10 min 39 s	53 min 01 s
L2L (5-6)	41 s	59 s
L2P (7)	1 min 42 s	3 min 55 s
temps total de calcul des interactions lointaines	14 min 46 s	1 h 01 min 01 s
temps de calcul des interactions proches	19 min 36 s	19 min 36 s
temps total	43 min 56 s	1 h 29 min 47 s

TABLE 2.5 – Précision relative et temps d'exécution pour le cas-test à 100 millions de points. P2M, M2M, M2L, L2L et L2P correspondent respectivement aux étapes (1), (2-3), (4), (5-6) et (7) dans l'Algorithme 2.

Chapitre 3

Travaux sur les méthodes de réduction de modèle physique

3.1 Application de la méthode des bases réduites à l'étude d'un problème aérothermique multi-échelles

Cette section correspond à la Publication [Pub.20](#).

Ce travail est un projet de recherche en modélisation, répondant à une problématique industrielle. L'approche utilise des méthodes récentes au moment de l'étude, mais ne contient pas de nouvel algorithme en réduction de modèle. Nous présentons alors très brièvement le contenu de l'étude et renvoyons vers la Publication [Pub.20](#) pour les détails.

Nous considérons un problème multi-échelles 2D d'aérothermie en aviation civile. Nous souhaitons déterminer le champ de température dans un avion en conditions de vol, avec présence d'une climatisation. Des composants électroniques ventilés sont présents dans la soute, et constituent une source de chaleur, introduisant une deuxième échelle dans notre problème.

Dans un premier temps, trois modèles sont développés en **FreeFEM++** (voir [64]), pour étudier le champ de température dans la cabine ; du plus compliqué au plus simple : (i) les équations de Navier-Stokes dans l'approximation de Boussinesq, (ii) un découplage entre les équations de Navier-Stokes incompressibles stationnaires et l'équation de la chaleur, et (iii) un découplage entre un écoulement potentiel et l'équation de la chaleur. Nous vérifions que dans le cas de la climatisation d'une cabine d'avion, seule l'approximation de Boussinesq permet de calculer correctement le champ de température : la thermique ayant un effet important sur le champ de vitesse. Ensuite, nous considérons un modèle de composant électronique refroidit par un écoulement forcé. Dans ce cas, le nombre de Péclet est important, environ 10^4 , indiquant que les effets convectifs forcés sont dominants devant la convection induite par les flux thermiques locaux. Effectivement, nous remarquons sur la Figure 3.1 que le découplage Navier-Stokes incompressible stationnaire avec l'équation de la chaleur donne le même résultat que l'approximation de Boussinesq (alors que le modèle potentiel est très différent).

Dans un second temps, nous appliquons la méthode des bases réduites pour déterminer les paramètres du composant électronique qui minimisent la température moyenne sur le circuit intégré. La formulation variationnelle de l'équation de la chaleur à résoudre est : trouver $T_\mu \in M_h$ tel que $\forall \Theta \in M_h$

$$\int_{\Omega} \rho_0 c_p (\mathbf{u}_{NS} \cdot \nabla \Theta) T + k \nabla \Theta \cdot \nabla T + \frac{1}{2} \int_{\Omega} \rho_0 c_p \operatorname{div}(\mathbf{u}_{NS}) T \Theta = q \int_{\Omega_{IC}} T_\mu \quad (3.1)$$

où ρ_0 est la densité supposée constante, c_p est la capacité thermique, k est la conductivité thermique, \mathbf{u}_{NS} est le champ de vitesse précalculé par notre solveur Navier-Stokes incompressible stationnaire (porteur de la convection forcée), Ω_{IC} est la partie du domaine correspondant au circuit intégré, et où M_h est un sous-espace de dimension finie de $H^1_{\text{per},0}(\Omega) = \{T \in H^1(\Omega) \mid T_{\Gamma_{\text{in}}} = 0, T \text{ périodique sur } \Gamma_{\text{per}}\}$, avec Γ_{per} une partie du bord du composant électronique. La quantité d'intérêt est $s_\mu = l(T_\mu) = (f, T_\mu)_{H^1_0(\Omega)} = q \int_{\Omega_{IC}} T_\mu$. La forme bilinéaire intervenant dans (3.1), notée a_μ , est coercive et non symétrique.

Pour ce faire, nous utilisons un algorithme glouton construisant un modèle réduit pour le problème de

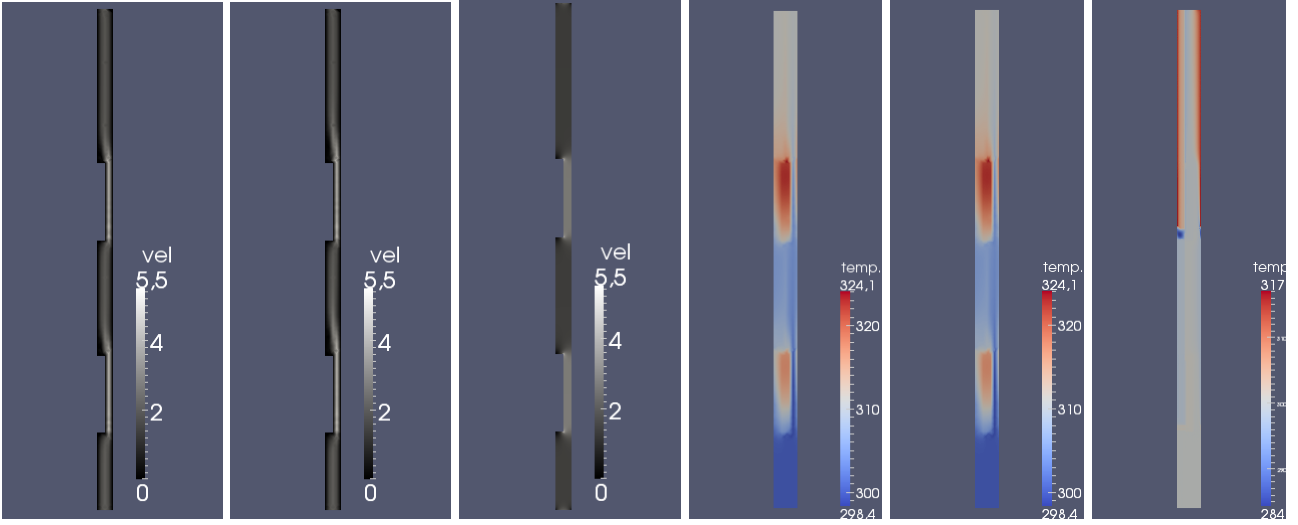


FIGURE 3.1 – Gauche : champ de vitesse, droite : champ de température, pour les cas de Boussinesq, Navier-Stokes et potentiel, dans le système international d’unités

thermique (3.1) et son problème adjoint, et nous utilisons l’estimateur d’erreur suivant

$$|s_{\mu}^{\text{RB},*\text{RB}} - s_{\mu}^{\text{FE}}| \leq \Delta_{\mu} := \frac{\|G_{\mu} T_{\mu}^{\text{RB}}\|_{H_0^1(\Omega)} \|G_{\mu}^* \Psi_{\mu}^{*\text{RB}}\|_{H_0^1(\Omega)}}{\alpha_{\text{LB},\mu}},$$

où $s_{\mu}^{\text{RB},*\text{RB}}$ est notre quantité d’intérêt construite par le modèle réduite, s_{μ}^{FE} correspond à son estimation par le modèle haute-fidélité, $\|G_{\mu} T_{\mu}^{\text{RB}}\|_{H_0^1(\Omega)}$ correspond à la norme duale du résidu (entre estimation réduite et haute-fidélité) du problème direct et $\|G_{\mu}^* \Psi_{\mu}^{*\text{RB}}\|_{H_0^1(\Omega)}$ correspond à la norme duale du résidu du problème adjoint, et où $\alpha_{\text{LB},\mu}$ est une borne inférieure pour la constante de coercivité α_{μ} et $a_{\mu}(\cdot, \cdot)$ (une borne dans le cas continu est $\alpha_{\text{LB},\mu} = \min(\kappa_{\mu})$). Les problèmes direct et adjoint, ainsi que l’estimateur d’erreur (3.1) sont calculés en complexité indépendante de la taille du problème haute-fidélité en utilisant une hypothèse de dépendance affine en les paramètres considérés.

Résultats numériques Le problème haute-fidélité a 9,012 degrés de liberté, et la procédure gloutonne entraînée sur un échantillonnage de 20,000 points de l’espace paramétrique a sélectionné 10 fonctions de base. Les paramètres choisis sont κ_{IC} , κ_{board} , κ_{air} les conductivités thermiques du circuit intégré, du support du composant et de l’air et $c_{p\text{air}}$ la capacité thermique de l’air, variant dans les intervalles suivants (dans le système international d’unités) :

	κ_{IC}	κ_{board}	κ_{air}	$c_{p\text{air}}$
min	0.5	0.06	0.028	1080
max	5	0.6	0.032	1120

Les paramètres relatifs à l’air sont ici des paramètres incertains et non pas des paramètres de design.

L’évolution du maximum de l’estimation d’erreur Δ_{μ} pour $\mu \in \mathcal{D}_{\text{trial}}$ et la valeur correspondante de l’erreur $|s_{\mu}^{\text{RB},*\text{RB}} - s_{\mu}^{\text{FE}}|$ sont représentées sur la Figure 3.2, avec la valeur de l’estimateur pour une valeur prise au hasard du paramètre.

Une fois le modèle réduit construit, nous l’appelons pour une valeur du paramètre prise au hasard (voir la Figure 3.3) :

- Estimateur d’erreur a posteriori : 7.24×10^{-6}
- Erreur : 1.20×10^{-6}
- Température du circuit intégré : estimation du modèle réduit 40.99 K ; estimation haute-fidélité 40.96 K
différence relative = 0.074 %

La durée des différentes étapes de l’algorithme sont

- phase offline : 8min37s

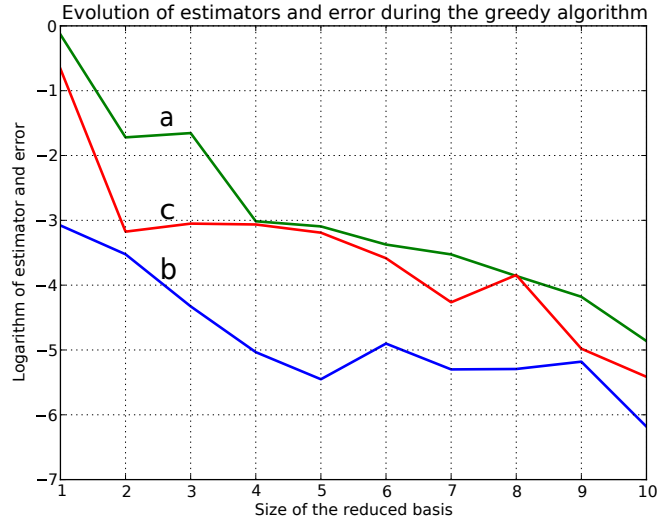


FIGURE 3.2 – Évolution du maximum de l’estimateur d’erreur Δ_μ pour un $\mu \in \mathcal{D}_{\text{trial}}$ (a), valeur correspondante de l’erreur $|s_\mu^{\text{RB},*\text{RB}} - s_\mu^{\text{FE}}|$ (b) et estimateur d’erreur $\Delta_{\hat{\mu}}$ pour un $\hat{\mu} \in \mathcal{D}$ pris aléatoirement (en dehors de $\mathcal{D}_{\text{trial}}$) (c)

- une évaluation du modèle haute-fidélité : 0.28s
- une évaluation du modèle haute réduit : 1.4×10^{-6} s

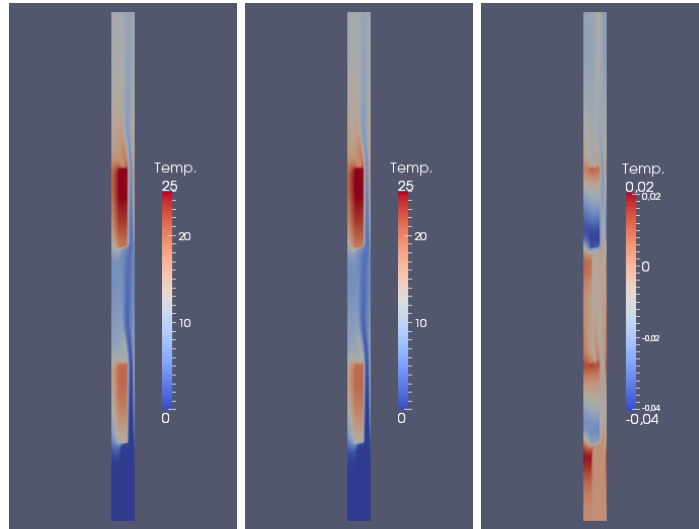


FIGURE 3.3 – Champs de température pour le composant électronique; gauche : modèle réduit u_{RB} , centre : modèle haute-fidélité u_{FE} , droite : différence $u_{\text{FE}} - u_{\text{RB}}$

3.2 Une méthode POD-EIM pour les problème de thermique transitoire non-linéaire avec une convection importante

Cette section correspond au Rapport [Rap.2](#) et aux Exposés [Exp.18](#) et [Exp.19](#).

Dans cette section, nous appliquons la réduction de modèle par projection sur l’équation de la chaleur transitoire non-linéaire, en présence d’effets de translation importants, pour modéliser le moulage de matériaux monocristallin, comme motivé dans l’introduction du manuscrit. La méthode “Apparent Heat Capacity” est

utilisée pour modéliser le changement de phase du superalliage, ce qui introduit de fortes non-linéarités. Nous détaillons comment l'EIM et la DEIM sont utilisées pour la construction efficace du problème réduit dans la phase online (la DEIM est appliquée sur des quantités intégrées comme proposé dans l'article original [35]). Dans les applications considérées, l'EIM est plus stable que la DEIM appliquée sur les quantités intégrées. Les effets de translation sont connus pour être mal réductibles : des bases POD locales sont utilisées pour améliorer la réductibilité et améliorer le speedup. Nous illustrons comment les instabilités du modèle réduit par POD-EIM peuvent être améliorées en enrichissant l'ensemble d'apprentissage d'EIM, stoppant la sélection gloutonne de façon prématurée et en remplaçant le système linéaire résolu dans la phase online d'EIM pour des moindres carrés régularisés. Enfin, la méthodologie est appliquée au moulage d'une aube de turbine basse pression industrielle.

Modèle haute-fidélité

Nous considérons une structure Ω , située dans un four à haute température. La structure est tradlatée hors du four par le bas : la frontière de la structure située à l'intérieur et à l'extérieur du four est notée respectivement $\partial\Omega_N$ et $\partial\Omega_R$. Nous nous plaçons dans le référentiel de Ω : la translation est ainsi prise en compte par des conditions aux limites dépendant du temps, voir Figure 3.4.

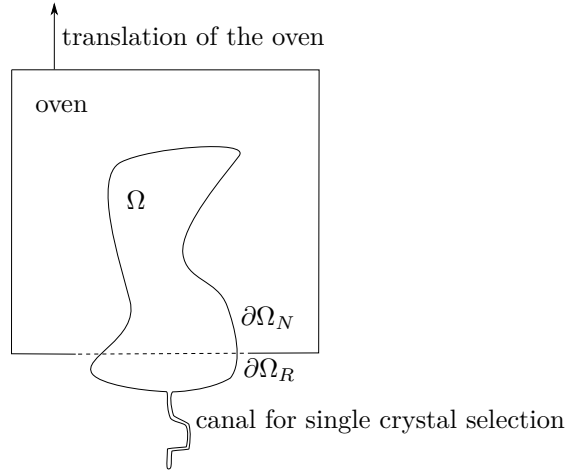


FIGURE 3.4 – Modélisation de la structure d'intérêt Ω

La conservation de l'énergie issue de la première loi de la thermodynamique en l'absence de travail s'écrit

$$\int_V \frac{\partial u}{\partial t} = - \int_{\partial V} q \cdot n + \int_V r, \quad (3.2)$$

où V est un volume de matière quelconque et ∂V sa frontière, n est la normale extérieure sur ∂V , u l'énergie interne volumique, q est la densité de flux thermique ($q \cdot n|_{\partial V}$ est la chaleur perdue par le volume de matière vers l'extérieur), et r est une source de chaleur volumique. Dans notre cas, $r = 0$. Le théorème de flux-divergence appliqué à l'intégrale de surface dans (3.2) fournit la relation locale $\frac{\partial u}{\partial t} + \nabla \cdot q = 0$. En utilisant la loi de Fourier $q = -\lambda \nabla T$, où T est l'inconnue de température, λ est la conductivité thermique (supposée être une fonction connue de la température), l'évolution de la structure satisfait l'équation de la chaleur transitoire non-linéaire suivante :

$$\frac{\partial u}{\partial t} - \nabla \cdot (\lambda \nabla T) = 0 \quad \text{in } \Omega \times [0, t_F] \quad (\text{équation de la chaleur}), \quad (3.3a)$$

$$\lambda \nabla T \cdot n = 0 \quad \text{in } \partial\Omega_N \times [0, t_F] \quad (\text{condition aux limites de Neumann homogène}), \quad (3.3b)$$

$$\lambda \nabla T \cdot n = \sigma \epsilon (T^4 - T_0^4) \quad \text{in } \partial\Omega_R \times [0, t_F] \quad (\text{condition aux limites de radiation}), \quad (3.3c)$$

$$T = T_{\text{oven}} \quad \text{in } \Omega \text{ at } t = 0 \quad (\text{condition initiale}), \quad (3.3d)$$

où T_{oven} et T_0 sont des températures constantes de référence, n est la normale extérieure sur $\partial\Omega_R$, σ est la constante de Stefan-Boltzmann et ϵ est le coefficient d'émission de surface du matériau considéré pour la structure Ω . Un schéma d'Euler implicite est utilisé pour la discrétisation temporelle et une base éléments-finis

$\{\varphi_i\}_{1 \leq i \leq N}$, dont l'espace engendré est noté \mathcal{V} , est utilisée pour la discrétisation spatiale. Le système d'équations non-linéaires obtenu s'écrit

$$\mathcal{F}_j(T_{t+\Delta t}) := \int_{\Omega} \frac{u(T_{t+\Delta t}) - u(T_t)}{\Delta t} \varphi_j + \int_{\Omega} \lambda(T_{t+\Delta t}) \nabla T_{t+\Delta t} \cdot \nabla \varphi_j - \int_{\partial\Omega_R} \sigma \epsilon (T_{t+\Delta t}^4 - T_0^4) \varphi_j = 0, 1 \leq j \leq N. \quad (3.4)$$

Nous utilisons un algorithme de Newton pour résoudre le système (3.4) à chaque pas de temps :

$$\frac{D\mathcal{F}}{DT}(T_{t+\Delta t}^k) (T_{t+\Delta t}^{k+1} - T_{t+\Delta t}^k) = -\mathcal{F}(T_{t+\Delta t}^k), \quad (3.5)$$

où les exposants k désignent la k -ème itération de l'algorithme de Newton, $T_{t+\Delta t}^0 = T_t$, les éléments de \mathbb{R}^N sont identifiés avec les éléments de \mathcal{V} (par exemple $T_{t+\Delta t}^k \in \mathbb{R}^N$ est identifié avec $\sum_{j=1}^N (T_{t+\Delta t}^k)_j \varphi_j$) et $\frac{D\mathcal{F}}{DT}(T_{t+\Delta t}^k) \in \mathbb{R}^{N \times N}$ est tel que

$$\begin{aligned} \left(\frac{D\mathcal{F}}{DT}\right)_{i,j}(T_{t+\Delta t}^k) &= \frac{1}{\Delta t} \int_{\Omega} c(T_{t+\Delta t}^k) \varphi_i \varphi_j + \int_{\Omega} \lambda(T_{t+\Delta t}^k) \nabla \varphi_i \cdot \nabla \varphi_j \\ &\quad - \int_{\partial\Omega_R} 4\sigma \epsilon (T_{t+\Delta t}^k)^3 \varphi_i \varphi_j + \int_{\Omega} \frac{d\lambda}{dT}(T_{t+\Delta t}^k) \varphi_i \nabla T_{t+\Delta t}^k \cdot \nabla \varphi_j, 1 \leq i, j \leq N, \end{aligned} \quad (3.6)$$

où l'énergie interne volumique $u(T)$ et la capacité thermique volumique $c(T) = \frac{du}{dT}$ sont supposées des fonctions connues de la température. Nous choisissons d'ignorer le dernier terme dans (3.6) pour obtenir une matrice tangente symétrique, comme cela est souvent fait en pratique. Nous renvoyons vers [72] pour des discussions sur l'utilisation de la partie symétrique de la matrice tangente dans des schémas de Newton pour l'équation de la chaleur non-linéaire.

Pour éviter le tracking effectif de l'interface solide/liquide, la chaleur latente de fusion est modélisée par la méthode Apparent Heat Capacity (AHC) [25, 91], qui consiste en un maximum local de la fonction de capacité thermique autour de la température de fusion, voir Figure 3.5 pour des représentations des dépendances en température de la capacité thermique volumique et de la conductivité thermique considérées. La méthode AHC introduit des effets non-linéaires autour de la température de fusion, qui doivent être pris en compte correctement par le modèle réduit, sans dégrader la stabilité du schéma numérique. Nous référons aux tests avec AHC quand la capacité thermique volumique c_1 de la Figure 3.5 est utilisée, et sans AHC quand c_2 est utilisée. Dans le reste de la section, lorsque nous étudions la précision du ROM, nous utilisons "erreur" pour faire référence à la moyenne temporelle des erreurs relatives en norme $L^2(\Omega)$.

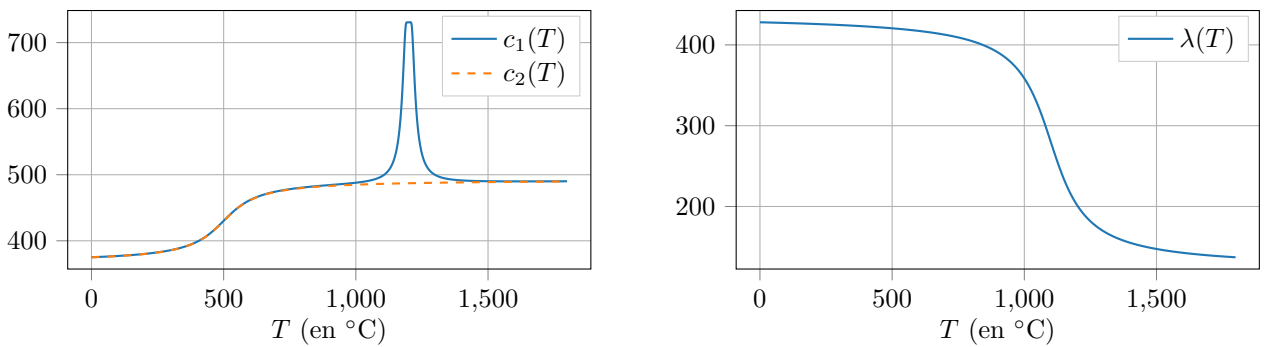


FIGURE 3.5 – Gauche : capacité thermique volumique (en $\text{J.m}^{-3}.\text{K}^{-1}$) (c_1 contient un modèle pour la transition de phase liquide-solide dans le contexte de la méthode Apparent Heat Capacity) ; droite : conductivité thermique (en $\text{W.m}^{-1}.\text{K}^{-1}$)

Considérons un domaine paramétrique \mathcal{P} , un compact de \mathbb{R}^r , $r \in \mathbb{N}^*$, nous introduisons le problème paramétrique : connaissant T_t^μ , trouver $T_{t+\Delta t}^\mu$ tel que

$$\mathcal{F}^\mu(T_{t+\Delta t}^\mu) = 0. \quad (3.7)$$

Modèle réduit

Supposons que le problème (3.7) a été résolu pour certaines valeurs paramétriques, produisant une collection de champs de température solution T_s , $1 \leq s \leq N_c$ appelés snapshots, où s est un indice décrivant les valeurs de paramètres et de temps des résolutions. Le modèle réduit consiste en une méthode de Galerkin similaire à (3.4), mais écrite sur une base d'ordre réduit $(\psi_i)_{1 \leq i \leq n}$, avec $n \ll N$, obtenue comme un post-traitement des snapshots T_s , au lieu de la base éléments-finis $\{\varphi_i\}_{1 \leq i \leq N}$. La procédure est décomposée en deux étapes : (i) l'étape offline, où l'information sur le modèle est apprise et toutes les opérations en complexité algorithmique dépendant de N sont exécutées, (ii) l'étape online, où le problème réduit est construit et résolu : c'est la phase d'exploitation, exécutée efficacement, c'est-à-dire en complexité algorithmique indépendante de N .

Étape online

Le modèle réduit est résolu avec un algorithme de Newton réduit :

$$\frac{D\hat{\mathcal{F}}}{DT} \left(\hat{T}_{t+\Delta t}^k \right) \left(\hat{T}_{t+\Delta t}^{k+1} - \hat{T}_{t+\Delta t}^k \right) = -\hat{\mathcal{F}}(\hat{T}_{t+\Delta t}^k), \quad (3.8)$$

où $\hat{T}_{t+\Delta t}^0 = \hat{T}_t$ (la solution réduite connue au pas de temps t), les éléments de \mathbb{R}^n sont identifiés avec les éléments de $\hat{\mathcal{V}} := \text{Span} \{ \Phi_j \}_{1 \leq j \leq n}$ (par exemple $\hat{T}_{t+\Delta t}^k \in \mathbb{R}^n$ est identifié avec $\sum_{j=1}^n \left(\hat{T}_{t+\Delta t}^k \right)_j \Phi_j$), et $\hat{\mathcal{F}}(\hat{T}_{t+\Delta t}^k) \in \mathbb{R}^n$ est tel que

$$\hat{\mathcal{F}}_j(\hat{T}_{t+\Delta t}^k) := \int_{\Omega} \frac{u(\hat{T}_{t+\Delta t}^k) - u(\hat{T}_t^k)}{\Delta t} \Phi_j + \int_{\Omega} \lambda(\hat{T}_{t+\Delta t}^k) \nabla \hat{T}_{t+\Delta t}^k \cdot \nabla \Phi_j - \int_{\partial\Omega_R} \sigma \epsilon \left(\left(\hat{T}_{t+\Delta t}^k \right)^4 - \hat{T}_0^4 \right) \Phi_j, \quad 1 \leq j \leq n, \quad (3.9)$$

et $\frac{D\hat{\mathcal{F}}}{DT} \left(\hat{T}_{t+\Delta t}^k \right) \in \mathbb{R}^{n \times n}$ est tel que

$$\left(\frac{D\hat{\mathcal{F}}}{DT} \right)_{i,j} \left(\hat{T}_{t+\Delta t}^k \right) = \int_{\Omega} c \left(\hat{T}_{t+\Delta t}^k \right) \Phi_i \Phi_j + \int_{\Omega} \lambda(\hat{T}_{t+\Delta t}^k) \nabla \Phi_i \cdot \nabla \Phi_j - \int_{\partial\Omega_R} 4\sigma \epsilon \hat{T}_{t+\Delta t}^3 \Phi_i \Phi_j, \quad 1 \leq i, j \leq n, \quad (3.10)$$

où la contribution non-symétrique a été ignorée, comme expliqué précédemment. La solution réduite est obtenue comme $\hat{T}_{t+\Delta t} = \sum_{i=1}^n \left(\hat{T}_{t+\Delta t} \right)_i \Phi_i$.

Notons que même si (3.8) est un système linéaire d'inconnue $\hat{T}_{t+\Delta t}^{k+1} - \hat{T}_{t+\Delta t}^k$ de taille n , et donc peut être résolu en complexité algorithmique indépendante de N , sa construction nécessite des intégrations sur le domaine Ω et la surface $\partial\Omega_R$. Ainsi, en plus de la construction de la base d'ordre réduit $\{ \Phi_j \}_{1 \leq j \leq n}$, l'étape offline doit contenir un traitement additionnel assurant la construction online efficace de (3.8), comme détaillé ci-dessous. Cette étape est souvent appelée hyperreduction.

Le speedup du modèle réduit est défini comme le rapport entre le temps de calcul du modèle haute-fidélité (3.83) et celui du modèle réduit (3.8).

Étape offline

La première partie consiste au calcul de la base d'ordre réduit $(\psi_i)_{1 \leq i \leq n}$. Nous utilisons la Snapshot POD, détaillée dans l'Algorithm 3 présenté dans la Section 3.5.3. La deuxième partie est l'hyperreduction, pour assurer une construction online efficace de (3.8).

Dans cette section, nous comparons l'EIM et la DEIM pour l'étape d'hyperreduction, comme décrit ci-dessous.

Empirical Interpolation Method

L'EIM a été présenté dans l'introduction, et la phase d'apprentissage offline est détaillée dans l'Algorithm 1. Considérons le premier terme dans (3.9). Nous disposons de la collection de snapshots T_s et des champs d'énergie interne volumique $u(T_s)$, $1 \leq s \leq N_c$. En appliquant EIM sur $\{1, \dots, N_c\} \times \Omega \ni (s, x) \mapsto u(T_s)(x) \in \mathbb{R}$, nous obtenons :

$$u(T_s)(x) \approx \sum_{l,m=1}^{d^u} D_{l,m}^u u(T_{s_l^u})(x) u(T_s)(x_m^u), \quad (3.11)$$

où l'échantillonnage Ω_{sample} pour Ω est la collection des nœuds du maillage et les exposants \cdot^u réfèrent à la quantité étant approchée. En appliquant EIM sur tous les termes requis, le membre de droite de (3.8) peut être

approché efficacement par

$$\begin{aligned} \hat{\mathcal{F}}_j(\hat{T}_{t+\Delta t}^k) &\approx \sum_{m=1}^{d^u} \left(u(\hat{T}_{t+\Delta t}^k)(x_m^u) - u(\hat{T}_t^k)(x_m^u) \right) Q_{m,j}^u + \sum_{m=1}^{d^\lambda} \lambda(\hat{T}_{t+\Delta t}^k)(x_m^\lambda) \sum_{i=1}^n \left(\hat{T}_{t+\Delta t}^k \right)_i Q_{m,i,j}^\lambda \\ &\quad - \sum_{m=1}^{d^{\text{quad}}} \left(\hat{T}_{t+\Delta t}^k \right)^4 (x_m^{\text{quad}}) Q_{m,j}^{\text{quad}} + \int_{\partial\Omega_R} \sigma \epsilon \hat{T}_0^4 \Phi_j, 1 \leq j \leq n, \end{aligned} \quad (3.12)$$

où $Q_{m,j}^u := \sum_{l=1}^{d^u} D_{l,m}^u \frac{1}{\Delta t} \int_{\Omega} u(T_{s_l^u}) \Phi_j$, $Q_{m,i,j}^\lambda := \sum_{l=1}^{d^\lambda} D_{l,m}^\lambda \int_{\Omega} \lambda(T_{s_l^\lambda}) \nabla \Phi_i \cdot \nabla \Phi_j$ et $Q_{m,j}^{\text{quad}} := \sum_{l=1}^{d^{\text{quad}}} D_{l,m}^{\text{quad}} \int_{\partial\Omega_R} \sigma \epsilon T_{s_l^{\text{quad}}}^4 \Phi_i \Phi_j$, avec les exposants quad signifiant que EIM a été appliqué sur la fonction $\{1, \dots, N_c\} \times \Omega \ni (s, x) \mapsto T_s^4(x) \in \mathbb{R}$. De même, l'opérateur linéaire dans le même de gauche de (3.8) peut être approché par

$$\begin{aligned} \left(\frac{D\hat{\mathcal{F}}}{D\hat{T}} \right)_{i,j} \left(\hat{T}_{t+\Delta t}^k \right) &\approx \sum_{m=1}^{d^c} c(\hat{T}_{t+\Delta t}^k)(x_m^c) Q_{m,i,j}^c + \sum_{m=1}^{d^\lambda} \lambda(\hat{T}_{t+\Delta t}^k)(x_m^\lambda) Q_{m,i,j}^\lambda \\ &\quad - \sum_{m=1}^{d^{\text{triple}}} \left(\hat{T}_{t+\Delta t}^k \right)^3 (x_m^{\text{triple}}) Q_{m,i,j}^{\text{triple}}, 1 \leq i, j \leq n, \end{aligned} \quad (3.13)$$

où $Q_{m,i,j}^c := \sum_{l=1}^{d^c} D_{l,m}^c \int_{\Omega} c(T_{s_l^c}) \Phi_i \Phi_j$ et $Q_{m,i,j}^{\text{triple}} := \sum_{l=1}^{d^{\text{triple}}} D_{l,m}^{\text{triple}} \int_{\partial\Omega_R} 4\sigma \epsilon T_{s_l^{\text{triple}}}^3 \Phi_j$, avec l'exposant triple signifiant que EIM a été appliqué sur la fonction $\{1, \dots, N_c\} \times \Omega \ni (s, x) \mapsto T_s^3(x) \in \mathbb{R}$.

Notons que toutes les opérations en complexité algorithmique dépendant de N concernent les termes $Q_{m,j}^u$, $Q_{m,i,j}^\lambda$, $Q_{m,j}^{\text{quad}}$, $Q_{m,i,j}^c$ et $Q_{m,i,j}^{\text{triple}}$, qui sont indépendants des valeurs courantes de paramètres et temps : ils peuvent être précalculés dans la phase offline.

Discrete Empirical Interpolation Method

Dans DEIM, une première POD est calculée avant que l'Algorithme 1 ne soit exécuté : considérons une base POD $\left\{ \Psi_k^f \right\}_{1 \leq k \leq d}$ telle que $f(x, y) \approx \sum_{k=1}^d \left(f(x, \cdot), \Psi_k^f \right)_{L^2(\mathcal{D}_y)} \Psi_k^f(y)$. Ensuite, l'Algorithme 1 est appliqué à la fonction $\{1, \dots, d\} \times \mathcal{D}_y \ni (k, y) \mapsto \Psi_k^f(y) \in \mathbb{R}$. Comme la première variable est déjà discrétisée, nous n'avons besoin que d'un échantillonnage $\mathcal{D}_{y\text{sample}}$ pour \mathcal{D}_y . Notons que puisque $\dim \left(\text{Span} \left(\Psi_k^f \right)_{1 \leq k \leq d} \right) = d$, nous pouvons construire une approximation avec d termes : $\Psi_k^f(y) = \sum_{l,m=1}^d \bar{D}_{l,m} \Psi_{\zeta(l)}^f(y) \Psi_k^f(y_m)$, où y_m sont des valeurs prises dans $\mathcal{D}_{y\text{sample}}$, ζ est une permutation de $\{1, \dots, d\}$ et $\bar{D} = \bar{F}^{-T}$ avec $\bar{F}_{l,m} = \Psi_{\zeta(l)}^f(y_m)$. Ainsi, cette fois, la compression de données n'est pas pilotée par un critère d'arrêt sur l'algorithme glouton, mais par une tolérance sur la compression de la POD. En utilisant cette approximation dans la décomposition de f sur la base POD $\left\{ \Psi_k^f \right\}_{1 \leq k \leq d}$, nous obtenons :

$$\begin{aligned} f(x, y) &\approx \sum_{l,m=1}^d \bar{D}_{l,m} \Psi_{\zeta(l)}^f(y) \left(\sum_{k=1}^d \left(f(x, \cdot), \Psi_k^f \right)_{L^2(\mathcal{D}_y)} \Psi_k^f(y_m) \right) \\ &\approx \sum_{l,m=1}^d \bar{D}_{l,m} \Psi_{\zeta(l)}^f(y) f(x, y_m). \end{aligned} \quad (3.14)$$

Bien que cette approximation puisse être utilisée sur les mêmes objets que dans la section précédente, la DEIM a été initialement proposée dans la littérature pour l'approximation des quantités intégrées directement. Par exemple, considérons à nouveau le premier terme dans (3.9). Nous disposons de la collection de snapshots T_s et des termes assemblés $\mathcal{F}_{s,i}^u := \frac{1}{\Delta t} \int_{\Omega} u(T_s) \varphi_i$, $1 \leq i \leq N$, $1 \leq s \leq N_c$. En utilisant la snapshot POD (Algorithme 1) avec la matrice de corrélation $C_{r,s}^{\bar{u}} := \sum_{i=1}^N \mathcal{F}_{r,i}^u \mathcal{F}_{s,i}^u$, $1 \leq r, s \leq N_c$, nous construisons une base $\Psi_k^{\bar{u}}$, $1 \leq k \leq d^{\bar{u}}$ pour $\mathcal{F}_{s,i}^u$. Enfin, en utilisant (3.14), nous obtenons l'approximation DEIM :

$$\frac{1}{\Delta t} \int_{\Omega} u(\hat{T}_{t+\Delta t}^k) \Phi_j \approx \sum_{i=1}^N \Phi_{j,i} \frac{1}{\Delta t} \int_{\Omega} u(\hat{T}_{t+\Delta t}^k) \varphi_i \approx \sum_{i=1}^N \Phi_{j,i} \sum_{l,m=1}^{d^{\bar{u}}} \bar{D}_{l,m}^{\bar{u}} \Psi_{\zeta(l)}^{\bar{u}} \left(\frac{1}{\Delta t} \int_{\Omega} u(\hat{T}_{t+\Delta t}^k) \varphi_{i_m^{\bar{u}}} \right), 1 \leq j \leq n, \quad (3.15)$$

où $i_m^{\bar{u}}$, $1 \leq m \leq d^{\bar{u}}$, sont des valeurs prises dans $\{1, \dots, N\}$ et associées à l'approximation de $\mathcal{F}_{s,i}^u$. Notons que bien que le terme $\frac{1}{\Delta t} \int_{\Omega} u(\hat{T}_{t+\Delta t}^k) \varphi_{i_m^{\bar{u}}}$ doit être évalué pendant la phase online, l'intégrale peut être évaluée en complexité algorithmique indépendante de N . Soit $m \in \{1, \dots, d^{\bar{u}}\}$, l'union des supports des fonctions de base éléments finis qui sont non nulles au nœud $i_m^{\bar{u}}$ correspond à l'union des éléments qui partagent le nœud $i_m^{\bar{u}}$: notons cet ensemble $\mathcal{S}_m^{\bar{u}}$. L'intégrale sur Ω est alors remplacée par une intégrale sur $\mathcal{S}_m^{\bar{u}}$. L'union de tous les domaines locaux $\mathcal{S}_m^{\bar{u}}$ est appelée le domaine d'intégration réduit.

La même procédure est appliquée pour l'approximation de chaque terme de l'opérateur tangent (3.10) après réorganisation des matrices en grands vecteurs, comme proposé dans [26] avec la Matrix-DEIM.

Expériences numériques et stabilité

Nous renvoyons au Rapport [Rap.2](#) pour une présentation exhaustive des résultats.

Implémentation

Pour résoudre le problème haute-fidélité (HFM) (3.4), le solveur éléments-finis Z-set [90] est utilisé, et une nouvelle condition aux limites de radiation dépendant du temps a été implémentée dans ce solveur pour calculer les termes $\int_{\partial\Omega_R} \sigma \epsilon (T_{t+\Delta t}^4 - T_0^4) \varphi_j$ et $\int_{\partial\Omega_R} \sigma \epsilon T_{t+\Delta t}^3 \varphi_i \varphi_j$ dans (3.6) avec une frontière $\partial\Omega_R$ dépendant du temps pour modéliser la translation du four. L'étape offline a été implémentée par modification des routines d'assemblage de Z-set, et le modèle réduit (ROM) (3.8) est exécuté par une librairie maison en python. Pour l'EIM, l'assemblage du ROM repose sur une librairie python externe, mais pour la DEIM, les termes intégrés online sont calculés avec des appels de routines d'assemblage de Z-set. Comme nous n'avons pas programmé de sessions interactives de Z-set, chaque intégration online avec DEIM nécessite de lancer une nouvelle session de Z-set, ce qui pénalise fortement les speedups.

Pour des besoins de visualisation et d'exploitation, un plugin paraview [5] a été développé sous la forme d'un "programmable filter", où les paramètres peuvent être modifiés directement par l'utilisateur dans l'interface, pour obtenir une mise à jour des champs réduits solutions en temps réel dans la fenêtre de visualisation.

Comparaison EIM/DEIM sans translation

Nous considérons un modèle simplifié d'aube de turbine haute pression, avec $N = 3,296$, où nous rappelons que N est le nombre de degrés de liberté de la solution du HFM, voir Figure 3.6. Nous résolvons (3.3a), avec une condition initiale (3.3d). Les conditions aux limites (3.3b)-(3.3c) sont remplacées par une condition aux limites de Neumann : un flux constant sur les surfaces des conduits de refroidissement internes et un flux nul sur le reste de la frontière. L'évolution transitoire est calculée pendant 20 secondes, discrétisées en 20 pas de temps. La base d'ordre réduit est construite à partir de cette résolution et le ROM est testé sur cette même configuration. La comparaison d'EIM avec DEIM est présentée dans la Table 3.1, et le domaine d'intégration réduite construit par DEIM est illustré dans la Figure 3.7.

	sans AHC		avec AHC	
temps HFM	124 s		174 s	
hyperréduction	EIM	DEIM	EIM	DEIM
temps offline	39 s	85 s	51 s	143 s
longueur decomp.	6 / 11	10 / 12	18 / 19	19 / 20
speedup	8×10^3	3.9	8×10^3	2.8
erreur	2×10^{-3}	2×10^{-2}	2×10^{-3}	68

TABLE 3.1 – cas test aube HP-Tu : comparaison des performances EIM et DEIM ; longueur decomp. : les deux nombres correspondent aux approximations du membre de droite et de l'opérateur linéaire de (3.8) respectivement

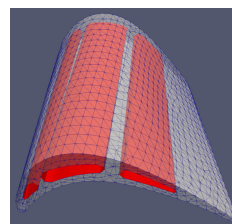


FIGURE 3.6 – cas test aube HP-Tu : maillage, canaux de refroidissement internes en rouge

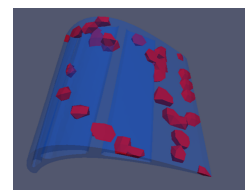


FIGURE 3.7 – cas test aube HP-Tu : domaine d'intégration réduite de la DEIM

Nous notons d'abord que avec AHC, les non-linéarités sont plus fortes et le temps de résolution du HFM est plus long, ceci étant dû à une convergence plus lente de l'algorithme de Newton. EIM et DEIM sélectionnent également plus de termes avec AHC. Notons que EIM conduit à un meilleur speedup dans les deux cas. En particulier avec AHC, DEIM conduit à des erreurs importantes.

Réductibilité, précision et stabilité en présence de translation

Nous sommes en présence des phénomènes suivants :

[P1] Il est connu que la POD a des performances limitées en présence de forts effets de convection : la matrice de corrélation construite par la snapshot POD (Algorithm 3) n'est pas réductible, et un nombre important de modes sont sélectionnés. Des solutions ont été proposées, par exemple dans [29] où les snapshots, sujets à des effets de convection importants, sont transformés par des changements de variable de sorte que leur combinaison linéaire conduit à une expression plus précise et compacte pour la solution réduite, en présence de conditions aux limites de périodicité. Des stratégies utilisant des éléments similaires ont été proposées dans [22, 29, 70, 93].

Nous proposons d'améliorer la réductibilité en utilisant des bases POD locales, comme introduites dans [11, 114] : les snapshots sont classifiés et regroupés en clusters selon une certaine notion de similarité, où la dimension peut être réduite. Dans notre cas, la translation est connue, et les clusters de snapshots similaires sont simplement construits en les regroupant par position proche du four. Dans la phase online, le ROM va automatiquement changer de base locale en fonction de la position du four de façon efficace grâce au précalcul des produits scalaires $(\Phi_i^A, \Phi_j^B)_{1 \leq i \leq n^A, 1 \leq j \leq n^B}$, pour toutes bases locales adjacentes Φ_i^A et Φ_i^B , $1 \leq i \leq n^A$, $1 \leq i \leq n^B$. L'utilisation de bases locales améliore le speedup, car chaque base locale contient un petit nombre de vecteurs de base et le système linéaire (3.8) à résoudre online est plus petit. Cependant, cela conduit à une plus mauvaise compression des données ; dans le cas extrême d'autant de bases locales que de snapshots, aucune compression de données n'est obtenue : toute l'information contenue dans les snapshots est conservée.

[P2] L'EIM a une mauvaise précision lorsque l'approximation est appelée loin de l'échantillonnage utilisé pendant la phase d'apprentissage, en particulier en régime d'extrapolation. Dans l'Algorithme 1, nous avons supposé que nous disposions de la valeur de la fonction $f(x, y)$ à approcher dans des sous-ensembles $\mathcal{D}_{x_{\text{sample}}}$ et $\mathcal{D}_{y_{\text{sample}}}$, suffisamment grands pour capturer le comportement de f sur les domaines continus complets \mathcal{D}_x et \mathcal{D}_y . Lorsque nous appliquons EIM pour approcher le terme non-linéaire $u(T_s)(x)$ dans (3.11), nous utilisons comme échantillonnage de la variable spatiale la collection des nœuds du maillage du HFM, ce que nous pouvons supposer adapté à la représentation de $u(T_s)(x)$. Cependant, nous utilisons seulement N_c valeurs pour le paramètre et le temps (correspondant aux snapshots T_s , $1 \leq s \leq N_c$), ce qui représente l'ensemble des snapshots à notre disposition. Nous ne pouvons faire mieux, puisque supposer un échantillonnage riche signifierait résoudre le HFM pour un grand nombre de valeurs de paramètres, ce qui n'est pas raisonnable dans notre contexte de réduction de modèle. En pratique, cet échantillonnage est insuffisant et un nouveau paramètre considéré dans la phase online peut conduire à évaluer l'approximation EIM en dehors de l'échantillonnage utilisé dans la phase d'apprentissage, et potentiellement en régime d'extrapolation.

Lorsque le ROM est résolu, l'approximation est appelée pour la solution courante $\hat{T}_{t+\Delta t}^k$. Comme $\hat{T}_{t+\Delta t}^k \in \text{Span}\{\Phi\}_{1 \leq i \leq n}$ par construction, nous proposons l'enrichir le domaine d'apprentissage d'EIM en construisant des valeurs test de T prises dans $\text{Span}\{\Phi\}_{1 \leq i \leq n}$. En pratique, nous prenons quelques perturbations aléatoires des valeurs de T rencontrées pendant les résolutions du HFM dans $\text{Span}\{\Phi\}_{1 \leq i \leq n}$, comme fait dans l'Exposé Exp.19.

[P3] A l'instar de toute méthode d'interpolation, EIM peut être sujet au sur-apprentissage. Nous observons dans nos simulations numériques que lorsqu'une grande précision est imposée dans l'étape gloutonne d'EIM, le conditionnement de la matrice B in (5) devient important, et la valeur absolue des coefficients λ de l'approximation (4) devient grande. Nous référons à l'introduction de [97] pour une revue de références traitant ce problème, où les solutions apportées consistent à prendre plus de points EIM pour une variable que pour l'autre et en remplaçant le système linéaire par un moindre carré. Même si ce phénomène est intrinsèque à la méthode EIM, et qu'une bonne convergence de l'algorithme glouton conduira à un système linéaire online mal conditionné, nous avons tout de même proposé une meilleure construction numérique de l'opérateur B dans la Publication Pub.18. Dans [14], les auteurs proposent de borner les coefficients de la décomposition EIM (les λ de l'approximation (4)), en fonction de leur comportement dans la phase *offline* de l'algorithme EIM.

Nous proposons d'appliquer des remèdes classiques du sur-apprentissage : un critère de convergence moins restrictif dans l'algorithme glouton et remplacer le système linéaire de la phase online d'EIM par une minimisation avec régularisation de Tikhonov (reg.-LS).

$$\operatorname{argmin}_{w \in \mathbb{R}^d} \|Fw - b\|_2 + \alpha \|w\|_2, \quad (3.16)$$

une minimisation aux moindres carrés sous contraintes d’une solution à composantes positives (NNLS)

$$\operatorname{argmin}_{w \in \mathbb{R}^+{}^d} \|Fw - b\|_2. \quad (3.17)$$

Dans la suite, nous présentons de façon non-exhaustive des expériences numériques illustrant les problèmes de stabilité décrits ci-dessus en présence de non-linéarités fortes de la méthode AHC et des effets de translation. Entin, une application est présentée sur le moulage d’une aube de turbine basse-pression (Tu-LP).

Expériences numériques sur la stabilité du ROM POD-EIM en présence de translation

Nous considérons deux expériences : le soudage de plaques métalliques et le moulage d’une barre métallique, voir Figure 3.8. Dans le premier cas, le problème (3.3a)-(3.3d) est considéré, tel que $N = 1698$, avec 200 pas de temps et un paramètre : la température du four. Dans le deuxième cas, une condition aux limites de Neumann avec un flux positif constant est imposé sur la projection du laser mobile sur les plaques et une condition aux limites de radiation (3.3c) est imposée sur le reste de la frontière, avec $N = 1764$, 200 pas de temps et un paramètre : la puissance du laser. Dans les deux cas, si cela n’est pas mentionné autrement, la capacité thermique volumique c_1 (avec AHC) et la conductivité thermique λ définis dans la Figure 3.5 sont utilisées ; les snapshots considérés dans la phase offline correspondent aux résolutions du HFM à deux valeurs distinctes du paramètre, et le ROM est ensuite testé à la valeur paramétrique moyenne.

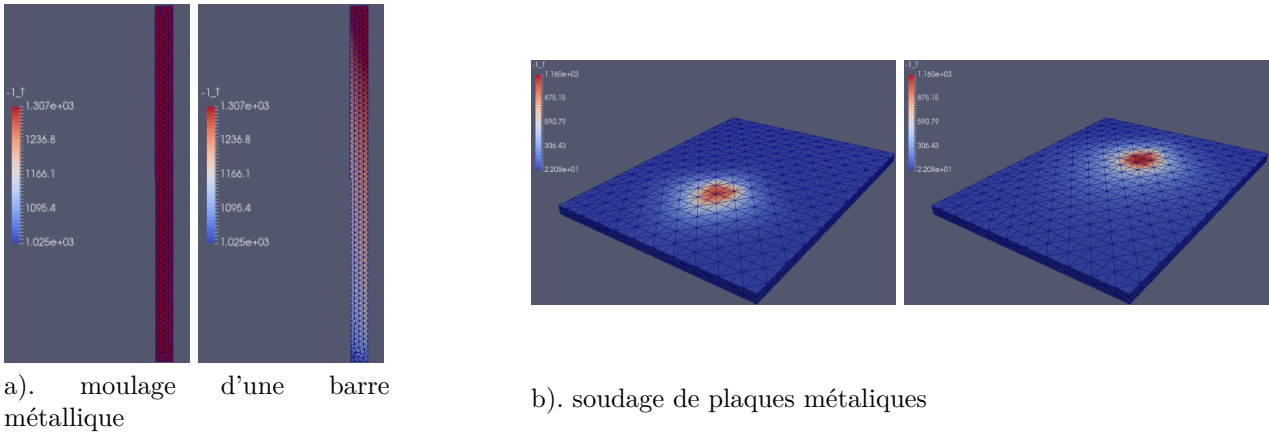


FIGURE 3.8 – Champs de température au début et à la fin des processus de moulage et soudage

Expérience 1 : moulage d’une barre métallique

La Figure 3.9 présente une comparaison de la précision du ROM sans hyperréduction, et quand EIM est utilisé sans régularisation et avec les régularisations NNLS et reg.-LS avec $\alpha = 0.01$ dans (3.16). Les intervalles paramétriques sont centrés sur 1320°C pour la température du four. Avec ce choix, la barre métallique est liquide au début de l’expérience et refroidie en deçà de la température de fusion. Nous notons que sans hyperréduction, la précision du ROM ne dépend pas beaucoup de la taille de l’intervalle paramétrique. Au contraire, avec EIM et ses stabilisations, plus l’intervalle paramétrique est grand, moins le ROM est précis. Avec AHC, les effets non-linéaires sont plus forts, et la précision du cas de référence sans hyperréduction est environ un ordre de grandeur plus mauvaise que sans AHC. Notons que avec AHC et l’EIM classique, seul le cas où le ROM calcule la même configuration que le HFM est stable. La stabilisation ne semble pas dégrader significativement la précision, puisque le ROM avec AHC reste un ordre de grandeur moins précis que sans AHC, à l’exception de reg.-LS pour les petites tailles d’intervalle paramétrique.

Dans une autre expérience, avec des tailles d’intervalle paramétriques de respectivement 40 et 80, avec AHC et l’EIM classique, le ROM est stable pour un critère d’arrêt de 10^{-3} pour EIM, et instable pour 10^{-4} , 10^{-5} et 10^{-6} .

Expérience 2 : soudage de plaques métalliques

Dans ce cas, nous considérons un enrichissement d’EIM avec 3 vecteurs additionnels par snapshot, obtenus comme modifications aléatoires des coefficients de la décomposition sur la base POD avec une amplitude de 10%. Plus précisément, prenons un snapshot T_s et projetons-le sur la base POD comme $\sum_{i=1}^n (T_s, \Phi_i) \Phi_i$. Un nouveau snapshot est généré comme $\sum_{i=1}^n \gamma_i (T_s, \Phi_i) \Phi_i$, où γ_i sont des tirages aléatoires d’une distribution uniforme sur $(0.95, 1.05)$. Avec une précision de la POD et de EIM fixés à 10^{-5} et des expériences avec respectivement 1 base POD globale et 4 bases POD globales, l’EIM est instable, tandis que l’EIM enrichi avec nos choix est stable,

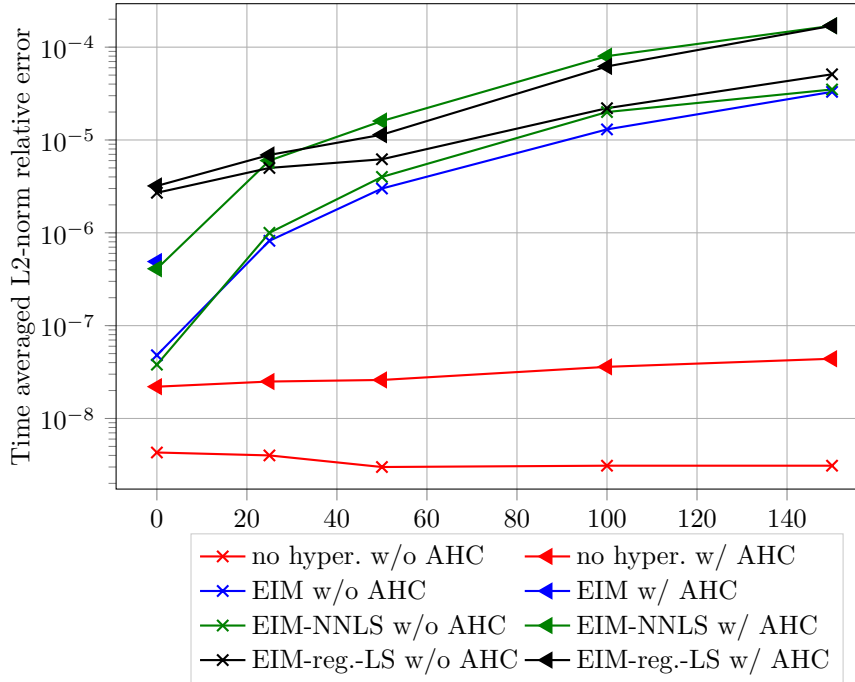


FIGURE 3.9 – Erreur du ROM par rapport au HFM avec et sans AHC, pour différentes techniques d’hyperréduction et différentes tailles d’intervalles paramétriques

avec des speedups de respectivement 10^2 et 1.4×10^2 . Notons qu’un EIM est calculé pour chaque POD locale, de sorte que l’hyperréduction soit dédiée à chaque variabilité locale.

En conclusion, dans notre contexte d’équation de la chaleur transitoire avec AHC et des effets de translation, la correction [P1] conduit à un meilleur speedup, et si le ROM devient instable, des modifications de EIM peuvent améliorer la stabilité : enrichir l’ensemble d’apprentissage d’EIM [P2] ou limiter le sur-apprentissage d’EIM [P3] avec un critère d’arrêt prématuré ou remplacer le système linéaire dans les évaluations online d’EIM par un moindre carré avec régularisation de Tikhonov ou un moindre carré à poids positifs.

Un ROM POD-EIM paramétrique pour le soudage d’une aube de turbine basse pression industrielle

Nous considérons une aube de turbine basse pression industrielle, dont le maillage est représenté dans la Figure 3.10, et le problème (3.3a)-(3.3d) avec $N = 25062$ et 200 pas de temps. Les solutions du HFM sont calculées en 3245 s. Les variabilités sont modélisées par 4 paramètres : trois pour la description de la capacité thermique volumique $c_p(T) = 3.7 \times 10^6 \mu_1 + 3.5 \times 10^5 \mu_2 \arctan(0.01 \mu_3 (T - 500))$, et un paramètre μ_4 pour la température du four. Les snapshots sont générés avec quatre résolutions HFM aux valeurs paramétriques $(\mu_1, \mu_2, \mu_3, \mu_4)$: $(0.95, 0.95, 1.15, 1282.5)$, $(1.15, 1.15, 1.05, 1307.5)$, $(0.85, 1.05, 0.85, 1332.5)$ et $(1.05, 0.85, 0.95, 1357.5)$. Nous illustrons dans la Figure 3.10 que lorsqu’une base POD globale est utilisée, les effets de translation conduisent à des modes oscillants et une réduction limitée. En utilisant 200 bases POD locales, le ROM est exécuté à la valeur paramétrique $(1, 1, 1, 1320)$ avec l’EIM classique. Utiliser autant de bases POD locales que de pas de temps revient à ne considérer que les corrélations paramétriques dans la snapshot POD, au lieu de considérer les corrélations paramétriques et temporelles avec une base unique. Le speedup est 1.4×10^4 et l’erreur est vérifiée à 10^{-4} . Une comparaison du ROM et du HFM est illustrée dans la Figure 3.11.

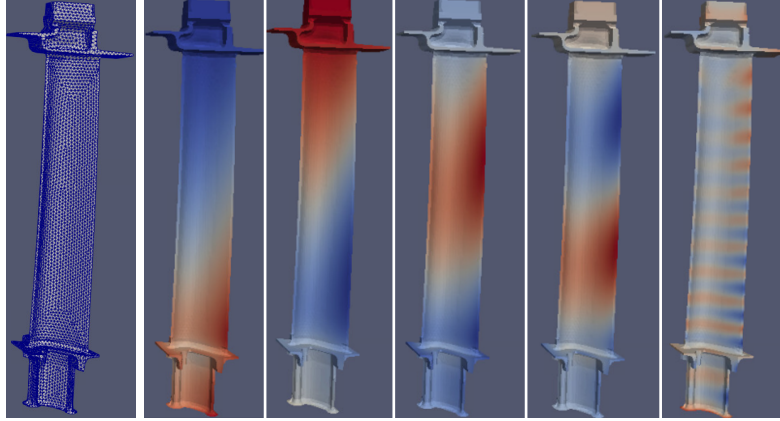


FIGURE 3.10 – De gauche à droite : maillage de l’aube Tu-LP ; 1^{er}, 2^{eme}, 3^{eme} et 31^{eme} mode POD (base globale)

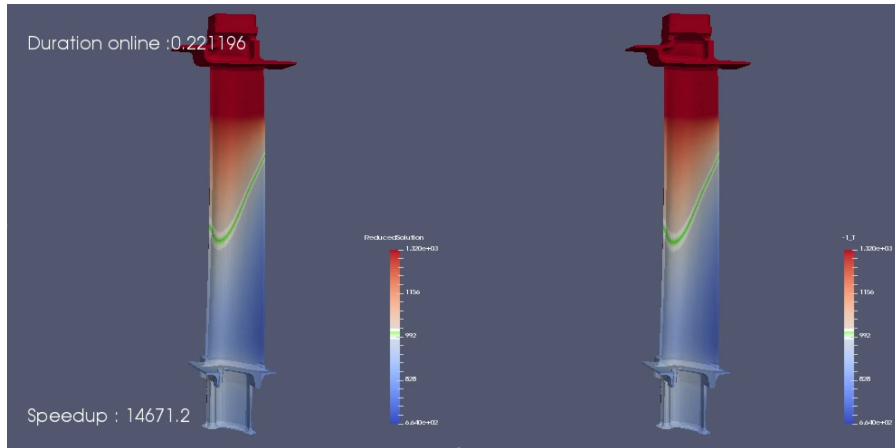


FIGURE 3.11 – Gauche : ROM-POD EIM avec 200 bases POD locales ; droite : solution de référence du HFM

3.3 Variantes de l’Empirical Interpolation Method : formulation symétrique, choix des normes et extension rectangulaire

Cette section correspond à la Publication [Pub.12](#).

Dans cette section, nous proposons une définition d’EIM équivalente à la définition classique, mais pour laquelle les deux variables de la fonction approchée jouent un rôle symétrique. Ensuite, nous donnons une preuve pour l’existence de cette approximation, et obtenons la convergence d’EIM, pour toute norme utilisée pour calculer l’erreur dans l’algorithme glouton. Enfin, nous étendons la présentation aux cas où le nombre de points sélectionnés pour chaque variable est différent, cas correspondant classiquement à la Gappy-POD. Dans le cas d’un champ physique mesuré par des capteurs, il peut être utile de prendre en compte le cas d’un capteur défectueux tout en gardant l’information fournie par le champ associé.

Comme présenté dans l’introduction, dans la phase offline d’EIM appliqué à l’approximation d’une fonction $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ sont sélectionnés des points $(x_1, \dots, x_d) \in \mathcal{X}^d$ et $(y_1, \dots, y_d) \in \mathcal{Y}^d$ de façon gloutonne, tels que

$$x_{k+1} = \arg \max_{x \in \mathcal{X}} \| (f - I_k(f))(x, \cdot) \|_{L^\infty(\mathcal{Y})}, \quad (3.18a)$$

$$y_{k+1} = \arg \max_{y \in \mathcal{Y}} | (f - I_k(f))(x_{k+1}, y) |, \quad (3.18b)$$

où $I_k(f)$ est l’approximation construite à partir des k premiers points (x_l, y_l) , $l \in \{1, \dots, k\}$. La méthode est efficace quand l’erreur $f - I_d(f)$ est petite pour des valeurs raisonnablement faibles de d . Nous rappelons que

la fonction $I_d(f)$ vérifie la propriété d'interpolation [83, Lemma 1] : pour tout $m \in \{1, \dots, d\}$,

$$I_d(f)(x, y_m) = f(x, y_m), \quad \text{for all } x \in \mathcal{X}, \quad (3.19a)$$

$$I_d(f)(x_m, y) = f(x_m, y), \quad \text{for all } y \in \mathcal{Y}. \quad (3.19b)$$

Soit $\mathcal{U} := \{f(x, \cdot), x \in \mathcal{X}\}$. Les éléments de \mathcal{U} sont des fonctions de \mathcal{Y} dans \mathbb{R} .

Théorème 3.1 (Existence de la décomposition, [83, Theorem 1]). *Supposons que les points d'interpolation soient choisis selon (3.18a)-(3.18b) et que $d < \dim \text{Vect}(\mathcal{U})$. Alors, la représentation à variables séparées $I_d(f)$ est bien définie.*

Nous avons montré par l'Equation (6) que $I_d(f)$ peut être réécrit dans la forme équivalente

$$I_d(f)(x, y) = \sum_{1 \leq l, m \leq d} D_{l,m} f(x_l, y) f(x, y_m), \quad (3.20)$$

où la matrice D dépend des points $x_l, y_m, l, m \in \{1, \dots, d\}$. La propriété d'interpolation (3.19a)-(3.19b) est vérifiée si et seulement si $D = F^{-T}$, où $F_{l,m} = f(x_l, y_m)$, ce qui motive une présentation alternative d'EIM, reposant sur l'Équation (3.20) où les variables x et y remplissent un rôle symétrique. La double somme dans (3.20) souligne ce rôle symétrique; rappelons que, par exemple, $I_d(f)(x, y) = \sum_{1 \leq l \leq d} \tilde{\lambda}_l(x) \tilde{q}_l(y)$, avec $\tilde{\lambda}_l(x) = \sum_{1 \leq m \leq d} D_{l,m} f(x, y_m)$ et $\tilde{q}_l(y) = f(x_l, y)$.

Soit $\|\cdot\|_{\mathcal{Y}}$ une norme sur \mathcal{Y} et supposons que les points d'interpolation sont maintenant sélectionnés comme

$$x_{k+1} = \arg \max_{x \in \mathcal{X}} \|(f - I_k(f))(x, \cdot)\|_{\mathcal{Y}}, \quad (3.21a)$$

$$y_{k+1} = \arg \max_{y \in \mathcal{Y}} |(f - I_k(f))(x_{k+1}, y)|, \quad (3.21b)$$

la différence avec (3.18a)-(3.18b) étant le choix arbitraire pour la norme $\|\cdot\|_{\mathcal{Y}}$ dans la première ligne, à la place de $\|\cdot\|_{L^\infty(\mathcal{Y})}$. Il est possible d'échanger les rôles de x et y dans l'algorithme précédent, conduisant à

$$y_{k+1} = \arg \max_{y \in \mathcal{Y}} \|(f - I_k(f))(\cdot, y)\|_{\mathcal{X}}, \quad (3.22a)$$

$$x_{k+1} = \arg \max_{x \in \mathcal{X}} |(f - I_k(f))(x, y_{k+1})|, \quad (3.22b)$$

pour une norme arbitraire $\|\cdot\|_{\mathcal{X}}$ sur \mathcal{X} . En général, le couple (x_{k+1}, y_{k+1}) résultant de (3.22a)-(3.22b) diffère de celui obtenu avec (3.21a)-(3.21b). En choisissant les normes L^∞ sur \mathcal{Y} et \mathcal{X} dans (3.21a) et (3.22a) respectivement, nous obtenons :

$$(x_{k+1}, y_{k+1}) = \arg \max_{(x,y) \in \mathcal{X} \times \mathcal{Y}} |(f - I_k(f))(x, y)|,$$

retrouvant ainsi le choix fait dans (3.18a)-(3.18b). Pour ce choix spécifique de normes L^∞ sur \mathcal{Y} et \mathcal{X} , (3.21a)-(3.21b) est équivalent à (3.22a)-(3.22b).

Théorème 3.2. *Supposons que les points d'interpolation sont choisis selon (3.21a)-(3.21b) ou (3.22a)-(3.22b) et que $d \leq \dim \text{span}(\mathcal{U})$. Alors, la représentation séparée (3.20) avec $D = F^{-T}$, où $F_{l,m} = f(x_l, y_m)$, est bien définie et vérifie (3.19a)-(3.19b).*

Le Théorème 3.2 étend le Théorème 3.1 de deux façons. Premièrement, il permet une sélection plus générale des points d'interpolation. Deuxièmement, l'existence de la décomposition est assurée jusqu'à convergence. Comme $\dim \text{Vect}(\mathcal{U})$ est en général inconnu, une conséquence pratique est que pour tout $k \in \mathbb{N}$, soit $f(x, y) = I_k(f)(x, y)$ pour tout $(x, y) \in \mathcal{X} \times \mathcal{Y}$, soit la procédure gloutonne peut être poursuivie. De plus, la preuve que nous donnons pour le Théorème 3.2 est relativement simple et ne repose pas explicitement sur une épaisseur de Kolmogorov. Notre présentation est compatible avec la Generalized Empirical Interpolation Method (GEIM) [81], et dans le cas où la valeurs de f à certains points y_i ne sont plus fiables (défaillances de capteurs par exemple), le champ $f(x_l, y)$ peut être conservé, conduisant à une formulation dite rectangulaire, où le nombre de points sélectionnés n'est pas le même pour les deux variables. Nous reconnaissons une ressemblance avec la Gappy-POD.

Nous énonçons quelques résultats intermédiaires intéressants, sans en reproduire la démonstration, utilisée pour la preuve du Théorème 3.2. Considérons les points d'interpolation $(x_1, \dots, x_d) \in \mathcal{X}^d$ et $(y_1, \dots, y_d) \in \mathcal{Y}^d$, et définissons la matrice $F \in \mathbb{R}^{d \times d}$ telle que $F_{l,m} = f(x_l, y_m)$, pour tout $l, m \in \{1, \dots, d\}$.

Lemme 3.1 (Existence). *Si la matrice F est inversible, alors il existe une fonction $I_d(f) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ de la forme (3.20) satisfaisant la propriété d'interpolation (3.19a)-(3.19a).*

Notons que l'interpolation EIM est ainsi donnée par $I_d(f)(x, y) = \sum_{1 \leq l, m \leq d} (F^{-1})_{m,l} f(x_l, y) f(x, y_m)$.

Lemme 3.2 (Critère d'arrêt). *Considérons un ensemble de points $(x_l, y_m)_{1 \leq l, m \leq k+1}$ et supposons que la matrice $F^k := (f(x_l, y_m))_{1 \leq l, m \leq k}$ est inversible. Si*

$$(f - I_k(f))(x_{k+1}, y_{k+1}) \neq 0,$$

alors la matrice $F^{k+1} := (f(x_l, y_m))_{1 \leq l, m \leq k+1}$ est aussi inversible.

À partir de (3.21a)-(3.21b) ou (3.22a)-(3.22b), nous déduisons que pour tout $k < \dim \text{span}(\mathcal{U})$, $(f - I_k(f))(x_{k+1}, y_{k+1}) \neq 0$.

Corollaire 3.1 (Terminaison). *Si (x_{k+1}, y_{k+1}) est sélectionné selon (3.21a)-(3.21b) ou (3.22a)-(3.22b), alors $|(f - I_k(f))(x_{k+1}, y_{k+1})| = 0$, implique que $(f - I_k(f))(x, y) = 0$ pour tout $(x, y) \in \mathcal{X} \times \mathcal{Y}$.*

La conséquence pratique des résultats ci-dessous est que, pour tout $k \in \mathbb{N}$, soit $|(f - I_k(f))(x_{k+1}, y_{k+1})| = 0$ et $f = I_k(f)$ (Corollaire 3.1), ou I_{k+1} peut être construit (Lemme 3.2) et la procédure gloutonne peut être poursuivie.

Remarque 3.1 (Approximation goal-oriented). *Nous avons vu que dans un contexte de bases réduite, la procédure EIM pour être utilisée pour obtenir une représentation approchée d'une fonction de deux variables, pour retrouver une hypothèse de dépendance affine des opérateurs par rapport aux paramètres considérés, et obtenir une procédure efficace pour la phase online. Par exemple, considérons le membre de droite $f(x, y)$ d'une équation aux dérivées partielles, discrétisée en utilisant une base de Galerkin $\{\varphi_1, \dots, \varphi_N\}$. Ici, x est le paramètre et y la variable d'espace. Le membre de droite discrétisé est alors $b(x)_j = \int_{\Omega} f(x, y) \varphi_j(y) dy$, et en utilisant l'approximation EIM, il peut être réécrit comme une combinaison linéaire de fonctions de x : $(I_d(b)(x))_j = \int_{\Omega} I_d(f)(x, y) \varphi_j(y) dy = \sum_{1 \leq l, m \leq d} D_{l,m} f(x, y_m) \int_{\Omega} f(x_l, y) \varphi_j(y) dy$. Considérons un sous-espace de dimension finie d'un espace de Hilbert ayant comme base $\{\varphi_1, \dots, \varphi_N\}$. Définissons $b(x)_j = \int_{\Omega} f(x, y) \varphi_j(y) dy$ pour tout $j \in \{1, \dots, N\}$, et son approximation EIM $(I_d(b)(x))_j = \int_{\Omega} I_d(f)(x, y) \varphi_j(y) dy$. Si l'approximation EIM I_d est construite en utilisant (3.21a)-(3.21b), l'erreur dans la procédure gloutonne est évaluée sur la qualité de l'approximation de $f(x, y)$. Une alternative goal-oriented est d'évaluer la qualité d'approximation de f sur l'objet d'intérêt b pour la réduction de modèle :*

$$x_{k+1} = \arg \max_{x \in \mathcal{X}} \|(b - I_d(b))(x)\|, \quad (3.23a)$$

$$y_{k+1} = \arg \max_{y \in \mathcal{Y}} |(f - I_k(f))(x_{k+1}, y)|, \quad (3.23b)$$

où $\|\cdot\|$ peut-être n'importe quelle norme sur \mathbb{R}^N .

Considérons une approximation EIM, où d couples (x_m, y_m) , $m \in \mathcal{N} := \{1, \dots, d\}$ ont été sélectionnés. Supposons maintenant que pour une quelconque raison, certains points y_{m_0} , $m_0 \in \mathcal{N}_0 \subsetneq \mathcal{N}$, doivent être ignorés. Par exemple, considérons que $f(x, y)$ représente l'évaluation d'un champ physique paramétré par x , par le moyen d'un capteur localisé y . Ignorer y_{m_0} correspondrait à la défaillance du capteur correspondant. La motivation étant la suivante : même si ce capteur défaille, nous pouvons toujours utiliser l'information champ physique associé à ce capteur dans l'approximation en représentation séparée. En s'inspirant du Lemme 3.1 nous choisissons $D = (F^T)^\dagger$, où \cdot^\dagger est la pseudo-inverse. Les propriétés d'interpolation (3.19a)-(3.19b) ne sont plus vérifiées, mais le but est bien de trouver une formule d'approximation, non pas nécessairement d'interpolation :

$$\sum_{l \in \mathcal{N}} \sum_{m \in \mathcal{N}_0} (F^T)^\dagger_{l,m} f(x_l, y) f(x, y_m). \quad (3.24)$$

Pour application numérique, fixons $\vec{v} = (1 \ 2 \ 3)^T$ et considérons la fonction $\mathbb{R}^3 \times \mathbb{R} \ni (\vec{x}, y) \mapsto f(\vec{x}, y) := \cos((\vec{v} \cdot \vec{x})y) \in (-1, 1)$. Supposons que $d = 8$ couples de points (x_k, y_k) , $k \in \mathcal{N} = \{1, \dots, d\}$, ont été sélectionnés par la procédure gloutonne selon (3.18a)-(3.18b). Considérons une paire $i, j \in \mathcal{N}$, $i \neq j$ et l'ensemble $\mathcal{N}_0 = \mathcal{N} \setminus \{i, j\}$. L'erreur relative en norme ℓ^2 sur un échantionnage de 1000 dans $(0, 1)^3 \times (0, 1)$ est calculée par les deux algorithmes suivants : (i) en utilisant la procédure EIM classique avec 6 couples de points (x_k, y_k) , $k \in \mathcal{N}_0$, et (ii) en utilisant l'approximation (3.24) avec les 6 points $\{x_k\}_{k \in \mathcal{N}_0}$ dans la dimension x -dimension et les 8 points $\{y_k\}_{k \in \mathcal{N}}$ dans la dimension y . L'erreur relative en norme ℓ^2 a été calculée pour tout $\mathcal{N}_0 = \mathcal{N} \setminus \{i, j\}$, $i, j \in \mathcal{N}$ tels

que $i \neq j$. Dans le cas (i), le maximum, minimum et erreur relative moyenne en norme ℓ^2 sont respectivement 6.8×10^{-4} , 3.0×10^{-6} , et 3.6×10^{-5} , tandis que dans le cas (ii), ces erreurs sont respectivement 2.3×10^{-5} , 7.6×10^{-7} , et 2.4×10^{-6} . Cet exemple illustre le fait que en gardant l'information associée à x_i et x_j , $i, j \in \mathcal{N}$ tel que $i \neq j$, donne une meilleure précision qu'en ignorant cette information et en utilisant l'approximation classique sur les capteurs non défaillants.

L'extension à GEIM est obtenue de la façon suivante. Considérons un ensemble \mathcal{F} de fonctions f définies sur $\Omega \subset \mathbb{R}$, et un dictionnaire Σ de formes linéaires σ définies sur \mathcal{F} . Considérons d fonctions $(f_1, \dots, f_d) \in \mathcal{F}^d$ et d formes linéaires $(\sigma_1, \dots, \sigma_d) \in \Sigma^d$, et définissons la matrice $\hat{F} = (\sigma_l(f_m))_{1 \leq l, m \leq d}$.

Lemme 3.3 (Existence for GEIM). *Supposons que \hat{F} est inversible. Alors, avec $D = \hat{F}^{-T}$, la combinaison linéaire*

$$I_d(f) = \sum_{1 \leq l, m \leq d} D_{l,m} \sigma_l(f) f_m \quad (3.25)$$

vérifie, pour tout $m \in \{1, \dots, d\}$,

$$\sigma_m(f) = \sigma_m(I_d(f)), \quad \text{pour tout } f \in \mathcal{F}, \quad (3.26a)$$

$$\sigma(f_m) = \sigma(I_d(f_m)), \quad \text{pour tout } \sigma \in \Sigma. \quad (3.26b)$$

Lemme 3.4 (Critère d'arrêt pour GEIM). *Supposons que la matrice $\hat{F}^k := (\sigma_l(f_m))_{1 \leq l, m \leq k}$ est inversible. Si $|\sigma_{k+1}(f_{k+1}) - \sigma_{k+1}(I_k(f_{k+1}))| \neq 0$, alors la matrice $\hat{F}^{k+1} := (\sigma_l(f_m))_{1 \leq l, m \leq k+1}$ est inversible.*

La généralisation du Théorème 3.2 à GEIM, en prenant n'importe quelle norme sur \mathcal{F} , est alors possible. Le contexte d'un capteur défaillant est pertinent avec GEIM : supposons que pour tout $\sigma \in \Sigma$, il existe un unique x_σ tel que $\sigma(f) = f(x_\sigma)$; les fonctions f représentent le champ physique paramétré, dépendant de x , et $\sigma(f) = f(x_\sigma)$ représente un capteur localisé au point x_σ , qui mesure la valeur de x à ce point. Le cas d'un capteur m_0 défaillant conduit à l'approximation

$$\sum_{l \in \mathcal{N}_0} \sum_{m \in \mathcal{N}} (\hat{F}^T)_{l,m}^\dagger \sigma_l(f) f_m, \quad (3.27)$$

qui devrait fournir une représentation plus précise de f que si la fonction f_{m_0} était ignorée dans le cas classique d'EIM. Nous citons [107], où une extension de ce type de décompositions au cas de fonctions multivariées est proposée.

3.4 Évaluation précise et efficace de la borne d'erreur a posteriori dans la méthode des bases réduites

Cette section correspond aux Publications Pub.18 et Pub.19, aux Exposés Exp.10 à Exp.15 et Exp.21 et au Poster Pos.1.

Nous considérons la méthode de réduction de modèle méthode des BR (bases réduites). Reprenons le formalisme utilisé dans l'introduction de ce manuscrit. Supposons que le problème d'intérêt admet la formulation variationnelle discrète suivante, dépendant du paramètre $\mu \in \mathcal{P}$: pour un espace de dimension finie \mathcal{V} de dimension $N \gg 1$, trouver $u_\mu \in \mathcal{V}$ tel que

$$E_\mu : a_\mu(u_\mu, v) = b(v), \quad \forall v \in \mathcal{V}, \quad (3.28)$$

où a_μ est une forme sesquilinéaire inf-sup stable sur $\mathcal{V} \times \mathcal{V}$ et b est une forme linéaire continue sur \mathcal{V} . Dans la suite, la conjugaison complexe de $z \in \mathbb{C}$ est notée z^* . On définit l'isomorphisme de Riesz J de \mathcal{V}' dans \mathcal{V} tel que pour tout $l \in \mathcal{V}'$ et tout $u \in \mathcal{V}$, $(Jl, u)_\mathcal{V} = l(u)$, où $(\cdot, \cdot)_\mathcal{V}$ représente le produit scalaire dans \mathcal{V} avec la norme associée $\|\cdot\|_\mathcal{V}$. On note $\beta_\mu := \inf_{u \in \mathcal{V}} \sup_{v \in \mathcal{V}} \frac{|a_\mu(u, v)|}{\|u\|_\mathcal{V} \|v\|_\mathcal{V}} > 0$ la constant inf-sup de a_μ et $\tilde{\beta}_\mu$ une borne inférieure calculable de β_μ . Par simplicité, nous considérons que la forme linéaire b est indépendante du paramètre μ . L'extension à un second même b dépendant du μ est évidente. Nous désignons la solution discrète u_μ par "solution de confiance". Nous expliquons maintenant comment l'approximation BR est calculée. Supposons qu'une base réduite, constituée de \tilde{N} solutions u_{μ_i} de E_{μ_i} , $i \in \{1, \dots, \tilde{N}\}$, a déjà été construite. Pour alléger les

notations, nous notons u_i la fonction u_{μ_i} . Étant donnée une valeur du paramètre $\mu \in \mathcal{P}$, le problème réduit est une procédure de Galerkin sur l'espace linéaire $\hat{\mathcal{V}} = \text{Span}\{u_1, \dots, u_{\hat{N}}\} \subset \mathcal{V}$: trouver $\hat{u}_\mu \in \hat{\mathcal{V}}$ tel que

$$\hat{E}_\mu : a_\mu(\hat{u}_\mu, u_j) = b(u_j), \quad \forall j \in \{1, \dots, \hat{N}\}. \quad (3.29)$$

La solution approchée sur la base réduite est écrite

$$\hat{u}_\mu = \sum_{i=1}^{\hat{N}} \gamma_i(\mu) u_i. \quad (3.30)$$

En notant respectivement $Q(u_\mu)$ et $\hat{Q}(\hat{u}_\mu)$ les quantités d'intérêt exactes et approchées, la qualité de l'approximation pour un $\mu \in \mathcal{P}$ est quantifiée par la mesure d'erreur $\|Q(u_\mu) - \hat{Q}(\hat{u}_\mu)\|$. Lorsque nous obtenons une erreur satisfaisante avec $\hat{N} \ll N$, la stratégie de BR fonctionne. Deux cas sont généralement considérés : (i) la quantité d'intérêt est la solution entière : $Q = \hat{Q} = \text{Id}$ et $\|\cdot\| = \|\cdot\|_{\mathcal{V}}$, et (ii) Q est une forme linéaire sur \mathcal{V} et $\|\cdot\| = |\cdot|$. L'opérateur \hat{Q} est construit de façon consistante, telle que $\|Q(u_\mu) - \hat{Q}(\hat{u}_\mu)\|$ s'annule pour $\mu = \mu_i$, $i \in \{1, \dots, \hat{N}\}$. Dans la méthode des BR, la borne d'erreur a posteriori (que nous appelons simplement borne d'erreur dans la suite) est basée sur le calcul d'un résidu. Comme la borne d'erreur est une borne supérieure, elle fournit un outil pour certifier l'erreur faite par l'approximation des BR. La borne d'erreur est donnée par : pour tout $\mu \in \mathcal{P}$,

$$\|u_\mu - \hat{u}_\mu\|_{\mathcal{V}} \leq \mathcal{E}_1(\mu) := \tilde{\beta}_\mu^{-1} \|G_\mu \hat{u}_\mu\|_{\mathcal{V}}, \quad (3.31)$$

avec G_μ la forme linéaire de \mathcal{V} dans \mathcal{V} telle que $\mathcal{V} \ni u \mapsto G_\mu u := J(a_\mu(u, \cdot) - b) \in \mathcal{V}$.

La méthode des BR est dite efficace online si, dans l'étape online, (i) les problèmes réduits peuvent être construits en complexité indépendante de N , et (ii) la borne d'erreur peut être calculée en complexité indépendante de N . Nous disons que la forme sesquilinéaire a_μ dépend de μ de façon affine s'il existe d fonctions $\alpha_k(\mu) : \mathcal{P} \rightarrow \mathbb{C}$ et d formes sesquilineaires a_k bornées sur $\mathcal{V} \times \mathcal{V}$ et indépendantes de μ telles que

$$a_\mu(u, v) = \sum_{k=1}^d \alpha_k(\mu) a_k(u, v), \quad \forall u, v \in \mathcal{V}. \quad (3.32)$$

Dans cette section 3.4, nous supposons que la décomposition affine (3.32) est toujours vérifiée. Nous montrons que si a_μ dépend de μ de façon affine, alors la méthode des BR est efficace online.

En décomposant le produit scalaire dans (3.31), nous obtenons une autre formule pour la borne d'erreur :

$$\begin{aligned} \mathcal{E}_2(\mu) = \tilde{\beta}_\mu^{-1} & \left((G_{00}, G_{00})_{\mathcal{V}} + 2\text{Re} \sum_{i=1}^{\hat{N}} \sum_{k=1}^d \gamma_i(\mu) \alpha_k(\mu) (G_k u_i, G_{00})_{\mathcal{V}} \right. \\ & \left. + \sum_{i,j=1}^{\hat{N}} \sum_{k,l=1}^d \gamma_i(\mu) \alpha_k(\mu) \gamma_j^*(\mu) \alpha_l^*(\mu) (G_k u_i, G_l u_j)_{\mathcal{V}} \right)^{\frac{1}{2}}, \end{aligned} \quad (3.33)$$

qui peut être calculée en complexité indépendante de N dans l'étape online si $(G_{00}, G_{00})_{\mathcal{V}}$, $(G_k u_i, G_{00})_{\mathcal{V}}$ et $(G_k u_i, G_l u_j)_{\mathcal{V}}$, où $G_{00} := -Jb \in \mathcal{V}$ et $G_k u := J a_k(u, \cdot) \in \mathcal{V}$ pour tout $k \in \{1, \dots, d\}$, sont précalculés pendant l'étape offline, et si une borne inférieure $\tilde{\beta}_\mu$ de la constante de stabilité de a_μ peut aussi être calculée en complexité indépendante de N . En précision arithmétique infinie, $\mathcal{E}_1(\mu) = \mathcal{E}_2(\mu)$: les indices 1 et 2 sont utilisés pour noter deux façons différentes de calculer la même quantité. En pratique, $\mathcal{E}_2(\mu)$ est beaucoup plus sensible aux erreurs d'arrondis machine que $\mathcal{E}_1(\mu)$, comme nous allons le voir ci-après. Cependant, $\mathcal{E}_1(\mu)$ n'est pas efficace online, alors que $\mathcal{E}_2(\mu)$ l'est.

Considérons les deux formules \mathcal{E}_1 , voir (3.31), et \mathcal{E}_2 , voir (3.33), pour calculer la borne d'erreur. La formule \mathcal{E}_k , $k = 1, 2$, est dite valide pour calculer la borne d'erreur avec une tolérance tol si

$$\max_{\mu \in \mathcal{P}_{\text{select}}} (\mathcal{E}_k(\mu)) \leq \text{tol}. \quad (3.34)$$

D'un point de vue théorique, l'erreur $\|u_\mu - \hat{u}_\mu\|_{\mathcal{V}}$ et le résidu $G_\mu u_\mu$ s'annulent pour tout $\mu \in \mathcal{P}_{\text{select}}$. Ainsi, toute formule basée sur un résidu pour calculer la borne d'erreur devrait s'annuler également, pour toute tolérance. Cependant, la validité d'une formule est à considérer en présence de phénomènes adverses introduisant des

erreurs de calcul, voir la Figure 3.12. L'algorithme glouton de l'étape offline s'arrête lorsque $\max_{\mu \in \mathcal{P}_{\text{trial}}} (\mathcal{E}_k(\mu)) < \text{tol}_{\text{RB}}$, où tol_{RB} représente l'erreur maximale acceptable faite par l'approximation BR. Ainsi, si la tolérance minimale pour laquelle une formule \mathcal{E}_k pour calculer une borne d'erreur est valide est plus grande que tol_{RB} , alors l'algorithme glouton ne convergera pas et continuera à enrichir la base réduite, même si l'erreur effective est très petite.

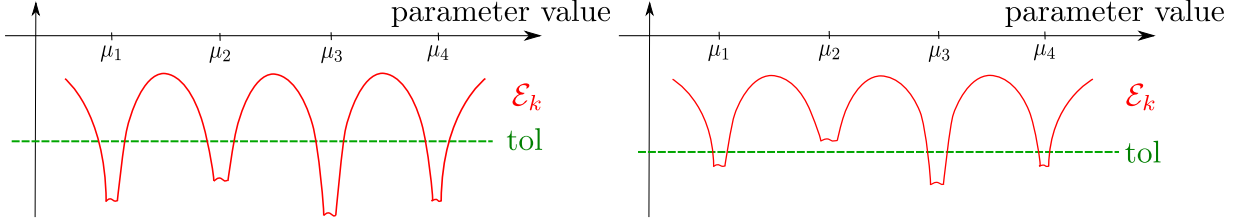


FIGURE 3.12 – Comportement de la formule \mathcal{E}_k en fonction du paramètre (illustration schématique de la définition de validité d'une formule), avec $\mathcal{P}_{\text{select}} = \{\mu_1, \dots, \mu_4\}$. Gauche : la formule \mathcal{E}_k est valide pour calculer la borne d'erreur avec tolérance tol ; droite : la formule n'est pas valide, car $\mathcal{E}_k(\mu_2) > \text{tol}$.

Nous examinons maintenant la validité des formules \mathcal{E}_1 et \mathcal{E}_2 pour calculer la borne d'erreur, en considérant comme phénomène adverse les erreurs d'arrondis machine de ces formules. Nous négligeons les erreurs d'arrondis machine introduits en résolvant E_μ et \hat{E}_μ , de sorte que les fonctions de la base réduite u_i et les solutions réduites \hat{u}_μ ne contiennent pas d'erreurs d'arrondis machine. Dans le manuscrit de thèse, nous considérons aussi comme phénomène adverse l'utilisation d'un solveur non exact pour résoudre E_{μ_i} . Nous supposons aussi que la borne inférieure calculable $\tilde{\beta}_\mu$ de la constante inf-sup ne contient pas d'erreurs d'arrondis machine.

Il est bien connu que l'évaluation d'un polynôme est plus précise en forme factorisée qu'en forme développée, tout particulièrement à des points proches de ses racines. D'après (3.33), $(\tilde{\beta}_\mu \mathcal{E}_2(\mu))^2$ est un polynôme multivarié d'ordre 2 calculé en forme développée, alors que le produit scalaire $(G_\mu u_\mu, G_\mu u_\mu)_\mathcal{V}$ utilisé dans le calcul de $\mathcal{E}_1(\mu)$ dans (3.31) n'est pas développé. Soit $\mu \in \mathcal{P}_{\text{select}}$ et notons $\text{maxfl}(\tilde{\beta}_\mu \mathcal{E}_k(\mu))$, $k = 1, 2$, l'évaluation de $\tilde{\beta}_\mu \mathcal{E}_k(\mu)$ lorsque l'accumulation d'erreurs d'arrondis machine maximale est atteinte. Nous montrons que

$$\begin{aligned} \text{maxfl}(\tilde{\beta}_\mu \mathcal{E}_1(\mu)) &\geq 2\delta\epsilon, \\ \text{maxfl}(\tilde{\beta}_\mu \mathcal{E}_2(\mu)) &\geq 2\delta\sqrt{\epsilon}, \end{aligned} \tag{3.35}$$

où $\delta = \|G_{00}\|_\mathcal{V}$ et ϵ est la précision machine (qui est l'erreur relative maximale de représentation d'un réel non nul en arithmétique à virgule flottante, $\epsilon \approx 5 \times 10^{-16}$ en double précision). Nous observons dans nos simulations numériques que les erreurs d'arrondis machine de \mathcal{E}_1 sont de l'ordre de ϵ , alors que ceux de \mathcal{E}_2 sont de l'ordre de $\sqrt{\epsilon}$. Ainsi, si nous supposons que les bornes inférieures dans (3.35) sont atteintes, les formules \mathcal{E}_1 et \mathcal{E}_2 sont valides pour calculer les bornes d'erreur avec tolérance tol si, respectivement,

$$\begin{aligned} \text{pour } \mathcal{E}_1, \quad & 2(\tilde{\beta}_{\min})^{-1} \delta\epsilon \leq \text{tol}, \\ \text{pour } \mathcal{E}_2, \quad & 2(\tilde{\beta}_{\min})^{-1} \delta\sqrt{\epsilon} \leq \text{tol}, \end{aligned} \tag{3.36}$$

où $\tilde{\beta}_{\min} = \inf_{\mu \in \mathcal{P}_{\text{select}}} (\tilde{\beta}_\mu)$.

La formule \mathcal{E}_1 pour calculer les bornes d'erreur est donc valide pour des tolérances de l'ordre de ϵ , mais n'est pas efficace online, tandis que la formule \mathcal{E}_2 est online efficace mais n'est valide que pour des tolérances significativement supérieures, à savoir des tolérances de l'ordre de $\sqrt{\epsilon}$. L'intuition numérique derrière la perte de précision de la formule \mathcal{E}_2 , est que lorsque nous faisons une combinaison linéaire dont le résultat est beaucoup plus petit que les sommes intermédiaires dans le calcul, il se produit une "catastrophic cancellation" (quand nous faisons une différence de deux réels proches en arithmétique en dimension finie, nous perdons autant de chiffres significatifs que le nombre de chiffres égaux entre les représentations de ces deux entiers). Nous allons construire une nouvelle formule efficace online qui soit en pratique valide pour des tolérances de l'ordre de ϵ .

Nous considérons qu'une base réduite de taille \hat{N} a été construite. Soit $\sigma := 1 + 2d\hat{N} + (d\hat{N})^2$. Soit $\mu \in \mathcal{P}_{\text{trial}}$ et $\hat{u}_\mu \in \text{Span}\{u_1, \dots, u_{\hat{N}}\}$ la solution réduite correspondante, nous définissons $X(\mu) \in \mathbb{C}^\sigma$ comme le vecteur de composantes $(1, x_{\mu_I}, x_{\mu_I}^*, x_{\mu_I}^* x_{\mu_J})$, où $x_{\mu_I} = \alpha_k(\mu) \gamma_i(\mu)$ (nous rappelons que $\gamma_i(\mu)$ sont les coefficients de la solution réduite dans la base réduite, voir (3.30), et $\alpha_k(\mu)$, les coefficients de la décomposition affine de a_μ

Propriété	\mathcal{E}_1	\mathcal{E}_2	\mathcal{E}_3
Efficacité online	No	Yes	Yes
Dépendance en ϵ de la précision observée	ϵ	$\sqrt{\epsilon}$	ϵ
Complexité de l'étape offline	–	$(d\hat{N} + 1)N_{\text{sol}}$	σN_{sol}
Complexité de l'étape online	–	$\hat{N}^3 + \sigma$	$\hat{N}^3 + \sigma^3$

TABLE 3.2 – Comparaison des formules considérées pour calculer la borne d'erreur, où N_{sol} est la complexité de la résolution de E_μ .

dans (3.32)), avec $1 \leq I, J \leq d\hat{N}$ (avec $I = i + \hat{N}(k - 1)$ tel que $1 \leq i \leq \hat{N}$, $1 \leq k \leq d$, et avec $J = j + \hat{N}(l - 1)$ tel que $1 \leq j \leq \hat{N}$, $1 \leq l \leq d$). Une observation importante est que la quantité entre parenthèses dans le membre de droite de (3.33) peut être écrite comme une forme linéaire en $X(\mu)$ de la façon suivante :

$$\delta^2 + 2\text{Re}(s^t x_\mu) + x_\mu^{*t} S x_\mu = \sum_{p=1}^{\sigma} t_p X_p(\mu), \quad (3.37)$$

où t_p est indépendant de μ (car δ , s , et S sont indépendants de μ) et $X_p(\mu)$ est la p -ième composante de $X(\mu)$. Considérons la fonction de deux variables $(p, \mu) \mapsto X_p(\mu)$, pour tout $p \in \{1, \dots, \sigma\}$ et tout $\mu \in \mathcal{P}_{\text{trial}}$. Nous cherchons une approximation de cette fonction de la forme

$$\forall \mu \in \mathcal{P}_{\text{trial}}, \forall p \in \{1, \dots, \sigma\}, X_p(\mu) \approx \sum_{r=1}^{\hat{\sigma}} \lambda_r(\mu) X_p(\mu_r), \quad (3.38)$$

pour un certain paramètre $\hat{\sigma} \leq \sigma$. L' EIM permet de construire cette approximation et de choisir les points d'interpolation.

Lorsque $\hat{\sigma}$ est grand, il peut être utile de stabiliser l'EIM vis-à-vis des erreurs d'arrondis machine. Nous proposons une stabilisation inspirée de la procédure d'orthonormalisation de Gram-Schmidt stabilisée. Nous pouvons maintenant injecter la décomposition (6) dans l'expression (3.67) de $(\beta\mathcal{E}(\mu))^2$, et réorganiser les sommes :

$$\begin{aligned} (\beta\mathcal{E}(\mu))^2 &\approx \sum_{p=1}^{\sigma} t_p \left\{ \sum_{l=1}^{\hat{\sigma}} \sum_{r=1}^{\hat{\sigma}} \Delta_{l,r} X_{pl}(\mu) X_p(\mu_r) \right\} \\ &= \sum_{r=1}^{\hat{\sigma}} \underbrace{\left\{ \sum_{l=1}^{\hat{\sigma}} \Delta_{l,r} X_{pl}(\mu) \right\}}_{\lambda_r(\mu)} \underbrace{\left\{ \sum_{p=1}^{\sigma} t_p X_p(\mu_r) \right\}}_{(\beta\mathcal{E}(\mu_r))^2} \end{aligned} \quad (3.39)$$

conduisant à la nouvelle formule pour calculer la borne d'erreur :

$$\mathcal{E}_3(\mu) := \tilde{\beta}_\mu^{-1} \left(\sum_{r=1}^{\hat{\sigma}} \lambda_r(\mu) (\beta\mathcal{E}_1(\mu_r))^2 \right)^{\frac{1}{2}}, \quad (3.40)$$

où nous pouvons choisir la formule \mathcal{E}_1 dans le membre de droite, car ces évaluations de la borne d'erreur se faisant à des valeurs fixes μ_i , elles peuvent être précalculées pendant l'étape offline, tout en garantissant que la formule \mathcal{E}_3 est efficace online. Dans la combinaison linéaire dans (3.40), les $\mathcal{E}(\mu_i)$ sont du même ordre de grandeur que le résultat de la combinaison linéaire, ce qui empêche les "catastrophic cancellations" qui interviennent avec la formule \mathcal{E}_2 et nuisent à la précision de cette formule. Les avantages et inconvénients des trois formules considérées pour calculer la borne d'erreur sont résumés dans la Table 3.2.

Nous considérons comme illustration numérique un problème de diffusion en une dimension, à savoir le problème aux limites $-u'' + \mu u = 1$ sur $]0, 1[$ avec $u(0) = u(1) = 0$, et un paramètre $\mu \in \mathcal{P} := [1, 100]$. La solution analytique est :

$$u(x) = -\frac{1}{\mu} (\cosh(\sqrt{\mu}x) - 1) + \frac{\cosh(\sqrt{\mu}) - 1}{\mu \sinh(\sqrt{\mu})} \sinh(\sqrt{\mu}x). \quad (3.41)$$

La théorie de Lax-Milgram est applicable, et la constante de coercivité est bornée par en-dessous par 1 dans la norme $H^1(]0, 1[)$. La borne d'erreur est donnée par $\mathcal{E}_1(\mu) = \|G_\mu \hat{u}_\mu\|_{H^1(]0, 1[)}$. Les éléments finis de Lagrange \mathbb{P}_1 sont utilisés avec des mailles uniformes de longueur 0.005. L'ensemble $\mathcal{P}_{\text{trial}}$ consiste en 1000 points uniformément distribués dans \mathcal{P} . La méthode des BR est conduite jusqu'à ce que la formule \mathcal{E}_2 souffre d'erreurs d'arrondis machine, ce qui se produit déjà pour une base réduite de taille $\hat{N} = 7$ (puisque $d = 2$, on a $\sigma = 225$). Un solveur direct est utilisé, si bien que le seul phénomène adverse pour calculer la borne d'erreur sont les erreurs d'arrondis machine. Dans la Figure 3.13, nous remarquons que la formule classique \mathcal{E}_2 n'est pas valide pour calculer une borne d'erreur avec n'importe quelle tolérance inférieure à 10^{-7} , alors que les formules \mathcal{E}_1 et \mathcal{E}_3 sont valides pour des tolérances aussi faibles que 10^{-14} . La différence est de 7 ordres de grandeur ; sachant que $\sqrt{\epsilon} \approx 10^{-7}$, ces observations sont cohérentes avec les relations (3.36).

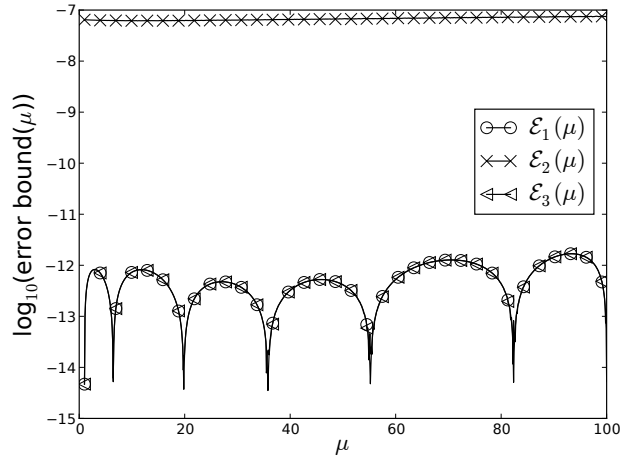


FIGURE 3.13 – Courbes de borne d'erreur en fonction du paramètre pour les trois formules considérées.

La formule \mathcal{E}_3 a été utilisée avec succès dans des problèmes aéroacoustiques plus compliqués dans le manuscrit de thèse, pour lesquels les faibles valeurs de la constante de stabilité requièrent une grande précision de la borne d'erreur, et par M. Yano au MIT [119], où la nécessité d'avoir une évaluation très précise de la borne d'erreur est imposée par la spécificité du problème non-linéaire (équation de Boussinesq). Ce travail initié sur la stabilité numérique vis à vis des erreurs d'arrondis machine des estimateurs a posteriori pour la méthode des bases réduites a été poursuivi dans d'autres équipes [28, 36].

3.5 Mise en œuvre des méthodes de réduction de modèle : non-intrusivité, problèmes non-linéaires de grande taille, indication d'erreur, variabilités non-paramétrées

3.5.1 Nonintrusive Empirical Interpolation Method : faible intrusivité pour la méthode des bases réduites

Cette section correspond à la Publications Pub.15, au Rapport Rap.5 et aux Exposés Exp.10 à Exp.13 et Exp.21.

Le but de cette section est d'obtenir une approximation non-intrusive, en utilisant une procédure offline-online, des objets suivants :

$$Q_t(\mu) = \sum_{s=1}^{\varsigma} \int_{\Omega} g_s(\mu, x) \Psi_{s,t}(x) dx, \quad \forall t \in \{1 \dots \mathcal{N}\}, \quad (3.42)$$

où $\varsigma \geq 2$, et \mathcal{N} est supposé grand. Les fonctions $\Psi_{s,t}$ sont des produits de fonctions de base qui interviennent dans les coefficients du vecteur C_μ et de la matrice A_μ , qui sont les approximations en dimension finie de respectivement l'opérateur a_μ et la forme linéaire b_μ (lorsque celle-ci dépend de μ) dans (3.28). Pour illustrer

les enjeux, prenons l'exemple de $a_\mu(u, v) = \int_\Omega g(\mu, x) \nabla u(x) \cdot \nabla v(x) dx$, conduisant à la matrice $(A_\mu)_{i,j} = \int_\Omega g(\mu, x) \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) dx$, $1 \leq i, j \leq N$. Ceci correspond à $\varsigma = 1$, $\mathcal{N} = N$, $t = (i, j)$ et $\Psi_{1,t}(x) = \varphi_i(x) \cdot \varphi_j(x)$ dans (3.42).

Le résultat principal de cette section est l'approximation de (3.42) de la forme

$$Q_t(\mu) \approx \sum_{r=1}^{d^z} \beta_r(\mu) Q_t(\mu_r), \quad (3.43)$$

pour un entier d^z , des coefficients $\{\beta_r(\mu)\}_{1 \leq r \leq d^z}$ et des valeurs de paramètre $\{\mu_r\}_{1 \leq r \leq d^z}$.

Pour illustrer notre idée, nous considérons dans un premier temps le cas où la fonction g_s dépend seulement de μ , mais pas de x . Ceci correspond au cas où l'opérateur a_μ a une dépendance affine en μ :

$$Q_t(\mu) = \sum_{s=1}^{\varsigma} g_s(\mu) \int_\Omega \Psi_{s,t}(x) dx, \quad \forall t \in \{1 \dots N\}. \quad (3.44)$$

Nous appliquons un EIM à la fonction $g_s(\mu)$ vue comme la fonction de deux variables

$$\gamma : (\mu, s) \mapsto g_s(\mu). \quad (3.45)$$

Nous utilisons les mêmes notations que dans l'introduction pour la matrice B , les fonctions à valeur vectorielle $q_m(\cdot)$, et les points sélectionnés μ_m , et nous introduisons les indices s_l sélectionnés par EIM pour approcher $\gamma(\mu, s)$. Il vient

$$g_s(\mu) \approx \sum_{r=1}^d \underbrace{\left\{ \sum_{l=1}^d \Delta_{l,r} g_{s_l}(\mu) \right\}}_{:=\beta_r(\mu)} g_s(\mu_r), \quad (3.46)$$

où d , qui représente maintenant le nombre de points d'interpolation EIM, est inférieur à ς . En injectant cette approximation dans (3.44) et en changeant l'ordre des sommations,

$$\begin{aligned} Q_t(\mu) &\approx \sum_{s=1}^{\varsigma} \sum_{r=1}^d \beta_r(\mu) g_s(\mu_r) \int_\Omega \Psi_{s,t}(x) dx \\ &= \sum_{r=1}^d \beta_r(\mu) Q_t(\mu_r), \end{aligned} \quad (3.47)$$

qui correspond à un approximation de la forme (3.43).

Quand la dépendance affine n'est pas disponible, la première étape consiste à approcher la fonction $g_s(\mu, x)$ pour tout $1 \leq s \leq \varsigma$ avec un premier EIM. Cela conduit à la construction de ς ensembles de points x , de points μ , de matrices B , Γ , Δ et de fonctions à valeurs vectorielles $q(\cdot)$. On note ces quantités avec un indice additionnel s ; par exemple, $q_s(\cdot)$, de composantes $q_{s,m} : x \mapsto q_{s,m}(x)$, pour tout $1 \leq m \leq d$. Par simplicité, nous supposons que les ς EIM ont le même nombre de points d'interpolation d . En injectant les approximations EIM de $g_s(\mu, x)$, pour tout $1 \leq s \leq \varsigma$, dans (3.42) donne l'approximation suivante de $Q_t(\mu)$, que nous notons $(\mathcal{I}_d Q_t)(\mu)$ et qui est donnée par

$$\begin{aligned} (\mathcal{I}_d Q_t)(\mu) &:= \sum_{s=1}^{\varsigma} \int_\Omega (I_d g_s)(\mu, x) \Psi_{s,t}(x) dx \\ &= \sum_{s=1}^{\varsigma} \sum_{m=1}^d \sum_{l=1}^d (\Delta_s)_{l,m} g_s(\mu, x_{s,l}) \int_\Omega g_s(\mu_{s,m}, x) \Psi_{s,t}(x) dx. \end{aligned} \quad (3.48)$$

Nous observons que l'expression (3.48) est une forme linéaire par rapport à un vecteur $z \in \mathbb{R}^{\varsigma d}$, dont les

composantes, notées $z_p(\mu)$, $1 \leq p \leq \varsigma d$, contiennent toutes les dépendances en μ . Notons

$$z_p(\mu) := \begin{cases} \sum_{l=1}^d (\Delta_1)_{m,l} g_1(\mu, x_{1,l}), & 1 \leq m \leq d, \quad p = m, \\ \sum_{l=1}^d (\Delta_2)_{m,l} g_2(\mu, x_{2,l}), & 1 \leq m \leq d, \quad p = m + d, \\ \vdots \\ \sum_{l=1}^d (\Delta_\varsigma)_{m,l} g_\varsigma(\mu, x_{\varsigma,l}), & 1 \leq m \leq d, \quad p = m + (\varsigma - 1)d, \end{cases} \quad (3.49)$$

et

$$Q_{t,p} := \begin{cases} \int_{\Omega} g_1(\mu_{1,m}, x) \Psi_{1,t}(x) dx, & 1 \leq m \leq d, \quad p = m, \\ \int_{\Omega} g_2(\mu_{1,m}, x) \Psi_{2,t}(x) dx, & 1 \leq m \leq d, \quad p = m + d, \\ \vdots \\ \int_{\Omega} g_\varsigma(\mu_{1,m}, x) \Psi_{\varsigma,t}(x) dx, & 1 \leq m \leq d, \quad p = m + (\varsigma - 1)d. \end{cases} \quad (3.50)$$

Ainsi,

$$(\mathcal{I}_d Q_t)(\mu) = \sum_{p=1}^{\varsigma d} z_p(\mu) Q_{t,p}. \quad (3.51)$$

Pour obtenir une approximation non-intrusive de $Q_t(\mu)$ de la forme (3.43), nous appliquons un autre EIM à $z_p(\mu)$ vue comme la fonction de deux variables :

$$\zeta : (\mu, p) \mapsto z_p(\mu). \quad (3.52)$$

Pour bien identifier les quantités associées à l'EIM de $\zeta(\mu, p)$, nous leur ajoutons un exposant (ζ). En injectant cette approximation EIM dans le membre de droite de (3.51), nous obtenons :

$$\begin{aligned} (\mathcal{I}_d Q_t)(\mu) &\approx \sum_{p=1}^{\varsigma d} (I_d^{(\zeta)} z)(\mu, p) Q_{t,p} \\ &= \sum_{p=1}^{\varsigma d} \sum_{l=1}^{d^\zeta} \sum_{r=1}^{d^\zeta} \Delta_{l,r}^{(\zeta)} z_{p_l^{(\zeta)}}(\mu) z_p(\mu_r^{(\zeta)}) Q_{t,p}. \end{aligned} \quad (3.53)$$

En échangeant les ordres de sommation dans (3.53), il vient :

$$\begin{aligned} (\mathcal{I}_d Q_t)(\mu) &\approx \sum_{r=1}^{d^\zeta} \sum_{l=1}^{d^\zeta} \Delta_{l,r}^{(\zeta)} z_{p_l^{(\zeta)}}(\mu) \sum_{p=1}^{\varsigma d} z_p(\mu_r^{(\zeta)}) Q_{t,p} \\ &= \sum_{r=1}^{d^\zeta} \sum_{l=1}^{d^\zeta} \Delta_{l,r}^{(\zeta)} z_{p_l^{(\zeta)}}(\mu) (\mathcal{I}_d Q_t)(\mu_r^{(\zeta)}), \end{aligned} \quad (3.54)$$

où (3.51) a été utilisé dans le deuxième ligne. En remplaçant enfin $\mathcal{I}_d Q_t$ par Q_t , nous obtenons l'approximation non-intrusive de $Q_t(\mu)$:

$$Q_t(\mu) \approx \sum_{r=1}^{d^\zeta} \underbrace{\left\{ \sum_{l=1}^{d^\zeta} \Delta_{l,r}^{(\zeta)} z_{p_l^{(\zeta)}}(\mu) \right\}}_{:= \beta_r(\mu)} Q_t(\mu_r^{(\zeta)}). \quad (3.55)$$

Pour construire le problème réduit (3.29), nous devons assembler la matrice $\hat{A}_{\mu,ij} = a_\mu(\hat{u}_i, \hat{u}_j) = \hat{U}_i^t Q_t(\mu) \hat{U}_j$, où \hat{U}_i est la i -ième fonction de la base réduite, sur la base des éléments finis associée au problème initial E_μ ,

et $Q_t(\mu) = A_{\mu,ij}$. Nous obtenons $\hat{A}_{\mu,ij} \approx \sum_{r=1}^{d^z} \beta_r(\mu) \hat{U}_i^t A_{\mu_r^{(\zeta)},ij} \hat{U}_j$: le problème réduit peut être construit sans avoir à modifier les routines d'assemblage élémentaire du code considéré. Ceci fournit un avantage certain dans l'implémentation des BR dans un contexte industriel. De plus, tout produit matrice-vecteur optimisé est directement utilisable.

Dans les implémentations d'éléments de frontière pour l'équation d'Helmholtz, la FMM permet de réaliser les produits matrice-vecteur de façon rapide (mais approchée), et ainsi de résoudre des systèmes linéaires pleins en utilisant des méthodes itératives, en complexité $n \log n$, où n est le nombre d'inconnues. Dans cette section, nous considérons un cas test où les matrices $A_{\mu_r^{(\zeta)}}$, où $\mu_r^{(\zeta)}$ sont les valeurs de paramètres sélectionnées en appliquant la formule (3.55) à l'approximation de A_μ , sont tellement grandes qu'elles ne peuvent pas être stockées sur disque dur. Ainsi, à chaque fois qu'un produit matrice-vecteur est requis, la matrice est assemblée et la FMM est utilisée. Nous considérons un problème de diffraction d'une onde acoustique incidente par un objet dont la surface est recouverte, sur trois zones, par des matériaux impédants. Cette formulation et son analyse sont détaillées dans le manuscrit de thèse, mais nous ne les reproduisons pas ici. L'objet considéré est un avion complet, voir la Figure 3.14. La source est un monopole acoustique, situé sous l'aile droite de l'avion

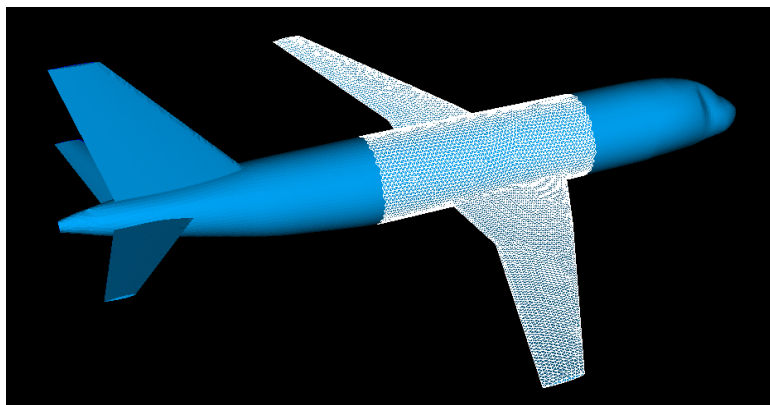


FIGURE 3.14 – Maillage d'avion utilisé, la deuxième surface impédante est mise en évidence.

(pour simuler la présence d'un moteur). Les paramètres considérés sont la fréquence de la source et les valeurs d'impédance des trois zones composant la surface de l'avion. La fréquence varie de 27 à 135 Hz et chaque coefficient d'impédance de 1 à 2. Nous considérons 532,400 valeurs paramétrique dans $\mathcal{P}_{\text{trial}}$ (400 valeurs de la fréquence et 11 valeurs pour chaque coefficient d'impédance).

Pour recouvrer l'hypothèse de dépendance affine nécessaire à la méthode des BR, nous utilisons la formule d'approximation non-intrusive (3.55) (telle que $d = 35$ et $d^z = 50$). Dans ce cas-test, une formule non-intrusive similaire doit être utilisée pour l'approximation du second membre (telle que $d = 50$ et $d^z = 60$). Une base réduite de taille $\hat{n} = 30$ est construite. Les deux étapes qui prennent le plus de temps de calcul sont les produits matrice-vecteur en FMM et l'exploration de $\mathcal{P}_{\text{trial}}$ par l'algorithme glouton. Ces deux étapes sont parallélisées dans notre code. Ainsi, nous nous attendons à ce que cette procédure soit efficace sur les clusters avec de nombreux processeurs. Les simulations ont été réalisées sur un ordinateur portable standard. La formule (3.55) permet d'utiliser directement la FMM, sans laquelle les simulations auraient été impossibles sur un ordinateur standard, dans la mesure où la stockage d'une seule matrice requiert 60 GB de mémoire. Une approche consistant à calculer et stocker les 50 matrices de la décomposition aurait nécessité 3 TB de mémoire.

L'étape online prend 1.5×10^{-2} secondes pour calculer une solution réduite et la borne d'erreur correspondante, tandis que le problème complet, même accéléré par la FMM parallèle, prend environ 40 minutes, ce qui correspond à un facteur d'accélération de 1.6×10^5 . L'étape offline prend environ 2 jours, et la dernière étape de l'algorithme glouton dans l'étape offline prend 1 heure. La FMM que nous utilisons pour calculer les produits matrice-vecteur a une précision relative d'environ 10^{-3} ; nous ne pouvons donc pas espérer atteindre une approximation BR plus précise.

Nous considérons comme quantité d'intérêt le champ acoustique diffracté sur un réseau de 1681 points localisés derrière l'avion. Un applet java a été développé pour calculer et afficher ce champ de pression en temps réel à des valeurs de paramètre (fréquence de la sources et coefficients d'impédance) choisies par l'utilisateur. Deux autres applets java pour d'autres simulations industrielles sont présentées dans le manuscrit de thèse. Considérons les valeurs de paramètres suivantes : fréquence = 122.3 Hz, $\mu_1 = 1.21$, $\mu_2 = 1.87$ et $\mu_3 = 1.45$. La borne d'erreur est 5.4×10^{-4} , et l'erreur relative entre la solution directe et la solution réduite, en norme

euclidienne, est 1%. Sur le réseau de 1681 points localisés derrière l’avion, l’erreur relative sur le champ acoustique diffracté est 1.4%. La Figure 3.15 montre le champ de pression acoustique calculé à partir de la solution de confiance, et la différence entre le champ de pression approché par les BR et le champ calculé à partir de la solution de confiance. Notons que la méthode des bases réduites a été appliquée à l’Electric Field Integral Equation dans [49, 67].

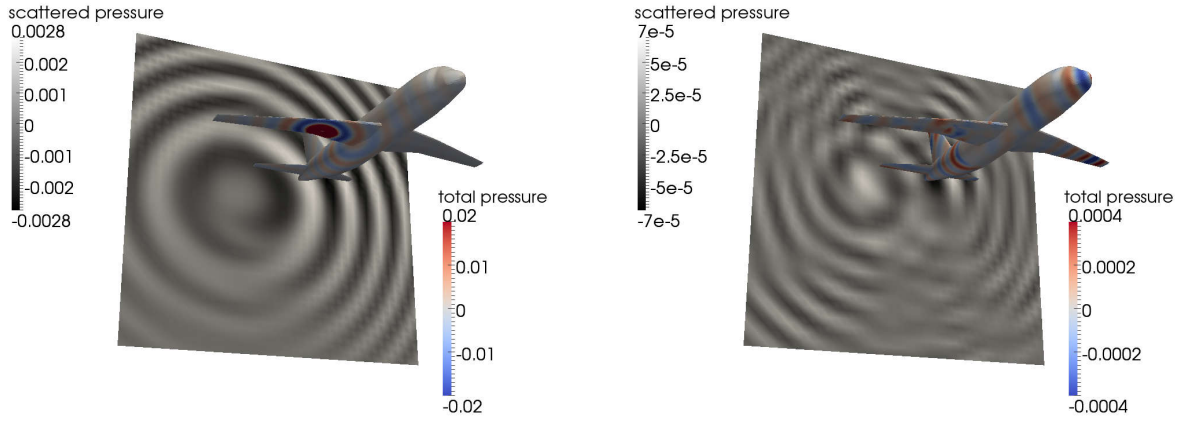


FIGURE 3.15 – Gauche : champ de pression acoustique sur l’avion et sur le réseau de 1681 points localisés derrière l’avion calculé à partir de la solution de confiance, droite : différence entre les champs approchés par BR et de confiance.

3.5.2 Approximation non-intrusive des limites d’algorithmes de puissances de matrices : application aux inverses et log-déterminants de matrices et à la réduction non-intrusive de modèles non-linéaires

Cette section correspond à la Publication [Pub.10](#) et au Rapport [Rap.1](#) et aux Exposés [Exp.6](#) et [Exp.18](#).

Dans cette section, nous présentons une méthode non-intrusive qui n’est pas de type Galerkin, et qui se rapproche des méthodes de régression statistique. La différence ici, c’est que nous tâchons d’injecter une information de l’équation sous-jacente directement dans la méthode de régression. Nous considérons une famille paramétrée de matrices inversibles, telle que la dépendance paramétrique est affine, et nous sommes intéressés par l’approximation non-intrusive de quantités qui sont obtenues comme limites d’algorithmes de puissance de ces matrices paramétrées, par exemple la matrice inverse et le logarithme du déterminant de la matrice (noté log-det), que nous exprimons comme combinaisons linéaires de ces quantités prises à des valeurs bien choisies de paramètres. Par exemple, de nombreuses évaluations du log-det d’une matrice définie positive sont requises pour l’estimation du maximum de vraisemblance en régression par processus gaussiens, voir [85, 123]. L’algorithme proposé calcule efficacement les coefficients de cette combinaison linéaire (i.e. en complexité algorithmique indépendante de la taille des matrices), et de façon non-intrusive, dans le sens où il ne repose que sur l’évaluation de ces quantités d’intérêt (dans notre cas la matrice inverse et le log-det), comme la plupart des méthodes de régression statistique. Sous l’hypothèse de dépendance affine de la matrice en les paramètres, nous utilisons l’EIM pour obtenir une approximation non-intrusive des puissances de cette matrice, à partir de laquelle nous construisons une approximation des limites mentionnées ci-dessus. Nous proposons des bornes supérieures pour cette approximation. Des applications numériques proposant des comparaisons avec quelques méthodes classiques de régression statistique sont présentées. Nous présentons la procédure seulement pour la matrice inverse et renvoyons vers la Publication [Pub.10](#) pour les détails sur le log-det.

Description de la méthode

Fixons $d \in \mathbb{N}$ et $\mu \in \mathcal{P}$ un paramètre, où \mathcal{P} est un espace paramétrique, pris comme un sous-ensemble compact de \mathbb{R}^r , $r \in \mathbb{N}^*$. Considérons $\{A_\mu\}_{\mu \in \mathcal{P}} \subset \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ un ensemble de matrices carrées paramétrées et

supposons une dépendance affine suivante :

$$A_\mu = \sum_{l=1}^d \alpha_l(\mu) A_l, \quad \mu \in \mathcal{P}, \quad (3.56)$$

où la famille de matrices carrées $\{A_l\}_{1 \leq l \leq d}$ est indépendante de μ . Ainsi, la matrice A_μ dépend de μ seulement à travers les coefficients $\alpha_l : \mathcal{P} \rightarrow \mathbb{R}$. En posant $\alpha_0(\mu) = 1$ et $A_0 = -\Psi_0$, nous obtenons une décomposition affine semblable à (3.56) pour la matrice $I - \Psi_0^{-1} A_\mu$. Soit $t \in \mathbb{N}$, nous notons un multi-indice $\vec{s} = (s_1, s_2, \dots, s_t) \in \mathbb{N}^t$ et définissons son poids par $|\vec{s}| := \sum_{l=1}^t s_l$. Notons également $\bar{\kappa}_{m,d} = \left\{ \vec{k} \in \llbracket 0; m \rrbracket^d \text{ tel que } |\vec{k}| \leq m \right\}$. Définissons enfin $g(\vec{k}, \mu) = \prod_{l=0}^d \alpha_l^{k_l}(\mu)$, $k \in \bar{\kappa}_{m,d}$, $\mu \in \mathcal{P}$.

Soit $m \in \mathbb{N}$, nous montrons qu'il existe une famille de matrices $\{T_{\vec{k},p}^-\}_{\vec{k} \in \bar{\kappa}_{m,d}, 0 \leq p \leq m} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, indépendantes de μ , telles que

$$(I - \Psi_0^{-1} A_\mu)^p = \sum_{\vec{k} \in \bar{\kappa}_{m,d}} g(\vec{k}, \mu) T_{\vec{k},p}^-, \quad \mu \in \mathcal{P}, \quad 1 \leq p \leq m. \quad (3.57)$$

Nous donnons une expression explicite pour les matrices $\{T_{\vec{k},p}^-\}_{\vec{k} \in \bar{\kappa}_{m,d}, 0 \leq p \leq m}$ dans la Publication [Pub.10](#) (la connaissance de cette expression n'est pas nécessaire pour notre présentation résumée). Comme $\#\{\vec{k} \in \llbracket 0; p \rrbracket^d \text{ tel que } |\vec{k}| = p\} = \frac{(p+d-1)!}{(d-1)!p!}$, le nombre de termes dans (3.57), noté $Q_{m,d} := \#\bar{\kappa}_{m,d}$, est $\frac{1}{(d-1)!} \sum_{k=0}^m \frac{(k+d-1)!}{k!}$. Notons que pour $d \geq 2$, $Q_{m,d} \leq \frac{m}{(d-1)!} (m+1)(m+2) \cdots (m+d-1) = P_d(m)$, où $P_d(m)$ est un polynôme de degré d en m .

Nous appliquons l'EIM, présenté dans l'introduction, à la fonction $g(\vec{k}, \mu)$, et obtenons :

$$(I - \Psi_0^{-1} A_\mu)^p \approx \sum_{l=1}^{N^{\text{EIM}}} \lambda_l(\mu) (I - \Psi_0^{-1} A_{\mu_l})^p, \quad \mu \in \mathcal{P}, \quad 1 \leq p \leq m, \quad (3.58)$$

où μ_l et la procédure de calcul des $\lambda_l^{N^{\text{EIM}}}$ sont construits lors de la phase offline de l'algorithme EIM.

Considérons le schéma itératif suivant :

$$\begin{cases} X_{0,\mu} = X_0 \\ X_{k+1,\mu} = (I - \Psi_0^{-1} A_\mu) X_{k,\mu} + \Psi_0^{-1}, \end{cases} \quad (3.59)$$

avec les conditions initiales suivantes, indépendantes du paramètre μ :

$$\begin{cases} \Psi_0 = \underset{M \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}, \text{ Minversible}}{\operatorname{argmin}} \sup_{\mu \in \mathcal{P}} \|I - M^{-1} A_\mu\|_2, \\ X_0 = \underset{X \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}}{\operatorname{argmin}} \sup_{\mu \in \mathcal{P}} \|X - A_\mu^{-1}\|_2, \end{cases} \quad (3.60)$$

et définissons

$$\rho = \sup_{\mu \in \mathcal{P}} \|I - \Psi_0^{-1} A_\mu\|_2, \quad (3.61)$$

et $\epsilon_0 = \sup_{\mu \in \mathcal{P}} \|X_0 - A_\mu^{-1}\|_2$. De $X_{m,\mu} = (I - \Psi_0^{-1} A_\mu)^m (X_0 - A_\mu^{-1}) + A_\mu^{-1}$, nous obtenons la borne indépendante de μ suivante

$$\|X_{m,\mu} - A_\mu^{-1}\|_2 \leq \epsilon_0 \rho^m, \quad (3.62)$$

qui, comme $\rho < 1$, assure la convergence uniforme du schéma itératif (3.59) en m .

Nous montrons par récurrence que

$$X_{m,\mu} = (I - \Psi_0^{-1} A_\mu)^m X_0 + \left(\sum_{k=0}^{m-1} (I - \Psi_0^{-1} A_\mu)^k \right) \Psi_0^{-1}, \quad \forall m \in \mathbb{N}, \quad (3.63)$$

où la limite A_μ^{-1} n'apparaît plus. En remplaçant les puissances de $(I - \Psi_0^{-1} A_\mu)$ dans (3.63) et en utilisant (3.58), nous obtenons :

$$X_{m,\mu} \approx \sum_{l=1}^{N^{\text{EIM}}} \lambda_l(\mu) X_{m,\mu_l}. \quad (3.64)$$

La convergence de $X_{m,\mu}$ vers A_μ^{-1} en fonction de m , voir (3.62), suggère de remplacer X_{m,μ_l} par les inverses $A_{\mu_l}^{-1}$ dans (3.64) et de définir

$$\mathcal{X}_\mu^{N^{\text{EIM}}} := \sum_{l=1}^{N^{\text{EIM}}} \lambda_l(\mu) A_{\mu_l}^{-1}. \quad (3.65)$$

où $\mathcal{X}_\mu^{N^{\text{EIM}}}$ est bien une approximation non-intrusive de A_μ^{-1} . En effet, dans l'Équation (3.65), nous rappelons que les coefficients sont obtenus à partir d'un EIM sur $g(\vec{k}, \mu)$, qui dépend uniquement de la dépendance paramétrique de A_{μ_l} , voir l'Équation (3.56). Ainsi, l'approximation obtenue est non-intrusive dans le sens où elle ne repose que sur des évaluations de la quantité d'intérêt (les inverses $A_{\mu_l}^{-1}$), tout en utilisant une information sur la forme particulière du problème (à travers les $\alpha_l(\mu)$). En particulier, nous n'avons jamais besoin de calculer les $(I - \Psi_0^{-1} A_\mu)^p$ ni les $\hat{T}_{\vec{k},p}$. Remarquons que les matrices A_l dans (3.56), $1 \leq l \leq d$, n'ont pas besoin non plus d'être calculées. De plus, nous pouvons calculer les $A_{\mu_l}^{-1}$ par la méthode de notre choix, et même si le schéma itératif considéré converge, nous pouvons utiliser les méthodes directes pour calculer les $A_{\mu_l}^{-1}$, et appliquer (3.65) pour obtenir notre approximation de A_μ^{-1} . En particulier, nous n'avons jamais besoin de construire Ψ_0 et X_0 , nous avons simplement besoin de l'existence d'une matrice $\hat{\Psi}$ telle que $\sup_{\mu \in \mathcal{P}} \|I - \hat{\Psi} A_\mu\|_2 < 1$. Remarquons enfin que nous obtenons directement une façon d'approcher les solutions de systèmes linéaires à second membre constant :

$$A_\mu^{-1} b \approx \sum_{l=1}^{N^{\text{EIM}}} \lambda_l(\mu) (A_{\mu_l}^{-1} b). \quad (3.66)$$

Un élément clef de la méthode est la linéarité de la fonction $g \mapsto X_{m,\mu}$, où, d'après (3.57) et (3.63),

$$X_{m,\mu} = \sum_{\vec{k} \in \bar{\kappa}_{m,d+1}} g(\vec{k}, \mu) \left(T_{\vec{k},m} X_0 + \sum_{l=0}^{m-1} T_{\vec{k},l} \Psi_0^{-1} \right). \quad (3.67)$$

Performance de l'approximation

Dans un contexte industriel avec des calculs de grand échelle et un budget contraint, le N^{EIM} dans (3.65) et (3.66) ne peut pas être aussi grand que nous souhaitons. Si l'approximation produit des erreurs trop grandes, le problème sera considéré non réductible par la méthode choisie et avec le budget alloué. Dans (3.60), Ψ_0^{-1} peut être vu comme le meilleur préconditionneur uniformément sur l'espace paramétrique. Le problème peut être réduit efficacement si ce préconditionneur est bon au sens $\sup_{\mu \in \mathcal{P}} \|I - \Psi_0^{-1} A_\mu\|_2 = \rho \ll 1$, voir Equation (3.62).

Dans les cas d'espace paramétrique de grand dimension, l'existence d'un bon préconditionneur est peu probable à cause de la malédiction de la dimensionnalité, en particulier si l'intervalle de variation des paramètres est grand. Nous rappelons que nous n'avons pas besoin de calculer Ψ_0^{-1} et avons seulement besoin de son existence. Le succès de la méthode sera évalué a posteriori, si une structure de rang faible existe, comme pour toute méthode de réduction de modèle (par exemple, avec la snapshot POD, si les valeurs propres de la matrice de corrélation décroît suffisamment rapidement). Dans ce contexte, à budget de calcul donné, nous comparons notre algorithme à d'autres procédures non-intrusives en calculant l'erreur par rapport à la solution haute-fidélité.

Considérons les formules d'approximation (3.65) et (3.66), qui consistent en l'interpolation de $A_{\mu_l}^{-1}$ et $A_{\mu_l}^{-1} b$. Le calcul de $A_{\mu_l}^{-1}$ et $A_{\mu_l}^{-1} b$, $1 \leq l \leq N^{\text{EIM}}$, est inhérent à toute méthode d'interpolation ou régression non-intrusive, où le modèle haute-fidélité doit être calculé un certain nombre de fois pour acquérir de l'information sur le modèle. Le coût de calcul de $A_{\mu_l}^{-1}$ et $A_{\mu_l}^{-1} b$, $1 \leq l \leq N^{\text{EIM}}$, est ainsi présent dans toute méthode non-intrusive, et n'est pas relié aux traitements supplémentaires que nous faisons dans la partie offline de notre méthode. L'analyse se réduit donc à évaluer le coût de calcul des coefficients $\lambda_l(\mu)$, $1 \leq l \leq N^{\text{EIM}}$, dans (3.65) et (3.66). Dans nos applications numériques, avec d imposé par la forme du problème, nous déterminons m_0 comme le plus large entier m tel que $Q_{m,d} = \#\bar{\kappa}_{m,d}$ est plus petit que le budget de calcul. Le coût offline correspond ainsi à l'EIM appliqué sur la fonction g sur l'échantillonnage $\bar{\kappa}_{m_0,d} \times \mathcal{P}_{\text{sample}}$. En pratique, comme le nombre d'appels au solveur haute-fidélité est contraint (la plus grande valeur que nous avons prise dans nos applications pour $Q_{m_0,d}$ est 680), nous avons la possibilité de prendre un échantillonnage important de \mathcal{P} , ce qui est désirable de toutes façons vu la possible grande dimension de l'espace paramétrique \mathcal{P} . Si l'EIM est calculé jusqu'à ce que tous les $Q_{m_0,d}$ multi-indices dans $\bar{\kappa}_{m_0,d}$ sont sélectionnés, la complexité algorithmique est alors proportionnelle à $Q_{m_0,d}^3 \times \#\mathcal{P}_{\text{sample}}$. Dans nos simulations numériques, l'étape offline de EIM avec

$\#\mathcal{P}_{\text{sample}} = 10^6$ prend environ autant de temps que la construction du Design Of Experiment (DOE) avec l'algorithme MaxProj lorsque nous comparons nos résultats avec des méthodes de régression statistique. Par exemple, avec $Q_{m_0,d} = 286$ ($m_0 = 2$ et $d = 10$), et $\#\mathcal{P}_{\text{sample}} = 10^6$, la construction du DOE et l'EIM prennent chacun environ 15 minutes.

Notons que dans l'utilisation classique de EIM pour la réduction de modèle non-linéaire, où la solution et/ou l'opérateur sont approchés, il faut évaluer le modèle haute-fidélité sur échantillonnage $\mathcal{P}_{\text{sample}}$, ce qui limite fortement sa taille. Cependant dans notre cas, la fonction g à approcher est connue sur l'ensemble $\bar{K}_{m_0,d} \times \mathcal{P}$ complet, sans résoudre le modèle haute-fidélité, ce qui rend possible un échantillonnage $\mathcal{P}_{\text{sample}}$ important.

Convergence de l'approximation

Nous donnons ici simplement les résultats et les remarques, et renvoyons vers la Publication [Pub.10](#) pour les démonstrations.

Nous donnons deux bornes pour l'erreur faite par l'approximation \mathcal{X}_μ^N de A_μ^{-1} : la première fait intervenir un espace vectoriel abstrait, la deuxième exploite la relation entre les fonctions g_μ , sur laquelle porte l'approximation EIM, et l'objet approché A_μ^{-1} , à travers les schémas itératifs $X_m g_\mu$.

Définissons $Z_N := \frac{1}{N} \sum_{n=1}^N \frac{\int_{\mu \in \mathcal{P}} (g_\mu, \Phi_n)_{\mathcal{U}}^2}{\frac{|\mathcal{P}|}{N} \sum_{p=1}^N (g_{\mu_p}, \Phi_n)_{\mathcal{U}}^2}$, où le dénominateur est une approximation du numérateur, et Φ_n sont les modes obtenus par une snapshots POD faisant intervenir les fonctions g et les valeurs paramétriques μ_i sélectionnées par EIM. Une présentation rigoureuse est disponible dans la Publication [Pub.10](#) (Section 4.1), mais n'est pas reproduite ici car nécessite l'introduction de nombreux objets et notations. Nous notons également τ_N la $N^{\text{ième}}$ valeur propre intervenant dans cette snapshot POD. Le rang de la matrice $\left(g(\vec{k}_i, \mu_j) \right)_{i,j}$, $1 \leq i \leq Q_{m,d}$, $1 \leq j \leq \#\mathcal{P}$ est noté $\hat{Q}_{m,d}(\leq Q_{m,d})$, et nous notons $\delta_{N\text{EIM}}$ une certaine norme de l'erreur de l'approximation EIM. Définissons également $\mathcal{S}^{\mathcal{N}^2 N}$, le plus petit sous-espace vectoriel de $\mathcal{C}^0(\mathcal{P}, V)$ contenant l'image de l'application $v \mapsto \mathcal{J}^N v$, définie par $\forall \mu \in \mathcal{P}$, $(\mathcal{J}^N v)(\mu) := \sum_{n=1}^N (g_\mu, \Phi_n)_{\mathcal{U}} \sum_{p=1}^N G_{np}^{-1} v(\mu_p)$, où $G_{np} = (g_{\mu_n}, \Phi_p)_{\mathcal{U}} \in \mathbb{R}^{N \times N}$ est une matrice inversible et $V := \mathbb{R}^{N \times N}$. Le caractère borné de $(Z_N)_N$, la dimension de $\mathcal{S}^{\mathcal{N}^2 N}$ égale à $\mathcal{N}^2 N$ et l'inversibilité de G sont démontrés dans la Publication [Pub.10](#) (Section 4.3).

Proposition 3.5.2.1. *Pour tout entier m et $1 \leq N < \hat{Q}_{m,d}$,*

$$\frac{1}{|\mathcal{P}|} \int_{\mu \in \mathcal{P}} \|\mathcal{X}_\mu^N - A_\mu^{-1}\|_V^2 \leq 4(1 + N^2 Z_N) (\theta^{sN})^2 + \frac{8}{|\mathcal{P}|} \sup_{\mu \in \mathcal{P}} \|A_\mu^{-1}\|_V^2 \frac{N \delta_N^2}{\tau_N}, \quad (3.68)$$

où

$$\theta^{\mathcal{N}^2 N} := \inf_{\varphi \in \mathcal{S}^{\mathcal{N}^2 N}} \sup_{\mu \in \mathcal{P}} \|A_\mu^{-1} - \varphi(\mu)\|_V. \quad (3.69)$$

Dans le cas $N = \hat{Q}_{m,d}$ où l'approximation EIM est exacte : pour tout entier m

$$\frac{1}{|\mathcal{P}|} \int_{\mu \in \mathcal{P}} \|\mathcal{X}_\mu^{\hat{Q}_{m,d}} - A_\mu^{-1}\|_V^2 \leq 4 \left(1 + \hat{Q}_{m,d}^2 Z_{\hat{Q}_{m,d}} \right) \left(\theta^{\mathcal{N}^2 \hat{Q}_{m,d}} \right)^2. \quad (3.70)$$

Proposition 3.5.2.2. *Pour tout entier m et $1 \leq N < \hat{Q}_{m,d}$,*

$$\frac{1}{|\mathcal{P}|} \int_{\mu \in \mathcal{P}} \|\mathcal{X}_\mu^N - A_\mu^{-1}\|_V^2 \leq 3\epsilon_0^2 \rho^{2m} \left(1 + 2N^2 Z_N + 8 \frac{1}{|\mathcal{P}|} \frac{N \delta_N^2}{\tau_N} \right) + 3 \frac{1}{|\mathcal{P}|} \int_{\mu \in \mathcal{P}} \|X_m(I^N(g_\mu) - g_\mu)\|_V^2. \quad (3.71)$$

Dans le cas $N = \hat{Q}_{m,d}$ où l'approximation EIM est exacte : pour tout entier m

$$\frac{1}{|\mathcal{P}|} \int_{\mu \in \mathcal{P}} \|\mathcal{X}_\mu^{\hat{Q}_{m,d}} - A_\mu^{-1}\|_V^2 \leq 3\epsilon_0^2 \rho^{2m} \left(1 + 2P_d^2(m) Z_{\hat{Q}_{m,d}} \right), \quad (3.72)$$

où nous rappelons que $P_d(m)$ est un polynôme de degré d en m , tel que $\hat{Q}_{m,d} \leq Q_{m,d} \leq P_d(m)$. L'approximation converge sous la condition que $\rho < 1$.

Remarque 3.5.2.1. Les bornes dans (3.68) et (3.71) font intervenir $\frac{N\delta_N^2}{\tau_N}$ et $\|X_m(I^N(g_\mu) - g_\mu)\|_V^2$, qui sont difficiles à décrire : d'une part le comportement asymptotique de $\frac{N\delta_N^2}{\tau_N}$ exhibe une forme indéterminée, d'autre part la norme d'opérateur de X_m est difficile à estimer. Cependant, grâce aux propriétés d'interpolation de EIM, nous savons que $\delta_{\hat{Q}_{m,d}} = 0$ et $I^{\hat{Q}_{m,d}}(g_\mu) = g_\mu$, tandis que $\tau_{\hat{Q}_{m,d}} > 0$: des bornes plus précises peuvent être obtenues dans ce cas particulier.

Dans (3.70), $Z_{\hat{Q}_{m,d}}$ est borné, et $\theta^{\mathcal{N}^2\hat{Q}_{m,d}}$ est relié à une certaine épaisseur de Kolmogorov, qui est habituellement supposée converger de sorte à compenser des croissances exponentielles ou polynomiales dans les problèmes d'approximation; dans notre cas, nous avons seulement besoin de supposer la convergence de $\hat{Q}_{m,d}\theta^{\mathcal{N}^2\hat{Q}_{m,d}}$. Dans (3.72), la convergence est assurée par les propriétés de l'algorithme de puissance considéré, sous l'hypothèse forte que $\rho < 1$, où nous rappelons que $\rho = \sup_{\mu \in \mathcal{P}} \|I - \Psi_0^{-1}A_\mu\|_2$. Cette hypothèse repose sur l'existence d'un bon préconditionneur Ψ_0 uniformément sur l'espace paramétrique \mathcal{P} , possiblement de grande dimensions, ce que nous avons relié plus haut à la réductibilité du problème.

Considérons un sous-ensemble compact de $\mathcal{C}^0(\mathcal{P}, V)$ contenant $\mu \mapsto A_\mu^{-1}$. La deuxième borne dans la Proposition 3.5.2.1 peut être affaiblie dans la forme suivante : pour tout entier m ,

$$\frac{1}{|\mathcal{P}|} \int_{\mu \in \mathcal{P}} \|\mathcal{X}_\mu^{\hat{Q}_{m,d}} - A_\mu^{-1}\|_V^2 \leq 4 \left(1 + \hat{Q}_{m,d}^2 Z_{\hat{Q}_{m,d}}\right) \eta_{\mathcal{S}^{\mathcal{N}^2\hat{Q}_{m,d}}}^2, \quad (3.73)$$

où

$$\eta_{\mathcal{S}^{\mathcal{N}^2\hat{Q}_{m,d}}} := \sup_{v \in F} \inf_{\varphi \in \mathcal{S}^{\mathcal{N}^2\hat{Q}_{m,d}}} \|v - \varphi\|_{\mathcal{C}^0(\mathcal{P}, V)}, \quad (3.74)$$

qui est relié à la $\mathcal{N}^2\hat{Q}_{m,d}$ -épaisseur de Kolmogorov :

$$d_{\mathcal{N}^2\hat{Q}_{m,d}}(F, \mathcal{C}^0(\mathcal{P}, V)) = \inf_{F^{\mathcal{N}^2\hat{Q}_{m,d}} \subset \mathcal{C}^0(\mathcal{P}, V)} \sup_{v \in F} \inf_{\varphi \in F^{\mathcal{N}^2\hat{Q}_{m,d}}} \|v - \varphi\|_{\mathcal{C}^0(\mathcal{P}, V)} = \inf_{F^{\mathcal{N}^2\hat{Q}_{m,d}} \subset \mathcal{C}^0(\mathcal{P}, V)} \eta_{F^{\mathcal{N}^2\hat{Q}_{m,d}}}. \quad (3.75)$$

Pour les besoins de la preuve, nous avons considéré une snapshot POD sur les ensembles $(g_{\mu_l})_{1 \leq l \leq \hat{Q}_{m,d}}$, avec des valeurs μ_n sélectionnées par un premier EIM sur g , ce qui a conduit à un projecteur fixe $(\mathcal{J}^{\hat{Q}_{m,d}v})(\mu) = \sum_{n=1}^{\hat{Q}_{m,d}} (g_\mu, \Phi_n)_{\mathcal{U}} \sum_{p=1}^{\hat{Q}_{m,d}} G_{np}^{-1} v(\mu_p)$ for $v \in \mathcal{C}^0(\mathcal{P}, V)$, et ainsi à un sous-espace fixe $\mathcal{S}^{\mathcal{N}^2\hat{Q}_{m,d}} \subset \mathcal{C}^0(\mathcal{P}, V)$, au lieu du sous-espace optimal $F^{\mathcal{N}^2\hat{Q}_{m,d}}$ in (3.75).

Dans [82], des bornes supérieures pour l'erreur EIM ont été proposées, pour des décroissances polynomiales et exponentielles de la n -épaisseur de Kolmogorov $d_n(\{g_\mu, \mu \in \mathcal{P}\}, \mathcal{U})$. Dans la Proposition 3.5.2.1 sont rendues explicites les dépendances de la borne obtenue par rapport à l'erreur EIM δ_N et à $\theta^{\mathcal{N}^2N}$, qui est relié à l'approximation de $\mu \mapsto A_\mu^{-1}$ dans $\mathcal{C}^0(\mathcal{P}, V)$, et non pas à l'approximation EIM de g_μ in \mathcal{U} .

La convergence des bornes obtenues dans la Proposition 3.5.2.1 est difficile à observer en pratique, compte-tenu des difficultés de l'estimation numérique de δ_N et $\theta^{\mathcal{N}^2N}$. Cependant, la convergence de la borne obtenue dans (3.70) semble raisonnable dans la mesure où, dans les cas réductibles, la convergence de $N\theta^{\mathcal{N}^2N}$ est une hypothèse modérée lorsque $\theta^{\mathcal{N}^2N}$ est remplacé par la \mathcal{N}^2N -épaisseur de Kolmogorov $d_{\mathcal{N}^2N}(F, \mathcal{C}^0(\mathcal{P}, V))$. Dans nos expériences numériques, nous observons que EIM ne donne une erreur d'approximation raisonnable que dans le cas où $N = \hat{Q}_{m,d} = Q_{m,d}$, probablement dû à la forme particulière de $\bar{\kappa}_{m,d}$, qui est un ensemble discret de multi-indices, et non pas l'échantillonnage d'une variable continue : les éléments \vec{k} de $\bar{\kappa}_{m,d}$ semblent générer des éléments linéairement indépendants $\mu \mapsto g_\mu(\vec{k})$ dans $\mathcal{C}^0(\mathcal{P}, V)$. Ceci peut être comparé à la discrete EIM (DEIM), où l'algorithme EIM est appliqué sur les indices d'une liste de vecteurs POD, qui sont tous gardés dans l'approximation [35]. Cependant, l'avantage principal d'EIM dans notre cas est la sélection des paramètres significatifs dans l'échantillonnage de grande taille $\mathcal{P}_{\text{sample}}$ pour notre approximation. Dans nos expériences numériques, nous avons pris $N = Q_{m,d}$ pour cette raison, et nous pouvons évaluer la convergence de l'approximation par rapport à la borne obtenue dans la Proposition 3.5.2.2.

Néanmoins, nous avons obtenu un résultat de convergence assez général dans la Proposition 3.5.2.1, qui pourrait être utile pour d'autres classes de modèles et problèmes. Notons également que avec $\hat{Q}_{m,d}$ évaluation du modèle haute-fidélité A_μ^{-1} , $\theta^{\mathcal{N}^2\hat{Q}_{m,d}}$ fait intervenir une approximation dans un sous-espace vectoriel de dimension de dimension $\mathcal{N}^2\hat{Q}_{m,d}$ de $\mathcal{C}^0(\mathcal{P}, V)$, où \mathcal{N}^2 est la dimension de V .

Expériences numériques

Nous présentons une application sur la matrice inverse d'un problème thermique en dimension paramétrique 2 et un problème de mécanique élastique en dimension paramétrique 14. Pour d'autres applications, notamment sur l'approximation du log-det, nous renvoyons vers la Publication [Pub.10](#) (Section 5).

Considérons un ensemble ouvert $\Omega \subset \mathbb{R}^3$ maillé avec des tétraèdres, voir Figure 3.16. Il s'agit d'un maillage très simplifié d'une aube de turbine haute pression avec 3 canaux internes de refroidissement. L'intersection de ces canaux avec Ω est notée $\partial\Omega_C$. Nous considérons le problème de thermique suivant :

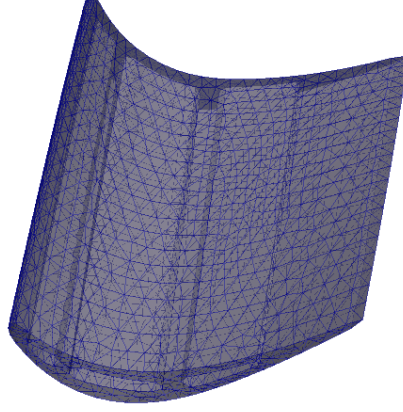


FIGURE 3.16 – Maillage du modèle simplifié d'aube de turbine haute pression

$$\begin{cases} -\vec{\nabla} \cdot \vec{q} = \mu_2 u_\mu & \text{in } \Omega, \\ \vec{q} \cdot \vec{n} = -1 & \text{on } \partial\Omega_C, \\ \vec{q} \cdot \vec{n} = 0 & \text{on } \partial\Omega \setminus \partial\Omega_C, \end{cases} \quad (3.76)$$

où u_μ est la température, $\vec{q} = -\mu_1 \vec{\nabla} u_\mu$ est la densité de flux de chaleur, et $\mu = (\mu_1, \mu_2) \in \mathcal{P} := (1, 4)^2$ est le paramètre. Dans ce problème, μ_1 est la conductivité thermique, et $\mu_2 u_\mu$ est un terme de source volumique dépendant de la solution u_μ .

Notons $\mathcal{V}_h(\Omega)$ l'espace des éléments finis P1 associés au maillage considéré de Ω , où h est la longueur caractéristique des tétraèdres du maillage. La forme faible de (3.76) peut être approchée par

$$A_\mu U_\mu = b, \quad (3.77)$$

où $A_\mu = \mu_1 A_1 + \mu_2 A_2$, avec $(A_1)_{i,j} = \int_\Omega \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi_j$ et $(A_2)_{i,j} = \int_\Omega \phi_i \phi_j$, et $b_i = \int_{\partial\Omega_C} \phi_i$; $\{\phi_i\}_{1 \leq i \leq \mathcal{N}}$ est la base éléments finis P1, où $\mathcal{N} = 3,296$ dans cet exemple. L'approximation $u_{\mu_h} \in \mathcal{V}_h$ de la solution u_μ de (3.76) est obtenue comme $u_{\mu_h} := \sum_{i=1}^{\mathcal{N}} U_{\mu_i} \phi_i$.

Nous comparons l'approximation (3.66) avec d'autres méthodes disponibles pour approcher des solutions de systèmes linéaires paramétrés :

1. (Minimisation de la norme de Frobenius) Soit $Y_{Q_{m,d}} = \text{Span}\{A_{\mu_1}^{-1}, \dots, A_{\mu_{Q_{m,d}}}^{-1}\}$, et

$$P_{Q_{m,d}}(\mu) := \underset{P \in Y_{Q_{m,d}}}{\operatorname{argmin}} \|I - PA(\mu)\|_F, \quad (3.78)$$

où $\|\cdot\|_F$ représente la norme de Frobenius. D'après [122], $P_{Q_{m,d}}(\mu) = \sum_{l=1}^{Q_{m,d}} \lambda_l(\mu) A_{\mu_l}^{-1}$, avec $\lambda(\mu)$ étant la solution de

$$M(\mu)\lambda(\mu) = S(\mu), \quad (3.79)$$

avec $M_{i,j}(\mu) = \text{trace}(A_\mu^T A_{\mu_i}^{-T} A_{\mu_j}^{-1} A_\mu)$ et $S_i(\mu) = \text{trace}(A_{\mu_i}^{-1} A_\mu)$. $P_{Q_{m,d}}(\mu)$ est ainsi la meilleure approximation de A_μ^{-1} exprimée comme une combinaison linéaire de matrices inverses, pour la distance choisie. Cependant, la construction de $M(\mu)$ et $S(\mu)$ requiert la construction online de $A_{\mu_i}^{-1} A_\mu$, $1 \leq i \leq Q_{m,d}$, ce qui est coûteux en temps de calcul – le but de [122] est de proposer des approximations efficaces

de (3.78). Ici, nous utilisons (3.56) pour écrire $M_{i,j}(\mu) = \sum_{l,m=1}^{Q_{m,d}} \alpha_l(\mu) \alpha_m(\mu) \text{trace}(A_l^T A_{\mu_i}^{-T} A_{\mu_j}^{-1} A_m)$ et $S_i(\mu) = \sum_{l=1}^{Q_{m,d}} \alpha_l(\mu) \text{trace}(A_{\mu_i}^{-1} A_l)$, où les matrices $(\text{trace}(A_l^T A_{\mu_i}^{-T} A_{\mu_j}^{-1} A_m))_{i,j}$, $1 \leq l, m \leq Q_{m,d}$ et membres de droite $(\text{trace}(A_{\mu_i}^{-1} A_l))_i$, $1 \leq l \leq Q_{m,d}$ peuvent être précalculés – ce qui est tout de même bien plus coûteux que l’algorithme que nous proposons.

Enfin, l’approximation de (3.77) peut être écrite $u_{\mu_h}(\mu) \approx \sum_{l=1}^{Q_{m,d}} \lambda_l(\mu) u_{\mu_h}(\mu_l)$, avec $\lambda(\mu)$ étant la solution de (3.79). Notons que cette méthode est intrusive, car elle nécessite d’accéder à la matrice A_μ , au lieu de seulement nécessiter les solutions $u_{\mu_h}(\mu)$ pour les stratégies non-intrusives.

2. (Proper Orthogonal Decomposition (POD) [105]) D’abord, nous construisons $M \in \mathbb{R}^{Q_{m,d} \times N}$ tel que $M_{i,j} = u_{\mu_h}(\mu_i)_j$. Ensuite, nous calculons la décomposition aux valeurs singulières (SVD) de $M : M = WSV$, avec $S \in \mathbb{R}^{Q_{m,d} \times N}$ contenant les valeurs singulières de M sur sa diagonale et zéro ailleurs, et $W \in \mathbb{R}^{Q_{m,d} \times Q_{m,d}}$ et $V \in \mathbb{R}^{N \times N}$ étant des matrices unitaires. Les \hat{N} plus grandes valeurs singulières sont gardées, suivant un critère de précision, et notons $\hat{V} \in \mathbb{R}^{\hat{N} \times N}$ tel que $\hat{V}_{i,j} = V_{i,j}$, $1 \leq i \leq \hat{N}$, $1 \leq j \leq N$, et $\{\hat{v}_i\}_{1 \leq i \leq \hat{N}}$, tel que $(\hat{v}_i)_j = \hat{V}_{i,j}$, $1 \leq i \leq \hat{N}$, $1 \leq j \leq N$. Enfin, l’approximation de $u_{\mu_h}(\mu)$ est calculées comme $u_{\mu_h}(\mu) \approx \sum_{i=1}^{\hat{N}} \theta_i(\mu) \hat{v}_i$, où $\theta(\mu)$ est la solution de $\hat{A}_\mu \theta(\mu) = \hat{b}$, with $\hat{A}_\mu = \alpha_1(\mu) \hat{V} A_1 \hat{V}^T + \alpha_2(\mu) \hat{V} A_2 \hat{V}^T$, et $\hat{b} = \hat{V} b$. Notons que cette méthode est intrusive, car elle nécessite d’accéder aux matrices A_μ .
3. (Meta-modélisation) La première étape est la même que la méthode précédente : la construction de la base \hat{v}_i , $1 \leq i \leq \hat{N}$ avec la SVD. Pour avoir une comparaison honnête, nous n’utilisons pas les snapshots $u_{\mu_h}(\mu_i)$ pour les μ_i sélectionnés par l’EIM sur $g(\vec{k}, \mu)$, mais nous utilisons un échantillonnage de type hypercube latin de la même taille, calculé avec l’algorithme MaxProj (see [99]), ce qui est la pratique habituelle pour les méthodes de méta-modélisation. L’échantillonnage obtenu de l’espace paramétrique est appelé Design Of Experiment (DOE). Ensuite, nous calculons les coefficients des snapshots sur la base construite : $\alpha_{i,j} = (u_{\mu_i h}, \hat{v}_j)$, $1 \leq i \leq Q_{m,d}$, $1 \leq j \leq \hat{N}$. Enfin, des méta-modèles non-intrusifs sont construits sur les coefficients $\alpha_{i,j}$, pour lesquels les prédictions $\theta_i(\mu)$ à une nouvelle valeur paramétrique μ sont utilisés dans l’approximation obtenue $u_{\mu_h}(\mu) \approx \sum_{i=1}^{\hat{N}} \theta_i(\mu) \hat{v}_i$. Les modèles statistiques considérés sont issus de la communauté du machine learning : (i) Gaussian processes, (ii) gradient boosting regression, (iii) random forests et (iv) Bayesian Ridge regression ; et calculés en utilisant le package python scikit-learn, voir [95].

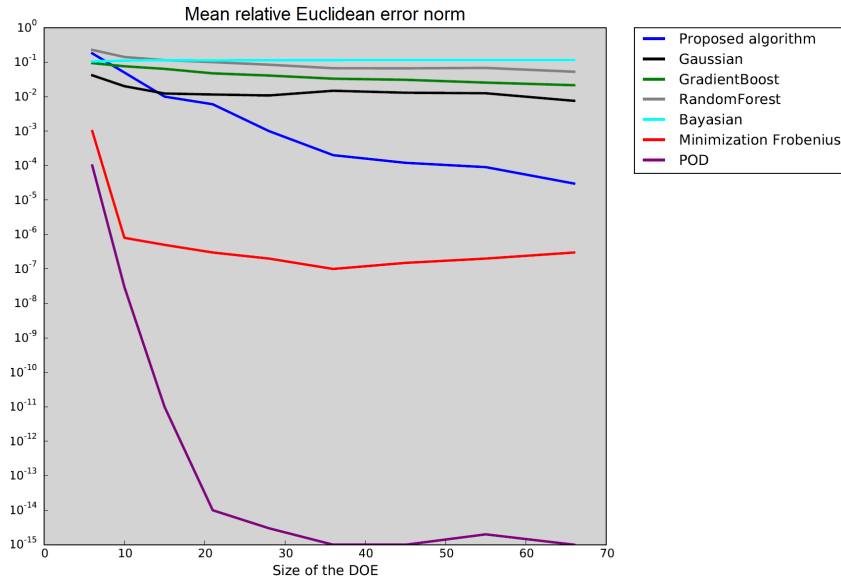


FIGURE 3.17 – Erreur relative moyenne en norme euclidienne des solutions de (3.77) en utilisant diverse méthodes d’approximation, pour 100 valeurs aléatoires du paramètre

La Figure 3.17 montre une comparaison pour les différentes approximations présentées ci-dessus. L’abscisse correspond à la taille du DOE pour les méthodes statistiques et à $Q_{m,d}$, le nombre de valeurs paramétriques sélectionnées par EIM pour notre algorithme, la minimisation de la norme de Frobenius, et la POD. Nous

remarquons que les méthodes intrusives, qui requièrent l'accès à la matrice A_μ sont beaucoup plus performantes que les méthodes non-intrusives. Parmi ces dernières, notre algorithme converge le plus rapidement, sur cet exemple.

Considérons un cube Ω maillé avec des hexaèdres linéaires, avec tous les déplacements fixes sur une face (notée Γ_D) et une pression imposée sur la face opposée (notée Γ_N), les autres faces étant libres. Le domaine contient 6 fibres $\Omega_1, \dots, \Omega_6$, voir Figure 3.18. Nous définissons $\Omega_0 := \Omega \setminus (\cup_{i=1}^6 \Omega_i)$ et considérons le problème suivant d'élasticité linéaire : trouver $u \in H_0^1(\Omega)^3$ tel que $\forall v \in H_0^1(\Omega)^3$

$$\int_{\Omega} \frac{\eta_1}{2} (\nabla u_\mu + {}^t \nabla u_\mu) \cdot (\nabla v + {}^t \nabla v) + \int_{\Omega} \eta_2 (\nabla \cdot u_\mu) (\nabla \cdot v) = \int_{\Gamma_N} (t \cdot n) v, \quad (3.80)$$

où, $H_0^1(\Omega)^3 = \{w \in L^2(\Omega)^3 \text{ tel que } \nabla w \in L^2(\Omega)^{3 \times 3} \text{ et } w|_{\Gamma_D} = 0\}$, η_1 et η_2 sont respectivement les premier et second coefficients de Lamé, $t = t_0 n$ (avec n la normale unitaire dirigée vers l'extérieur et $t_0 = -100 \text{ N.m}^{-2}$) est la pression imposée sur Γ_N , et u_μ est le déplacement inconnu. Notons $\eta_{1,k}$ et $\eta_{2,k}$ respectivement le premier et second coefficients de Lamé dans Ω_k , $0 \leq k \leq 6$. Nous choisissons comme paramètres $\mu = (\eta_{1,0}, \eta_{2,0}, \eta_{1,1}, \eta_{2,1}, \dots, \eta_{1,6}, \eta_{2,6})$, et la décomposition affine (3.56) est obtenue comme $A_\mu = \sum_{l=1}^{14} \alpha_l(\mu) A_l$ avec $(A_{2k})_{i,j} = \int_{\Omega_k} \frac{1}{2} (\nabla \phi_i + {}^t \nabla \phi_i) \cdot (\nabla \phi_j + {}^t \nabla \phi_j)$ and $(A_{2k+1})_{i,j} = \int_{\Omega_k} (\nabla \cdot \phi_i) (\nabla \cdot \phi_j)$, $0 \leq k \leq 6$, $1 \leq i, j \leq \mathcal{N}$ (où $(\phi_i)_{1 \leq i \leq \mathcal{N}}$ est la base d'un espace éléments finis approchant $H_0^1(\Omega)^3$, avec $\mathcal{N} = 27,783$), et $\alpha_{2k} = \eta_{1,k}$, $\alpha_{2k+1} = \eta_{2,k}$, $0 \leq k \leq 6$. L'espace paramétrique est défini comme suit : le coefficient de Poisson de référence est 0.3 dans le cube complet et le module de Young pour la fibre est 2×10^9 , et 2×10^6 dans le reste du domaine. A partir de ces valeurs, nous calculons les coefficients de Lamé de référence $(\eta_{1,k}, \eta_{2,k}) = (1.15 \times 10^9, 7.7 \times 10^8)$ pour les fibres (pour $1 \leq k \leq 6$), et $(\eta_{1,0}, \eta_{2,0}) = (1.15 \times 10^6, 7.7 \times 10^5)$ pour le reste du domaine. Trois espaces paramétriques sont considérés, comme le produit tensoriel des intervalles centrées en ces valeurs de référence, et de largeur respectivement 1%, 5% et 10% de la valeur de référence correspondante.

Dans la Figure 3.18 sont représentées les erreurs relatives entre l'algorithme proposé et les Gaussian processes. Nous tirons les mêmes conclusions que l'expérience numérique précédente.

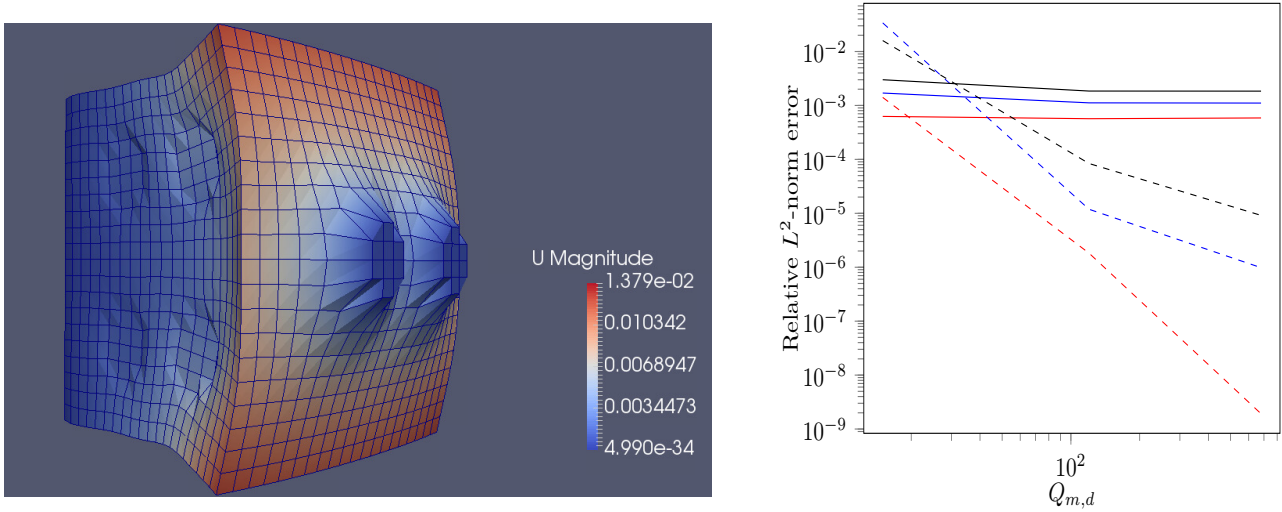


FIGURE 3.18 – Gauche : solution haute-fidélité de (3.80), où le maillage est déformé proportionnellement à la solution ; droite : Erreur relative moyenne en norme euclidienne des solutions en utilisation notre algorithme et les Gaussian processes : — Gaussian 1%, — Gaussian 5%, — Gaussian 10%, - - - algorithm proposé 1%, - - - algorithm proposé 5%, - - - algorithm proposé 10%, le % indique la taille de l'espace paramétrique considéré.

Extension à l'approximation non-intrusive de solutions de problèmes non-linéaires paramétrés

Cette section correspond au Rapport [Rap.1](#).

Nous adaptons l'algorithme présenté à l'approximation non-intrusive de solutions de problèmes non-linéaires paramétrés. Nous illustrons directement la procédure sur un problème de thermique non-linéaire instationnaire. Pour simplifier les notations, nous n'explicitons pas les dépendances en μ . En prenant les mêmes notations pour

la géométrie que celles du problème (3.76) :

$$\begin{cases} \frac{\partial e(u)}{\partial t} - \nabla \cdot (\lambda(u) \nabla u) = q_V \\ \lambda(u) \nabla u = q_S & \text{on } \partial\Omega_C \\ \lambda(u) \nabla u = 0 & \text{on } \partial\Omega \setminus \partial\Omega_C \\ u = u^0 & \text{on } \Omega, \text{ at } t = 0 \end{cases} \quad (3.81)$$

où u est le champ de température inconnu, e est l'énergie interne massique, q_V est une source de température volumique, q_S est une source de température massique, $\lambda(u)$ est la conductivité thermique dépendant de la température, u^0 est la condition initiale. Considérons la formulation variationnelle suivante trouver $u \in H^1(\Omega)$ tel que pour tout $v \in H^1(\Omega)$,

$$\int_{\Omega} \frac{\partial e(u)}{\partial t} v + \int_{\Omega} \lambda(u) \nabla u \cdot \nabla v = \int_{\partial\Omega_N} q_S \cdot n v + \int_{\Omega} q_V v. \quad (3.82)$$

Après discrétisation en temps par un schéma d'Euler implicite, un algorithme de Newton peut être utilisé pour résoudre le système d'équations non-linéaires obtenu :

$$\frac{D\mathcal{F}}{Du}(u_{i\Delta t}^k)(u_{i\Delta t}^{k+1} - u_{i\Delta t}^k) = -\mathcal{F}(u_{i\Delta t}^k), \quad (3.83)$$

où

$$\mathcal{F}(u_{i\Delta t})_r = \int_{\Omega} \frac{e(u_{i\Delta t}) - e(u_{(i-1)\Delta t})}{\Delta t} \varphi_r + \int_{\Omega} \lambda(u_{i\Delta t}) \nabla u_{i\Delta t} \cdot \nabla \varphi_r - \int_{\partial\Omega_N} q_{S,i\Delta t} \cdot n \varphi_r - \int_{\Omega} q_{V,i\Delta t} \varphi_r, \quad (3.84)$$

et

$$\frac{D\mathcal{F}}{Du}(u_{i\Delta t}^k)_{r,s} = \frac{1}{\Delta t} \int_{\Omega} c_p(u_{i\Delta t}^k) \varphi_r \varphi_s + \int_{\Omega} \lambda(u_{i\Delta t}^k) \nabla \varphi_r \cdot \nabla \varphi_s + \int_{\Omega} \frac{d\lambda}{du}(u_{i\Delta t}^k) \varphi_r (\nabla u_{i\Delta t}^k \cdot \nabla \varphi_s), \quad (3.85)$$

avec $c_p := \frac{de}{du}$, la capacité thermique massique et φ_s , une base éléments-finis engendrant un espace approchant $H^1(\Omega)$.

Notons que le dernier terme dans (3.85) est parfois ignoré dans certains solveurs, pour garder une matrice tangente $\frac{D\mathcal{F}_{\mu}}{Du}(u_{i\Delta t}^k)$ symétrique : nous le faisons ici, comme discuté dans la Section 3.2 et [72]. Nous supposons que nous connaissons les dépendances en μ et v des opérateurs $\frac{D\mathcal{F}_{\mu}}{Du}(v)$ et $-\mathcal{F}_{\mu}(v)$, ce qui correspond à connaître les équations (3.81) de la physique considérée. Dans notre cas, cette hypothèse revient à considérer que nous connaissons les fonctions c_p , λ , q_V et q_S . Comme depuis le début de cette section, nous supposons aussi que nous n'avons accès qu'aux solutions calculées par le code à notre disposition : nous pouvons seulement évaluer la fonction $\mathcal{H} : \mathcal{P} \ni \mu \mapsto \mathcal{H}(\mu) := (u_{\mu,i\Delta t})_{1 \leq i \leq N_{\Delta t}}$. Sous ces hypothèses, nous proposons de construire une approximation non-intrusive de \mathcal{H} sous la forme

$$\mathcal{H}(\mu) \approx \left(u_{\mu,(i-1)\Delta t} + \sum_{j=1}^n \lambda_j(\mu, i) \Phi_j \right)_{1 \leq i \leq N_{\Delta t}} \quad (3.86)$$

où $\lambda_j(\mu, i)$, $1 \leq i \leq N_{\Delta t}$, $\mu \in \mathcal{P}$, est évalué efficacement (c'est-à-dire en complexité algorithmique indépendante du nombre de degrés de liberté de la solution) et Φ_j sont des fonctions de l'espace construites au cours d'une phase offline.

L'idée ici est de remplacer la résolution du système linéaire (3.83) par un approximation d'inverse de matrice selon (3.65), où l'hypothèse de décomposition affine (3.56) est obtenue de façon approchée dans un contexte non-linéaire. En approchant le membre de droite par une décomposition affine en paramètre également, nous montrons comment la solution à chaque pas de temps peut être approchée de façon non-intrusive.

Dans un premier temps, la connaissance des équations et l'utilisation d'EIM nous permettent d'obtenir une décomposition affine approchée de la matrice tangente et du second membre de (3.83) pour la première itération de Newton à chaque pas de temps (notons que $u_{\mu,i\Delta t}^0 = u_{\mu,(i-1)\Delta t}$). Nous construisons une approximation EIM pour les fonctions de deux variables $((\mu, t), x) \mapsto c_p(u_{\mu,t}(x))$ et $((\mu, t), x) \mapsto \lambda(u_{\mu,t}(x))$ (le paramètre et le temps ont été regroupés dans la même variable); la phase d'apprentissage de EIM requiert un échantillonnage du

domaine de définition de ces fonctions : nous prenons des valeurs de μ choisies par une Design Of Experiment \mathcal{P}_{DOE} , les valeurs de t sont celles du schéma numérique ($t = i\Delta t$, $1 \leq i \leq N_{\Delta t}$), et les valeurs de x sont les positions des nœuds du maillage de Ω . Notons qu'en pratique, nous n'avons pas besoin de manipuler le maillage ni de connaître la positions des nœuds, nous travaillons seulement avec les composantes du vecteur solution $u_{\mu,t}$. Les approximations EIM sont :

$$\begin{aligned} c_p(u_{\mu,t}(x)) &\approx \sum_{l=1}^{d^{c_p}} \beta_l^{c_p}(\mu, t) c_p(u_{(\mu,t)_l}(x)), \\ \lambda(u_{\mu,t}(x)) &\approx \sum_{m=1}^{d^\lambda} \beta_m^\lambda(\mu, t) \lambda(u_{(\mu,t)_m}(x)). \end{aligned} \quad (3.87)$$

Nous remplaçons les approximations (3.87) dans (3.85) pour obtenir :

$$\begin{aligned} \frac{D\mathcal{F}_\mu}{Du}(u_{\mu,(i-1)\Delta t})_{r,s} &\approx \sum_{l=1}^{d^{c_p}} \beta_l^{c_p}(\mu, (i-1)\Delta t) \left(\frac{1}{\Delta t} \int_{\Omega} c_p(u_{(\mu,(i-1)\Delta t)_l}) \varphi_r \varphi_s \right) \\ &+ \sum_{m=1}^{d^\lambda} \beta_m^\lambda(\mu, (i-1)\Delta t) \left(\int_{\Omega} \lambda(u_{(\mu,(i-1)\Delta t)_m}) \nabla \varphi_r \cdot \nabla \varphi_s \right). \end{aligned} \quad (3.88)$$

Notons que les matrices entre parenthèse sont indépendantes du paramètre et du temps : $((\mu, (i-1)\Delta t)_l)$, $1 \leq l \leq d^{c_p}$, et $(\mu, (i-1)\Delta t)_m$, $1 \leq m \leq d^\lambda$, sont des valeurs choisies du couple $(\mu, (i-1)\Delta t)$. Avec $d = d^{c_p} + d^\lambda$, let $\alpha_l(\mu, (i-1)\Delta t) = \beta_l^{c_p}(\mu, (i-1)\Delta t)$ for $1 \leq l \leq d^{c_p}$ et $\alpha_l(\mu, (i-1)\Delta t) = \beta_{l-d^{c_p}}^\lambda(\mu, (i-1)\Delta t)$ pour $d^{c_p} \leq l \leq d$; et posons $(A_l)_{r,s} = \frac{1}{\Delta t} \int_{\Omega} c_p(u_{(\mu,(i-1)\Delta t)_l}) \varphi_r \varphi_s$ pour $1 \leq l \leq d^{c_p}$ et $(A_l)_{r,s} = \int_{\Omega} \lambda(u_{(\mu,(i-1)\Delta t)_{l-d^{c_p}}}) \nabla \varphi_r \cdot \nabla \varphi_s$ pour $d^{c_p} \leq l \leq d$. Alors, (3.88) peut être réécrit de façon équivalente comme :

$$\frac{D\mathcal{F}_\mu}{Du}(u_{\mu,(i-1)\Delta t}) \approx \sum_{l=1}^d \alpha_l(\mu, (i-1)\Delta t) A_l. \quad (3.89)$$

Ensuite, en considérant l'approximation EIM de $g(\vec{k}, (\mu, t)) = \prod_{l=1}^d \alpha_l^{k_l}(\mu, t) \approx \sum_{l=1}^D \beta_l(\mu, t) g(\vec{k}, (\mu, t)_l)$ nous obtenons, en utilisant (3.65) :

$$\left(\frac{D\mathcal{F}_\mu}{Du}(u_{\mu,(i-1)\Delta t}) \right)^{-1} \approx \sum_{l=1}^D \beta_l(\mu, (i-1)\Delta t) Q_l, \quad (3.90)$$

où $Q_l = \left(\frac{D\mathcal{F}_\mu}{Du}(u_{(\mu,(i-1)\Delta t)_l}) \right)^{-1}$.

Considérons maintenant le membre de droite dans (3.83) pour la première itération de Newton. Comme $u_{\mu,i\Delta t}^0 = u_{\mu,(i-1)\Delta t}$, en utilisant (3.84), nous obtenons :

$$-\mathcal{F}(u_{\mu,(i-1)\Delta t})_r = - \int_{\Omega} \lambda(u_{\mu,(i-1)\Delta t}) \nabla u_{\mu,(i-1)\Delta t} \cdot \nabla \varphi_r + \int_{\partial\Omega_N} q_{\mathcal{S},\mu,(i-1)\Delta t} \cdot n \varphi_r + \int_{\Omega} q_{\mathcal{V},\mu,(i-1)\Delta t} \varphi_r. \quad (3.91)$$

Avec des EIM additionnels sur $((\mu, t), x) \mapsto u_{\mu,t}(x)$, $((\mu, t), x) \mapsto q_{\mathcal{S},\mu,t}(x)$ et $((\mu, t), x) \mapsto q_{\mathcal{V},\mu,t}(x)$, nous obtenons (en utilisant des notations sur le même modèle que (3.87)) :

$$\begin{aligned} &-\mathcal{F}(u_{\mu,(i-1)\Delta t})_r \approx \\ &-\sum_{l=1}^{d^u} \sum_{m=1}^{d^\lambda} \beta_l^u(\mu, (i-1)\Delta t) \beta_m^\lambda(\mu, (i-1)\Delta t) \int_{\Omega} \lambda(u_{(\mu,(i-1)\Delta t)_m}) \nabla u_{\mu,i\Delta t} u_{(\mu,(i-1)\Delta t)_l} \cdot \nabla \varphi_r \\ &+ \sum_{l=1}^{d^{q_s}} \beta_l^{q_s}(\mu, (i-1)\Delta t) \int_{\partial\Omega_N} q_{\mathcal{S},(\mu,(i-1)\Delta t)_l} \cdot n \varphi_r + \sum_{m=1}^{d^{q_v}} \beta_m^{q_v}(\mu, (i-1)\Delta t) \int_{\Omega} q_{\mathcal{V},(\mu,(i-1)\Delta t)_m} \varphi_r. \end{aligned} \quad (3.92)$$

Avec $d' = d^u d^\lambda + d^{q_s} + d^{q_v}$, posons $\gamma_l(\mu, (i-1)\Delta t) = \beta_{l,d^\lambda}^u(\mu, (i-1)\Delta t) \beta_{l\%d^\lambda}^\lambda(\mu, (i-1)\Delta t)$ pour $1 \leq l \leq d^u d^\lambda$ (où $l : d^\lambda$ et $l\%d^\lambda$ sont respectivement le quotient et le reste de la division euclidienne de l par d^λ), et $\gamma_l(\mu, (i-1)\Delta t) =$

$\beta_{l-d^u d^\lambda}^{qs}(\mu, (i-1)\Delta t)$ pour $d^u d^\lambda \leq l \leq d^u d^\lambda + d^{qs}$, et $\gamma_l(\mu, (i-1)\Delta t) = \beta_{l-d^u d^\lambda - d^{qs}}^{qv}(\mu, (i-1)\Delta t)$ pour $d^u d^\lambda + d^{qs} \leq l \leq d^u d^\lambda + d'$; et posons $(b_l)_r = -\int_{\Omega} \lambda(u_{(\mu, (i-1)\Delta t)_{l\%d^\lambda}}) \nabla u_{\mu, i\Delta t} u_{(\mu, (i-1)\Delta t)_{l\%d^\lambda}} \cdot \nabla \varphi_r$ pour $1 \leq l \leq d^u d^\lambda$, et $(b_l)_r = \int_{\partial\Omega_N} q_{S, (\mu, (i-1)\Delta t)_{l-d^u d^\lambda}} \cdot n \varphi_r$ pour $d^u d^\lambda \leq l \leq d^u d^\lambda + d^{qs}$, et $(b_l)_r = \int_{\Omega} q_{V, (\mu, (i-1)\Delta t)_{l-d^u d^\lambda - d^{qs}}} \varphi_r$ pour $d^u d^\lambda + d^{qs} \leq l \leq d^u d^\lambda + d'$. Alors, (3.92) peut être écrit de façon équivalente comme

$$-\mathcal{F}(u_{\mu, (i-1)\Delta t}) \approx \sum_{l=1}^{d'} \gamma_l(\mu, (i-1)\Delta t) b_l. \quad (3.93)$$

Ensuite, en utilisant (3.90) et (3.93) dans (3.83), nous obtenons :

$$\begin{aligned} (u_{\mu, i\Delta t}^1 - u_{\mu, (i-1)\Delta t}) &= \left(\frac{D\mathcal{F}_\mu}{Du} (u_{\mu, (i-1)\Delta t}) \right)^{-1} (-\mathcal{F}(u_{\mu, (i-1)\Delta t})) \\ &\approx \sum_{l=1}^D \sum_{m=1}^{d'} \beta_l(\mu, (i-1)\Delta t) \gamma_m(\mu, (i-1)\Delta t) Q_l b_m. \end{aligned} \quad (3.94)$$

Définissons $\tau_l(\mu, t) = \beta_{l:d'}(\mu, t) \gamma_{l\%d'}(\mu, t)$ et $q_l = Q_{l:d'} b_{l\%d'}$, $1 \leq l \leq Dd'$, de sorte que (3.94) est écrit de façon équivalente comme

$$(\Delta u)_{\mu, i\Delta t} \approx \sum_{l=1}^{Dd'} \tau_l(\mu, (i-1)\Delta t) q_l, \quad (3.95)$$

où $(\Delta u)_{\mu, i\Delta t} := u_{\mu, i\Delta t} - u_{\mu, (i-1)\Delta t}$ est la variation de la solution entre les pas de temps $i-1$ and i . Notons que les q_l sont indépendants de μ et t . Calculons une dernière approximation EIM sur $\tau_l : \tau_l(\mu, t) \approx \sum_{l'=1}^{d''} \delta_{l'}(\mu, t) \tau_l((\mu, t)_{l'})$ et utilisons-là dans (3.95) pour obtenir

$$\begin{aligned} (\Delta u)_{\mu, i\Delta t} &\approx \sum_{l'=1}^{d''} \delta_{l'}(\mu, (i-1)\Delta t) \sum_{l=1}^{Dd'} \tau_l((\mu, (i-1)\Delta t)_{l'}) q_l \\ &= \sum_{l'=1}^{d''} \delta_{l'}(\mu, (i-1)\Delta t) (\Delta u)_{(\mu, i\Delta t)_{l'}}. \end{aligned} \quad (3.96)$$

Nous remplaçons la solution de la première itération de Newton $u_{\mu, i\Delta t}^1$ par la solution $u_{\mu, i\Delta t}$ pour obtenir :

$$u_{\mu, i\Delta t} \approx u_{\mu, (i-1)\Delta t} + \sum_{l'=1}^{d''} \delta_{l'}(\mu, (i-1)\Delta t) (\Delta u)_{(\mu, i\Delta t)_{l'}}. \quad (3.97)$$

Remarque 3.5.2.2. Nous identifions $u_{\mu, i\Delta t}^1$ par $u_{\mu, i\Delta t}$ car, avec nos hypothèses de non-intrusivité, nous n'avons pas accès aux solutions intermédiaires de l'algorithme de Newton. Cependant, lorsque le modèle haute-fidélité calcule plusieurs itérations de Newton, nous faisons mieux que simplement négliger les itérations suivant la première, car dans le membre de droite de (3.97), nous réalisons une combinaison linéaire de variations totales entre pas de temps, et non pas entre un pas de temps et la première itération de Newton. De plus, dans le modèle haute-fidélité, la non-linéarité peut être résolue par d'autres techniques qu'un algorithme de Newton, et notre expression (3.97) peut toujours être calculée car elle ne dépend d'aucune quantité intermédiaire du Newton.

Enfin, pour compresser encore les données et obtenir une expression de la même forme que (3.86), nous calculons une SVD tronquée de la matrice $\left((\Delta u)_{\mu_r: N_{\Delta t}, (r\%N_{\Delta t})\Delta t} \right)_s \in \mathbb{R}^{(\#(\mathcal{P}_{\text{DOE}})N_{\Delta t}) \times \mathcal{N}}$ et nous gardons les vecteurs singuliers droits Φ_j , $1 \leq j \leq n$, pour obtenir l'approximation suivante : $(\Delta u)_{(\mu, i\Delta t)_{l'}} \approx \sum_{j=1}^n ((\Delta u)_{(\mu, i\Delta t)_{l'}}, \Phi_j) \Phi_j$. En utilisant cette approximation dans (3.97) et en notant $\lambda_j(\mu, i) = \sum_{l'=1}^{d''} \delta_{l'}(\mu, (i-1)\Delta t) ((\Delta u)_{(\mu, i\Delta t)_{l'}}, \Phi_j)$, nous obtenons

$$u_{\mu, i\Delta t} \approx u_{\mu, (i-1)\Delta t} + \sum_{j=1}^n \lambda_j(\mu, i) \Phi_j, \quad (3.98)$$

qui est de la forme cherchée (3.86).

Comme application numérique, nous considérons le maillage d'aube de turbine haute pression représenté dans la Figure 3.16, sur lequel le problème (3.81) est résolu, avec $q_V = 0$, $\partial\Omega_N$ étant la surface des canaux

de refroidissement, et q_S étant constant, et $\lambda(u) = 28 - 10 \arctan(11 - 0.01u)$. Les dépendances paramétriques μ_1, μ_2, μ_3 concernent la capacité thermique massique, qui dépend aussi de la température : $c_p(\mu) = 3440\mu_1 - 320\mu_2 \arctan(0.01\mu_3(u - 500))$. Les intervalles de variations considérées sont $\mu_i \in [0.8, 1.2]$, $1 \leq i \leq 3$. La Figure 3.19 montre l'erreur relative en norme euclidienne par rapport à la taille du Design Of Experiment, pour notre algorithme et les méthodes statistiques présentées lors de l'application numérique précédente.

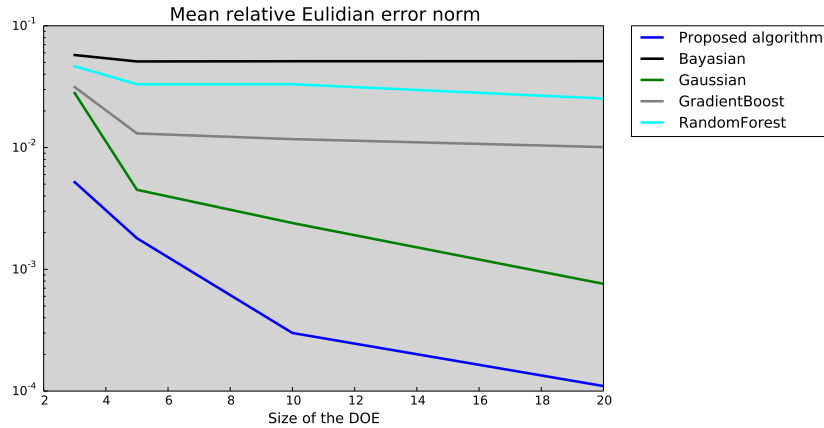


FIGURE 3.19 – Erreur relative de la prédiction en norme euclidienne par rapport à la taille du design of experiment

3.5.3 Un environnement non-intrusif et parallèle en mémoire distribuée pour la réduction de modèle non-linéaire en mécanique des structures : application à l'extrapolation cyclique de matériaux élastoviscoplastiques

Cette section correspond à la Publication [Pub.7](#) et aux Exposés [Exp.3](#), [Exp.4](#) et [Exp.5](#).

Le problème considéré ici est l'estimation rapide de la durée de vie des aubes de turbine, dont la motivation a été donnée dans l'introduction du manuscrit.

Dans l'industrie, les méthodes non-intrusives pour accélérer les simulations numériques sont couramment utilisées. Outre leur avantage pratique, des raisons plus importantes justifient leur utilisation : toutes les méthodologies de simulation sont certifiées en utilisant une certaine version d'un code, et modifier ce code conduirait à une nouvelle procédure de certification ; par ailleurs, les codes commerciaux utilisés par les bureaux d'études bénéficient de contrats d'assistance fournis par les éditeurs du code. Ainsi, développer des méthodes non-intrusives est important pour la diffusion de la réduction de modèle dans l'industrie. La littérature sur les méthodes non-intrusives en réduction de complexité algorithmique est vaste, mais dans leur grande majorité, ces méthodes ne sont pas de type Galerkin, et ont tendance à "oublier" l'équation d'origine. Concernant les méthodes d'apprentissage statistique et les régressions, nous pouvons citer les revues bibliographiques [[13](#), [44](#), [78](#), [88](#), [109](#)]. Dans les méthodes de réduction de modèle *a posteriori*, des efforts ont été fournis pour réduire les contraintes d'intrusivités. Dans la Section 3.5.1, nous avons modifié la méthode des bases réduites en ne requérant au code haute-fidélité que de multiplier l'opérateur linéaire construit par des second-membres fournis par l'utilisateur. Dans [[55](#)], des problèmes paramétriques sont approchés en n'utilisant que la première itération d'un solveur itératif haute-fidélité. Dans [[16](#), [68](#)], une méthode de bases réduites est utilisée, et les coefficients de la solution réduite sur cette base sont calculés en utilisant des méthodes d'interpolation ou de régression au lieu d'utiliser une méthode de Galerkin dans la phase online. Dans [[32](#)], ces coefficients sont approchés en utilisant une méthode à deux grilles et en utilisant un code élément-finis de façon non-intrusive.

Dans cette section, nous proposons une suite logicielle capable de construire des modèles réduits pour la mécanique des structures non-linéaire, de façon non-intrusive, et qui peut traiter des simulations de grande échelle. Nous identifions trois étapes, qui peuvent être exécutées séparément en temps, et possiblement sur des ordinateurs différents : (i) la production des solutions haute-fidélité par un logiciel commercial, (ii) la phase offline de la réduction de modèle et (iii) la phase online où le modèle réduit est exploité. La non-intrusivité suppose que seul le champ solution de déplacement est connu, et toutes les étapes supplémentaires nécessaires à la phase offline sont traitées par des routines python internes. L'applicabilité à un nouveau code commercial

nécessite simplement de convertir le maillage et résultat de calcul vers notre format interne. L'ensemble de la chaîne de calcul peut-être utilisée en séquentiel ou en parallèle en mémoire distribuée, de la génération des solutions haute-fidélité (reposant dans notre cas sur la décomposition de domaine par FETI - Finite Element Tearing and Interconnect [52, 58]), à la visualisation de la solution réduite, en passant par les phases offline et online. La non-intrusivité est illustrée sur des problèmes élastoviscoplastiques utilisant les versions commerciales de Z-set et Ansys Mechanical. La motivation industrielle de ces développements est la simulation efficace du chargement cyclique d'une aube de turbine haute pression, pour la prédiction de la durée de vie.

Problème de référence en mécanique des structures non-linéaire

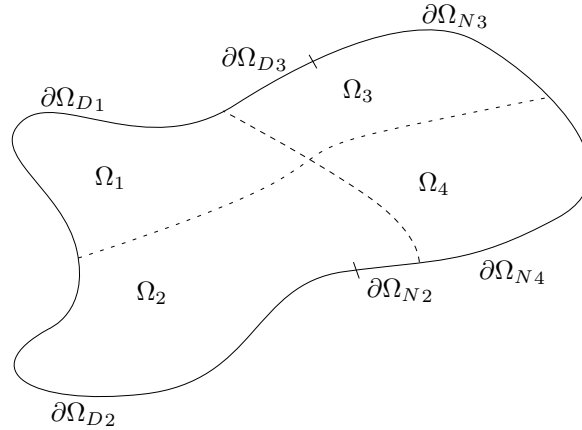


FIGURE 3.20 – Modélisation de la structure d'intérêt Ω dans un contexte sous-structuré.

Considérons une structure, notée Ω , de frontière $\partial\Omega$ partitionnée comme $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ tel que $\partial\Omega_D \cap \partial\Omega_N = \emptyset$, voir la Figure 3.20. La structure est soumise à un chargement quasi-statique et dépendant du temps : une condition de Dirichlet homogène sur $\partial\Omega_D$, une pression dépendant du temps sur $\partial\Omega_N$ et une force volumique f . La réponse de la structure est gouvernée par les équations suivantes :

$$\epsilon(u) = \frac{1}{2} (\nabla u + \nabla^T u) \quad \text{in } \Omega \times [0, T] \quad (\text{compatibilité}), \quad (3.99a)$$

$$\text{div} \sigma + f = 0 \quad \text{in } \Omega \times [0, T] \quad (\text{équilibre}), \quad (3.99b)$$

$$\sigma = \sigma(u, y) \quad \text{in } \Omega \times [0, T] \quad (\text{loi de comportement}), \quad (3.99c)$$

$$u = 0 \quad \text{in } \partial\Omega_D \times [0, T] \quad (\text{encastrement}), \quad (3.99d)$$

$$\sigma \cdot n = T_N \quad \text{in } \partial\Omega_N \times [0, T] \quad (\text{pression}), \quad (3.99e)$$

$$u = 0, y = 0 \quad \text{in } \Omega \quad (\text{condition initiale}), \quad (3.99f)$$

où u est l'inconnue, ϵ est le tenseur des déformations, σ est le tenseur des contraintes de Cauchy, y est le vecteur des variables d'état, et n est la normale extérieure sur $\partial\Omega$. Soit $H_0^1(\Omega) = \{w \in L^2(\Omega) \mid \frac{\partial w}{\partial x_i} \in L^2(\Omega), 1 \leq i \leq 3 \text{ et } w|_{\partial\Omega_D} = 0\}$. L'évolution des variables internes y est mise à jour en résolvant la loi de comportement.

Comme notre procédure est non-intrusive, elle a seulement besoin d'une évaluation de la solution de (3.99a)-(3.99f) par un code commercial. Supposons qu'une quantification de la précision du résultat soit possible, et fixée à $\epsilon_{\text{Newton}}^{\text{HFM}}$.

Une procédure non-intrusive et parallèle en mémoire distribuée

La suite logicielle non-intrusive consiste en un ensemble de classes python, parallélisées en utilisant MPI for Python (mpi4py, voir [43]). Lorsqu'elle est utilisée en parallèle, elle suppose que le résultat haute-fidélité, les chargements et le maillage sont fournis par le code commercial indépendamment pour une collection de sous-domaines Ω_l , $1 \leq l \leq n_d$, constituant une partition de la structure Ω : $\Omega_l \cap \Omega_{l'} = \emptyset$ pour tout $l \neq l'$, et

$$\bigcup_{l=1}^{n_d} \Omega_l = \Omega, \text{ voir la Figure 3.20.}$$

Phase *offline* : une procédure en trois étapes

Les trois étapes de la procédure sont

1. (Génération des données) C'est la génération des snapshots u_s , $1 \leq s \leq N_c$, par le modèle haute-fidélité considéré, où snapshot réfère aux solutions haute-fidélité à notre disposition. Remarquons que le comportement global de la structure n'est supposé connu que via ces données. L'histoire complète des quantités duales est reconstruite sur un maillage de Ω en utilisant un solveur de loi de comportement et l'information des champs de déplacement. Pour cela, notre code est interfacé au solveur de loi de comportement Zmat [90].
2. (Compression des données) C'est le calcul de la base d'ordre réduit $(\psi_i)_{1 \leq i \leq n}$, où $n \ll N$ (avec N le nombre de degrés de liberté du modèle haute-fidélité associé), obtenue par post-traitement des snapshots, en cherchant une structure de rang-faible. Nous utilisons la snapshot POD pour ses bonnes performances en parallèle avec mémoire distribuée, (voir Algorithm 3 et [34, 105]).

Algorithm 3 Compression des données par snapshot POD

1. Choisir une tolérance ϵ_{POD}
 2. Calculer la matrice des corrélations $C_{i,j} = \int_{\Omega} u_i \cdot u_j$, $1 \leq i, j \leq N_c$
 3. Calculer une décomposition aux valeurs propres tronquée à ϵ_{POD} de C , avec ξ_i et λ_i , $1 \leq i \leq n$, respectivement les vecteurs propres et valeurs propres de C
 4. Calculer la base d'ordre réduit $\psi_i(x) = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{N_c} u_j(x) \xi_{ij}$, $1 \leq i \leq n$
-

Chaque sous-domaine peut construire sa matrice de corrélation locale, et la matrice de corrélation globale peut être reconstruite par chaque processus au prix d'une communication MPI de type all-to-all, pour un objet de taille $\frac{1}{2}N_c(N_c + 1)$. La construction des matrices des produits scalaires $L^2(\Omega_l)$, $1 \leq l \leq n_d$, est également faite en parallèle.

3. (Compression des opérateurs) C'est le traitement additionnel requis pour garantir l'efficacité de l'étape online. Cette étape est présente pour toutes les applications en réduction de modèle, indépendamment de la linéarité ou la non-linéarité du modèle haute-fidélité ou de la structure des dépendances paramétriques – mais dans les cas les plus compliqués, une approximation est introduite. Nous renvoyons aux discussions dans la Publication Pub.7.

Trois méthodes ont été introduites simultanément pour l'étape de compression des opérateurs pour les problèmes non-linéaire : la Missing Point Estimation (MPE) [15], l'Empirical Interpolation Method (EIM) [17] (présenté dans l'introduction), et l'hyperreduction [100]. D'autres méthodes ont été introduites plus tard, comme la Best Point Interpolation Method (BPIM) [92], la Discrete Empirical Interpolation Method (DEIM) [35], la méthode Gauss–Newton with Approximated Tensors (GNAT) [30], l'Energy Conserving Sampling and Weighting (ECSW) [50], l'Empirical Cubature Method (ECM) [66] et la procédure LP empirical quadrature [120].

Nous choisissons de construire une quadrature réduite à poids positifs, comme proposé dans les trois dernières de ces méthodes. Considérons le vecteur des forces internes calculé par quadrature exacte par la méthode des éléments finis :

$$\hat{F}_i^{\text{int}}(t) = \sum_{e \in E} \sum_{k=1}^{n_e} \omega_k \sigma(\hat{u}, y)(x_k, t) : \epsilon(\psi_i)(x_k), \quad (3.100)$$

où E est l'ensemble des éléments du maillage, n_e est le nombre de points de quadrature de l'élément e , ω_k et x_k les poids et points d'intégration de l'élément considéré et où le tenseur des contraintes $\sigma(\hat{u}, y)$ pour la solution réduite considérée \hat{u} et les variables d'état y est vu comme une fonction de l'espace et du temps. Notons N_G le nombre total de points d'intégration. Des approximations de (3.101) sont cherchées sous la forme d'une quadrature réduite à poids positifs :

$$\hat{F}_i^{\text{int}}(t) := \int_{\Omega} \sigma(\epsilon(\hat{u}), y)(x, t) : \epsilon(\psi_i)(x) \approx \sum_{k'=1}^d \hat{\omega}_{k'} \sigma(\hat{u}, y)(x_{k'}, t) : \epsilon(\psi_i)(x_{k'}), \quad (3.101)$$

où $\omega_{k'} > 0$ et $x_{k'}$ sont les poids et points d'intégration, et où d est la longueur de la quadrature réduite. Proposée originalement dans [12] en imagerie par ordinateur, la positivité des poids garantit la préservation par l'opérateur réduit des structures spectrales de l'opérateur du modèle haute-fidélité.

Nous cherchons à construire une quadrature la mieux adaptée aux solutions du modèle haute-fidélité à notre disposition : Notons d'abord $f_q := \sigma(u_{(q//n)+1}, y) : \epsilon(\psi_{(q\%n)+1})$, où $//$ et $\%$ sont respectivement le quotient et le reste de la division euclidienne, et \mathcal{Z} est un sous-ensemble de $[1; N_G]$ de taille d . Ensuite, notons $J_{\mathcal{Z}} \in \mathbb{R}^{nN_c \times d}$ et $b \in \mathbb{N}^{nN_c}$ tel que pour tout $1 \leq q \leq nN_c$ et tout $1 \leq q' \leq N_G$,

$$J_{\mathcal{Z}} = \left(f_q(x_{\mathcal{Z}_{q'}}) \right)_{1 \leq q \leq nN_c, q' \in \mathcal{Z}}, \quad b = \left(\int_{\Omega} f_q \right)_{1 \leq q \leq nN_c}, \quad (3.102)$$

où $\mathcal{Z}_{q'}$ denote le q' -ème élément de \mathcal{Z} . Soit $\hat{\omega} \in \mathbb{R}^{+d}$, $(J_{\mathcal{Z}}\hat{\omega})_q = \sum_{q'=1}^d \hat{\omega}_{q'} \sigma(\epsilon(u_{(q//n)+1}), y)(x_{\mathcal{Z}_{q'}}) : \epsilon(\psi_{(q\%n)+1})(x_{\mathcal{Z}_{q'}})$, $1 \leq q \leq nN_c$ est une approximation candidate pour $\int_{\Omega} \sigma(\epsilon(u_{(q//n)+1}), y) : \epsilon(\psi_{(q\%n)+1}) = b_q$, $1 \leq q \leq nN_c$.

Le problème consistant à trouver la meilleure quadrature de longueur d adaptée au calcul du vecteur des énergies internes est écrit (c.f. [66, Equation (23)])

$$(\hat{\omega}, \mathcal{Z}) = \arg \min_{\hat{\omega}' > 0, \mathcal{Z}' \subset [1; N_G]} \|J_{\mathcal{Z}'}\hat{\omega}' - b\|, \quad (3.103)$$

où $\|\cdot\|$ est la norme euclidienne. La notion de bonne formule de quadrature réduite implique un compromis entre la vitesse d'évaluation de l'approximation et sa précision. Comme indiqué dans [50, Section 5.3], qui cite [8], prendre la longueur de la formule de quadrature réduite dans la fonction objectif de l'optimisation conduit à un problème NP-complet. Le calcul d'une solution approchée est un problème qui a été étudiée par la communauté du traitement du signal, où différentes variantes d'algorithmes gloutons ont été proposées. Dans [50], un Sparse NonNegative Least-Squares (NNLS), proposé dans [96], est utilisé pour atteindre une solution en temps raisonnable dans notre contexte de réduction de modèle. Ici, nous considérons une modification de l'Orthogonal Matching Pursuit [84] prenant en compte les contraintes de non-négativité, voir [118, Algorithme 1] et Algorithme 4. Dans ce travail, nous considérons une heuristique

Algorithm 4 Nonnegative Orthogonal Matching Pursuit.

Entrée : J , b , tolérance $\epsilon_{\text{Op.comp.}}$, x_k , $1 \leq k \leq N_G$

Initialisation : $\mathcal{Z} = \emptyset$, $k' = 0$, $\hat{\omega} = 0$ et $r_0 = b$

1. **while** $\|r_{k'}\|_2 > \epsilon \|b\|_2$ **do**
2. $\tau \leftarrow$ indice $\max \left((J_{[1; N_G]})^T r_{k'} \right)$
3. $\mathcal{Z} \leftarrow \mathcal{Z} \cup \tau$
4. $\hat{\omega} \leftarrow \arg \min_{\hat{\omega}' > 0} \|b - J_{\mathcal{Z}}\hat{\omega}'\|_2$
5. $r_{k'+1} \leftarrow b - J_{\mathcal{Z}}\hat{\omega}$
6. $k' \leftarrow k' + 1$
7. **end while**
8. $d \leftarrow k'$
9. $\hat{x}_k := x_{\mathcal{Z}_k}$, $1 \leq k \leq d$

Sortie : $\hat{\omega}_k$, \hat{x}_k , $1 \leq k \leq d$.

simple adaptée au contexte parallèle en mémoire distribuée, introduite dans [51, Section 3.3] et consistant à chercher des quadratures réduites indépendamment dans chaque sous-domaine. Les intégrations sur le domaine complet Ω sont alors simplement obtenues en sommant toutes les quadratures locales.

Nous renvoyons vers la Publication [Pub.7](#) pour des remarques et commentaires sur les détails techniques.

Workflow parallèle en mémoire distribuée

Le workflow des données et opérations parallèles en mémoire distribuée est illustré dans la Figure [3.21](#).

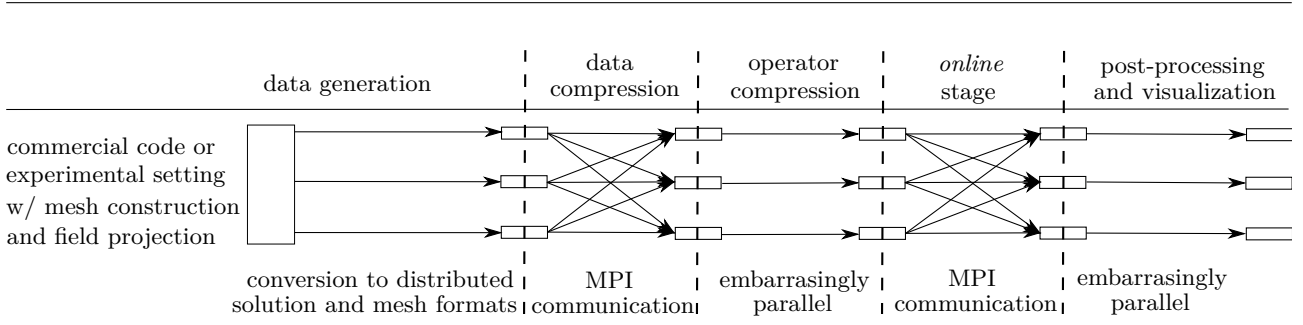


FIGURE 3.21 – Workflow parallèle en mémoire distribuée

Phase *online* : le problème réduit

Le système d'équations non-linéaires réduites est résolu par un algorithme de Newton :

$$\frac{D\mathcal{F}}{Du}(\hat{u}^k)(\hat{u}^{k+1} - \hat{u}^k) = -\mathcal{F}(\hat{u}^k), \quad (3.104)$$

où $\hat{u}^k \in \hat{\mathcal{V}} := \text{Span}(\psi_i)_{1 \leq i \leq n}$ est la k -ème itération du déplacement réduit pour le pas de temps considéré et

$\hat{\mathbf{u}}^k = (\hat{u}_i^k)_{1 \leq i \leq n} \in \mathbb{R}^n$ est tel que $\hat{u}^k = \sum_{i=1}^n \hat{u}_i^k \psi_i$ et où,

$$\frac{D\mathcal{F}}{Du}(\hat{u}^k)_{ij} = \int_{\Omega} \epsilon(\psi_j) : \frac{\partial \sigma}{\partial \epsilon}(\hat{u}^k, y) : \epsilon(\psi_i) \quad (3.105)$$

et

$$\mathcal{F}(\hat{u}^k, y)_i = \int_{\Omega} \sigma(\hat{u}^k, y) : \epsilon(\psi_i) - \int_{\Omega} f \cdot \psi_i - \int_{\partial\Omega_N} T_N \cdot \psi_i. \quad (3.106)$$

L'algorithme de Newton s'arrête lorsque la norme du résidu réduit divisée par la norme du vecteur réduit des forces extérieures est plus petite qu'une certaine tolérance, notée $\epsilon_{\text{Newton}}^{\text{ROM}}$. L'étape online est dite efficace si (3.104) peut être construit en complexité algorithmique indépendante de N , ce qui est le cas ici par le calcul des intégrales sur Ω par les quadratures réduites construites dans l'étape de compression des opérateurs.

Notons que le modèle réduit permet de construire une approximation du déplacement sur le maillage haute-fidélité, mais les quantités duales (tenseurs des contraintes et déformations et variables d'état de la loi de comportement) ne sont calculées que aux points d'intégration sélectionnés dans l'étape de compression des opérateurs. Pour reconstruire ces quantités duales sur le maillage complet, nous utilisons une procédure basée sur la DEIM et la Gappy-POD, adaptée au contexte parallèle en mémoire distribuée et présentée dans l'Algorithme 5 (voir [48] pour la présentation originelle de la Gappy-POD).

Algorithm 5 Reconstruction des quantités duales (DEIM et Gappy-POD) : exemple de la plasticité cumulée p

Phase offline : considérons le sous-domaine Ω_l , $1 \leq l \leq n_d$

1. Choisir une tolérance $\epsilon_{\text{Gappy-POD}}$
2. Appliquer la snapshot POD (Algorithme 3) sur les snapshots haute-fidélité p_s^l , $1 \leq s \leq N_c$ pour obtenir les vecteurs ψ_i^l , $1 \leq i \leq n^l$, orthonormés par rapport au produit scalaire $[L^2(\Omega)]^3$
3. Appliquer EIM sur la collection de vecteurs ψ_i^l , $1 \leq i \leq n^l$ pour sélectionner n^l indices distincts $\{j_k^l\}_{1 \leq k \leq n^l}$ ($1 \leq j_k^l \leq N_G$), et compléter (sans répétition) cet ensemble d'indices pour les indices de la quadrature locale pour obtenir les ensembles $\{j_k^l\}_{1 \leq k \leq m^l}$ ($1 \leq j_k^l \leq N_G$), où $m^l \leq n^l + d^l$
4. Construire les matrices $M^l \in \mathbb{N}^{n^l \times n^l}$ telles que $M_{i,j}^l = \sum_{k=1}^{m^l} \psi_i^{p^l}(x_{j_k^l}) \psi_j^{p^l}(x_{j_k^l})$ (produits scalaires "gappy" des modes POD)

Phase online : considérons un instant t et un sous-domaine l , $1 \leq l \leq n_d$

1. Construire $\beta \in \mathbb{R}^{n^l}$, avec $\beta_i = \sum_{k=1}^{m^l} \psi_i^p(\hat{x}_{j_k^l}) \hat{p}_k$, où $\hat{\mathbf{p}} \in \mathbb{R}^{m^l}$ est tel que \hat{p}_k est la prédiction *online* de p au temps t et au point d'intégration $\hat{x}_{j_k^l}$ (obtenus à partir d'évaluations de la loi de comportement)
 2. Résoudre le système linéaire (de petite taille) : $M\mathbf{w} = \beta$
 3. Calculer la reconstruction de p^l sur le sous-domaine complet Ω^l avec $\sum_{i=1}^{n^l} w_i \psi_i^l$
-

Proposition 3.5.3.1 (La reconstruction est bien définie). *Le système linéaire considéré dans la phase online de l’Algorithme 5 est inversible.*

Extrapolation cyclique pour une aube de turbine haute pression avec Z-set

Nous renvoyons vers la Publication [Pub.7](#) pour l’ensemble des illustrations numériques. Pour cette application, nous utilisons la suite logicielle présentée pour calculer un grand nombre de cycles de chargement d’une aube de turbine haute pression. Nous utilisons la procédure suivante : le premier cycle est calculé avec le modèle haute-fidélité, puis un modèle réduit est construit et exécuté sur les cycles suivants. Notons que ce n’est pas le contexte habituel d’utilisation d’un modèle réduit, dans la mesure où il est exploité pour un seul calcul. Ainsi, nous nous employons à optimiser toutes les étapes de la phase offline et incluant le temps d’exécution de cette phase dans nos mesures de speed-up (qui est donc défini comme le ratio entre les temps d’exécution du modèle haute-fidélité et de notre procédure complète).

Nous considérons un modèle d’aube de turbine haute pression avec quatre canaux de refroidissement. La partie basse de l’aube, appelée le pied, est modélisée par un matériau élastique (même si de la plasticité a effectivement lieu dans cette partie, elle n’est pas dimensionnante pour la durée de vie, donc nous choisissons de ne pas la modéliser) ; tandis que la partie haute de l’aube est modélisée avec un matériau élastoviscoplastique.

La loi de comportement de ce dernier matériau est une loi de Norton avec écrouissage cinématique non-linéaire : la partie élastique est donnée par $\sigma = \mathcal{A} : (\epsilon - \epsilon^{\text{th}} - \epsilon^P)$, où $\epsilon^{\text{th}} = \alpha(T - T_0)I$ est le tenseur des déformations thermiques, avec I le tenseur identité du second ordre et α le coefficient d’expansion thermique en K^{-1} dépendant de la température, et où \mathcal{A} est le tenseur des rigidités et ϵ^P est le tenseur des déformations plastiques. La partie viscoplastique nécessite de résoudre le système d’équations différentielles ordinaires :

$$\begin{cases} \dot{\epsilon}^P = \dot{p} \sqrt{\frac{3}{2}} \frac{s - \frac{2}{3}C\alpha}{\sqrt{(s - \frac{2}{3}C\alpha) : (s - \frac{2}{3}C\alpha)}}, \\ \dot{\alpha} = \dot{\epsilon}^P - \dot{p}D\alpha, \\ \dot{p} = \left\langle \frac{f_r}{K} \right\rangle^m, \end{cases} \quad (3.107)$$

où le critère de plasticité est $f_r \leq 0$ (avec la surface $f_r = \sqrt{\frac{3}{2}} \sqrt{(s - \frac{2}{3}C\alpha) : (s - \frac{2}{3}C\alpha)} - R_0$), p est la plasticité cumulée, α (sans dimension) est la variable interne associée au tenseur back-stress $X = \frac{2}{3}C\alpha$ représentant le centre du domaine élastique dans l’espace des contraintes, $s := \sigma - \frac{1}{3}\text{Tr}(\sigma)I$ (avec Tr l’opérateur de trace) est la partie déviatorique du tenseur des contraintes, et $\langle \cdot \rangle$ denote l’opérateur partie positive. Les coefficients suivants dépendent de la température : limite d’élasticité initiale R_0 (en MPa), coefficient matériau de Norton K (en $\text{MPa} \cdot \text{s}^{\frac{1}{m}}$), coefficient matériau exposant de Norton m (sans dimension) et coefficients d’écrouissage C (en MPa) et D (sans dimension). Les variables d’état considérées ici sont ϵ^P et p , et l’équation différentielle ordinaire est complétée avec les conditions initiales $\epsilon^P = 0$ et $p = 0$ à $t = 0$.

Le modèle haute-fidélité est calculé en parallèle sur 48 cœurs avec Z-set et le solveur AMPFETI (Adaptive MultiPreconditioned FETI [\[27\]](#)), voir Figure [3.22](#). Le chargement est représenté sur la Figure [3.23](#) : l’axe de rotation est dirigé le long de l’axe x (et localisé loin sous l’aube) et la vitesse de rotation a un profil croissant linéairement, puis décroissant linéairement au milieu du cycle ; le champ de température commence par une uniforme 20°C à $t = 0\text{s}$, est interpolé linéairement nœud par nœud jusqu’au champ représenté sur la Figure [3.23 b\)](#) à $t = 20\text{s}$ et décroît linéairement jusqu’à une uniforme 20°C à $t = 40\text{s}$. Les caractéristiques du cas-test sont données dans la Table [3.3](#).

Nombre de degrés de liberté	4,892'463
Nombre de tétraèdres (quadratiques)	1'136'732
Nombre de points de quadrature	5'683'660
Nombre de pas de temps	40 par cycle

TABLE 3.3 – Caractéristique du cas-test d'aube de turbine haute pression

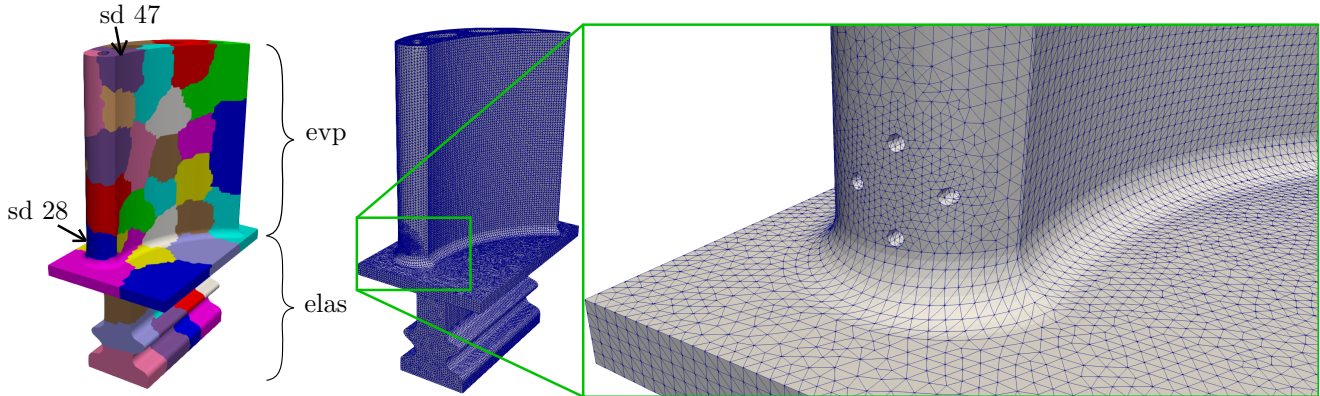
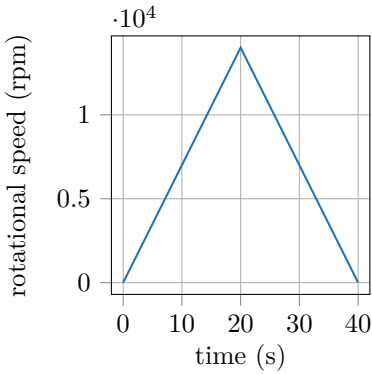
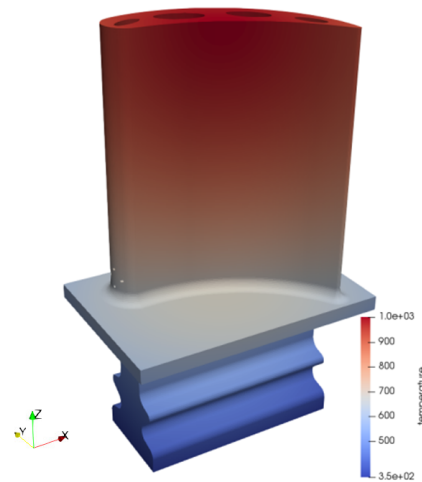


FIGURE 3.22 – a) structure découpée en 48 sous-domaines - la partie supérieure de l'aube est modélisée par une loi élastoviscoplastique et le pied de l'aube par une loi élastique, b) maillage pour l'aube de turbine haute pression avec un zoom autour des canaux de refroidissement.



a) rotational speed



b) temperature field at max loading

FIGURE 3.23 – a) Vitesse de rotation en fonction du temps, b) champ de température au chargement maximum ($t = 20s$).

Des détails sur la procédure de calcul sont donnés dans la Table 3.4. En particulier, toutes les étapes sont exécutées en parallèle en mémoire distribuée, en utilisant le protocole MPI pour la communication entre les processus (48 cœurs sur 2 nœuds). Nous remarquons que dans notre chaîne de calcul, la visualisation est aussi faite en parallèle en mémoire distribuée avec paraview [5]. Les choix pour le critère d'arrêt sont empiriques : ϵ_{Newton}^{HFM} et ϵ_{Newton}^{ROM} sont choisis à 10^{-6} , $\epsilon_{POD} = 10^{-7}$ et $\epsilon_{Gappy-POD} = 10^{-7}$ fournissent la meilleure

Étape	algorithme	tps. de retour (CPUs)
Génération données	solveur AMPFETI (Z-set), $\epsilon_{\text{Newton}}^{\text{HFM}} = 10^{-6}$	2h18min (48)
Compression données	Snapshot POD distribuée, $\epsilon_{\text{POD}} = 10^{-7}$	9min (48)
Compression opérateurs	NonNegative Orthogonal Matching Pursuit distribué, $\epsilon_{\text{Op.comp.}} = 10^{-5}$	44min (48)
Calcul online	ROM pour 100 cycles, $\epsilon_{\text{Newton}}^{\text{ROM}} = 10^{-6}$	3h13min (48)
Reconstruction de σ , ϵ et p	Gappy-POD distribuée, $\epsilon_{\text{Gappy-POD}} = 10^{-7}$	7min (48)

TABLE 3.4 – Description de la procédure complète (p est la plasticité cumulée)

compression possible (en-dessous de ces valeurs, la représentation des données sur la base n'est pas plus précise), $\epsilon_{\text{Op.comp.}} = 10^{-5}$: en dessous de cette valeur, la convergence du Distributed NonNegative Orthogonal Matching Pursuit est très lente, et parfois jamais atteinte dans certains de nos tests numériques. La procédure complète prend 6h31min pour calculer un approximation de la solution sur 100 cycles, tandis que le modèle haute-fidélité prendrait 9 jours et 14h, ce qui correspond à un speedup d'environ 35. Si ce speedup peut apparaître faible, nous soulignons que toute la phase offline a été prise en compte, et que nous utilisons un temps de référence utilisant un solveur AMPFETI à l'état de l'art (un speedup supérieur aurait été obtenu en comparant avec une référence calculée par Abaqus, par exemple). Avec cet exemple, nous illustrons comment un calcul qui prend plusieurs jours avec des solveurs haute-fidélité à l'état de l'art peut être approché en une nuit, ce qui est de grande intérêt pratique dans des phases de design de pièces mécaniques dans l'industrie.

Les points d'intégration sélectionnés à l'étape de compression des opérateurs sont représentés dans la Figure 3.24.

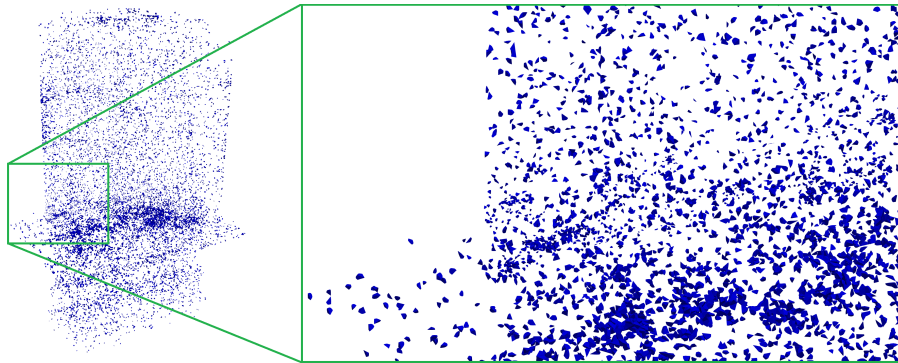


FIGURE 3.24 – Points d'intégration sélectionnés à l'étape de compression des opérateurs.

Pour évaluer la précision du ROM et sa capacité à extrapoler le chargement cyclique, le modèle haute-fidélité a été exécuté sur 10 cycles, pour une durée de 23h20min. En comparant le ROM et le HFM au 10^{ème} cycle, nous vérifions les capacités d'extrapolation entre les cycles 1 à 10. Pour des raisons de ressources, le HFM n'a pas été calculé jusqu'au 100^{ème} cycle. Nous sélectionnons trois sous-domaines critiques sur la structure, et trouvons le point (noeud pour U et point d'intégration pour p) où le champ considéré a la plus grande amplitude. L'erreur relative est définie comme la différence entre le ROM et le HFM, divisé par la prédiction du HFM à ce point. Ci-dessous nous considérons les moyennes de ces erreurs relatives sur les trois sous-domaines identifiés, à chaque pas de temps. Pour $\epsilon_{\text{POD}} = 10^{-3}$, 10^{-5} et 10^{-7} , la snapshot POD sélectionne respectivement 3, 16 et 31 modes POD. Dans la Figure 3.25 sont représentées les erreurs relatives pour l'amplitude du champ de déplacement U et la plasticité cumulée p en fonction du temps, pour différentes valeurs de ϵ_{POD} et $\epsilon_{\text{Op.comp.}}$. Nous observons que, comme anticipé, l'erreur relative est plus faible pour des petites tolérances de ϵ_{POD} et $\epsilon_{\text{Op.comp.}}$. Pour $\epsilon_{\text{POD}} = 10^{-3}$, des petites valeurs de $\epsilon_{\text{Op.comp.}}$ n'améliorent pas la précision de la prédiction, car les erreurs importantes sont causées par la troncature POD. D'autres quantifications et représentations de l'erreur du ROM sont disponibles dans la Publication Pub.7.

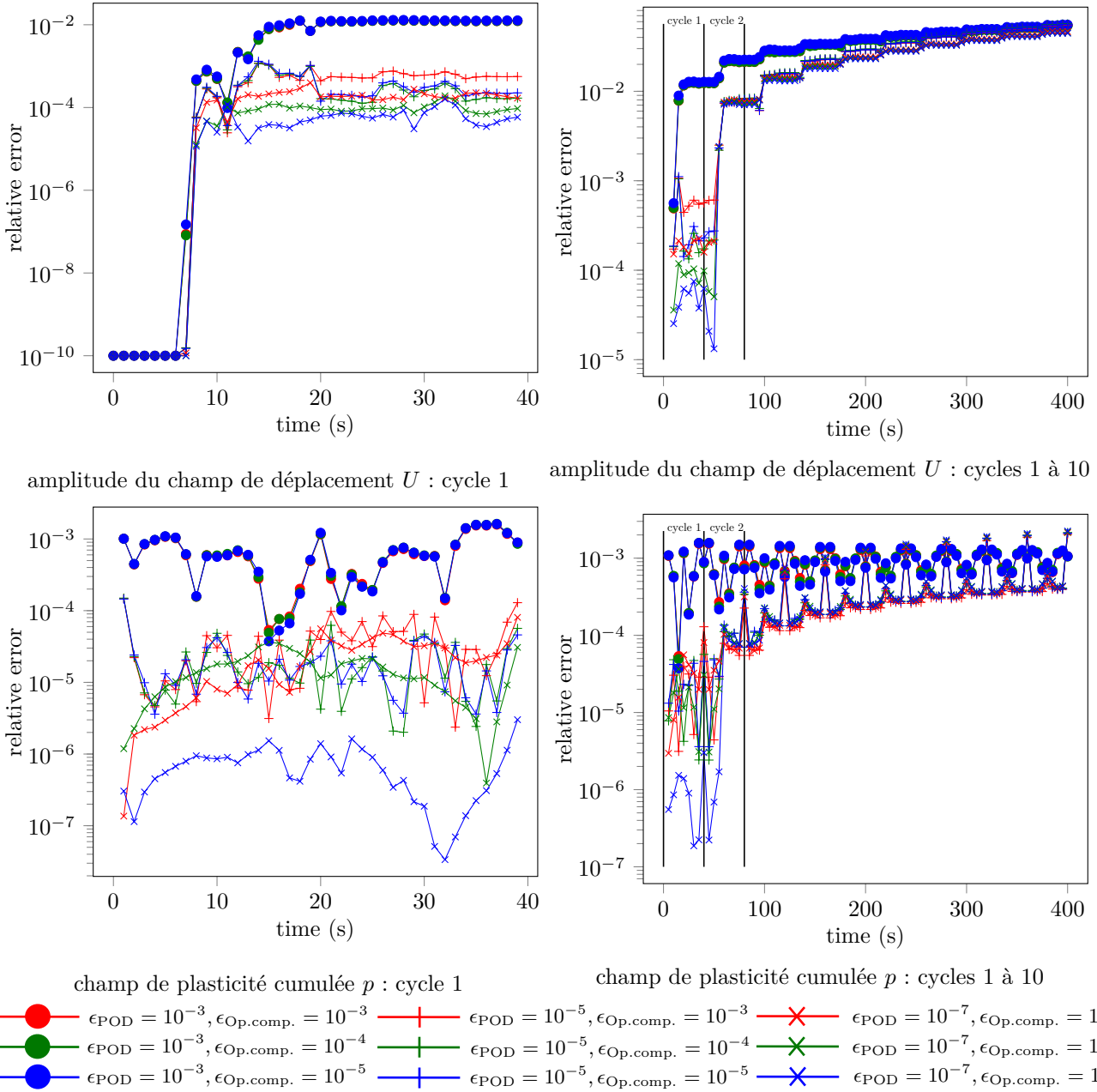


FIGURE 3.25 – Erreur relative pour l’amplitude du champ de déplacement U et le champ de plasticité cumulée p en fonction du temps, pour différentes valeurs de ϵ_{POD} et $\epsilon_{\text{Op.comp.}}$.

3.5.4 Un modèle réduit non-intrusif adaptatif basé sur un indicateur d’erreur, pour la mécanique des structures non-linéaire avec variabilités non-paramétrées

Cette section correspond à la Publication [Pub.9](#) et aux Exposés [Exp.1](#) et [Exp.2](#).

L’application d’intérêt industriel est la même que la Section 3.5.3, et a été motivée dans l’introduction. Dans ce travail, nous cherchons d’une part une indication de l’erreur réalisée par le ROM, et d’autre part à enrichir le modèle réduit par un snapshot calculé par le HFM à chaque fois que cet indicateur d’erreur devient trop important, dans un contexte de variation non-paramétrées. Dans notre contexte de prédiction de la durée de vie des aubes de turbines haute pression, cela consiste à enrichir le ROM au cours de l’extrapolation temporelle de sorte à maîtriser les erreurs. Une autre application consiste en la quantification des incertitudes sur le champ de température que subit l’aube (en pratique, la température en sortie de chambre de combustion est mal connue).

L’estimation d’erreur pour la prédiction des modèles réduits est une discipline qui a été abondamment

traîtée dans la littérature. Nous renvoyons vers la Section 3.4 pour des discussions et des références sur les estimations d'erreur pour la méthode des bases réduites. Même s'ils ne sont pas nécessaires à leur exécution, les estimateurs d'erreur sont désirables dans toutes les autres méthodes de réduction de modèle. Dans la Proper Generalized Decomposition [39], des estimations d'erreur reposant sur la méthode de constitutive relation error sont disponibles [33, 74, 75]. Dans la Proper Orthogonal Decomposition (POD) [34, 105], des estimateurs d'erreur ont été développés pour des problèmes de contrôle optimal pour des problèmes linéaires-quadratiques [108], pour l'approximation de problèmes éléments finis mixtes [79], le contrôle optimal d'équations aux dérivées partielles non-linéaires [71], et pour la réductions de problèmes magnétostatiques [65] et les équations de Navier-Stokes [113]. Pour la réduction de problèmes non-linéaires, la POD a été couplée à des stratégies d'intégration réduite appelées hyperréduction, comme décrit dans la Section 3.5.3, pour lesquels des estimations en relation de constitution ont été proposés [101, 102]. Des études de sensibilités a priori pour l'approximation par POD d'équations paraboliques quasi-non-linéaires sont disponibles [6].

Dans notre contexte de matériau élastoviscoplastique avec modèle de plasticité cristalline, il n'existe pas à notre connaissance d'estimation d'erreur a posteriori : nous cherchons alors un indicateur d'erreur heuristique sur une quantité d'intérêt duale (par exemple plasticité cumulée ou endommagement), calibré sur nos données. La contribution de ce travail consiste en la construction d'un nouvel indicateur d'erreur, adapté à la réduction de modèle non-linéaire en mécanique des structures, où nous sommes intéressés dans la prédiction de quantités duales, comme la plasticité cumulée ou le tenseur des contraintes. Ces quantités duales nécessitent une étape de reconstruction pour être représentées sur la structure d'intérêt complète, ce qui peut être fait par un algorithme de Gappy-POD reposant sur la solution réduite. Nous illustrons que le résidu ROM-Gappy-POD des quantités d'intérêt est fortement corrélé à l'erreur de prédiction du ROM dans nos applications. A partir de cette observation, nous proposons une étape de calibration, reposant sur les données calculées lors de la phase *offline* de la réduction de modèle, pour construire un indicateur d'erreur adapté au problème considéré. L'indicateur d'erreur est ensuite utilisé dans une stratégie d'enrichissement qui améliore la précision de la prédiction réduite. Dans ce travail, nous considérons des variations non-paramétrées entre les configurations d'intérêt, que nous appelons variabilité et notons μ . La variabilité contient les pas de temps et n'importe quelle description non-paramétrée de la configuration, qui dans notre cas est le chargement de l'aube, indiqué par un label. Par exemple, $\mu = \{t = 3, \text{"computation 1"}\}$, signifie que nous considérons le 3ème pas de temps de la configuration "computation 1", pour laquelle nous disposons d'une description du chargement (centre, vitesse et axe de rotation, champs de température et de pression dans nos applications).

Une indicateur d'erreur heuristique

Nous introduisons quelques notations : les vecteurs sont notés en gras (\mathbf{p}_μ est le vecteurs de composantes la valeur de plasticités cumulées haute-fidélité aux points d'intégration réduits), les quantités calculées par le modèle réduit sont notées avec un chapeau (\hat{u}_μ est la prédiction réduite du déplacement et $\hat{\mathbf{p}}_\mu$ est le vecteur de composantes la valeur de la plasticité cumulée réduite aux points d'intégration réduits), alors que les quantités duales reconstruites par Gappy-POD (par exemple \tilde{p}) sont notées avec des tildes. Les symboles gras et avec un tilde, par exemple $\tilde{\mathbf{p}}_\mu$, indiquent le vecteur de composantes la quantités duale reconstruite aux points d'intégration réduits : $\tilde{p}_{\mu,k} = \tilde{p}_\mu(\hat{x}_k)$, $1 \leq k \leq m^p$. Nous remarquons que dans le cas général, $\tilde{\mathbf{p}}_\mu \neq \hat{\mathbf{p}}_\mu$: cet écart est à la base de notre indicateur d'erreur.

L'erreur relative de prédiction est définie par

$$E_\mu^p := \begin{cases} \frac{\|p_\mu - \tilde{p}_\mu\|_{L^2(\Omega)}}{\|p_\mu\|_{L^2(\Omega)}} & \text{si } \|p_\mu\|_{L^2(\Omega)} \neq 0 \\ \frac{\|p_\mu - \tilde{p}_\mu\|_{L^2(\Omega)}}{\max_{\mu \in \mathcal{P}_{\text{off}}} \|p_\mu\|_{L^2(\Omega)}} & \text{sinon,} \end{cases} \quad (3.108)$$

où nous rappelons que p_μ et \tilde{p}_μ sont respectivement la prédiction haute-fidélité et réduite du champ de plasticité cumulé à la variabilité μ , et \mathcal{P}_{off} est l'ensemble des variabilités rencontrées durant la phase *offline*.

Nous définissons le résidu ROM-Gappy-POD comme

$$\mathcal{E}_\mu^p := \begin{cases} \frac{\|\tilde{\mathbf{p}}_\mu - \hat{\mathbf{p}}_\mu\|_2}{\|\hat{\mathbf{p}}_\mu\|_2} & \text{si } \|\hat{\mathbf{p}}_\mu\|_2 \neq 0 \\ \frac{\|\tilde{\mathbf{p}}_\mu - \hat{\mathbf{p}}_\mu\|_2}{\max_{\mu \in \mathcal{P}_{\text{off}}} \|\hat{\mathbf{p}}_\mu\|_2} & \text{sinon,} \end{cases} \quad (3.109)$$

où $\|\cdot\|_2$ dénote la norme euclidienne. Remarquons que l'erreur relative E_μ^p fait intervenir des champs et des normes $L^2(\Omega)$ définies sur la structure complète, alors que le résidu ROM-Gappy-POD \mathcal{E}_μ^p fait intervenir des

vecteurs de quantités duales définies sur les points d'intégration réduits et des normes euclidiennes. Nous pouvons montrer que dans (3.109), $\|\tilde{\mathbf{p}}_\mu - \hat{\mathbf{p}}_\mu\|_2$ est la norme du résidu du problème de minimisation aux moindres carrés intervenant dans la phase online de la reconstruction Gappy-POD de l'Algorithme 5.

Supposons que $K := \{p_\mu, \text{ pour toute variabilité possible } \mu\}$ est un sous-ensemble compact de $L^2(\Omega)$ et définissons la n -épaisseur de Kolmogorov par $d_n(K)_{L^2(\Omega)} := \inf_{\dim(W)=n} d(K, W)_{L^2(\Omega)}$, où $d(K, W)_{L^2(\Omega)} := \sup_{v \in K} \inf_{w \in W} \|v - w\|_{L^2(\Omega)}$, avec W un sous-espace de dimension finie de $L^2(\Omega)$. Notons aussi $\mathbf{\Pi}_\mu := \left((p_\mu, \psi_i^p)_{L^2(\Omega)} \right)_{1 \leq i \leq n^p} \in \mathbb{R}^{n^p}$, où nous rappelons que $\{\psi_i^p\}_{1 \leq i \leq n^p}$ sont les modes Gappy-POD obtenus par l'Algorithm 5 et où $(\cdot, \cdot)_{L^2(\Omega)}$ dénote le produit scalaire $L^2(\Omega)$.

Nous contrôlons d'abord le numérateur dans l'erreur relative E_μ^p par rapport au numérateur dans le résidu ROM-Gappy-POD \mathcal{E}_μ^p dans la Proposition 3.5.4.1.

Proposition 3.5.4.1. *Il existe deux constantes positives C_1 et C_2 indépendantes de μ (mais dépendant de n^p) telles que*

$$\|p_\mu - \tilde{p}_\mu\|_{L^2(\Omega)}^2 \leq C_1 \|\tilde{\mathbf{p}}_\mu - \hat{\mathbf{p}}_\mu\|_2^2 + C_1 \|p_\mu - \hat{p}_\mu\|_2^2 + C_2 d(K, \text{Span}\{\psi_i^p\}_{1 \leq i \leq n^p})_{L^2(\Omega)}^2. \quad (3.110)$$

Nous contrôlons maintenant le numérateur dans le résidu ROM-Gappy-POD \mathcal{E}_μ^p par rapport au numérateur dans l'erreur relative E_μ^p dans la Proposition 3.5.4.2.

Proposition 3.5.4.2. *Il existe deux constantes positives K_1 et K_2 indépendantes de μ telles que*

$$\|\tilde{\mathbf{p}}_\mu - \hat{\mathbf{p}}_\mu\|_2^2 \leq K_1 \|p_\mu - \tilde{p}_\mu\|_{L^2(\Omega)}^2 + K_2 \|p_\mu - \hat{p}_\mu\|_2^2. \quad (3.111)$$

Le corollaire suivant fournit une notion de consistance :

Corollaire 3.5.4.1. *Supposons que la solution réduite soit exacte jusqu'au pas de temps considéré pour la variabilité online $\mu : p_\mu = \tilde{p}_\mu$ dans $L^2(\Omega)$. En particulier, le solver de loi de comportement a été évalué avec le tenseur des déformations et les variables d'état exacts aux points d'intégration x_k , conduisant à $\hat{p}_\mu(\hat{x}_k) = p_\mu(\hat{x}_k)$, $1 \leq k \leq m^d$. En utilisant la Proposition 3.5.4.2, $\|\tilde{\mathbf{p}}_\mu - \hat{\mathbf{p}}_\mu\|_2 = 0$, et $\mathcal{E}_\mu^p = 0$.*

Un indicateur d'erreur calibré

Comme nous allons l'illustrer avec les applications numériques, les évaluations du résidu ROM-Gappy-POD \mathcal{E}_μ^p (3.109) et de l'erreur E_μ^p (3.108) sont très corrélées dans nos configurations. Notre idée est d'exploiter cette corrélation en entraînant un regresseur par processus gaussiens pour la fonction $\mathcal{E}_\mu^p \mapsto E_\mu^p$. À la fin de l'étape *offline*, nous proposons de calculer la prédiction réduite aux valeurs de variabilité $\{\mu_i\}_{1 \leq i \leq N_c}$ rencontrées lors de l'étape de génération des données, et les couples correspondant $(E_{\mu_i}^p, \mathcal{E}_{\mu_i}^p)$, $1 \leq i \leq N_c$. Le regresseur est entraîné sur ces valeurs et nous définissons la fonction

$$\mathcal{E}_\mu^p \mapsto \text{Gpr}^p(\mathcal{E}_\mu^p) \quad (3.112)$$

d'approximation de l'erreur E_μ^p à la variabilité μ comme la moyenne plus 3 fois l'écart type de la distribution de prédiction au point \mathcal{E}_μ^p : c'est l'indicateur d'erreur proposé. Nous utilisons la classe python "GaussianProcessRegressor" de scikit-learn [95]. Nous appelons ces opérations "calibration de l'indicateur d'erreur", voir Figure 3.26 pour une présentation du workflow contenant cette étape de calibration.

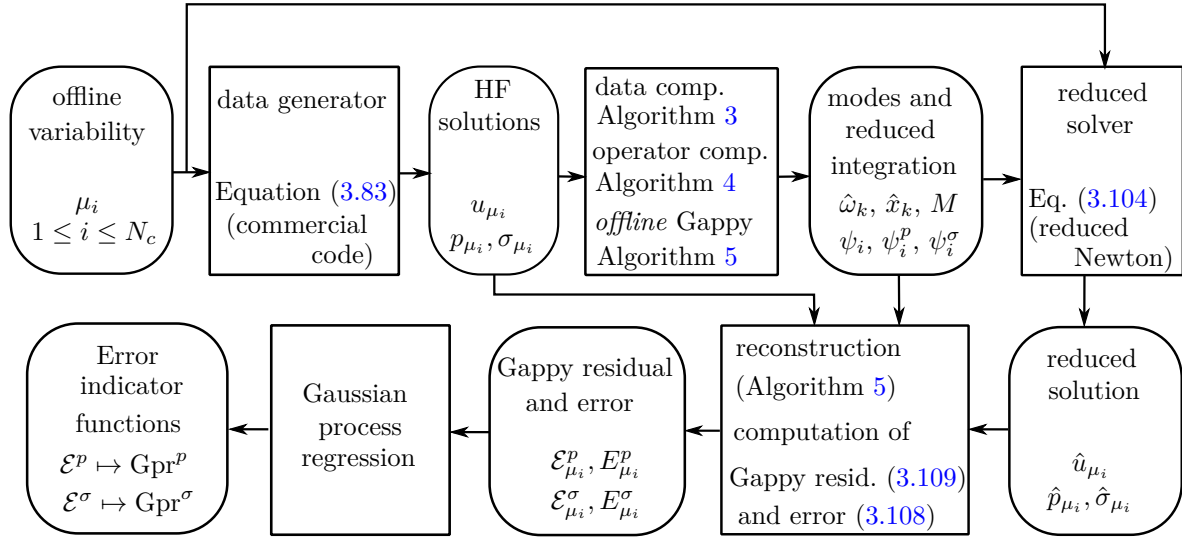


FIGURE 3.26 – Workflow de la phase *offline* avec calibration de l'indicateur d'erreur.

Pour les quantités primales et duales, les modes sont calculés en cherchant une structure de rang faible dans les données haute-fidélité. Les coefficients de la combinaison linéaire pour la reconstruction des quantités primales sont donnés par l'algorithme de Newton réduit (3.104). À convergence, le résidu est petit, même dans les cas où la prédiction réduite n'est pas précise : ce résidu ne donne aucune information sur la distance entre la solution réduite et la solution haute-fidélité. Cependant, dans la phase *online* de la reconstruction Gappy-POD dans Algorithm 5, les coefficients $\hat{p}_{\mu,k}$ contiennent une information haute-fidélité par le biais du solveur de loi de comportement. De plus, un moindre carré sur-déterminé est résolu, ce qui peut fournir un résidu non-nul qui contient implicitement cette information haute-fidélité, c'est la distance entre la prédiction du solveur de loi de comportement et l'espace vectoriel généré par les modes Gappy-POD (restreint aux points d'intégration réduits) : il s'agit du terme $\|\tilde{\mathbf{p}}_{\mu} - \hat{\mathbf{p}}_{\mu}\|_2$ dans (3.110).

Dans la phase *online*, nous avons seulement besoin d'évaluer \mathcal{E}_{μ}^p et n'avons pas besoin d'estimer les autres termes ou les constantes qui apparaissent dans les Propositions 3.5.4.1 and 3.5.4.2.

Équipés d'un indicateur d'erreur efficace, nous pouvons maintenant évaluer la qualité de la prédiction réduite dans la phase *online*. Si l'indicateur d'erreur est trop grand, une étape d'enrichissement se produit : le modèle haute-fidélité est utilisé pour calculer un nouveau snapshot haute-fidélité, qui est ensuite utilisé pour mettre à jour les bases POD et Gappy-POD, ainsi que le schéma d'intégration réduit. Notons que pour que l'étape d'enrichissement soit calculée, le champ de déplacement et toutes les variables d'état doivent être reconstruites sur le maillage complet Ω pour fournir l'état matériau complet au solveur haute-fidélité. Le workflow pour l'étape *online* avec enrichissement est présenté dans la Figure 3.27.

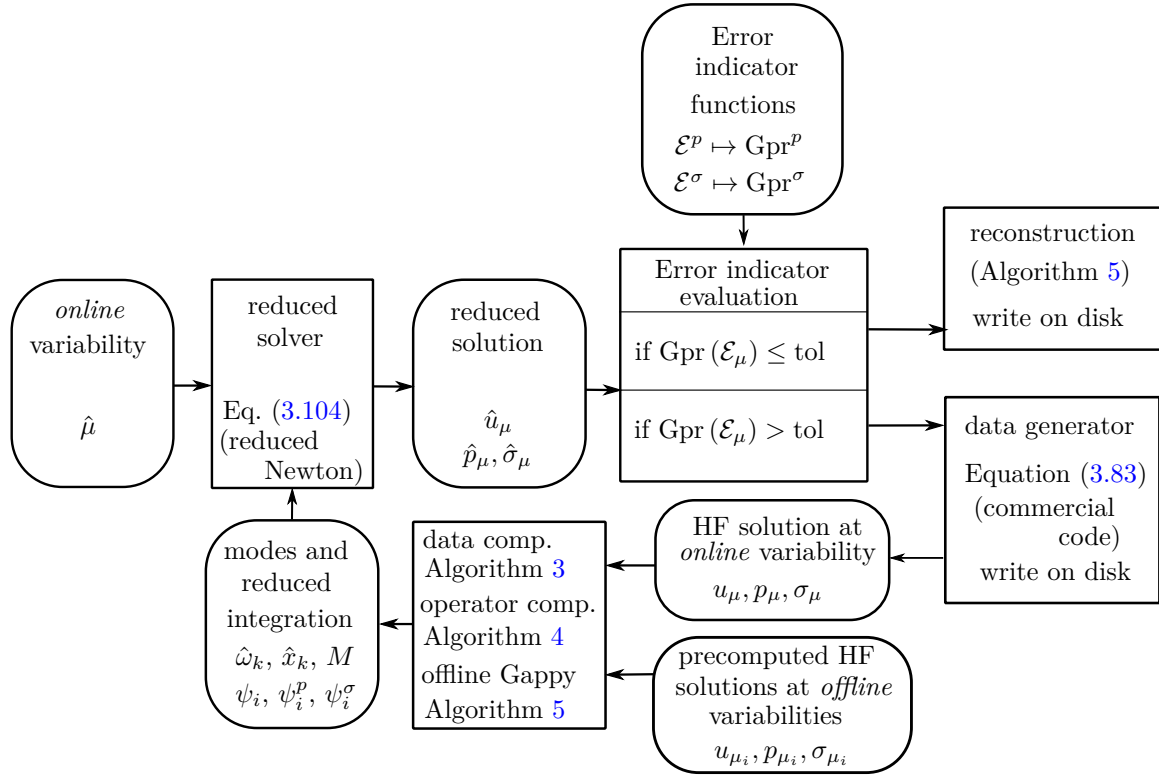


FIGURE 3.27 – Workflow pour l'étape *online* avec enrichissement.

Remarque 3.5.4.1 (efficacité *online*). Le calcul du résidu ROM-Gappy-POD (3.109) est efficace, car $\tilde{\mathbf{p}}_\mu$ et $\hat{\mathbf{p}}_\mu$ sont déjà calculés pour la reconstruction, et m^p dépendant seulement de l'approximation de $\sigma : \epsilon$ et p , ce calcul est en complexité algorithmique indépendante de N . L'évaluation de $\text{Gpr}^p(\mathcal{E}_\mu^p)$ est aussi en complexité indépendante de N .

Application numérique

Une campagne numérique détaillée est présentée dans la Publication [Pub.9](#), où les corrélations entre E_μ^p et \mathcal{E}_μ^p sont illustrées, ainsi que l'amélioration de la prédiction réduite grâce aux enrichissements déclenchés par l'indicateur d'erreur, pour plusieurs configurations académiques et l'aube de turbine présentée dans la Section 3.5.3. Dans ce manuscrit, nous reproduisons uniquement une expérience sur l'aube de turbine.

La géométrie du cas test est représentée dans la Figure 3.22. Le chargement de l'aube diffère de celui de la Section 3.5.3 et est représenté dans la Figure 3.28 : 10 champs de température sont considérés : le plus froid est appliqué aux vitesses de rotation les plus faibles, tandis que le plus chaud est appliqué aux vitesses de rotation les plus élevées. La variabilité *online* diffère de la variabilité *offline* aux trois pas de temps localisés autour des trois derniers maximums du profil de vitesse de rotation, où seulement le champ de température change comme indiqué dans les deux images dans la partie droite de la Figure 3.28 : le maximum de la température est déplacé du centre vers l'avant de l'aube. Comme nous allons voir, cette modification va entraîner de grandes erreurs dans la prédiction du ROM, si aucun enrichissement n'est utilisé.

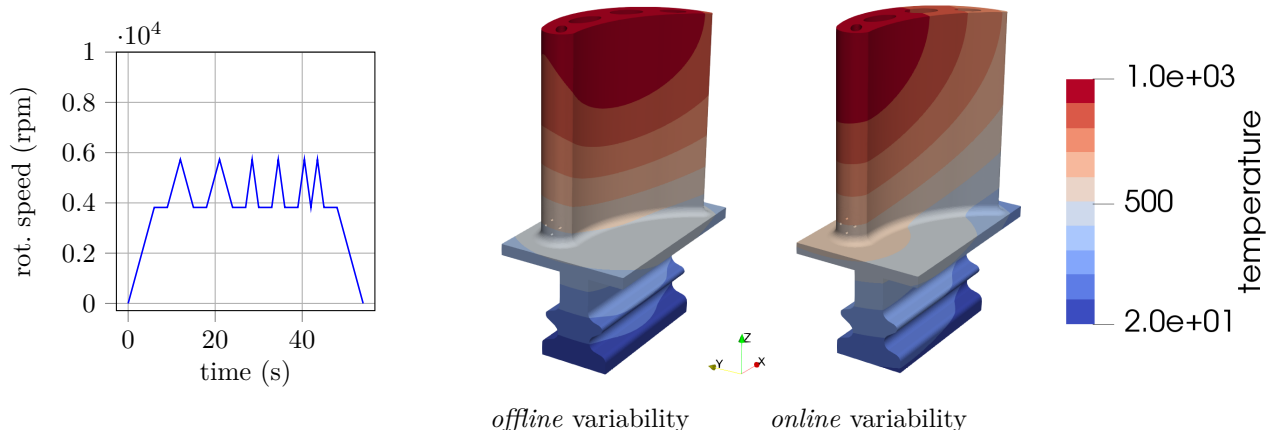


FIGURE 3.28 – Cas test d’aube de turbine haute pression : gauche) vitesse de rotation en fonction du temps ; droite) représentation des champs de température maximums utilisés pour les calculs *offline* et *online* computations ; l’axe de rotation est localisé sous l’aube, le long de l’axe x .

Les corrélations entre le résidu ROM-Gappy-POD \mathcal{E} (3.109) et l’erreur E (3.108) sur les quantités duales plasticité cumulée p et tenseur des contraintes σ sont illustrées dans la Table 3.5. Pour la plasticité cumulée, les valeurs avant les premiers effets plastiques sont négligées. Une corrélation forte apparaît dans tous les cas considérés, bien que quelques outliers soient observés dans les derniers pas de temps, où l’augmentation des contraintes résiduelles pour des chargements faibles sont plus difficiles à prédire avec le modèle réduit.

	p	σ_{xx}
subdomain 28		
subdomain 47		

TABLE 3.5 – Illustration de la corrélation entre le résidu ROM-Gappy-POD \mathcal{E} (3.109) et l’erreur E (3.108) sur les quantités duales plasticité cumulée p et une composante du tenseur des contraintes σ .

Dans la Table 3.6, nous comparons l’indicateur d’erreur (3.112) avec l’erreur (3.108) pour les variabilités *offline* et *online* considérées. Les valeurs de l’indicateur d’erreur sont plus grandes que l’erreur, sauf dans des cas d’erreur très importantes (pour lesquelles le ROM est inutile), et quelques fois dans les derniers pas de temps, où les contraintes résiduelles sont importantes. Sans enrichissement, le ROM fait des erreurs très importantes. Nous observons que le sous-domaine dans lequel le critère d’enrichissement est utilisé permet de contrôler l’erreur dans

le sous-domaine correspondant, alors que l'erreur est plus importante dans les autres sous-domaines. Cela illustre que des quantités d'intérêt locales (en espace) peuvent être utilisées pour éviter que des étapes d'enrichissement se produisent trop souvent.

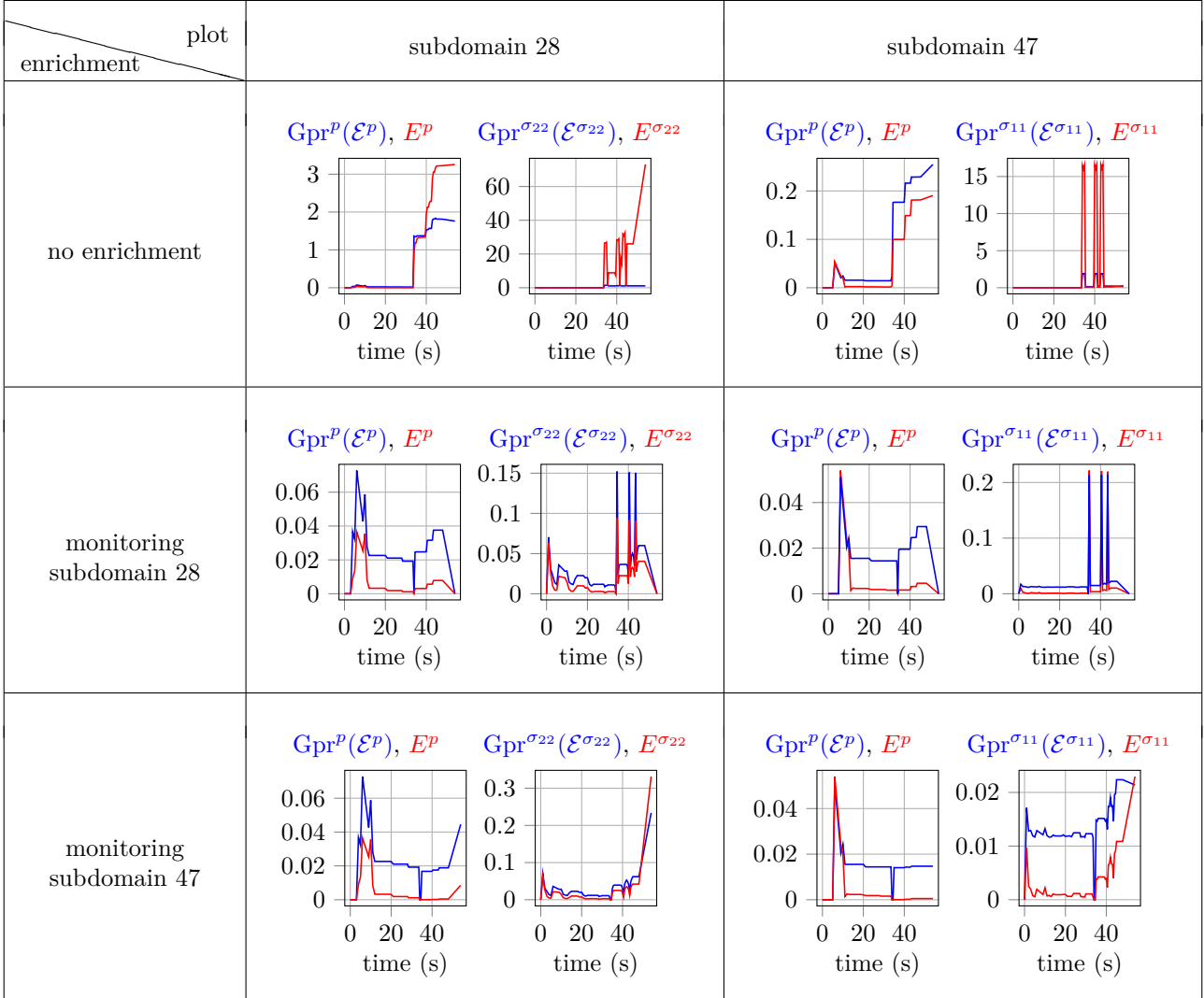


TABLE 3.6 – Comparaison de l'indicateur d'erreur (3.112) avec l'erreur (3.108) pour les variabilités *offline* et *online* considérées. La catégorie “plot” pour les colonnes réfère au sous-domaine où l'indicateur d'erreur et l'erreur sont tracés, tandis que la catégorie “enrichment” pour les lignes réfère au sous-domaine duquel l'indicateur est utilisé pour décider des étapes d'enrichissement.

Dans la Figure 3.29 sont illustrées plusieurs prédictions pour les quantités duales : l'indice “off.” réfère à la prédiction haute-fidélité pour la variabilité *offline*, “ref.” à la référence haute-fidélité pour la variabilité *online*, “nores.” au ROM sans enrichissement, “sd28” au ROM avec enrichissement en monitorant l'indicateur d'erreur sur le sous-domaine 28, et “sd47” au ROM avec enrichissement en monitorant l'indicateur d'erreur sur le sous-domaine 47. Nous observons que sans enrichissement, le ROM souffre d'erreurs importantes. Avec enrichissement, le sous-domaine monitoré a une prédiction réduite précise.

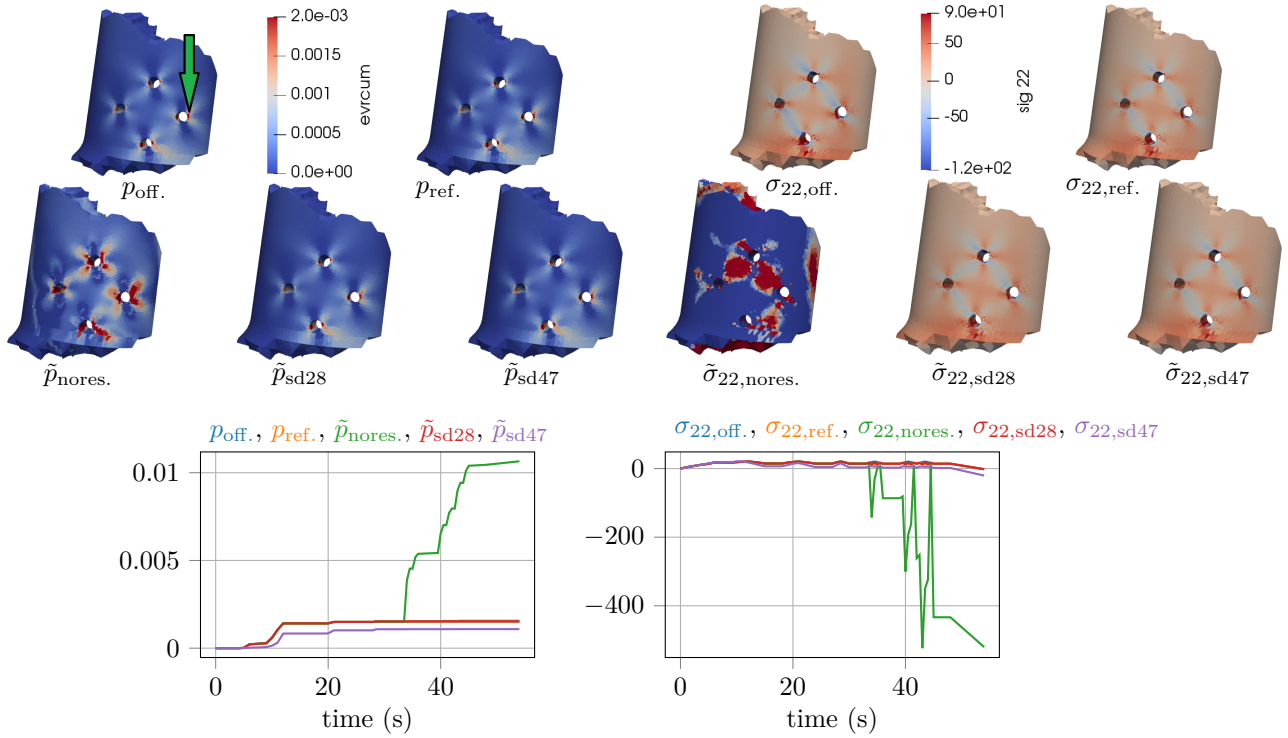


FIGURE 3.29 – Haut : diverses prédiction hautes-fidélités et réduites pour des quantités duales à $t = 43.5s$ pour le sous-domaine 28 : gauche p , droite σ_{22} ; bas : comparaison au point identifié par une flèche verte sur l'image en haut à gauche. Les composantes du tenseur des contraintes sont en MPa.

Les cas tests présentés permettent les observations suivantes :

- [O1] dans la réduction *a posteriori* des calculs élastoviscoplastiques, les variabilités *online* du chargement de température non rencontrées au cours de la phase *offline* peut conduire à des erreurs importantes,
- [O2] le résidu ROM-Gappy-POD (3.109) est fortement corrélé à l'erreur (3.108), de sorte que l'indicateur d'erreur proposé (3.112) peut être utilisé dans l'étape *online* comme décrit dans le workflow de la Figure 3.27 pour corriger les variabilités *online* du chargement en température non rencontrées lors de la phase *offline*.

Nous renvoyons vers la Publication Pub.9 pour plus de détails sur les applications numériques. Nous précisons que dans ce travail, les enrichissements du ROM ont été fait avec la référence haute-fidélité à la variabilité *online* et non en relançant le code haute-fidélité sur le pas de temps correspondant, car les interfaces pour faire ces restarts n'ont pas encore été codées.

3.5.5 Un modèle réduit non-intrusif pour un problème de thermique transitoire non-linéaire avec variations non-paramétrées

Cette section correspond à la Publication Pub.4.

L'application d'intérêt dans cette section est le compresseur haute pression, voir Figure 3. Nous considérons à nouveau le problème de thermique transitoire non-linéaire (3.3a)-(3.3d), dont la forme faible a été décrite dans (3.4), les différences étant que pour l'application industrielle considérée ici, les coefficients matériaux sont uniformes et constants en temps, et des conditions aux limites de convection $\lambda \nabla T \cdot n = h(x, t) (T(x, t) - T_{1,e}(x, t))$ sont présentes en plus des conditions aux limites de radiation. Par ailleurs, la structure est décomposée en 39 flags de surface, sur lesquelles les conditions aux limites de convections varient en temps selon des scénarios complexes issus d'une étape de prétraitement : ces variations, ainsi que la condition initiale, constituent la variabilité non-paramétrée du problème, voir Figure 3.30 et Equation (3.113).

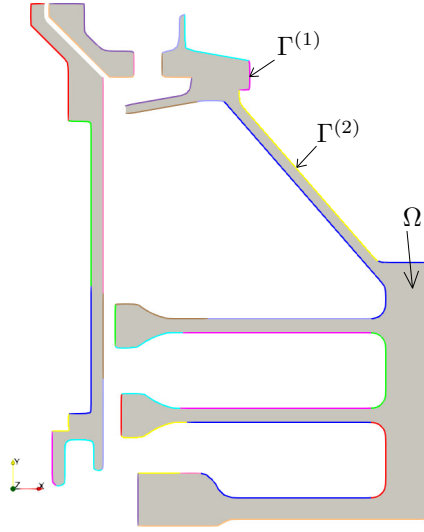


FIGURE 3.30 – Cas test industriel du compresseur haute pression, constitué de 3 corps, avec un partitionnement de la surface $\partial\Omega = \cup_{i=1}^d \Gamma^{(i)}$, $\Gamma^{(i)\circ} \cap \Gamma^{(j)\circ} = \emptyset$, $1 \leq i, j \leq d = 39$.

$$\begin{cases} h(x, t) = h^{(i)}(t), & \Gamma^{(i)}(t) \times [0, t_f] \\ T_{1,e}(x, t) = T_{1,e}^{(i)}(t), & \Gamma^{(i)} \times [0, t_f] \\ (\sigma\epsilon)(x, t) = \sigma\epsilon, & \partial\Omega \times [0, t_f] \\ T_{2,e}(x, t) = T_{2,e}, & \partial\Omega \times [0, t_f] \end{cases} \quad (3.113)$$

Dans ce travail, l'hypperréduction est obtenue par une adaption de l'ECM sur le terme de radiation du vecteur des forces internes réduit :

$$\begin{aligned} \hat{b}_i^{\text{int,rad}} \left(\hat{T}^{(p)}(s+1) \right) &:= -\sigma\epsilon \int_{\partial\Omega} \left[\sum_{k=1}^n \hat{T}_k^{(p)}(s+1) \psi_k(x) \right]^4 \psi_i(x) dx \\ &\approx -\sigma\epsilon \sum_{k'=1}^d \hat{\omega}_{k'} \left[\sum_{j=1}^n \hat{T}_j^{(p)}(s+1) \psi_j(\hat{x}_{k'}) \right]^4 \psi_i(\hat{x}_{k'}), 1 \leq i \leq n, \end{aligned} \quad (3.114)$$

la formule de radiation obtenue étant également utilisée pour le calcul du terme de radiation de l'opérateur tangent global, tandis que l'assemblage des autres termes du modèle réduit peut être précalculé sous la forme de tenseurs d'ordre au plus deux.

Nous implémentons ce modèle dans la librairie non-intrusive présentée dans la Section 3.5.3, et utilisons des snpashots haute-fidélité calculés par Z-set et Abaqus [106]. Nous remarquons qu'avec les choix de modélisation associés à la mise en donnée industrielle, la solution Z-set présente de grandes oscillations non-physiques dans certains coins de la pièce, alors que la solution Abaqus est régulière. Ce dernier code étant commercial, il doit certainement contenir des post-traitements de stabilisation et des gardes fous numériques pour corriger ces oscillations. Nous remarquons que bien que notre modèle réduit résout les mêmes équations que le modèle haute-fidélité Z-set, lorsqu'il est construit avec des snapshots générés par Abaqus, la solution réduit ne contient pas ces oscillations. Ainsi, sans connaître ni avoir implémenté le probable traitement de stabilisation propriétaire présent dans Abaqus, notre modèle réduit hérite directement, à travers la qualité des snapshots, de cette propriété de régularité.

La POD sélectionne 13 modes avec $\epsilon_{\text{POD}} = 10^{-3}$, et ECM sélectionne 39 points d'intégration avec $\epsilon_{\text{ECM}} = 10^{-3}$. Les durées des étapes *offline* et *online* sont données dans la Table 3.7. Dans notre cas, la partie POD est relativement longue (le calcul de la matrice des corrélations prendre 38.2 s à elle seule), car les données d'entraînement contiennent 7 scénarios de 300 snapshots chacun. Le modèle réduit est testé pour une 8^{ème} configuration, différente des scénarios d'entraînement. Le modèle haute-fidélité prend 52 s avec Abaqus, conduisant à un speedup de 91 dans ce cas.

étape <i>offline</i>		étape <i>online</i>	
POD	41.1 s	calcul du chargement <i>online</i>	0.09 s
ECM	7.7 s	ROM	0.45 s
Reste	17.3 s	Ecriture de la solution dans le format ROM	0.03 s
Total	66.1 s	Total	0.57 s

TABLE 3.7 – Durée des étapes *offline* et *online*

La comparaison de la Figure 3.31 montre la prédiction du ROM pour la variabilité du 8^{ème} scénario. Les points B et C sont choisis pour illustrer les plus grands écarts, et une erreur de -11°C a été mesurée à $t = 24$ s, ce qui correspond à une erreur relative de 2.4% à ce moment du scénario.

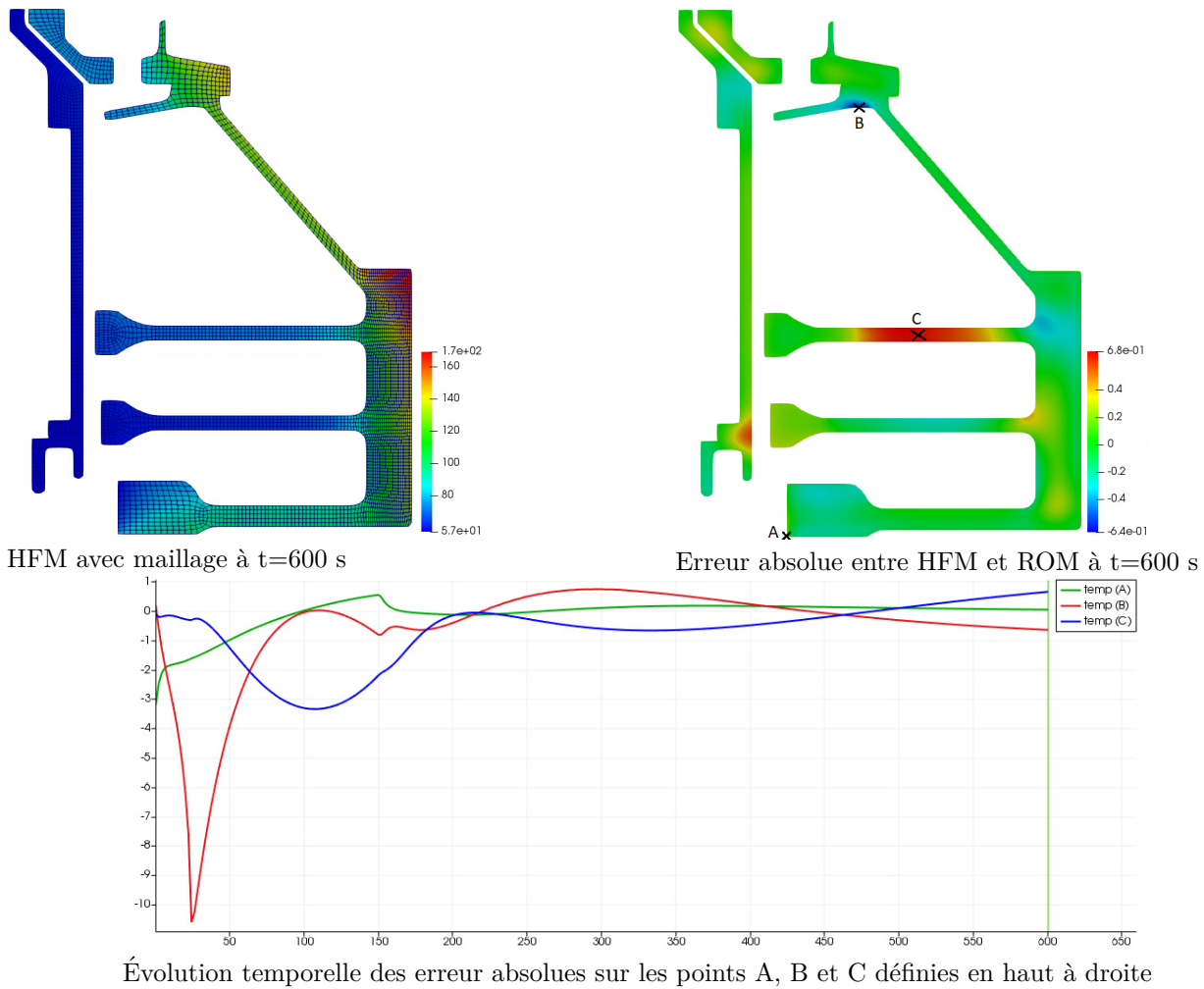


FIGURE 3.31 – Application industrielle

Nous menons maintenant une analyse d'erreur sur ce cas, pour mesurer la précision de la prédiction réduite en fonction de l'hyperparameter ϵ_{POD} . Nous définissons l'erreur relative temporelle comme $\mathcal{E}(t) := \frac{\|\hat{T}(\cdot, t) - T(\cdot, t)\|_{L^2(\Omega)}}{\|T(\cdot, t)\|_{L^2(\Omega)}}$, où T et \hat{T} sont respectivement les prédictions du HFM et du ROM à la variabilité du 8^{ème} scénario. L'erreur relative moyenne est définie par $\bar{\mathcal{E}} := \frac{1}{J} \sum_{s=1}^J \mathcal{E}(sdt)$. Les performances du ROM pour plusieurs valeurs de ϵ_{POD} sont évaluées dans la Table 3.8 et la Figure 3.32, où $\epsilon_{\text{ECM}} = 10^{-3}$. Dans nos expériences, la valeur de ϵ_{ECM} n'a pas d'impact significatif (pour les valeurs 10^{-2} , 10^{-3} et 10^{-4}), les raisons pouvant être

que l'erreur est dominée par la troncature POD et que la non-linéarité polynomiale dans notre cas est déjà bien approchée avec une tolérance moyenne. Dans la Section 3.5.3, où nous avons considéré un problème de mécanique non-linéaire avec matériaux élastoviscoplastiques, la tolérance ϵ_{ECM} avait un impact significatif sur la qualité de la prédiction réduite. Nous remarquons également que le speedup est intéressant pour les valeurs de ϵ_{POD} plus grandes que 10^{-3} et que la précision de la prédiction n'augmente plus pour des valeurs inférieures à $\epsilon_{\text{POD}} = 10^{-4}$: avec un ROM construit à partir de snapshots générés par Abaqus, nous avons vu que nous ne résolvons pas exactement le même modèle que celui présent dans Abaqus.

ϵ_{POD}	nbe modes POD	nbe points ECM	durée <i>offline</i>	speedup	$\bar{\mathcal{E}}$
10^{-5}	62	114	335 s	0.2	7.6×10^{-4}
10^{-4}	31	78	110 s	2	8.4×10^{-4}
10^{-3}	13	39	66 s	91	2.5×10^{-3}
10^{-2}	4	19	57 s	96	2.3×10^{-2}

TABLE 3.8 – Comparaison des performances du ROM pour plusieurs valeurs de ϵ_{POD} .

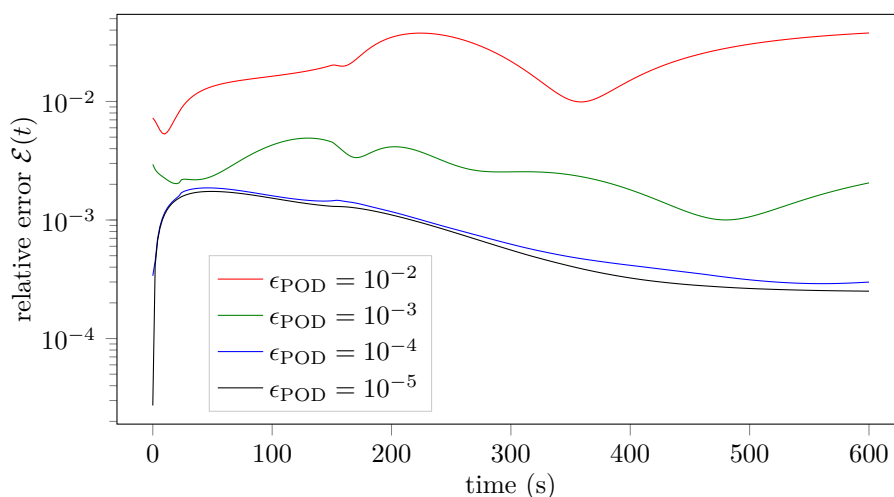


FIGURE 3.32 – Évolution temporelle de l'erreur relative $\mathcal{E}(t)$ pour plusieurs valeurs de ϵ_{POD} .

Nous renvoyons vers la Publication [Pub.4](#) pour le reste des applications numériques.

3.6 POD-Galerkine stable pour l'équation de Navier-Stokes incompressible en régime turbulent avec variations géométriques non-paramétrées.

Cette section correspond aux Publications [Pub.1](#) et [Pub.8](#), pour lesquelles Nissrine Akkari est la contributrice principale : elle sera, pour cette raison, très synthétique.

Modèle réduit stable

La réduction de modèle de l'équation de Navier-Stokes incompressible en régime turbulent est un sujet de recherche très actif. De tels écoulements contiennent plusieurs échelles énergétiques, et même les petites échelles ont une contribution importante dans le profil des champs de vitesse. La POD classique, qui tronque les petites échelles, échoue alors à réduire le modèle de façon satisfaisante : soit la précision est mauvaise, soit le nombre de modes à garder est tel qu'aucun speedup ne sera mesuré en pratique. Cette question a reçu un intérêt important dans la littérature, et les solutions existantes consistent à stabiliser le modèle réduit en ajoutant des termes de dissipation artificiels, à changer le produit scalaire servant à calculer les corrélations dans le calcul de la

snapshot POD (voir Algorithme 3) ou à proposer une nouvelle façon de construire une base pour donner de l'importance aux petites échelles énergétiques.

Dans la Publication [Pub.8](#), nous proposons une façon originale de construire une base POD où les petites échelles énergétiques sont bien représentées. En plus d'un snapshot POD sur les champs de vitesse, une deuxième snapshot POD sur les gradients des champs de vitesse est aussi réalisée. Dans ce deuxième cas, nous gardons seulement les valeurs propres et vecteurs propres obtenus, et les recombinaisons directement avec les snapshots de vitesse (au lieu des gradients). Les modes ainsi obtenus sont orthonormalisés avec la base POD classique sur la vitesse. Avec cette base, les ROM-Galerkine obtenus sont parfaitement stables, sans qu'aucune stabilisation numérique ou artificielle ne soit nécessaire, sur un cas d'injecteur industriel de moteur d'avion à 14 millions de cellules, et en extrapolation temporelle d'un facteur 10, avec un speedup de l'ordre du million. Notons que la pression est imposée sur la surface d'entrée par une méthode de pénalisation. Une autre originalité de ce travail est que les schémas numériques des modèle réduit et haute-fidélité sont différents : Galerkine sans modélisation de la turbulence et schéma implicite en temps pour le modèle réduit, contre volumes finis avec turbulence modélisée par Large Eddy Simulation (LES) et schéma explicite en temps pour le modèle haute-fidélité.

Variations géométrique non-paramétrées

Supposons que nous disposons d'un calcul LES pour une géométrie d'injecteur aéronautique. Nous souhaitons calculer rapidement l'écoulement dans l'injecteur pour des designs quelconques de la zone centrale de l'injecteur, de sorte qu'une quantité d'intérêt métier soit optimisée (stabilité et forme de la flamme, minimisation de la consommation et des polluants). Les variabilités géométriques sont difficiles à considérer dans un contexte de réduction de modèle, car le support des snapshots change avec la variabilité, et le calcul des corrélations entre les snapshots n'est alors plus défini de façon directe. De nombreuses contributions existent dans le cas des variations paramétriques, mais les variations géométriques non-paramétriques ont été bien moins traitées dans la littérature. La Gappy-POD, présentée dans l'Algorithme 5, a été utilisée pour faire de l'optimisation de design paramétré.

Dans la Publication [Pub.1](#), une stratégie à base de Gappy-POD est développée pour prendre en compte des variabilités non-paramétrées du design. La géométrie étant susceptible de changer seulement dans une zone limitée de l'injecteur (appelée masque), une première étape consiste à calculer le champ de vitesse localement dans cette zone, pour la nouvelle géométrie. Ceci est fait de façon efficace en considérant un maillage très déraffiné à l'extérieur du masque, où le champ de vitesse associé à la géométrie initiale est imposé : seule la vitesse dans le masque est calculée. Le champ de vitesse obtenu est projeté sur le maillage initial, et la Gappy-POD classique est utilisée pour mettre à jour la vitesse à l'extérieur du masque (les modes Gappy-POD sont calculés en utilisant la POD stabilisée présentée dans la section précédente). Une POD est calculée sur les nouveaux champs de vitesse ainsi obtenus, et un ROM-Galerkine est calculé sur ces modes. Enfin, les coefficients ainsi calculés sont combinés avec la base POD stabilisée associée au calcul LES sur la géométrie de référence, pour former la prédiction à l'extérieur du masque. Cette procédure complète a un speedup de 100 sur un injecteur industriel et sa précision est meilleure que la Gappy-POD classique sur des indicateurs associés aux zones de recirculation.

Chapitre 4

Perspectives

Nous proposons des perspectives de recherche en réduction de complexité des algorithmes pour la simulation numérique des équations de la physique, qui permettront d’adresser certains besoins futurs du domaine aéronautique.

4.1 Amélioration des méthodes de réduction d’ordre de modèle

Dans la Section 3.5.4, nous avons introduit un indicateur d’erreur adapté aux quantités duales, comme l’endommagement, pour la mécanique des structures non-linéaire. Nous avons illustré que lorsqu’on utilise cet indicateur pour enrichir un modèle réduit préalablement construit par la solution exacte, la précision du modèle réduit peut être grandement améliorée avec un nombre d’enrichissements limité, ce qui rend la procédure rentable. Cependant, dans une situation d’exploitation du modèle réduit, nous ne disposons pas de la solution exacte, mais il est possible d’appeler le modèle haute-fidélité pour calculer le snapshot au pas de temps considéré : ceci nécessite néanmoins de pouvoir reconstruire l’état matériau complet sur le maillage haute-fidélité à ce pas de temps. Sachant que les modèles industriels d’alliages monocristallins peuvent contenir jusqu’à une centaines de variables internes, cette reconstruction limite la performance de la réduction de modèle. Une piste consiste à essayer de minimiser l’indication d’erreur du modèle réduit en cherchant une direction de descente dans l’espace haute-fidélité pour enrichir la base d’ordre réduit.

Dans la Section 3.2, nous avons illustré que l’Empirical Interpolation Method (EIM) appliquée à la réduction de modèle non-linéaire peut devenir instable, ce constat étant partagé par la communauté. Notre interprétation est que l’EIM est sujette à un sur-apprentissage. Nous souhaitons poursuivre le recours aux méthodes de régularisation issues de la communauté statistiques pour améliorer la situation et peut-être permettre une utilisation plus simple d’EIM pour réduire des problèmes non-linéaires. A partir de cette interprétation, nous avons remarqué que l’instabilité se produit lorsque que l’amplitude des coefficients de la décomposition EIM s’emballe. A l’instar de l’indicateur d’erreur de la Section 3.5.4, il pourrait être également possible de proposer un indicateur de stabilité ou d’erreur qui serait appris à partir des coefficients de la décomposition EIM directement.

La collaboration avec Nissrine Akkari sur la réduction des équations de Navier-Stokes incompressibles en régime turbulent Pub.8, notamment avec variation géométriques Pub.1 sera poursuivie. Dans la construction du modèle réduit, nous avons remarqué d’importantes améliorations des résultats sur un critère a priori empirique, lié à l’ordre de l’affectation des coefficients aux modes POD. Nous souhaitons continuer ces travaux pour comprendre les phénomènes sous-jacents et éventuellement continuer à faire progresser l’approche.

4.2 Implémentation générique et non-intrusive de la réduction d’ordre de modèle

Une librairie de réduction de modèle non-intrusive pour la mécanique des structures Pub.7 et la thermique non-linéaires Pub.4 a été développée. Le modèle réduit physique est construit à partir de résultats de codes de calculs commerciaux. L’originalité de cette approche donne l’avantage d’un potentiel important de contributions scientifiques. Comme nous pouvons construire un modèle réduit uniquement à partir de solutions d’équations (et donc sans code de calcul associé), nous pouvons considérer des champs obtenus par l’expérimentation (assimilation de données), de résolutions différentes (multi-fidélité) ou même de très grande taille (incalculables par

éléments finis, donc donnés soit analytiquement, soit encore par l'expérimentation, soit sous la forme de première prédiction locale par un réseau de neurones). Une application industrielle à moyen terme serait la construction d'un modèle réduit pour la dynamique des fluides à partir de mesures PIV (particle image velocimetry) dans une chambre de combustion.

Avec une bonne architecture de code, il apparaît possible de standardiser la plupart des opérations réalisées dans le cadre de la réduction de modèle physique pour pouvoir les appliquer à de nombreuses physiques et cas d'étude. Une première implémentation en ce sens a été initiée, dans une librairie où nous espérons pouvoir cumuler non-intrusivité, calcul parallèle à mémoire distribuée, multi-physique (mécanique solide et fluide, thermique) avec non-linéarités génériques et variabilités non-paramétriques (notamment géométriques) et, autant que possible, indication d'erreur. Même si certains éléments de réponse ont déjà été proposés dans [Pub.7](#), et que d'autres sont disponibles dans la littérature, les challenges scientifiques et techniques restant sont importants. En particulier, l'indication d'erreur et les variations géométriques non-paramétriques pour toutes les physiques restent difficiles.

4.3 Intelligence artificielle au service de la réduction d'ordre de modèle

L'utilisation des réseaux de neurones profond pour la prédiction de résultats de simulation est potentiellement difficile et pour l'instant non certifiable ni interprétable. Cependant, leur potentiel pour assister certaines étapes de la réduction du modèle est très important, la prédiction physique étant toujours confiée à un modèle réduit résolvant les équations de la physique.

Des travaux ont commencé, dans le cadre de la thèse de Thomas Daniel [PhD.2](#) et de la Publication [Pub.3](#), sur la construction d'un dictionnaire de modèles hyperréduits pouvant prendre en compte une grande variabilité non-paramétrisable du chargement thermique d'une pièce mécanique, telle que le problème n'est pas réductible avec un seul modèle réduit. En particulier dans [Pub.3](#) et la thèse de Thomas Daniel, la recommandation du modèle réduit local n'est pas faite à partir de données du problème, comme ce qui est fait actuellement dans l'état de l'art, mais dans une approche originale utilisant les réseaux de neurones profonds. Après un grand nombre de calculs candidats et un clustering sur les distances de Grassman, un réseau de neurones profond est entraîné, dans une tâche de classification, à reconnaître le numéro du cluster à partir de la donnée du chargement thermique. Ainsi, le bon modèle réduit est appelé en phase d'exploitation sans avoir à évaluer la prédiction mécanique. Dans l'état de l'art actuel, la recommandation du modèle réduit est faite sur les données du problème, alors que dans nos travaux, elle est spécialement adaptée à l'exercice : ici la réduction de modèle. En effet, les distances de Grassman quantifient la proximité des sous-espaces vectoriels générés par les bases d'ordre réduit. Les premiers résultats montrent un net avantage de cette approche par rapport à des approches plus naïves pour construire le dictionnaire de modèles réduits. Dans ces travaux, nous avons formalisé la notion de ROM-net, une procédure contenant un réseau de neurones pour assister une ou plusieurs étapes de la réduction de modèle qui sont difficilement traitées dans l'état de l'art actuel.

Un auto-encodeur peut être imaginé pour faire de la prédiction temporelle, qui serait à nouveau corrigée par un modèle réduit. Ces technologies permettent d'envisager des variabilités géométriques compliquées ou des phénomènes de transport mal réductibles, avec une mise en œuvre relativement simple. L'application de ces idées est prévue dans le cadre du projet de thèse [PhD.1](#), sur la réduction de modèle en dynamique rapide. L'utilisation d'un Generative Adversarial Network (GAN) pour générer des simulations fluides 2D incompressible et turbulentes est illustrée dans [Pub.6](#), et reprennent des éléments commencés dans le stage [M2R.1](#). Enfin, dans le travail récent [Pub.5](#), nous proposons un ROM-net permettant de passer d'une résolution basse fidélité à son erreur de discrétisation ; cette prédiction est ensuite corrigée par un modèle réduit pour lui faire respecter les conditions aux limites et l'équilibre global. En pratique, de nombreuses géométries sont générées et des problèmes de thermique transitoire sont résolus sur des maillages fins et grossiers associées aux mêmes géométries. Les solutions grossières et la différence entre solution fine et grossière (sur une fenêtre carrée) sont projetées sur une grille régulière fine et le réseau de neurones apprend la projection de l'écart à partir de la projection de la solution grossière. Les procédures d'entraînement et d'utilisation de ce réseau sont illustrées dans la Figure 4.1. Des premiers résultats sont illustrés sur la Figure 4.2 sur des échantillons de validation. Les modèles réduits obtenus à partir des snapshots des solutions grossières et des prédictions de l'erreur de discrétisation par l'application du réseau de neurones sur ces solutions grossières ont conduit à des prédictions plus précises, sur des indicateurs physiques, pour toutes les configurations testées, alors que le réseau de neurone empire parfois la prédiction grossière. Des améliorations d'architecture et de fonctions loss adaptées au problème sont envisagées.

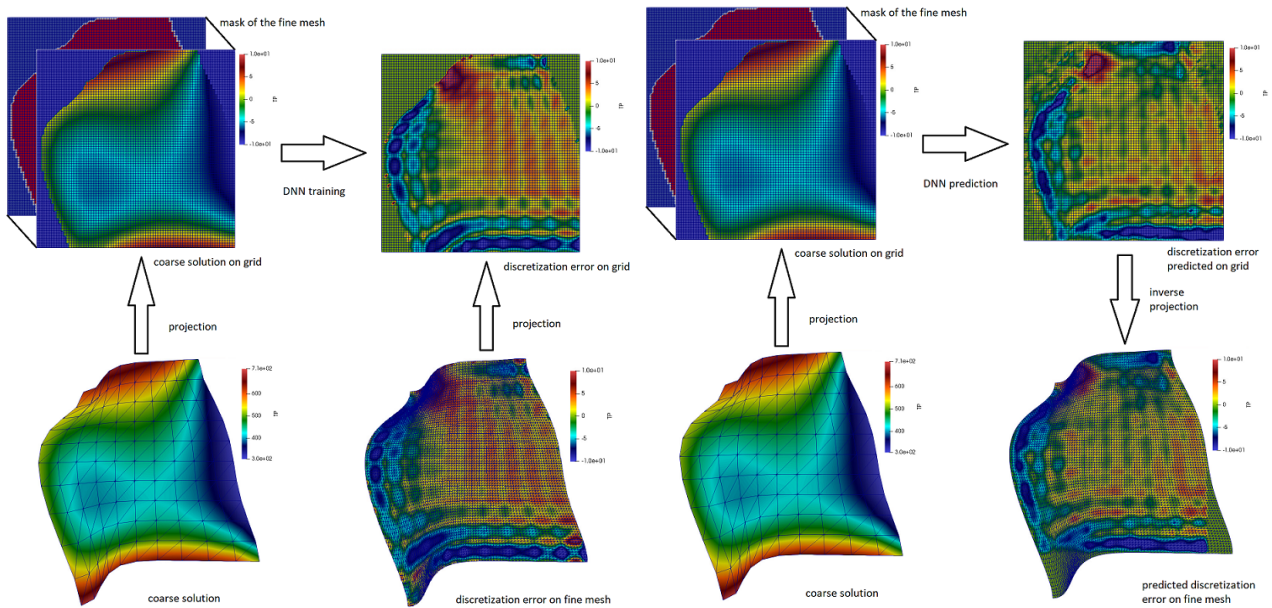


FIGURE 4.1 – Réseau de neurones convolutifs; gauche : procédure d’entraînement, droite : procédure d’utilisation.

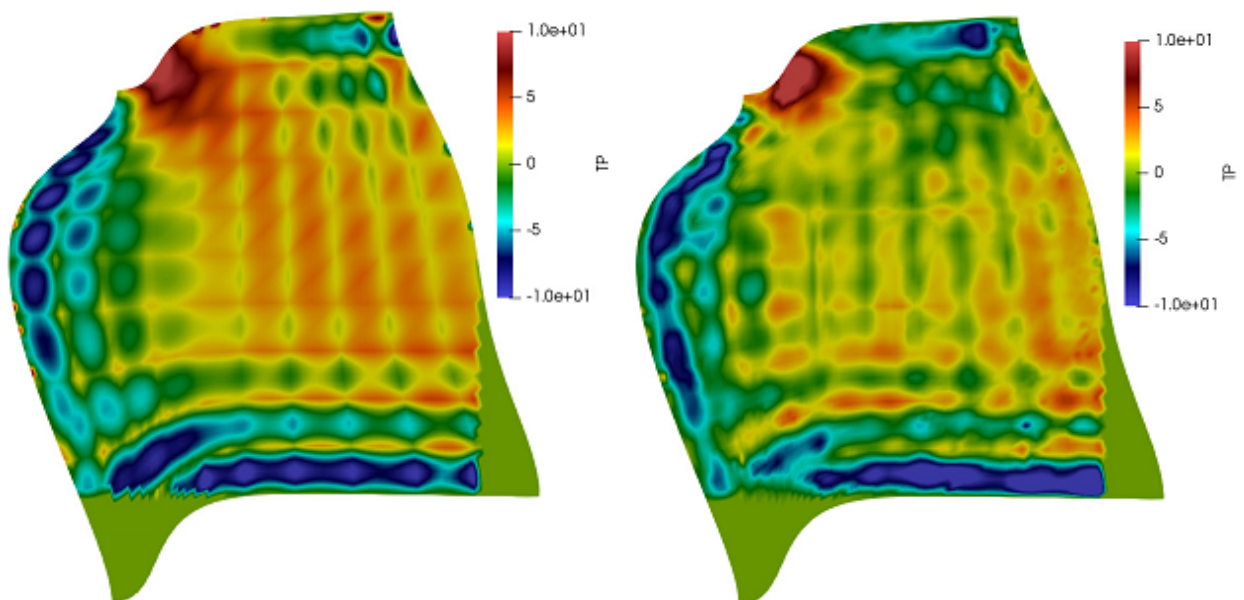


FIGURE 4.2 – Résultat de l’apprentissage de l’erreur de discrétisation d’un problème thermique par réseau de neurones convolutifs sur un échantillon de validation; gauche : prédiction de l’erreur de discrétisation à partir de la solution grossière, droite : erreur de discrétisation de référence.

Bibliographie

- [1] GOCE mission. <https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/goce>.
- [2] GRACE mission. https://www.nasa.gov/mission_pages/Grace/index.html.
- [3] File :gaturbineblade.svg. Wikipedia, the free encyclopedia, image under the Creative Commons Attribution-Share Alike 3.0 Unported license, 2009.
- [4] File :turbinenschaufel rb199.jpg. Wikipedia, the free encyclopedia, image under the Creative Commons Attribution-Share Alike 3.0 Unported license, 2009.
- [5] J. Ahrens, B. Geveci, and C. Law. Paraview : An end-user tool for large data visualization, visualization handbook. Elsevier, 2005.
- [6] N. Akkari, A. Hamdouni, E. Liberge, and M. Jazar. On the sensitivity of the pod technique for a parameterized quasi-nonlinear parabolic equation. *Advanced Modeling and Simulation in Engineering Sciences*, 1(1) :1–14, Aug 2014.
- [7] F. Alted, I. Vilata, et al. PyTables : Hierarchical datasets in Python, 2002.
- [8] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2) :237–260, 1998.
- [9] C. Amante and B.W. Eakins. Etopo1 1 arc-minute global relief model : Procedures, data sources and analysis. *NOAA Technical Memorandum NESDIS NGDC-24. National Geophysical Data Center*, 2009.
- [10] S. Amaral, T. Verstraete, R. Van den Braembussche, and T. Arts. Design and optimization of the internal cooling channels of a high pressure turbine blade—part i : Methodology. *Journal of Turbomachinery*, 132, 2010.
- [11] D. Amsellem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10) :891–916, 2012.
- [12] S. S. An, T. Kim, and D. L. James. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)*, 27(5) :165, 2008.
- [13] C. V. Ananth and D. G. Kleinbaum. Regression models for ordinal responses : a review of methods and applications. *International Journal of Epidemiology*, 26(6) :1323–1333, 1997.
- [14] J.P. Argaud, B. Bouriquet, H. Gong, Y. Maday, and O. Mula. Stabilization of (g) eim in presence of measurement noise : application to nuclear reactor physics. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 133–145. Springer, 2017.
- [15] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10) :2237–2251, Nov 2008.
- [16] C. Audouze, F. De Vuyst, and P. B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5) :1587–1628, 2013.
- [17] M. Barrault, Y. Maday, N.-C. Nguyen, and A. T. Patera. An 'empirical interpolation' method : application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9) :667 – 672, 2004.
- [18] J. Baumgardner and P. Frederickson. Icosahedral discretization of the two-sphere. *SIAM Journal on Numerical Analysis*, 22(6) :1107–1115, 1985.
- [19] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4) :565–589, 2000.
- [20] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1) :1–24, 2003.
- [21] M. Bebendorf and R. Venn. Constructing nested bases approximations from the entries of non-local operators. *Numerische Mathematik*, 121(4) :609–635, 2012.
- [22] W. Beyn and V. Thümmler. Freezing solutions of equivariant evolution equations. *SIAM Journal on Applied Dynamical Systems*, 3(2) :85–116, 2004.
- [23] B. K. Bhattacharyya. Some general properties of potential fields in space and frequency domain : A review. *Geoexploration*, 5(3) :127 – 143, 1967.
- [24] R. J. Blakely. *Potential theory in gravity and magnetic applications*. Cambridge University Press, Cambridge (UK), 1995.
- [25] C. Bonacina, G. Comini, A. Fasano, and M. Primicerio. Numerical solution of phase-change problems. *International Journal of Heat and Mass Transfer*, 16(10) :1825 – 1832, 1973.
- [26] D. Bonomi, A. Manzoni, and A. Quarteroni. A matrix discrete empirical interpolation method for the efficient model reduction of parametrized nonlinear pdes : application to nonlinear elasticity problems. *MATHICSE, École Polytechn. Fédérale de Lausanne, Lausanne, Switzerland, Tech. Rep*, 14 :2016, 2016.
- [27] C. Bovet, A. Parret-Fréaud, N. Spillane, and P. Gosselet. Adaptive multipreconditioned feti : Scalability results and robustness assessment. *Computers & Structures*, 193 :1 – 20, 2017.

-
- [28] A. Buhr, C. Engwer, M. Ohlberger, and Rave S. A numerically stable a posteriori error estimator for reduced basis approximations of elliptic equations. *11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014*, pages 4094–4102, 2014.
- [29] N. Cagniard, Y. Maday, and B. Stamm. *Model Order Reduction for Problems with Large Convection Effects*, pages 131–150. Springer International Publishing, Cham, 2019.
- [30] K. Carlberg, F. Charbel, J. Cortial, and D. Amsallem. The gnat method for nonlinear model reduction : Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242 :623 – 647, 2013.
- [31] P. Caron and O. Lavigne. Recent studies at onera on superalloys for single crystal turbine blades. *AerospaceLab*, pages 1–14, 2011.
- [32] R. Chakir and Y. Maday. Une méthode combinée d’éléments finis à deux grilles/bases réduites pour l’approximation des solutions d’une e.d.p. paramétrique. *Comptes Rendus Mathématique*, 347(7) :435 – 440, 2009.
- [33] L. Chamoin, F. Pled, P.-E. Allier, and P. Ladevèze. A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 327 :118–146, 2017.
- [34] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7) :808–817, 2000.
- [35] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5) :2737–2764, 2010.
- [36] Y. Chen, J. Jiang, and A. Narayan. A robust error estimator and a residual-free error indicator for reduced basis methods. *Computers & Mathematics with Applications*, 2018.
- [37] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings. An overview of the Proper Generalized Decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11) :578 – 592, 2011. XVIth International Workshop on Numerical Methods for Non-Newtonian Flows.
- [38] F. Chinesta, P. Ladeveze, and C. Elías. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18 :395–404, 2011.
- [39] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. González, I. Alfaro, A. Ammar, and A. Huerta. Pgd-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1) :31–59, 2013.
- [40] O. Coulaud, B. Bramas, and C. Piacibello. Scalfmm, C++ Fast Multipole Method Library for HPC. <http://scalfmm-public.gforge.inria.fr/doc/>.
- [41] B. A. Cowles. High cycle fatigue in aircraft gas turbines—an industry perspective. *International Journal of Fracture*, 80(2-3) :147–163, 1996.
- [42] credit : ESA. Geoid. <https://phys.org/news/2012-10-atomic-clocks-good-earth-geoid.html>.
- [43] L. Dalcín, R. Paz, M. Storti, and Jo. D’Elía. Mpi for python : Performance improvements and mpi-2 extensions. *Journal of Parallel and Distributed Computing*, 68(5) :655–662, 2008.
- [44] A. Dasgupta, Y. V. Sun, I. R. König, J. E. Bailey-Wilson, and J. D. Malley. Brief review of regression-based and machine learning methods in genetic epidemiology : the genetic analysis workshop 17 experience. *Genetic Epidemiology*, 35(S1) :S5–S11, 2011.
- [45] A. Dutt, M. Gu, and V. Rokhlin. Fast algorithms for polynomial interpolation, integration, and differentiation. *SIAM Journal on Numerical Analysis*, 33(5) :1689–1711, 1996.
- [46] A. Dutt and V. Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing*, 14(6) :1368–1393, 1993.
- [47] G. L. Erickson. A new, third-generation, single-crystal, casting superalloy. *JOM*, 47(4) :36–39, Apr 1995.
- [48] R. Everson and L. Sirovich. Karhunen–loève procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8) :1657–1664, Aug 1995.
- [49] M. Fares, J.S. Hesthaven, Y. Maday, and B. Stamm. The reduced basis method for the electric field integral equation. *Journal of Computational Physics*, 230(14) :5532 – 5555, 2011.
- [50] C. Farhat, P. Avery, T. Chapman, and J. Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9) :625–662, 2014.
- [51] C. Farhat, T. Chapman, and P. Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *International Journal for Numerical Methods in Engineering*, 102(5) :1077–1110, 2015.
- [52] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6) :1205–1227.
- [53] W. Fong and E. Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228(23) :8712 – 8725, 2009.
- [54] K. Giebermann. Multilevel approximation of boundary integral operators. *Computing*, 67(3) :183–207, 2001.
- [55] L. Giraldi, A. Litvinenko, D. Liu, H. Matthies, and A. Nouy. To be or not to be intrusive? the solution of parametric and stochastic equations—the “plain vanilla” galerkin case. *SIAM Journal on Scientific Computing*, 36(6) :A2720–A2744, 2014.
- [56] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1–3) :1 – 21, 1997.
- [57] K.M. Gorski, Eric Hivon, A.J. Banday, B.D. Wandelt, F.K. Hansen, et al. HEALPix - A Framework for high resolution discretization, and fast analysis of data distributed on the sphere. *Astrophys.J.*, 622 :759–771, 2005.

-
- [58] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13(4) :515, Dec 2006.
- [59] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2) :325–348, 1987.
- [60] L. Greengard and J. Strain. The Fast Gauss Transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1) :79–94, 1991.
- [61] T. Grombein, K. Seitz, and B. Heck. Optimized formulas for the gravitational field of a tesseroid. *Journal of Geodesy*, 87(7) :645–660, 2013.
- [62] W. Hackbusch and S. Börm. H2-matrix approximation of integral operators by interpolation. *Applied Numerical Mathematics*, 43(1–2) :129 – 143, 2002. 19th Dundee Biennial Conference on Numerical Analysis.
- [63] T. Hartmann and H.-G. Wenzel. The hw95 tidal potential catalogue. *Geophysical Research Letters*, 22(24) :3553–3556.
- [64] F. Hecht, J. Morice, and O. Pironneau. freeFEM++, www.freefem.org/.
- [65] T. Henneron, H. Mac, and S. Clenet. Error estimation of a proper orthogonal decomposition reduced model of a permanent magnet synchronous machine. In *Computation in Electromagnetics (CEM 2014), 9th IET International Conference on*, pages 1–6, London, United Kingdom, 2014. IEEE.
- [66] J. A. Hernandez, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer methods in applied mechanics and engineering*, 313 :687–722, 2017.
- [67] J. Hesthaven, B. Stamm, and S. Zhang. Certified reduced basis method for the electric field integral equation. *SIAM Journal on Scientific Computing*, 34(3) :A1777–A1799, 2012.
- [68] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363 :55 – 78, 2018.
- [69] R. Hiptmair and P. Meury. Stabilized FEM-BEM coupling for Helmholtz transmission problems. *SIAM J. Numer. Anal.*, 44(5) :2107–2130, 2006.
- [70] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2) :022923, 2014.
- [71] E. Kammann, F. Tröltzsch, and S. Volkwein. A method of a-posteriori error estimation with application to proper orthogonal decomposition. 2012.
- [72] R. Kouhia. On the solution of non-linear diffusion equation. *Rakenteiden mekaniikka*, 46(4) :116–130, 2013.
- [73] A. Kounitzky, J. Wortmann, and P.N. Agarwal. A single crystal casting process for high-temperature components. *Materials & Design*, 12(6) :323 – 330, 1991.
- [74] P. Ladevèze and L. Chamoin. Toward guaranteed PGD-reduced models. *Bytes and Science. CIMNE : Barcelona*, pages 143–154, 2013.
- [75] P. Ladevèze and A. Chouaki. Application of a posteriori error estimation for structural model updating. *Inverse problems*, 15(1) :49, 1999.
- [76] V. Levillain. *Couplage éléments finis-équations intégrales pour la résolution des équations de Maxwell en milieu hétérogène*. PhD thesis, Ecole Polytechnique, 1991.
- [77] X. Li and M. Chouteau. Three-dimensional gravity modeling in all space. *Surveys in Geophysics*, 19(4) :339–368, 1998.
- [78] G. Luo. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1) :18, 2016.
- [79] Z. Luo, J. Zhu, R. Wang, and I. M. Navon. Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical pacific ocean reduced gravity model. *Computer Methods in Applied Mechanics and Engineering*, 196(41) :4184 – 4195, 2007.
- [80] L. Machiels, Y. Maday, A. T. Patera, C. Prud’homme, D. V. Rovas, G. Turinici, and K. Veroy. Reliable real-time solution of parametrized partial differential equations : Reduced-basis output bound methods. *CJ Fluids Engineering*, 124 :70–80, 2002.
- [81] Y. Maday and O. Mula. *A Generalized Empirical Interpolation Method : Application of Reduced Basis Techniques to Data Assimilation*, pages 221–235. Springer Milan, Milano, 2013.
- [82] Y. Maday, O. Mula, and G. Turinici. Convergence analysis of the Generalized Empirical Interpolation Method. *SIAM Journal on Numerical Analysis*, 54(3) :1713–1731, 2016.
- [83] Y. Maday, N. C. Nguyen, A. T. Patera, and S. Pau. A general multipurpose interpolation procedure : the magic points. *Communications On Pure And Applied Analysis*, 8(1) :383–404, 2008.
- [84] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12) :3397–3415, Dec 1993.
- [85] R. J. Martin. Approximations to the determinant term in Gaussian maximum likelihood estimation of some spatial models. *Communications in Statistics - Theory and Methods*, 22(1) :189–205, 1992.
- [86] P. G. Martinsson and V. Rokhlin. An accelerated kernel-independent fast multipole method in one dimension. *SIAM J. Sci. Comput.*, 26 :1389–1404, 2005.
- [87] Z. Mazur, A. Luna-Ramírez, J.A. Juárez-Islas, and A. Campos-Amezcuca. Failure analysis of a gas turbine blade made of inconel 738lc alloy. *Engineering Failure Analysis*, 12(3) :474 – 486, 2005.
- [88] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim. Robust regression methods for computer vision : A review. *International Journal of Computer Vision*, 6(1) :59–70, 1991.
- [89] M. Messner, B. Bramas, D. Coulaud, and E. Darve. Optimized M2L Kernels for the Chebyshev Interpolation based Fast Multipole Method. *CoRR*, abs/1210.7292, 2012.
- [90] Mines ParisTech and ONERA the French aerospace lab. Zset : nonlinear material & structure analysis suite. <http://www.zset-software.com/>, 1981-present.

-
- [91] M. Muhieddine, E. Canot, and R. March. Various approaches for solving problems in heat conduction with phase change. *International Journal on Finite Volumes*, 6(1) :19, 2009.
- [92] N. C. Nguyen, A. T. Patera, and J. Peraire. A ‘best points’ interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4) :521–543.
- [93] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathématique*, 351(23) :901 – 906, 2013.
- [94] M. Okabe. Analytical expressions for gravity anomalies due to homogeneous polyhedral bodies and translations into magnetic anomalies. *Geophysics*, 44 :730, 1979.
- [95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.
- [96] R. Peharz and F. Pernkopf. Sparse nonnegative matrix factorization with l0-constraints. *Neurocomputing*, 80 :38 – 46, 2012. Special Issue on Machine Learning for Signal Processing 2010.
- [97] B. Peherstorfer, Z. Drmač, and S. Gugercin. Stabilizing discrete empirical interpolation via randomized and deterministic oversampling. *arXiv preprint arXiv :1808.10473*, 2018.
- [98] C. Ronchi, R. Iacono, and P.S. Paolucci. The cubed sphere : A new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics*, 124(1) :93 – 114, 1996.
- [99] V. Roshan Joseph, E. Gul, and S. Ba. Maximum projection designs for computer experiments. *Biometrika*, (102) :371–380, 2015.
- [100] D. Ryckelynck. A priori hyperreduction method : an adaptive approach. *Journal of Computational Physics*, 202(1) :346–366, 2005.
- [101] D. Ryckelynck. Estimation d’erreur d’hyperréduction de problèmes élastoviscoplastiques. *21ème Congrès Français de Mécanique, 2013, Bordeaux, France*, 2013.
- [102] D. Ryckelynck, L. Gallimard, and S. Jules. Estimation of the validity domain of hyper-reduction approximations in generalized standard elastoviscoplasticity. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1) :19 p., 2015.
- [103] M. Schmidt, M. Fengler, T. Mayer-Gürr, A. Eicker, J. Kusche, L. Sánchez, and S. C. Han. Regional gravity modeling in terms of spherical base functions. *Journal of Geodesy*, 81(1) :17–38, 2007.
- [104] U. Schulz, C. Leyens, K. Fritscher, M. Peters, B. Saruhan, O. Lavigne, J.-M. Dorvaux, M. Poulain, R. Mévrel, and M. Caliez. Some recent trends in research and technology of advanced thermal barrier coatings. 7 :73–80, 2003.
- [105] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I, II and III. *Quarterly of Applied Mathematics*, XLV :561–590, 1987.
- [106] M. Smith. *ABAQUS/Standard User’s Manual, Version 6.9*. Simulia, 2009.
- [107] A. Toumi and F. De Vuyst. Empirical interpolation decomposition. *Acta Applicandae Mathematicae*, Nov 2018.
- [108] F. Tröltzsch and S. Volkwein. Pod a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44(1) :83, 2009.
- [109] I. Uysal and H. A. Güvenir. An overview of regression techniques for knowledge discovery. *Knowl. Eng. Rev.*, 14(4) :319–340, December 1999.
- [110] M. Van Camp, M. Vanclooster, O. Crommen, T. Petermans, K. Verbeeck, B. Meurers, T. van Dam, and A. Dassargues. Hydrogeological investigations at the membach station, belgium, and application to correct long periodic gravity variations. *Journal of Geophysical Research : Solid Earth*, 111(B10).
- [111] F. I. Versnyder and M. E. Shank. The development of columnar grain and single crystal high temperature materials through directional solidification. *Materials Science and Engineering*, 6(4) :213 – 247, 1970.
- [112] T. Verstraete, S. Amaral, R. Van den Braembussche, and T. Arts. Design and optimization of the internal cooling channels of a high pressure turbine blade—part ii : Optimization. *Journal of Turbomachinery*, 132, 2010.
- [113] A. Wang and Y. Ma. An error estimate of the proper orthogonal decomposition in model reduction and data compression. *Numerical Methods for Partial Differential Equations*, 25(4) :972–989, 2009.
- [114] K. Washabaugh, D. Amsellem, M. Zahr, and C. Farhat. Nonlinear model reduction for cfd problems using local reduced-order bases. *42nd AIAA Fluid Dynamics Conference and Exhibit. New Orleans, Louisiana, USA*, 2012.
- [115] H.-G. Wenzel. The nanogal software : Earth tide data processing package eterna 3.30. *Bull. Inf. Marées Terrestres*, 124 :9425–9439.
- [116] M. A. Wieczorek. SHTOOLS - Tools for working with spherical harmonics (v2.9.1), oct 2014.
- [117] Wikipedia, the free encyclopedia. Cf6-6 engine cutaway, 2007. [Online; accessed August 17, 2019], image in the public domain.
- [118] M. Yaghoobi, D. Wu, and M. E. Davies. Fast non-negative orthogonal matching pursuit. *IEEE Signal Processing Letters*, 22(9) :1229–1233, 2015.
- [119] M. Yano. A space-time Petrov–Galerkin certified reduced basis method : Application to the Boussinesq equations. *SIAM Journal on Scientific Computing*, 36(1) :A232–A266, 2014.
- [120] M. Yano and A. T. Patera. An lp empirical quadrature procedure for reduced basis treatment of parametrized nonlinear pdes. *Computer Methods in Applied Mechanics and Engineering*, 344 :1104 – 1123, 2019.
- [121] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196(2) :591–626, 2004.
- [122] O. Zahm and A. Nouy. Interpolation of inverse operators for preconditioning parameter-dependent equations. *SIAM Journal on Scientific Computing*, 38(2) :A1044–A1074, 2016.
- [123] Y. Zhang, W. E. Leithead, D. J. Leith, and L. Walshe. Log-det approximation based on uniformly distributed seeds and its application to Gaussian process regression. *Journal of Computational and Applied Mathematics*, 220(1-2) :198–214, 2008.