Reduced order modeling in thermo-mechanics

Amina Benaceur

UNIVERSITÉ PARIS-EST

THÈSE

PRÉSENTÉE POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ PARIS EST

École doctorale MSTIC: mention Mathématiques Appliquées

par **Amina BENACEUR**

Réduction de modèles en thermique et mécanique non-linéaires

Soutenue publiquement le 21 Décembre 2018 devant le jury de thèse composé de:

Dr. Virginie Ehrlacher	ENPC - Université Paris Est	Examinatrice
Pr. Alexandre Ern	ENPC - Université Paris Est	Directeur de thèse
Pr. Yvon Maday	Sorbonne université	Examinateur
Dr. Sébastien Meunier	EDF R&D	Examinateur
Dr. Olga Mula	Université Paris Dauphine	Examinatrice
Pr. Gianluigi Rozza	SISSA	Rapporteur
Pr. David Ryckelynck	École des Mines ParisTech	Examinateur
Pr. Julien Salomon	INRIA Paris	Rapporteur

Aux êtres qui me sont chers,
pour leur soutien immuable.

Encore faut-il pousser une porte pour savoir qu'elle nous est close.
MONTAIGNE.

RÉSUMÉ

Cette thèse propose trois nouveaux développements de la méthode des bases réduites (RB) et de la méthode d'interpolation empirique (EIM) pour des problèmes non-linéaires. La première contribution est une nouvelle méthodologie, la méthode progressive RB-EIM (PREIM) dont l'objectif est de réduire le coût de la phase de construction du modèle réduit tout en maintenant une bonne approximation RB finale. L'idée est d'enrichir progressivement l'approximation EIM et l'espace RB, contrairement à l'approche standard où leurs constructions sont disjointes. La deuxième contribution concerne la RB pour les inéquations variationnelles avec contraintes non-linéaires. Nous proposons une combinaison RB-EIM pour traiter la contrainte. En outre, nous construisons une base réduite pour les multiplicateurs de Lagrange via un algorithme hiérarchique qui conserve la positivité des vecteurs de cette base. Nous appliquons cette stratégie aux problèmes de contact élastique sans frottement pour les maillages non-coïncidents. La troisième contribution concerne la réduction de modèles avec assimilation de données. Une méthode dédiée a été introduite dans la littérature pour combiner un modèle numérique avec des mesures expérimentales. Nous élargissons son cadre d'application aux problèmes instationnaires en exploitant la méthode POD-greedy afin de construire des espaces réduits pour tout le transitoire temporel. Enfin, nous proposons un nouvel algorithme qui produit des espaces réduits plus représentatifs de la solution recherchée tout en minimisant le nombre de mesures nécessaires pour le problème réduit final.

ABSTRACT

This thesis introduces three new developments of the reduced basis method (RB) and the empirical interpolation method (EIM) for nonlinear problems. The first contribution is a new methodology, the Progressive RB-EIM (PREIM) which aims at reducing the cost of the phase during which the reduced model is constructed without compromising the accuracy of the final RB approximation. The idea is to gradually enrich the EIM approximation and the RB space, in contrast to the standard approach where both constructions are separate. The second contribution is related to the RB for variational inequalities with nonlinear constraints. We employ an RB-EIM combination to treat the nonlinear constraint. Also, we build a reduced basis for the Lagrange multipliers via a hierarchical algorithm that preserves the non-negativity of the basis vectors. We apply this strategy to elastic frictionless contact for non-matching meshes. Finally, the third contribution focuses on model reduction with data assimilation. A dedicated method has been introduced in the literature so as to combine numerical models with experimental measurements. We extend the method to a time-dependent framework using a POD-greedy algorithm in order to build accurate reduced spaces for all the time steps. Besides, we devise a new algorithm that produces better reduced spaces while minimizing the number of measurements required for the final reduced problem.

REMERCIEMENTS

Mes trois années de thèse sont désormais écoulées, et il est grand temps de rendre hommage aux personnes ayant contribué à mon arrivée à bon port. La première personne envers laquelle je tiens à témoigner mon immense gratitude est assurément mon directeur de thèse, Alexandre Ern. J'ai eu le privilège d'être encadrée par un chercheur dont les qualités humaines sont aussi admirables que les compétences scientifiques et pédagogiques qui ne font guère débat. En particulier, observer et comprendre sa démarche de recherche scientifique, ainsi que sa déontologie, fut un grand enseignement qui me servira indéniablement dans l'avenir. Je souhaite lui adresser mes remerciements les plus sincères pour la qualité prodigieuse de son encadrement.

Je remercie ensuite mon encadrante académique, Virginie Ehrlacher, pour m'avoir apporté ses conseils tout au long de cette thèse. Son calme, sa patience et sa propension à positiver m'ont particulièrement marqué. D'autant plus que j'ai souvent puisé la motivation dans ses encouragements qui m'ont incité à maintenir le rythme.

En sa qualité d'encadrant industriel, je remercie également Sébastien Meunier. Ma compréhension de la recherche industrielle s'est principalement forgée en travaillant avec lui. Notamment, il m'a transmis avec volonté et application, et non sans humour, les méthodes de travail dans l'industrie.

Je souhaite maintenant remercier Gianluigi Rozza et Julien Salomon, pour avoir accepté de rapporter ma thèse. Je tiens également à remercier Yvon Maday, Olga Mula et David Ryckelynck pour leur présence dans mon jury de thèse. Aussi, une reconnaissance particulière est celle que j'éprouve envers Anthony Patera qui m'a accordé l'honneur de collaborer avec lui et m'a cordialement accueillie au MIT.

Plusieurs personnes que j'ai côtoyées durant cette thèse ont contribué à son bon déroulement. En particulier à EDF, je souhaite remercier Mickaël pour ses conseils et ses avis pertinents lors de mes comités de thèse. Mes remerciements vont également à Géraud, Marc, Chu, Guilhem, Jean-Philippe, Jérôme, Jean-François, Nicolas et Cécile pour l'intérêt porté à mes travaux. Toujours à EDF, une pensée particulière va à ma remarquable amie et collègue de bureau, Maoyuan, ainsi qu'à la bande de copains

avec qui je me suis longuement attelée à refaire le monde lors des pauses déjeuner ou à l'occasion de moments partagés hors labo, je cite en particulier Georges-Arthur, Ziling, Arina, Qiwei et Aboubakr. Je remercie aussi Jean-Christophe, Dominique, Anna, Gilles, ainsi que tous mes autres collègues pour leur accueil et leur affabilité notable. Au CERMICS, je souhaite remercier Isabelle pour son aide administrative, son efficacité, et la bonne humeur qu'elle veille à préserver au sein du laboratoire. Je remercie également les désormais docteurs Yannick, Athmane, Pierre et François pour leur accueil, puis Oumaïma, Riccardo, Alexandre, Julien, Frédéric et tous les autres doctorants et post-doctorants pour les discussions parfois à bâtons rompus lors des multiples pauses partagées ensemble. Je remercie chaleureusement la clique de l'INRIA: Ani, Jad, Karol, Matteo, Mohammad, Nicolas, Patrik et Simon. Merci pour le soutien, les gags et les débats quelquefois indécidables qui égayaient mon quotidien parmi vous; sans oublier les différentes sorties. Je remercie aussi Michel et Géraldine pour leurs encouragements. Je tiens par ailleurs à reconnaître les rôles de Mohamed El Hachmi et de Vincent Brunel dont les enseignements m'ont donné de l'élan plusieurs années durant.

Je saisis cette occasion afin d'éterniser par le biais de l'encre ma profonde gratitude à l'égard des plus proches. Maman, Papa, Abderrahim, Oussama et Mariam, mille mercis d'avoir constamment et inconditionnellement cru en moi, surtout lorsque je craignais sérieusement de faillir. Ce dernier diplôme, tout comme ceux qui l'ont précédé, n'aurait pu être décroché sans votre soutien indéfectible. Mes mots, aussi expressifs soient-ils, ne sauraient restituer la reconnaissance que je vous dois. Enfin, je remercie tous mes amis et les autres membres de ma famille, notamment les Chraas et les Jamils.

CONTENTS

CHAPTER 1

INTRODUCTION

Ce chapitre développe plusieurs éléments permettant de situer les travaux présentés dans ce manuscrit. Tout d'abord, nous détaillons le contexte industriel ayant donné lieu à cette thèse. Ensuite, nous abordons des aspects théoriques généraux sur la réduction de modèles, en particulier sur la méthode des bases réduites. Ces éléments serviront d'introduction aux chapitres suivants pour le lecteur peu familier avec le sujet. Enfin, nous dressons un bref état de l'art suivi d'une description des principaux résultats de cette thèse.

1.1 Contexte industriel

Cette thèse se consacre à une classe de problèmes thermo-mécaniques, qui est en particulier celle rencontrée dans les études de robinetterie au sein d'EDF R&D. Dans cette section, nous présentons brièvement le cadre de ces études.

1.1.1 Études de robinetterie et enjeux industriels

En tant qu'équipements importants pour la sûreté et la disponibilité des centrales nucléaires, certains matériels de robinetterie doivent être qualifiés aux conditions normales, incidentelles et accidentelles de fonctionnement. La qualification a pour objectif de vérifier la capacité du matériel à assurer sa fonction dans des conditions de fonctionnement et pour une durée de vie spécifiées.

Dans le cadre d'un partenariat entre le département Mécanique des Matériaux et Composants (MMC) d'EDF R&D et un fabricant de robinets, un essai sur un robinet à maintenance allégée (RAMA) fortement instrumenté a été mis en place sur la boucle de cyclage thermique (CYTHER) (cf. Figure 1.1). L'expérience consiste en une succession de chocs thermiques alternés par passage d'un fluide sous pression. Il résulte de cet essai une quantité importante de données expérimentales, telles que des températures mesurées par des thermocouples, des efforts de serrage dans des goujons, des déformations résiduelles et bien d'autres mesures. Une étude récente

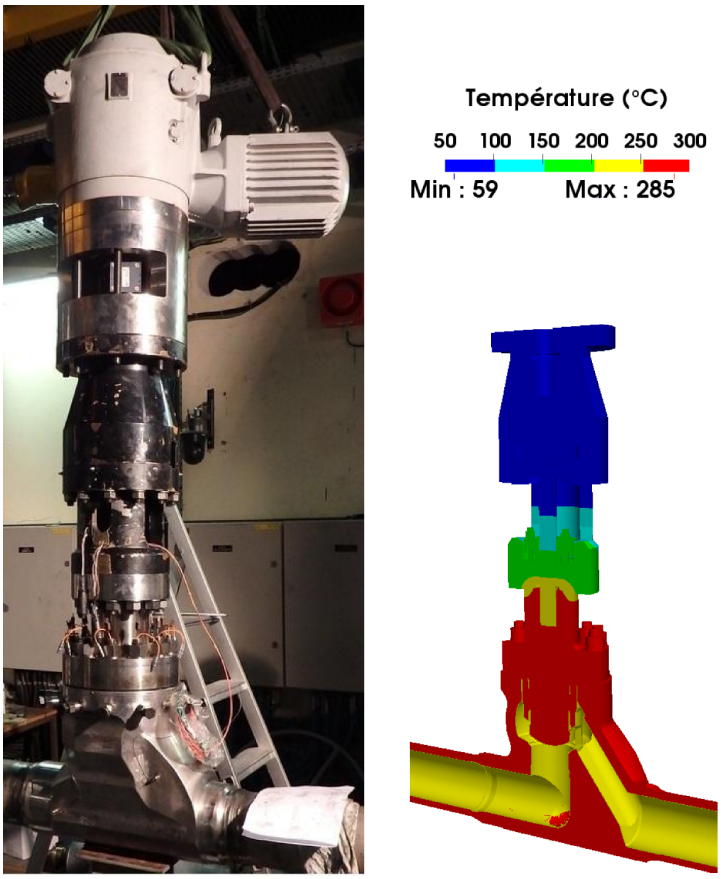

Figure 1.1 – Prototype du robinet RAMA instrumenté. À gauche: image réelle du robinet. À droite: Simulation numérique.

avait pour objectif d'étudier et de comparer les données obtenues lors de l'essai aux résultats des simulations numériques multi-physiques. Les accords sont des éléments de validation et les désaccords indiquent les limites des modèles et outils de calcul.

Pour la robinetterie et bien d'autres projets, plusieurs travaux sont initiés à EDF R&D sur la réduction de modèles et l'assimilation de données. Afin d'éviter des coûts d'essais prohibitifs, les constructeurs de robinets pour les centrales nucléaires ont de plus en plus recours à la simulation numérique afin d'étudier le comportement de leur matériel en conditions de qualification. EDF R&D recourt également à la simulation numérique pour évaluer les simulations des fabricants. Dans certaines configurations, même avec les ressources de calcul actuelles, il peut s'avérer difficile, voire impossible, d'utiliser les méthodes numériques standards. Ces calculs (par éléments finis) sont généralement lourds et impliquent des maillages très fins, beaucoup de pas de temps, des lois de comportement complexes et parfois la prise en compte de contraintes physiques non-linéaires comme les conditions de non-interpénétration dans les phénomènes de contact [42]. De plus, ces calculs sont souvent indispensables pour étudier l'influence de paramètres sur un modèle ou recaler des paramètres. De telles analyses nécessitent beaucoup d'appels à un modèle éléments finis complexe; elles sont alors rendues difficiles à cause des temps de calcul considérables qu'elles requièrent. Les méthodes de 'méta-modèles' consistent à remplacer un modèle com-

plexe par un modèle approché beaucoup moins coûteux. Les méthodes dites de 'chaos polynomial' sont une famille de méthodes de méta-modèles étudiées [9, 11] et implémentées [7] depuis plusieurs années à EDF R&D.

En raison de contraintes de temps et de moyens, il serait d'un grand intérêt industriel de concevoir des modélisations simplifiées, permettant des simulations à temps de calcul réduit, et dont les résultats seraient similaires (à une marge d'erreur acceptable près) à ceux des simulations plus complexes qu'EDF R&D développe en interne, et par conséquent au coût de calcul nettement plus élevé.

1.1.2 Réduction de modèles

Plusieurs domaines de l'ingénierie requièrent de pouvoir résoudre numériquement des équations aux dérivées partielles (EDP) modélisant des phénomènes physiques paramétriques. Deux configurations sont particulièrement récurrentes

- **Les études multi-requêtes:** Ce type d'études se caractérise par le besoin de résoudre plusieurs problèmes du même genre mais qui diffèrent légèrement les uns des autres; ce qui soulève naturellement la question du coût excessif de l'utilisation répétée de la même méthode numérique. Bien que ces méthodes puissent être très performantes en soi, elles n'ont pas été développées pour une utilisation répétitive.

- **Les simulations en temps réel:** Ce type d'études exige l'immédiateté de la résolution. Néanmoins, les méthodes standard ne parviennent pas à fournir des résultats aussi rapidement.

Les techniques de réduction de modèle [46, 47] présentent un intérêt considérable dans un tel contexte où on considère un problème dont la solution dépend de paramètres. Ainsi, pour un ensemble donné de paramètres, ces méthodes permettent d'obtenir la solution correspondante après un temps de calcul relativement court. Pour parvenir à de tels résultats, il est crucial de calculer et stocker en amont certaines quantités caractéristiques du problème qui seront utilisées lors du calcul des futures solutions. Ainsi, les méthodes de réduction de modèle se structurent en deux étapes consécutives. Pendant la première étape, dite 'hors-ligne', les calculs coûteux utilisant les modèles fins sont réalisés, permettant ainsi de créer une bibliothèque de calculs. La caractéristique principale de la phase hors-ligne est d'être effectuée une seule et unique fois. A l'opposé, la deuxième partie, dite 'en-ligne', consiste en la résolution de systèmes réduits découlant d'une reformulation du problème initial basée sur l'apprentissage hors-ligne. La phase en-ligne est donc effectuée autant de fois qu'un nouveau paramètre est choisi pour la résolution. Son coût de résolution est très faible afin d'assurer l'efficacité de l'approche.

Dans cette thèse, nous nous intéressons à deux applications importantes de la réduction de modèles. Une première application est l'étude et la simulation des problèmes thermiques instationnaires. Cela est motivé conjointement par la multitude d'applications de la thermique non-linéaire instationnaire (robinetterie, générateur de vapeur, cuve, etc) et la simplicité théorique des problèmes de thermique, au

moins dans le cas linéaire, par rapport aux problèmes de mécanique. La deuxième application est celle des problèmes de mécanique de contact qui sont également très importants, notamment pour les calculs chaînés thermo-mécaniques. En effet, la robinetterie a la caractéristique d'impliquer du contact unilatéral, car les robinets sont des assemblages de plusieurs pièces (cf. Figure 1.2). En outre, les quantités

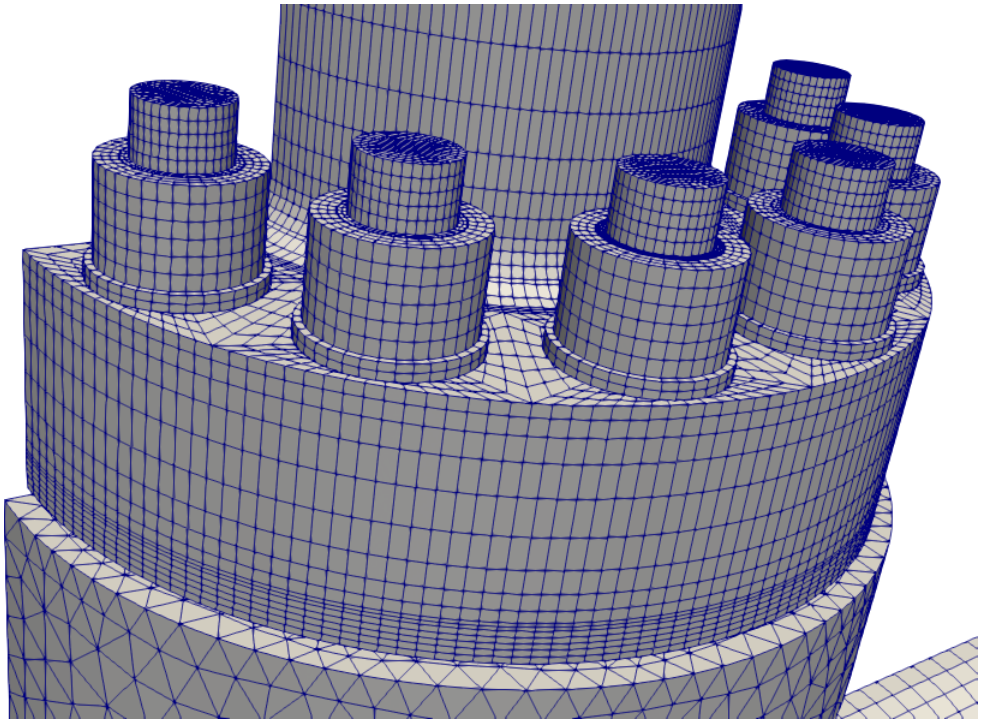

Figure 1.2 – Exemple de contact entre six goujons et une pièce du robinet RAMA (Remerciement à VELAN SAS pour la fourniture des plans nécessaires pour le maillage).

d'intérêt pour EDF R&D sont celles issues d'un chaînage thermo-mécanique, d'où la complémentarité des deux problèmes de réduction thermique et mécanique. Il est également à noter que les calculs mécaniques en robinetterie sont nettement plus coûteux que les calculs thermiques. En effet, les calculs thermiques se résolvent en un temps qui est de l'ordre de l'heure alors que les calculs mécaniques se résolvent en un temps de calcul qui est de l'ordre de la dizaine d'heures. Cela justifie l'intérêt porté à la réduction de modèles en mécanique de contact où les gains escomptés sont importants. De plus, il s'agit d'un domaine de recherche relativement neuf où les premiers travaux datent de 2016 [4], alors que les premiers travaux sur la thermique non-linéaire datent de 2007 [24].

1.1.3 Assimilation de données

Plusieurs projets au sein d'EDF R&D traitent à la fois de modèles numériques et de mesures expérimentales. Ainsi, la capacité à exploiter conjointement les résultats issus des deux approches de modélisation et d'expérimentation est d'un grand intérêt.

Parmi les applications concernées à EDF R&D figurent la fatigue au département MMC, le génie civil à travers la maquette de vérification réaliste du confinement des réacteurs (VERCORS) au département MMC, la neutronique au département PERICLES (Performance et Prévention des Risques Industriels du Parc par la Simulation et les Études) ainsi que la robinetterie à MMC (cf. Section 1.1.1). L'application industrielle envisagée dans le futur concerne les tests liés à ce sujet. De telles campagnes d'essais sont particulièrement onéreuses et chronophages et ne peuvent donc être systématiquement envisagées. Compte tenu de ces contraintes de temps et de moyens, les constructeurs s'appuient sur des modélisations simplifiées, permettant des simulations à un temps de calcul raisonnable. Dans ce contexte, disposer à la fois d'un modèle complexe, de modèles plus ou moins simplifiés et de mesures expérimentales ouvre plusieurs perspectives. En premier lieu, la capacité d'EDF R&D à porter un regard critique sur les modèles numériques utilisés en se servant des mesures expérimentales comme d'une référence. En outre, l'assimilation de données combinée avec la réduction de modèles pourra être considérée dans l'étude d'un robinet RAMA.

1.2 Réduction de modèles

Dans cette section, nous rappelons quelques notions relatives à la réduction de modèles qui seront utiles pour la lecture de ce manuscrit. En particulier, nous détaillons la méthode des bases réduites (RBM; de l'anglais: Reduced Basis Method). La RBM est une méthode dite *a priori*. Elle ne nécessite aucune connaissance de la solution, car cette connaissance est construite à mesure que le calcul progresse et doit donc être réadaptée à tout nouveau type de problème. Deux méthodes de construction lors de la phase d'apprentissage hors-ligne d'une base réduite dans le cadre de la RBM seront présentées: une compression d'un ensemble de solutions précalculées par décomposition propre orthogonale (POD) [35] et un algorithme glouton - ou greedy - pour générer ces solutions de manière progressive [10, 12]. La POD est une méthode dite *a posteriori* car elle requiert des informations préalables sur la solution du problème d'intérêt.

1.2.1 Problème modèle

Cette section se limite aux problèmes linéaires. Nous nous intéressons à un modèle mathématique qui décrit le comportement physique d'un système en s'appuyant sur une EDP paramétrique. L'ensemble des paramètres sert à identifier une configuration particulière du système décrit par le modèle (et donc les EDPs sous-jacentes) et sera identifié comme entrée du modèle. Notons $\mathcal{P}$ l'ensemble de paramètres dont les éléments seront notés $\mu \in \mathcal{P} \subset \mathbb{R}^P$, avec $P \geqslant 1$. Ces paramètres d'entrée peuvent être des propriétés physiques de matériaux, des variables caractérisant la géométrie, des efforts variables, etc. Soit Ω un domaine borné, régulier, dans $\mathbb{R}^d$ ($d = 1, 2, 3$). Le domaine Ω peut dépendre de μ. Ce sera le cas lorsque nous étudierons les problèmes de contact, mais dans cette introduction, nous supposons pour simplifier que Ω ne

dépend pas de μ. Soit X un espace de fonctions sur Ω muni d'un produit scalaire $(\cdot,\cdot)_X$ et de la norme associée $\|\cdot\|_X$. Nous notons $X_\mathcal{N} \subset X$ un sous-espace vectoriel de X de dimension $\mathcal{N}$ ($X_\mathcal{N}$ est typiquement un espace d'approximation éléments finis [20]). Soient une forme bilinéaire symétrique et paramétrique $a : \mathcal{P} \times X \times X \to \mathbb{R}$ et une forme linéaire paramétrique $f : \mathcal{P} \times X \to \mathbb{R}$. Les caractères bilinéaire symétrique de a et linéaire de f sont par rapport à leurs variables d'entrée dans X. Nous nous intéressons à la résolution du problème paramétrique suivant: trouver $u(\mu) \in X$ vérifiant

$$a(\mu; u(\mu), v) = f(\mu; v), \quad \forall v \in X. \tag{1.1}$$

Le caractère bien posé du problème (1.1) est garanti par le théorème de Lax-Milgram sous les hypothèses suivantes:

(i) $a(\mu;\cdot,\cdot)$ est coercive et continue sur $X \times X$ uniformément pour tout $\mu \in \mathcal{P}$, i.e., il existe une constante positive $\alpha > 0$ et une constante $\beta^a < \infty$ vérifiant

$$\begin{cases} \forall v \in X,\ \forall \mu \in \mathcal{P}, \quad \alpha\|v\|_X^2 \leqslant \alpha_{\mathrm{LB}}(\mu)\|v\|_X^2 \leqslant a(\mu, v, v), \\ \forall v, w \in X,\ \forall \mu \in \mathcal{P}, \quad a(\mu; v, w) \leqslant \beta^a\|v\|_X\|w\|_X. \end{cases} \tag{1.2}$$

(ii) $f(\mu;\cdot)$ est uniformément continue pour tout $\mu \in \mathcal{P}$, i.e., il existe une constante $\beta^f < \infty$ vérifiant

$$\forall v \in X,\ \forall \mu \in \mathcal{P}, \quad f(\mu; v) \leqslant \beta^f\|v\|_X. \tag{1.3}$$

Sous ces hypothèses, le problème (1.1) est bien posé. Toutefois, pour la majorité des cas pratiques, la solution exacte du problème (1.1) est inaccessible. En revanche, la projection de Galerkin de (1.1) dans $X_\mathcal{N}$, notée $u_\mathcal{N}(\mu)$, est accessible au calcul. Elle est solution du problème suivant: trouver $u_\mathcal{N}(\mu) \in X_\mathcal{N}$ vérifiant

$$a(\mu; u_\mathcal{N}(\mu), v_\mathcal{N}) = f(\mu; v_\mathcal{N}), \quad \forall v_\mathcal{N} \in X_\mathcal{N}. \tag{1.4}$$

Ici, l'entier $\mathcal{N}$ désigne la dimension de l'espace $X_\mathcal{N}$ et permet de quantifier le coût de calcul de $u_\mathcal{N}(\mu)$ pour chaque $\mu \in \mathcal{P}$. Dorénavant, la solution $u_\mathcal{N}(\mu)$ de (1.4) sera dite solution 'haute-fidélité'. Cette solution est habituellement calculée par des solveurs éléments finis (`code_aster` [19], `FreeFem++` [28], etc). En général, la résolution de (1.4) pour une valeur $\mu \in \mathcal{P}$ est onéreuse.

1.2.2 La méthode des bases réduites (RBM)

Nous introduisons la notion de variété - ou manifold - de solutions $\mathcal{M}$ qui est l'ensemble des solutions du problème paramétrique (1.1) pour l'ensemble des paramètres $\mathcal{P}$. La variété $\mathcal{M}$ est donc définie par

$$\mathcal{M} = \{u(\mu) \in X, \forall \mu \in \mathcal{P}\} \subset X. \tag{1.5}$$

De la même manière, nous définissons la version discrète $\mathcal{M}_\mathcal{N}$ de la variété de solutions $\mathcal{M}$:

$$\mathcal{M}_\mathcal{N} = \{u_\mathcal{N}(\mu) \in X_\mathcal{N}, \forall \mu \in \mathcal{P}\} \subset X_\mathcal{N}. \tag{1.6}$$

La méthode des bases réduites (RBM) repose sur la prémisse selon laquelle la variété $\mathcal{M}_\mathcal{N}$ peut être approchée avec une très bonne précision par un sous-espace vectoriel 'réduit' $X_N \subset X_\mathcal{N}$ de dimension N bien plus faible devant $\mathcal{N}$. Ainsi, la RBM se base sur une projection de type Galerkin sur l'espace X_N construit de telle sorte à ce qu'il approche au mieux la variété $\mathcal{M}_\mathcal{N}$ (cf. Figure 1.3). Ainsi, le problème (1.4)

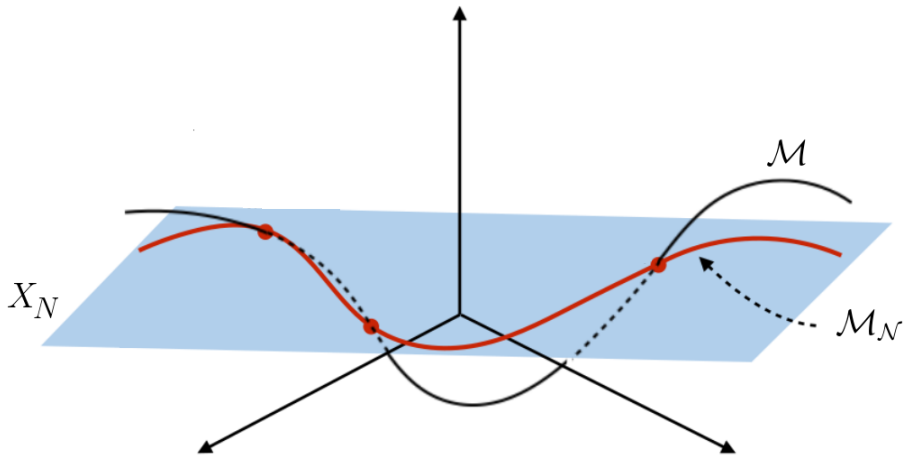

Figure 1.3 – Illustration de l'espace réduit X_N et des variétés continue $\mathcal{M}$ et discrète $\mathcal{M}_\mathcal{N}$ de solutions [48].

en formulation réduite s'écrit: trouver $u_N(\mu) \in X_N$ tel que

$$a(\mu; u_N(\mu), v_N) = f(\mu; v_N), \quad \forall v_N \in X_N. \tag{1.7}$$

Soient $\{\phi_1, \dots, \phi_\mathcal{N}\}$ une base de l'espace éléments finis $X_\mathcal{N}$ et $\{\theta_1, \dots, \theta_N\}$ une base de l'espace réduit X_N dont la représentation matricielle sera notée $\mathbf{X}_N \in \mathbb{R}^{\mathcal{N}\times N}$. La construction de la base $\{\theta_1, \dots, \theta_N\}$ sera discutée dans la section 1.2.3 ci-dessous. La solution réduite $u_N(\mu)$ se décompose dans cette base sous la forme $u_N(\mu) = \sum_{n=1}^N u_{N,n}(\mu)\theta_n$. Par conséquent, le problème (1.7) est reformulé sous la forme

$$\sum_{n=1}^N u_n(\mu) a(\mu; \theta_n, \theta_p) = f(\mu, \theta_p), \quad \forall 1 \leqslant p \leqslant N, \tag{1.8}$$

À ce stade, nous définissons la matrice réduite $\hat{\mathbf{A}}$ et le vecteur réduit $\hat{\mathbf{f}}$ tels que

$$(\hat{\mathbf{A}}(\mu))_{np} = (a(\mu; \theta_n, \theta_p))_{np}, \quad \text{et} \quad (\hat{\mathbf{f}}(\mu))_p = (f(\mu; \theta_p))_p, \quad \forall 1 \leqslant n, p \leqslant N, \tag{1.9}$$

qui sont facilement calculables via les formules

$$\hat{\mathbf{A}}(\mu) = \mathbf{X}_N^T \mathbf{A}(\mu) \mathbf{X}_N, \quad \text{et} \quad \hat{\mathbf{f}}(\mu) = \mathbf{X}_N^T \mathbf{f}(\mu), \tag{1.10}$$

où la matrice $\mathbf{A}$ et le vecteur $\mathbf{f}$ sont issus du problème haute-fidélité (1.4) résolu par éléments finis et sont évalués en utilisant la base $\{\phi_1, \dots, \phi_\mathcal{N}\}$. De cette manière, la RBM revient pour tout paramètre $\mu \in \mathcal{P}$ à résoudre le système linéaire réduit

$$\hat{\mathbf{A}}(\mu)\mathbf{u}_N(\mu) = \hat{\mathbf{f}}(\mu). \tag{1.11}$$

Pour une résolution rapide et peu onéreuse de (1.7) ou de sa formulation matricielle (1.11), nous faisons l'hypothèse cruciale que a et f admettent une dépendance affine en le paramètre $\mu \in \mathcal{P}$, i.e. il existe $Q^a, Q^f \in \mathbb{N}^*$ **petits** devant $\mathcal{N}$ tels que

$$a(\mu; v, w) = \sum_{q=1}^{Q^a} \sigma_q^a(\mu) a_q(v, w), \quad \text{et} \quad f(\mu; v) = \sum_{q=1}^{Q^f} \sigma_q^f(\mu) f_q(v), \tag{1.12}$$

où pour tout entier q, $\sigma_q^a : \mathcal{P} \to \mathbb{R}$ et $\sigma_q^f : \mathcal{P} \to \mathbb{R}$ sont des fonctions du paramètre uniquement, $a_q : X \times X \to \mathbb{R}$ est une forme bilinéaire continue et symétrique et $f_q : X \to \mathbb{R}$ est une forme linéaire continue. Cette hypothèse restrictive sur a et f augmente significativement les performances des calculs numériques en évitant de recalculer *ex nihilo* la forme bilinéaire $a(\mu; v, w)$ et la forme linéaire $f(\mu; v)$ pour chaque nouveau paramètre $\mu \in \mathcal{P}$. En effet, une fois les formes bilinéaires $(a_1, \ldots, a_{Q^a})$ et les formes linéaires $\left(f_1, \ldots, f_{Q^f}\right)$ stockées lors de la phase d'apprentissage hors-ligne, il suffit d'évaluer les quantités $\left(\sigma_1^a(\mu), \ldots, \sigma_{Q^a}^a(\mu)\right)$ et $\left(\sigma_1^f(\mu), \ldots, \sigma_{Q^f}^f(\mu)\right)$ pendant la phase en-ligne pour assembler ensuite les expressions (1.12) et obtenir ainsi les valeurs de $a(\mu; v, w)$ et de $f(\mu; v)$. Le raisonnement est identique pour la version matricielle (1.11) du problème. Le calcul et le stockage des matrices $(\mathbf{A}_1, \ldots, \mathbf{A}_{Q^a})$ et des vecteurs $\left(\mathbf{f}_1, \ldots, \mathbf{f}_{Q^f}\right)$ s'effectuent hors-ligne, tandis que le calcul des quantités $\left(\sigma_1^a(\mu), \ldots, \sigma_{Q^a}^a(\mu)\right)$ et $\left(\sigma_1^f(\mu), \ldots, \sigma_{Q^f}^f(\mu)\right)$ se fait en-ligne. Cela permet l'assemblage de l'équivalent matriciel de (1.12) donné par les formules

$$\mathbf{A}(\mu) = \sum_{q=1}^{Q^a} \sigma_q^a(\mu) \mathbf{A}_q, \quad \text{et} \quad \mathbf{f}(\mu) = \sum_{q=1}^{Q^f} \sigma_q^f(\mu) \mathbf{f}_q, \tag{1.13}$$

L'hypothèse (1.12) est rarement satisfaite d'emblée. Nous verrons dans la section 1.2.5 que dans le cas général où l'EDP ne dépend pas des paramètres de manière affine, il est possible de se ramener au cadre (1.12) moyennant une méthode d'interpolation empirique (EIM); ce qui introduit une nouvelle source d'erreur d'approximation qui peut être *a priori* contrôlée par le biais des entiers Q^a et Q^f.

1.2.3 Génération de la base réduite

Le meilleur espace réduit X_N est -sous réserve d'existence- celui qui minimise l'erreur maximale de projection, dite épaisseur de Kolmogorov d'ordre N de la variété discrète $\mathcal{M}_\mathcal{N}$, et donnée par la formule

$$d_N(\mathcal{M}_\mathcal{N}) = \inf_{\dim(W)=N} \sup_{\mu \in \mathcal{P}} \inf_{w \in W} \|u_\mathcal{N}(\mu) - w\|_X. \tag{1.14}$$

Quand l'épaisseur d_N est petite, unebonne approximation de la variété discrète $\mathcal{M}_\mathcal{N}$ est possible par un espace X_N de très petite dimension $N \ll \mathcal{N}$. Toutefois, l'espace qui réalise le minimum dans la définition de d_N est souvent inaccessible. Dans la littérature, on recense plusieurs stratégies de construction d'espaces réduits. Les plus largement utilisées sont la POD [35] et l'algorithme glouton - ou greedy - [10, 12].

Dans les deux cas, nous partons d'un ensemble d'apprentissage $\mathcal{P}^{\text{tr}} \subset \mathcal{P}$. À titre d'exemple, $\mathcal{P}^{\text{tr}}$ peut être constitué par un échantillonnage uniforme, log-uniforme ou aléatoire de $\mathcal{P}$. Cet ensemble d'apprentissage peut aussi être déterminé grâce à une connaissance physique de valeurs de paramètres pertinentes pour le problème d'intérêt.

1.2.3.1 La décomposition propre orthogonale (POD)

Supposons disposer d'un ensemble de solutions $\mathcal{S} \subset \mathcal{M}_{\mathcal{N}}$ de (1.4) pour un sous-ensemble de paramètres d'apprentissage $\mathcal{P}^{\text{tr}} \subset \mathcal{P}$. Si $\mathcal{S}$ est suffisamment riche, alors l'espace vectoriel engendré par ses éléments constitue une bonne approximation de $\mathcal{M}_{\mathcal{N}}$. Le but de la POD est de compresser (i.e. réduire la taille) de l'ensemble $\mathcal{S}$ tout en conservant des bonnes propriétés d'approximation de l'espace vectoriel engendré par ses éléments. La qualité de l'approximation est quantifiée par un paramètre ϵ_{POD} donné par l'utilisateur. Nous utilisons la notation

$$(\theta_1, \dots, \theta_N) = \text{POD}(\mathcal{S}, \epsilon_{\text{POD}}), \tag{1.15}$$

où $\mathcal{S} = (v_1, \dots, v_R)$ est composé de $R \geqslant 1$ fonctions de l'espace $X_{\mathcal{N}}$ et ϵ_{POD} est la tolérance définie par l'utilisateur. Nous adoptons une description algébrique pour plus de clarté, et nous renvoyons le lecteur à [25] pour un exposé détaillé de la méthode POD. Soit $(\varrho_1, \dots, \varrho_{\mathcal{N}})$ une base de $X_{\mathcal{N}}$. Pour une fonction $w \in X_{\mathcal{N}}$, on note $\mathbf{w} := (w_j)_{1 \leqslant j \leqslant \mathcal{N}}$ son vecteur de composantes dans $\mathbb{R}^{\mathcal{N}}$, de telle sorte que $w = \sum_{j=1}^{\mathcal{N}} w_j \varrho_j$. La version algébrique de la procédure (2.29) consiste à se donner R vecteurs formant une matrice rectangulaire $\mathbf{S} := (\mathbf{v}_1, \dots, \mathbf{v}_R) \in \mathbb{R}^{\mathcal{N} \times R}$, et à chercher N vecteurs formant la matrice rectangulaire $\boldsymbol{\Theta} := (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) \in \mathbb{R}^{\mathcal{N} \times N}$. Les vecteurs $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N)$ doivent être orthonormaux par rapport à la matrice de Gram du produit scalaire de X. Dans ce contexte, on considère la matrice de Gram $\mathbf{C}_{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ telle que

$$\mathbf{C}_{\mathcal{N}} = \Big(a(\mu_0; \varrho_n, \varrho_p) \Big)_{1 \leqslant p,n \leqslant \mathcal{N}}, \tag{1.16}$$

où la forme bilinéaire $a(\mu_0; \cdot, \cdot)$ est celle utilisée dans (1.1) pour une valeur représentative μ_0 du paramètre. Ainsi, nous cherchons à vérifier $\boldsymbol{\theta}_n^{\mathrm{T}} \mathbf{C}_{\mathcal{N}} \boldsymbol{\theta}_p = \delta_{n,p}$, où $\delta_{n,p}$ est l'indice de Kronecker, pour tous $n, p \in \{1, \dots, N\}$.

Soit $\mathbf{T} := (\mathbf{C}_{\mathcal{N}})^{\frac{1}{2}} \mathbf{S} \in \mathbb{R}^{\mathcal{N} \times R}$ et soit l'entier $D = \min(\mathcal{N}, R)$ (en général, $D = R$ et $D \ll \mathcal{N}$, ce que nous supposons par la suite). La décomposition en valeurs singulières (SVD) [44] de la matrice $\mathbf{T}$ retourne les réels $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_D \geqslant 0$, la famille de vecteurs colonnes orthonormaux $(\boldsymbol{\xi}_n)_{1 \leqslant n \leqslant D} \in (\mathbb{R}^{\mathcal{N}})^D$ (tels que $\boldsymbol{\xi}_n^{\mathrm{T}} \boldsymbol{\xi}_p = \delta_{p,n}$) et la famille de vecteurs colonnes orthonormaux $(\hat{\boldsymbol{\psi}}_n)_{1 \leqslant n \leqslant D} \in (\mathbb{R}^R)^D$ (tels que $\hat{\boldsymbol{\psi}}_n^{\mathrm{T}} \hat{\boldsymbol{\psi}}_p = \delta_{p,n}$) de telle manière que soit satisfaite la relation

$$\mathbf{T} = \sum_{n=1}^{D} \sigma_n \boldsymbol{\xi}_n \hat{\boldsymbol{\psi}}_n^{\mathrm{T}}. \tag{1.17}$$

À partir de (2.31), nous déduisons que $\mathbf{T} \hat{\boldsymbol{\psi}}_n = \sigma_n \boldsymbol{\xi}_n$ et $\mathbf{T}^{\mathrm{T}} \boldsymbol{\xi}_n = \sigma_n \hat{\boldsymbol{\psi}}_n$ pour tout $n \in \{1, \dots, D\}$. Enfin, les vecteurs recherchés sont donnés par $\boldsymbol{\theta}_n := (\mathbf{C}_{\mathcal{N}})^{-\frac{1}{2}} \boldsymbol{\xi}_n$ pour

tout $n \in \{1, \dots, N\}$ avec $N := \max\{1 \leqslant n \leqslant D \mid \sigma_n \geqslant \epsilon_{\text{POD}}\}$. Il est intéressant de remarquer que, parmi tous les sous-espaces $\mathbf{Z}_N$ de dimension N dans $\mathbb{R}^{\mathcal{N}}$, l'espace de dimension N engendré par les vecteurs $(\boldsymbol{\theta}_n)_{1 \leqslant n \leqslant N}$ est celui qui minimise la quantité

$$\sum_{r=1}^{R} \inf_{\mathbf{z} \in \mathbf{Z}_N} (\mathbf{v}_r - \mathbf{z})^{\mathrm{T}} \mathbf{C}_{\mathcal{N}} (\mathbf{v}_r - \mathbf{z}) \tag{1.18}$$

En notant $Z_N = \text{vect}\{\theta_1, \dots, \theta_N\}$, nous avons de manière équivalente, que Z_N minimise la quantité

$$\sum_{r=1}^{R} \inf_{z \in Z_N} \|v_r - z\|_X^2 \tag{1.19}$$

De plus, nous avons $\|v - \Pi_{Z_N} v\|_X \leqslant \sigma_{N+1} \|v\|_X$, pour tout $v \in \mathcal{S}$, où Π_{Z_N} désigne le projecteur X-orthogonal sur Z_N. Dans la pratique, il est possible d'éviter le calcul de la matrice $(\mathbf{C}_{\mathcal{N}})^{\frac{1}{2}}$ et de son inverse en considérant la matrice de plus petite dimension $\mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_{\mathcal{N}}\mathbf{S} \in \mathbb{R}^{R \times R}$, dite matrice d'auto-corrélation. Comme $\mathbf{T}^{\mathrm{T}}\mathbf{T}\hat{\boldsymbol{\psi}}_n = \sigma_n \mathbf{T}^{\mathrm{T}}\boldsymbol{\xi}_n = \sigma_n^2 \hat{\boldsymbol{\psi}}_n$, la résolution du problème aux valeurs propres associé à $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ donne les vecteurs $\hat{\boldsymbol{\psi}}_n$ et leurs valeurs propres associées σ_n^2. Ainsi, les vecteurs $(\boldsymbol{\theta}_n)_{1 \leqslant n \leqslant N}$ sont obtenus à travers la relation

$$\boldsymbol{\theta}_n = (\mathbf{C}_{\mathcal{N}})^{-\frac{1}{2}} \boldsymbol{\xi}_n = \frac{1}{\sigma_n} (\mathbf{C}_{\mathcal{N}})^{-\frac{1}{2}} \mathbf{T}\hat{\boldsymbol{\psi}}_n = \frac{1}{\sigma_n} \mathbf{S}\hat{\boldsymbol{\psi}}_n. \tag{1.20}$$

L'intérêt de la méthode POD est qu'elle permet en général de capturer un maximum d'information en norme X en utilisant très peu de modes. Par conséquent, en cherchant la solution RBM de (1.7) sous la forme

$$u_N(\mu) = \sum_{n=1}^{N} u_n(\mu) \theta_n, \tag{1.21}$$

on obtient une troncature raisonnable en norme X puisque les vecteurs propres associés aux plus grandes valeurs propres sont conservés. Par ailleurs, une tolérance définie par l'utilisateur guide la POD dans la sélection des modes dominants. Concrètement, sont conservés les premiers modes dominants dont la somme des valeurs propres associées par rapport à la trace de la matrice d'auto-corrélation est supérieure à $1 - \epsilon_{\text{POD}}$ pour une tolérance ϵ_{POD} donnée (10^{-2} par exemple).

1.2.3.2 L'algorithme glouton

Contrairement à la POD, l'algorithme glouton [10, 12] est une méthode itérative dans laquelle, à chaque itération, une nouvelle fonction de base réduite est ajoutée et la précision globale de la base est en général améliorée. Pour cela, il faut calculer une solution du problème haute-fidélité par itération et un total de N solutions haute-fidélité pour générer l'espace réduit de dimension N. On notera au passage que la POD nécessite le calcul de $R \geqslant N$ solutions haute-fidélité. L'algorithme glouton nécessite un estimateur d'erreur $\Delta_N(\mu)$ qui prédit l'erreur due à la réduction de dimension du problème, c'est-à-dire qui fournit une estimation de l'erreur induite

en remplaçant $X_{\mathcal{N}}$ dans le problème haute-fidélité par l'espace réduit X_N. Nous détaillerons le calcul de $\Delta_N : \mathcal{P} \to \mathbb{R}$ en Section 1.2.4 ci-dessous. L'estimation d'erreur permet non seulement de quantifier et de contrôler l'erreur commise par la RBM, mais aussi d'explorer l'espace des paramètres de manière optimale. En général, l'estimateur d'erreur Δ_N vérifie l'inégalité

$$\|u_{\mathcal{N}}(\mu) - u_N(\mu)\|_X \leqslant \Delta_N(\mu), \quad \forall \mu \in \mathcal{P}. \tag{1.22}$$

L'ensemble des solutions réduites pour $\mu \in \mathcal{P}$ ne pouvant en général pas être exploré de manière exhaustive, on explore uniquement un sous-ensemble $\mathcal{P}^{\text{tr}} \subset \mathcal{P}$ qu'on appelle ensemble d'apprentissage.

L'initialisation de la base réduite se fait par un premier calcul haute-fidélité pour un paramètre quelconque μ_1 permettant de définir $X_1 = \text{vect}\{u_{\mathcal{N}}(\mu_1)\}$. Ensuite, pour chacune des itérations $n \geqslant 1$ de cet algorithme, étant donnée une base $\{u_{\mathcal{N}}(\mu_1), \ldots, u_{\mathcal{N}}(\mu_n)\}$, la prochaine fonction de base à sélectionner est celle pour laquelle nous approchons le moins bien la sortie du modèle, i.e. celle qui maximise l'estimateur sur l'erreur de réduction par rapport à l'ensemble des paramètres d'apprentissage $\mathcal{P}^{\text{tr}}$. Ainsi, l'algorithme glouton sélectionne un paramètre

$$\mu_{n+1} \in \underset{\mu \in \mathcal{P}^{\text{tr}}}{\text{argmax}}\ \Delta_n(\mu), \tag{1.23}$$

calcule la solution haute-fidélité $u_{\mathcal{N}}(\mu_{n+1})$, et enrichit l'espace réduit pour obtenir $X_{n+1} = \text{vect}\{u_{\mathcal{N}}(\mu_1), \ldots, u_{\mathcal{N}}(\mu_{n+1})\}$. Cette démarche est répétée jusqu'à atteindre une estimation d'erreur $\max_{\mu \in \mathcal{P}^{\text{tr}}} \Delta_N(\mu)$ en-dessous d'une tolérance fixée par l'utilisateur $\epsilon_{\text{RB}} > 0$. En pratique, le critère $\max_{\mu \in \mathcal{P}^{\text{tr}}} \Delta_N(\mu) \leqslant \epsilon_{\text{RB}}$ est atteint pour une valeur de N petite ($N \ll \mathcal{N}$).

1.2.4 Estimation d'erreur *a posteriori*

Un ingrédient essentiel à l'évaluation de la qualité de l'approximation d'un modèle réduit par RBM est un estimateur d'erreur garanti, précis et peu coûteux [53]. De plus, un tel estimateur est indispensable à l'exécution de l'algorithme glouton. Nous considérons l'estimateur d'erreur d'énergie

$$\Delta_N(\mu) = \frac{\|r(\mu; \cdot)\|_{X'_{\mathcal{N}}}}{\alpha_{\text{LB}}(\mu)}, \tag{1.24}$$

où $r(\mu; \cdot)$ est le résidu

$$r(\mu; v) := f(\mu; v) - a(\mu; u_N(\mu), v), \quad \forall v \in X_{\mathcal{N}}, \tag{1.25}$$

et $\alpha_{\text{LB}}(\mu)$ est la borne inférieure de la constante de coercivité $\alpha(\mu)$ définie dans (1.2). Calculons maintenant le résidu (1.25). Le théorème de représentation de Riesz assure l'existence d'une unique fonction $\hat{e} : \mathcal{P} \to X_{\mathcal{N}}$ vérifiant

$$r(\mu; v) = (\hat{e}(\mu), v)_X, \quad \forall v \in X_{\mathcal{N}}. \tag{1.26}$$

Ainsi,

$$\begin{aligned} r(\mu;v) =& f(\mu;v) - a(\mu;u_N(\mu),v), \\ =& f(\mu;v) - a(\mu;\sum_{n=1}^{N} u_n(\mu)\theta_n, v), \\ =& f(\mu;v) - \sum_{n=1}^{N} u_n(\mu) a(\mu;\theta_n,v), \\ =& \sum_{q=1}^{Q^f} \sigma_q^f(\mu) f_q(v) - \sum_{n=1}^{N} u_n(\mu) \sum_{q=1}^{Q^a} \sigma_q^a(\mu) a_q(\theta_n,v), \end{aligned} \tag{1.27}$$

si bien que

$$(\hat{e}(\mu),v)_X = r(\mu;v) \tag{1.28}$$

$$= \sum_{q=1}^{Q^f} \sigma_q^f(\mu) f_q(v) - \sum_{n=1}^{N}\sum_{q=1}^{Q^a} \sigma_q^a(\mu) a_q(\theta_n,v) u_n(\mu)\ . \tag{1.29}$$

Nous obtenons donc

$$\hat{e}(\mu) = \sum_{q=1}^{Q^f} \sigma_q^f(\mu)\mathcal{C}^q + \sum_{n=1}^{N}\sum_{q=1}^{Q^a} \sigma_q^a(\mu)\mathcal{L}_n^q u_n(\mu)\ , \tag{1.30}$$

où $\mathcal{C}^q \in X_\mathcal{N}$ et $\mathcal{L}_n^q \in X_\mathcal{N}$ sont les solutions éléments finis du problème

$$\begin{cases} (\mathcal{C}^q, v)_X = f_q(v) & \forall v \in X_\mathcal{N}, \\ (\mathcal{L}_n^q, v)_X = -a_q(\theta_n, v) & \forall v \in X_\mathcal{N}. \end{cases} \tag{1.31}$$

La norme du résidu est alors calculée par la formule

$$\begin{aligned} ||\hat{e}(\mu)||_X^2 =& \sum_{q=1}^{Q^f}\sum_{q'=1}^{Q^f} \sigma_q^f(\mu)\sigma_{q'}^f(\mu)(\mathcal{C}^q,\mathcal{C}^{q'})_X + 2\sum_{q'=1}^{Q^f}\sum_{q=1}^{Q^a}\sum_{n=1}^{N} \sigma_{q'}^f(\mu)\sigma_q^a(\mu)(\mathcal{C}^{q'},\mathcal{L}_n^q)_X u_n(\mu) \\ &+ \sum_{q=1}^{Q^a}\sum_{n=1}^{N} \sigma_q^a(\mu) \left\{ \sum_{q'=1}^{Q^a}\sum_{n'=1}^{N} \sigma_{q'}^a(\mu)(\mathcal{L}_n^q,\mathcal{L}_{n'}^{q'})_X u_n'(\mu) \right\} u_n(\mu). \end{aligned} \tag{1.32}$$

Durant la phase hors-ligne, sont calculées les quantités indépendantes du paramètre, à savoir $(\mathcal{C}^q,\mathcal{C}^{q'})_X$, $(\mathcal{C}^{q'},\mathcal{L}_n^q)_X$ et $(\mathcal{L}_n^q,\mathcal{L}_{n'}^{q'})_X$ pour $1 \leqslant n,n' \leqslant N$ et $1 \leqslant q,q' \leqslant Q^a$ ou Q^f. Durant la phase en-ligne, seules les quantités $\sigma_q^a(\mu)$ et $\sigma_q^f(\mu)$ pour $1 \leqslant q \leqslant Q$ restent à calculer; l'assemblage du résidu se fait en utilisant les calculs hors-ligne et la formule (1.32).

L'estimation précise d'une borne inférieure α_{LB} pour la constante de coercivité est un point délicat lors de l'estimation d'erreur *a posteriori*. La méthode des contraintes successives (SCM) est une méthode robuste qui a été introduite pour la première fois dans [34] afin de pallier ce problème; d'autres améliorations de cette technique

sont présentées dans [16, 33]. La méthode repose sur une stratégie hors-ligne/en-ligne efficace qui conduit à un problème d'optimisation linéaire. L'inconvénient de la SCM est son coût de calcul. Une stratégie alternative moins onéreuse a été récemment proposée dans [41]. Cette méthode se base sur le calcul d'une approximation interpolatrice de α_{LB}.

1.2.5 La méthode d'interpolation empirique (EIM)

L'objectif de la méthode d'interpolation empirique (EIM) [5, 39] est de construire une approximation affine du type décrit dans (1.12). Comme indiqué plus haut, ce type d'approximation est crucial pour le succès d'une décomposition hors-ligne/en-ligne dans le cadre de la méthode des bases réduites. Dans le contexte le plus général, l'EIM a été introduite pour approcher une fonction continue bivariée $\gamma : \mathcal{P} \times \Omega \to \mathbb{R}$. Cette approximation est décrite par un opérateur d'interpolation γ_M qui interpole la fonction γ en des points d'interpolation $(x_i)_{1\leqslant i\leqslant M}$ dans $\Omega^{\mathrm{tr}} \subsetneq \Omega$ en tant que combinaison linéaire de fonctions $(q_j)_{1\leqslant j\leqslant M}$ avec $q_j : \Omega \to \mathbb{R}$. Ces dernières fonctions ne sont pas des fonctions polynomiales ou trigonométriques mais sont déduites directement à partir de la famille de fonctions $\{\gamma(\mu;\cdot),\ \forall \mu \in \mathcal{P}^{\mathrm{tr}}\}$ par des combinaisons linéaires de M 'échantillons' $\gamma(\mu_1;\cdot), \ldots, \gamma(\mu_M;\cdot)$, où les paramètres $\mu_1, \ldots, \mu_M \in \mathcal{P}^{\mathrm{tr}}$ sont sélectionnés par un algorithme glouton dans un ensemble d'apprentissage $\mathcal{P}^{\mathrm{tr}} \subset \mathcal{P}$ (qui peut être différent de celui introduit plus haut). Ainsi, pour tout paramètre $\mu \in \mathcal{P}$ et tout $x \in \Omega$, l'approximation de rang M, notée γ_M, est donnée par la formule

$$\gamma_M(\mu; x) = \sum_{j=1}^{M} \varphi_j(\mu) q_j(x). \tag{1.33}$$

Le but de la séparation de variables dans (1.33) est de permettre le calcul des fonctions $(q_j)_{1\leqslant j\leqslant M}$ qui sont indépendantes de μ pendant la phase hors-ligne afin de n'avoir que les fonctions $(\varphi_j)_{1\leqslant j\leqslant M}$ qui dépendent de μ à évaluer lors de phase en-ligne; celles-ci vérifient la propriété d'interpolation

$$\gamma_M(\mu, x_i) = \sum_{j=1}^{M} \varphi_j(\mu) q_j(x_i), \quad \forall 1 \leqslant i \leqslant M. \tag{1.34}$$

Pour des fonctions à valeurs réelles v définies sur Ω, on définit $\|v\|_{\ell^\infty(\Omega^{\mathrm{tr}})} := \max_{x\in\Omega^{\mathrm{tr}}} |v(x)|$. Soit un compteur d'itérations $m \geqslant 1$ et une fonction γ_{m-1} définie sur $\mathcal{P}^{\mathrm{tr}} \times \Omega$, avec la convention $\gamma_0 \equiv 0$. Une itération EIM se définit comme suit: premièrement, on définit $\mu_m \in \mathcal{P}^{\mathrm{tr}}$ par

$$\mu_m \in \operatorname*{argmax}_{\mu\in\mathcal{P}^{\mathrm{tr}}} \|\gamma(\mu;\cdot) - \gamma_{m-1}(\mu;\cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}. \tag{1.35}$$

Une fois le paramètre μ_m déterminé, on pose

$$r_m(\cdot) := \gamma(\mu_m;\cdot) - \gamma_{m-1}(\mu_m;\cdot), \quad \text{et} \quad x_m \in \operatorname*{argmax}_{x\in\Omega^{\mathrm{tr}}} |r_m(x)|. \tag{1.36}$$

Est vérifiée ensuite la satisfaction ou pas du critère $|r_m(x_m)| < \epsilon_{\text{EIM}}$ pour une tolérance $\epsilon_{\text{EIM}} > 0$ définie par l'utilisateur. Si ce critère est satisfait, on définit la nouvelle fonction EIM par

$$q_m(\cdot) := \frac{r_m(\cdot)}{r_m(x_m)}, \tag{1.37}$$

et on calcule la dernière ligne de la matrice d'interpolation $\mathbf{B}$ donnée par

$$\mathbf{B}_{mi} := (q_i(x_m)), \quad \forall 1 \leqslant i \leqslant m. \tag{1.38}$$

La construction de l'approximation EIM ainsi décrite satisfait trois propriétés importantes:

(i) Les fonctions de base $(q_j)_{1\leqslant j\leqslant M}$ sont linéairement indépendantes.

(ii) La matrice d'interpolation $\mathbf{B}$ est triangulaire inférieure à diagonale unitaire, donc inversible.

(iii) Pour M grand, l'approximation γ_M tend vers γ en norme infinie $\ell^\infty(\mathcal{P}^{\text{tr}} \times \Omega^{\text{tr}})$.

1.2.6 Brève bibliographie sur la RBM et ses applications

L'idée d'utilisation de bases réduites a été introduite dans des travaux anciens [1, 45]. Forgée sous sa forme actuelle, la RBM a été étudiée dans les travaux fondateurs [37, 46] et deux monographies récentes présentent en détail ses différents aspects [30, 48]. La RBM a été appliquée à un large éventail de problèmes. Par exemple, elle a été utilisée pour la quantification d'incertitudes [6, 15] et pour l'échantillonnage par éléments finis avec comme but l'accélération des calculs spatio-temporels d'un problème d'élasticité [32]. La méthode RBM a également été étendue aux problèmes de contrôle de flux optimaux pour les équations elliptiques non-coercives [43], après avoir permis de générer des sorties en temps réel des équations de Navier–Stokes incompressibles avec des estimateurs d'erreur [53]. Un autre estimateur d'erreur *a posteriori* a été développé dans [13] en utilisant la méthode d'interpolation empirique (EIM) afin de surmonter les erreurs d'arrondi. La méthode EIM est utilisée dans plusieurs contextes et a donné lieu à plusieurs variantes [14, 38]. Concernant l'application à la modélisation thermique, une autre application de la RBM a été la création de solutions fiables en temps réel pour la modélisation d'ailettes thermiques et autres structures complexes [46]. Par ailleurs, des travaux récents présentent une version adaptée à la solution des problèmes décrits par les EDP paraboliques [49]. Bien que dans un premier temps l'algorithme glouton fut utilisé seul pour construire l'espace réduit X_N dans le cas parabolique, il est généralement couplé avec une POD pour améliorer l'efficacité de la méthode. Cette méthode couplée relativement récente a été baptisée POD-greedy dans la littérature [26].

1.3 Objectifs de la thèse

Cette thèse a pour objectif de traiter trois problématiques principales.

1. Il est communément admis dans la littérature que la phase de calcul amont de la méthode des bases réduites, dite phase 'hors-ligne', peut être onéreuse, voire très onéreuse. Ce critère est rarement pris en compte dans les études académiques car la phase hors-ligne est réalisée une fois pour toutes. Cependant, le contexte industriel détaillé plus haut (et bien d'autres) impose des contraintes de ressources et de coût de calcul même pendant cette phase. Notre objectif sera donc de remédier à ce problème en concevant une méthode qui optimise la phase hors-ligne sans pour autant dégrader la qualité de l'approximation réduite finale. Cette problématique n'a été que très peu étudiée jusqu'à présent [17, 18].

2. Les méthodes de réduction de modèles sont pensées pour réduire des équations variationnelles. L'extension de ces travaux aux inégalités variationnelles a fait l'objet de quelques travaux récents [4, 21, 27]. Une application d'intérêt majeur est la mécanique du contact. Néanmoins, les travaux sur ce sujet supposent des hypothèses simplificatrices que sont la linéarité des contraintes et la coïncidence des maillages. Par conséquent, notre second objectif sera d'établir une stratégie de réduction de modèles applicable pour une classe plus générale de problèmes de contact élastique.

3. Le troisième objectif de cette thèse est de valoriser les campagnes d'essai et leurs résultats expérimentaux pour des études par simulation numérique tout en réduisant les moyens et temps de calcul. Nous considérons plus particulièrement une stratégie d'assimilation de données couplée à la réduction de modèles [40] que nous étendrons aux problèmes de thermique non-linéaire instationnaire.

1.4 État de l'art et principaux résultats

Nous présentons finalement un résumé des travaux effectués dans le cadre de cette thèse. Ces travaux sont présentés plus amplement dans les trois chapitres suivants. Dans le premier de ces chapitres, nous nous concentrons sur un problème de thermique non-linéaire instationnaire pour lequel nous concevons une méthode de réduction du temps de calcul hors-ligne afin de diminuer davantage les ressources requises pour la réduction de cette classe de problèmes. Dans le deuxième chapitre, nous étudions ensuite la réduction de modèles pour un problème de mécanique de contact. Finalement, le troisième chapitre est dédié à l'élaboration d'une stratégie d'assimilation de données pour les problèmes instationnaires, dont la classe de problèmes traitée au premier chapitre.

1.4.1 Réduction de modèle en thermique non-linéaire

Nous nous intéressons aux problèmes paraboliques non-linéaires pour lesquels une méthode RBM a été développée dans [23, 24]. Ces travaux ne prennent pas en compte l'éventuel coût de la phase offline. Concernant l'optimisation des coûts hors-ligne de cette classe de problèmes, quelques travaux récents existent dans la littérature. L'idée d'un enrichissement progressif de l'approximation EIM et de l'espace réduit a été récemment proposée dans [17] pour les EDP non-linéaires stationnaires. La construction progressive de l'EIM et de la base réduite a été également abordée dans [18]. Dans cette méthode, le critère d'enrichissement est commun à la fois à l'EIM et à la base réduite, et l'échantillon maximisant un estimateur d'erreur a posteriori est sélectionné pour enrichir les deux bases. Une autre méthode est développée dans [52]; on y propose une construction progressive de l'EIM à l'aide d'approximations basées sur une POD des trajectoires haute-fidélité.

Nous étudions dans le chapitre 2 de nouveaux développements de la méthode des bases réduites et de la méthode d'interpolation empirique (RB-EIM) pour des problèmes paraboliques non-linéaires. Nous développons une nouvelle méthodologie: la méthode progressive RB-EIM (PREIM) pour les problèmes non-linéaires paraboliques. Ici, le but est de réduire le coût hors-ligne final tout en maintenant une bonne approximation RB dans la phase en-ligne. L'idée de base est un enrichissement progressif de l'approximation EIM et de l'espace RB, contrairement à l'approche standard où l'approximation EIM et l'espace RB sont construits séparément. PREIM utilise des calculs haute-fidélité chaque fois que ceux-ci sont disponibles et des calculs RB dans le cas contraire. Ce chapitre correspond à un article paru chez SIAM Journal on Scientific Computing (SISC).

1.4.2 Réduction de modèle pour les problèmes de contact

Dans la littérature, trois articles récents abordent la réduction de modèle pour les inéquations variationnelles. Le premier article [27] étend la méthode RBM standard aux inégalités variationnelles linéaires résolues par une formulation mixte. En ce qui concerne la construction des bases, les bases 'primale' (pour la solution primale) et duale (pour les multiplicateurs de Lagrange) sont directement composées d'échantillons bien choisis. Aucune phase de compression supplémentaire n'est considérée. Dans la méthode dite 'Projection-Based' (PB) [4] qui a été introduite spécifiquement pour résoudre les problèmes de contact avec des contraintes linéaires, les bases primale et duale sont construites différemment. Une base primale est obtenue par POD, comme c'est souvent le cas pour la réduction des égalités variationnelles. Vu que [4] se concentre sur des problèmes de dynamique avec de nombreux instants pour réaliser l'échantillonnage temporel, l'ensemble des échantillons des multiplicateurs de Lagrange s'avère souvent assez important. Sa compression devient alors une préoccupation majeure. L'idée est de construire la base duale réduite en appliquant l'algorithme de factorisation de matrices non-négatives (NMF) [36] à l'ensemble des échantillons de multiplicateurs de Lagrange. La NMF garantit la positivité des vecteurs de base ainsi qu'une dimension de base relativement faible, mais la base

duale qui en résulte est beaucoup moins précise que la base primale. De plus, l'utilisateur ne spécifie pas une tolérance requise mais un nombre de vecteurs de base dominants à conserver. Enfin, [21] étend un autre type de méthodes de réduction de modèles, dites méthodes d'hyper-réduction (HR), aux problèmes de contact avec des contraintes linéaires. L'extension proposée de la méthode HR aux problèmes de contact consiste à conserver peu de vecteurs de la base duale haute-fidélité, car le nombre de nœuds de contact est limité à un domaine d'intégration réduit (RID). Par conséquent, seuls les nœuds de contact du RID sont traités, mais avec une haute fidélité locale. Jusqu'à présent, tous les résultats existants se limitent aux contraintes linéaires.

Nous étudions dans le chapitre 3 de nouveaux développements de la RBM pour les inéquations variationnelles avec contraintes non-linéaires. Nous proposons une méthode de base réduite combinée à la méthode d'interpolation empirique pour traiter la contrainte non-linéaire. Dans ce contexte, une base réduite 'primale' est nécessaire pour la solution primale et une base réduite 'duale' est requise pour les multiplicateurs de Lagrange. Nous proposons de construire cette dernière en utilisant un algorithme hiérarchique qui conserve la non-négativité des vecteurs de la base duale. Cette stratégie de réduction est ensuite appliquée aux problèmes de contact élastique sans frottement pour les maillages non-coïncidents. Ce type de maillages induit une non-linéarité de discrétisation et son traitement est d'autant plus important que le cas de non-coïncidence est très majoritaire dans la pratique. Ces deux dernières conditions issues de la physique du problème n'ont pas encore été abordées, à notre connaissance, dans la littérature.

1.4.3 Assimilation de données

Une méthode d'assimilation de données a été récemment couplée à la réduction de modèle, à savoir la méthode dite 'Parameterized-Background Data-Weak' (PBDW) [40]. Elle a pour but de traiter les divergences pouvant apparaître entre les prédictions d'une simulation numérique basée sur un modèle numérique et des mesures expérimentales. Cette problématique est fréquemment rencontrée dans la pratique, et il est judicieux de prendre en compte les données expérimentales afin d'améliorer les modèles mathématiques. La méthode PBDW est une formulation variationnelle pour les problèmes d'assimilation de données modélisés par des équations aux dérivées partielles. Plusieurs tests numériques indiquent une grande amélioration de la prédiction par le modèle mathématique grâce à la prise en compte des données expérimentales. Une étude approfondie des liens entre la méthode PBDW et d'autres techniques d'assimilation de données est proposée dans [50, 51]. D'autres travaux ont par la suite émergé dans ce même contexte. Dans [3], une technique basée sur la Generalized EIM (GEIM) est proposée pour le placement des capteurs de mesures expérimentales.

Le chapitre 4 a été réalisé à l'occasion d'une collaboration avec le Pr. Patera initiée par une mobilité d'un mois au Massachusetts Institute of Technology (MIT). Il est dédié à une extension de la méthode PBDW. Initialement introduite pour

les problèmes elliptiques, ce chapitre élargit son cadre d'application aux problèmes paraboliques. Pour ce faire, nous exploitons la POD-greedy [26] qui permet d'obtenir des espaces réduits convenables pour tout le transitoire temporel. Ensuite, nous présentons un nouvel algorithme pour la phase hors-ligne de la méthode PBDW. Cet algorithme présente deux intérêts majeurs. D'un côté, il permet d'obtenir des espaces réduits plus représentatifs de la solution recherchée. D'un autre côté, il conduit à une diminution significative du nombre de mesures expérimentales nécessaires lors de la phase en-ligne et en accélère davantage la résolution.

1.4.4 Développements informatiques

Cette thèse a donné lieu à plusieurs développements informatiques. Ceux-ci sont capitalisés dans deux outils phares d'EDF R&D. Une première partie est capitalisée dans la 'forge Pléiade', qui est un dépôt de codes informatiques permettant l'accès et la modification de codes sources à plusieurs acteurs impliqués dans un même projet. Quant à la seconde partie, elle est intégrée à la plateforme MAP (Materials Ageing Platform). Cette plateforme permet l'automatisation du code avec un processus de validation industriel plus avancé, une documentation et un accès à un public d'utilisateurs plus large. Ce processus inclut des tests réguliers de non-régression au fur et à mesure de l'évolution des versions des différents langages de programmation et outils impliqués dans l'élaboration du code. Ainsi, les codes informatiques réalisés pendant cette thèse se divisent comme suit:

- **Dans MAP** sont intégrés les codes de réduction de modèle pour la thermique linéaire pour des problèmes d'échange thermique avec conditions aux limites de Robin. De plus, la partie hors-ligne d'un problème de thermique non-linéaire avec une non-linéarité qui porte sur la conductivité thermique y est finalisée et la partie en-ligne est bien avancée. L'intégration a été initiée pour une paramétrisation qui porte sur la conductivité thermique pour un clapet de régulation de débit (cf Figure 1.4). Ensuite, elle a été élargie à une paramétrisation qui porte sur le coefficient d'échange thermique aussi bien pour le clapet de régulation de débit que pour le robinet RAMA (cf Figure 1.1) dans le cadre du stage de fin d'études d'Alaeddine Jlaiel, encadré par l'auteur du manuscrit et Sébastien Meunier.

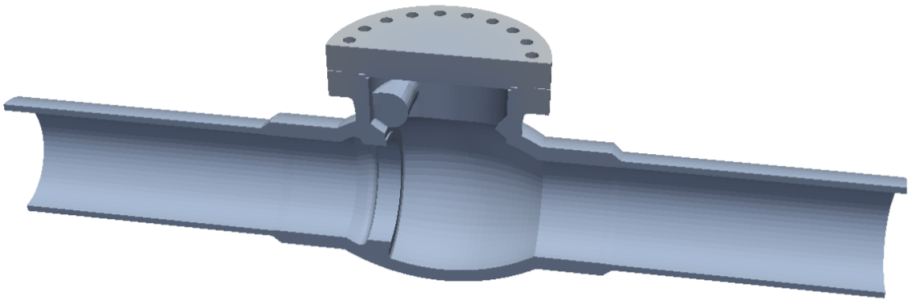

Figure 1.4 – Demi-volume du clapet de régulation de débit.

- **Dans la forge Pléiade** sont déposés les codes restants, à savoir les fichiers sources de réduction de modèles pour la mécanique de contact élastique non-frottant et l'assimilation de données.

CHAPTER 2

A PROGRESSIVE REDUCED BASIS/EMPIRICAL INTERPOLATION METHOD FOR NONLINEAR PARABOLIC PROBLEMS

The material in this chapter has been published in the SIAM Journal on Scientific Computing [8].

Abstract

We investigate new developments of the combined Reduced-Basis and Empirical Interpolation Methods (RB-EIM) for parametrized nonlinear parabolic problems. In many situations, the cost of the EIM in the offline stage turns out to be prohibitive since a significant number of nonlinear time-dependent problems need to be solved using the high-fidelity (or full-order) model. In the present work, we develop a new methodology, the Progressive RB-EIM (PREIM) method for nonlinear parabolic problems. The purpose is to reduce the offline cost while maintaining the accuracy of the RB approximation in the online stage. The key idea is a progressive enrichment of both the EIM approximation and the RB space, in contrast to the standard approach where the EIM approximation and the RB space are built separately. PREIM uses high-fidelity computations whenever available and RB computations otherwise. Another key feature of each PREIM iteration is to select twice the parameter in a greedy fashion, the second selection being made after computing the high-fidelity trajectory for the firstly selected value of the parameter. Numerical examples are presented on nonlinear heat transfer problems.

2.1 Introduction

The Reduced-Basis (RB) method devised in [37, 46] (see also the recent textbooks [30, 48]) is a computationally effective approach to approximate parametrized Partial Differential Equations (PDEs) encountered in many problems in science and engineering. For instance, the RB method is often used in real-time simulations, where a problem needs to be solved very quickly under limited computational resources, or in multi-query simulations, where a problem has to be solved repeatedly for a large number of parameter values. Let $\mathcal{P}$ denote the parameter set. The RB method is split into two stages: (i) an offline stage where a certain number of so-called High-Fidelity (HF) trajectories are computed for a training subset of parameters $\mathcal{P}^{\text{tr}} \subset \mathcal{P}$(typically a finite element space based on a fine mesh); (ii) an online stage for real-time or multi-query simulations where the parameter set $\mathcal{P}$ is explored more extensively. The output of the offline phase includes an approximation space of small dimension spanned by the so-called RB functions. The reduced space then replaces the much larger HF space in the online stage. The crucial point for the computational efficiency of the overall procedure is that computations in the HF space are allowed only in the offline stage.

In the present work, we are interested in nonlinear parabolic problems for which a RB method has been successfully developed in [23, 24]. A key ingredient to treat the nonlinearity so that the online stage avoids HF computations is the Empirical Interpolation Method (EIM) [5, 39]. The EIM provides an approximation of the nonlinear (or non-affine) terms in the PDE. This approximation is built using a greedy algorithm as the sum of M functions, where the dependence on the space variable is separated from the dependence on the parameter (and the time variable for parabolic problems). The integer M is called the rank of the EIM and controls the accuracy of the approximation. Although the EIM is performed during the offline stage of the RB method, its cost can become a critical issue since the EIM can require an important number of HF computations for an accurate approximation of the nonlinearity. The cost of the EIM typically scales with the size of the training set $\mathcal{P}^{\text{tr}}$.

The goal of the present work is to overcome this issue. To this purpose, we devise a new methodology, the Progressive RB-EIM (PREIM) method, which aims at reducing the computational cost of the offline stage while maintaining the accuracy of the RB approximation in the online stage. The key idea is a progressive enrichment of both the EIM approximation and the RB space, in contrast to the standard approach, where the EIM approximation and the RB space are built separately. In PREIM, the number of HF computations is at most M, and it is in general much lower than M in a time-dependent context where the greedy selection of the pair (μ, k) to build the EIM approximation (where μ is the parameter and k refers to the discrete time node) can lead to repeated values of μ for many different values of k. In other words, PREIM can select multiple space fields within the same HF trajectory to build the EIM space functions. In this context, only a modest number of HF trajectories needs to be computed, yielding significant computational savings

with respect to the standard offline stage. PREIM is driven by convergence criteria on the quality of both the EIM and the RB approximation, as in the standard RB-EIM procedure. PREIM is devised in order to have a guaranteed termination, and in the worst-case scenario, the same number of HF trajectories is computed as in the standard RB-EIM algorithm, thus reaching the same level of accuracy for the representation of the nonlinearity and the construction of the RB functions (if this level of accuracy turns out to be insufficient, the parameter training set has to be enlarged as usual in the standard algorithm). In this worst-case scenario, the computational cost of PREIM may be slightly larger than that of the standard algorithm because of the way the intermediate calculations of trajectories are organized in PREIM. However, we expect that in many practical situations, e.g., when the computation of HF trajectories dominates the cost of the progressive construction of the EIM, PREIM can bring computational benefits with respect to the standard approach. These benefits, which are particularly sizeable whenever the nonlinearity can be represented by an EIM approximation of relatively modest rank, are illustrated in this work on three test cases, including one derived from a three-dimensional industrial prototype. Yet, the present study remains heuristic, and a theoretical analysis of the possible computational gains of PREIM can be pursued in future work.

The idea of a progressive enrichment of both the EIM approximation and the RB space has been recently proposed in [17] for stationary nonlinear PDEs, where it is called Simultaneous EIM/RB (SER). Thus, PREIM extends this idea to time-dependent PDEs. In addition, there is an important difference in the greedy algorithms between SER and PREIM. Whereas SER uses only RB computations, PREIM uses HF computations whenever available, both for the greedy selection of the parameters and the time nodes, as well as for the space-dependent functions in the EIM approximation. These aspects are particularly relevant since they improve the accuracy of the EIM approximation. This is illustrated in our numerical experiments on nonlinear parabolic PDEs. The progressive construction of the EIM and the RB has been recently addressed within the Empirical Interpolation Operator Method in [18]. Therein, the enrichment criterion is common to both the EIM and the RB, and the snapshot maximizing an a posteriori error estimator is selected to enrich both bases. Instead, PREIM has dedicated criteria for the quality of the EIM approximation and for the RB approximation. Furthermore, PREIM systematically exploits the knowledge of the HF trajectories whenever available, and an update step is performed in order to confirm the current parameter selection. We also mention the Proper Orthogonal Empirical Interpolation Method from [52], where a progressive construction of the EIM approximation is devised using POD-based approximations of the HF trajectories.

The chapter is organized as follows. In Section 2.2, we introduce the model problem. In Section 2.3, we briefly recall the main ideas of the nonlinear RB method devised in [23, 24], and in Section 2.4, we briefly recall the EIM procedure in the standard offline stage as devised in [5, 39]. The reader familiar with the material can jump directly to Section 2.5, where PREIM is introduced and discussed. Section 2.6 presents numerical results illustrating the performance of PREIM on nonlin-

ear parabolic problems related to heat transfer including a three-dimensional valve prototpye for flow regulation. Finally, Section 2.7 provides a technical complement.

2.2 Model problem

In this section, we present a prototypical example of a nonlinear parabolic PDE. The methodology we propose is illustrated on this model problem but can be extended to other types of parabolic equations. We consider a spatial domain (open, bounded, connected subset) $\Omega \subset \mathbb{R}^d$, $d \geqslant 1$, with a Lipschitz boundary, a finite time interval $I = [0, T]$, with $T > 0$, and a parameter set $\mathcal{P} \subset \mathbb{R}^p$, $p \geqslant 1$, whose elements are generically denoted by $\mu \in \mathcal{P}$. Our goal is to solve the following nonlinear parabolic PDE for many values of the parameter $\mu \in \mathcal{P}$: find $u_\mu : I \times \Omega \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial u_\mu}{\partial t} - \nabla \cdot \big((\kappa_0 + \Gamma(\mu, u_\mu))\nabla u_\mu\big) = f, & \text{in } I \times \Omega, \\ -\big(\kappa_0 + \Gamma(\mu, u_\mu)\big)\dfrac{\partial u_\mu}{\partial n} = \phi_e, & \text{on } I \times \partial\Omega, \\ u_\mu(t = 0, \cdot) = u_0(\cdot), & \text{in } \Omega, \end{cases} \tag{2.1}$$

where $\kappa_0 > 0$ is a fixed positive real number, $\Gamma : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$ is a given nonlinear function, $f : I \times \Omega \to \mathbb{R}$ is the source term, $\phi_e : I \times \partial\Omega \to \mathbb{R}$ is the time-dependent Neumann boundary condition on $\partial\Omega$, and $u_0 : \Omega \to \mathbb{R}$ is the initial condition. For simplicity, we assume without loss of generality that f, ϕ_e, and u_0 are parameter-independent. We assume that $f \in L^2(I; L^2(\Omega))$ and $\phi_e \in L^2(I; L^2(\partial\Omega))$ (this means that $f(t) \in L^2(\Omega)$ and $\phi_e(t) \in L^2(\partial\Omega)$ for (almost every) $t \in I$), and we also assume that $u_0 \in H^1(\Omega)$. We make the standard uniform ellipticity assumption $\beta_1 \leqslant \kappa_0 + \Gamma(\mu, z) \leqslant \beta_2$ with $0 < \beta_1 < \beta_2 < \infty$, for all $(\mu, z) \in \mathcal{P} \times \mathbb{R}$. With the above assumptions, it is reasonable to look for a weak solution $u_\mu \in L^2(I; Y) \cap H^1(I; Y')$.

Remark 2.1 (Initial condition)**.** *For parabolic PDEs, the initial condition is often taken to be in a larger space, e.g., $u_0 \in L^2(\Omega)$. Our assumption that $u_0 \in Y$ is motivated by the RB method, where basis functions in Y are sought as solution snapshots in time and for certain parameter values. In this context, we want to include the possibility to select the initial condition as a RB function.*

Remark 2.2 (Heat transfer)**.** *One important application we have in mind for (4.68) is heat transfer problems. In this context, the PDE can take the slightly more general form*

$$\alpha(u_\mu)\frac{\partial u_\mu}{\partial t} - \nabla \cdot \big((\kappa_0 + \Gamma(\mu, u_\mu))\nabla u_\mu\big) = f, \quad \text{in } I \times \Omega,$$

where $\alpha(u_\mu)$ stands for the mass density times the heat capacity. Moreover, the quantity $(\kappa_0 + \Gamma(\mu, u_\mu))$ represents the thermal conductivity. Note also that $\phi_e > 0$ means that the system is heated.

In practice, one way to solve (4.68) is to use a Y-conforming Finite Element Method [20] to discretize in space and a time-marching scheme to discretize in time.

The Finite Element Method is based on a finite element subspace $X \subsetneq Y$ defined on a discrete nodal subset $\Omega^{\rm tr} \subsetneq \Omega$, where $\mathrm{Card}(\Omega^{\rm tr}) = \mathcal{N}$. To discretize in time, we consider an integer $K \geqslant 1$, we let $0 = t^0 < \cdots < t^K = T$ be $(K+1)$ distinct time nodes over I, and we set $\mathbb{K}^{\rm tr} = \{1, \cdots, K\}$, $\overline{\mathbb{K}}^{\rm tr} = \{0\} \cup \mathbb{K}^{\rm tr}$, $I^{\rm tr} = \{t^k\}_{k \in \overline{\mathbb{K}}^{\rm tr}}$, and $\Delta t^k = t^k - t^{k-1}$ for all $k \in \mathbb{K}^{\rm tr}$. As is customary with the RB method, we assume henceforth that the mesh-size and the time-steps are small enough so that the above space-time discretization method delivers HF approximate trajectories within the desired level of accuracy. These trajectories, which then replace the exact trajectories solving (4.68), are still denoted u_μ for all $\mu \in \mathcal{P}$. Henceforth, we use the convention that the superscript k always indicates a time index; thus, we write $u_\mu^k(\cdot) = u_\mu(t^k, \cdot) \in X$, $f^k(\cdot) = f(t^k, \cdot) \in L^2(\Omega)$, and $\phi_e^k(\cdot) = \phi_e(t^k, \cdot) \in L^2(\partial\Omega)$. Applying a semi-implicit Euler scheme, our goal is, given $u_\mu^0 = u_0 \in X$, to find $(u_\mu^k)_{k \in \mathbb{K}^{\rm tr}} \in X^K$ such that, for all $k \in \mathbb{K}^{\rm tr}$,

$$\forall v \in X, \;\; m(u_\mu^k, v) + \Delta t^k a_0\big(u_\mu^k, v\big) + \Delta t^k n_\Gamma\big(\mu, u_\mu^{k-1}, v\big) = m(u_\mu^{k-1}, v) + \Delta t^k l^k(v), \tag{2.2}$$

with the bilinear forms $m : Y \times Y \to \mathbb{R}$, $a_0 : Y \times Y \to \mathbb{R}$ and the linear forms $l^k : Y \to \mathbb{R}$ such that

$$m(v, w) = \int_\Omega vw, \qquad a_0(v, w) = \kappa_0 \int_\Omega \nabla v \cdot \nabla w, \qquad l^k(v) = \int_\Omega f^k v + \int_{\partial\Omega} \phi_e^k v, \tag{2.3}$$

and the nonlinear form $n_\Gamma : \mathcal{P} \times Y \times Y \to \mathbb{R}$ such that

$$n_\Gamma(\mu, v, w) = \int_\Omega \Gamma(\mu, v) \nabla v \cdot \nabla w, \tag{2.4}$$

for all $\mu \in \mathcal{P}$ and all $v, w \in Y$. In (2.2), the nonlinearity is treated explicitly, whereas the diffusive term is treated implicitly. This choice avoids dealing with a nonlinear solver at each time-step. The computation of derivatives of discrete operators within Newton's method is addressed, e.g., in [18].

2.3 The Reduced-Basis method

In this section, we briefly recall the Reduced-Basis (RB) method for the nonlinear problem (2.2) [24, 23]. Let $\hat{X}_N \subset X$ be a so-called reduced subspace such that $N = \dim(\hat{X}_N) \ll \dim(X) = \mathcal{N}$. Let $(\theta_n)_{1 \leqslant n \leqslant N}$ be a Y-orthonormal basis of $\hat{X}_N$. For all $\mu \in \mathcal{P}$ and $k \in \overline{\mathbb{K}}^{\rm tr}$, the RB solution $\hat{u}_\mu^k \in \hat{X}_N$ that approximates the HF solution $u_\mu^k \in X$ is decomposed as

$$\hat{u}_\mu^k = \sum_{n=1}^N \hat{u}_{\mu,n}^k \theta_n, \tag{2.5}$$

with real numbers $\hat{u}_{\mu,n}^k$ for all $n \in \{1, \ldots, N\}$. Let us introduce the component vector $\hat{\mathbf{u}}_\mu^k := (\hat{u}_{\mu,n}^k)_{1 \leqslant n \leqslant N} \in \mathbb{R}^N$, for all $\mu \in \mathcal{P}$ and $k \in \overline{\mathbb{K}}^{\rm tr}$. Let $\hat{u}^0$ be the Y-orthogonal projection of the initial condition $u_0 \in X$ onto $\hat{X}_N$ with associated component vector

$\hat{\mathbf{u}}^0 \in \mathbb{R}^N$. Replacing $u_\mu^k \in X$ in the weak form (2.2) by the approximation $\hat{u}_\mu^k \in \hat{X}_N$ with associated component vector $\hat{\mathbf{u}}_\mu^k \in \mathbb{R}^N$, and using the test functions $(\theta_p)_{1\leqslant p\leqslant N}$, we obtain the following problem written in algebraic form: Given $\hat{\mathbf{u}}_\mu^0 = \hat{\mathbf{u}}^0 \in \mathbb{R}^N$, find $(\hat{\mathbf{u}}_\mu^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^N)^K$ such that, for all $k \in \mathbb{K}^{\mathrm{tr}}$,

$$(\mathbf{M} + \Delta t^k \mathbf{A}_0)\hat{\mathbf{u}}_\mu^k = \Delta t^k \mathbf{f}^k + \mathbf{M}\hat{\mathbf{u}}_\mu^{k-1} - \Delta t^k \mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1}), \tag{2.6}$$

with the matrices $\mathbf{M}, \mathbf{A}_0 \in \mathbb{R}^{N\times N}$ and the vectors $\mathbf{f}^k \in \mathbb{R}^N$ such that

$$\mathbf{M} = \Big(m(\theta_n,\theta_p)\Big)_{1\leqslant p,n\leqslant N}, \qquad \mathbf{A}_0 = \Big(a_0(\theta_n,\theta_p)\Big)_{1\leqslant p,n\leqslant N}, \qquad \mathbf{f}^k = \big(l^k(\theta_p)\big)_{1\leqslant p\leqslant N}, \tag{2.7}$$

and the vector $\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1}) \in \mathbb{R}^N$ such that

$$\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1}) = \left(\sum_{n=1}^N \hat{u}_{\mu,n}^{k-1} \int_\Omega \Gamma\Big(\mu, \sum_{n'=1}^N \hat{u}_{\mu,n'}^{k-1}\theta_{n'}\Big)\nabla\theta_n \cdot \nabla\theta_p\right)_{1\leqslant p\leqslant N}. \tag{2.8}$$

The difficulty is that the computation of $\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1})$ requires a parameter-dependent reconstruction using the RB functions in order to compute the integral over Ω. To avoid this, we need to build an approximation γ_M of the nonlinear function $\gamma : \mathcal{P} \times \overline{\mathbb{K}}^{\mathrm{tr}} \times \Omega \to \mathbb{R}$ such that

$$\gamma(\mu, k, x) := \Gamma(\mu, u_\mu^k(x)), \tag{2.9}$$

in such a way that the dependence on x is separated from the dependence on (μ, k). More precisely, for some integer $M > 0$, we are looking for an (accurate) approximation $\gamma_M : \mathcal{P} \times \overline{\mathbb{K}}^{\mathrm{tr}} \times \Omega \to \mathbb{R}$ of γ under the separated form

$$\gamma_M(\mu, k, x) := \sum_{j=1}^M \varphi_{\mu,j}^k q_j(x), \tag{2.10}$$

where M is called the rank of the approximation and $\varphi_{\mu,j}^k$ are real numbers that we find by interpolation over a set of M points $\{x_1, \ldots, x_M\}$ in Ω^{tr} by requiring that

$$\gamma_M(\mu, k, x_i) = \gamma(\mu, k, x_i) = \Gamma(\mu, u_\mu^k(x_i)), \quad \forall i \in \{1, \cdots, M\}. \tag{2.11}$$

The interpolation property (2.11) is achieved by setting

$$\varphi_{\mu,j}^k = (\mathbf{B}^{-1}\boldsymbol{\gamma}_\mu^k)_j, \ \forall j \in \{1, \cdots, M\}, \quad \text{where} \quad \boldsymbol{\gamma}_\mu^k := \big(\Gamma(\mu, u_\mu^k(x_i))\big)_{1\leqslant i\leqslant M} \in \mathbb{R}^M, \tag{2.12}$$

and $\mathbf{B} = (q_j(x_i))_{1\leqslant i,j\leqslant M} \in \mathbb{R}^{M\times M}$ must be an invertible matrix. Therefore, (2.10) can be rewritten as follows:

$$\gamma_M(\mu, k, x) = \sum_{j=1}^M (\mathbf{B}^{-1}\boldsymbol{\gamma}_\mu^k)_j q_j(x). \tag{2.13}$$

The points $(x_i)_{1\leqslant i\leqslant M}$ in Ω^{tr} and the functions $(q_j)_{1\leqslant j\leqslant M}$ defined on Ω are determined by the EIM algorithm [5] which is further described in Section 2.4 below.

Let us now describe how we can use the EIM approximation (2.13) to allow for an offline/online decomposition of the computation of the vector $\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1})$ defined in (3.44). Under the (reasonable) assumptions

$$\hat{u}_\mu^k \approx u_\mu^k \quad \text{and} \quad \Gamma(\mu, \hat{u}_\mu^k(x)) \approx \Gamma\big(\mu, u_\mu^k(x)\big), \tag{2.14}$$

we obtain

$$\begin{aligned}\Gamma(\mu, \hat{u}_\mu^k(x)) \approx \Gamma\big(\mu, u_\mu^k(x)\big) = \gamma(\mu, k, x) \approx \gamma_M(\mu, k, x) \\ = \sum_{j=1}^M (\mathbf{B}^{-1}\boldsymbol{\gamma}_\mu^k)_j q_j(x) \approx \sum_{j=1}^M (\mathbf{B}^{-1}\hat{\boldsymbol{\gamma}}_\mu^k)_j q_j(x),\end{aligned} \tag{2.15}$$

with the vector $\hat{\boldsymbol{\gamma}}_\mu^k := (\Gamma(\mu, \hat{u}_\mu^k(x_i)))_{1\leqslant i\leqslant M} \in \mathbb{R}^M$. The problem (2.6) then becomes: Given $\hat{\mathbf{u}}_\mu^0 = \hat{\mathbf{u}}^0 \in \mathbb{R}^N$, find $(\hat{\mathbf{u}}_\mu^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^N)^K$ such that, for all $k \in \mathbb{K}^{\mathrm{tr}}$,

$$(\mathbf{M} + \Delta t^k \mathbf{A}_0)\hat{\mathbf{u}}_\mu^k = \Delta t^k \mathbf{f}^k + \big(\mathbf{M} - \Delta t^k \mathbf{D}_\mu^{k-1}\big)\hat{\mathbf{u}}_\mu^{k-1}, \tag{2.16}$$

with the matrix $\mathbf{D}_\mu^{k-1} \in \mathbb{R}^{N\times N}$ such that

$$\mathbf{D}_\mu^{k-1} = \sum_{j=1}^M \big(\mathbf{B}^{-1}\hat{\boldsymbol{\gamma}}_\mu^{k-1}\big)_j \mathbf{C}^j, \tag{2.17}$$

where

$$\mathbf{C}^j = \left(\int_\Omega q_j \nabla\theta_n \cdot \nabla\theta_p\right)_{1\leqslant p,n\leqslant N} \in \mathbb{R}^{N\times N}, \ \forall 1 \leqslant j \leqslant M. \tag{2.18}$$

The overall computational procedure can now be split into two stages:

(i) An offline stage where one precomputes on the one hand the RB functions $(\theta_n)_{1\leqslant n\leqslant N}$ leading to the vectors $\hat{\mathbf{u}}^0 \in \mathbb{R}^N$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^N)^K$ and the matrices $\mathbf{M}, \mathbf{A}_0 \in \mathbb{R}^{N\times N}$, and on the other hand the EIM points $(x_i)_{1\leqslant i\leqslant M}$ and the functions $(q_j)_{1\leqslant j\leqslant M}$ leading to the matrices $\mathbf{B} \in \mathbb{R}^{M\times M}$ and $\mathbf{C}^j \in \mathbb{R}^{N\times N}$, for all $j \in \{1, \ldots, M\}$. The offline stage is discussed in more detail in Section 2.4.

(ii) An online stage to be performed each time one wishes to compute a new trajectory for a parameter $\mu \in \mathcal{P}$. All that remains to be performed is to compute the vector $\hat{\boldsymbol{\gamma}}_\mu^{k-1} \in \mathbb{R}^M$ and the matrix $\mathbf{D}_\mu^{k-1} \in \mathbb{R}^{N\times N}$ and to solve the N-dimensional linear problem (2.16) for all $k \in \mathbb{K}^{\mathrm{tr}}$. The online stage is summarized in Algorithm 2.1.

2.4 The standard offline stage

There are two tasks to be performed during the offline stage:

(T_1) Build the rank-M EIM approximation (2.10) of the nonlinear function γ defined by (2.9);

Algorithm 2.1 Online stage

Input : μ, $(\theta_n)_{1\leqslant n\leqslant N}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, $\mathbf{A}_0$, $(x_i)_{1\leqslant i\leqslant M}$, $(q_j)_{1\leqslant j\leqslant M}$, and $(\mathbf{C}^j)_{1\leqslant j\leqslant M}$

Set $k=1$ and $\hat{\mathbf{u}}^0_\mu=\hat{\mathbf{u}}^0$
while $k\in\mathbb{K}^{\mathrm{tr}}$ **do**
  Compute $\mathbf{D}^{k-1}_\mu$ using (2.17) and $\hat{\mathbf{u}}^{k-1}_\mu$
  Solve the reduced system (2.16) to obtain $\hat{\mathbf{u}}^k_\mu$
  Set $k=k+1$
end while

Output : $(\hat{\mathbf{u}}^k_\mu)_{k\in\mathbb{K}^{\mathrm{tr}}}$

Algorithm 2.2 Standard EIM

Input : $\mathcal{P}^{\mathrm{tr}}$, $\overline{\mathbb{K}}^{\mathrm{tr}}$, Ω^{tr}, and $\epsilon_{\mathrm{EIM}}>0$

Compute $\mathcal{S}=(u^k_\mu)_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ P `HF trajectories`
Set $m=1$ and $\gamma_0\equiv 0$
Search $(\mu_m,k_m)\in\underset{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}{\operatorname{argmax}}\|\Gamma(\mu,u^k_\mu(\cdot))-\gamma_{m-1}(\mu,k,\cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$
Set $r_m(\cdot):=\Gamma(\mu_m,u^{k_m}_{\mu_m}(\cdot))-\gamma_{m-1}(\mu_m,k_m,\cdot)$ and $x_m\in\underset{x\in\Omega^{\mathrm{tr}}}{\operatorname{argmax}}|r_m(x)|$
while $(|r_m(x_m)|>\epsilon_{\mathrm{EIM}})$ **do**
  Set $q_m:=r_m/r_m(x_m)$ and compute $(\mathbf{B}_{mi})_{1\leqslant i\leqslant m}$ by setting $\mathbf{B}_{mi}:=(q_i(x_m))$
  Set $m=m+1$
  Search $(\mu_m,k_m)\in\underset{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}{\operatorname{argmax}}\|\Gamma(\mu,u^k_\mu(\cdot))-\gamma_{m-1}(\mu,k,\cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$
  Set $r_m(\cdot):=\Gamma(\mu_m,u^{k_m}_{\mu_m}(\cdot))-\gamma_{m-1}(\mu_m,k_m,\cdot)$ and $x_m\in\underset{x\in\Omega^{\mathrm{tr}}}{\operatorname{argmax}}|r_m(x)|$
end while
Set $M:=m-1$

Output : $(x_i)_{1\leqslant i\leqslant M}$ and $(q_j)_{1\leqslant j\leqslant M}$

(T_2) Explore the solution manifold in order to construct a linear subspace $\hat{X}_N\subset X$ of dimension N.

In the standard offline stage, these two tasks are performed independently.

Let us first discuss Task (T_1), i.e., the construction of the rank-M EIM approximation. Recall from Section 2.3 that the goal is to find the interpolation points $(x_i)_{1\leqslant i\leqslant M}$ in $\Omega^{\mathrm{tr}}\subsetneq\Omega$ and the functions $(q_j)_{1\leqslant j\leqslant M}$ with $q_j:\Omega\to\mathbb{R}$. The construction of the EIM approximation additionally uses a training set $\mathcal{P}^{\mathrm{tr}}\subset\mathcal{P}$ for the parameter values; in what follows, we denote by P the cardinality of $\mathcal{P}^{\mathrm{tr}}$. For a real-valued function v defined on Ω^{tr}, we define $\|v\|_{\ell^\infty(\Omega^{\mathrm{tr}})}:=\max_{x\in\Omega^{\mathrm{tr}}}|v(x)|$. Given an iteration counter $m\geqslant 1$ and a function γ_{m-1} defined on $\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}\times\Omega$, with the convention that $\gamma_0\equiv 0$, an EIM iteration consists of the following steps. First, one

defines $(\mu_m, k_m) \in \mathcal{P}^{\mathrm{tr}} \times \overline{\mathbb{K}}^{\mathrm{tr}}$ by

$$(\mu_m, k_m) \in \operatorname*{argmax}_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}} \|\Gamma(\mu, u_\mu^k(\cdot)) - \gamma_{m-1}(\mu, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}, \tag{2.19}$$

where we notice the use of the HF trajectories for all values of the parameter μ in the training set $\mathcal{P}^{\mathrm{tr}}$. Once (μ_m, k_m) has been determined, one sets

$$r_m(\cdot) := \Gamma(\mu_m, u_{\mu_m}^{k_m}(\cdot)) - \gamma_{m-1}(\mu_m, k_m, \cdot), \qquad x_m \in \operatorname*{argmax}_{x\in\Omega^{\mathrm{tr}}} |r_m(x)|, \tag{2.20}$$

and one checks whether $|r_m(x_m)| > \epsilon_{\mathrm{EIM}}$ for some user-defined positive threshold ϵ_{EIM}. If this is the case, one sets

$$q_m(\cdot) := \frac{r_m(\cdot)}{r_m(x_m)}, \tag{2.21}$$

and one computes the new row of the matrix $\mathbf{B}$ by setting $\mathbf{B}_{mi} := (q_i(x_m))$, for all $1 \leqslant i \leqslant m$. The standard EIM procedure is presented in Algorithm 2.2.

Let us now briefly discuss Task (T_2) above, i.e., the construction of a set of RB functions with cardinality N. First, as usual in RB methods, the solution manifold is explored by considering a training set for the parameter values; for simplicity, we consider the same training set $\mathcal{P}^{\mathrm{tr}}$ as for the EIM approximation. This way, one only explores the collection of points $\{u_\mu^k\}_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ in the solution manifold. For this exploration to be informative, the training set $\mathcal{P}^{\mathrm{tr}}$ has to be chosen large enough. The exploration can be driven by means of an a posteriori error estimator (see, e.g., [49]) which allows one to evaluate only N HF trajectories. However, in the present setting where HF trajectories are to be computed for all the parameters in $\mathcal{P}^{\mathrm{tr}}$ when constructing the EIM approximation, it is natural to exploit these computations by means of a Proper Orthogonal Decomposition (POD) [31, 35] to define the RB. This technique is often considered in the literature to build the RB in a time-dependent setting, see, e.g., [26, 30, 48]. In practice, a POD of the whole collection of snapshots may be computationally demanding (or even unfeasible) when a very large number of functions is considered. Thus, we adopt a POD-based progressive construction of the reduced basis in the spirit of the POD-greedy algorithm from [26]. Therein, one additional RB function is picked at a time, whereas here we can pick more than one function. The progressive construction of the RB is presented in Algorithm 2.3, where we have chosen an enumeration of the parameters in $\mathcal{P}^{\mathrm{tr}}$ from 1 to P. The initialization of Algorithm 2.3 is made by computing $(\theta_n)_{1\leqslant n\leqslant N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$ for the trajectory $\mathcal{S}_1$ associated with the parameter μ_1. That is, we select the first N^1 POD modes out of the set $\mathcal{S}_1$ with error threshold ϵ_{POD} (for completeness, this procedure is briefly outlined in Section 2.7). The next steps of the algorithm are performed in an iterative fashion. For each new trajectory, we first subtract its projection on the current RB, and then perform a POD on the projection and merge the result with the current RB. This specific part of the procedure, called UPDATE_RB, is presented in Algorithm 2.4; this part of the procedure is presented separately since it will be re-used later on.

Algorithm 2.3 Progressive RB

Input : $\mathcal{P}^{\mathrm{tr}}$, $\overline{\mathbb{K}}^{\mathrm{tr}}$, and $\epsilon_{\mathrm{POD}} > 0$
1: Compute $\big(\mathcal{S}_p\big)_{1\leqslant p\leqslant P} = \big((u^k_{\mu_p})_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}\big)_{1\leqslant p\leqslant P}$ P `HF trajectories`
2: Compute $(\theta_n)_{1\leqslant n\leqslant N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$
3: Set $p = 1$
4: **while** $p < P$ **do**
5: Set $p = p + 1$
6: Compute $(\theta_n)_{1\leqslant n\leqslant N^p} = \mathrm{UPDATE\_RB}\big((\theta_n)_{1\leqslant n\leqslant N^{p-1}}, \mathcal{S}_p, \epsilon_{\mathrm{POD}}\big)$
7: **end while**
8: Set $N := N^P$
9: Compute $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, and $\mathbf{A}_0$
10: Compute the matrices $(\mathbf{C}^j)_{1\leqslant j\leqslant M}$
Output : $(\theta_n)_{1\leqslant n\leqslant N}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, $\mathbf{A}_0$, and $(\mathbf{C}^j)_{1\leqslant j\leqslant M}$

Remark 2.3 (Threshold ϵ_{POD})**.** *For the initialization (line 2 of Algorithm 2.3), one can use a relative error threshold for ϵ_{POD} (for instance, $\epsilon_{\mathrm{POD}} = 1\%$). Instead, for the iterative loop (line 6 of Algorithm 2.3), the threshold ϵ_{POD} can be set to the greatest singular value that has been truncated at the initialization step.*

Remark 2.4 (Order of EIM and RB)**.** *Algorithms 2.2 and 2.3 can be performed in whatever order. If Algorithm 2.3 is performed first, the computation of the matrices $(\mathbf{C}^j)_{1\leqslant j\leqslant M}$ is postponed to the end of Algorithm 2.2. Moreover, the HF trajectories $(u^k_\mu)_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ appearing in both algorithms are computed only once.*

2.5 The Progressive RB-EIM method (PREIM)

In this section, we first present the main ideas of the PREIM algorithm. Then we describe one important building block called UPDATE_EIM. Finally, using this building block together with the procedure UPDATE_RB from Algorithm 2.4, we present the PREIM algorithm.

2.5.1 Main ideas

PREIM consists in a progressive construction of the EIM approximation and of the RB. The key idea is that, unlike the standard EIM for which HF trajectories are computed for all the parameter values in the training set $\mathcal{P}^{\mathrm{tr}}$ (Algorithm 2.2, line 1), PREIM works with an additional training subset $\mathcal{P}^{\mathrm{HF}}_m \subset \mathcal{P}^{\mathrm{tr}}$ that is enriched progressively with the iteration index m of PREIM. The role of $\mathcal{P}^{\mathrm{HF}}_m$ is to collect the parameter values for which a HF trajectory has already been computed. PREIM is designed such that $\mathrm{Card}(\mathcal{P}^{\mathrm{HF}}_m) \leqslant m$ for all $m \in \{1, \ldots, M\}$. This means that when the final rank-M EIM approximation has been computed, at most M HF trajectories have been evaluated, whence the computational gain with respect to the standard offline stage provided $M \ll P$.

Algorithm 2.4 UPDATE_RB

Input : $\Theta = (\theta_n)_{1\leqslant n\leqslant N}$, $\mathcal{S}$, and $\epsilon_{\text{POD}} > 0$
if $\mathcal{S} = \varnothing$ **then**
  Θ remains unchanged
else
  Define $\tilde{\mathcal{S}} := (u - \Pi_{\text{span}(\Theta)} u)_{u\in\mathcal{S}}$
  Set $\Xi := \text{POD}(\tilde{\mathcal{S}}, \epsilon_{\text{POD}})$
  if $\Xi = \varnothing$ **then**
    Θ remains unchanged
  else
    Set $\Theta := \Theta \cup \Xi$
  end if
end if
Output : Θ

At the iteration $m \geqslant 1$ of PREIM, the trajectories for all $\mu \in \mathcal{P}_m^{\text{HF}}$ are HF trajectories, whereas they are approximated by RB trajectories for all $\mu \in \mathcal{P}^{\text{tr}} \backslash \mathcal{P}_m^{\text{HF}}$. The RB trajectories can be modified at each iteration m of PREIM. This happens whenever a new value of the parameter is selected in the greedy stage of the EIM so that the approximation of the nonlinearity is modified. To reflect this dependency, we add a superscript m to the RB trajectories which are now denoted $(\hat{u}_\mu^{m,k})_{k\in\overline{\mathbb{K}}^{\text{tr}}}$ for all $\mu \in \mathcal{P}^{\text{tr}} \backslash \mathcal{P}_m^{\text{HF}}$. It is convenient to introduce the notation

$$\bar{u}_\mu^{m,k} := \begin{cases} u_\mu^k & \text{if } \mu \in \mathcal{P}_m^{\text{HF}}, \\ \hat{u}_\mu^{m,k} & \text{otherwise,} \end{cases} \tag{2.22}$$

and the nonlinear function

$$\bar{\gamma}^m(\mu, k, x) := \Gamma(\mu, \bar{u}_\mu^{m,k}(x)). \tag{2.23}$$

The goal of every PREIM iteration is twofold:

(i) produce a set of RB functions $(\theta_n^m)_{1\leqslant n\leqslant N^m}$ (the RB functions and their number depend on m);

(ii) produce a rank-m approximation of the nonlinear function $\bar{\gamma}^m$ defined by (2.23) in the form

$$\bar{\gamma}^m_{[\mathcal{P}_m^{\text{HF}}, \mathcal{X}_m, \mathcal{Q}_m]}(\mu, k, x) := \sum_{j=1}^m (\bar{\varphi}^m)_{\mu,j}^k \bar{q}_j(x). \tag{2.24}$$

The notation $\bar{\gamma}^m_{[\mathcal{P}_m^{\text{HF}}, \mathcal{X}_m, \mathcal{Q}_m]}$ in (2.24) indicates the data $[\mathcal{P}_m^{\text{HF}}, \mathcal{X}_m, \mathcal{Q}_m]$ that is used to build the approximation of the nonlinearity. More precisely, this construction uses the PREIM training set $\mathcal{P}_m^{\text{HF}}$, the sequence of interpolation points $\mathcal{X}_m := (\bar{x}_i)_{1\leqslant i\leqslant m}$ in Ω^{tr} (with $\bar{x}_m$ computed at iteration m), and the sequence of functions $\mathcal{Q}_m :=$

$(\bar{q}_j)_{1\leqslant j\leqslant m}$ defined on Ω (with $\bar{q}_m$ computed at iteration m). The progressive construction of these three ingredients is described below. Then, considering the (invertible) lower-triangular matrix $\bar{\mathbf{B}} \in \mathbb{R}^{m\times m}$ whose last row is calculated using $\bar{\mathbf{B}}_{mj} = \bar{q}_j(\bar{x}_m)$ for all $j \in \{1,\dots,m\}$, we compute the real numbers $(\bar{\varphi}^m)^k_{\mu,j}$ in (2.24) from the relations

$$\sum_{j=1}^{m} \bar{\mathbf{B}}_{ij}(\bar{\varphi}^m)^k_{\mu,j} = \bar{\gamma}^m(\mu,k,\bar{x}_i), \qquad \forall i \in \{1,\dots,m\}, \tag{2.25}$$

for all $(\mu,k) \in \mathcal{P}\times\overline{\mathbb{K}}^{\mathrm{tr}}$. All the real numbers $(\bar{\varphi}^m)^k_{\mu,j}$ depend on m since the right-hand side of (2.25) depends on m.

2.5.2 The procedure UPDATE_EIM

Algorithm 2.5 UPDATE_EIM

Input : $(\theta_n)_{1\leqslant n\leqslant N^{m-1}}$, $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$, $\mathcal{X}_{m-1}$, $\mathcal{Q}_{m-1}$, and ϵ_{EIM}

Compute $(\bar{u}^k_\mu)_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ using $(\theta_n)_{1\leqslant n\leqslant N^{m-1}}$

Search $(\mu_m,k_m) \in \underset{(\mu',k')\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}{\operatorname{argmax}} \|\Gamma\big(\mu',\bar{u}^{k'}_{\mu'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu',k',\cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ based on RB/HF

Define $\widetilde{r}_m(\cdot) = \Gamma\big(\mu_m,\bar{u}^{k_m}_{\mu_m}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu_m,k_m,\cdot)$.

if $\mu_m \notin \mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$ **then**

  Compute $\mathcal{S}_{\mathrm{out}} = (u^k_{\mu_m})_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}$ and set $\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}} = \mathcal{P}^{\mathrm{HF}}_{\mathrm{in}} \cup \{\mu_m\}$ one HF trajectory

  Search $(\bar{\mu}_m,\bar{k}_m) \in \underset{(\mu',k')\in\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}{\operatorname{argmax}} \|\Gamma\big(\mu',u^{k'}_{\mu'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu',k',\cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$

else

  Set $\mathcal{S}_{\mathrm{out}} = \varnothing$, $\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}} = \mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$, and $(\bar{\mu}_m,\bar{k}_m) = (\mu_m,k_m)$

end if

Define $\bar{r}_m(\cdot) := \Gamma\big(\bar{\mu}_m,u^{\bar{k}_m}_{\bar{\mu}_m}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\bar{\mu}_m,\bar{k}_m,\cdot)$

if $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})} < \epsilon_{\mathrm{EIM}}$ **then**

  Set incr_rk = FALSE

  Define $r_m(\cdot) = \widetilde{r}_m(\cdot)$ discard the EIM selection

  Set $\mathcal{X}_{\mathrm{out}} = \mathcal{X}_{m-1}$ and $\mathcal{Q}_{\mathrm{out}} = \mathcal{Q}_{m-1}$

else

  Set incr_rk = TRUE

  Define $r_m(\cdot) = \bar{r}_m(\cdot)$

  Set $\mathcal{X}_{\mathrm{out}} = (\mathcal{X}_{m-1},\bar{x}_m)$ and $\mathcal{Q}_{\mathrm{out}} = (\mathcal{Q}_{m-1},\bar{q}_m)$ with $\bar{x}_m$, $\bar{q}_m$ as in Algorithm 2.2 (lines 6 and 9).

end if

Define $\delta^{\mathrm{EIM}}_m = \|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$

Output : incr_rk, $\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}}$, $\mathcal{X}_{\mathrm{out}}$, $\mathcal{Q}_{\mathrm{out}}$, $\mathcal{S}_{\mathrm{out}}$, and δ^{EIM}_m

An essential building block of PREIM is the procedure UPDATE_EIM described in Algorithm 2.5. The input is the RB functions $(\theta_n)_{1\leqslant n\leqslant N^{m-1}}$, the triple

$[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}, \mathcal{X}_{m-1}, \mathcal{Q}_{m-1}]$ describing the current approximation of the nonlinearity (the choice for the indices will be made clearer in the next section, and is not important at this stage), and the threshold ϵ_{EIM}. The output is the flag incr_rk which indicates whether or not the rank of the EIM approximation has been increased, and if incr_rk = TRUE, the additional output is the triple $[\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}}, \mathcal{X}_{\mathrm{out}}, \mathcal{Q}_{\mathrm{out}}]$ to devise the new EIM approximation, possibly a new HF trajectory $\mathcal{S}_{\mathrm{out}}$, and a measure δ^{EIM}_m on the EIM error.

First (see line 2), one selects a new pair $(\mu_m, k_m) \in \mathcal{P}^{\mathrm{tr}} \times \overline{\mathbb{K}}^{\mathrm{tr}}$ in a greedy fashion as follows:

$$(\mu_m, k_m) \in \operatorname*{argmax}_{(\mu', k') \in \mathcal{P}^{\mathrm{tr}} \times \overline{\mathbb{K}}^{\mathrm{tr}}} \|\Gamma\big(\mu', \bar{u}^{k'}_{\mu'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}, \mathcal{X}_{m-1}, \mathcal{Q}_{m-1}]}(\mu', k', \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}. \tag{2.26}$$

In (2.26), $\bar{u}^{k'}_{\mu'}$ is defined as in (2.22) using the set $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$. Therefore, the selection criterion (2.26) exploits the knowledge of the HF trajectory for all the parameter values in $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$, and otherwise uses a RB trajectory. This is an important difference with respect to the standard offline stage. There are now two possibilities: (i) either μ_m is already in $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$; then, no new HF trajectory is computed and we set $\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}} := \mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$ (line 8); (ii) or μ_m is not in $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$; then we compute a new HF trajectory for the parameter μ_m and we set $\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}} := \mathcal{P}^{\mathrm{HF}}_{\mathrm{in}} \cup \{\mu_m\}$ (line 5). Our numerical experiments reported in Section 2.6 below will show that at many iterations of PREIM, the pair (μ_m, k_m) selected in (2.26) differs from the previously selected pair by the time index and not by the parameter value; this means that for many PREIM iterations, no additional HF computation is performed. In case of non-uniqueness of the maximizer in (2.26), one selects, if possible, a trajectory for which the parameter is not already in the set $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$ in order to trigger a computation of a new HF trajectory.

An additional feature of PREIM is that, whenever a new HF trajectory is actually computed, one can either confirm or update the selected pair (μ_m, k_m) using the following HF-based re-selection criterion (see line 6):

$$(\bar{\mu}_m, \bar{k}_m) \in \operatorname*{argmax}_{(\mu', k') \in \mathcal{P}^{\mathrm{HF}}_{\mathrm{out}} \times \overline{\mathbb{K}}^{\mathrm{tr}}} \|\Gamma\big(\mu', u^{k'}_{\mu'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}, \mathcal{X}_{m-1}, \mathcal{Q}_{m-1}]}(\mu', k', \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}. \tag{2.27}$$

We notice that this re-selection criterion only handles HF trajectories since the parameter values are in $\mathcal{P}^{\mathrm{HF}}_{\mathrm{out}}$. Moreover, (2.27) only requires to probe the values for μ_m, since the values for the other parameters, which are in $\mathcal{P}^{\mathrm{HF}}_{\mathrm{in}}$, have already been evaluated in (2.26). Finally, to prevent division by small quantities, the value of the residual $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ is checked in line 11. If this value is too small, the pair $(\bar{\mu}_m, \bar{k}_m)$ is rejected and the rank of the EIM approximation is not increased.

2.5.3 The PREIM algorithm

We are now ready to describe the PREIM procedure, see Algorithm 2.6. PREIM is an iterative method that builds progressively the RB and the EIM approximation. The iteration is controlled by three tolerances: $\epsilon_{\mathrm{POD}} > 0$ which is used in the

progressive increment of the RB, $\epsilon_{\text{EIM}} > 0$ which is used to check the quality of the EIM approximation, and $\epsilon_{\text{RB}} > 0$ which is used to check the quality of the RB. The termination criterion involves the quality of both the EIM and the RB approximations, see line 7. Note that this is the same criterion as in the standard RB-EIM approach.

Within each PREIM iteration, the two previously-described procedures UPDATE_EIM and UPDATE_RB are called. First, one attempts to improve the EIM approximation (line 9). If this is successful (i.e., if incr_rk = TRUE), the RB is updated by using the possibly new HF trajectory $\mathcal{S}_m$ (line 21). Otherwise (i.e., if incr_rk = FALSE), the RB is possibly updated (line 12) and a new improvement of the EIM is attempted (line 13). In general, the RB is improved because a new HF trajectory has been computed. Whenever this is not the case, a new HF trajectory is anyway computed in line 18 (cf. Remark 2.5 below). The choice of this new HF trajectory can be driven by a standard greedy algorithm based on the use of a classical a posteriori error estimator. More precisely, for a given reduced basis $(\theta_n)_{1\leqslant n\leqslant N}$ and given sets of training points $\mathcal{X}$ and functions $\mathcal{Q}$ used for the current EIM approximation of the nonlinearity, the associated a posteriori error estimator for a given value of the parameter $\mu \in \mathcal{P}$ is denoted by $\Delta^{\mathcal{X},\mathcal{Q}}_{(\theta_n)_{1\leqslant n\leqslant N}}(\mu)$. Finally, we observe that the reduced matrices and vectors in line 22 of Algorithm 2.6 need to be updated since these quantities depend on the RB functions which can change at every iteration.

Remark 2.5 (Worst-case scenario)**.** *The worst-case scenario is that in which PREIM would compute as many trajectories as the standard EIM. In this situation, the RB space would be identical to that of the standard RB-EIM. Regarding the approximation of the nonlinearity, if PREIM is carried on until $M = M_{\max} := P \times K$, the resulting rank-$M$ approximation would be exact for all the parameters in $\mathcal{P}^{\text{tr}}$. Hence, as ϵ_{RB}, ϵ_{POD}, and ϵ_{EIM} tend to zero, RB-EIM and PREIM produce the same approximations at termination (recall that termination is guaranteed for both algorithms).*

Let us now discuss the initialization of PREIM. In line 2, one can choose an initial PREIM training set $\mathcal{P}_1^{\text{HF}}$ composed of a single parameter, as is often the case with greedy algorithms. Although the nonlinearity may not be well-described initially, one can expect that the description will improve progressively. Still, to allow for more robustness in the initialization, one can consider an initial set $\mathcal{P}_1^{\text{HF}}$ composed of several parameters. One can then compute the HF trajectories for all $\mu \in \mathcal{P}_1^{\text{HF}}$ and compress them using the POD procedure with threshold ϵ_{POD} (if $\mathcal{P}_1^{\text{HF}}$ contains more than one value, a progressive version is used). Finally, one selects

$$(\mu_1, k_1) \in \operatorname*{argmax}_{(\mu',k')\in\mathcal{P}_1^{\text{HF}}\times\overline{\mathbb{K}}^{\text{tr}}} \|\Gamma\big(\mu', u^{k'}_{\mu'}(\cdot)\big)\|_{\ell^\infty(\Omega^{\text{tr}})}, \tag{2.28}$$

one defines $r_1(\cdot) = \Gamma(\mu_1, u^{k_1}_{\mu_1}(\cdot))$ and computes $\mathcal{X}_1 = (\bar{x}_1)$, $\mathcal{Q}_1 = (\bar{q}_1)$ (as in the standard EIM procedure), and one sets $\delta_1^{\text{EIM}} = \|r_1\|_{\ell^\infty(\Omega^{\text{tr}})}$. Let us finally point out that a good initialization of PREIM can favor its early termination. For instance,

one can try to select the first parameter as one for which the nonlinearity has a sizeable effect.

Remark 2.6 (PREIM-NR and U-SER variants)**.** *We can consider two variants in the procedure UPDATE_EIM (Algorithm 2.5) and therefore in PREIM. A first variant consists in skipping the re-selection step in line 6 of Algorithm 2.5. This variant, which we call PREIM-NR (for 'no re-selection'), will be tested numerically in the next section in order to highlight the actual benefits brought by the re-selection. A second variant is to replace* $\bar{u}_\mu^{m,k}$ *with* $\hat{u}_\mu^{m,k}$ *in lines 1 and 2 of Algorithm 2.5, and to skip the re-selection step in line 6. We call this variant U-SER since it can be viewed as an extension of SER [17] to the unsteady setting. The crucial difference between PREIM-NR and U-SER is that U-SER uses RB trajectories to compute the space-dependent functions in the EIM approximation, whereas PREIM-NR uses HF trajectories.*

Algorithm 2.6 Progressive RB-EIM (PREIM)

Input : $\mathcal{P}^{\mathrm{tr}}$, $\overline{\mathbb{K}}^{\mathrm{tr}}$, Ω^{tr}, $\epsilon_{\mathrm{POD}} > 0$, $\epsilon_{\mathrm{EIM}} > 0$, and $\epsilon_{\mathrm{RB}} > 0$

Set $m = 1$

Choose $\mathcal{P}_1^{\mathrm{HF}} \subsetneq \mathcal{P}^{\mathrm{tr}}$ of cardinality J and compute $\mathcal{S}_1 = (u_\mu^k)_{(\mu,k)\in\mathcal{P}_1^{\mathrm{HF}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ $J \geqslant 1$ `HF trajectories`

Compute $(\theta_n^1)_{1\leqslant n\leqslant N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$.

Compute $\hat{\mathbf{u}}^0 \in \mathbb{R}^{N^1}$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^{N^1})^K$, $\mathbf{M} \in \mathbb{R}^{N^1\times N^1}$, and $\mathbf{A}_0 \in \mathbb{R}^{N^1\times N^1}$

Compute $(\mathcal{X}_1, \mathcal{Q}_1, \delta_1^{\mathrm{EIM}}) = \mathrm{INIT\_EIM}(\mathcal{P}_1^{\mathrm{HF}})$ and $\mathbf{C}^1 \in \mathbb{R}^{N^1\times N^1}$

Compute $\delta_1^{\mathrm{RB}} = \max_{\mu\in\mathcal{P}^{\mathrm{tr}}} \Delta_1(\mu)$

while $(\delta_m^{\mathrm{EIM}} > \epsilon_{\mathrm{EIM}}$ or $\delta_m^{\mathrm{RB}} > \epsilon_{\mathrm{RB}})$ **do**

  Set $m = m + 1$ and $\mathcal{P}_{\mathrm{in}}^{\mathrm{HF}} := \mathcal{P}_{m-1}^{\mathrm{HF}}$

  (incr_rk, $\mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$, $\mathcal{X}_{\mathrm{out}}$, $\mathcal{Q}_{\mathrm{out}}$, $\mathcal{S}_{\mathrm{out}}$ δ_m^{EIM}) = UPDATE_EIM $((\theta_n^{m-1})_{1\leqslant n\leqslant N^{m-1}}, \mathcal{P}_{\mathrm{in}}^{\mathrm{HF}}, \mathcal{X}_{m-1}, \mathcal{Q}_{m-1}, \epsilon_{\mathrm{EIM}})$

  while incr_rk = FALSE **do**

    $\mathcal{P}_{\mathrm{in}}^{\mathrm{HF}} = \mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$

    $(\theta_n^{m-1})_{1\leqslant n\leqslant N^{m-1}} = \mathrm{UPDATE\_RB}\big((\theta_n^{m-1})_{1\leqslant n\leqslant N^{m-1}}, \mathcal{S}_{\mathrm{out}}, \epsilon_{\mathrm{POD}}\big)$

    (incr_rk, $\mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$, $\mathcal{X}_{\mathrm{out}}$, $\mathcal{Q}_{\mathrm{out}}$, $\mathcal{S}_{\mathrm{out}}$, δ_m^{EIM}) = UPDATE_EIM $((\theta_n^{m-1})_{1\leqslant n\leqslant N^{m-1}}, \mathcal{P}_{\mathrm{in}}^{\mathrm{HF}}, \mathcal{X}_{m-1}, \mathcal{Q}_{m-1}, \epsilon_{\mathrm{EIM}})$

    if incr_rk = TRUE **then**

      Step to line 20

    end if

    Compute $\mu_m \in \underset{\mu\in\mathcal{P}^{\mathrm{tr}}}{\mathrm{argmax}}\, \Delta^{\mathcal{X}_{\mathrm{out}},\mathcal{Q}_{\mathrm{out}}}_{(\theta_n^{m-1})_{1\leqslant n\leqslant N^{m-1}}}(\mu)$

    Compute $\mathcal{S}_{\mathrm{out}} = (u_{\mu_m}^k)_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}$ `one HF trajectory`

  end while

  Set $\mathcal{P}_m^{\mathrm{HF}} = \mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$, $\mathcal{S}_m = \mathcal{S}_{\mathrm{out}}$, $\mathcal{X}_m = \mathcal{X}_{\mathrm{out}}$, and $\mathcal{Q}_m = \mathcal{Q}_{\mathrm{out}}$

  Compute $(\theta_n^m)_{1\leqslant n\leqslant N^m} = \mathrm{UPDATE\_RB}\big((\theta_n^{m-1})_{1\leqslant n\leqslant N^{m-1}}, \mathcal{S}_m, \epsilon_{\mathrm{POD}}\big)$

  Update $\hat{\mathbf{u}}^0 \in \mathbb{R}^{N^m}$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^{N^m})^K$, and the matrices $\mathbf{M}$, $\mathbf{A}_0$, $(\mathbf{C}^j)_{1\leqslant j\leqslant m}$ in $\mathbb{R}^{N^m\times N^m}$

  Compute $\delta_m^{\mathrm{RB}} = \max_{\mu\in\mathcal{P}^{\mathrm{tr}}} \Delta^{\mathcal{X}_m,\mathcal{Q}_m}_{(\theta_n^m)_{1\leqslant n\leqslant N^m}}(\mu)$

end while

Set $M := m$

Output : $(\theta_n)_{1\leqslant n\leqslant N^M}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, $\mathbf{A}_0$, $\mathcal{X}_M$, $\mathcal{Q}_M$, and $(\mathbf{C}^j)_{1\leqslant j\leqslant M}$

2.6 Numerical results

In this section, we illustrate the above developments on three test cases related to transient heat transfer problems. The first two test cases use the idealized 2D geometry of a perforated square plate; the first test case involves a nonlinearity on the solution, whereas the second test case considers a nonlinearity on its partial derivatives. The third test case is based on the three-dimensional geometry of an industrial valve prototype used for flow regulation in nuclear reactor operation, while we use the same type of nonlinearity as in the first test case. Our goal is to illustrate the computational performance of PREIM and to compare it to the standard EIM approach described in Section 2.4 and to the variants PREIM-NR and U-SER described in Remark 2.6. HF trajectories are computed using a Finite Element subspace $X \subset Y = H^1(\Omega)$ consisting of continuous, piecewise affine functions. The HF computations use the industrial software `code_aster` [19] for the first test case, `FreeFem++` [28] for the second test case, and a combination of the industrial software `Salomé` and `FreeFem++` for the third test case. The reduced-order modeling algorithms have been developed in Python. In all the test cases, the dominant error component turns out to be the one resulting from the approximation of the nonlinearity, rather than the one resulting from the RB. For this reason, PREIM has been run using only the stopping criterion $\delta_m^{\mathrm{EIM}} > \epsilon_{\mathrm{EIM}}$ in line 7 of Algorithm 2.6.

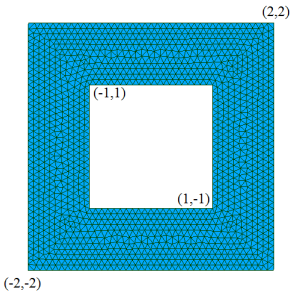

Figure 2.1 – Test cases (a) and (b): The computational domain is a perforated plate.

2.6.1 Test case (a): Nonlinearity on the solution

We consider a two-dimensional setting based on the perforated plate illustrated in Figure 2.1 with $\Omega = (-2,2)^2 \backslash [-1,1]^2 \subset \mathbb{R}^2$. We consider the nonlinear parabolic problem (4.68) with the nonlinear function $\Gamma(\mu, v) := \sin\left(\frac{2\pi\mu}{20}\left(\frac{v-u_0}{u_m-u_0}\right)^2\right)$, with $u_0 = 293$ K (20 °C) and $u_m = 323$ K (50 °C). We define $\kappa_0 = 1.05$ m^2·K^{-2}·s^{-1} and $\phi_e = 3$ K·m·s^{-1} (these units result from our normalization by the density times the heat capacity). For space discretization, we use a mesh containing $\mathcal{N} = 1438$ nodes (see Figure 2.1). Regarding time discretization, we consider the time interval $I = [0,5]$, the set of discrete times nodes $\mathbb{K}^{\mathrm{tr}} = \{1, \cdots, 50\}$, and a constant time step $\Delta t^k = 0.1$ s for all $k \in \mathbb{K}^{\mathrm{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1,20]$, the training set $\mathcal{P}^{\mathrm{tr}} = \{1, \cdots, 20\}$, and we use the larger set $\{0.25i \mid 0 \leqslant i \leqslant 80\}$ to verify our numerical results. In Figure 2.2, we show the HF temperature profiles over

the perforated plate at two different times and for two different parameter values. We can see that, as the simulation time increases, the temperature is, overall, higher for larger values of the parameter μ than for smaller values. Also, for larger values of μ, the temperature variation tends to be less uniform over the plate than for smaller values of μ.

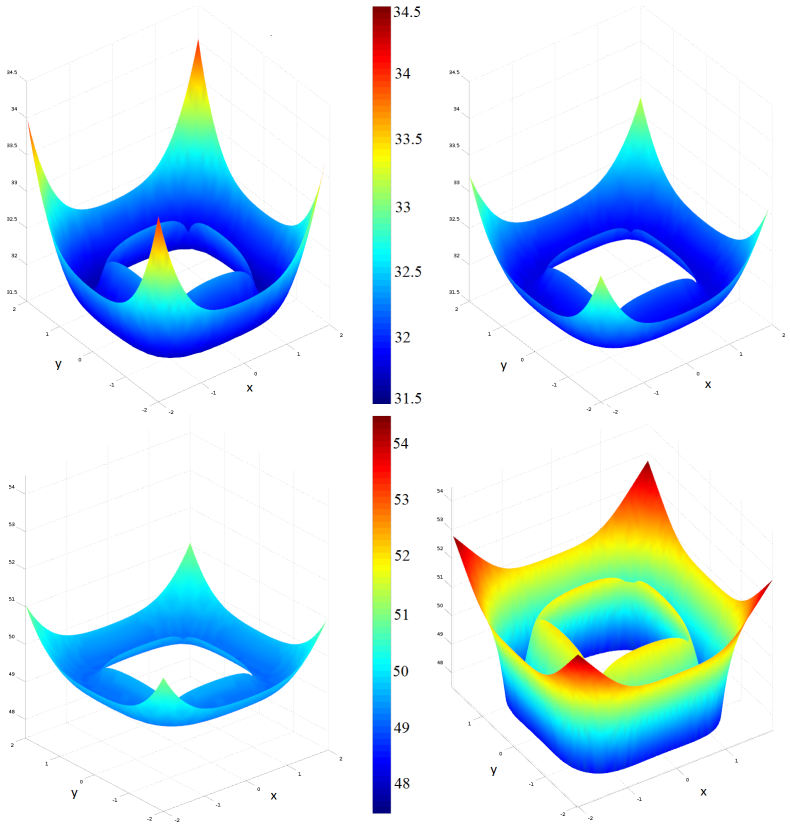

Figure 2.2 – Test case (a): HF solutions for the parameter values $\mu = 1$ (left) and $\mu = 18$ (right) at $t = 2$ s (top) and $t = 5$ s (bottom).

m	1	2	6	14	15	20	25
$\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$	2.0	8.1E−1	1.1E−1	5.2E−3	2.6E−3	1.1E−3	1.6E−4

Table 2.1 – Test case (a): Evolution of the standard EIM error. m is the rank of the EIM approximation.

During the standard offline stage, we perform $P = 20$ HF computations. Knowing that $K = 50$, the set $\mathcal{S}$ (Algorithm 2.2, line 1) contains 1020 fields, each consisting of $\mathcal{N} = 1438$ nodal values. Applying the POD in a progressive manner (see Algorithm 2.3 with the parameters enumerated using increasing values) based on the H^1-norm and a truncation threshold $\epsilon_{\mathrm{POD}} = 10^{-3}$, we obtain $N = 6$ RB functions. Afterwards, we perform the standard EIM algorithm whose convergence is reported in Table 2.1 for selected values of the rank of the EIM approximation. For $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$, the final rank of the EIM approximation is $M = 8$, whereas for $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$, the final rank of the EIM approximation is $M = 15$.

m		1	2	3	4	5	6	7	8	9	10	11	12	13
PREIM	$\bar{\mu}$	1	20	20	20	20	20	20	20	16	20	20	18	20
	μ	1	20	20	20	20	18	20	20	16	20	20	18	20
	$\bar{k}$	50	45	48	50	43	42	39	46	50	49	33	50	47

Table 2.2 – Test case (a): Selected parameters and time nodes in PREIM. The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

We now investigate PREIM, which we first run with thresholds $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-2}$. Table 2.2 shows the selected parameters and discrete time nodes at each stage of PREIM. We can make two important observations from this table. First, after 13 iterations, PREIM has only selected four different parameter values, and has therefore computed only four HF trajectories (the iterations for which a new parameter value is selected are indicated in gray in Table 2.2). In the other 9 out of the 13 iterations, a different time snapshot of an already existing HF trajectory has been selected. Second, by comparing the lines in Table 2.2 related to μ and $\bar{\mu}$, we can see that a parameter re-selection happened at iteration $m = 7$.

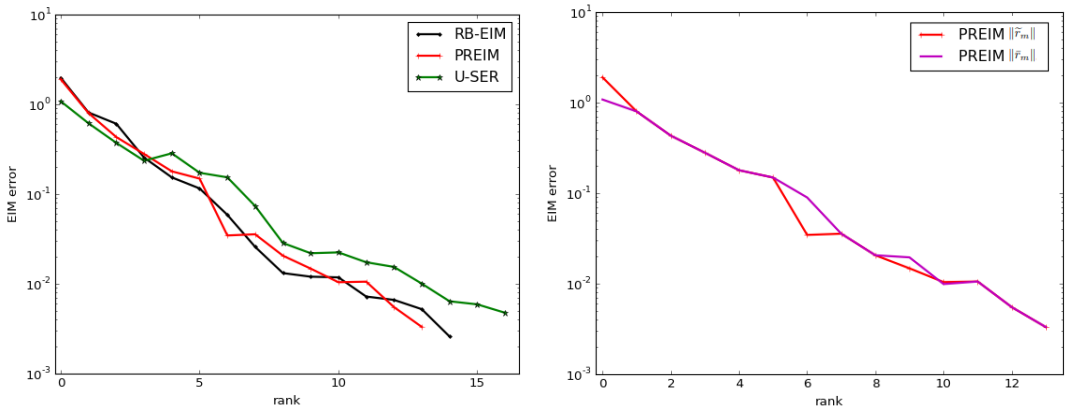

Figure 2.3 – Test case (a): EIM approximation error as a function of m for $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-2}$. Left: Errors for the standard RB-EIM procedure, PREIM, and U-SER. Right: Errors $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ for PREIM.

The left panel of Figure 2.3 displays the error on the approximation of the nonlinear function Γ for the standard RB-EIM procedure and for PREIM as a function of the iteration number m (the additional curve concerning U-SER will be commented afterwards), i.e., we plot $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ (line 3 of Algorithm 2.5) and $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ (line 10 of Algorithm 2.5) as a function of m, see (2.24). We can see that the quality of the approximation of the nonlinearity is almost the same for PREIM as for the standard RB-EIM procedure; yet, the former achieves this accuracy by computing 20% of the HF trajectories computed by the latter (4 instead of 20 HF trajectories). The right panel of Figure 2.3 shows the values of $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ as a function of m. The two quantities differ when the parameter μ_m in line 2 of Algorithm 2.5 is not in the set $\mathcal{P}^{\text{HF}}_{m-1}$ so that $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is computed using a RB approximation, whereas

$\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ results from a HF trajectory. Discarding the initialization, this happens for $m \in \{6, 9, 10\}$. The fact that $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ take rather close values indicates that the RB provides an accurate approximation of the HF trajectory.

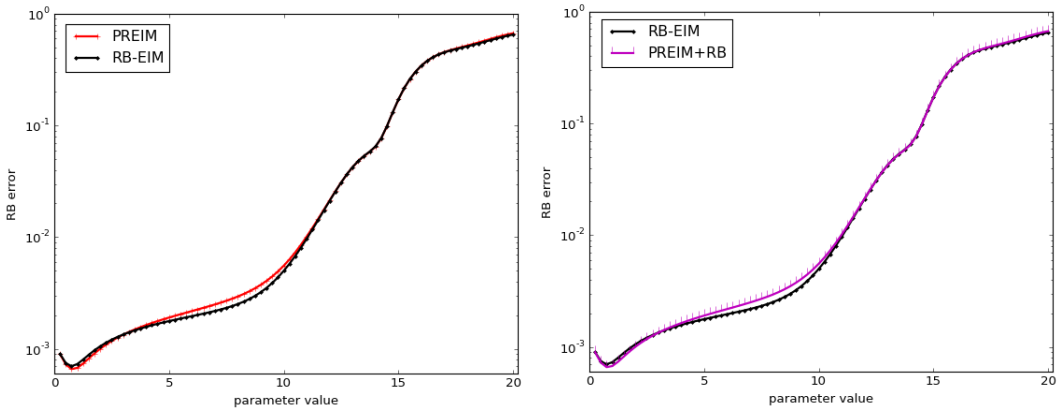

Figure 2.4 – Test case (a): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 10^{-3}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$.

The left panel of Figure 2.4 compares the space-time errors (measured using the ℓ^2-norm in time and the H^1-norm in space) on the trajectories produced by the standard RB-EIM and the PREIM procedures for the whole parameter range. The error is generically denoted $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$. We observe an excellent agreement over the whole parameter range. In the right panel of Figure 2.4, we also consider the space-time errors on the trajectories produced using the approximation of the nonlinearity resulting from PREIM with the RB resulting from the standard algorithm. We do not observe any significant change with respect to the left panel, which indicates that the dominant error component is that associated with the approximation of the nonlinearity. We consider the tighter couple of thresholds $\epsilon_{\mathrm{POD}} = 10^{-5}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$ in Figure 2.5. Here, we can observe some differences in the errors produced by the standard RB-EIM and PREIM procedures, although both errors remain comparable and reach similar maximum values over the parameter

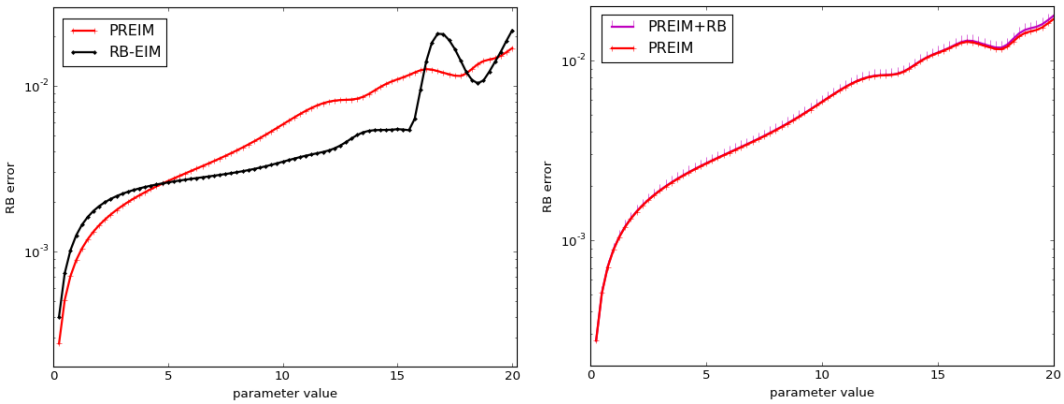

Figure 2.5 – Test case (a): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 10^{-5}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$.

range. While the standard procedure is slightly more accurate for most parameter values, the conclusion is reversed for some other values. Moreover, the curves on the right panel of Figure 2.5 corroborate the fact that once again, the dominant error component is that associated with the approximation of the nonlinearity.

m		1	2	3	4	5	6	7	8	9	10	11	12	13
U-SER	μ	1	20	20	20	16	20	19	20	20	19	17	20	19
	k	50	49	50	46	42	49	44	39	50	49	48	47	50
PREIM-NR	μ	1	20	20	20	20	16	20	20	20	20	20	17	19
	k	50	47	50	46	42	49	48	46	39	50	45	50	50

Table 2.3 – Test case (a): Selected parameters and time nodes in U-SER and PREIM-NR. The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

Let us further explore the PREIM algorithm by comparing it to its variants U-SER and PREIM-NR introduced in Remark 2.6. Table 2.3 reports the selected parameters and time nodes in U-SER and PREIM-NR (compare with Table 2.2 for PREIM). Both U-SER and PREIM-NR need to compute five HF trajectories, which is only 25% of those needed with the standard RB-EIM procedure, but this is still one more HF trajectory than with PREIM. One difference between U-SER and PREIM-NR is that new parameters are selected earlier with U-SER. Interestingly, after 13 iterations, U-SER and PREIM-NR have selected the same five parameters. Another interesting observation is that U-SER actually selects the same couple (μ, k) twice (this happens for $m = 2$ and $m = 6$); this can be interpreted by observing that owing to the improvement of the RB using HF trajectories between iterations $m = 2$ and $m = 6$, the algorithm detects the need to improve the approximation of the nonlinearity by using the same pair (μ, k). The same observation can be made for PREIM-NR (this happens for $m = 4$ and $m = 8$). We emphasize that re-selecting the same pair (μ, k) is not possible within PREIM since the selection is based on HF trajectories. The left panel of Figure 2.3 displays the error on the approximation of the nonlinear function Γ obtained with U-SER and compares it to the error obtained with the standard RB-EIM and PREIM procedures that were already discussed. The U-SER error is evaluated as $\sup_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}} \|\Gamma(\mu, \hat{u}^k_\mu(\cdot)) - \overline{\gamma}^m_{[\mathcal{P}^{\mathrm{HF}}_m, \mathcal{X}_m, \mathcal{Q}_m]}(\mu, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$. We observe that the approximation of the nonlinearity is somewhat less sharp with U-SER than with PREIM. Figure 2.6 reports the space-time errors (measured using the ℓ^2-norm in time and the H^1-norm in space) on the trajectories produced by PREIM and U-SER for the whole parameter range. We observe that the U-SER error is always larger, sometimes up to a factor of five, but for the larger parameter values which produce the larger errors, the quality of the results produced by PREIM and U-SER remains comparable.

Finally, we provide an assessment of the runtimes in Table 2.4. We can see that for the standard RB-EIM procedure, the computation of the HF trajectories dominates the cost of the offline phase. For both PREIM and U-SER, the cost of these HF computations is substantially reduced. At the same time, the cost of the

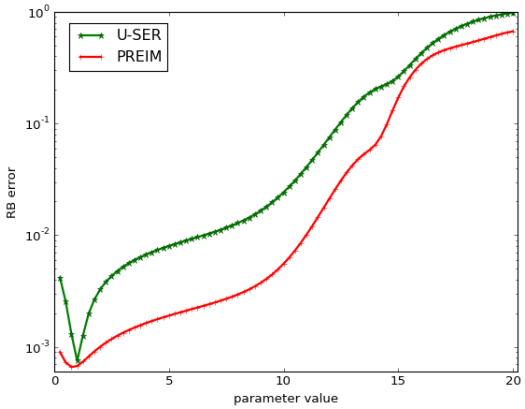

Figure 2.6 – Test case (a): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 10^{-3}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$.

	RB-EIM	PREIM	U-SER
HF computations	99%	20.0%	25.0%
greedy runtime	1%	1.5%	2.3%
Total runtime	100%	21.5%	26.3%

Table 2.4 – Test case (a): Runtime measurements.

greedy algorithm (which includes the construction of the EIM and of the RB) is increased by 50% with respect to the standard RB-EIM procedure. However, the impact on the total runtime is marginal.

2.6.2 Test case (b): Nonlinearity on the partial derivatives

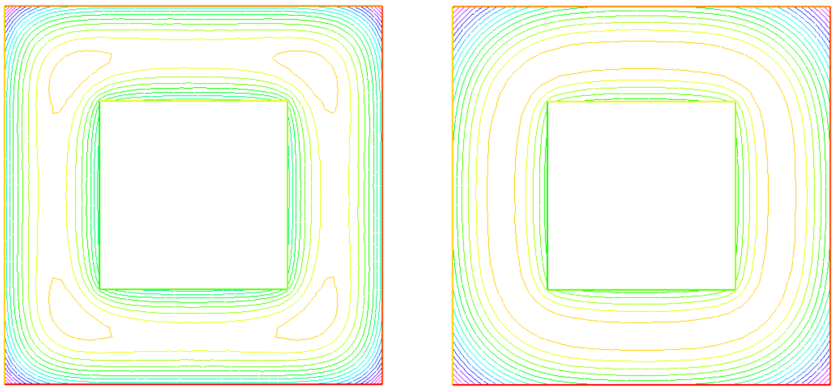

Figure 2.7 – Test case (b): HF solutions for the parameter values $\mu = 1$ at $t = 1$ s (left, values from 20.2 to 22.1) and at $t = 2.5$ s (right, values from 34.5 to 37.3).

We consider the nonlinear parabolic problem (4.68) with the nonlinear function $\Gamma(\mu, z) := \sin\left(\omega\mu\left((\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2\right)\right)^2$, where $\omega = 6.25 \cdot 10^{-3}$. We define $u_0 = 293$ K (20 ^{o}C), $\kappa_0 = 1$ m^2·K^{-2}·s^{-1} and $\phi_e = 3$ K·m·s^{-1} (these units result from our nor-

malization by the density times the heat capacity). For the space discretization, we use a mesh containing $\mathcal{N} = 1429$ nodes. Regarding time discretization, we consider the time interval $I = [0, 2.5]$, the set of discrete times nodes $\mathbb{K}^{\mathrm{tr}} = \{1, \dots, 50\}$, and a constant time step $\Delta t^k = 0.05$ s for all $k \in \mathbb{K}^{\mathrm{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1, 40]$ and the training set $\mathcal{P}^{\mathrm{tr}} = \{1, \dots, 40\}$. In Figure 2.7, we show the temperature isovalues over the perforated plate at two different times for $\mu = 1$. We can observe different boundary layers depending on the time (the same observation can be made by varying the parameter value).

p	1	2	3	8	20	23	24	26	32	33	36	37	39	40
RB size	3	4	5	6	7	8	9	10	11	12	13	14	15	15

Table 2.5 – Test case (b): Size of the reduced basis in the standard algorithm with $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$.

m	2	10	13	20	30	36	37	79	96	144
$\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$	1.6	1.3	9.7E−1	4.7E−1	1.7E−1	1.2E−1	8.0E−2	9.1E−3	4.6E−3	9.4E−4

Table 2.6 – Test case (b): Evolution of the standard EIM error. m is the rank of the EIM approximation and $\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ is the residual norm in (2.20).

During the standard offline stage, we perform $P = 40$ HF computations. Knowing that $K = 50$, the set $\mathcal{S}$ (Algorithm 2.2, line 1) contains 2040 fields, each consisting of $\mathcal{N} = 1429$ nodal values. Applying Algorithm 2.3 based on the H^1-norm, a truncation threshold $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$, and parameters enumerated with increasing values, we obtain $N = 15$ RB functions. Table 2.5 shows the dimension of the RB space as a function of the enumeration index p. Table 2.6 shows the evolution of the error on the nonlinearity within the standard EIM for selected values of the rank of the EIM approximation. The fact that the nonlinearity depends on the partial derivatives of the solution challenges the EIM; indeed, the error decay is not as fast as in the previous test case. This observation is corroborated by the fact that the functions $(q_j)_{1 \leqslant j \leqslant M}$ all look quite different (not shown for brevity).

We now investigate the performance of PREIM, which we run with thresholds $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$ and either $\epsilon_{\mathrm{EIM}} = 10^{-1}$ or $\epsilon_{\mathrm{EIM}} = 10^{-3}$. Table 2.7 shows the selected parameters and time nodes at each iteration. For $\epsilon_{\mathrm{EIM}} = 10^{-1}$, PREIM performs 9 iterations, and three parameters are selected for HF computations, whereas for $\epsilon_{\mathrm{EIM}} = 10^{-3}$, PREIM performs 11 further iterations and six more HF computations to reach the requested threshold. Moreover, the evolution of the size of the reduced basis within PREIM is shown in Table 2.8 for selected values of the rank of the EIM approximation. As can be noticed, the approximation of the nonlinearity requires more computational effort than that of the solution manifold.

Figure 2.8 shows the decrease of the EIM approximation error on the nonlinearity for PREIM with $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-3}$. We observe that each time a new HF trajectory is computed, i.e., whenever the quantities $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$

m	1	2	3	4	5	6	7	8	9
$\bar{\mu}$	21	8	21	8	21	21	21	8	9
μ	21	8	21	8	21	21	21	8	9
$\bar{k}$	2	5	3	2	50	4	49	3	4

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\bar{\mu}$	21	21	21	8	21	21	21	8	9	21	9	21	9	9	9	6	21	21	40	40
μ	21	8	21	8	21	21	21	8	9	21	9	7	6	9	9	5	4	3	40	40
$\bar{k}$	2	5	3	2	50	4	49	3	4	10	50	25	49	5	10	4	6	9	15	40

Table 2.7 – Test case (b): Selected parameters and time nodes in PREIM for $\epsilon_{\text{EIM}} = 10^{-1}$ (top) and $\epsilon_{\text{EIM}} = 10^{-3}$ (bottom). The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

m	1	2	9	17	18	20
RB size	5	6	6	7	9	9

Table 2.8 – Test case (b): Size of RB generated within PREIM for $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$; for $\epsilon_{\text{EIM}} = 10^{-1}$, one stops at $m = 9$, and for $\epsilon_{\text{EIM}} = 10^{-3}$, one stops at $m = 20$.

differ, the difference is actually rather small, thereby confirming the already accurate approximation of the nonlinearity by the RB solutions in PREIM. The left panel of Figure 2.9 illustrates the space-time errors (measured in the $\ell^2(I^{\text{tr}}; H^1(\Omega^{\text{tr}}))$-norm) on the trajectories produced by the standard RB-EIM and the PREIM procedures for the whole parameter range. We observe that for lower parameter values, PREIM delivers somewhat less accurate results, whereas the conclusion is reversed for higher parameter values. Altogether, both errors stay within comparable upper bounds. The right panel of Figure 2.9 shows that the error component associated with the approximation of the nonlinearity is still the dominant one, except for the parameter range $[1, 5]$, where the RB from the standard algorithm improves the error. Incidentally, we observe that these smaller values of the parameter were not selected within PREIM for approximating the nonlinearity. Finally, Figure 2.10 shows the same results for the tighter thresholds $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\text{EIM}} = 10^{-4}$. Here, 14 HF computations and 100 PREIM iterations were needed. We can see that the PREIM error closely matches that of the standard RB-EIM procedure.

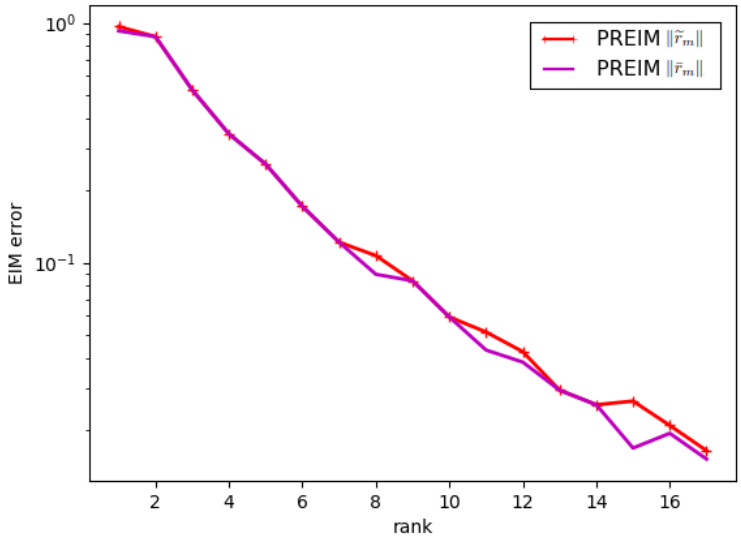

Figure 2.8 – Test case (b): EIM approximation errors $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ as a function of m for PREIM with $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-3}$.

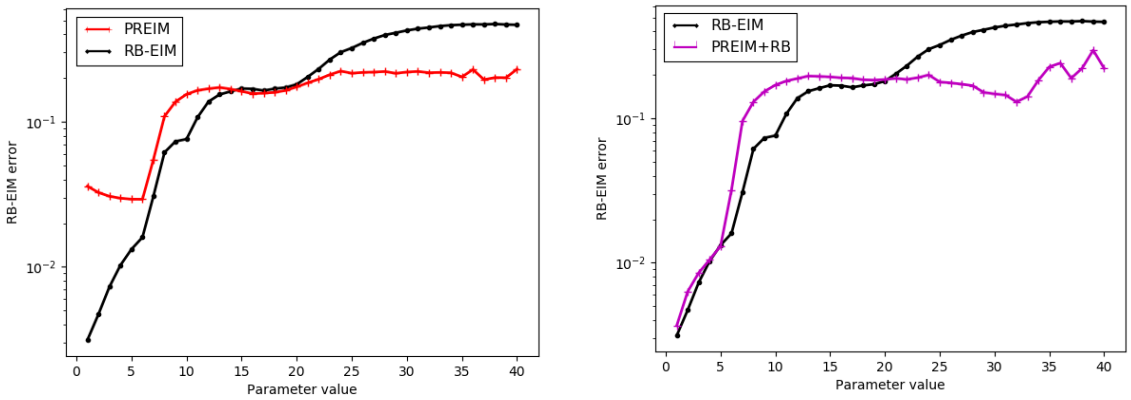

Figure 2.9 – Test case (b): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-3}$.

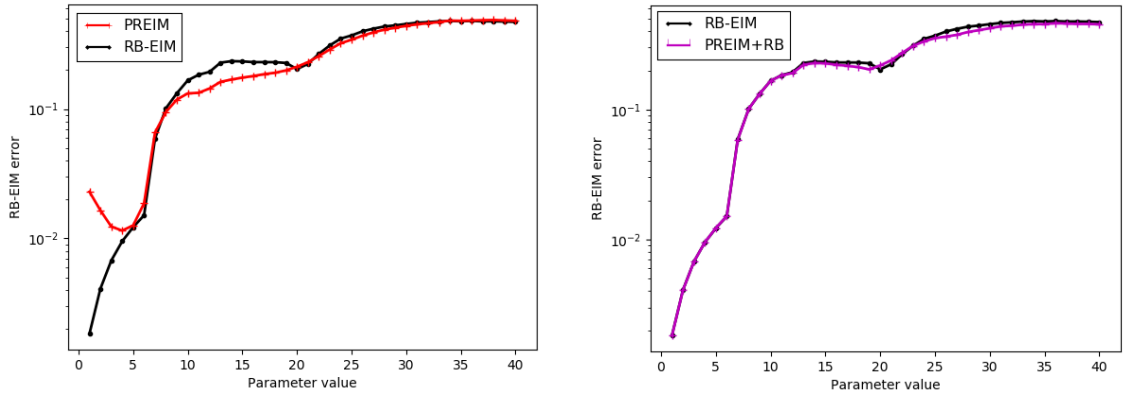

Figure 2.10 – Test case (b): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 2.5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-4}$.

2.6.3 Test case (c): 3D industrial valve prototype

Here, we present a three-dimensional test case whose geometry is based on a flow regulation valve used in nuclear reactor operation. We consider the nonlinear parabolic problem (4.68) with the nonlinear function $\Gamma(\mu, v) := \sin\left(\frac{\pi\mu}{20}\left(\frac{v-u_0}{u_m-u_0}\right)^2\right)$, with $u_0 = 293$ K (20 ^{o}C) and $u_m = 303$ K (30 ^{o}C). We define $\kappa_0 = 1.05$ m^2·K^{-2}·s^{-1} and $\phi_e = 3$ K·m·s^{-1}. For space discretization, we use a mesh containing $\mathcal{N} = 46,018$ nodes (see Figure 2.11). Regarding time discretization, we consider the time inter-

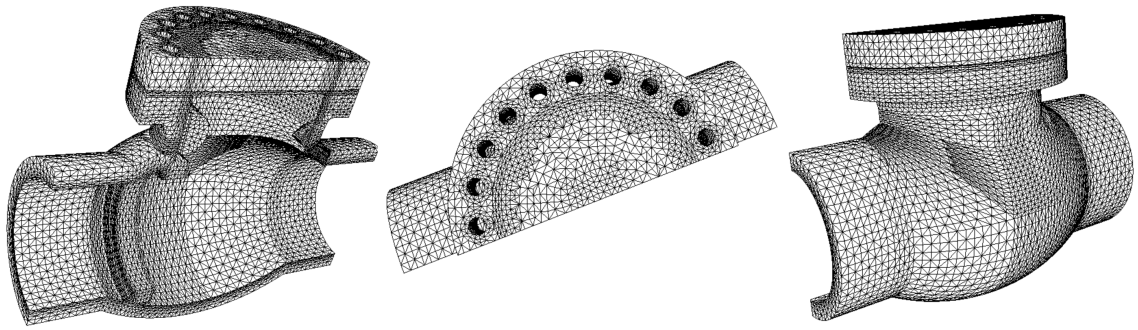

Figure 2.11 – Test case (c): half-section of an industrial flow-regulation valve (Acknowledgment to VELAN SAS for providing the design data necessary for the mesh).

val $I = [0, 1.5]$, the set of discrete times nodes $\mathbb{K}^{\text{tr}} = \{1, \cdots, 30\}$, and a constant time step $\Delta t^k = 0.05$ s for all $k \in \mathbb{K}^{\text{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1, 20]$ and the training set $\mathcal{P}^{\text{tr}} = \{1, \cdots, 20\}$. During the standard offline

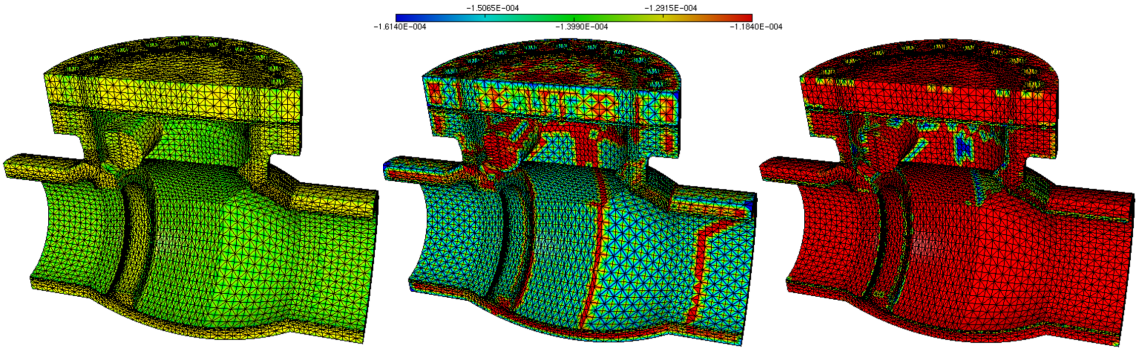

Figure 2.12 – Test case (c): 2$^{\text{nd}}$, 3$^{\text{rd}}$, and 4$^{\text{th}}$ progressive-POD modes.

m	1	2	3	5	7	8	11
$\|r_m\|_{\ell^\infty(\Omega^{\text{tr}})}$	1.0	5.5E−1	2.7E−1	5.2E−2	1.1E−2	7.5E−3	1.5E−3

Table 2.9 – Test case (c): Evolution of the standard EIM error. m is the rank of the EIM approximation and $\|r_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is the residual norm in (2.20).

stage, we perform $P = 20$ HF computations. Knowing that $K = 31$, the set $\mathcal{S}$ (Algorithm 2.2, line 1) contains 620 fields, each consisting of $\mathcal{N} = 46,018$ nodal values.

Applying the POD in a progressive manner based on the H^1-norm and a relative truncation threshold $\epsilon_{\text{POD}} = 10^{-3}$ defined as suggested in Remark 2.3, we obtain $N = 4$ RB functions. Figure 2.12 shows three POD modes. Afterwards, we perform the standard EIM algorithm whose convergence is reported in Table 2.9.

m		1	2	3	4	5	6	7	8	9	10
PREIM	$\bar{\mu}$	20	20	10	20	20	20	8	20	20	13
	k	30	14	30	21	5	26	22	3	9	30
	μ	20	11	10	20	19	9	8	12	13	7
	k	31	31	31	21	31	31	31	31	31	31

Table 2.10 – Test case (c): Selected parameters and time nodes in PREIM for $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. The gray cells correspond to a new parameter selection.

We now investigate PREIM, which we first run with thresholds $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. Table 2.10 shows the selected parameters and discrete time nodes at each stage of PREIM. Out of ten iterations, we can see that a parameter re-selection happened at four iterations. For some of the remaining iterations, the selected time nodes have been changed at the re-selction step.

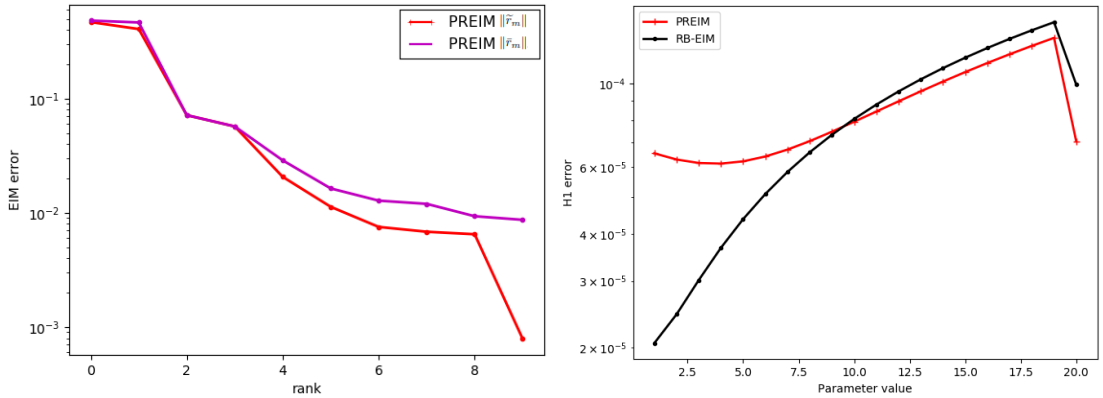

Figure 2.13 – Test case (c): Left: EIM approximation errors $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ as a function of m for PREIM with $\epsilon_{\text{POD}} = 10^{-4}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. Right: RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\text{tr}};H^1(\Omega^{\text{tr}}))} / \|u_\mu\|_{\ell^2(I^{\text{tr}};H^1(\Omega^{\text{tr}}))}$ for $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$.

The left panel of Figure 2.13 shows the decrease of the EIM approximation error on the nonlinearity within PREIM for $\epsilon_{\text{POD}} = 10^{-4}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. Overall, the difference between $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is rather small except for $m = 9$ where Table 2.10 shows that the first selection has been discarded. The right panel of Figure 2.13 illustrates the space-time errors (measured in the relative $\ell^2(I^{\text{tr}}; H^1(\Omega^{\text{tr}}))$-norm) on the trajectories produced by the standard RB-EIM and the PREIM procedures for the whole parameter range; here the tolerances are set to $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. We observe that for higher parameter values, PREIM delivers somewhat more accurate results, whereas the conclusion is reversed for lower parameter

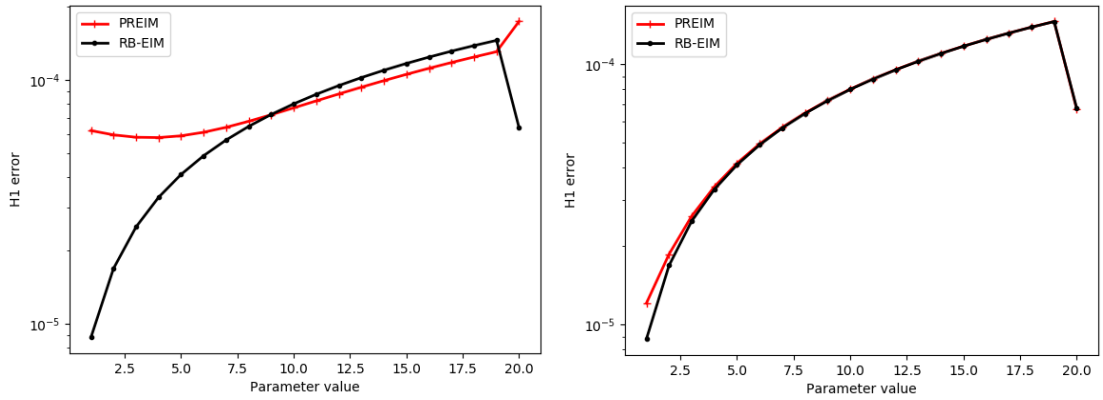

Figure 2.14 – Test case (c): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))} / \|u_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$.
Left: $\epsilon_{\mathrm{POD}} = 10^{-4}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-1}$. Right: $\epsilon_{\mathrm{POD}} = 10^{-4}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$.

values. Altogether, both errors stay within comparable upper bounds. Figure 2.14 displays the space-time errors for the tighter tolerance $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-4}$ that delivers more accurate reluts as expected. Moreover, tightening ϵ_{EIM} makes both the RB-EIM and PREIM errors overlap. Thus, one can infer that the dominant error in this test case is rather the RB error; whence the numerous HF computations that need to be performed, as seen in Table 2.10. Still, for such tight tolerances and with quasi-identical output errors (cf. right panel of Figure 2.14), PREIM makes less than half of the HF computations incurred in the standard RB-EIM.

$\epsilon_{\mathrm{EIM}} = 5.10^{-1}$	RB-EIM	PREIM
HF computations	99.8%	10.0%
greedy runtime	0.2%	0.6%
Total runtime	100%	10.6%

$\epsilon_{\mathrm{EIM}} = 5.10^{-2}$	RB-EIM	PREIM
HF computations	99.6%	20.0%
greedy runtime	0.4%	2.4%
Total runtime	100%	22.4%

Table 2.11 – Test case (c): Runtime measurements with $\epsilon_{\mathrm{POD}} = 10^{-4}$. Left: $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-1}$. Right: $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$.

Finally, we provide an assessment of the runtimes in Table 2.11. One can notice that the greedy procedure accounts for a slightly greater percentage of the offline stage in PREIM compared to the standard RB-EIM. This is mainly due to the additional intermediate calculations in PREIM. However, as previously shown, the dominant part of the offline stage are the HF computations; this illustrates again the relevance of using PREIM.

2.7 Technical complement : Proper Orthogonal Decomposition

The goal of this complement is to briefly describe the procedure associated with the notation

$$(\theta_1, \dots, \theta_N) = \mathrm{POD}(\mathcal{S}, \epsilon_{\mathrm{POD}}), \tag{2.29}$$

which is used in Algorithms 2.3, 2.4, and 2.6, where $\mathcal{S} = (v_1, \dots, v_R)$ is composed of $R \geqslant 1$ functions in the space X and ϵ_{POD} is a user-prescribed tolerance. For simplicity, we adopt an algebraic description, and we refer the reader to [25] for further insight. Let $(\varrho_1, \dots, \varrho_{\mathcal{N}})$ be a basis of X, where $\dim(X) = \mathcal{N}$. For a function $w \in X$, we denote by $\mathbf{w} := (w_j)_{1 \leqslant j \leqslant \mathcal{N}}$ its coordinate vector in $\mathbb{R}^{\mathcal{N}}$, so that $w = \sum_{j=1}^{\mathcal{N}} w_j \varrho_j$. The algebraic counterpart of (2.29) is that we are given R vectors forming the rectangular matrix $\mathbf{S} := (\mathbf{v}_1, \dots, \mathbf{v}_R) \in \mathbb{R}^{\mathcal{N} \times R}$, and we are looking for N vectors forming the rectangular matrix $\boldsymbol{\Theta} := (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) \in \mathbb{R}^{\mathcal{N} \times N}$. The vectors $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N)$ are to be orthonormal with respect to the Gram matrix of the inner product in X. In the present setting, we consider the Gram matrix $\mathbf{C}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ such that

$$\mathbf{C}^{\mathcal{N}} = \Big(m(\varrho_n, \varrho_p) + \eta a_0(\varrho_n, \varrho_p) \Big)_{1 \leqslant p,n \leqslant \mathcal{N}}, \tag{2.30}$$

where $\eta > 0$ is a user-prescribed weight and the bilinear forms m and a_0 are defined in (2.3). Thus, we want to have $\boldsymbol{\theta}_n^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} \boldsymbol{\theta}_p = \delta_{n,p}$, the Kronecker delta, for all $n, p \in \{1, \dots, N\}$.

Let us set $\mathbf{T} := (\mathbf{C}^{\mathcal{N}})^{\frac{1}{2}} \mathbf{S} \in \mathbb{R}^{\mathcal{N} \times R}$ and consider the integer $D = \min(\mathcal{N}, R)$ (in most situations, we have $D = R$ and $D \ll \mathcal{N}$). Computing the Singular Value Decomposition [44] of the matrix $\mathbf{T}$, we obtain the real numbers $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_D \geqslant 0$, the orthonormal family of column vectors $(\boldsymbol{\xi}_n)_{1 \leqslant n \leqslant D} \in (\mathbb{R}^{\mathcal{N}})^D$ (so that $\boldsymbol{\xi}_n^{\mathrm{T}} \boldsymbol{\xi}_p = \delta_{p,n}$) and the orthonormal family of column vectors $(\hat{\boldsymbol{\psi}}_n)_{1 \leqslant n \leqslant D} \in (\mathbb{R}^R)^D$ (so that $\hat{\boldsymbol{\psi}}_n^{\mathrm{T}} \hat{\boldsymbol{\psi}}_p = \delta_{p,n}$), and we have

$$\mathbf{T} = \sum_{n=1}^{D} \sigma_n \boldsymbol{\xi}_n \hat{\boldsymbol{\psi}}_n^{\mathrm{T}}. \tag{2.31}$$

From (2.31), it follows that $\mathbf{T} \hat{\boldsymbol{\psi}}_n = \sigma_n \boldsymbol{\xi}_n$ and $\mathbf{T}^{\mathrm{T}} \boldsymbol{\xi}_n = \sigma_n \hat{\boldsymbol{\psi}}_n$ for all $n \in \{1, \dots, D\}$. The vectors we are looking for are then given by $\boldsymbol{\theta}_n := (\mathbf{C}^{\mathcal{N}})^{-\frac{1}{2}} \boldsymbol{\xi}_n$ for all $n \in \{1, \dots, N\}$ with $N := \max\{1 \leqslant n \leqslant D \mid \sigma_n \geqslant \epsilon_{\mathrm{POD}}\}$. It is well-known that the N-dimensional space spanned by the vectors $(\boldsymbol{\theta}_n)_{1 \leqslant n \leqslant N}$ minimizes the quantity $\sum_{r=1}^{R} \inf_{\mathbf{z} \in \mathbf{Z}_N} (\mathbf{v}_r - \mathbf{z})^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} (\mathbf{v}_r - \mathbf{z})$ among all the N-dimensional subspaces $\mathbf{Z}_N$ of $\mathbb{R}^{\mathcal{N}}$. Moreover, we have $\|v - \Pi_{Z_N} v\|_X \leqslant \sigma_{N+1} \|v\|_X$, for all $v \in \mathcal{S}$.

In practice, when $D = R$, we can avoid the computation of the matrix $(\mathbf{C}^{\mathcal{N}})^{\frac{1}{2}}$ and of its inverse by considering the matrix of smaller dimension $\mathbf{T}^{\mathrm{T}} \mathbf{T} = \mathbf{S}^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} \mathbf{S} \in \mathbb{R}^{R \times R}$. Solving for the eigenvalues of $\mathbf{T}^{\mathrm{T}} \mathbf{T}$, we obtain the vectors $\hat{\boldsymbol{\psi}}_n$ with associated eigenvalues σ_n^2 since we have $\mathbf{T}^{\mathrm{T}} \mathbf{T} \hat{\boldsymbol{\psi}}_n = \sigma_n \mathbf{T}^{\mathrm{T}} \boldsymbol{\xi}_n = \sigma_n^2 \hat{\boldsymbol{\psi}}_n$. Then, the vectors $(\boldsymbol{\theta}_n)_{1 \leqslant n \leqslant N}$ are obtained as $\boldsymbol{\theta}_n = \left(\mathbf{C}^{\mathcal{N}}\right)^{-\frac{1}{2}} \boldsymbol{\xi}_n = \frac{1}{\sigma_n} \left(\mathbf{C}^{\mathcal{N}}\right)^{-\frac{1}{2}} \mathbf{T} \hat{\boldsymbol{\psi}}_n = \frac{1}{\sigma_n} \mathbf{S} \hat{\boldsymbol{\psi}}_n$.

CHAPTER 3

A REDUCED-BASIS METHOD FOR PARAMETRIZED VARIATIONAL INEQUALITIES WITH NONLINEAR CONSTRAINTS

Abstract

We investigate new developments of the Reduced-Basis (RB) method for parametrized optimization problems with nonlinear constraints. In this chapter, we propose a reduced-basis scheme in a saddle-point form combined with the Empirical Interpolation Method to deal with the nonlinear constraint. In this setting, a 'primal' reduced-basis is needed for the primal solution and a 'dual' one is needed for Lagrange multipliers. We suggest to construct the latter using a 'cone-projected' hierarchical algorithm that conserves the non-negativity of the dual basis vectors. The reduction strategy is applied to elastic frictionless contact problems including the possibility of using non-matching meshes. We study test cases that are inspired from existing work on finite elements for contact mechanics. The numerical examples confirm the efficiency of the reduction strategy.

3.1 Introduction

Constrained optimization problems are of great importance in numerous engineering applications. Owing to the nonlinear nature of some constraints, the algorithms designed for solving nonlinearly-constrained optimization problems often suffer from slow convergence; thereby entailing subsequent computational costs. Besides, the Reduced-Basis (RB) method [30, 48] is a computationally effective approach to approximate the solutions of parametrized Partial Differential Equations (PDEs) encountered in many problems in science and engineering. In particular, it is highly

beneficial in nonlinear settings that tend to substantially increase computational complexity. Although significant progress has been achieved in this field [4, 21, 27], the model reduction of parametrized optimization problems involving nonlinear constraints remains in need of further advances.

Here, the problem of interest is a parametrized optimization problem with nonlinear constraints which is formulated as a saddle-point problem and numerically solved using Lagrange multipliers. We consider a situation where this problem must be solved in a multi-query or real-time context, i.e., the problem has to be solved repeatedly for a large number of parameter values or it needs to be solved very quickly under limited computational resources. For standard PDEs in variational form, RB methods provide efficient tools for complexity reduction. More precisely, instead of the High-Fidelity (HF) problem, which is typically infinite-dimensional or rather high-dimensional after a finite element discretization, a low-dimensional model is generated. This low-dimensional problem can then be solved significantly faster for a wide range of parameters. Lately, the reduced modeling of variational inequalities has gained growing interest. In the literature, three recent papers address somewhat related problems. The first paper [27] extends the standard RB method to linear variational inequalities solved through a mixed formulation. Regarding the construction of the bases, the 'primal' basis (for the primal solution) and the 'dual' one (for the Lagrange multipliers) are directly composed of well-chosen snapshots. No additional compression phase is considered. In the so-called Projection-Based method of [4], which has been specifically introduced to address time-dependent contact problems with linear constraints, the primal and the dual bases are built differently. An efficient primal RB is obtained using the POD. Since [4] focuses on dynamic problems with numerous time instants, the set of Lagrange multiplier snapshots can rapidly become sizable and its compression is therefore an important task. Therein, a dual basis is built by applying the Non-negative Matrix Factorization (NMF) algorithm [36] to the set of Lagrange multiplier snapshots. The NMF guarantees non-negative basis vectors and a limited RB dimension, but the resulting dual RB can be (far) less accurate than the primal RB. Another concern is that the user does not specify a required error tolerance as an input but a number of dominant basis vectors to retain. Finally, the work in [21] extends another type of model-order reduction methods called Hyper-Reduction (HR) to contact problems with linear constraints. The proposed extension of the HR method consists in conserving a few vectors of the High-Fidelity (HF) dual basis because the number of contact nodes is limited to a Reduced Integration Domain (RID). Hence, only the contact nodes in the RID are treated but with a local high fidelity. So far, all the existing results are restricted to linear constraints.

In this paper, we propose to extend constrained model reduction to the framework of nonlinear constraints. Motivated by industrial applications in contact mechanics, we address a nonlinear type of constraints that can be written in a quasi-linear form so as to use a fixed-point iterative scheme. We express the problem of interest in a saddle-point form, and we apply the Empirical Interpolation Method (EIM) [5, 39] to allow for an offline/online decomposition of the nonlinear con-

straints. Regarding basis construction, a 'primal' reduced basis is needed for the primal solution and a 'dual' one is needed for the Lagrange multipliers. Whereas the primal basis is constructed using a standard POD, we introduce a 'cone-projected' algorithm that builds nested dual bases while preserving the non-negativity of the basis vectors. The forthcoming analysis is meant to address static parametrized optimization problems, the extension to a time-dependent setting being straightforward once a time-discretization scheme has been chosen. An important application we have in mind is frictionless contact in a generic framework, namely without the small displacement assumption and for various types of contact constraints, including the intricate case of non-matching meshes. Both of the previous features lead to nonlinear operators in the definition of the constraint. On the one hand, the small displacement hypothesis allows one to consider the same normal vector on both contact boundaries. This work aims at going beyond this assumption. On the other hand, assuming that the meshes match eludes the nonlinearity induced by the spatial discretization of the constraint. Unfortunately, the latter assumption is not realistic in many engineering scenarios. Therefore, we also aim at addressing the case of non-matching meshes.

This chapter is organized as follows. In Section 3.2, we introduce the model problem. In Section 3.3, we consider elastic contact problems. Since we do not consider the simplifying hypotheses discussed above, we describe the way we derive the nonlinear non-interpenetration condition in some detail. In Section 3.4, we return to the general setting and we apply the RB method to derive a reduced resolution scheme. In Section 3.5, we discuss the offline stage in some detail, we present the EIM procedure for the nonlinear constraint, and we describe the construction of the primal and dual RB spaces. In Section 3.6, we present numerical results illustrating the performance of the method in the framework of elastic frictionless contact. We consider the contact problem between two spheres introduced by Hertz [29] with a parametrization of the radius of one of the spheres. Finally, Section 3.7 collects some technical results.

3.2 Model problem

Let $\mathcal{V}$ be a separable Hilbert space composed of functions defined on a spatial domain (open, bounded, connected subset) $\Omega \subset \mathbb{R}^d$, $d \geqslant 1$, with a Lipschitz boundary $\partial\Omega$. Let $\overline{\Omega}$ denote the closure of Ω and let $\mathcal{P}$ denote a parameter set. We define a continuous, symmetric and coercive bilinear form $a : \mathcal{P} \times \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ (the attributes of a are with respect to its second and third arguments), and a continuous linear form $f : \mathcal{P} \times \mathcal{V} \to \mathbb{R}$ (the attributes of f are with respect to its second argument). We also define a nonlinear continuous map $k : \mathcal{P} \times \mathcal{V} \times \mathcal{V} \to L^2(\Gamma_1^c)$ and a nonlinear continuous map $g : \mathcal{P} \times \mathcal{V} \to L^2(\Gamma_1^c)$, for a subset $\Gamma_1^c \subset \partial\Omega$. For simplicity, we consider at this stage that the domain Ω and the subset Γ_1^c are parameter-independent. A more general setting with parameter-dependent $\Omega(\mu)$ and $\Gamma_1^c(\mu)$ will be considered from Section 3.3.2 onwards.

For all $\mu \in \mathcal{P}$, we want to solve the following nonlinear minimization problem:

Find $u(\mu) \in \mathcal{V}$ such that

$$\begin{cases} u(\mu) = \underset{v \in \mathcal{V}}{\operatorname{argmin}} \ \dfrac{1}{2} a(\mu; v, v) - f(\mu; v) \\ k(\mu, u(\mu); u(\mu)) \leqslant g(\mu, u(\mu)) \quad \text{a.e. on } \Gamma_1^c. \end{cases} \tag{3.1}$$

The semicolons in (3.1) indicate that a is linear with respect to its second and third arguments and that k is linear with respect to its third argument.

Remark 3.1 (Nonlinear constraint)**.** *The nonlinear constraint in* (3.1) *can be formulated more compactly as* $\zeta(\mu, u(\mu)) \leqslant 0$ *for the nonlinear continuous map* $\zeta(\mu, v) : \mathcal{P} \times \mathcal{V} \to L^2(\Gamma_1^c)$ *defined as*

$$\zeta(\mu, v) := k(\mu, v; v) - g(\mu, v). \tag{3.2}$$

The adopted decomposition of $\zeta(\mu, v)$ *in* (3.2) *is natural in the context of nonlinear contact problems. Note that this decomposition is not unique since one can write* $\zeta(\mu, v) = \tilde{k}(\mu, v; v) - \tilde{g}(\mu, v)$ *with* $\tilde{k}(\mu, v; v) := k(\mu, v; v) + \delta(\mu, v; v)$, $\tilde{g}(\mu, v) := g(\mu, v) + \delta(\mu, v; v)$, *and an arbitrary map* $\delta(\mu, v; v) : \mathcal{P} \times \mathcal{V} \times \mathcal{V} \to L^2(\Gamma_1^c)$.

In the present setting, we make three assumptions. First, we assume that the inequality constraint in (3.1) is quasi-linear, i.e., that k is linear with respect to its third argument. This assumption will be exploited below in setting up an iterative solver for the discrete version of (3.1). Second, we assume that g satisfies $g(\mu, 0) \geqslant 0$. Third, we assume that the problem (3.1) is well-posed. Note that the functional minimized in (3.1) is strongly convex and continuous. Moreover, the set of admissible states

$$\mathcal{K} = \{v \in \mathcal{V} \mid k(\mu, v; v) \leqslant g(\mu, v)\} \tag{3.3}$$

is non-empty since $0 \in \mathcal{K}$ because $g(\mu, 0) \geqslant 0$, and the set is closed owing to the continuity of k and g. Therefore, the existence of a minimizer is guaranteed. Our third assumption then means that we assume the uniqueness of the searched minimizer in $\mathcal{K}$. In fact, the above setting is motivated by contact constrained problems that will be described in more detail in Section 3.3 below.

Let $\mathcal{W}$ be a non-empty closed convex cone composed of functions defined on the subset $\Gamma_1^c \subset \partial\Omega$. We assume that $\mathcal{W} := L^2(\Gamma_1^c, \mathbb{R}_+)$, with $\mathbb{R}_+ := [0, +\infty)$. Using the test space $\mathcal{W}$, the variational formulation of the inequality constraint in (3.1) reads as follows:

$$\int_{\Gamma_1^c} k(\mu, u(\mu); u(\mu))\eta \leqslant \int_{\Gamma_1^c} g(\mu, u(\mu))\eta, \quad \forall \eta \in \mathcal{W}. \tag{3.4}$$

Using a Lagrangian formulation, the optimization problem (3.1) is rewritten as a saddle-point problem. Specifically, (3.1) can be recast as: Find $(u(\mu), \lambda(\mu)) \in \mathcal{V} \times \mathcal{W}$ such that

$$(u(\mu), \lambda(\mu)) = \arg \underset{v \in \mathcal{V}, \eta \in \mathcal{W}}{\operatorname{minmax}} \ \mathcal{L}(\mu)(v, \eta), \tag{3.5}$$

where the Lagrangian $\mathcal{L}(\mu) : \mathcal{V} \times \mathcal{W} \to \mathbb{R}$ is defined as

$$\mathcal{L}(\mu)(v, \eta) := \frac{1}{2} a(\mu; v, v) - f(\mu; v) + \left(\int_{\Gamma_1^c} k(\mu, v; v)\eta - \int_{\Gamma_1^c} g(\mu, v)\eta \right), \tag{3.6}$$

and $u(\mu)$ and $\lambda(\mu)$ are respectively called the primal and the dual solutions of the saddle-point problem (3.5).

In practice, one uses a conforming Finite Element Method (FEM) [20] to discretize (3.5) in space. The FEM is based on a finite element subspace $V_{\mathcal{N}} := \mathrm{span}\{\phi_1, \ldots, \phi_{\mathcal{N}}\} \subsetneq \mathcal{V}$ defined using a discrete nodal subset $\Omega^{\mathrm{tr}} \subsetneq \overline{\Omega}$, where $\mathrm{Card}(\Omega^{\mathrm{tr}}) = \mathcal{N}$. Besides, one introduces the subset $W_{\mathcal{R}} := \mathrm{span}_+\{\psi_1, \ldots, \psi_{\mathcal{R}}\} \subsetneq \mathcal{W}$ defined using a discrete nodal subset $\Gamma_1^{c,\mathrm{tr}} \subsetneq \Gamma_1^c$, where $\mathrm{Card}(\Gamma_1^{c,\mathrm{tr}}) = \mathcal{R}$. The notation span_+ means that linear combinations are restricted to non-negative coefficients. The discrete sadlle-point problem reads: Find $(u_{\mathcal{N}}(\mu), \lambda_{\mathcal{R}}(\mu)) \in V_{\mathcal{N}} \times W_{\mathcal{R}}$ such that

$$(u_{\mathcal{N}}(\mu), \lambda_{\mathcal{R}}(\mu)) = \arg \min_{v \in V_{\mathcal{N}}, \eta \in W_{\mathcal{R}}} \max \mathcal{L}(\mu)(v, \eta), \tag{3.7}$$

with the Lagrangian defined in (3.6). Note that the discrete inequality constraint reads

$$\int_{\Gamma_1^c} k(\mu, u_{\mathcal{N}}(\mu); u_{\mathcal{N}}(\mu))\psi_i \leqslant \int_{\Gamma_1^c} g(\mu, u_{\mathcal{N}}(\mu))\psi_i, \quad \forall i \in \{1, \ldots, \mathcal{R}\}. \tag{3.8}$$

As is customary with the RB method, we assume henceforth that the mesh-size is small enough so that the above space discretization method delivers HF approximate primal and dual solutions within the desired level of accuracy. Introducing the component vectors $\mathbf{u}(\mu) := (u_n(\mu))_{1 \leqslant n \leqslant \mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$ and $\boldsymbol{\lambda}(\mu) := (\lambda_n(\mu))_{1 \leqslant n \leqslant \mathcal{R}} \in \mathbb{R}^{\mathcal{R}}_+$ of $u_{\mathcal{N}}(\mu)$ and $\lambda_{\mathcal{N}}(\mu)$, the algebraic formulation of (3.7) reads: Find $(\mathbf{u}(\mu), \boldsymbol{\lambda}(\mu)) \in \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{R}}_+$ satisfying

$$(\mathbf{u}(\mu), \boldsymbol{\lambda}(\mu)) \in \arg \min_{\mathbf{v} \in \mathbb{R}^{\mathcal{N}}, \boldsymbol{\eta} \in \mathbb{R}^{\mathcal{R}}_+} \max \frac{1}{2}\mathbf{v}^T\mathbf{A}(\mu)\mathbf{v} - \mathbf{v}^T\mathbf{f}(\mu) + \boldsymbol{\eta}^T\big(\mathbf{K}(\mu, \mathbf{v})\mathbf{v} - \mathbf{g}(\mu, \mathbf{v})\big), \tag{3.9}$$

with the matrices $\mathbf{A}(\mu) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ and $\mathbf{K}(\mu, w) \in \mathbb{R}^{\mathcal{R} \times \mathcal{N}}$ such that

$$\begin{cases} \mathbf{A}(\mu)_{ij} = a(\mu; \phi_j, \phi_i), \\ \mathbf{K}(\mu, w)_{ij} = \displaystyle\int_{\Gamma_1^c} k(\mu, w; \phi_j)\psi_i, \end{cases} \tag{3.10}$$

and the vectors $\mathbf{f}(\mu) \in \mathbb{R}^{\mathcal{N}}$ and $\mathbf{g}(\mu, w) \in \mathbb{R}^{\mathcal{R}}$ such that

$$\begin{cases} \mathbf{f}(\mu)_j = f(\mu; \phi_j), \\ \mathbf{g}(\mu, w)_j = \displaystyle\int_{\Gamma_1^c} g(\mu, w)\psi_j. \end{cases} \tag{3.11}$$

In the sequel, we will solve (3.9) using an iterative algorithm, where the terms $\mathbf{K}(\mu, \mathbf{v})$ and $\mathbf{g}(\mu, \mathbf{v})$ are treated explicitly. This amounts to a so-called 'secant method' or Kačanov iterative method [22]. The Kačanov iteration consists in solving the following problem : For all $k \geqslant 1$,

$$\begin{aligned} (\mathbf{u}^k(\mu), \boldsymbol{\lambda}^k(\mu)) = \arg \min_{\mathbf{v} \in \mathbb{R}^{\mathcal{N}}, \eta \in \mathbb{R}^{\mathcal{R}}_+} \max \frac{1}{2}\mathbf{v}^T\mathbf{A}(\mu)\mathbf{v} - \mathbf{v}^T\mathbf{f}(\mu) \\ + \boldsymbol{\eta}^T\big(\mathbf{K}(\mu, \mathbf{u}^{k-1}(\mu))\mathbf{v} - \mathbf{g}(\mu, \mathbf{u}^{k-1}(\mu))\big). \end{aligned} \tag{3.12}$$

For a user-defined tolerane $\epsilon_{\text{KA}} > 0$, the stopping criterion of the Kačanov iterative algorithm reads

$$\frac{\|\mathbf{u}^k(\mu) - \mathbf{u}^{k-1}(\mu)\|_{\mathbb{R}^{\mathcal{N}}}}{\|\mathbf{u}^{k-1}(\mu)\|_{\mathbb{R}^{\mathcal{N}}}} \leqslant \epsilon_{\text{KA}}. \tag{3.13}$$

Depending on the problem and output of interest, an additional check on the dual increment $\|\boldsymbol{\lambda}^k(\mu) - \boldsymbol{\lambda}^{k-1}(\mu)\|_{\mathbb{R}^{\mathcal{R}}} / \|\boldsymbol{\lambda}^{k-1}(\mu)\|_{\mathbb{R}^{\mathcal{R}}}$ can be performed. A brief comparison with the Newton method is presented in Section 3.7.3. The advantage of the Kačanov iterative method is its simplicity. Indeed, unlike the standard Newton method, Kačanov iterations do not require any computation of Jacobian preconditioners, thereby achieving significant computational savings when solving (3.9). On the other hand, if the Newton method converges, it is (much) faster than the Kačanov iteration. In Section 3.4 below, the reduced problem will be solved using the Kačanov iteration as well. Therein, we will shortly discuss the influence of the solver on the reduction scheme.

3.3 Prototypical example: elastic contact

The model reduction of mechanical problems involving contact remains an important issue in computational solid mechanics. In this section, we consider the case of frictionless contact for linear elasticity. An important difference with regard to the previous section is that the domain over which the problem is posed is now parameter-dependent.

3.3.1 Linear elasticity

For all $\mu \in \mathcal{P}$, the domain $\Omega(\mu) \subset \mathbb{R}^d$, $d \in \{2, 3\}$ represents the initial configuration of a deformable medium initially at equilibrium and to which an external load $\ell(\mu) : \Omega(\mu) \to \mathbb{R}^d$ is applied. The standard linear elasticity problem consists in finding the displacement field $u(\mu) : \Omega(\mu) \to \mathbb{R}^d$ induced by the externally applied force field $\ell(\mu)$ once the system has reached equilibrium. Let $\sigma(u(\mu)) : \Omega(\mu) \to \mathbb{R}^{d\times d}$ be the stress tensor in the medium. The equilibrium conditions can be expressed as

$$\nabla \cdot \sigma(u(\mu)) = \ell(\mu), \quad \text{in } \Omega(\mu). \tag{3.14}$$

We define the functional space $\mathcal{V}(\mu)$ such that

$$\mathcal{V}(\mu) := H^1(\Omega(\mu); \mathbb{R}^d). \tag{3.15}$$

For all $v \in \mathcal{V}(\mu)$, let $\varepsilon : \Omega(\mu) \to \mathbb{R}^{d\times d}$ be the linearized strain tensor defined as

$$\varepsilon(v) := \frac{1}{2}(\nabla v + \nabla v^T). \tag{3.16}$$

In the framework of linear isotropic elasticity, the stress tensor is related to the linearized strain tensor by the formula

$$\sigma(v) = \frac{E\nu}{(1+\nu)(1+2\nu)}\text{tr}(\varepsilon(v))\mathcal{I} + \frac{E}{(1+\nu)}\varepsilon(v), \tag{3.17}$$

where E is the Young modulus, ν is the Poisson coefficient and $\mathcal{I}$ is the identity tensor in $\mathbb{R}^{d\times d}$. For simplicity, we have supposed that E and ν are parameter-independent. At this stage, we define the bilinear form $a : \mathcal{P} \times \mathcal{V}(\mu) \times \mathcal{V}(\mu) \to \mathbb{R}$ introduced in (3.1) as

$$a(\mu; v, w) = \int_{\Omega(\mu)} \sigma(v) : \varepsilon(w), \tag{3.18}$$

and the linear form $f : \mathcal{P} \times \mathcal{V}(\mu) \to \mathbb{R}$ introduced in (3.1) as

$$f(\mu; w) = \int_{\Omega(\mu)} \ell(\mu) w. \tag{3.19}$$

3.3.2 Non-interpenetration condition

We intend to model the non-interpenetration condition in a general framework. We use the previously introduced notation and only define the new quantities dedicated to this particular formulation. We consider that the domain $\Omega(\mu)$ can be partitionned as

$$\overline{\Omega}(\mu) = \overline{\Omega}_1(\mu) \cup \overline{\Omega}_2(\mu),$$

where $\Omega_1(\mu)$ and $\Omega_2(\mu)$ represent the initial configuration of the two disjoint deformable media. For all $\mu \in \mathcal{P}$, let $\Gamma_1^c(\mu)$ and $\Gamma_2^c(\mu)$ be the potential contact boundaries of $\Omega_1(\mu)$ and $\Omega_2(\mu)$, such that

$$\overline{\Gamma}^c(\mu) := \overline{\Gamma}_1^c(\mu) \cup \overline{\Gamma}_2^c(\mu).$$

For all $v \in \mathcal{V}(\mu)$ and all $i \in \{1, 2\}$, we introduce the following functions $v_i : \Omega_i(\mu) \to \mathbb{R}^d$ such that

$$v_i := v_{|\Omega_i(\mu)}. \tag{3.20}$$

In order to formulate the contact conditions in a general setting, we need to introduce some auxiliary geometric mappings. An illustration of the various geometric mappings is given in Figure 3.1. For all $v \in \mathcal{V}$, all $\mu \in \mathcal{P}$ and all $i \in \{1, 2\}$, we define the geometric mappings

$$\begin{aligned} \psi_i(\mu, v_i) : \ \Gamma_i^c(\mu) &\to \Upsilon_i^c(\mu, v_i) \\ z &\mapsto z + v_i(z), \end{aligned} \tag{3.21}$$

where $\Upsilon_i^c(\mu, v_i) := \psi_i(\mu, v_i)\big(\Gamma_i^c(\mu)\big)$. In what follows, we assume implicitly that v_i is injective so that $\psi_i(\mu, v_i)$ is injective as well. Therefore, $(\psi_i(\mu, v_i))^{-1} : \Upsilon_i^c(\mu, v_i) \to \Gamma_i^c(\mu)$ is well defined. This assumption is natural in the context of solid mechanics. Under a local convexity assumption on $\Upsilon_2^c(\mu, v_2)$, the contact mapping

$$\begin{aligned} \vartheta(\mu, v) : \Upsilon_1^c(\mu, v_1) &\to \Upsilon_2^c(\mu, v_2) \\ z_1 &\mapsto \operatorname*{argmin}_{z_2 \in \Upsilon_2^c(\mu, v_2)} \|z_1 - z_2\|, \end{aligned} \tag{3.22}$$

is well defined. The contact mapping $\vartheta(\mu, v)$ can be physically interpreted as the function relating every point on $\Upsilon_1^c(\mu, v_1)$ to a potential contact point on $\Upsilon_2^c(\mu, v_2)$.

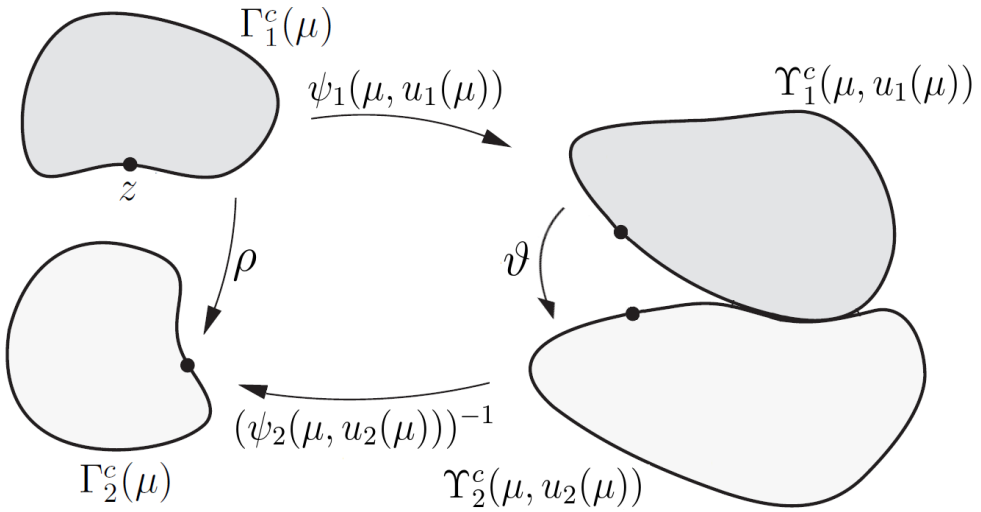

Figure 3.1 – Generic two-body contact problem. For simplicity of the scheme, the entire boundary is taken to be the potential contact boundary.

To be more precise, for all $z \in \Upsilon_1^c(\mu, v_1)$, $\vartheta(\mu, v)(z)$ is the orthogonal projection of z onto $\Upsilon_2^c(\mu, v_2)$. The contact mapping $\vartheta(\mu, v)$ depends on the displacement field v and is therefore unknown *a priori* for the solution $v = u(\mu)$. For all $v \in \mathcal{V}$ and all $\mu \in \mathcal{P}$, we define the mapping

$$\begin{cases} \rho(\mu, v) : \Gamma_1^c(\mu) \to \Gamma_2^c(\mu) \\ \rho(\mu, v) := (\psi_2(\mu, v_2))^{-1} \circ \vartheta(\mu, v) \circ \psi_1(\mu, v_1), \end{cases} \tag{3.23}$$

and the vector field of outward normals on $\Upsilon_2^c(\mu, v_2)$

$$n_2(\mu, v_2) : \Upsilon_2^c(\mu, v_2) \to \mathbb{R}^d. \tag{3.24}$$

It is also convenient to introduce the vector field

$$\begin{aligned} &\widetilde{n}_2(\mu, v) : \Gamma_1^c(\mu) \to \mathbb{R}^d \\ &\widetilde{n}_2(\mu, v) := n_2(\mu, v_2) \circ \vartheta(\mu, v) \circ \psi_1(\mu, v_1), \end{aligned} \tag{3.25}$$

which corresponds to the outward normal on $\Upsilon_2^c(\mu, v_2)$ but defined at the corresponding point in $\Gamma_1^c(\mu)$ through the mapping $\vartheta(\mu, v) \circ \psi_1(\mu, v_1)$.

For an admissible solution $u(\mu) = (u_1(\mu), u_2(\mu)) \in \mathcal{V}(\mu)$, the non-interpenetration condition reads: For all $z \in \Gamma_1^c(\mu)$,

$$\begin{aligned} \left(u_1(\mu)(z) - \left(u_2(\mu) \circ \rho(\mu, u(\mu))\right)(z)\right) \cdot \widetilde{n}_2(\mu, u(\mu))(z) \\ \geqslant \left(\rho(\mu, u(\mu))(z) - z\right) \cdot \widetilde{n}_2(\mu, u(\mu))(z). \end{aligned} \tag{3.26}$$

At this stage, we can define the displacement map k and the gap map g in (3.1) as

$$k(\mu, w; v)(z) = -\left(v_1(z) - \left(v_2 \circ \rho(\mu, w)\right)(z)\right) \cdot \widetilde{n}_2(\mu, w)(z), \tag{3.27}$$

and

$$g(\mu, w)(z) = -\left(\rho(\mu, w)(z) - z\right) \cdot \widetilde{n}_2(\mu, w)(z), \tag{3.28}$$

for all $z \in \Gamma_1^c(\mu)$. Hence, (3.26) can be recast as

$$k(\mu, u(\mu); u(\mu)) \leqslant g(\mu, u(\mu)), \tag{3.29}$$

leading to the same inequality constraint as in (3.1).

Lemma 3.1. *The constraint* (3.26) *is equivalent to the physical non-interpenetration condition*

$$\overline{\Omega}_1(\mu, u_1(\mu)) \cap \overline{\Omega}_2(\mu, u_2(\mu)) \subset \Upsilon^c(\mu, u(\mu)), \tag{3.30}$$

where

$$\overline{\Omega}_i(\mu, u_i(\mu)) := (Id + u_i(\mu))(\overline{\Omega}_i(\mu)), \quad \forall i \in \{1, 2\}, \tag{3.31}$$

and

$$\Upsilon^c(\mu, u(\mu)) := \Upsilon_1^c(\mu, u_1(\mu)) \cap \Upsilon_2^c(\mu, u_2(\mu)), \tag{3.32}$$

Proof. The proof is postponed to Section 3.7.3. □

Remark 3.2 (Physical interpretation)**.** *Incidentally,* (3.30) *means that the intersection of the two deformed solids* $\overline{\Omega}_1(\mu, u_1(\mu))$ *and* $\overline{\Omega}_2(\mu, u_2(\mu))$ *is necessarily a subset of their contact boundaries.*

Remark 3.3 (Symmetry)**.** *Note that the indices* 1 *and* 2 *play symmetric roles in* (3.30)*.*

3.4 The reduced-basis model

In this section, we return to the general setting of Section 3.2 and we derive a general RB formulation for the nonlinear minimization problem (3.1), and more precisely its algebraic saddle-point formulation (3.9). Yet, following Section 3.3, we now consider the more general setting of a parameter-dependent domain $\Omega(\mu)$.

3.4.1 Reference domain

In view of model reduction, we assume that there exists a bi-Lipschitz diffeomorphism called geometric mapping $h(\mu)$ and defined on a parameter-independent reference domain $\breve{\Omega}$ such that

$$\begin{aligned} h(\mu) : \breve{\Omega} &\to \Omega(\mu) \\ x &\mapsto \sum_{i=1}^{I} h_i(\mu, x) \mathbf{1}_{\breve{\Omega}_i}(x), \end{aligned} \tag{3.33}$$

where $\{\breve{\Omega}_i\}_{i=1}^I$ is a partition of $\breve{\Omega}$. Using this geometric mapping, we introduce the reference Hilbert space $\breve{\mathcal{V}} := H^1(\breve{\Omega}; \mathbb{R}^d)$ composed of functions defined on $\breve{\Omega}$ so that

$$\mathcal{V}(\mu) = \breve{\mathcal{V}} \star h(\mu)^{-1} = \{v \circ h(\mu)^{-1} \mid v \in \breve{\mathcal{V}}\}. \tag{3.34}$$

In what follows, for all $i \in \{1, \dots, I\}$, we set

$$\Omega_i(\mu) = h(\mu)(\breve{\Omega}_i), \tag{3.35}$$

so that $\breve{\Omega}_i$ is parameter-independent.

We assume for simplicity that $I = 2$ in (3.33), which corresponds to the situation where there are two disjoint solids $\Omega_1(\mu)$ and $\Omega_2(\mu)$ that can come into contact. We then define the reference contact boundaries such that

$$\Gamma_i^c(\mu) = h(\mu)(\breve{\Gamma}_i^c), \quad \forall i \in \{1, 2\}. \tag{3.36}$$

We also define $\mathcal{W}(\mu) := L^2(\Gamma_1^c(\mu); \mathbb{R}_+)$ and $\breve{\mathcal{W}} := L^2(\breve{\Gamma}_1^c; \mathbb{R}_+)$ such that

$$\mathcal{W}(\mu) = \breve{\mathcal{W}} \star h^{-1}(\mu)_{|\Gamma_1^c(\mu)} = \{\breve{\eta} \circ \star h^{-1}(\mu)_{|\Gamma_1^c(\mu)} \mid \breve{\eta} \in \breve{\mathcal{W}}\}. \tag{3.37}$$

3.4.2 Reduced basis spaces

Let $V_\mathcal{N}(\mu)$ and $W_\mathcal{R}(\mu)$ be the FEM discretizations of $\mathcal{V}(\mu)$ and $\mathcal{W}(\mu)$, respectively. The reference counterparts of these spaces are $\breve{V}_\mathcal{N}$ and $\breve{W}_\mathcal{R}$. Towards an accurate approximation of the solution manifold, we consider a so-called 'primal' RB subspace $\widehat{V}_N(\mu)$ and a so-called 'dual' RB subcone $\widehat{W}_R(\mu)$ such that

$$\widehat{V}_N(\mu) \subset V_\mathcal{N}(\mu) \subset \mathcal{V}(\mu), \quad \text{and} \quad \widehat{W}_R(\mu) \subset W_\mathcal{R}(\mu) \subset \mathcal{W}(\mu). \tag{3.38}$$

The dimensions of these spaces are such that $N = \dim(\widehat{V}_N(\mu)) \ll \dim(V_\mathcal{N}(\mu)) = \mathcal{N}$ and $R = \dim(\widehat{W}_R(\mu)) \ll \dim(W_\mathcal{R}(\mu)) = \mathcal{R}$. The reference counterparts of the parameter-dependent RB spaces in (3.38) are spaces $\breve{V}_N$ and $\breve{W}_R$ that satisfy

$$\breve{V}_N \subset \breve{V}_\mathcal{N} \subset \breve{\mathcal{V}}, \quad \text{and} \quad \breve{W}_R \subset \breve{W}_\mathcal{R} \subset \breve{\mathcal{W}}. \tag{3.39}$$

Let $(\breve{\theta}_n)_{1 \leqslant n \leqslant N}$ be an orthonormal basis of $\breve{V}_N$ and let $(\breve{\xi}_n)_{1 \leqslant n \leqslant R}$ be generating vectors of the cone $\breve{W}_R$, i.e., $\breve{W}_R = \mathrm{span}_+\{\breve{\xi}_1, \dots, \breve{\xi}_R\} = \{\sum_{n=1}^R \omega_n \breve{\xi}_n | \ \omega_n \geqslant 0\}$. For all $\mu \in \mathcal{P}$, the primal RB solution $\widehat{u}(\mu) \in \widehat{V}_N(\mu)$ and the dual RB solution (Lagrange multipliers) $\widehat{\lambda}(\mu) \in \widehat{W}_R(\mu)$ that approximate the HF solution $(u_\mathcal{N}(\mu), \lambda_\mathcal{R}(\mu)) \in V_\mathcal{N}(\mu) \times W_\mathcal{R}(\mu)$ are decomposed as

$$\widehat{u}(\mu) = \sum_{n=1}^N \widehat{u}_n(\mu) \breve{\theta}_n \circ h(\mu)^{-1}, \qquad \widehat{\lambda}(\mu) = \sum_{n=1}^R \widehat{\lambda}_n(\mu) \breve{\xi}_n \circ h(\mu)^{-1}, \tag{3.40}$$

with real numbers $\widehat{u}_n(\mu)$ for all $n \in \{1, \dots, N\}$ and non-negative real numbers $\widehat{\lambda}_n(\mu)$ for all $n \in \{1, \dots, R\}$. Let us introduce the component vectors $\widehat{\mathbf{u}}(\mu) := (\widehat{u}_n(\mu))_{1 \leqslant n \leqslant N} \in \mathbb{R}^N$ and $\widehat{\boldsymbol{\lambda}}(\mu) := (\widehat{\lambda}_n(\mu))_{1 \leqslant n \leqslant R} \in \mathbb{R}_+^R$, for all $\mu \in \mathcal{P}$. The RB formulation of (3.9) reads: Find $(\widehat{\mathbf{u}}(\mu), \widehat{\boldsymbol{\lambda}}(\mu)) \in \mathbb{R}^N \times \mathbb{R}_+^R$ such that

$$(\widehat{\mathbf{u}}(\mu), \widehat{\boldsymbol{\lambda}}(\mu)) \in \arg \min_{\widehat{\mathbf{v}} \in \mathbb{R}^N} \max_{\widehat{\boldsymbol{\eta}} \in \mathbb{R}_+^R} \frac{1}{2} \widehat{\mathbf{v}}^T \widehat{\mathbf{A}}(\mu) \widehat{\mathbf{v}} - \widehat{\mathbf{v}}^T \widehat{\mathbf{f}}(\mu) + \widehat{\boldsymbol{\eta}}^T \big(\widehat{\mathbf{K}}(\mu, \widehat{\mathbf{v}}) \widehat{\mathbf{v}} - \widehat{\mathbf{g}}(\mu, \widehat{\mathbf{v}}) \big), \tag{3.41}$$

with the matrices $\widehat{\mathbf{A}}(\mu) \in \mathbb{R}^{N\times N}$ and $\widehat{\mathbf{K}}(\mu, \widehat{\mathbf{v}}) \in \mathbb{R}^{R\times N}$ such that

$$\widehat{\mathbf{A}}(\mu)_{pn} = a(\mu; \breve{\theta}_n \circ h(\mu)^{-1}, \breve{\theta}_p \circ h(\mu)^{-1}), \tag{3.42a}$$

$$\widehat{\mathbf{K}}(\mu, \widehat{\mathbf{v}})_{pn} = \int_{\Gamma_1^c(\mu)} k\left(\mu, \sum_{i=1}^{N} \widehat{v}_i \breve{\theta}_i \circ h(\mu)^{-1}; \breve{\theta}_n \circ h(\mu)^{-1}\right) \breve{\xi}_p \circ h(\mu)^{-1}, \tag{3.42b}$$

and the vectors $\widehat{\mathbf{f}}(\mu) \in \mathbb{R}^N$ and $\widehat{\mathbf{g}}(\mu, \widehat{\mathbf{v}}) \in \mathbb{R}^R$ such that

$$\widehat{\mathbf{f}}(\mu)_p = f(\mu; \breve{\theta}_p \circ h(\mu)^{-1}), \tag{3.43a}$$

$$\widehat{\mathbf{g}}(\mu, \widehat{\mathbf{v}})_p = \int_{\Gamma_1^c(\mu)} g\left(\mu, \sum_{i=1}^{N} \widehat{v}_i \breve{\theta}_i \circ h(\mu)^{-1}\right) \breve{\xi}_p \circ h(\mu)^{-1}. \tag{3.43b}$$

3.4.3 Separation of the elastic energy

We assume the existence of two integers J^a and J^f and of matrices $\widehat{\mathbf{A}}_j \in \mathbb{R}^{N\times N}$, with $1 \leqslant j \leqslant J^a$, and vectors $\widehat{\mathbf{f}}_j \in \mathbb{R}^N$, with $1 \leqslant j \leqslant J^f$, such that

$$\widehat{\mathbf{A}}_{j,pn} = a_j(\breve{\theta}_n, \breve{\theta}_p), \quad \text{and} \quad \widehat{\mathbf{f}}_{j,pn} = f_j(\breve{\theta}_p), \tag{3.44}$$

defined using the reference basis functions such that the matrix $\widehat{\mathbf{A}}(\mu)$ defined in (3.42a) and the vector $\widehat{\mathbf{f}}(\mu)$ defined in (3.43a) can be affinely decomposed under the form

$$\left(\widehat{\mathbf{A}}(\mu)\right)_{np} = \sum_{j=1}^{J^a} \alpha_j^a(\mu) \widehat{\mathbf{A}}_{j,np}, \text{ and } \left(\widehat{\mathbf{f}}(\mu)\right)_p = \sum_{j=1}^{J^f} \alpha_j^f(\mu) \widehat{\mathbf{f}}_{j,p}, \ \forall 1 \leqslant n, p \leqslant N, \tag{3.45}$$

where the dependencies on μ and n, p are separated. The key point is that the quantities in (3.44) are offline-computable. During the online stage, all that remains to be performed is the assembly of $\widehat{\mathbf{A}}(\mu)$ and $\widehat{\mathbf{f}}(\mu)$ using (3.45) for each new parameter value $\mu \in \mathcal{P}$.

In order to clarify how the separated representations in (3.45) are derived, we consider the elastic problem defined in Section 3.3 with homogenous load function $\ell(\mu)$ and a test function w. We have

$$\begin{aligned} f(\mu; w) &= \int_{\Omega(\mu)} \ell(\mu) w(x) \, dx \\ &= \int_{\breve{\Omega}} \ell(\mu) w(h(\mu)(\breve{x})) \det\left(\mathrm{Jac}(h(\mu))(\breve{x})\right) \, d\breve{x}, \end{aligned} \tag{3.46}$$

where the notation $\det\left(\mathrm{Jac}(h(\mu))\right)$ refers to the determinant of the Jacobian matrix of the geometric mapping $h(\mu)$. Using the definition of $h(\mu)$ in (3.33), we obtain

$$\begin{aligned} f(\mu; w) &= \sum_{j=1}^{2} \int_{\breve{\Omega}_j} h_j(\mu) \ell(\mu) \breve{w}(\breve{x}) \det\left(\mathrm{Jac}(\mathbf{1}_{\breve{\Omega}_j})(x)\right) \, d\breve{x} \\ &= \sum_{j=1}^{2} h_j(\mu) \ell(\mu) \int_{\breve{\Omega}_j} \breve{w}(\breve{x}) \, d\breve{x}. \end{aligned} \tag{3.47}$$

Consequently, $J^f = 2$, and in (3.45), we have $\alpha_j^f(\mu) = h_j(\mu)\ell(\mu)$ and $\widehat{\mathbf{f}}_{j,p} = \int_{\breve{\Omega}_j} \breve{\theta}_p(\breve{x}) \, d\breve{x}$, for all $j \in \{1, 2\}$.

3.4.4 Separation of the constraint

The remaining bottleneck is the computation of $\widehat{\mathbf{K}}(\mu, \hat{\mathbf{v}}(\mu))$ and $\widehat{\mathbf{g}}(\mu, \hat{\mathbf{v}}(\mu))$ in (3.42b) and (3.43b) which requires parameter-dependent reconstructions using the FEM basis functions in order to compute the integrals over $\Gamma_1^c(\mu)$.

The key idea is to introduce a further approximation. Namely, we search for approximations κ_{M^k} and γ_{M^g} of the nonlinear maps $\kappa : \mathcal{P} \times \{1, \dots, \mathcal{N}\} \times \breve{\Gamma}_1^c \to \mathbb{R}$ and $\gamma : \mathcal{P} \times \breve{\Gamma}_1^c \to \mathbb{R}$ defined such that

$$\kappa(\mu, n, \breve{x}) := k(\mu, u(\mu); \phi_n)(h(\mu)(\breve{x})), \text{ and } \gamma(\mu, \breve{x}) := g(\mu, u(\mu))(h(\mu)(\breve{x})). \quad (3.48)$$

Our goal in building the approximations κ_{M^k} and γ_{M^g} is to separate the dependence on μ from the dependence on the other variables. More precisely, for some integers $M^k, M^g \geqslant 1$, we look for (accurate) approximations $\kappa_{M^k} : \mathcal{P} \times \{1, \dots, \mathcal{N}\} \times \breve{\Gamma}_1^c \to \mathbb{R}$ of κ and $\gamma_{M^g} : \mathcal{P} \times \breve{\Gamma}_1^c \to \mathbb{R}$ of γ in the separated form

$$\kappa_{M^k}(\mu, n, \breve{x}) := \sum_{j=1}^{M^k} \varphi_j^\kappa(\mu) q_j^\kappa(n, \breve{x}), \qquad \gamma_{M^g}(\mu, \breve{x}) := \sum_{j=1}^{M^g} \varphi_j^\gamma(\mu) q_j^\gamma(\breve{x}), \quad (3.49)$$

where M^k (resp. M^g) is called the rank of the approximation and $\varphi_j^\kappa(\mu)$ (resp. $\varphi_j^\gamma(\mu)$) are real numbers that are found by interpolation. For κ_{M^k}, we interpolate over a set of M^k pairs $\{(n_1^\kappa, \breve{x}_1^\kappa), \dots, (n_{M^k}^\kappa, \breve{x}_{M^k}^\kappa)\}$ in $\{1, \dots, \mathcal{N}\} \times \breve{\Gamma}_1^c$, whereas for γ_{M^g}, we interpolate over a set of M^g points $\{\breve{x}_1^\gamma, \dots, \breve{x}_{M^g}^\gamma\}$ in $\breve{\Gamma}_1^c$. The interpolation is performed using the EIM [5] and leads to the quantities $\hat{\boldsymbol{\kappa}}(\mu, \hat{\mathbf{v}}) \in \mathbb{R}^{M^k}$, $\mathbf{B}^\kappa \in \mathbb{R}^{M^k \times M^k}$, $\hat{\boldsymbol{\gamma}}(\mu, \hat{\mathbf{v}}) \in \mathbb{R}^{M^g}$ and $\mathbf{B}^\gamma \in \mathbb{R}^{M^g \times M^g}$ defined as follows:

$$\begin{cases} \hat{\boldsymbol{\kappa}}(\mu, \hat{\mathbf{v}})_i := \left(k\left(\mu, \hat{v}; \phi_{n_i}^\kappa\right)(h(\mu)(\breve{x}_i^\kappa))\right)_i, \\ \mathbf{B}_{ij}^\kappa = (q_j^\kappa(n_i^\kappa, \breve{x}_i^\kappa))_{ij}, \\ \hat{\boldsymbol{\gamma}}(\mu, \hat{\mathbf{v}})_i := \left(g\left(\mu, \hat{v}\right)(h(\mu)(\breve{x}_i^\gamma))\right)_i, \\ \mathbf{B}_{ij}^\gamma = (q_j^\gamma(\breve{x}_i^\gamma)). \end{cases} \quad (3.50)$$

Note that the EIM guarantees the invertibility of the matrices $\mathbf{B}^\kappa$ and $\mathbf{B}^\gamma$. The problem (3.41) becomes (we keep the same notation for its solution)

$$\begin{aligned} (\hat{\mathbf{u}}(\mu), \hat{\boldsymbol{\lambda}}(\mu)) \in \arg \underset{\hat{\mathbf{v}} \in \mathbb{R}^N, \hat{\boldsymbol{\eta}} \in \mathbb{R}_+^R}{\operatorname{minmax}} \Big\{ & \frac{1}{2} \hat{\mathbf{v}}^T \widehat{\mathbf{A}}(\mu) \hat{\mathbf{v}} - \hat{\mathbf{v}}^T \widehat{\mathbf{f}}(\mu) \\ & + \hat{\boldsymbol{\eta}}^T \left(\mathbf{D}^\kappa(\mu, \hat{\mathbf{v}}) \hat{\mathbf{v}} - \mathbf{D}^\gamma(\hat{\mathbf{v}}) \hat{\boldsymbol{\gamma}}(\mu, \hat{\mathbf{v}}) \right) \Big\}, \end{aligned} \quad (3.51)$$

with the matrices

$$\mathbf{D}^\kappa(\mu, \hat{\mathbf{v}}) = \sum_{j=1}^{M^k} \mathbf{C}_j^\kappa ((\mathbf{B}^\kappa)^{-1} \hat{\boldsymbol{\kappa}}(\mu; \hat{\mathbf{v}}))_j, \quad \text{and} \quad \mathbf{D}^\gamma = \mathbf{C}^\gamma (\mathbf{B}^\gamma)^{-1}, \quad (3.52)$$

where $\mathbf{C}_j^\kappa \in \mathbb{R}^{R \times N}$ and $\mathbf{C}^\gamma \in \mathbb{R}^{R \times M^g}$ are given by

$$\mathbf{C}_{j,pn}^\kappa = \left(\sum_{i=1}^{\mathcal{N}} \int_{\breve{\Gamma}_1^c} \breve{\theta}_{n,i} q_j^\kappa(i, \cdot) \breve{\xi}_p \right)_{pn}, \quad \text{and} \quad \mathbf{C}_{pj}^\gamma = \left(\int_{\breve{\Gamma}_1^c} q_j^\gamma \breve{\xi}_p \right)_{pj}. \quad (3.53)$$

The overall computational procedure can now be split into two stages:

(i) An offline stage where one precomputes on the one hand the RB subspace $\widehat{V}_N$ and the RB subcone $\widehat{W}_R$ leading to the vectors $\{\widehat{\mathbf{f}}_r\}_{1\leqslant j\leqslant J^a}$ in $\mathbb{R}^N$ and the matrices $\{\widehat{\mathbf{A}}_r\}_{1\leqslant j\leqslant J^f}$ in $\mathbb{R}^{N\times N}$, and on the other hand the EIM pairs $\{(n_i^\kappa, \breve{x}_i^\kappa)\}_{1\leqslant i\leqslant M^k}$, the EIM points $\{\breve{x}_i^\gamma\}_{1\leqslant i\leqslant M^g}$, the EIM functions $\{q_j^\kappa\}_{1\leqslant j\leqslant M^k}$, and the EIM functions $\{q_j^\gamma\}_{1\leqslant j\leqslant M^g}$, leading to the matrices $\mathbf{B}^\kappa \in \mathbb{R}^{M^k\times M^k}$, $\mathbf{B}^\gamma \in \mathbb{R}^{M^g\times M^g}$, $\{\mathbf{C}_j^\kappa\}_{1\leqslant j\leqslant M^k}$ in $\mathbb{R}^{R\times N}$ and $\mathbf{C}^\gamma \in \mathbb{R}^{R\times M^g}$. The offline stage is discussed in more detail in Section 3.5.

(ii) An online stage to be performed each time one wishes to compute a new solution for a parameter $\mu \in \mathcal{P}$. All that remains to be performed is to assemble the vector $\widehat{\mathbf{f}}(\mu) \in \mathbb{R}^N$ and the matrix $\widehat{\mathbf{A}}(\mu) \in \mathbb{R}^{N\times N}$ using (3.45), to compute the vectors $\widehat{\boldsymbol{\kappa}}(\mu, \hat{\mathbf{v}}) \in \mathbb{R}^{M^k}$ and $\widehat{\boldsymbol{\gamma}}(\mu, \hat{\mathbf{v}}) \in \mathbb{R}^{M^g}$ defined in (3.50), to assemble the matrix $\mathbf{D}^\kappa(\mu, \hat{\mathbf{v}})$ defined in (3.52), and to solve the reduced saddle-point problem (3.51). The online stage is summarized in Algorithm 3.1.

Algorithm 3.1 Online stage

Input : μ, $\{\widehat{\mathbf{f}}_j\}_{1\leqslant j\leqslant J^f}$, $\{\widehat{\mathbf{A}}_j\}_{1\leqslant j\leqslant J^a}$, $\{(n_i^\kappa, \breve{x}_i^\kappa)\}_{1\leqslant i\leqslant M^k}$, $\{\breve{x}_i^\gamma\}_{1\leqslant i\leqslant M^g}$, $\{q_j^\kappa\}_{1\leqslant j\leqslant M^k}$, $\{q_j^\gamma\}_{1\leqslant j\leqslant M^g}$, $\mathbf{B}^\kappa$, $\{\mathbf{C}_j^\kappa\}_{1\leqslant j\leqslant M^k}$ and $\mathbf{D}^\gamma$.

1: Assemble the vector $\widehat{\mathbf{f}}(\mu)$ and the matrix $\widehat{\mathbf{A}}(\mu)$ using (3.45)
2: Compute $\widehat{\boldsymbol{\kappa}}(\mu, \hat{\mathbf{v}})$ and $\widehat{\boldsymbol{\gamma}}(\mu, \hat{\mathbf{v}})$ using (3.50)
3: Compute $\mathbf{D}^\kappa(\mu)$ using $\widehat{\boldsymbol{\kappa}}(\mu, \hat{\mathbf{v}})$ and (3.52)
4: Solve the reduced saddle-point problem (3.51) to obtain $\hat{\mathbf{u}}(\mu)$ and $\widehat{\boldsymbol{\lambda}}(\mu)$

Output : $\hat{\mathbf{u}}(\mu)$ and $\widehat{\boldsymbol{\lambda}}(\mu)$

Remark 3.4 (EIM matrix)**.** *The computations in Algorithm 3.1 only require the knowledge of the matrix* $(\mathbf{B}^\kappa)^{-1}$*. In order to optimize the computational costs,* $\mathbf{B}^\kappa$ *is inverted prior to the algorithm, i.e. during the offline stage. The matrix* $\mathbf{B}^\gamma$ *is also inverted when computing the matrix* $\mathbf{D}^\gamma$ *(see (3.52)), which is an input of the offline stage.*

Remark 3.5 (EIMs on k and g)**.** *Owing to the quasi-linear structure of the inequality constraint, the reduced problem (3.51) is solved using the Kačanov algorithm. At first glance, the influence of this resolution choice is that we have to perform the EIM twice since the maps* k *and* g *are separated one at a time. Were we to use a standard Newton method by considering the one-term constraint* $\zeta(\mu, u(\mu)) \leqslant 0$ *(see (3.2)), we would only perform a single EIM. However, an additional EIM would be needed in the Newton method in order to compute the Jacobian preconditioning matrix. Thus, both options (Kačanov or Newton) lead to two distinct EIMs and the storage cost is essentially the same.*

3.5 The offline stage

There are two tasks to be performed during the offline stage:

(T_1) Build the rank-M^k and the rank-M^g EIM approximations in (3.49);

(T_2) Explore the solution manifold in order to construct the linear subspace $\check{V}_N \subset \check{V}_{\mathcal{N}}$ of dimension N and the subcone $\widetilde{W}_R \subset \widetilde{W}_{\mathcal{R}}$ of dimension R.

Tasks (T_1) and (T_2) can be performed independently and in whatever order. Since Task (T_1) can be considered to be standard, we only discuss Task (T_2), i.e., the construction of the sets of primal and dual RB functions with cardinalities N and R respectively. First, as usual in RB methods, the solution manifold is explored by considering a training set for the parameter values. For simplicity, one can consider the same training set $\mathcal{P}^{\mathrm{tr}}$ as for the EIM approximations. This way, one only explores the collection of points $\mathcal{S}_{\mathrm{PRI}} = \{u(\mu)\}_{\mu\in\mathcal{P}^{\mathrm{tr}}}$ and $\mathcal{S}_{\mathrm{DU}} = \{\lambda(\mu)\}_{\mu\in\mathcal{P}^{\mathrm{tr}}}$ respectively in the primal and dual solution manifolds. For this exploration to be informative, the training set $\mathcal{P}^{\mathrm{tr}}$ has to be chosen large enough. In the present setting where HF solutions are to be computed for all the parameters in $\mathcal{P}^{\mathrm{tr}}$ when constructing the EIM approximations, it is natural to compress these computations by means of a Proper Orthogonal Decomposition (POD) [31, 35] to define the primal RB subspace $\check{V}_N$. This technique is often considered in the literature to build the RB, see, e.g., [26, 30, 48]. Bearing in mind that the dual RB cone $\widetilde{W}_R$ is meant to represent the set of Lagrange multipliers, its spanning vectors should all have non-negative components. Consequently, the POD is not appropriate to build the dual RB cone. If the training set has a reasonable size, one could keep all the snapshots, especially if they have been computed via *a posteriori* error estimation. In [4], it is suggested to use the Non-negative Matrix Factorization (NMF) algorithm [36] whenever the number of training snapshots is relatively large, for instance in the case of a time-dependent problem. For a set of positive snapshots $\mathcal{S}_{\mathrm{DU}}$ and an integer R, the procedure $\mathrm{NMF}(\mathcal{S}_{\mathrm{DU}}, R)$ returns R vectors $(w_1, \dots, w_R)$ with non-negative components (cf. Section 3.7.2). Nonetheless, the resulting dual RB can be less accurate than the primal RB and the user does not specify a required error tolerance as an input but only the cardinality of the dual RB. In practice, it is often difficult to anticipate the approximation capacity of the dual RB from its cardinality.

Here, we suggest to build a dual hierarchical basis from the Lagrange multiplier snapshots computed offline. In the spirit of weak greedy algorithms, the idea is to order the snapshots depending on their relevance to represent the entire set of snapshots. For all $\mu \in \mathcal{P}^{\mathrm{tr}}$, we define the reference Lagrange multiplier snapshot

$$\check{\lambda}(\mu;\cdot) := \lambda(\mu;\cdot) \circ h(\mu) : \check{\Gamma}_1^c \to \mathbb{R}. \tag{3.54}$$

First, we choose $\mu_1 \in \mathcal{P}^{\mathrm{tr}}$ such that

$$\mu_1 \in \operatorname*{argmax}_{\mu\in\mathcal{P}^{\mathrm{tr}}} \|\check{\lambda}(\mu;\cdot)\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}, \tag{3.55}$$

where the discrete subset $\check{\Gamma}_1^{c,\mathrm{tr}} \subsetneq \check{\Gamma}_1^c$ is introduced to compute the maximizer in space. Afterwards, at each iteration $n \geqslant 2$, we define the convex cone $\check{K}_n = \mathrm{span}_+\{\lambda(\mu_1,\cdot)\circ h(\mu), \dots, \lambda(\mu_{n-1},\cdot)\circ h(\mu)\}$ and select a new parameter value $\mu_n \in \mathcal{P}^{\mathrm{tr}}$ using the

criterion

$$\mu_n \in \underset{\mu\in\mathcal{P}^{\mathrm{tr}}}{\operatorname{argmax}} \|\check{\lambda}(\mu;\cdot) - \Pi_{\check{K}_{n-1}}(\check{\lambda}(\mu;\cdot))\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}, \tag{3.56}$$

where $\Pi_{\check{K}_{n-1}}$ is the orthogonal projector onto the convex cone $\check{K}_{n-1}$. At each iteration, we check whether or not the stopping criterion

$$\max_{\mu\in\mathcal{P}^{\mathrm{tr}}} \|\check{\lambda}(\mu;\cdot) - \Pi_{\check{K}_{n-1}}(\check{\lambda}(\mu;\cdot))\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})} \leqslant \epsilon_{\mathrm{DU}}, \tag{3.57}$$

is fulfilled. The steps of the 'cone-projected' weak greedy algorithm are summarized in Algorithm 3.2.

Algorithm 3.2 Cone-projected weak greedy algorithm

Input : $\mathcal{P}^{\mathrm{tr}}$, $\check{\Gamma}_1^{c,\mathrm{tr}}$ and $\epsilon_{\mathrm{DU}} > 0$

1: Compute $\mathcal{S}_{\mathrm{DU}} = \{\check{\lambda}(\mu;\cdot)\}_{\mu\in\mathcal{P}^{\mathrm{tr}}}$ # `HF solutions`
2: Set $\check{K}_0 = \{0\}$
3: Set $n = 1$
4: Set $r_1 = \max_{\mu\in\mathcal{P}^{\mathrm{tr}}} \|\check{\lambda}(\mu;\cdot)\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}$
5: **while** $(r_n > \epsilon_{\mathrm{DU}})$ **do**
6: Search $\mu_n \in \underset{\mu\in\mathcal{P}^{\mathrm{tr}}}{\operatorname{argmax}} \|\check{\lambda}(\mu;\cdot) - \Pi_{\check{K}_{n-1}}(\check{\lambda}(\mu;\cdot))\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}$
7: Set $\check{K}_n := \mathrm{span}_+\{\check{\lambda}(\mu_1;\cdot), \ldots, \check{\lambda}(\mu_n;\cdot)\}$
8: Set $n = n + 1$
9: Set $r_n := \max_{\mu\in\mathcal{P}^{\mathrm{tr}}} \|\check{\lambda}(\mu;\cdot) - \Pi_{\check{K}_{n-1}}(\check{\lambda}(\mu;\cdot))\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}$
10: **end while**
11: Set $R := n - 1$

Output : $\check{W}_R := \check{K}_R$.

Remark 3.6 (Cone projections)**.** *The projection onto the cone $\check{K}_{n-1}$ in line 6 of Algorithm 3.2 is not trivial. We use the off-the-shelf solver from [2].*

Remark 3.7 (Relative error)**.** *Algorithm 3.2 can be run using a relative error criterion instead of an absolute one. Towards this end, one replaces the absolute error $\|\check{\lambda}(\mu;\cdot) - \Pi_{\check{K}_{n-1}}(\check{\lambda}(\mu;\cdot))\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}$ in lines 6 and 9 by the relative error*

$$\|\check{\lambda}(\mu;\cdot) - \Pi_{\check{K}_{n-1}}(\check{\lambda}(\mu;\cdot))\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})} / \|\check{\lambda}(\mu;\cdot)\|_{\ell^\infty(\check{\Gamma}_1^{c,\mathrm{tr}})}. \tag{3.58}$$

Remark 3.8 (Elementary compression)**.** *For the reduced problem, if we choose to conserve all the snapshots of the Lagrange multipliers, we can still check for some computational savings. This can be achieved by suppressing the constraints that are never saturated for any of the parameters in the training set $\mathcal{P}^{\mathrm{tr}}$ but were initially introduced in the HF model. In practice, we reduce the dimensions of the matrix $\mathbf{K}(\mu, \mathbf{u}(\mu))$ and the vector $\mathbf{g}(\mu, \mathbf{u}(\mu))$ by removing the lines and columns of $\mathbf{K}(\mu, \mathbf{u}(\mu))$ and the components of $\mathbf{g}(\mu, \mathbf{u}(\mu))$ that always vanish no matter the value of the parameter $\mu \in \mathcal{P}^{\mathrm{tr}}$.*

3.6 Numerical results

In this section, we illustrate the above developments by a numerical example related to elastic contact in a two-dimensional framework. The investigated test case is the seminal contact problem between two spheres introduced by Hertz [29]. The HF computations use the software `Freefem++` [28] in order to generate the mesh and the matrices and vectors of the saddle-point problem. The HF solution of the saddle-point problem is then retrieved using `Python` and its convex optimization package `cvxopt` [2]. The RB algorithms are developed in `Python`. Our goal is to illustrate the computational performance of the method. We choose a Galerkin discretization for the primal solution and a collocation method for the contact nodes. We introduce a reference mesh grid $\breve{\Omega}^{\mathrm{tr}} \subset \breve{\Omega}$ with $\mathrm{Card}(\breve{\Omega}^{\mathrm{tr}}) = \mathcal{N}$ and the corresponding finite element test functions $\breve{\phi}_{j,x} = (\breve{\phi}_j, 0)$ and $\breve{\phi}_{j,y} = (0, \breve{\phi}_j)$ for $d = 2$ (the extension being straightforward for $d = 3$). Also, we introduce a set of ordered sampling points $\breve{\Gamma}_1^{c,\mathrm{tr}} := \{\breve{z}_1, \cdots, \breve{z}_{\mathcal{R}}\} \subset \breve{\Gamma}_1^c \cap \breve{\Omega}^{\mathrm{tr}}$ with $\mathrm{Card}(\breve{\Gamma}_1^{c,\mathrm{tr}}) = \mathcal{R}$. By means of collocation over this set of points, the expressions of the matrices $\mathbf{C}_j^\kappa$ and $\mathbf{C}^\gamma$ in (3.52) and (3.53) are recast as

$$\begin{cases} \mathbf{C}_j^\kappa = \widehat{\mathbf{W}}_R \mathbf{Q}_j^\kappa, \quad \text{where} \quad \mathbf{C}_j^\kappa = \left(\sum_{i=1}^{\mathcal{N}} \int_{\Gamma_1^c} \theta_{n,i} q_j^\kappa(i, \breve{z}_p) \right)_{1 \leqslant p \leqslant R, 1 \leqslant n \leqslant N} \in \mathbb{R}^{R \times N}, \\ \mathbf{C}^\gamma = \widehat{\mathbf{W}}_R \mathbf{Q}^\gamma, \quad \text{where} \quad \mathbf{Q}^\gamma = \left(q_j^\gamma(\breve{z}_p) \right)_{1 \leqslant p \leqslant \mathcal{R}, 1 \leqslant j \leqslant M^g} \in \mathbb{R}^{R \times N}, \end{cases} \tag{3.59}$$

where $\widehat{\mathbf{W}}_R \in R \times \mathcal{R}$ is the matrix of dual snapshots.

Consider the parameter set $\mathcal{P} = [0.9, 1.12]$, the discrete training set $\mathcal{P}^{\mathrm{tr}} = \{0.905 + 0.01i \mid 0 \leqslant i \leqslant 22\}$ and a two-dimensional setting based on two half-disks with an upper constant radius $R_{\mathrm{up}} = 1\mathrm{m}$, a lower parametric radius $R_{\mathrm{low}} = \mu$ and an initial gap between the centers of the disks $\gamma_0 = 0.1m$. The left panel of Figure 3.2 shows the reference domain $\breve{\Omega} = \Omega(\mu = 1)$. HF solutions are computed using a finite element subspace defined on a mesh of Ω^{tr} with 675 nodes consisting of continuous, piecewise affine functions and the potential contact zone contains 51 nodes on each disk. Consequently, our problem has $\mathcal{N} = 1350$ degrees of freedom and $\mathcal{R} = 51$ Lagrange multipliers. The materials of both hald-disks are identical and the HF computations are run with a Young modulus $E = 15\mathrm{Pa}$ and a Poisson coefficient $\nu = 0.35$. Regarding boundary conditions, we consider a homogeneous Dirichlet condition $u_x = 0$ and $u_y = 0$ on the lower horizontal edge, a homogeneous Dirichlet condition $u_x = 0$ on the upper horizontal edge, and an imposed displacement $u_y = -0.4\mathrm{m}$ on the upper horizontal edge. The right panel of Figure 3.2 displays the HF displacement field for the parameter value $\mu = 1.12\mathrm{m}$, whereas Figure 3.3 displays the normal contact stress as a function of the reference configuration's abscissas for the parameter values $\mu \in \mathcal{P}^{\mathrm{tr}}$. The normal contact is zero on the nodes where the contact between the two half-disks is not established at equilibrium.

During the offline stage, we perform $P = 23$ HF computations. Applying the POD to $\mathcal{S}_{\mathrm{PRI}}$ based on the energy norm and an absolute truncation threshold $\epsilon_{\mathrm{POD}} = 10^{-3}$, the primal space $\breve{V}_N$ is composed of 11 RB functions. Table 3.1 shows the

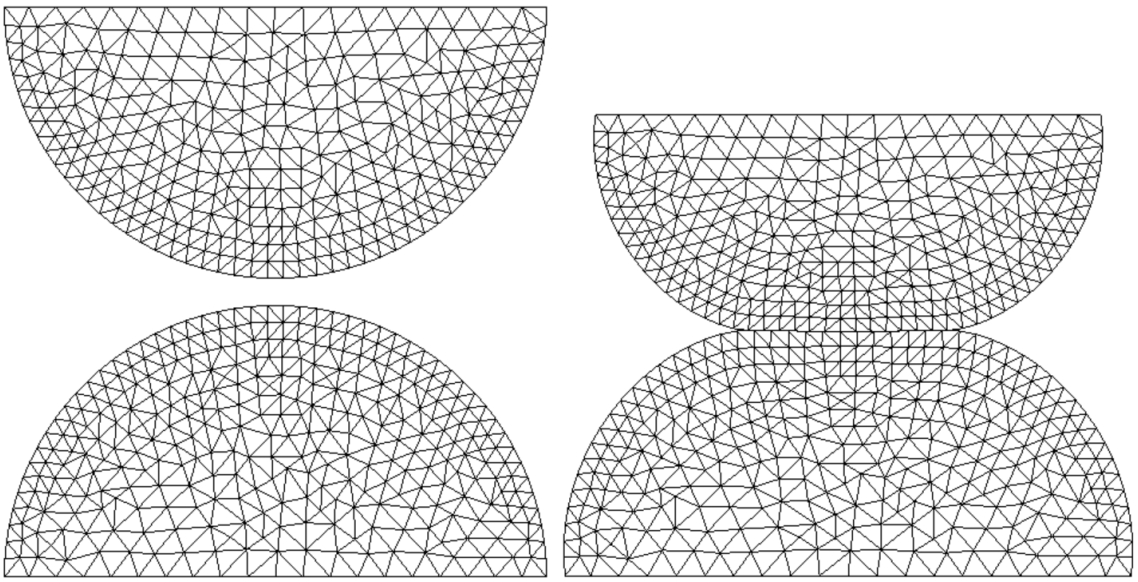

Figure 3.2 – Left: Reference domain $\breve{\Omega}$ and mesh with $\mathcal{N} = 1350$. Right: HF displacement field $u_{\mathcal{N}}(\mu)$ for $\mu = 1.12$m.

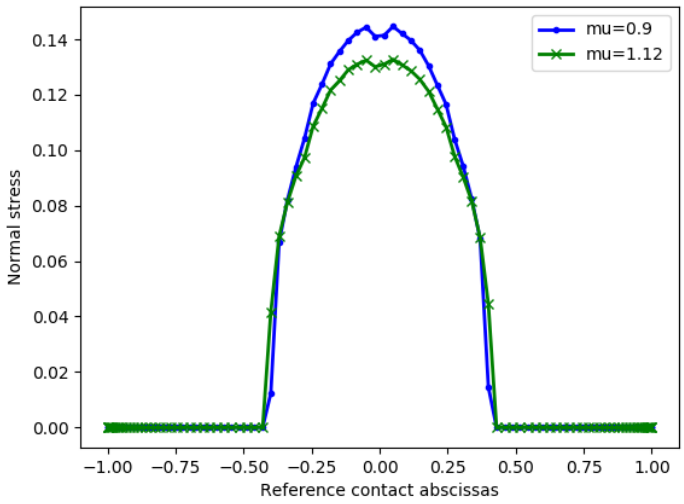

Figure 3.3 – Normal contact stress components for the HF solutions. Vanishing values correspond to nodes where the contact between the two disks is not established at equilibrium.

size of the reduced basis as a function of the tolerance ϵ_{POD}. As can be seen, the number of offline computations is equal to the dimension of the primal space for $\epsilon_{\text{POD}} = 4.10^{-6}$. The left panel of Figure 3.4 illustrates the decrease of the singular values associated with the POD modes. The decrease is not as sharp as is often the case for variational equalities. Moreover, the higher the rank of the singular value, the milder the decrease of the error. In order to build the dual reduced basis, we test both the NMF suggested in [4] and the cone-projected algorithm (see Algorithm 3.2). Table 3.2 shows the dimension R of the dual space as a function of the truncation threshold ϵ_{DU} for both algorithms. The cone-projected greedy algorithm achieves

ϵ_{POD}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	4.10^{-6}
N	5	11	16	20	22

Table 3.1 – Primal basis dimension N as a function of the truncation threshold ϵ_{POD}.

ϵ_{DU}		$5 \cdot 10^{-2}$	10^{-2}	$5 \cdot 10^{-3}$	10^{-3}
NMF	R	4	10	19	20
Cone-projected greedy	R	2	5	6	13

Table 3.2 – Dual basis dimension R as a function of the truncation threshold ϵ_{DU}.

the same accuracies with less basis function than the NMF. Notice that, at the first iterations of the procedure, the NMF uses at least twice as much functions as the cone-projected greedy algorithm. The right panel of Figure 3.4 displays the approximation error for the dual $\breve{W}_R$ space as its dimension R increases. This figure clearly shows that the cone-projected greedy algorithm outperforms the NMF. Note that it is actually pointless to perform an NMF with an input number of modes equal to the total number of modes. The same reasoning applies to the cone-projected greedy algorithm as well. However, as the cone-projected greedy algorithm is (in principle) steered by an accuracy threshold rather than a number of modes, the case in which all the modes are retained can still be justified. Hence, the right panel of figure 3.4 is meant to compare the performances of both algorithms at convergence. We now perform the EIM twice so as to allow for an offline/online decomposition

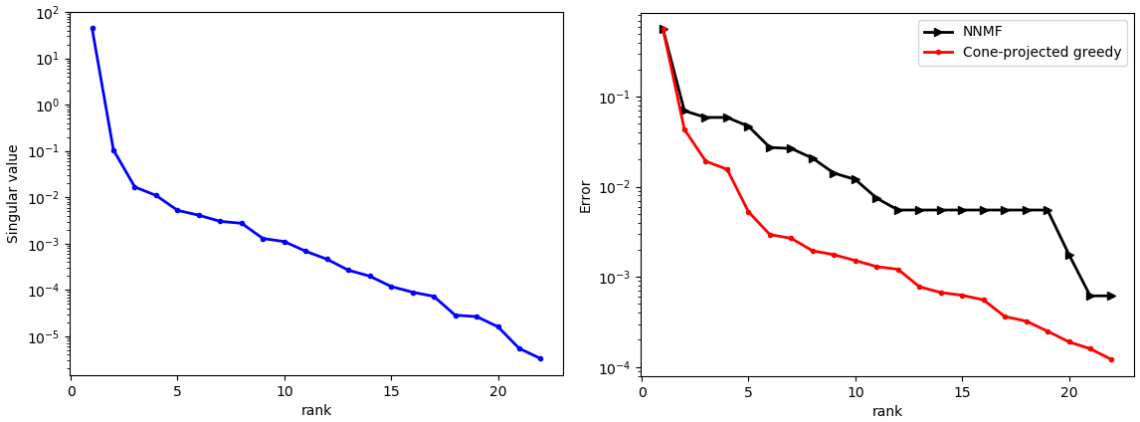

Figure 3.4 – Offline basis construction. Left: Singular values resulting from the POD for the primal space $\breve{V}_N$. Right: Approximation error for the dual space $\breve{W}_R$.

of both terms in the inequality constraint. The convergence of the EIM is reported in Figure 3.5. The approximation error decreases clearly faster for the nonlinear gap function $\mathbf{g}(\mu, \mathbf{u}(\mu))$ than for the nonlinear contact operator $\mathbf{K}(\mu, \mathbf{u}(\mu))$. This observation is not counter-intuitive since the contact map is a trivariate function, whereas the gap map is a bivariate function. Therefore, the EIM on $\mathbf{K}(\mu; \mathbf{u})$ needs a higher rank M^k to be accurate.

Let us now investigate the online stage for the entries $\epsilon_{\text{DU}} = 10^{-4}$, $\epsilon^k_{\text{EIM}} = 10^{-2}$,

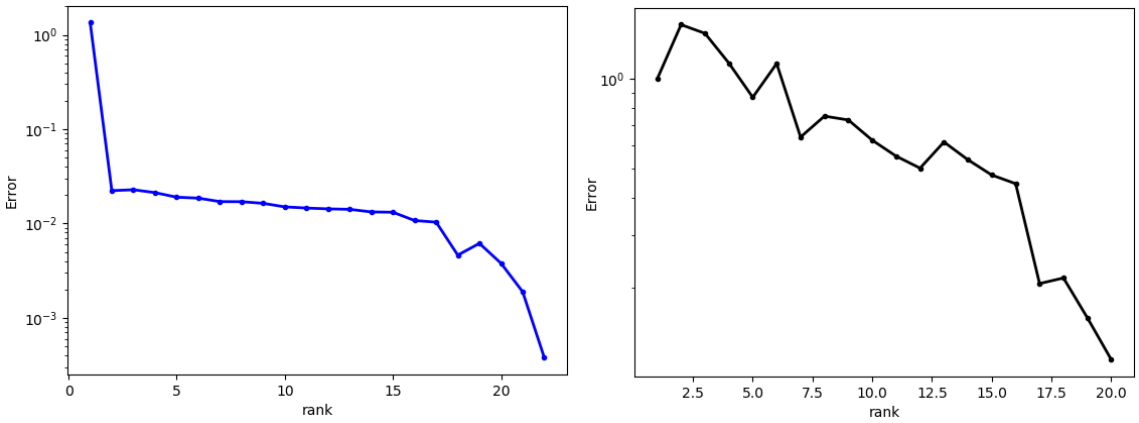

Figure 3.5 – EIM error as a function of the rank M of the EIM approximation. Left: for the nonlinear gap map $\mathbf{g}(\mu, \mathbf{u}(\mu))$. Right: for the nonlinear contact map $\mathbf{K}(\mu, \mathbf{u}(\mu))$.

$\epsilon^g_{\text{EIM}} = 10^{-3}$ and $\epsilon_{\text{POD}} = 10^{-5}$. We define the error on the minimum energy as follows:

$$e_{\text{ener}}(\mu) := \frac{1}{2}|a(\mu, \widehat{u}(\mu), \widehat{u}(\mu)) - f(\mu, \widehat{u}(\mu)) - a(\mu, u(\mu), u(\mu)) + f(\mu, u(\mu))|. \quad (3.60)$$

The left panel of Figure 3.6 displays the error $e_{\text{ener}}(\mu)$. One can notice that the error

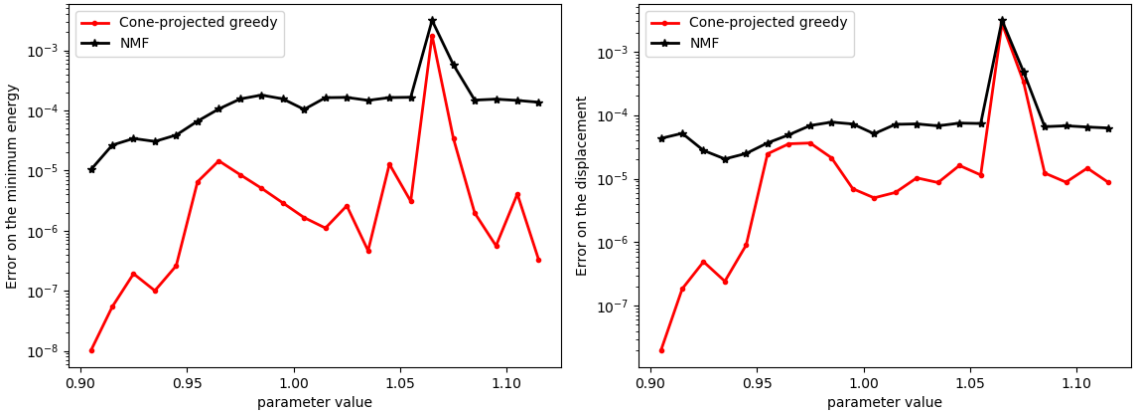

Figure 3.6 – Left: Error on the minimum energy $e_{\text{ener}}(\mu)$. Right: Relative H^1-error for the displacement field $e_{\text{displ}}(\mu)$.

for the cone-projected greedy algorithm is always below that of the NMF. Moreover, the right panel of Figure 3.6 shows the relative H^1-error error on the displacement field defined as follows:

$$e_{\text{displ}}(\mu) := \frac{\|\widehat{u}(\mu) - u_{\mathcal{N}}(\mu)\|_{H^1(\Omega^{\text{tr}})}}{\|u_{\mathcal{N}}(\mu)\|_{H^1(\Omega^{\text{tr}})}}. \quad (3.61)$$

The conlusion is similar as for $e_{\text{ener}}(\mu)$. Finally, Figure 3.7 displays a quantification of the interpenetration, i.e., the violation of the inequality constraint, by means of the error indicator

$$e_{\text{inter}}(\mu) := \sqrt{\int_{\Gamma_1^c} \min\big(0, k(\mu, \widehat{u}(\mu); \widehat{u}(\mu)) - g(\mu, \widehat{u}(\mu))\big)^2}. \quad (3.62)$$

For low parameter values, there is no interpenetration, but the conclusion is reversed

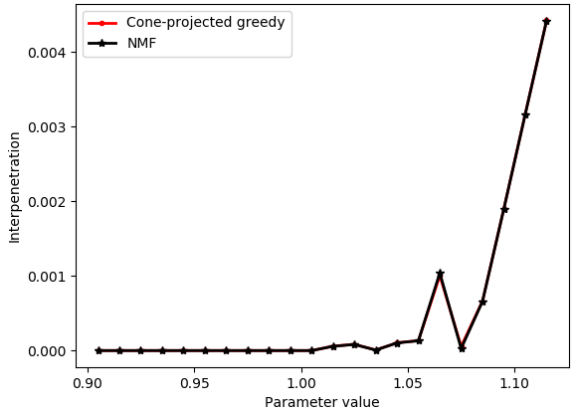

Figure 3.7 – Estimation of the interpenetration $e_{\text{inter}}(\mu)$.

for larger parameter values, i.e. $\mu \in \{1.01, \ldots, 1.12\}$. The reason for this is that the spatial discretization becomes coarser with the increase of the parameter value. Notice that the interpenetration curves for the NMF and the cone-projected greedy almost overlap.

As a second test, we run both methods with the looser tolerances $\epsilon_{\text{DU}} = 1 \cdot 10^{-2}$, $\epsilon^k_{\text{EIM}} = 2 \cdot 10^{-1}$, $\epsilon^g_{\text{EIM}} = 10^{-2}$ and $\epsilon_{\text{POD}} = 10^{-2}$. Figure 3.8 displays the errors $e_{\text{ener}}(\mu)$ and $e_{\text{displ}}(\mu)$ using only $R = 4$ basis vectors for the cone-projected greedy and $R = 10$ basis vectors for the NMF. The overall error is larger than in the previous cases since all the tolerances have been loosened.

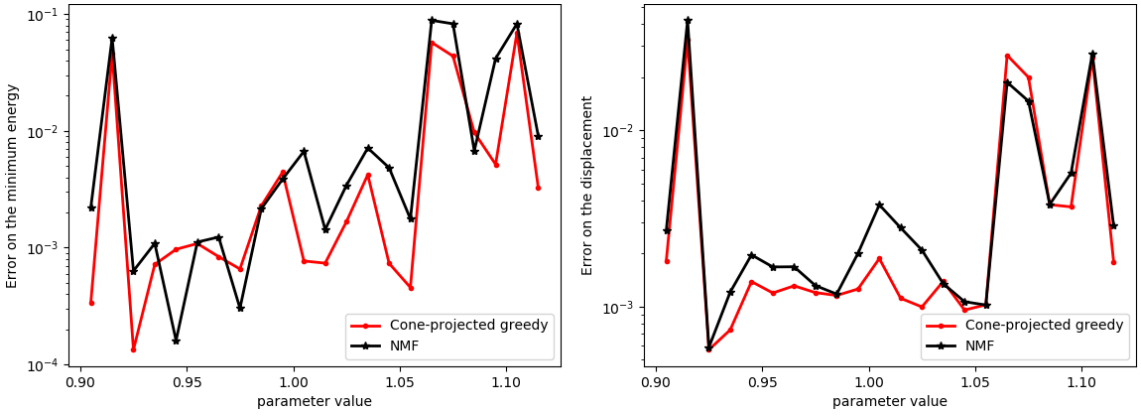

Figure 3.8 – Error quantification with $R = 10$ for the NMF and $R = 4$ for the cone-projected greedy. Left: Error on the minimum energy $e_{\text{ener}}(\mu)$. Right: Relative H^1-error for the displacement field $e_{\text{displ}}(\mu)$.

In order to get a clearer insight on the impact of the dual space $\widetilde{W}_R$, we display the same plots as in the first simulation with the larger truncation threshold $\epsilon_{\text{DU}} = 5 \cdot 10^{-3}$. The other tolerances are $\epsilon^k_{\text{EIM}} = 10^{-2}$, $\epsilon^g_{\text{EIM}} = 10^{-4}$ and $\epsilon_{\text{POD}} = 10^{-4}$. In this configuration, the NMF conserves $R = 18$ dual basis vectors, whereas the cone-projected greedy only conserves $R = 5$ dual basis vectors. The minimum energy

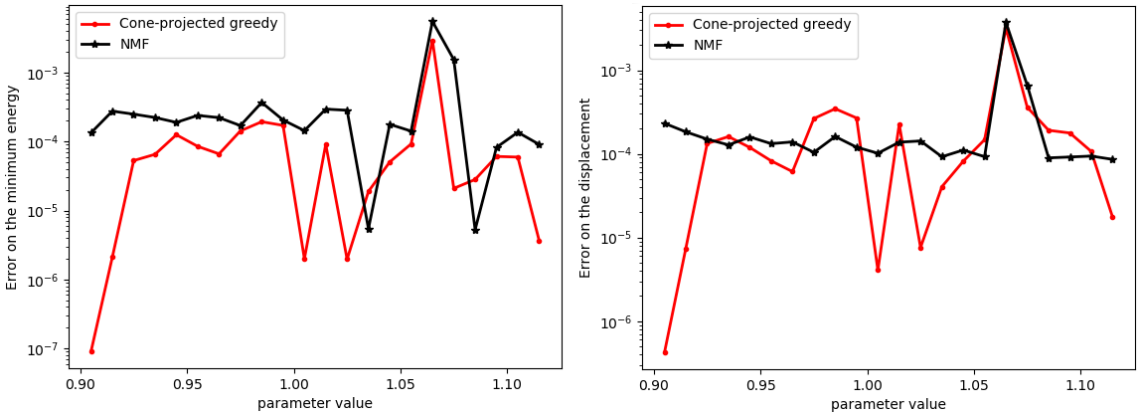

Figure 3.9 – Error quantification with $R = 18$ for the NMF and $R = 5$ for the cone-projected greedy. Left: Error on the minimum energy $e_{\text{ener}}(\mu)$. Right: Relative H^1-error for the displacement field $e_{\text{displ}}(\mu)$.

error $e_{\text{ener}}(\mu)$ and the relative H^1-error $e_{\text{displ}}(\mu)$ are plotted in Figure 3.9. In spite of the substantial difference between the sizes of the NMF space and the cone-projected greedy space, Figure 3.9 shows that the cone-projected greedy algorithm still delivers accurate approximations and, in average, produces smaller errors. For the comparison between the two algorithms to be fairer, we keep $R = 5$ basis vectors for the NMF and display the error indicators $e_{\text{ener}}(\mu)$ and $e_{\text{displ}}(\mu)$ in Figure 3.10. In this situation, the error for the cone-projected greedy algorithm is always below

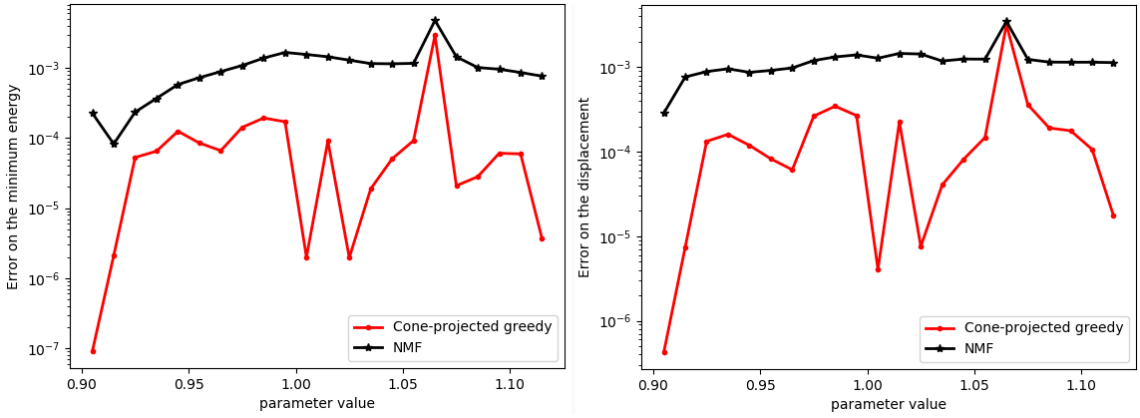

Figure 3.10 – Error quantification with $R = 5$ for both algorithms. Left: Error on the minimum energy $e_{\text{ener}}(\mu)$. Right: Relative H^1-error for the displacement field $e_{\text{displ}}(\mu)$.

that of the NMF. The quantification of the interpenetration displayed in Figure 3.11 corroborates the previous comment.

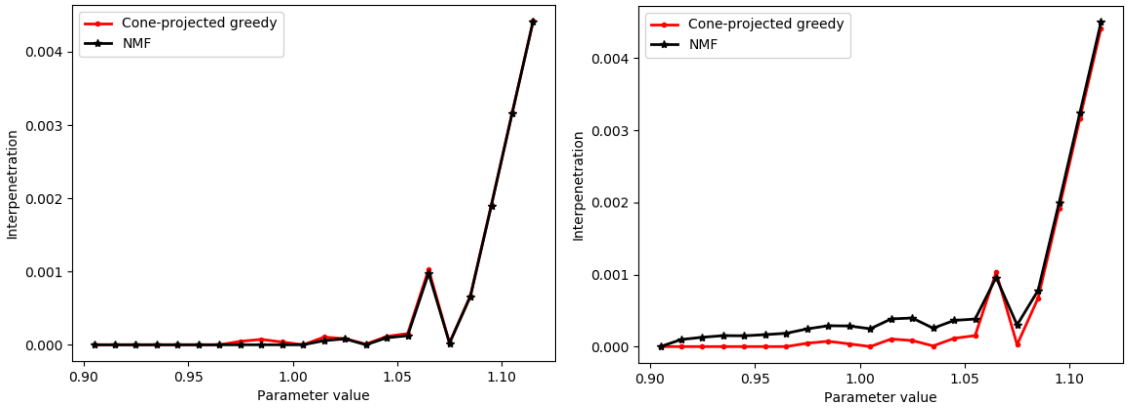

Figure 3.11 – Estimation of the interpenetration $e_{\text{inter}}(\mu)$. Left: $R = 18$ for the NMF and $R = 5$ for the cone-projected greedy. Right: $R = 5$ for both algorithms.

3.7 Technical complements

This section gathers three technical complements that give further insight on some of the aspects discussed within this chapter.

3.7.1 Kačanov vs. Newton

Suppose that we need to solve numerically the nonlinear problem: Find $\mathbf{u} \in \mathbb{R}^N$ such that $\mathbf{F}(\mathbf{u}) = \mathbf{0}$, where $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$. Given an initialization $\mathbf{u}_0 \in \mathbb{R}^N$, the standard Newton iteration reads

$$D\mathbf{F}(\mathbf{u}_k)(\mathbf{u}_{k+1} - \mathbf{u}_k) = -\mathbf{F}(\mathbf{u}_k), \quad \forall k \geqslant 0. \tag{3.63}$$

If the nonlinear operator $\mathbf{F}$ is of the form $\mathbf{F}(\mathbf{v}) = \mathbf{A}(\mathbf{v})\mathbf{v} - \mathbf{g}(\mathbf{v})$, with $\mathbf{A} : \mathbb{R}^N \to \mathbb{R}^{N\times N}$ and $\mathbf{g} : \mathbb{R}^N \to \mathbb{R}^N$, the iteration (3.63) becomes

$$\big(D\mathbf{A}(\mathbf{u}_k) \cdot \mathbf{u}_k + \mathbf{A}(\mathbf{u}_k) - D\mathbf{g}(\mathbf{u}_k)\big)(\mathbf{u}_{k+1} - \mathbf{u}_k) = -\mathbf{A}(\mathbf{u}_k)\mathbf{u}_k + \mathbf{g}(\mathbf{u}_k), \tag{3.64}$$

leading to

$$\big(D\mathbf{A}(\mathbf{u}_k) \cdot \mathbf{u}_k - D\mathbf{g}(\mathbf{u}_k)\big)(\mathbf{u}_{k+1} - \mathbf{u}_k) + A(\mathbf{u}_k)\mathbf{u}_{k+1} = \mathbf{g}(\mathbf{u}_k). \tag{3.65}$$

Provided that $D\mathbf{A}(\mathbf{v}) \cdot \mathbf{v} - D\mathbf{g}(\mathbf{v}) \approx 0$, (3.65) leads to the Kačanov iteration

$$\mathbf{A}(\mathbf{u}_k)\mathbf{u}_{k+1} = \mathbf{g}(\mathbf{u}_k). \tag{3.66}$$

3.7.2 Non-negative Matrix Factorization (NMF)

Let us give some details on the NMF for completeness. The following results can be found in the standard literature on clustering [36]. The goal is to briefly describe the procedure associated with the notation

$$(\mathbf{w}_1, \dots, \mathbf{w}_R) = \text{NMF}(\mathbf{T}, R), \tag{3.67}$$

which is mentioned in Section 3.5, where we are given P vectors $(\mathbf{t}_1, \cdots, \mathbf{t}_P)$ forming the rectangular matrix $\mathbf{T} \in \mathbb{R}_+^{\mathcal{R}\times P}$ whose entries are all non-negative, and we are looking for R positive vectors forming the rectangular matrix $\mathbf{W} := (\mathbf{w}_1, \ldots, \mathbf{w}_R) \in \mathbb{R}_+^{\mathcal{R}\times R}$. We define the error function that quantifies the quality of the approximation of the matrix $\mathbf{A}$ by a matrix $\mathbf{B}$ as the Frobenius norm of the difference between $\mathbf{A}$ and $\mathbf{B}$:

$$\|\mathbf{A} - \mathbf{B}\|^2 := \sum_{ij} (\mathbf{A}_{ij} - \mathbf{B}_{ij})^2, \tag{3.68}$$

which clearly vanishes if and only if $\mathbf{A} = \mathbf{B}$. The NMF optimization problem then reads:

$$(\mathbf{W}, \mathbf{H}) = \underset{\substack{\tilde{\mathbf{W}}\in\mathbb{R}_+^{\mathcal{R}\times R} \\ \tilde{\mathbf{H}}\in\mathbb{R}_+^{R\times P}}}{\operatorname{argmin}} \|\mathbf{T} - \tilde{\mathbf{W}}\tilde{\mathbf{H}}\|^2, \tag{3.69}$$

where $\mathbf{W} \in \mathbb{R}_+^{\mathcal{R}\times R}$ and $\mathbf{H} \in \mathbb{R}_+^{R\times P}$. The function $\|\mathbf{T} - \mathbf{W}\mathbf{H}\|$ is not convex in both variables $\mathbf{W}$ and $\mathbf{H}$ together. Thus, only the recovery of local minima is feasible. Regarding the search algorithm, the decrease of the Frobenius norm is proven in [36] for the following iterative update rules

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij} \frac{(\mathbf{W}^T\mathbf{T})_{ij}}{(\mathbf{W}^T\mathbf{W}\mathbf{H})_{ij}}, \qquad \mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{(\mathbf{T}\mathbf{H}^T)_{ij}}{(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ij}}. \tag{3.70}$$

The first update rule is equivalent to a gradient descent algorithm

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij} + \eta_{ij}\big((\mathbf{W}^T\mathbf{T})_{ij} - (\mathbf{W}^T\mathbf{W}\mathbf{H})_{ij}\big), \tag{3.71}$$

with $\eta_{ij} = \mathbf{H}_{ij}/(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ij}$. The reasoning for $\mathbf{W}_{ij}$ is similar. The motivation for (3.70) is that if the pair $(\mathbf{W}, \mathbf{H})$ yields an exact reconstruction, i.e. $\mathbf{T} = \mathbf{W}\mathbf{H}$, then $(\mathbf{W}, \mathbf{H})$ is a fixed-point of the algorithm. Finally, note that, in contrast to the POD, the integer R is a required input for the NMF. Moreover, the uniqueness of the NMF is not guaranteed. In fact, any positive matrix $\mathbf{D} \in \mathbb{R}^{R\times R}$ satisfies $\mathbf{T} = \mathbf{W}\mathbf{D}\mathbf{D}^{-1}\mathbf{H}$, thereby leading to another NMF decomposition.

3.7.3 Proof of Lemma 3.1

Proof. We first prove the converse implication, i.e. (3.30) $\Longrightarrow$ (3.26). It is readily verified that

$$(3.30) \Longrightarrow \Upsilon_1^c(\mu, u_1(\mu)) \subset \complement_{\overline{\Omega}(\mu,u(\mu))}\overline{\Omega}_2(\mu, u_2(\mu)) \cup \Upsilon_2^c(\mu, u_2(\mu)), \tag{3.72}$$

where the union in (3.72) is disjoint. For all $z \in \Gamma_1^c(\mu)$, we have $\psi_1(\mu, u_1(\mu))(z) = z + u_1(\mu)(z) \in \Upsilon_1^c(\mu, u_1(\mu))$. Thus, the inclusion (3.72) implies that, for all $z \in \Gamma_1^c(\mu)$:

$$\psi_1(\mu, u_1(\mu))(z) \notin \overline{\Omega}_2(\mu, u_2(\mu)) \text{ or } \psi_1(\mu, u_1(\mu))(z) \in \Upsilon_2^c(\mu, u_2(\mu)). \tag{3.73}$$

By definition of $\vartheta(\mu, u(\mu))$, we have

$$\begin{aligned}
&\psi_1(\mu, u_1(\mu))(z) \notin \overline{\Omega}_2(\mu, u_2(\mu)) \\
&\quad \Longleftrightarrow \left(\psi_1(\mu, u_1(\mu))(z) - \left(\vartheta(\mu, u(\mu)) \circ \psi_1(\mu, u_1(\mu))\right)(z) \cdot \tilde{n}_2(\mu, u(\mu))(z)\right) > 0,
\end{aligned}$$

and

$$\psi_1(\mu,u_1(\mu))(z)\in\Upsilon_2^c(\mu,u_2(\mu))\Longleftrightarrow\left(\psi_1(\mu,u_1(\mu))-\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu))\right)(z)=0$$
$$\Longleftrightarrow\left(\left(\psi_1(\mu,u_1(\mu))-\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu))\right)(z)\cdot\widetilde{n}_2(\mu,u(\mu))(z)\right)=0.$$

The latter equivalence follows from the colinearity of the two vectors involved in the scalar product. Thus,

$$(3.30)\Longrightarrow\left(\left(\psi_1(\mu,u_1(\mu))-\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu))\right)(z)\cdot\widetilde{n}_2(\mu,u(\mu))(z)\right)\geqslant 0. \tag{3.74}$$

Moreover, by definition of $\psi_2(\mu,u_2(\mu))$, we have

$$\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu))=(\psi_2(\mu,u_2(\mu)))^{-1}\circ\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu))$$
$$+u_2(\mu)\circ(\psi_2(\mu,u_2(\mu)))^{-1}\circ\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu)),$$

leading to

$$\vartheta(\mu,u(\mu))\circ\psi_1(\mu,u_1(\mu))=\rho(\mu,u(\mu))+u_2(\mu)\circ\rho(\mu,u(\mu)).$$

A straightforward replacement in (3.74) yields that, for all $z\in\Gamma_1^c(\mu)$,

$$\left(\psi_1(\mu,u_1(\mu))-u_2(\mu)\circ\rho(\mu,u(\mu))-\rho(\mu,u(\mu))(\,z)\cdot\widetilde{n}_2(\mu,u(\mu))(z)\geqslant 0.\right. \tag{3.75}$$

Using the definition of $\psi_1(\mu,u_1(\mu))$, (3.75) yields, for all $z\in\Gamma_1^c(\mu)$,

$$\left(u_1(\mu)-u_2(\mu)\circ\rho(\mu,u(\mu))\right)(z)\cdot n_2(\mu,u(\mu))(z)\geqslant(\rho(\mu,u(\mu))(z)-z)\cdot\widetilde{n}_2(\mu,u(\mu))(z),$$

thereby showing that (3.30) $\Longrightarrow$ (3.26).

Let us now prove the direct implication, i.e. (3.26) $\Longrightarrow$ (3.30). Let $\tilde{z}\in\overline{\Omega}_1(\mu,u_1(\mu))\cap\overline{\Omega}_2(\mu,u_2(\mu))$. Since in particular, $\tilde{z}\in\overline{\Omega}_1(\mu,u_1(\mu))$, we consider two cases:

- $\tilde{z}$ is a boundary point of the first solid, i.e. $\tilde{z}\in\Upsilon_1^c(\mu,u_1(\mu))$;
- $\tilde{z}$ is an interior point of the first solid, i.e. $\tilde{z}\in\Omega_1(\mu,u_1(\mu))$.

If $\tilde{z}$ is a boundary point, we have

$$\tilde{z}\in\Upsilon_1^c(\mu,u_1(\mu))\Longrightarrow\left(\tilde{z}-\vartheta(\mu,u(\mu))(\tilde{z})\cdot\widetilde{n}_2(\mu,u(\mu))\circ(\psi_1(\mu,u_1(\mu)))^{-1}(\tilde{z})\right)\geqslant 0$$
$$\Longleftrightarrow\tilde{z}=\vartheta(\mu,u(\mu))(\tilde{z})\text{ or }\left(\tilde{z}-\vartheta(\mu,u(\mu))(\tilde{z})\cdot\widetilde{n}_2(\mu,u(\mu))\circ(\psi_1(\mu,u_1(\mu)))^{-1}(\tilde{z})\right)>0$$
$$\Longrightarrow\tilde{z}\in\Upsilon_1^c(\mu,u_1(\mu))\cap\Upsilon_2^c(\mu,u_2(\mu)),$$

where we have used that

$$\left(\tilde{z}-\vartheta(\mu,u(\mu))(\tilde{z})\cdot\widetilde{n}_2(\mu,u(\mu))\circ(\psi_1(\mu,u_1(\mu)))^{-1}(\tilde{z})\right)\geqslant 0\Longrightarrow\tilde{z}\notin\overline{\Omega}_2(\mu,u_2(\mu)), \tag{3.76}$$

which cannot hold true since $\tilde{z} \in \overline{\Omega}_2(\mu, u_2(\mu))$ by assumption. Actually, the implication (3.76) means that

$$\exists z^b \in \Upsilon_2^c(\mu, u_2(\mu)) : \begin{cases} \left(\tilde{z} - z^b \cdot \tilde{n}_2(\mu, u(\mu)) \circ (\psi_1(\mu, u_1(\mu)))^{-1}(\tilde{z})\right) < 0 \\ z^b = \vartheta(\mu, u(\mu))(\tilde{z})), \end{cases}$$

which is in contradiction with the definition of the contact mapping $\vartheta(\mu, u(\mu))$. If $\tilde{z}$ is an interior point of the first solid, we have

$$\tilde{z} \in \Omega_1(\mu, u_1(\mu)) \cap \overline{\Omega}_2(\mu, u_2(\mu)) \Longrightarrow \exists z^b \in \Upsilon_1^c(\mu, u_1(\mu)) : \ z^b \in \Omega_2(\mu, u_2(\mu)). \quad (3.77)$$

The implication (3.77) means that a non-empty intersection of both solids necessarily induces that at least a boundary point of each solid is in the interior of the other. From (3.77), we infer that the constraint (3.26) is violated. Therefore, $\Omega_1(\mu, u_1(\mu)) \cap \overline{\Omega}_2(\mu, u_2(\mu)) = \varnothing$, which concludes the proof. □

CHAPTER 4

MODEL REDUCTION WITH DATA ASSIMILATION

The ideas introduced in this chapter have been devised in collaboration with A. T. Patera during two two-week visits of the author of the manuscript to the Massachusetts Institute of Technology.

4.1 Introduction

State estimation is a task in which the quantity of interest is the 'true' state u^{true} of a physical system over a space or space-time domain of interest. However, numerical prediction based on a given mathematical model may be deficient due to limitations imposed by available knowledge. In other words, the mathematical model can only take anticipated or parametric uncertainty into account. A more accurate prediction requires the incorporation of experimental observations in order to accommodate unanticipated or nonparametric uncertainty.

The Parameterized-Background Data-Weak (PBDW) formulation for variational data assimilation is a data-driven reduced order modeling approach that was initially devised in [40] so as to merge prediction by model with prediction by data. The PBDW approach has been developed in order to estimate the true state u^{true} for several configurations of a physical system. Supposing that the true state u^{true} depends on some unknown parameter ω in an unknown parameter set Θ that represents the unanticipated uncertainty, the goal is to account for the dependency of the true state $u^{\text{true}}(\omega)$ on uncertain parameters by means of the sole knowledge of data. In this chapter, whenever the context is unambiguous, the parameter ω is dropped.

The formulation combines a so-called 'best-knowledge' (`bk`) model represented by a parametrized partial differential equation (PDE) and experimentally observable measurements. The use of data in the PBDW approach is fundamental not only to reconstruct the quantities of interest, but also to correct the possible bias in the

mathematical `bk` model. The PBDW approach provides the following attractive features:

- The PBDW variational formulation simplifies the construction of *a priori* error estimates which can guide the optimal choice of the experimental observations.
- The PBDW formulation uses a `bk` model that accommodates anticipated uncertainty associated with the parameters of the model in a computationally convenient way. This `bk` model is typically built using model-order reduction techniques.

Note that the PBDW formulation does not explicitly include the equations of the `bk` model, but only a finite collection of solutions to the `bk` model. Thus, another important feature of the PBDW approach is its non-intrusiveness. In fact, once the subspace $\mathcal{Z}_N$ has been generated, we no longer need the `bk` model.

The PBDW approach was devised in [40] for steady problems. In this chapter, we propose an extension of the PBDW approach to time-dependent state estimation. Two main contributions to the standard PBDW approach are presented:

- We build appropriate background spaces for the time-dependent setting using the `POD-greedy` algorithm [26].
- We propose a modified offline stage so as to alleviate its computational cost which can be sizeable in a time-dependent setting. The new offline stage allows for a better computational efficiency owing to a smaller online system. Moreover, it achieves substantial cost savings associated with data collection since it diminishes the number of observation sensors needed online. Note that this modified offline stage can also be applied to a steady framework.

This chapter is organized as follows. In Section 4.2, we recall the standard PBDW approach for steady problems as introduced in [40]. In Section 4.3, we extend the PBDW approach to the time-dependent framework. In Section 4.4, we discuss the offline stage. Finally, in Section 4.5, we illustrate our method by numerical results.

4.2 Parametrized-Background Data-Weak (PBDW) approach

In this section, we first introduce the notation that will be used throughout the chapter. Here, we focus on a time-independent setting. We consider a spatial domain (open, bounded, connected subset) $\Omega \subset \mathbb{R}^d$, $d \geqslant 1$, with a Lipschitz boundary. We introduce a Hilbert space $\mathcal{U}$ composed of functions defined over Ω. The space $\mathcal{U}$ is endowed with an inner product $(\cdot, \cdot)$ and we denote by $\|\cdot\|$ the induced norm; $\mathcal{U}$ consists of functions $\{w : \Omega \to \mathbb{R} \mid \|w\| < \infty\}$. To fix the ideas, we assume that $H_0^1(\Omega) \subset \mathcal{U} \subset H^1(\Omega)$, and we denote the dual space of $\mathcal{U}$ by $\mathcal{U}'$. The Riesz operator $R_{\mathcal{U}} : \mathcal{U}' \to \mathcal{U}$ satisfies, for each $\ell \in \mathcal{U}'$,

$$(R_{\mathcal{U}}(\ell), v) = \ell(v), \quad \forall v \in \mathcal{U}. \tag{4.1}$$

For any closed subspace $\mathcal{Q} \subset \mathcal{U}$, the orthogonal complement of $\mathcal{Q}$ is defined as

$$\mathcal{Q}^{\perp} := \{w \in \mathcal{U} \mid (w, v) = 0, \ \forall v \in \mathcal{Q}\}. \tag{4.2}$$

Finally, we introduce a parameter set $\mathcal{P} \subset \mathbb{R}^p, \ p \geqslant 1$, whose elements are generically denoted by $\mu \in \mathcal{P}$.

4.2.1 Best-knowledge (bk) model

The first source of information we shall afford ourselves in the PBDW approach is a so-called 'best-knowledge' (bk) mathematical model in the form of a parameterized PDE posed over the domain Ω (or more generally, over a domain Ω^{bk} such that $\Omega \subset \Omega^{\text{bk}}$). Given a parameter value μ in the parameter set $\mathcal{P}$, we denote the solution to the bk parameterized PDE as $u^{\text{bk}}(\mu) \in \mathcal{U}$. Then, we introduce the manifold associated with the solutions of the bk model

$$\mathcal{M}^{\text{bk}} := \{u^{\text{bk}}(\mu) \mid \mu \in \mathcal{P}\} \subset \mathcal{U}. \tag{4.3}$$

In ideal situations, the true solution u^{true} is well approximated by the bk manifold, i.e., the model error

$$\epsilon^{\text{bk}}_{\text{mod}}(u^{\text{true}}) := \inf_{z \in \mathcal{M}^{\text{bk}}} \|u^{\text{true}} - z\|, \tag{4.4}$$

is very small.

We introduce nested background subspaces $\mathcal{Z}_1 \subset \ldots \subset \mathcal{Z}_N \subset \ldots \subset \mathcal{U}$ that are generated to approximate the bk manifold $\mathcal{M}^{\text{bk}}$ to a certain accuracy. These subspaces can be built using various model-order reduction techniques, for instance, the RBM described in the previous chapters. Note that the indices of the subspaces conventionally indicate their dimensions. To measure how well the true solution is approximated by the background space $\mathcal{Z}_N$, we define the quantity

$$\epsilon^{\text{bk}}_{N}(u^{\text{true}}) := \inf_{z \in \mathcal{Z}_N} \|u^{\text{true}} - z\|. \tag{4.5}$$

When N is large enough, we have

$$\epsilon^{\text{bk}}_{N}(u^{\text{true}}) \approx \epsilon^{\text{bk}}_{\text{mod}}(u^{\text{true}}). \tag{4.6}$$

Moreover, we introduce the reduction error

$$\epsilon^{\text{bk}}_{\text{red},N} := \sup_{u \in \mathcal{M}^{\text{bk}}} \inf_{z \in \mathcal{Z}_N} \|u - z\|, \tag{4.7}$$

which encodes the loss of accuracy caused by solving the bk model in the N-dimensional background space $\mathcal{Z}_N$. Figure 4.1 illustrates both the model and reduction errors, where $\Pi_{\mathcal{Z}_N}(u^{\text{true}})$ and $\Pi_{\mathcal{M}^{\text{bk}}}(u^{\text{true}})$ are closest points to u^{true} in $\mathcal{Z}_N$ and $\mathcal{M}^{\text{bk}}$, respectively. Note that $\Pi_{\mathcal{Z}_N}$ is the $\mathcal{U}$-orthogonal projection onto $\mathcal{Z}_N$. The background space $\mathcal{Z}_N$ can be interpreted as a prior space that approximates the bk manifold which we hope approximates well the true state u^{true}. As previously alluded to, u^{true} rarely lies in $\mathcal{M}^{\text{bk}}$ in realistic engineering study cases.

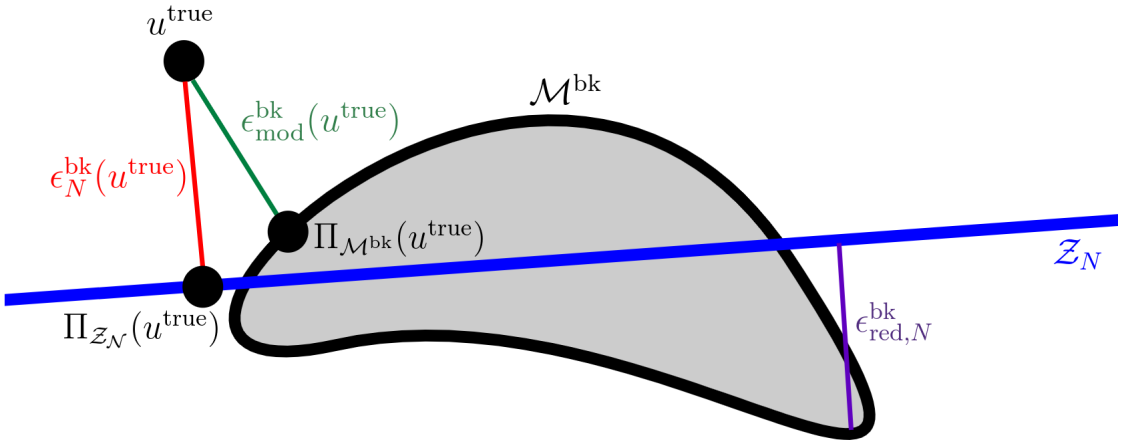

Figure 4.1 – Model and reduction errors.

4.2.2 Unlimited-observations statement

Let us first describe an ideal situation. The unlimited-observations PBDW statement reads: find $(u_N^*, z_N^*, \eta_N^*) \in \mathcal{U} \times \mathcal{Z}_N \times \mathcal{U}$ such that

$$(u_N^*, z_N^*, \eta_N^*) = \underset{\substack{u_N \in \mathcal{U} \\ z_N \in \mathcal{Z}_N \\ \eta_N \in \mathcal{U}}}{\operatorname{arginf}} \|\eta_N\|^2, \tag{4.8}$$

subject to

$$(u_N, v) = (\eta_N, v) + (z_N, v), \quad \forall v \in \mathcal{U}, \tag{4.9a}$$
$$(u_N, \phi) = (u^{\text{true}}, \phi), \quad \forall \phi \in \mathcal{U}. \tag{4.9b}$$

A direct consequence of (4.9a) is that

$$u_N^* = z_N^* + \eta_N^*, \tag{4.10}$$

whereas (4.9b) implies that $u_N^* = u^{\text{true}}$. We will be using the following terminology:

- **State estimate:** The goal of the PBDW statement (4.8)-(4.9) being to estimate the true state u^{true}, the first component of its solution, u_N^*, is called the 'state estimate'. In the present ideal situation of unlimited observations, the state estimate coincides with the true state.

- **Deduced background estimate:** The first contribution z_N^* in (4.10) lies in the background space $\mathcal{Z}_N$ and is deduced from the PBDW statement, which takes the observations into account. Hence, z_N^* is called the 'deduced background estimate'.

- **Update estimate:** The second contribution η_N^* in (4.10) is brought by the inclusion of the observations in the PBDW statement. The observations supplement the `bk` model. Thus, η_N^* is called the 'update estimate'.

The deduced background estimate z_N^* can only represent anticipated uncertainty. Since the `bk` mathematical model of a physical system is often deficient, one cannot realistically assume that the state estimate u_N^* of u^{true} lies completely in the `bk` manifold (or in the background space $\mathcal{Z}_N$). Therefore, the update estimate η_N^* is meant to cure the deficiency of the `bk` model by capturing unanticipated uncertainty. In other words, the key idea of the PBDW statement (4.8) is to search for the smallest correction to the `bk` manifold. The following result is proved in [40].

Proposition 4.1 (Unlimited observations)**.** *The solution of* (4.8)-(4.9) *is given by*

$$u_N^* = u^{\text{true}}, \quad z_N^* = \Pi_{\mathcal{Z}_N}(u^{\text{true}}), \quad \eta_N^* = \Pi_{\mathcal{Z}_N^\perp}(u^{\text{true}}). \tag{4.11}$$

Proof. We have already seen that $u_N^* = u^{\text{true}}$. Next, we deduce from (4.9a) that $u^{\text{true}} = z_N^* + \eta_N^*$. Since (4.8) is a minimization of $\|\eta_N^*\|$, it follows that $z_N^* = \Pi_{\mathcal{Z}_N}(u^{\text{true}})$. Thus, $\eta_N^* = \Pi_{\mathcal{Z}_N^\perp}(u^{\text{true}})$. □

The Euler–Lagrange saddle-point problem associated with the PBDW statement (4.8)-(4.9) reads: find $(z_N^*, \eta_N^*) \in \mathcal{Z}_N \times \mathcal{U}$ such that

$$\begin{cases} (\eta_N^*, q) + (z_N^*, q) = (u^{\text{true}}, q), & \forall q \in \mathcal{U}, \\ (\eta_N^*, p) = 0, & \forall p \in \mathcal{Z}_N, \end{cases} \tag{4.12}$$

and set

$$u_N^* = z_N^* + \eta_N^*. \tag{4.13}$$

As mentioned earlier, the saddle-point problem (4.12) is purely geometric and does not include any explicit reference to the `bk` model. The unique link to the `bk` model is through the background space $\mathcal{Z}_N$. Therefore, the PBDW approach is applicable to a wide class of engineering problems. Moreover, the non-intrusiveness of (4.12) simplifies its implementation.

4.2.3 Observable space

The evaluation of the right-hand side (u^{true}, q) in (4.12) requires the full knowledge of the true state u^{true} which is unrealistic. In practice, one can only afford a limited number of experimental observations of the true state u^{true}. In the present setting, the experimental observations are interpreted as the application of prescribed observation functionals $\ell_m^{\text{obs}} \in \mathcal{U}'$ for all $m \in \{1, \dots, M\}$ such that the m-th experimental observation is given by

$$\ell_m^{\text{obs}}(u^{\text{true}}) \in \mathbb{R}, \quad \forall m \in \{1, \dots, M\}. \tag{4.14}$$

One can consider any observation functional that renders the behaviour of some physical sensor. In the case of sensors measuring the state locally over user-defined subsets $\mathcal{R}_m \subset \Omega$, where $m \in \{1, \dots, M\}$, one possibility is to model each sensor through uniform local integration

$$\ell_m^{\text{obs}}(v) = \frac{1}{|\mathcal{R}_m|} \int_{\mathcal{R}_m} v(x)dx, \quad \forall v \in \mathcal{U}. \tag{4.15}$$

Another plausible option is, as introduced in [40], to consider

$$\ell_m^{\text{obs}}(v) = \frac{1}{\sqrt{2\pi r_m^2}} \int_{\mathcal{R}_m} v(x) \exp\left(\frac{-(x - x_m^c)^2}{2r_m^2}\right) dx, \tag{4.16}$$

where x_m^c is the center of the Gaussian that reflects the location of the sensor, and $r_m \ll |\mathcal{R}_m|^{\frac{1}{d}}$ is the standard deviation of the Gaussian that reflects the filter width of the sensor.

Generally, we introduce the M-dimensional experimentally observable space

$$\mathcal{U}_M := \text{Span}\{q_m\}_{1 \leqslant m \leqslant M} \subset \mathcal{U}, \tag{4.17}$$

where $q_m := R_{\mathcal{U}}(\ell_m^{\text{obs}})$ is the Riesz representation of $\ell_m^{\text{obs}} \in \mathcal{U}'$, for all $m \in \{1, \dots, M\}$. The experimental observations of the true state satisfy

$$\left(u^{\text{true}}, q_m\right) = \ell_m^{\text{obs}}(u^{\text{true}}), \quad \forall m \in \{1, \dots, M\}. \tag{4.18}$$

Hence, for all $q \in \mathcal{U}_M$ such that

$$q = \sum_{m=1}^{M} \alpha_m q_m, \quad \text{with } (\alpha_m)_{1 \leqslant m \leqslant M} \in \mathbb{R}^M, \tag{4.19}$$

the inner product (u^{true}, q) can be deduced from the experimental observations as a linear combination of the M available observations:

$$\left(u^{\text{true}}, q\right) = \sum_{m=1}^{M} \alpha_m \left(u^{\text{true}}, q_m\right) = \sum_{m=1}^{M} \alpha_m \ell_m^{\text{obs}}(u^{\text{true}}). \tag{4.20}$$

4.2.4 Limited-observations statement

Let us now describe the PBDW statement in the case of limited observations. Henceforth, we make the crucial assumption that

$$\mathcal{Z}_N \cap \mathcal{U}_M^{\perp} = \{0\}, \tag{4.21}$$

which is meant to ensure the well-posedness of the PBDW statement with limited observations (cf. Proposition 4.3 below). This assumption can be viewed as a requirement to have enough sensors (note that $\mathcal{Z}_N \cap \mathcal{U}^{\perp} = \{0\}$). The limited-observations PBDW statement reads: find $(u_{N,M}^*, z_{N,M}^*, \eta_{N,M}^*) \in \mathcal{U} \times \mathcal{Z}_N \times \mathcal{U}$ such that

$$(u_{N,M}^*, z_{N,M}^*, \eta_{N,M}^*) = \underset{\substack{u_{N,M} \in \mathcal{U} \\ z_{N,M} \in \mathcal{Z}_N \\ \eta_{N,M} \in \mathcal{U}}}{\text{arginf}} \|\eta_{N,M}\|^2, \tag{4.22}$$

subject to

$$(u_{N,M}, v) = (\eta_{N,M}, v) + (z_{N,M}, v), \quad \forall v \in \mathcal{U}, \tag{4.23a}$$

$$(u_{N,M}, \phi) = (u^{\text{true}}, \phi), \quad \forall \phi \in \mathcal{U}_M. \tag{4.23b}$$

As above, (4.23a) implies that the limited-observations state estimate $u^*_{N,M}$ satisfies

$$u^*_{N,M} = z^*_{N,M} + \eta^*_{N,M}. \tag{4.24}$$

One can show (e.g., by introducing the Lagrangian) that the limited-observations problem (4.22)-(4.23) is equivalent to the limited-observations saddle-point problem: find $(z^*_{N,M}, \eta^*_{N,M}) \in \mathcal{Z}_N \times \mathcal{U}_M$ such that

$$(\eta^*_{N,M}, q) + (z^*_{N,M}, q) = (u^{\text{true}}, q), \quad \forall q \in \mathcal{U}_M, \tag{4.25a}$$

$$(\eta^*_{N,M}, p) = 0, \quad \forall p \in \mathcal{Z}_N, \tag{4.25b}$$

and define $u^*_{N,M}$ according to (4.24). We will see in Proposition 4.3 below that the linear system (4.25) is well posed under the assumption (4.21).

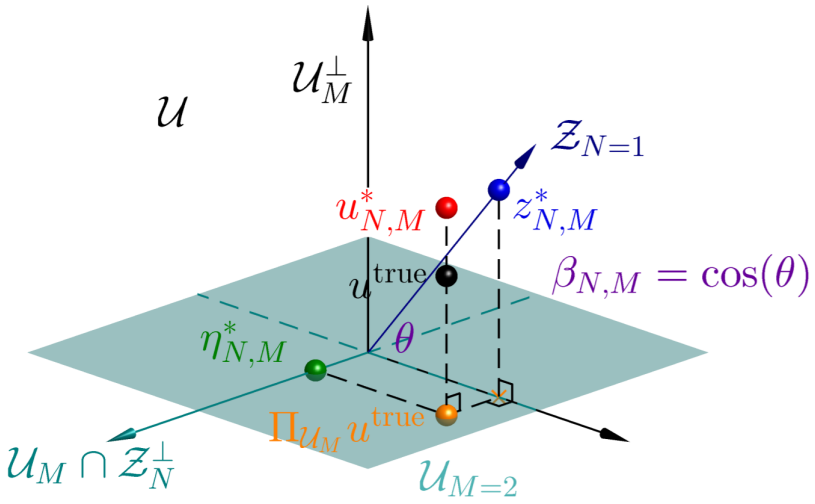

Figure 4.2 – PBDW state estimation (courtesy of A. T Patera).

Proposition 4.2 (Update estimate)**.** *The update estimate is given by*

$$\eta^*_{N,M} = \Pi_{\mathcal{Z}_N^\perp \cap \mathcal{U}_M}(u^{\text{true}}). \tag{4.26}$$

Proof. By definition of the saddle-point problem (4.25), $\eta^*_{N,M} \in \mathcal{U}_M$. From (4.25b), we infer that $\eta^*_{N,M} \in \mathcal{Z}_N^\perp$. Hence, $\eta^*_{N,M} \in \mathcal{Z}_N^\perp \cap \mathcal{U}_M$. Thus, we have

$$u^*_{N,M} = z^*_{N,M} + \eta^*_{N,M} \in \mathcal{Z}_N \oplus (\mathcal{Z}_N^\perp \cap \mathcal{U}_M).$$

Then, (4.25a) yields

$$(\eta^*_{N,M}, q) = (u^{\text{true}}, q), \quad \forall q \in \mathcal{Z}_N^\perp \cap \mathcal{U}_M.$$

We conclude that $\eta^*_{N,M} = \Pi_{\mathcal{Z}_N^\perp \cap \mathcal{U}_M}(u^{\text{true}})$. □

The decomposition of the state estimate $u^*_{N,M}$ is illustrated in Figure 4.2.

Remark 4.1 (Perfect background space)**.** *The choice of the background space $\mathcal{Z}_N$ and of the observable space $\mathcal{U}_M$ may lead to several specific configurations. In particular, the background space $\mathcal{Z}_N$ is said to be perfect if the reduction error (cf. Figure 4.1) vanishes, i.e., $u^{\text{true}} \in \mathcal{Z}_N$. In this case, the pair $(u^{\text{true}}, 0) \in \mathcal{Z}_N \times \mathcal{U}_M$ is the unique solution to (4.25). Hence, the state estimate $u^*_{N,M}$ also belongs to $\mathcal{Z}_N$ and the update estimate satisfies $\eta^*_{N,M} = 0$.*

4.2.5 Algebraic formulation

We now present the algebraic formulation of the limited-observations PBDW statement. We first introduce an $\mathcal{N}$-dimensional approximation space $\mathcal{U}^{\mathcal{N}}$ of the infinite-dimensional space $\mathcal{U}$ as well as discrete approximation spaces $\mathcal{Z}_N \subset \mathcal{U}^{\mathcal{N}}$ and $\mathcal{U}_M \subset \mathcal{U}^{\mathcal{N}}$ of the subspaces $\mathcal{Z}_N$ and $\mathcal{U}_M$, respectively. These spaces are built using finite elements [20]. We assume that the size of the mesh is small enough so that the $\mathcal{N}$-dimensional space discretization delivers High-Fidelity (HF) approximations within the requested level of accuracy. To alleviate the notation, we have dropped the superscript $\mathcal{N}$; hence, the discrete FEM spaces are denoted $\mathcal{Z}_N$ and $\mathcal{U}_M$ instead of $\mathcal{Z}_N^{\mathcal{N}}$ and $\mathcal{U}_M^{\mathcal{N}}$, but we still keep the notation $\mathcal{U}^{\mathcal{N}}$ for the FEM-discretization space. Then, we introduce a basis for the background space $\mathcal{Z}_N := \mathrm{Span}\{\zeta_n\}_{1\leqslant n\leqslant N}$. The update space is spanned by the Riesz representations of the observation functionals in $\mathcal{U}^{\mathcal{N}}$, i.e., $\mathcal{U}_M := \mathrm{Span}\{q_m\}_{1\leqslant m\leqslant M}$, where $q_m \in \mathcal{U}^{\mathcal{N}}$, for all $m \in \{1,\dots,M\}$. The high-fidelity (HF) discretization of the saddle-point problem (4.25) is: Find $(z^*_{N,M}, \eta^*_{N,M}) \in \mathcal{Z}_N \times \mathcal{U}_M$ such that

$$(\eta^*_{N,M}, q) + (z^*_{N,M}, q) = (u^{\mathrm{true}}, q), \quad \forall q \in \mathcal{U}_M, \tag{4.27a}$$

$$(\eta^*_{N,M}, p) = 0, \quad \forall p \in \mathcal{Z}_N. \tag{4.27b}$$

The solution of (4.27) is then searched under the form

$$z^*_{N,M} = \sum_{n=1}^{N} z_n \zeta_n, \quad \text{and} \quad \eta^*_{N,M} = \sum_{m=1}^{M} \eta_m q_m, \tag{4.28}$$

and we introduce the component vectors

$$\boldsymbol{z}^*_{N,M} := (z_n)_{1\leqslant n\leqslant N} \in \mathbb{R}^N \quad \text{and} \quad \boldsymbol{\eta}^*_{N,M} := (\eta_m)_{1\leqslant m\leqslant M} \in \mathbb{R}^M. \tag{4.29}$$

We also introduce the basis matrices $\mathbf{Z}_N \in \mathbb{R}^{\mathcal{N}\times N}$ and $\mathbf{U}_M \in \mathbb{R}^{\mathcal{N}\times M}$ whose column vectors are the components of the functions $\{\zeta_n\}_{1\leqslant n\leqslant N}$ and $\{q_m\}_{1\leqslant m\leqslant M}$ respectively in the basis of $\mathcal{U}^{\mathcal{N}}$. In algebraic form, the FEM-discretized saddle-point problem (4.27) reads: find $(\boldsymbol{z}^*_{N,M}, \boldsymbol{\eta}^*_{N,M}) \in \mathbb{R}^N \times \mathbb{R}^M$ such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}^*_{N,M} \\ \boldsymbol{z}^*_{N,M} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\ell}^{\mathrm{obs}}_M \\ \mathbf{0} \end{pmatrix}, \tag{4.30}$$

with the matrices

$$\mathbf{A} = \Big((q_{m'}, q_m)\Big)_{1\leqslant m,m'\leqslant M} \in \mathbb{R}^{M\times M}, \quad \mathbf{B} = \Big((\zeta_n, q_m)\Big)_{1\leqslant m\leqslant M, 1\leqslant n\leqslant N} \in \mathbb{R}^{M\times N}, \tag{4.31}$$

and the vector of observations

$$\boldsymbol{\ell}^{\mathrm{obs}}_M = \big(\ell^{\mathrm{obs}}_m(u^{\mathrm{true}})\big)_{1\leqslant m\leqslant M} \in \mathbb{R}^M. \tag{4.32}$$

Proposition 4.3 (Well-posedness)**.** *The PBDW statement (4.30) has a unique solution if and only if $\mathcal{Z}_N \cap \mathcal{U}_M^{\perp} = \{0\}$. Equivalently, under this assumption, the stability constant*

$$\beta_{N,M} := \inf_{w\in\mathcal{Z}_N} \sup_{v\in\mathcal{U}_M} \frac{(w,v)}{\|w\|\,\|v\|} \in (0,1]. \tag{4.33}$$

Proof. The system (4.30) is a saddle-point problem with a symmetric positive definite matrix $\mathbf{A}$. Therefore, it has a unique solution if and only if the matrix $\mathbf{B}$ is injective, i.e., if and only if $\ker(\mathbf{B}) = \{0\}$. Using the definition of $\mathbf{B}$ in (4.31), we have

$$\begin{aligned} \ker(\mathbf{B}) \neq \{0\} &\iff \exists z \in \mathcal{Z}_N \backslash \{0\} : \ \forall q \in \mathcal{U}_M : (z, q) = 0, \\ &\iff \mathcal{Z}_N \cap \mathcal{U}_M^\perp \neq \{0\}. \end{aligned} \tag{4.34}$$

Thus, (4.30) is well-posed if and only if $\mathcal{Z}_N \cap \mathcal{U}_M^\perp = \{0\}$ and this statement is equivalent to $\beta_{N,M} > 0$. Finally, we readily verify that $\beta_{N,M} \leqslant 1$ owing to the Cauchy–Schwarz inequality. □

Remark 4.2 (Stability constant)**.** *In terms of geometry, the stability constant $\beta_{N,M}$ is equal to the cosine of the angle between the linear subspaces $\mathcal{Z}_N$ and $\mathcal{U}_M^\perp$ (cf. Figure 4.2). Furthermore, it is readily verified that*

$$\beta_{N,M} = 0 \iff \mathcal{Z}_N \cap \mathcal{U}_M^\perp \neq \{0\}, \tag{4.35a}$$

$$\beta_{N,M} = 1 \iff \mathcal{Z}_N \subset \mathcal{U}_M. \tag{4.35b}$$

The case (4.35b) can hardly occur in practice with a reasonable (not too high) number of observation sensors. Loosely speaking, a sensor is localized in space. Thus, it concerns only a limited number of degrees of freedom.

Remark 4.3 (Insufficient observations)**.** *If $M < N$, then (4.30) is necessarily ill-posed. Indeed, we have*

$$M < N \iff \dim(\mathcal{U}_M) < \dim(\mathcal{Z}_N).$$

Moreover, we have

$$\begin{aligned} \dim(\mathcal{Z}_N) - \operatorname{codim}(\mathcal{U}_M^\perp) \leqslant \dim(\mathcal{Z}_N \cap \mathcal{U}_M^\perp) &\iff \dim(\mathcal{Z}_N) - \dim(\mathcal{U}_M) \leqslant \dim(\mathcal{Z}_N \cap \mathcal{U}_M^\perp) \\ &\implies 1 \leqslant \dim(\mathcal{Z}_N \cap \mathcal{U}_M^\perp) \\ &\iff \mathcal{Z}_N \cap \mathcal{U}_M^\perp \neq \{0\} \\ &\iff \beta_{N,M} = 0, \end{aligned}$$

where the last equivalence follows from Proposition 4.3.

In practice, the matrices $\mathbf{A}$ and $\mathbf{B}$ are computed using the algebraic formulas

$$\mathbf{A} = \mathbf{U}_M \mathbf{M} \mathbf{U}_M, \quad \text{and} \quad \mathbf{B} = \mathbf{Z}_N \mathbf{M} \mathbf{U}_M, \tag{4.36}$$

where $\mathbf{M}$ is the Gram matrix of the inner product in $\mathcal{U}$. Thus, solving (4.30) allows for a straightforward reconstruction of the components of the state estimate in the basis of $\mathcal{U}^\mathcal{N}$ as follows:

$$\boldsymbol{u}_{N,M}^* = \mathbf{Z}_N \boldsymbol{z}_{N,M}^* + \mathbf{U}_M \boldsymbol{\eta}_{N,M}^*. \tag{4.37}$$

Offline/online procedure: Since several realizations $u^{\text{true}}(\omega)$ of the true state are considered, an offline/online procedure can be employed. During the offline stage, one precomputes the RB functions $(\zeta_n)_{1\leqslant n\leqslant N}$ and the Riesz respresenters $(q_m)_{1\leqslant m\leqslant M}$ leading to the matrices $\mathbf{A}\in\mathbb{R}^{M\times M}$ and $\mathbf{B}\in\mathbb{R}^{N\times M}$ once and for all. Then, during the online stage, for each new set of observations corresponding to a new realization of the true state $u^{\text{true}}(\omega)$, all that remains to be performed is to form the vector of observations $\boldsymbol{\ell}_M^{\text{obs}}(\omega)$ and to retrieve the deduced background estimate $\boldsymbol{z}_{N,M}^*(\omega)$ and the update estimate $\boldsymbol{\eta}_{N,M}^*(\omega)$ by solving the $(N+M)$-dimensional linear problem (4.30). The PBDW state estimate $\boldsymbol{u}_{N,M}^*(\omega)$ is then computed using (4.37).

4.2.6 *A priori* error analysis

Here, we state an important result proved in [40] and related to the *a priori* error analysis of the PBDW statement.

Proposition 4.4 (Error estimate)**.** *The PBDW update estimate, deduced background estimate and state estimate satisfy the following error bounds:*

$$\|\eta_{N,M}^*-\eta_N^*\|\leqslant\inf_{q\in\mathcal{Z}_N^\perp\cap\mathcal{U}_M}\inf_{z\in\mathcal{Z}_N}\|u^{\text{true}}-z-q\|, \tag{4.38}$$

$$\|z_{N,M}^*-z_N^*\|\leqslant\frac{1}{\beta_{N,M}}\inf_{q\in\mathcal{Z}_N^\perp\cap\mathcal{U}_M}\inf_{z\in\mathcal{Z}_N}\|u^{\text{true}}-z-q\|, \tag{4.39}$$

$$\|u_{N,M}^*-u^{\text{true}}\|\leqslant\left(1+\frac{1}{\beta_{N,M}}\right)\inf_{q\in\mathcal{Z}_N^\perp\cap\mathcal{U}_M}\inf_{z\in\mathcal{Z}_N}\|u^{\text{true}}-z-q\|. \tag{4.40}$$

Proposition 4.4 shows that, the larger the stability constant $\beta_{N,M}$, the smaller the error on the deduced background estimate and on the state estimate. A straightforward upper bound with $q=0$ on the right-hand side of (4.38) yields

$$\|\eta_{N,M}^*-\eta_N^*\|\leqslant\inf_{z\in\mathcal{Z}_N}\|u^{\text{true}}-z\|. \tag{4.41}$$

Hence, the error on the update estimate depends on the quality of approximation of the true state u^{true} in $\mathcal{Z}_N$, i.e., $\inf_{z\in\mathcal{Z}_N}\|u^{\text{true}}-z\|$. The same comment is valid for the error on the deduced background estimate and the state estimate. Moreover, since $\mathcal{Z}_N$ and $\mathcal{Z}_N^\perp\cap\mathcal{U}_M$ form a $\mathcal{U}$-orthogonal direct sum, we have

$$\begin{aligned}\inf_{q\in\mathcal{Z}_N^\perp\cap\mathcal{U}_M}\inf_{z\in\mathcal{Z}_N}\|u^{\text{true}}-z-q\|&=\inf_{q\in\mathcal{Z}_N^\perp\cap\mathcal{U}_M}\|u^{\text{true}}-\Pi_{\mathcal{Z}_N}(u^{\text{true}})-q\|\\&=\inf_{q\in\mathcal{Z}_N^\perp\cap\mathcal{U}_M}\|\Pi_{\mathcal{Z}_N^\perp}(u^{\text{true}})-q\|,\end{aligned} \tag{4.42}$$

which shows that the three errors from Proposition 4.4 also depend on how well the subspace $\mathcal{Z}_N^\perp\cap\mathcal{U}_M$ cures the lack of information in the background space $\mathcal{Z}_N$.

Remark 4.4 (Choice of spaces)**.** *The subspaces $\mathcal{Z}_N$ and $\mathcal{U}_M$ must be chosen carefully. In fact, we want a small angle between the spaces $\mathcal{Z}_N$ and $\mathcal{U}_M$ in order to increase the stability constant, but we need some overlap between the spaces $\mathcal{Z}_N^\perp$ and $\mathcal{U}_M$ to improve the approximation capacity of $\mathcal{Z}_N^\perp\cap\mathcal{U}_M$.*

4.3 Time-dependent PBDW

Consider a finite time interval $I = [0, T]$, with $T > 0$. To discretize in time, we consider an integer $K \geqslant 1$, we define $0 = t^0 < \cdots < t^K = T$ as $(K+1)$ distinct time nodes over I, and we set $\mathbb{K}^{\mathrm{tr}} = \{1, \ldots, K\}$, $\overline{\mathbb{K}}^{\mathrm{tr}} = \{0\} \cup \mathbb{K}^{\mathrm{tr}}$ and $I^{\mathrm{tr}} = \{t^k\}_{k \in \overline{\mathbb{K}}^{\mathrm{tr}}}$.

Remark 4.5 (Initial condition)**.** *In the present setting, we choose not to solve the PBDW statement for the initial time node $k = 0$. It is straightforward to consider a setting where the initial time node is also included.*

4.3.1 Unlimited-observations statement

In this ideal setting, we assume that $u^{\mathrm{true}} \in \mathcal{C}^0(I; \mathcal{U})$. The time-dependent unlimited-observations PBDW statement reads: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(u_N^{k,*}, z_N^{k,*}, \eta_N^{k,*}) \in \mathcal{U} \times \mathcal{Z}_N \times \mathcal{U}$ such that

$$(u_N^{k,*}, z_N^{k,*}, \eta_N^{k,*}) = \underset{\substack{u_N \in \mathcal{U} \\ z_N \in \mathcal{Z}_N \\ \eta_N \in \mathcal{U}}}{\operatorname{arginf}} \|\eta_N\|^2, \tag{4.43}$$

subject to

$$(u_N, v) = (\eta_N, v) + (z_N, v), \qquad \forall v \in \mathcal{U}, \tag{4.44a}$$
$$(u_N, \phi) = (u^{k,\mathrm{true}}, \phi), \qquad \forall \phi \in \mathcal{U}. \tag{4.44b}$$

where $u^{k,\mathrm{true}} := u^{\mathrm{true}}(t^k, \cdot)$. For each $k \in \mathbb{K}^{\mathrm{tr}}$, the solution of (4.43)-(4.44) is given by

$$u_N^{k,*} = u^{k,\mathrm{true}}, \quad z_N^{k,*} = \Pi_{\mathcal{Z}_N}(u^{k,\mathrm{true}}), \quad \eta_N^{k,*} = \Pi_{\mathcal{Z}_N^\perp}(u^{k,\mathrm{true}}). \tag{4.45}$$

The Euler–Lagrange saddle-point problem associated with the time-dependent PBDW statement (4.43)-(4.44) reads: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(z_N^{k,*}, \eta_N^{k,*}) \in \mathcal{Z}_N \times \mathcal{U}$ such that

$$(\eta_N^{k,*}, q) + (z_N^{k,*}, q) = (u^{k,\mathrm{true}}, q), \qquad \forall q \in \mathcal{U}, \tag{4.46a}$$
$$(\eta_N^{k,*}, p) = 0, \qquad \forall p \in \mathcal{Z}_N. \tag{4.46b}$$

The unlimited-observations state estimate is then

$$u_N^{k,*} = z_N^{k,*} + \eta_N^{k,*}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.47}$$

4.3.2 Limited-observations statement

We now weaken the regularity assumption on the true state and only assume that $u^{\mathrm{true}} \in L^1(I; \mathcal{U})$. We introduce the time-integration intervals

$$\mathcal{I}^k = [t^k - \delta t_k, t^k + \delta t_k], \quad \forall k \in \mathbb{K}^{\mathrm{tr}}, \tag{4.48}$$

where $\delta t^k > 0$ is a parameter related to the precision of the sensor (ideally, $\delta t^k < \min(t^{k+1} - t^k, t^k - t^{k-1})$ with obvious adaptation if k=K). Then, for any function $v \in L^1(I;\mathcal{U})$, we define the time-averaged snapshots

$$v^k(x) := \frac{1}{|\mathcal{I}^k|} \int_{\mathcal{I}^k} v(t,x)\ dt \in \mathcal{U}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.49}$$

As in the steady case, we consider observation functionals that render the behaviour of given sensors. We use the same observation functionals as in the time-independent setting, but we let them act on the time-averaged snapshots of the true solution, i.e., we consider

$$\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) := \ell_m^{\mathrm{obs}}(u^{k,\mathrm{true}}), \quad \forall m \in \{1,\dots,M\},\ \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.50}$$

For instance, if the sensors act through local uniform time integration (see (4.15)), we have

$$\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) = \frac{1}{|\mathcal{R}_m|} \int_{\mathcal{R}_m} u^{k,\mathrm{true}}(x)\ dx = \frac{1}{|\mathcal{R}_m|} \frac{1}{|\mathcal{I}_k|} \int_{\mathcal{R}_m} \int_{\mathcal{I}_k} u^{\mathrm{true}}(t,x)\ dxdt, \tag{4.51}$$

whereas if the sensors act through integration against a Gaussian (see (4.16)), we have

$$\begin{aligned} \ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) &= \frac{1}{\sqrt{2\pi r_m^2}} \int_{\mathcal{R}_m} u^{k,\mathrm{true}}(x) \exp\left(\frac{-(x-x_m^c)^2}{2r_m^2}\right)\ dxdt, \\ &= \frac{1}{|\mathcal{I}_k|} \frac{1}{\sqrt{2\pi r_m^2}} \int_{\mathcal{I}_k} \int_{\mathcal{R}_m} u^{\mathrm{true}}(x) \exp\left(\frac{-(x-x_m^c)^2}{2r_m^2}\right)\ dxdt. \end{aligned} \tag{4.52}$$

Generally, we introduce the time-independent observable space $\mathcal{U}_M \subset \mathcal{U}$ such that

$$\mathcal{U}_M = \mathrm{Span}\{q_1,\dots,q_M\}. \tag{4.53}$$

The observation functionals in $\mathcal{U}'$ are then defined as

$$\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) = (u^{k,\mathrm{true}}, q_m), \quad \forall m \in \{1,\dots,M\}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.54}$$

Note that, for fixed sensor locations, the computational effort to compute the Riesz representations of the observation functionals is time-independent and is incurred only once so that the experimental observations of the true state satisfy

$$\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) = \left(u^{k,\mathrm{true}}, q_m\right) = \frac{1}{|\mathcal{I}_k|} \int_{\mathcal{I}^k} \ell_m^{\mathrm{obs}}(u^{\mathrm{true}}(t,\cdot))dt,\ \forall m \in \{1,\dots,M\},\ \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.55}$$

Hence, for all $q \in \mathcal{U}_M$ such that,

$$q = \sum_{m=1}^{M} \alpha_m q_m, \tag{4.56}$$

the inner product $\left(u^{k,\mathrm{true}}, q\right)$ is deduced from the experimental observations as follows:

$$\left(u^{k,\mathrm{true}}, q\right) = \frac{1}{|\mathcal{I}_k|}\int_{\mathcal{I}^k}\sum_{m=1}^{M}\alpha_m\left(u^{\mathrm{true}}(t,\cdot), q_m\right)dt = \frac{1}{|\mathcal{I}_k|}\sum_{m=1}^{M}\alpha_m\int_{\mathcal{I}^k}\ell_m^{\mathrm{obs}}(u^{\mathrm{true}}(t,\cdot))dt. \tag{4.57}$$

We are now ready to write the limited-observations PBDW statement: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(u_{N,M}^{k,*}, z_{N,M}^{k,*}, \eta_{N,M}^{k,*}) \in \mathcal{U} \times \mathcal{Z}_N \times \mathcal{U}$ such that

$$(u_{N,M}^{k,*}, z_{N,M}^{k,*}, \eta_{N,M}^{k,*}) = \underset{\substack{u_{N,M}\in\mathcal{U}\\ z_{N,M}\in\mathcal{Z}_N\\ \eta_{N,M}\in\mathcal{U}}}{\operatorname{arginf}} \|\eta_{N,M}\|^2, \tag{4.58}$$

subject to

$$(u_{N,M}, v) = (\eta_{N,M}, v) + (z_{N,M}, v), \qquad \forall v \in \mathcal{U}, \tag{4.59a}$$
$$(u_{N,M}, \phi) = (u^{k,\mathrm{true}}, \phi), \qquad \forall \phi \in \mathcal{U}_M. \tag{4.59b}$$

The limited-observations saddle-point problem associated with (4.58) reads: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(z_{N,M}^{k,*}, \eta_{N,M}^{k,*}) \in \mathcal{Z}_N \times \mathcal{U}_M$ such that

$$(\eta_{N,M}^{k,*}, q) + (z_{N,M}^{k,*}, q) = (u^{k,\mathrm{true}}, q), \quad \forall q \in \mathcal{U}_M, \tag{4.60a}$$
$$(\eta_{N,M}^{k,*}, p) = 0, \qquad \forall p \in \mathcal{Z}_N, \tag{4.60b}$$

and the limited-observations state estimate is

$$u_{N,M}^{k,*} = z_{N,M}^{k,*} + \eta_{N,M}^{k,*}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.61}$$

Remark 4.6 (Pointwise measurements)**.** *For simplicity of implementation, assuming that $u^{\mathrm{true}} \in \mathcal{C}^0(I;\mathcal{U})$, one may consider pointwise measurements in time, i.e.,*

$$\left(u^{k,\mathrm{true}}, q_m\right) = \ell_m^{\mathrm{obs}}(u^{\mathrm{true}}(t^k,\cdot)), \quad \forall m \in \{1,\dots,M\}, \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{4.62}$$

The assumption (4.62) is typically reasonable for a sensor of small precision δt^k.

In algebraic form, the limited-observations PBDW statement reads: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(\boldsymbol{z}^{k,*}, \boldsymbol{\eta}^{k,*}) \in \mathbb{R}^N \times \mathbb{R}^M$ such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix}\begin{pmatrix} \boldsymbol{\eta}^{k,*} \\ \boldsymbol{z}^{k,*} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\ell}^{k,\mathrm{obs}} \\ \mathbf{0} \end{pmatrix}, \tag{4.63}$$

with the matrices

$$\mathbf{A} = \Big((q_{m'}, q_m)\Big)_{1\leqslant m,m'\leqslant M} \in \mathbb{R}^{M\times M}, \quad \mathbf{B} = \Big((\zeta_n, q_m)\Big)_{1\leqslant m\leqslant M, 1\leqslant n\leqslant N} \in \mathbb{R}^{M\times N}, \tag{4.64}$$

and the vector of observations

$$\boldsymbol{\ell}^{k,\mathrm{obs}} = \left(\ell_m^{\mathrm{obs}}(u^{k,\mathrm{true}})\right)_{1\leqslant m\leqslant M} \in \mathbb{R}^M. \tag{4.65}$$

Similarly to the steady PBDW linear system (4.30), we solve (4.63) through an offline/online decomposed computational procedure whenever several realizations $u^{\mathrm{true}}(\omega)$ of the true state are to be considered.

Remark 4.7 (PBDW matrices)**.** *Notice that the PBDW matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *are time-independent; only the right-hand side in (4.63) depends on k.*

4.4 Offline stage

In this section, we discuss the offline stage. Our main goal is to address the construction of the background space $\mathcal{Z}_N$.

4.4.1 Background space construction via `POD-greedy`

Suppose that we have computed a set of high-fidelity (HF) trajectories

$$\mathcal{S} = (\mathcal{S}_k)_{k\in\mathbb{K}^{\mathrm{tr}}} = \left((u^k(\mu))_{\mu\in\mathcal{P}^{\mathrm{tr}}}\right)_{k\in\mathbb{K}^{\mathrm{tr}}}, \tag{4.66}$$

where $u^k(\mu) := u(\mu)(t^k,\cdot)$, for all $k \in \mathbb{K}^{\mathrm{tr}}$. If we were to consider the PBDW statement (4.58)-(4.59) for each $k \in \mathbb{K}^{\mathrm{tr}}$ as an independent steady PBDW statement, we would be working with the time-dependent background spaces

$$\mathcal{Z}^k_{N^k} = \mathtt{POD}(\mathcal{S}_k, \epsilon_{\mathrm{POD}}), \quad \forall k \in \mathbb{K}^{\mathrm{tr}}, \tag{4.67}$$

where the procedure `POD` is defined in Chapter 1. However, this strategy is not convenient since the sizes N^k of the background spaces $\mathcal{Z}^k_{N^k}$ would depend on k. Since the observable space $\mathcal{U}_M$ is fixed, the same non-homogeneity between time nodes would also arise in the stability constant $\beta_{N^k,M}$. Thus, we propose to apply a `POD-greedy` algorithm [26] in order to build a time-independent background space $\mathcal{Z}_N$ that will be used for all $k \in \mathbb{K}^{\mathrm{tr}}$. The advantage is that the PBDW matrices $\mathbf{A}$ and $\mathbf{B}$ and the stability constant $\beta_{N,M}$ remain unchanged regardless of the discrete time node. The offline stage using the `POD-greedy` algorithm is summarized in Algorithm 4.1.

Algorithm 4.1 Offline stage via `POD-greedy`

Input : $\mathcal{S}$ and ϵ_{POD}.
$\mathcal{Q}^{\mathrm{init}}$: an initial set of Riesz representations for the observations.

1: Compute $\mathcal{Z}_N := \mathtt{POD\text{-}greedy}(\mathcal{S}, \epsilon_{\mathrm{POD}})$.
2: Set $\mathcal{U}_M := \mathrm{Span}\{\mathcal{Q}^{\mathrm{init}}\}$.
3: Compute the matrices $\mathbf{A}$ and $\mathbf{B}$ using $\mathcal{Z}_N$ and $\mathcal{U}_M$.

Output : $\mathcal{Z}_N$, $\mathcal{U}_M$, $\mathbf{A}$ and $\mathbf{B}$.

4.4.2 Background space construction via state estimation

We now devise a new algorithm in the context of time-dependent PBDW to perform the offline stage. Here, the construction of the background space $\mathcal{Z}_N$, the choice of the observation space $\mathcal{U}_M$ and the PBDW matrices are modified. The key idea of the new procedure is to precompute the PBDW state estimates of the parameters in the training set $\mathcal{P}^{\mathrm{tr}}$ during the offline stage. The background space is then deduced from these PBDW state estimates. The benefit is that the newly created

background space incorporates data-based knowledge. The modified offline stage of the PBDW for time-dependent problems is described in Algorithm 4.2. Within the modified offline stage, we use the so-called 'Greedy stability maximization' (`S-Greedy`) algorithm (considered in [40]) in line 6 in order to identify the least stable mode and then take the best measurement. The algorithm uses an input space $\mathcal{Z}_N$ that results from a `POD-greedy` procedure so that $\mathcal{Z}_1$ contains the dominant mode, and so forth. The `S-greedy` algorithm selects the observations progressively. Thus, the enrichement of the observable space $\mathcal{U}_M$ stops once the minimum stability β_{MIN} is reached. The procedure `S-greedy` is described in Algorithm 4.3 below. Altogether,

Algorithm 4.2 Modified offline stage of the time-dependent PBDW

Input : $\mathcal{P}^{\text{tr}}$, $\mathbb{K}^{\text{tr}}$, $\mathcal{S}$, ϵ_{POD}, $\epsilon_{\text{POD}}^{\text{init}}$ and β_{MIN}.
$\mathcal{Q}^{\text{init}}$: an initial set of Riesz representations of observations.

1: Compute $\mathcal{Z}_{N^{\text{init}}}^{\text{init}} :=$ `POD-greedy`$(\mathcal{S}, \epsilon_{\text{POD}}^{\text{init}})$.
2: Set $\mathcal{U}_{M^{\text{init}}}^{\text{init}} := \text{Span}\{\mathcal{Q}^{\text{init}}\}$.
3: Compute the matrices $\mathbf{A}^{\text{init}}$ and $\mathbf{B}^{\text{init}}$ using $\mathcal{Z}_{N^{\text{init}}}^{\text{init}}$ and $\mathcal{U}_{M^{\text{init}}}^{\text{init}}$.
4: Estimate the state $u^{k,*}(\mu)$ for all $(\mu, k) \in \mathcal{P}^{\text{tr}} \times \mathbb{K}^{\text{tr}}$.
5: Compute $\mathcal{Z}_N :=$ `POD-greedy`$(\{u^{k,*}(\mu)\}_{\mu \in \mathcal{P}^{\text{tr}}, k \in \mathbb{K}^{\text{tr}}}, \epsilon_{\text{POD}})$.
6: Compute $\mathcal{U}_M :=$ `S-Greedy`$(\mathcal{P}^{\text{tr}}, \mathbb{K}^{\text{tr}}, N, \{\mathcal{Z}_n\}_{n=1}^{N}, \beta_{\text{MIN}}, \mathcal{Q}^{\text{init}})$.
7: Compute the matrices $\mathbf{A}$ and $\mathbf{B}$ using $\mathcal{Z}_N$ and $\mathcal{U}_M$.

Output : $\mathcal{Z}_N$, $\mathcal{U}_M$, $\mathbf{A}$ and $\mathbf{B}$.

the modified offline stage in the proposed algorithm offers four major advantages:

- **Improved background space:** Since the background space $\mathcal{Z}_N$ is built using both the `bk` model and the observations, it is expected to have better approximation capacities of the true state.
- **Reduced number of online observations:** In line 6 of Algorithm 4.2, we select each new data point so as to maximize the stability constant $\beta_{N,M}$. Thus, the observations that will be used during the online stage are mainly needed only for stability and not for accuracy.
- **Reduced dimension of the online PBDW statement:** Since the number of observations is significantly reduced, the modified PBDW matrices are of smaller size compared to the matrices of the standard PBDW. Thus, using the modified offline algorithm, the online PBDW formulation is solved faster.
- **Reduced storage cost:** Owing to the reduced number of online measurements, the dimensions of the observable space $\mathcal{U}_M$ and of the matrices $\mathbf{A}$ and $\mathbf{B}$ are smaller, whence the storage gain.

Regarding computational efficiency, the modified procedure consists of more steps than in the standard PBDW. However, all the additional steps of the algorithm are

Algorithm 4.3 `S-Greedy`: Stability-maximization algorithm

Input : N, $\mathcal{Z}_N$ and $\beta_{\text{MIN}} \in (0, 1]$.
$\mathcal{Q}^{\text{init}}$: an initial set of Riesz representations of the observations.

1: Choose a random $q_1 \in \mathcal{Q}^{\text{init}}$.
2: Set $\mathcal{U}_1 := \text{Span}\{q_1\}$.
3: Compute the stability constant $\beta_{1,1}$ using $\mathcal{Z}_1$ and $\mathcal{U}_1$.
4: Set $m := 2$.
5: **while** $\beta_{N,m-1} < \beta_{\text{MIN}}$ or $m < M$ **do**
6: Compute the least stable mode and the associated supremizer

$$w_{\text{inf}} \in \underset{w \in \mathcal{Z}_N}{\text{arginf}} \sup_{v \in \mathcal{U}_{m-1}} \frac{(w, v)}{\|w\| \, \|v\|}, \quad \text{and} \quad v_{\text{sup}} = \Pi_{\mathcal{U}_{m-1}}(w_{\text{inf}}).$$

7: Identify the least well-approximated vector $q_m = \underset{q \in \mathcal{Q}^{\text{init}}}{\text{argsup}} \, |(q, w_{\text{inf}} - v_{\text{sup}})|$.
8: Set $\mathcal{U}_m := \text{Span}\{\mathcal{U}_{m-1}, q_m\}$.
9: Compute the stability constant $\beta_{N,m}$.
10: $m = m + 1$.
11: **end while**
12: $M := m$.

Output : $\mathcal{U}_M$.

performed offline. As for all reduced-order modeling techniques, the goal of the algorithm is to further improve the online efficiency. Hence, the computational savings brought by the new PBDW formulation come, in our opinion, at a reasonable offline price. Indeed, the resolution of the (online) standard PBDW statement for each parameter $\mu \in \mathcal{P}^{\text{tr}}$ has a reduced computational cost. The only relevant additional computational cost incurred offline is related to the second `POD-greedy` (cf. line 5 of Algorithm 4.2). We believe this computational effort remains acceptable.

Remark 4.8 (Least stable mode)**.** *Line 6 of Algorithm 4.3 may return several infimums. Among these infimums, we select a function whose norm in* $\mathcal{U}$ *is maximal.*

Remark 4.9 (Steady setting)**.** *In a time-dependent framework, the computational savings induced by the modified offline stage are substantial; in particular because of the influence of the time steps. However, Algorithm 4.3 can be applied in the steady setting as well.*

4.5 Numerical results

In this section, we illustrate the above developments on test cases related to the heat equation. The goal is to illustrate the computational performance of our algorithms. In all our test cases, we consider a two-dimensional setting based on the

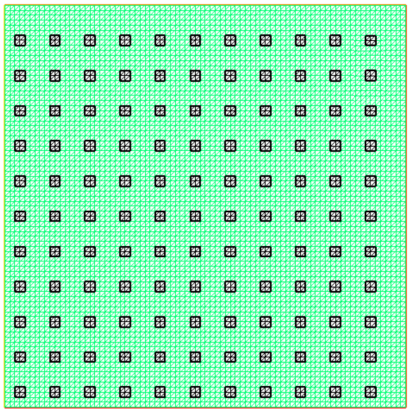

Figure 4.3 – Computational domain and mesh with $\mathcal{N} = 6561$. The little black squares are the observation subsets $\{\mathcal{R}_m\}_{m=1}^{121}$.

plate illustrated in Figure 4.3 with $\Omega = (-2, 2)^2 \subset \mathbb{R}^2$. We use a finite element subspace $\mathcal{U}^{\mathcal{N}} \subset \mathcal{U} = H^1(\Omega)$ consisting of continuous, piecewise affine functions in order to generate high-fidelity (HF) trajectories. The FEM subspace $\mathcal{U}^{\mathcal{N}}$ is based on a mesh that contains $\mathcal{N} = 6561$ nodes. The experimental data is generated synthetically and the observation subsets $\{\mathcal{R}_m\}_{1 \leqslant m \leqslant M}$ are uniformly selected over the plate as illustrated in Figure 4.3. Regarding the implementation, the HF computa-

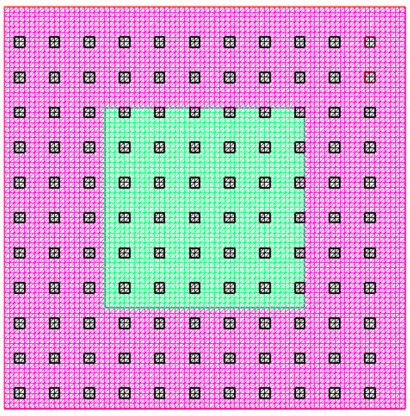

Figure 4.4 – Computational domain and mesh for a bi-material plate with $\mathcal{N} = 6561$. The little black squares are the observation subsets $\{\mathcal{R}_m\}_{m=1}^{121}$.

tions use the software `FreeFem++` [28], whereas the reduced-order modeling and the PBDW-related algorithms have been developed in `Python`.

4.5.1 Physical model problem

We apply the above methodology to the following parabolic PDE: For many values of the parameter $\mu \in \mathcal{P}$, find $u(\mu) : I \times \Omega \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial u(\mu)}{\partial t} - \nabla \cdot (D(\mu)\nabla u(\mu)) = 0, & \text{in } I \times \Omega, \\ u(\mu)(t = 0, \cdot) = u_0, & \text{in } \Omega, \\ \text{Boundary conditions}, & \text{on } I \times \partial\Omega, \end{cases} \tag{4.68}$$

where $u_0 = 293.15\text{K}$ (20^oC). We will supplement (4.68) with two types of boundary conditions:

1. **Linear heat equation:** We apply a homogeneous Neumann boundary condition on $\partial\Omega_0$ and a non-homogeneous Neumann boundary condition on $\partial\Omega_n$, i.e.,

$$\begin{cases} -D(\mu)\dfrac{\partial u(\mu)}{\partial n} = 0, & \text{on } I \times \partial\Omega_0, \\ -D(\mu)\dfrac{\partial u(\mu)}{\partial n} = \phi_e, & \text{on } I \times \partial\Omega_n, \end{cases} \tag{4.69}$$

   with $\phi_e = 3\text{K}{\cdot}\text{m}{\cdot}\text{s}^{-1}$ and

$$\partial\Omega_0 = (-2, 2) \times \{2\} \cup \{2\} \times (-2, 2), \tag{4.70}$$
$$\partial\Omega_n = (-2, 2) \times \{-2\} \cup \{-2\} \times (-2, 2). \tag{4.71}$$

   Thus, the resulting problem (4.68)−(4.69) is linear. Note that $\partial\Omega_0$ consists of the upper and right sides of the plate and $\partial\Omega_n$ consists of its lower and left sides, so that $\partial\Omega = \overline{\partial}\Omega_0 \cup \overline{\partial}\Omega_n$.

2. **Nonlinear heat equation:** We apply Stefan–Boltzmann boundary conditions on $\partial\Omega$, i.e.,

$$-D(\mu)\frac{\partial u}{\partial n} = \sigma\varepsilon(u^4 - u_r^4), \qquad \text{on } I \times \partial\Omega, \tag{4.72}$$

   where $u_r = 303.15\text{K}$ (30^oC) is an enclosure temperature, $\sigma = 5.67 \times 10^{-8}\text{W.m}^{-2}\text{.K}^{-4}$ is the Stefan–Boltzmann constant and $\varepsilon = 3.10^{-3}$ is the emissivity. The Stefan–Boltzmann boundary condition is nonlinear and so is the resulting problem (4.68)−(4.72).

In what follows, the background spaces $\mathcal{Z}_N$ will be generated by solving either the linear PDE (4.68)−(4.69) or the nonlinear PDE (4.68)−(4.72) with a uniform diffusivity function $D(\mu)$ such that

$$D(\mu)(x) = D_{\text{uni}}(\mu)(x) := \mu \mathbf{1}_\Omega(x), \quad \forall x \in \Omega. \tag{4.73}$$

4.5.2 Synthetic data generation

We synthesize the data by first synthesizing a true solution and then applying to it the linear functionals by means of their Riesz representations in the observable space $\mathcal{U}_M$. In order to synthetize the true solution, we consider a 'true model' based on the bi-material plate (cf. Figure 4.4) where we choose a fixed internal diffusivity $D_{\text{int}} = 1$ and define, for each $\mu \in \mathcal{P}$, the diffusivity function $D(\mu)$ as

$$D(\mu)(x) = D_{\text{syn}}(\mu)(x) := \mu D_{\text{int}} \mathbf{1}_{\Omega_{\text{ext}}}(x) + D_{\text{int}} \mathbf{1}_{\Omega_{int}}(x), \quad \forall x \in \Omega, \tag{4.74}$$

where

$$\Omega_{\text{int}} = (-1,1)^2, \quad \text{and} \quad \Omega_{\text{ext}} = (-2,2)^2 \backslash (-1,1)^2, \tag{4.75}$$

so that $\overline{\Omega} = \overline{\Omega}_{\text{int}} \cup \overline{\Omega}_{\text{ext}}$ and $\Omega_{\text{int}} \cap \Omega_{\text{ext}} = \varnothing$. The synthetic true solutions are then defined as the solutions of (4.68) for all $\mu \in \mathcal{P}$, with either the linear boundary condition (4.69) or the nonlinear boundary condition (4.72).

4.5.2.1 Test configurations

In order to investigate the PBDW formulation, we perform test cases on two distinct configurations:

1. **Perfect model:** The `bk` model is said to be perfect when $\epsilon^{\text{bk}}_{\text{mod}}(u^{\text{true}}(\omega)) = 0$, for every $\omega \in \Theta$ (see (4.4)) (we recall that ω represents the unanticipated uncertainty). In this situation, $u^{\text{true}}(\omega) \in \mathcal{M}^{\text{bk}}$ for all $\omega \in \Theta$. Although the model is perfect, some discrepancies between the HF solutions and the measurements might arise from model-order reduction since $\mathcal{M}^{\text{bk}} \neq \mathcal{Z}_N$ (cf. Figure 4.1). Note that this scenario seldom occurs in engineering situations. This test configuration is meant to assess the accuracy of the PBDW formulation when the observable space $\mathcal{U}_M$ scarcely has additional information compared to $\mathcal{Z}_N$.

2. **Imperfect model:** The `bk` model is said to be imperfect when the modeling error does not vanish. In this situation, there exists at least one (and in general many) $\omega \in \Theta$ such that $\epsilon^{\text{bk}}_{\text{mod}}(u^{\text{true}}(\omega)) \neq 0$, i.e., $u^{\text{true}}(\omega) \notin \mathcal{M}^{\text{bk}}$. Consider for instance the plates in Figures 4.3 and 4.4. If the true solution is generated synthetically using the bimaterial plate, an example of an imperfect `bk` model can be the one for which we solve the same PDE that has generated the true states without accounting for the difference in diffusivity between the subdomains of the plate.

4.5.3 Background space construction via `POD-greedy`

In this section, four test cases are considered to study the PBDW approach.

- Test case (a): Linear perfect.
- Test case (b): Linear imperfect.

- Test case (c): Nonlinear perfect.
- Test case (d): Nonlinear imperfect.

4.5.3.1 Linear case

Regarding time discretization, we consider the time interval $I = [0, 10]$s, the set of discrete times nodes $\mathbb{K}^{\text{tr}} = \{1, \ldots, 200\}$, and a constant time step $\Delta t^k = 0.05$s for all $k \in \mathbb{K}^{\text{tr}}$. Finally, we introduce the parameter interval $\mathcal{P} = [0.05, 1]$ and the training set $\mathcal{P}^{\text{tr}} = 0.05 \times \{1, \ldots, 20\}$. In Figure 4.5, we show the HF temperature

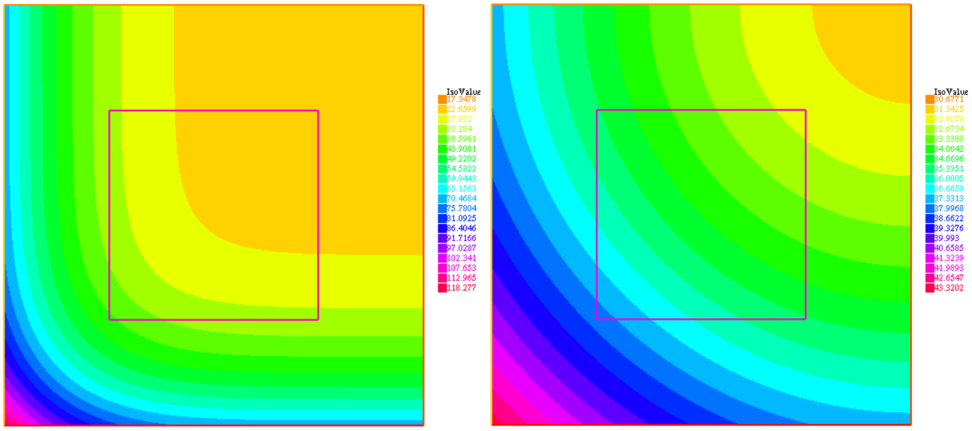

Figure 4.5 – Test cases (a) and (b) : HF solutions for the `bk` model with Neumann boundary conditions. Left: $\mu = 1$ in D_{uni} (values from 17.3^{o}C to 118.3^{o}C). Right: $\mu = 20$ in D_{uni} (values from 30.7^{o}C to 43.3^{o}C).

profiles for the model problem (4.68)−(4.69) over the homogeneous plate at the end of the simulation, i.e., for $t^K = 10$s and for two parameter values. We recall that these solutions will be used as true solutions for the perfect linear case. As the time evolves, the energy related to the flux ϕ_e propagates through the plate which is progressively heated. Moreover, the overall temperature is higher for smaller values of the parameter μ than for larger values. As physically expected, the thermal diffusion over the plate is stronger for larger values of μ than for smaller values.

Test case (a): Linear perfect model We consider the case of a perfect `bk` model for which the diffusivity is uniform over the entire domain Ω. Thus, the true solutions correspond to the HF computations of the `bk` model. The resulting trajectories are reduced using the `POD-greedy` algorithm. For instance, for a tolerance value $\epsilon_{\text{POD}} = 10^{-2}$, the background space $\mathcal{Z}_N$ is composed of $N = 5$ modes. Regarding observations, the initial set $\mathcal{Q}^{\text{init}}$ is obtained using $M = \text{Card}(\mathcal{Q}^{\text{init}}) = 121$ sensors that are uniformly placed over the plate (see Figure 4.3). Using both the background space $\mathcal{Z}_N$ and the observable space $\mathcal{U}_M(\mathcal{Q}^{\text{init}})$, we build the offline matrices $\mathbf{A}$ and $\mathbf{B}$. During the online stage, we estimate the state $u^*_{N,M}$ for every parameter μ in the training set $\mathcal{P}^{\text{tr}}$. Using the weighted H^1-norm, the state estimation relative H^1-error

$e^k(\mu)$ defined as

$$e^k(\mu) := \frac{\|u^{k,\text{true}}(\mu) - u^{k,*}_{N,M}(\mu)\|_{H^1(\Omega)}}{\|u^{k,\text{true}}(\mu)\|_{H^1(\Omega)}}, \quad \forall \mu \in \mathcal{P}, \tag{4.76}$$

is displayed in Figure 4.6 as a function of the value of the parameter μ for several values of ϵ_{POD}. In this first configuration, one can notice that the error decreases

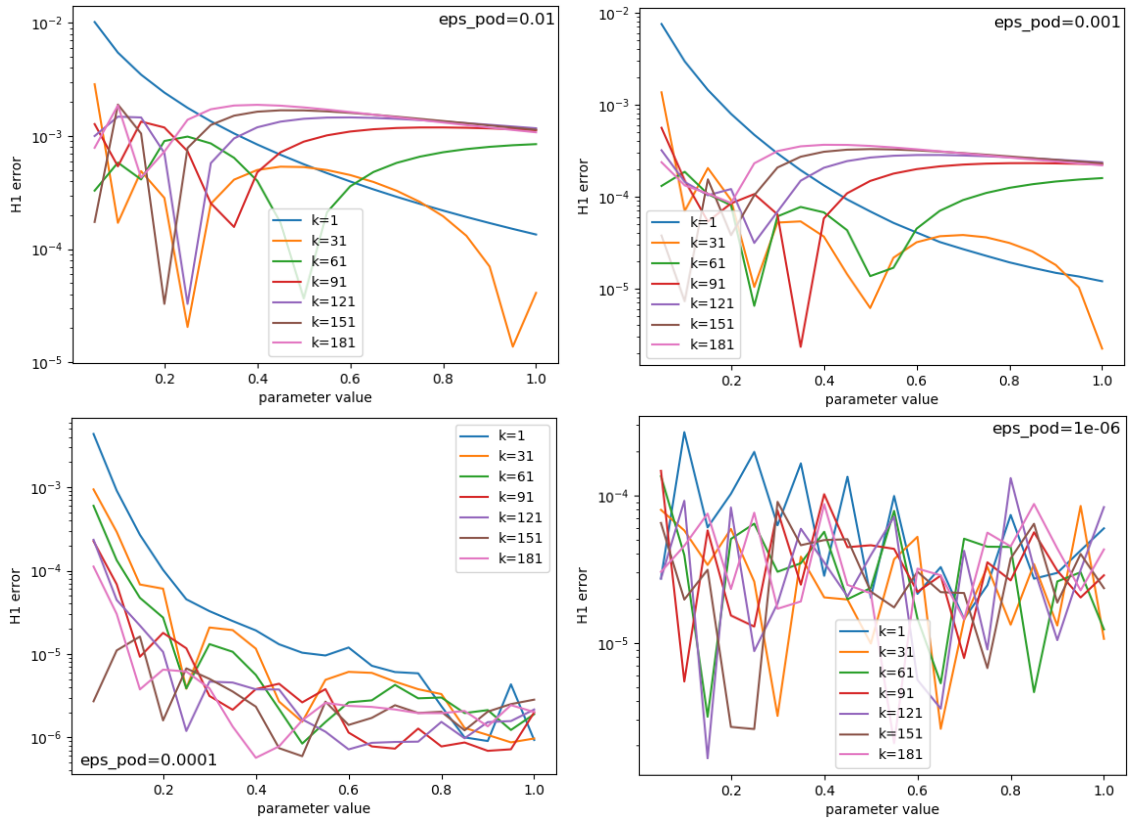

Figure 4.6 – Test case (a): Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\text{tr}}$. Top left: $\epsilon_{\text{POD}} = 10^{-2}$ ($N = 5$). Top right: $\epsilon_{\text{POD}} = 10^{-3}$ ($N = 7$). Bottom left: $\epsilon_{\text{POD}} = 10^{-4}$ ($N = 10$). Bottom right: $\epsilon_{\text{POD}} = 10^{-6}$ ($N = 15$).

for smaller tolerances ϵ_{POD}, i.e., with the dimension N of the background space $\mathcal{Z}_N$. However, the bottom-right panel of Figure 4.6 shows a starting increase in the relative H^1-error $e^k(\mu)$ for $\epsilon_{\text{POD}} = 10^{-6}$ and an oscillatory behaviour of the relative H^1-error $e^k(\mu)$. Although counter-intuitive in the reduced-basis context, this phenomenon is due to the deterioration of the stability constant $\beta_{N,M}$. This observation confirms the claims made in Remark 4.4.

Test case (b): Linear imperfect model This second test investigates the case of a linear imperfect `bk` model. In Figure 4.7, we show the HF temperature profiles for the true solutions over the bimaterial plate at the end of the simulation, i.e., at $t^K = 10$s and for two different parameter values. The temperature fields exhibit the same overall behaviour as in Figure 4.5. Additionally, we notice that the difference

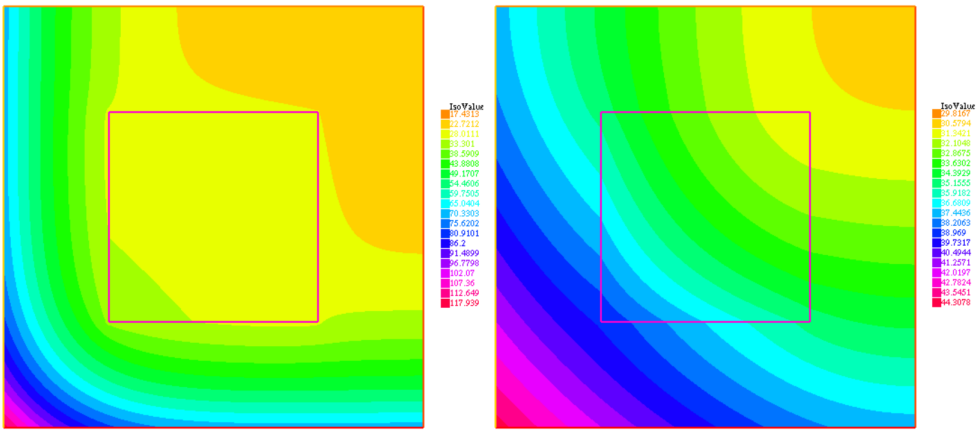

Figure 4.7 – Test case (b) : Synthetic true solutions with Neumann boundary conditions. Left: $\mu = 1$ in $D_{\text{syn}}(\mu)$ (values from 17.4^{o}C to 117.9^{o}C). Right: $\mu = 20$ in $D_{\text{syn}}(\mu)$ (values from 29.8^{o}C to 44.3^{o}C).

in diffusivity between Ω_{int} and Ω_{ext} leads as expected to a kink in the temperature isolines. When $\mu < 1$, the thermal diffusion is stronger in the inner plate corresponding to Ω_{int}, whereas for $\mu > 1$, the thermal diffusion is weaker in the inner plate. Using the HF trajectories produced by the `bk` model, we generate a background space $\mathcal{Z}_N$ by means of a `POD-greedy` algorithm. We use $M = 121$ observations to build the observable space $\mathcal{U}_M$. The relative H^1-errors $e^k(\mu)$ defined in (4.76) are shown in Figure 4.8 as a function of the value of the parameter μ. For instance, for a tolerance value $\epsilon_{\text{POD}} = 10^{-3}$, $\mathcal{Z}_N$ is spanned by $N = 7$ vectors. Notice that the error vanishes for $\mu = 0.5$ since this configuration is equivalent to a perfect `bk` model. However, the bottom panels of Figure 4.8 show a gradual error increase with the dimension N of the `bk` space. This tendency was already observed for the linear perfect test case, although in smaller proportions. As before, the stability constant $\beta_{N,M}$ is degraded when increasing the dimension N of the background space $\mathcal{Z}_N$. Moreover, the enrichment of $\mathcal{Z}_N$ does not add relevant modes anymore (in terms of associated singular values). For the sake of comparison, we enrich the observable space $\mathcal{U}_M$ such that $M = 676$ and plot the relative H^1-errors $e^k(\mu)$ for the same values of ϵ_{POD} in Figure 4.9. Our interpretation is confirmed since the stability issues do not arise anymore for $\epsilon_{\text{POD}} = 10^{-4}$. Owing to the increase of M, the stability decrease with respect to N is somewhat compensated. Finally, the bottom-right panel of Figure 4.9 shows the beginning of an error increase. Using the same reasoning as above, we conclude that more observations are needed for $\epsilon_{\text{POD}} = 10^{-6}$.

Let us now investigate the relative H^1-errors as a function of the dimension M of the observable space $\mathcal{U}_M$. Figure 4.10 shows that the larger the set of observations, the smaller the error. Finally, we visualize the stability constant $\beta_{N,M}$ as a function of the number of observations M in Figure 4.11. The left panel of the figure shows a single curve for clarity, whereas the right panel includes curves for several values of the tolerance ϵ_{POD} (note that the two panels do not use the same rule). As expected, for a constant value of N, the more the observations, the better the stability. For a number of observations $M = 3000$, the PBDW formulation is perfectly stable (or

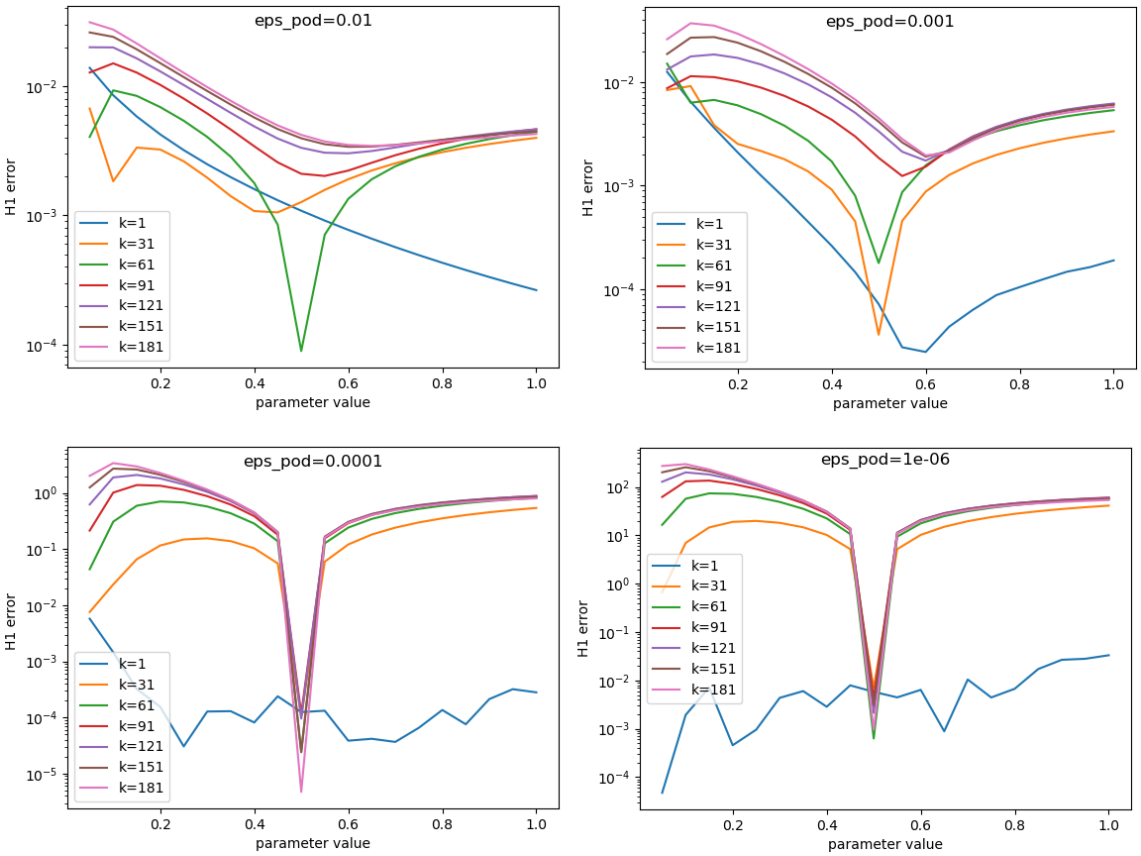

Figure 4.8 – Test case (b) : Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\mathrm{tr}}$ and $M = 121$. Top left: $\epsilon_{\mathrm{POD}} = 10^{-2}$ ($N = 5$). Top right: $\epsilon_{\mathrm{POD}} = 10^{-3}$ ($N = 7$). Bottom left: $\epsilon_{\mathrm{POD}} = 10^{-4}$ ($N = 10$). Bottom right: $\epsilon_{\mathrm{POD}} = 10^{-6}$ ($N = 15$).

close to) for all the considered values of ϵ_{POD}.

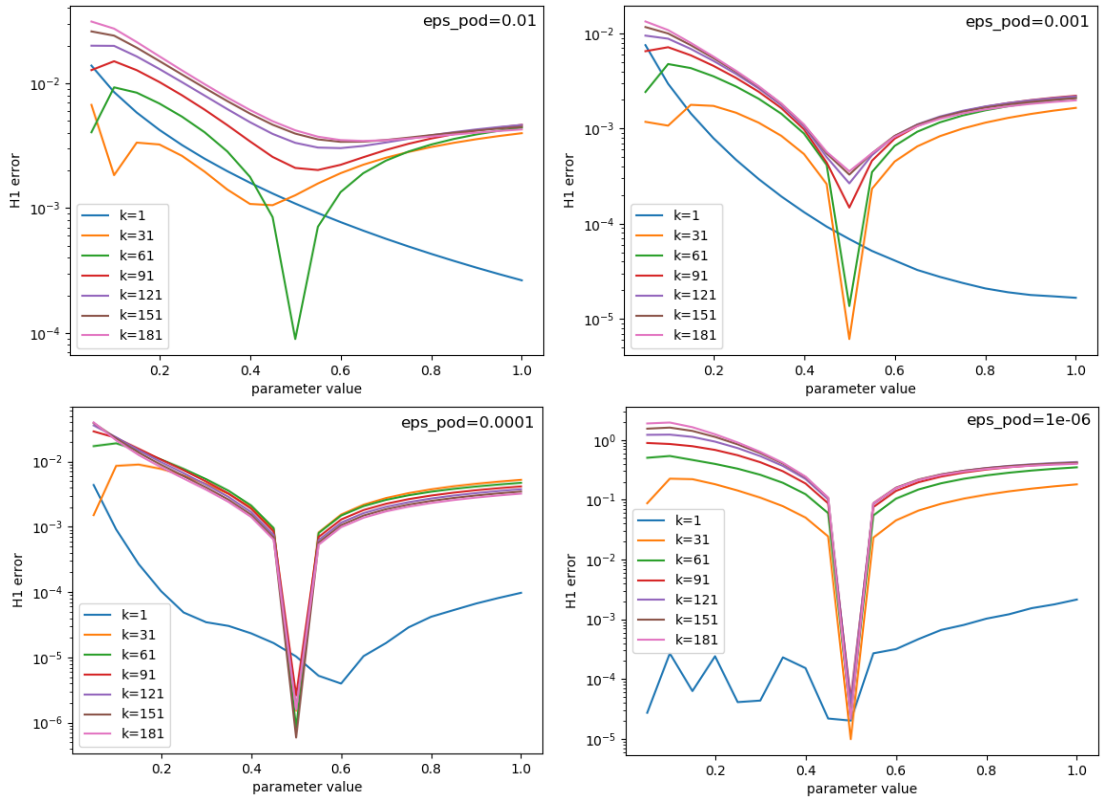

Figure 4.9 – Test case (b) : Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\text{tr}}$ and $M = 676$. Top left: $\epsilon_{\text{POD}} = 10^{-2}$ ($N = 5$). Top right: $\epsilon_{\text{POD}} = 10^{-3}$ ($N = 7$). Bottom left: $\epsilon_{\text{POD}} = 10^{-4}$ ($N = 10$). Bottom right: $\epsilon_{\text{POD}} = 10^{-6}$ ($N = 15$).

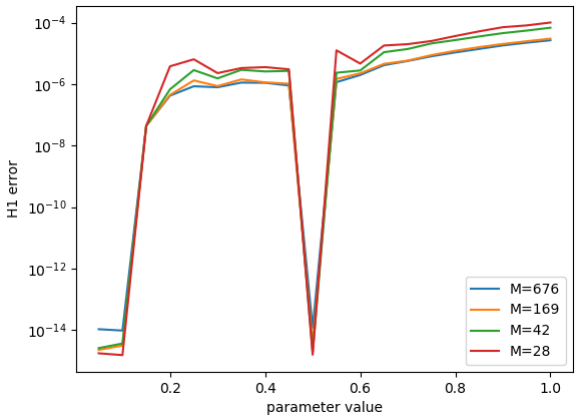

Figure 4.10 – Test case (b) : Relative H^1-error $e^k(\mu)$ as a function of the number M of observations for $t^K = 10$.

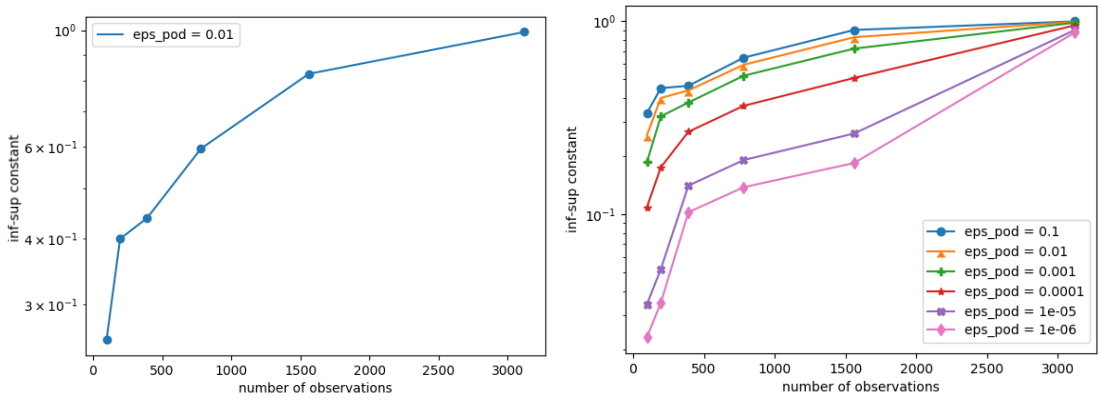

Figure 4.11 – Test case (b) : Stability constant $\beta_{N,M}$ as a function of M. On the right panel, the values of N are respectively 3, 5, 7, 10, 13, 15 for the values of ϵ_{POD} in decreasing order.

4.5.3.2 Nonlinear case

Here, we consider the PDE (4.68)−(4.72) with $u_r = 303.15\text{K}$, $\sigma = 5.67 \times 10^{-8}\text{W.m}^{-2}\text{.K}^{-4}$ and $\varepsilon = 3.10^{-3}$. Except for the parameter interval $\mathcal{P} = [0.1, 2]$, the set $\mathcal{P}^{\text{tr}} = \{0.1i, 1 \leqslant i \leqslant 20\}$ and the time step $\Delta t^k = 0.1$, all the other numerical data remain the same as for the linear test case from the previous section.

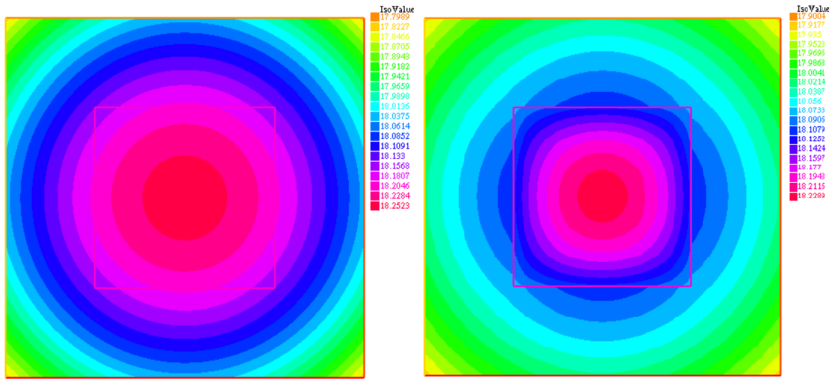

Figure 4.12 – Test cases (c) and (d) : Left: HF solution for the `bk` model (values from 17.80^{o}C to 18.25^{o}C). Right: Synthetic true solution using a bi-material plate (values from 17.90^{o}C to 18.23^{o}C).

4.5.3.3 Test case (c): Nonlinear perfect model

We consider the case with a perfect `bk` model. Thus, the true solutions correspond to the HF computations of the `bk` model (cf. left panel of Figure 4.12). The resulting trajectories are reduced using the `POD-greedy` algorithm. For instance, for a tolerance value $\epsilon_{\text{POD}} = 10^{-2}$, the background space $\mathcal{Z}_N$ consists of $N = 3$ modes. Regarding observations, the initial set $\mathcal{Q}^{\text{init}}$ is obtained using $M = \text{Card}(\mathcal{Q}^{\text{init}}) = 121$ sensors that are uniformly placed over the plate (see Figure 4.3). During the online stage, we estimate the state $u^*_{N,M}$ for every parameter μ in the training set $\mathcal{P}^{\text{tr}}$. In Figure 4.13, we display the state estimation relative H^1-error $e^k(\mu)$ defined in (4.76) as a function of the value of the parameter μ for several values of ϵ_{POD}. In contrast to the linear case, the error always decreases for smaller tolerances ϵ_{POD}, i.e., with the dimension N of the background space $\mathcal{Z}_N$. However, we expect that, for some very small tolerance value (e.g. ϵ_{POD} such that $N > M$), the stability issues mentioned above would arise again.

4.5.3.4 Test case (d): Nonlinear imperfect model

This test case investigates a nonlinear imperfect `bk` model for which the HF `bk` solutions and the true solutions are respectively displayed in the left and right panels of Figure 4.12. The temperature profile for the true solution over the bimaterial plate at the end of the simulation, i.e., at $t^K = 10\text{s}$ clearly shows a different behaviour at the boundaries of the inner material. Regarding the PBDW state estimation, Figure 4.14 shows the relative H^1-error $e^k(\mu)$ defined in (4.76) using $M = 121$

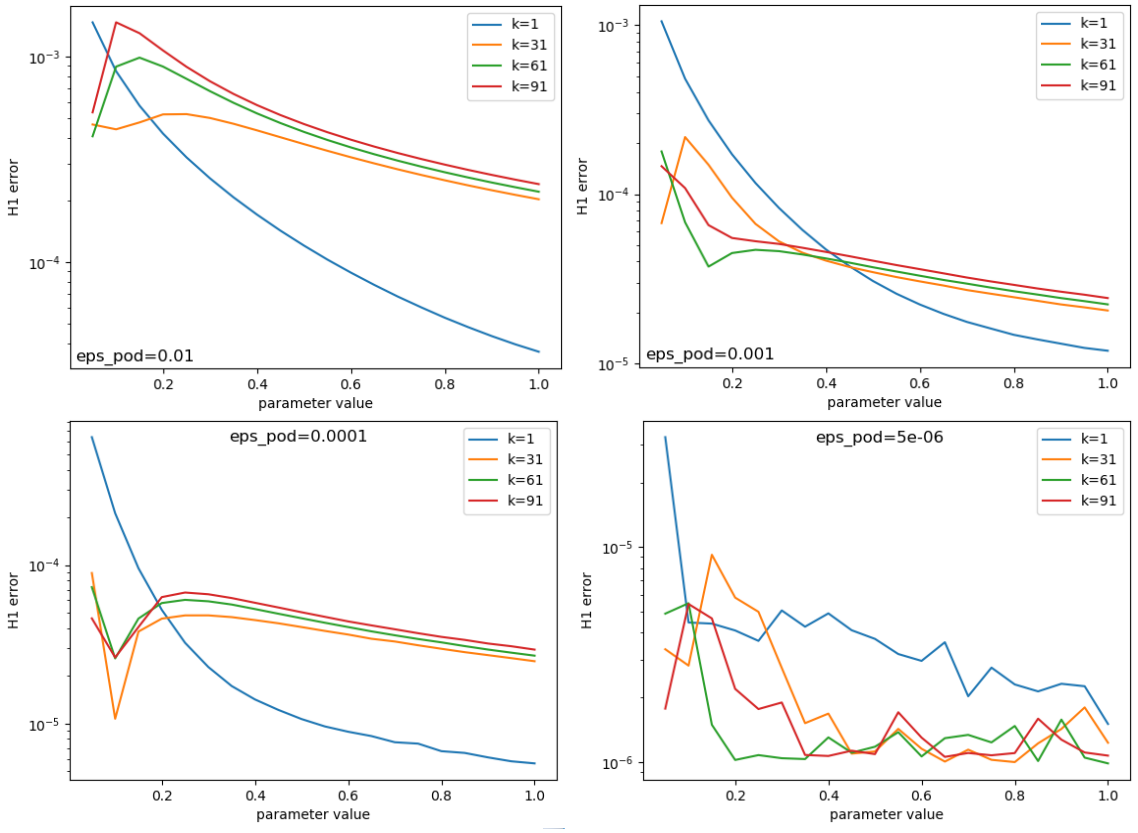

Figure 4.13 – Test case (c) : Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\text{tr}}$ and $M = 121$. Top left: $\epsilon_{\text{POD}} = 10^{-2}$ $(N = 3)$. Top right: $\epsilon_{\text{POD}} = 10^{-3}$ $(N = 5)$. Bottom left: $\epsilon_{\text{POD}} = 10^{-4}$ $(N = 7)$. Bottom right: $\epsilon_{\text{POD}} = 5.10^{-6}$ $(N = 11)$.

observations to build the observable space $\mathcal{U}_M$. For $\epsilon_{\text{POD}} = 10^{-3}$, $\mathcal{Z}_N$ is spanned by $N = 5$ vectors. Notice that the error vanishes for $\mu = 0.25$ since this configuration is equivalent to a perfect `bk` model. We notice that the relative H^1-error $e^k(\mu)$ increases because the stability constant decreases. Figure 4.15 visualizes the relative H^1-error $e^k(\mu)$ for a higher number of observations $M = 676$. We observe that augmenting the dimension of the observable space $\mathcal{U}_M$ cures the stability issues. Also, the errors are lower owing to the higher number of observations. Finally, Figure 4.16 shows the stability constant $\beta_{N,M}$ as a function of the number of observations M. The behaviour is quite similar to the linear case. Hence, the nonlinear character of the problem does not influence the overall features of the PBDW statement. This observation corroborates the independence with regard to the `bk` model.

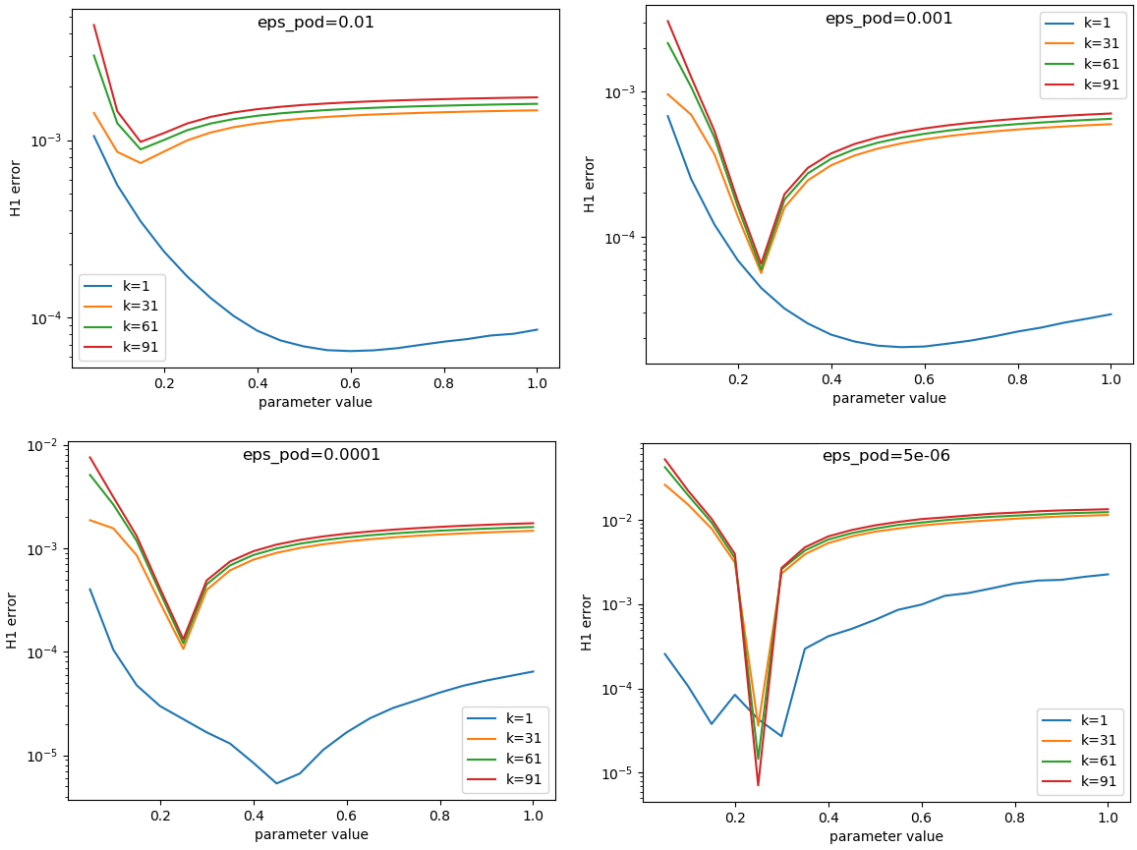

Figure 4.14 – Test case (d) : Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\text{tr}}$ and $M = 121$. Top left: $\epsilon_{\text{POD}} = 10^{-2}$ $(N = 3)$. Top right: $\epsilon_{\text{POD}} = 10^{-3}$ $(N = 5)$. Bottom left: $\epsilon_{\text{POD}} = 10^{-4}$ $(N = 7)$. Bottom right: $\epsilon_{\text{POD}} = 10^{-6}$ $(N = 11)$.

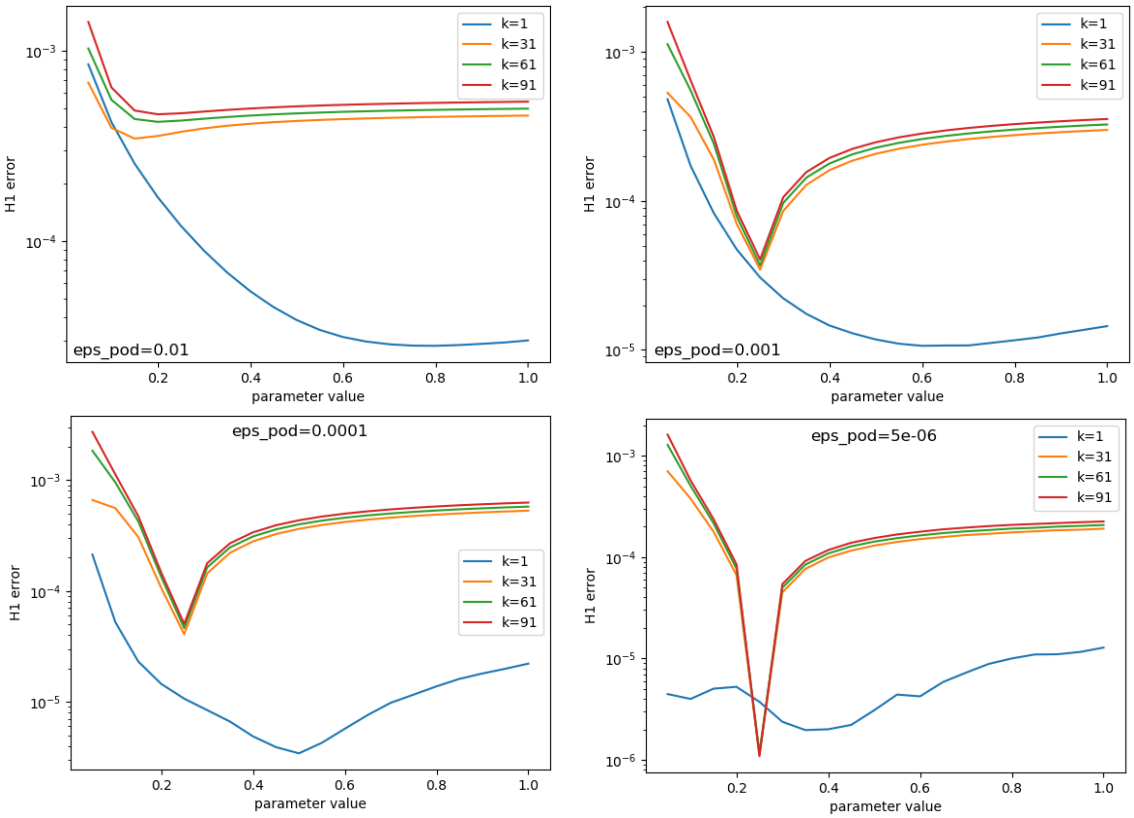

Figure 4.15 – Test case (d) : Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\text{tr}}$ and $M = 676$. Top left: $\epsilon_{\text{POD}} = 10^{-2}$ $(N = 3)$. Top right: $\epsilon_{\text{POD}} = 10^{-3}$ $(N = 5)$. Bottom left: $\epsilon_{\text{POD}} = 10^{-4}$ $(N = 7)$. Bottom right: $\epsilon_{\text{POD}} = 5.10^{-6}$ $(N = 11)$.

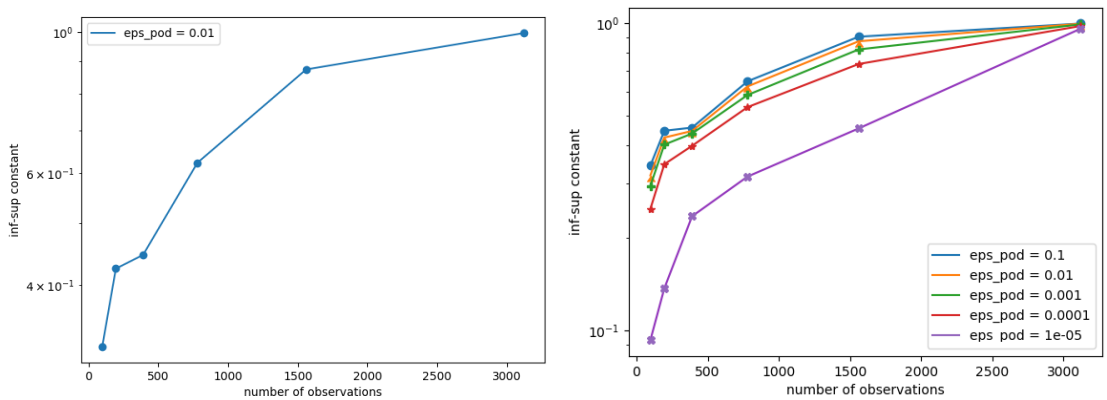

Figure 4.16 – Test case (d) : Stability constant $\beta_{N,M}$. On the right panel, the values of N are respectively 2, 3, 5, 7, 11 for the values of ϵ_{POD} in decreasing order.

4.5.4 Background space construction via state estimation

We now illustrate the performances of Algorithm 4.2 for the following linear imperfect case:

- Test case (e): We consider a simulation duration $T = 4$s and a time step $\Delta t = 0.1$s. Test truths are synthetized with an internal diffusivity $D_{\text{int}} = 0.2$.

As opposed to the previous section, we choose a non-parametric `bk` model based on an HF computation for $\mu = 0.5$. The resulting unique trajectory is then reduced using a `POD` algorithm, which is equivalent to a `POD-greedy` for a single trajectory (cf. line 1 of Algorithm 4.2). For a tolerance value $\epsilon_{\text{POD}}^{\text{init}} = 10^{-2}$, we obtain a background space $\mathcal{Z}_{N^{\text{init}}}^{\text{init}}$ composed of $N^{\text{init}} = 34$ modes. Figure 4.17 shows the singular values that are retained. As regards observations, the initial set $\mathcal{Q}^{\text{init}}$ consists of

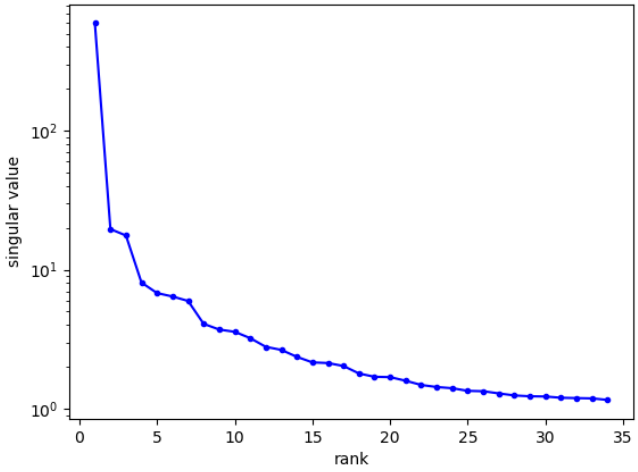

Figure 4.17 – Test case (e) : Singular values for $\epsilon_{\text{POD}}^{\text{init}} = 10^{-2}$.

$M^{\text{init}} = \text{Card}(\mathcal{Q}^{\text{init}}) = 1521$ sensors that are uniformly placed over the plate (cf. line 2 of Algorithm 4.2). Using both the background space $\mathcal{Z}_{N^{\text{init}}}^{\text{init}}$ and the observable space $\mathcal{U}_{M^{\text{init}}}^{\text{init}}(\mathcal{Q}^{\text{init}})$, we estimate the state $u_{N^{\text{init}},M^{\text{init}}}^{*}$ for every parameter μ in the training set $\mathcal{P}^{\text{tr}} = \{0, 4, 8, 12, 16\}$ (cf. line 4 of Algorithm 4.2). The state estimation leads to the relative H^1-error $e^k(\mu)$ shown in Figure 4.18. We also plot in Figure 4.19 the absolute H^1-norms of the deduced background estimate $z_{N^{\text{init}},M^{\text{init}}}^{*}$ and the update estimate $\eta_{N^{\text{init}},M^{\text{init}}}^{*}$. One can notice that the latter is non-negligible compared to the former. Once the first part of the modified offline stage has been performed, we use the resulting state estimates in order to build the modified background space (cf. line 5 of Algorithm 4.2). For a tolerance $\epsilon_{\text{POD}} = 5.10^{-2}$, the `POD-greedy` algorithm selects four modes. Then, we build the observable space $\mathcal{U}_M$ using $M = 121$ uniformly distributed sensors (the optimal choice can be made using the `S-Greedy` algorithm, see Algorithm 4.3). Figure 4.20 displays the errors for the verification set $\mathcal{P}^{\text{verif}} = \{0, \ldots, 19\}$. The state estimation relative H^1-error $e^k(\mu)$ remains comparable to that of the five parameters used for the offline construction. Regarding the online observations, we highlight that the online results are achieved using only $M \approx 8\% M^{\text{init}}$. Finally, Figure 4.21 shows the absolute H^1-norms of

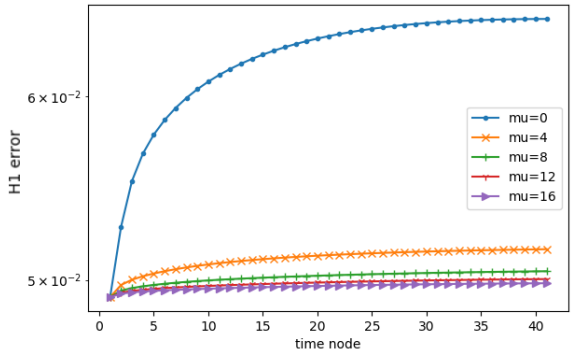

Figure 4.18 – Test case (e) : Relative H^1-error $e^k(\mu)$ for the state estimate as a function of the time nodes. The various curves correspond to the different values of μ.

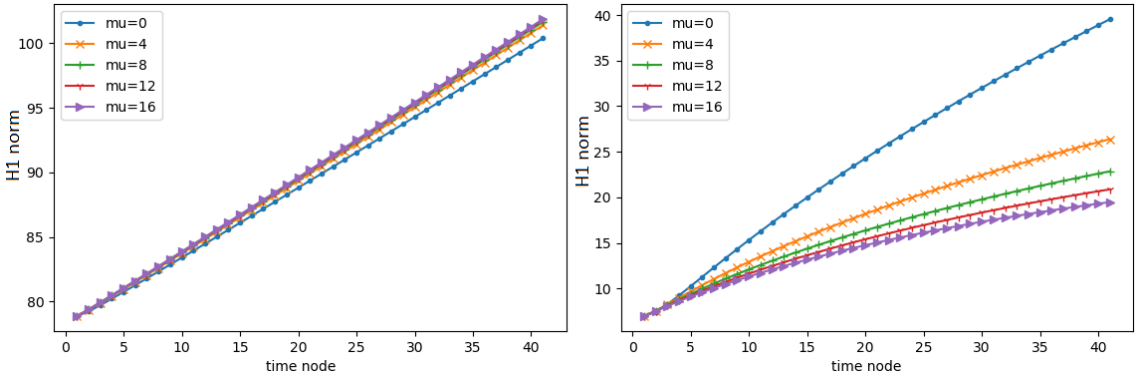

Figure 4.19 – Test case (e) : Absolute H^1-norms of the contributions $z^*_{N^{\text{init}},M^{\text{init}}}$ and $\eta^*_{N^{\text{init}},M^{\text{init}}}$ as a function of the time nodes. The various curves correspond to the different values of μ.

the deduced background estimate $z^*_{N,M}$ and the update estimate $\eta^*_{N,M}$. We observe that the update estimate $\eta^*_{N,M}$ has a lower norm compared to Figure 4.19, whereas the deduced background estimate $z^*_{N,M}$ has a larger norm. This is due to the offline inclusion of observations in the new background space $\mathcal{Z}_N$ through offline state estimation. Therefore, we deduce that the modified offline algorithm achieves the expected objective.

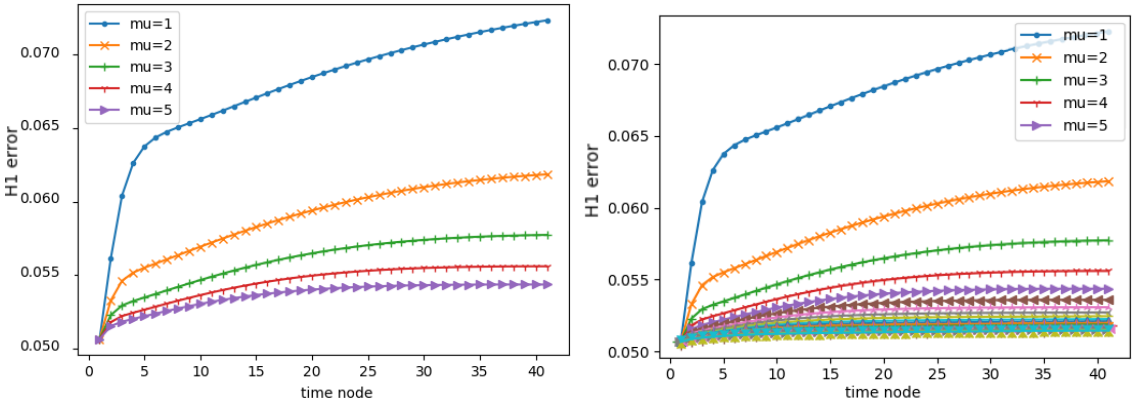

Figure 4.20 – Test case (e) : Relative H^1-error $e^k(\mu)$ for the state estimate as a function of the time nodes during the online stage. The various curves correspond to the different values of μ. Left: for all $\mu \in \mathcal{P}^{\text{tr}}$. Right: for all $\mu \in \mathcal{P}^{\text{verif}}$.

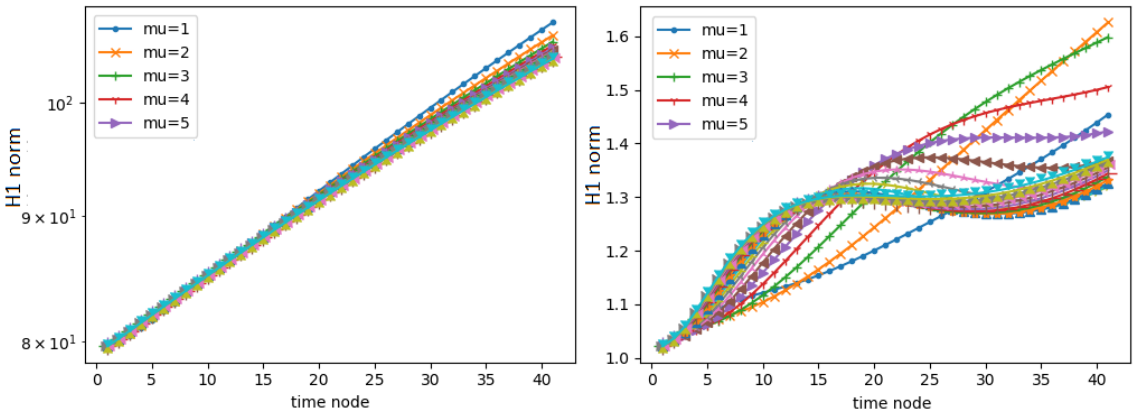

Figure 4.21 – Test case (e) : Absolute H^1-norms of the contributions $z^*_{N,M}$ and $\eta^*_{N,M}$ as a function of the time nodes. The various curves correspond to the different values of μ.

CHAPTER 5

CONCLUSIONS AND PERSPECTIVES

In this thesis, we have devised three new methodologies in the Reduced Basis (RB) context. First, we have introduced a Progresssive RB Empirical Interpolation Method (PREIM) that allows one to diminish the offline costs incurred in the nonlinear RB method applied to unsteady nonlinear PDEs. The main reason for this computational gain is that the computation of high-fidelity trajectories is the dominant part of the offline cost. Numerical tests on both two-dimensional and industrial three-dimensional heat transfer problems with nonlinear thermal conductivities have illustrated the computational efficiency and the accuracy of the proposed algorithm. In the present study, the computational benefits of PREIM have been evaluated by comparing the results to those of the reference method, i.e. the standard RB-EIM and also to the Simultaneous EIM-RB method (SER) which is the closest progressive method to PREIM available in the literature. Comparisons with other progressive RB-EIM methods, such as those mentioned in the introductive section of Chapter 2, and a more theoretical study of PREIM can be considered for future work. Another relevant perspective is the application of PREIM to a more systematic study of 3D flow regulation in industrial applications related to nuclear reactor operation.

Second, we have presented a RB scheme for parametrized variational inequalities, more particularly in the framework of contact mechanics. We have addressed the conceptional issue raised by nonlinear inequality constraints in the RB scheme when considering a general setting where neither small displacements nor matching meshes are assumed, as was the case so far in the literature. The present RB scheme allows one to reduce the dual basis of Lagrange multipliers while maintaining their positivity by means of a greedy algorithm based on the projection onto a convex cone. A systematic comparison of the present method to the few methods so far available in the literature such as the Non-negative Matrix Factorization (NMF) and Hyper-Reduction (HR) is a relevant perspective. The method that we have proposed can be used under a local convexity assumption on the solids that come into contact. A challenging research perspective is the extension of our developments to concave solids and to multi-body contact problems. Moreover, we have only considered

equilibrium problems. Thus, the treatment of dynamic contact problems remains on the agenda.

Third, we have addressed data assimilation in the context of RB schemes. Our first contribution is the extension of the Parametrized-Background Data-Weak (PBDW) approach to the time-dependent setting by means of a `POD-greedy` reduced basis construction. Since the construction of the basis is performed offline, the algorithm renders the time dependence of the problem we are addressing while the time stepping scheme remains unchanged. An interesting research direction is to devise an online time-stepping PBDW scheme so as to take advantage of the potential interactions between the time steps. As a second contribution, we have devised a modified offline algorithm that exploits offline state estimates in order to diminish both the dimension of the online PBDW statement and the number of required sensors collecting data. The idea is to exploit *in situ* observations in order to update the best-knowledge model, thereby improving the approximation capacity of the background space. The proposed algorithm is sequential, i.e., we first build the background space before chosing the observable space. An interesting perspective is to study a progressive construction of both spaces. As regards the quality of the spaces built offline, the PBDW suggests separate criteria for the background space and the observable space. Hence, another relevant research direction is to study a joint stopping criterion for the construction of both spaces.

BIBLIOGRAPHY

[1] B. Almroth, P. Stern, and F. A Brogan. Automatic choice of global shape functions in structural analysis. *Aiaa Journal*, 16(5):525–528, 1978.

[2] M. S. Andersen, J. Dahl, and L. Vandenberghe. Cvxopt: A python package for convex optimization. Open source on `http://www.abel.ee.ucla.edu/cvxopt`, 2008.

[3] J-P. Argaud, B. Bouriquet, F. de Caso, H. Gong, Y. Maday, and O. Mula. Sensor placement in nuclear reactors based on the generalized empirical interpolation method. *J. Comput. Phys.*, 363:354–370, 2018.

[4] M. Balajewicz, D. Amsallem, and C. Farhat. Projection-based model reduction for contact problems. *Internat. J. Numer. Methods Engrg.*, 106(8):644–663, 2016.

[5] 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. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

[6] A. Barrett and G. Reddien. On the reduced basis method. *Z. Angew. Math. Mech.*, 75(7):543–549, 1995.

[7] M. Baudin, A. Dutfoy, B. Iooss, and A-L. Popelin. Openturns: An industrial software for uncertainty quantification in simulation. 2015.

[8] A. Benaceur, V. Ehrlacher, A. Ern, and S. Meunier. A Progressive Reduced Basis/Empirical Interpolation Method for Nonlinear Parabolic Problems. *SIAM J. Sci. Comput.*, 40(5):A2930–A2955, 2018.

[9] M. Berveiller. *Eléments finis stochastiques: approches intrusive et non intrusive pour des analyses de fiabilité*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2005.

[10] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.*, 43(3):1457–1472, 2011.

[11] G. Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. Phd thesis, Université Blaise Pascal - Clermont-Ferrand II, October 2009.

[12] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici. *A priori* convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM Math. Model. Numer. Anal.*, 46(3):595–603, 2012.

[13] F. Casenave, A. Ern, and T. Lelièvre. Accurate and online-efficient evaluation of the *a posteriori* error bound in the reduced basis method. *ESAIM Math. Model. Numer. Anal.*, 48(1):207–229, 2014.

[14] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[15] P. Chen and C. Schwab. Sparse-grid, reduced-basis Bayesian inversion. *Comput. Methods Appl. Mech. Engrg.*, 297:84–115, 2015.

[16] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodríguez. Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2D Maxwell's problem. *M2AN Math. Model. Numer. Anal.*, 43(6):1099–1116, 2009.

[17] C. Daversin and C. Prud'homme. Simultaneous empirical interpolation and reduced basis method for non-linear problems. *C. R. Math. Acad. Sci. Paris*, 353(12):1105–1109, 2015.

[18] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.*, 34(2):A937–A969, 2012.

[19] Electricité de France. Finite element *code_aster*, analysis of structures and thermomechanics for studies and research. Open source on www.code-aster.org, 1989–.

[20] A. Ern and J-L. Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[21] J. Fauque, I. Ramière, and D. Ryckelynck. Hybrid hyper-reduced modeling for contact mechanics problems. *Internat. J. Numer. Methods Engrg.*, 115(1):117–139, 2018.

[22] S. Fučík, A. Kratochvíl, and J. Nečas. Kačanov-Galerkin method and its application. *Acta Universitatis Carolinae. Mathematica et Physica*, 15(1):31–33, 1974.

[23] M. A. Grepl. Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations. *Math. Models Methods Appl. Sci.*, 22(3):1150015, 40, 2012.

[24] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[25] M. Gubisch and S. Volkwein. Chapter 1: Proper orthogonal decomposition for linear-quadratic optimal control. In *Model Reduction and Approximation*, pages 3–63.

[26] B. Haasdonk. Convergence rates of the POD-greedy method. *ESAIM Math. Model. Numer. Anal.*, 47(3):859–873, 2013.

[27] B. Haasdonk, J. Salomon, and B. Wohlmuth. A reduced basis method for parametrized variational inequalities. *SIAM J. Numer. Anal.*, 50(5):2656–2676, 2012.

[28] F. Hecht. New developments in freefem++. Open source on `http://www.freefem.org`, 2012.

[29] H. Hertz. Über die berührung fester elastischer körper. *Journal für die reine und angewandte Mathematik*, 1882(92):156–171, 1882.

[30] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. SpringerBriefs in Mathematics. Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016.

[31] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control. In *Dimension reduction of large-scale systems*, volume 45 of *Lect. Notes Comput. Sci. Eng.*, pages 261–306. Springer, Berlin, 2005.

[32] K. C. Hoang, P. Kerfriden, B. C. Khoo, and S. P. A. Bordas. An efficient goal-oriented sampling strategy using reduced basis method for parametrized elasto-dynamic problems. *Numer. Methods Partial Differential Equations*, 31(2):575–608, 2015.

[33] D. B. P. Huynh, D. J. Knezevic, Y. Chen, J. S. Hesthaven, and A. T. Patera. A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Methods Appl. Mech. Engrg.*, 199(29-32):1963–1975, 2010.

[34] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Math. Acad. Sci. Paris*, 345(8):473–478, 2007.

[35] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.*, 90(1):117–148, 2001.

[36] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press.

[37] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Acad. Sci. Paris Sér. I Math.*, 331(2):153–158, 2000.

[38] Y. Maday and O. Mula. A generalized empirical interpolation method: application of reduced basis techniques to data assimilation. In *Analysis and numerics of partial differential equations*, pages 221–235. Springer, 2013.

[39] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau. A general multipurpose interpolation procedure: the magic points. *Commun. Pure Appl. Anal.*, 8(1):383–404, 2009.

[40] Y. Maday, A. T. Patera, J. D. Penn, and M. Yano. A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. *Internat. J. Numer. Methods Engrg.*, 102(5):933–965, 2015.

[41] A. Manzoni and F. Negri. Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs. *Adv. Comput. Math.*, 41(5):1255–1288, 2015.

[42] S. Meunier, J. Ferrari, J-F. Rit, D. Hersant, and J.P. Mathieu. On the influence of flows in clearances for thermal shocks in a globe valve. In *ASME 2017 Pressure Vessels and Piping Conference*, pages V002T02A016–V002T02A016. American Society of Mechanical Engineers, 2017.

[43] F. Negri, A. Manzoni, and G. Rozza. Reduced basis approximation of parametrized optimal flow control problems for the Stokes equations. *Comput. Math. Appl.*, 69(4):319–336, 2015.

[44] B. Noble and J. W. Daniel. *Applied Linear Algebra*. Prentice-Hall, 3rd edition, 1988.

[45] A. K Noor and J. M Peters. Reduced basis technique for nonlinear analysis of structures. *Aiaa journal*, 18(4):455–462, 1980.

[46] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods. *Journal of Fluids Engineering*, 124(1):70–80, November 2001.

[47] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. Patera, and G. Turinici. Reduced–basis output bound methods for parametrized partial differential equations. *Proceedings SMA Symposium*, 2002.

[48] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations*. La Matematica per il 3+2. Springer International Publishing, 2016.

[49] D. V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. *IMA J. Numer. Anal.*, 26(3):423–445, 2006.

[50] T. Taddei. An adaptive parametrized-background data-weak approach to variational data assimilation. *ESAIM Math. Model. Numer. Anal.*, 51(5):1827–1858, 2017.

[51] T. Taddei and A. T Patera. A localization strategy for data assimilation; application to state estimation and parameter estimation. *SIAM J. Sci. Comput.*, 40(2):B611–B636, 2018.

[52] K. Urban and B. Wieland. Affine decompositions of parametric stochastic processes for application within reduced basis methods. *IFAC Proceedings Volumes*, 45(2):716–721, 2012.

[53] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Internat. J. Numer. Methods Fluids*, 47(8-9):773–788, 2005.