Model reduction for forward simulation and inverse problems : towards non-linear approaches

Agustin Somacal

Sorbonne Université

LJLL

Doctoral School **Sciences Mathématiques de Paris Centre**

University Department **Laboratoire Jacques-Louis Lions**

Thesis defended by **Agustín Somacal**

Defended on **May 6, 2024**

In order to become Doctor from Sorbonne Université

Academic Field **Applied Mathematics**

Model reduction for forward simulation and inverse problems: towards non-linear approaches

Thesis supervised by Albert Cohen
Olga Mula

Committee members

Referees	Sébastien Boyaval	Research Director at Ecole des Ponts Paris Tech	
	Francesc Arandiga	Professor at Universidad de Valencia	
Examiners	Damiano Lombardi	Researcher at Inria Paris	
	Bruno Despres	Professor at Sorbonne Université	
	Anthony Nouy	Professor at Ecole Centrale de Nantes	
	Rachida Chakir	Researcher at Université Gustave Eiffel	
	Bruno Despres	Professor at Sorbonne Université	Committee President
Supervisors	Albert Cohen	Professor at Sorbonne Université	
	Olga Mula	Associate professor at Eindhoven University of Technology	

Colophon

Doctoral dissertation entitled “Model reduction for forward simulation and inverse problems: towards non-linear approaches”, written by Agustín Somacal, completed on July 11, 2024, typeset with the document preparation system LaTeX and the yathesis class dedicated to theses prepared in France.

Sorbonne Université

LJLL

École doctorale **Sciences Mathématiques de Paris Centre**

Unité de recherche **Laboratoire Jacques-Louis Lions**

Thèse présentée par **Agustín Somacal**

Soutenue le **6 mai 2024**

En vue de l'obtention du grade de docteur de Sorbonne Université

Discipline **Mathématiques Appliquées**

Réduction de modèle pour des problèmes directs et inverses: vers des approches non linéaires

Thèse dirigée par Albert Cohen
Olga Mula

Composition du jury

Rapporteurs	Sébastien Boyaval	Directeur de Recherche à l'Ecole des Ponts Paris Tech	
	Francesc Arandiga	Professeur à l'Universidad de Valencia	
Examinateurs	Damiano Lombardi	Chargé de recherche à Inria Paris	
	Bruno Despres	Professeur à Sorbonne Université	
	Anthony Nouy	Professeur à Ecole Centrale de Nantes	
	Rachida Chakir	Chargée de recherche à l' Université Gustave Eiffel	
	Bruno Despres	Professeur à Sorbonne Université	président du jury
Directeurs de thèse	Albert Cohen	Professeur à Sorbonne Université	
	Olga Mula	Professeur associée à Eindhoven University of Technology	

Model reduction for forward simulation and inverse problems: towards non-linear approaches

Abstract

Model reduction is a technique used to compute fast and accurate approximations of physical systems' states when they are described through parametric *Partial Differential Equations* (PDEs). In the classical setting a linear subspace is carefully built, in an *offline* stage, using a set of high resolution descriptions of possible states of the system of interest. Afterwards the subspace is used to quickly and accurately solve *forward* or *inverse* problems. It is known that these strategies can approximate well the solution of elliptic PDEs but they fail on hyperbolic PDEs or when states present jump discontinuities. In this context, this thesis focuses on developing efficient non-linear strategies to tackle the limitations of linear approximation spaces.
Chapter 2 extends the approximation guarantees offered by linear spaces for the stationary diffusion equation when extreme levels of contrast in the diffusivity constants are possible.
Chapter 3 presents a theoretical framework to analyse the effectiveness of non-linear strategies for *inverse* problems while Chapter 4 focuses on the practical implementation of high-order techniques to locally reconstruct interfaces from cell average data. In Chapter 5, we show a method to accelerate the reconstruction of 1d characteristic functions by a machine learning strategy trained to learn a mapping from lower order Fourier coefficient values to higher order ones. In Chapter 6, we turn the attention to another learning technique, known as Physics Informed Neural Networks (PINNs), to tackle a linear advection-diffusion equation when the diffusivity vanishes and shocks appear.
Finally, in Chapter 7, we apply a combination of linear and non-linear methods to a real case scenario in which the objective is to predict the pollution on every point in a city using heterogeneous sources of data like temporal pollution series on specified locations, the geometry of the streets, and Google Maps traffic information.
Chapters 2, 3, 5 and 6 are based on the published articles [51, 50, 55, 20] respectively while Chapters 4 and 7 are based on the submitted articles [56, 67].

Keywords: Non-linear approximation, Reduced order modelling, Numerical approximation

Laboratoire Jacques-Louis Lions
Sorbonne Université – Campus Pierre et Marie Curie – 4 place Jussieu – 75005 Paris – France

Réduction de modèle pour des problèmes directs et inverses: vers des approches non linéaires

Résumé

La réduction de modèle est une technique utilisée pour calculer des approximations rapides et précises des états de systèmes physiques, décrits par des *Équations* aux *Dérivées Partielles* (EDP) paramétriques. Dans le cadre classique, un sous-espace linéaire est construit dans une étape *offline* en utilisant un ensemble de descriptions à haute résolution des états possibles du système d'intérêt. Ensuite, le sous-espace est utilisé pour résoudre rapidement et avec précision des problèmes *directes* ou *inverses*. Il est connu que ces stratégies peuvent bien approximer la solution des EDP elliptiques avec peu d'éléments de base mais échouent sur les EDP hyperboliques ou lorsque les états présentent des discontinuités. Dans ce contexte, cette thèse se concentre sur le développement de stratégies non linéaires efficaces pour aborder les limitations des espaces linéaires.
Le Chapitre 2 étend les garanties d'approximation offertes par les espaces linéaires pour l'équation de diffusion stationnaire pour des niveaux extrêmes de contraste dans les champs de diffusion.
Le Chapitre 3 présente un cadre théorique pour analyser l'efficacité des stratégies non linéaires pour les problèmes *inverses* tandis que le Chapitre 4 se concentre sur la mise en œuvre pratique des techniques d'ordre élevé pour reconstruire localement des interfaces à partir des moyennes. Dans le Chapitre 5, nous montrons une méthode pour accélérer la reconstruction de fonctions caractéristiques en 1d par une stratégie d'apprentissage automatique entraînée à fournir une correspondance entre les valeurs des coefficients de Fourier d'ordre inférieur et celles d'ordre supérieur. Dans le Chapitre 6, nous portons notre attention sur une autre technique d'apprentissage connue sous le nom de réseaux neuronaux informés par la physique (PINN) pour traiter une équation de transport-diffusion linéaire lorsque la diffusivité tend vers zéro et que des chocs apparaissent.
Enfin, dans le Chapitre 7, nous appliquons une combinaison de méthodes linéaires et non linéaires à un scénario réel dans lequel l'objectif est de prédire la pollution en tout point d'une ville en utilisant des sources de données hétérogènes telles que des séries temporelles de pollution sur des emplacements spécifiques, la géométrie des rues et les informations de traffic de Google Maps.
Les Chapitres 2, 3, 5 et 6 sont basés sur les articles publiés [51, 50, 55, 20] tandis que les Chapitres 4 et 7 sont basés sur les articles soumis [56, 67].

Mots clés : Approximation non-lineaire, Modeles reduites, Approximation numérique

This thesis has been prepared at

Laboratoire Jacques-Louis Lions
Sorbonne Université
Campus Pierre et Marie Curie
4 place Jussieu
75005 Paris
France

Web Site https://ljll.math.upmc.fr/

Wishing to say too many things I tried and failed; consistently. So I took the pieces and made this weird collage hoping you will infer the rest out of this broken mirror.

Desde la Pampa

Aquí me pongo a cantar
al final *de* esta tarea
que de un mar-rio al Sena,
¡aventura *estraordinaria*!,
me trajera una mañana,
al encuentro de esta tierra.

Pido a las leyes *del* tiempo
que rigen cada momento
me permitan expresar
la gratitud que hoy siento
por tanta gente y sus gestos
que ahora son en mi andar.

The visible hand

It is not an invisible hand
the one that constantly guides our lives.
Neither is that heavy impostor
we are used to name by "I".
It is a very visible one,
though its length may extend
far centuries back afar,
one just have to look behind
to see the faces and the smiles
of those who stand to help us rise.

Where should I start?
Where should I finish?
The task has, indeed, no limits
as the self is nothing more
than one delayed melody
in the world's vast Fuga.

By looking at that mirror
I see, of course, my good advisors,
Albert, Olga;
Not many have the chance
to weekly meet

Discuss and speak
Receive advice and consider, all the while,
one's owns views on the things we had to do.

Albert, merci pour ta patience et passion pour les mathématiques; toujours chaque discussion commencé avec un pointilleux rappel des notations, des motivations que petit à petit j'espère avoir fini pour comprendre. Pour la bienveillance, toujours attentif a que je me trouve bien. Pour me laisser la liberté d' explorer tant de sujets intéressants et à mon propre rythme. Pour me montrer le chemin qu'amène vers l'écriture scientifique, c'était comme voir Borges travailler, j'espère avoir appris au moins une partie de ça.

Olga, gracias por crear un espacio de discusión cálido al que acudir cuando el entendimiento se me ofuscaba. Por tomarte siempre el tiempo de explicarme los mecanismos de métodos y teoremas que me resultaban ajenos. Por alentarme y motivarme, por resaltar lo bueno para poder trabajar en mejorar lo malo.

Merci Rachida, Damiano, Bruno et Anthony pour accepter de faire partie de mon jury de thèse. Aussi, à Anthony pour chaleureusement m'offrir une possibilité pour la suite avec un univers de projets intéressants et motivants que j'espère bientôt explorer. Merci Damiano pour des amicales discussions dans différentes conférences. Merci d'avoir organisé avec Olga et Virginie le CEMRACS 2021, une expérience très fatigante pour vous mais unique et inoubliable pour nous que je garde avec beaucoup d'affection avec l'apprentissage et les amis qui se sont forgés depuis le CIRM. A Bruno pour le bon humeur, le CEMRACS 2023, les conviviales discussions pendant mon temps au Labo et pour nous amener vers l'existence de LVIRA.

Merci Sébastien pour rapporter la thèse, et nous apporter les références derrière la longue histoire de LVIRA. Gracias Paco por el entusiasmo, la motivación y la amabilidad con las que me invitaste a descubrir Valencia y algunos de sus maravillosos rincones. Por invitarme a participar de la ICIAM. Y por las discusiones matemáticas felizmente interminables que me dieron la confianza que a veces se me escabullía. *Bueno*, y gracias a ti ahora entiendo *WENO*.

Merci Simon et Julien pour faire partie de mon comité de thèse. A Luis pour ta préoccupation pour l'environnement et ta bonne humeur. Yvon pour des projets super intéressants et variés. Emmanuel pour diriger le labo. A Catherine, Malika, Salima, Erika, Corentin pour résoudre, m'aider ou m'orienter avec toutes les procédures administratives. Kash et Hugues pour résoudre mes problèmes informatiques. Nora pour l'amabilité et la gestion de la bourse de l'ISCD. Corentin et Jean-François pour résoudre mes problèmes avec l'école doctorale.

University, universality, bridges and family

I wouldn't be here if it wasn't for the free, public and quality education received back in Argentina which is only possible by the combined effort of a nation in the hope that its children would have better chances and live in a future free from the chains of unjust life-long debt and slavery. For similar reasons, I thank Alexandra Elbakyan (my secret hero) for fighting for a world with knowledge and science equally accessible for everyone.

I wouldn't be here neither if it weren't for Dominique, aujourd'hui je peux le dire en français, merci pour voir, il y a 5 ans maintenant, quelque chose en moi pour penser à m'offrir un lien et construir le pont qui m'a amené jusqu'à ici. Je ne serai pas le moi que je suis aujourd'hui, je serai un autre mais pas celui ci. Et aussi, merci Matt, pour la bienveillance, y junto a Leo por permitirme hacer mis primeros pasos de trabajo e investigación en Aristas con tantos proyectos interesantes y variados. Y así mismo, aquellos consejos que me dieran una tarde sobre cómo pensar y afrontar un doctorado o, en fin, cualquier tipo de proyecto.

Claro está que sin mi mamá ni mi papá o mi ¡Emanitooo! nada de esto sería posible. Desde la fascinación por la ciencia y la naturaleza alimentada mirando estrellas, caminando Andes, y fogoneando el espíritu reflexivo con preguntas, problemas y experimentos que seguro tenían que incluir ondas en algún momento. El cariño, la lectura de cuentos, la libertad de explorar nuestra existencia aunque eso nos llevara lejos de casa. La posibilidad de poder hablar y contar lo que nos pasa y confiar que buscaremos soluciones en vez de culpables. En fin, un conjunto innumerable de gestos y sensaciones que puedo llamar hogar, por ser cálido como el fuego y sagrado como el de los antiguos Lares.

La buela que vuela vuela por mates, pastas, y tardes de bella Pampa. A Celia y los asados sancochados. A Silvio por maravillarme de chico con tus viajes a la Antártida (¿Sabías que antes quise ser paleontólogo también?). A Juli y Chachi por charlas, trucos, siempre traernos y llevarnos. A mis primas, Cami, Pau y Coti por interminables juegos y maravillosas vacaciones a la sombra de los Caldenes.

Paris

Georges merci de m'enseigner le français.

Muchachada, gracias por hacer de los tumultuosos años del CoVid y de la llegada a París una experiencia fantástica. Ana Guevara, la incansable capitana, siempre lista a todo. Gracias por las caminatas por el Sena y por estar presente a la escucha en los momentos álgidos. Emilio, gran inventor del café de las 5 junto con su contracara, la meditación sigilosa. Ramon, fiel al crous tardío, que ya extrañamos con brío, siempre listo para la Barge, un football, un tenis o una amigable caminata. Suney, guerrera cual Asuna en SAO, admiro tu valentía (y tu inigualable cocina). Claudia, las discusiones políticas y filosóficas nos han llevado siempre a los límites de lo explorable, espero que podamos continuarlas en el futuro y mejor si es en torno a alguno de tu arsenal de juegos! Paula, por caminatas en París o Lyon. Jesus, por el cariño que siempre irradias, la música que siempre llevas. A Nicolás por las frases que aun en constante renovación siempre suman 42. Y esa pasión por las lenguas que, sin embargo, se muestra bajo la forma de una.

Vamos piratas! Giorgia e Noemi, grazie por tanta bella pazzia nella teuf e nel Portogallo. Emma e Chiara per il bello spirito.

Violeta por hacerme descubrir museos, secretos recovecos de París y convertirte en mi Invaders nemesis. Gaston y Mona por la familiaridad, la sencillez y la fuerza que derraman y con la que afrontan la indomable existencia. Nicolas K, por pasarnos un contacto clave.

Elena grazie per la tua presenza piena di luce. Maria, madame de las madames, queen de las queenes, gracias por estar. Estar cuando todo parecía irreal, por comprender y

ayudarme a comprender. Y también por darme bella literatura para leer. Nilo, siempre en mil proyectos y aun así presente ante cualquier dificultad. Gracias por tu bella amistad y por abrir las puertas de tu casa. Cristobal, siempre listo para un matecito y conversaciones que me hacen sentir en casa, en la querida Latinoamérica. Apolline, merci pour le rire, pour les secrets de Paris et ses alentours, pour venir et partager avec nous.

Roberta, Maria et Adrian, pour le projet CEMRACS mais encore plus pour des sorties, luttes de végétaux, une longue balade à vélo qui m'a presque tué avec les diaboliques pedales automatiques. Siwar on a réussi! Et maintenant on doit fêter! Gaspard, pour m'enseigner ce jeux de cartes. Laurent pour tant d'histoires, de blagues et de belles musiques. Giulia per tante risate e noi convidare la bella musica nel CEMRACS. Pierre M. pour m'inviter à parler à Reims, on a encore un voyage à vélo à faire! Mi-Song, Matthias pour une belle balade à Cassis et le temps partagé au CEMRACS. Avec Etienne, Ludovica and Beatrice, et also Haibo, Mateo and Elham. Y Sara siempre atenta a todo. Louis-Pierre et une longue longue balade.

Antoine, merci pour les merveilleuses rencontres dans Le Salon remplies d' amitié, musique, fromage, pendus... Pour prendre toutes les responsabilités au labo y siii es a la izquierda. Alexiane et Pauline pour de belles et longues discussions au bureau. Sylvain, tes apparitions surprises et amicales. Jana, les lundis Mamba et les longues discussions. Andrea, corta pero bella presenza. JG pour la bonne humeur et le sarcasme. Marcel, Lucia, Nicola, Alessandro, Archit, now the office will be all yours! Et maintenant le dernier magnifique ajoute au Team Bureau: Nicolai! Le merveilleux conseiller de la Norvège, l'apprentie de magie noir des GPU bientôt le pouvoir sera a toi. Zheng Ping for the long philosophical discussions that so easily reach every corner of the world.

Matthieu merci de m'expliquer tant de choses que je ne comprenais pas, sans jugement et toujours avec un sourire passionné et contagieux pour les mathématiques, les aventures soit dans la nature, les montagnes ou le vélo. Merci Yvonne de me faire découvrir la comédie française et tant de discussions amiables au labo. Et merci de partager avec moi la découverte de la Corée et du Japon!

Jules G. pour organiser les journées 1A et maintenir le livret d'accueil. Anais pour l'ambiance chaleureuse et Anatole pour la magie, le trantranzai, et codiriger le thé du labo, cierto?. Rui to take on the lab tea as well as for the multiple advice and sharing the CS conference, and sorry for the interminable walk, upsi. Now is Ruikang and Eleanor's time. By the way, thanks Eleanor for the positive vibe.

Thomas, pour le sourire, certain sticky song that starts like this: "What do we do... mathemaaatics". Et merci de me montrer une certaine porte en bois. Robin, toujours malin, pour nous bien représenter. Jules P. pour insister infatigablement, certaines fois avec raison, sur tout sujet, toujours dans le bon esprit de maintenir allumé la flamme du labo. And now the task is continued by the new marvelous team: Aleksandra, Aloïs, Federica & Siguang.

Charles pour certaines chansons recueillis dans un papier. Pierre, tes blagues et sarcasmes avec références à l'intérieur de la culture française de 5 niveaux de profondeur. Lucas P. pour m'amener a la découverte de Manim (et sa malédiction). Ludovic et Zhe Chen, et Guillaume avant eux pour faire du GTT une réalité. Et Ioanna, pour les infomaths.

Roxane pour la conscience de lutte et nous rappeler des réunions pour les doctorants

d'autoformation pour réfléchir à comment faire une science plus proche de la société. Annamaria per una giornata de scienze e educazione.

Allen, discusiones sobre locos dibujos que se vuelven notaciones. Liang Ying your incomparable knowledge on paintings and french culture. Ming Yue always with a smile. Lucas J. pour m'enseigner à dire *Enti halue*. Lise pour me poser une question clé en espagnol. Rémi pour le pouvoir de einsum. Gong, sorry for sometimes over-using the server. Juliette pour le comité environnement. Alexandre et une passion contagieuse pour le vélo. Eugenio sempre tranquillo. David, pour être, par chance, l'espion a l'enceinte de l'administration. Et aussi Valentin, Toai, Nga, Assane, Yi Peng, Lucas E, Gabriela, ...

Argentina

El Tai Chi, y todo lo que ello implica, de la infatigable mano de Carlota que sigue reuniéndonos virtual o presencialmente con la esperanza de que podamos descubrirnos un poco más a nosotros mismos. Sin este pilar yo sería definitivamente otro, probablemente mucho más perdido. Y también a Walter, Ceci, Juan, Yara, Luz, Gisela, Jorge, Diana, Esteban, Maggi, Mel, Matias, Gonzalo. ¡André, por recibirme también en Suiza! Andrey, siempre con una buena mirada de la tierra. Y luego en España a la banda de los "Más o menos Ma Tsun Kuenos".

Cynthia por impulsarme a la física.

A Fulgura por sus locuras y esa felicidad que solo se encuentra al cantar.

A la familia Napoli por estar presente cuando yo estaba lejos de la mía.

Enrico aventurero sin igual, ya van más de 5 ciudades en las que te he visitado, ¿Dónde te encontraré en la próxima? Flor, la pasión por las montañas y la posibilidad de un mundo mejor. Fede por tener siempre una Hackathon bajo el brazo. Lauri por tantos años de amistad. Flavia por tantas bellas discusiones filosóficas y una amistad que perdura con el paso de los años. Silvia, por brindarme tus consejos en momentos clave y así permitirme entender, actuar y resolver.

Tomy, el famoso crepe bucloso cuyo retorno aguarda cuando vengas a París. Gracias por las discusiones que perduran a la distancia siempre renovadas en las experiencias. Camilo y tus maravillosos encuentros filosóficos, mucho he aprendido este año y siempre que nos hemos encontrado con vos y con Nico C. con el pretexto de una película o por hablar de Boca.

Los "Wichis" hoy desperdigados por el mundo, haciendo amalgama cada año en renovados encuentros. Dani, conversaciones que siempre aplazamos pero que cuando ocurren nos reencuentran felizmente; la más viajera. ¡Hasta Nueva Zelanda tenemos que ir a visitarte ahora! Pili y los adorables mellis, gracias por tanto cariño. Tomas, buenos y calmos consejos, referencias memeticas de orden 5 o más, siempre bello el reencuentro. Vicky ya un día volveremos a levantar un foque, evitar la botavara y poner rumbo en ceñida compartiendo la pasión por los veleros. Nacho tu sarcasmo y sentido del humor siempre hacen reír. Nico P, por tus imitaciones e historias sin igual. Sofi siempre sonriente me devuelves la esperanza en la gente. Belu, acroyogista y luchadora, por una química en armonía con el ambiente.

Gonzalo, un crack, en mil cosas y siempre encontrando la risa; que decir más que tuki. Ara, por transmitirme esa pasión por las estrellas, por una didáctica sin par. Por hacer posible que pudiéramos ver un eclipse de sol; por estar siempre lista a ayudar. Yami, corazón libre, siempre con un nuevo viaje en la manga, ya en bici o en canoa, agradezco que aunque pasa el tiempo y estamos lejos la amistad perdura aguardando siempre el cálido reencuentro.

El "Jamón es una lechuga" tampoco podía faltar con sus variadas discusiones y amistades que ya se acercan a las dos décadas. Mati, por los libros que no te devolví. Eri, aunque lejos, siempre dispuesta a ayudar. Juan, por no olvidar "Zamba para olvidar" y por ¡Viva el Rey León ... de Francia!. Wan por dar origen a tan desconcertante nombre. Fabi, por tantas discusiones a la distancia sobre humanidades digitales, historia, NLPs, epistemología y más; siempre me mantienen entusiasmado y pensando. Mañu, viejo, antiguo, rupestre amigo a la antigua, siempre con toda esa energía contagiosa.

Facu, viajero infatigable, apasionado por la ciencia y el mundo, me alegra que nuestros caminos se hayan podido cruzar más a menudo de lo que 9 horas de diferencia horaria preveían. Caminatas que nos llevaron de Machu Picchu a París y Noruega y ya veremos que le seguirá. En fin, gracias por la amistad.

Nico T., maestro Jedi, ya encontraremos el camino hacia la nueva ciencia.

Finalmente hemos llegado
al corazón de mi ciencia,
la última reverencia.
Búscala en la ausencia
ilógica en apariencia
mas lógica por prudencia.
Lunai, Rosai,
para vos, la existencia.

Contents

Abstract v

Contents xv

1 Introduction 1
- 1.1 Linear reduced modelling for forward simulation and inverse problems . . . 1
- 1.2 Towards non linear model reduction, motivation, objective and outline of this thesis . . . 9
- 1.3 Linear reduced order modelling for high contrast diffusivity . . . 10
  - 1.3.1 Compactness and convergence . . . 11
  - 1.3.2 Forward modelling . . . 13
  - 1.3.3 Inverse problem . . . 14
- 1.4 Non-linear approximation spaces . . . 15
- 1.5 High order non-linear interface reconstruction strategies . . . 19
- 1.6 Non-linear reduced basis . . . 22
- 1.7 Physics Informed Neural Networks for singularly perturbed convection-diffusion equations . . . 23
- 1.8 Modelling pollution at a city scale . . . 25
- 1.9 Python package for reproducible research . . . 27

2 Reduced order modelling for elliptic problems with high contrast diffusion coefficients 29
- 2.1 Introduction . . . 29
  - 2.1.1 Reduced models for parametrized PDEs . . . 29
  - 2.1.2 Parametrized elliptic PDEs . . . 31
  - 2.1.3 High constrast problems . . . 32
  - 2.1.4 Outline . . . 33
- 2.2 Uniform approximation in relative error . . . 34
  - 2.2.1 Limit solutions and the extended solution manifold . . . 34
  - 2.2.2 A compactness result . . . 36
- 2.3 Approximation rates . . . 39
  - 2.3.1 Polynomial approximation on inner rectangles . . . 41
  - 2.3.2 Polynomial approximation on infinite rectangles . . . 44

2.3.3 Approximation rates and n-widths . . . 48
2.4 Forward modelling and inverse problems . . . 51
2.4.1 Galerkin projection . . . 51
2.4.2 State and parameter estimation . . . 53
2.5 Numerical illustration . . . 56
2.5.1 Parameter selection . . . 58
2.5.2 Influence of dimensionality and geometry . . . 60

3 Non-linear approximation spaces for inverse problems 63
3.1 Introduction . . . 63
3.1.1 The recovery problem . . . 63
3.1.2 State estimation with reduced models for parametrized PDE's . . . 64
3.1.3 The PBDW method . . . 65
3.1.4 Towards nonlinear approximation spaces . . . 66
3.1.5 Objective and outline . . . 67
3.2 Nonlinear reduction of inverse problems . . . 69
3.2.1 A general framework . . . 69
3.2.2 The best fit estimator . . . 70
3.3 Linear observations . . . 71
3.3.1 Optimal norms . . . 72
3.3.2 The generalized interpolation estimator . . . 73
3.4 Shape recovery from cell averages . . . 75
3.4.1 The shape recovery problem . . . 75
3.4.2 The failure of linear reconstruction methods . . . 76
3.5 Shape recovery by nonlinear least-squares . . . 79
3.5.1 Nonlinear reconstruction on a stencil . . . 79
3.5.2 Global nonlinear reconstruction . . . 82
3.5.3 Numerical illustration . . . 83
3.6 Relation to compressed sensing . . . 84
3.6.1 Compressed sensing and best n-term approximation . . . 84
3.6.2 Stability and the null space property . . . 87
3.6.3 The case of ℓ^p norms . . . 88
3.A Proof of Proposition 3.5.1 . . . 88

4 High order recovery of geometric interfaces from cell-average data 97
4.1 Introduction . . . 97
4.1.1 Reconstruction from cell-averages . . . 97
4.1.2 Reconstruction of discontinuous interfaces . . . 100
4.1.3 Outline . . . 102
4.2 Numerical analysis of local recovery methods . . . 103
4.2.1 Local approximation by nonlinear families . . . 103
4.2.2 Near optimal recovery from cell averages . . . 108
4.3 Reconstruction by optimization (OBERA) . . . 110
4.3.1 Presentation of the method . . . 110

4.3.2 Analysis of the recovery error . . . 112
4.4 Reconstruction on oriented stencils (AEROS) . . . 115
4.4.1 Presentation of the method . . . 115
4.4.2 Analysis of the recovery error . . . 119
4.5 Numerical experiments . . . 120
4.5.1 Recovery of smooth domains . . . 120
4.5.2 The treatment of corner domains . . . 126
4.5.3 Finite volume evolution in time . . . 129
4.6 Conclusion and perspectives . . . 129
4.A The orientation test . . . 130
4.A.1 The case of a linear interface . . . 131
4.A.2 A perturbation analysis . . . 133

5 Nonlinear compressive reduced basis approximation for PDE's 135
5.1 Introduction . . . 135
5.2 Linear and nonlinear notions of m-widths . . . 137
5.3 Nonlinear compressive Reduced Basis approximation . . . 139
5.4 Analysis of a model framework : periodic step functions . . . 141
5.5 Numerical illustrations . . . 145

6 Deep learning-based schemes for singularly perturbed convection-diffusion problems 155
6.1 Introduction . . . 155
6.1.1 Scientific context and goals . . . 155
6.1.2 Contribution . . . 156
6.2 A singularly perturbed convection-diffusion equation . . . 157
6.2.1 Problem definition . . . 157
6.2.2 General formulation . . . 159
6.2.3 Vanilla (V) formulation . . . 159
6.2.4 Weak variational (W) formulation . . . 160
6.2.5 Rescaled formulation . . . 162
6.2.6 Summary of the methods . . . 163
6.3 Neural networks based numerical schemes . . . 164
6.3.1 General principle . . . 164
6.3.2 Neural Network classes of functions . . . 165
6.3.3 Sampling schemes . . . 166
6.3.4 Comparison with finite element schemes . . . 168
6.4 Numerical Results . . . 169
6.4.1 Test case and comparison criteria . . . 169
6.4.2 Our code and practical implementation details . . . 170
6.4.3 Discussion . . . 171
6.4.3.1 Impact of the number K of training points . . . 171
6.4.3.2 Impact of Machine Precision . . . 174
6.4.3.3 Impact of Sampling Strategy . . . 175

6.4.4 Conclusions from the numerical experiments 176
6.5 Future research directions and extensions 176
6.A l2 error plots 177

7 State estimation of urban air pollution with statistical, physical, and super-learning graph models 179
7.1 Introduction 179
7.1.1 Background and motivation 179
7.1.2 Urban air pollution modelling 180
7.1.3 Contributions and layout of the paper 181
7.2 Available data and pre-processing 182
7.2.1 Pollution sensors 182
7.2.2 Meteorological conditions 182
7.2.3 Traffic 183
7.2.4 Graph of Paris 184
7.2.5 Pre-processing of traffic data 186
7.2.6 Summary 187
7.3 Reconstruction methods 187
7.3.1 Spatial average 187
7.3.2 Best unbiased linear estimator 188
7.3.3 Kriging 189
7.3.4 Source model 190
7.3.5 Physical modelling 191
7.3.6 Super-Learning as a collaborative approach 195
7.4 Reconstruction benchmarks and Leave-One-Out 195
7.5 Numerical results 197
7.6 Conclusion and future works 199
7.A Metric graphs 200

Bibliography 203

Chapter 1

Introduction

When working with complex physical systems, we frequently need to have in hand a reliable description of their state to take fast but still sufficiently informed decisions. Situations of this type arise typically in engineering problems, for example, when assessing, via sensors, the state of a machine to decide if a replacement is needed or not, or while planing new infrastructures with optimized geometries and materials. Costly simulations are usually needed to attain the sometimes strict accuracy requirements. Therefore, the aim of this thesis is to develop new strategies for the approximation of system states faster than conventional solvers and with certified accuracy bounds when possible.

1.1 Linear reduced modelling for forward simulation and inverse problems

One way of framing these diverse situations in mathematical terms is by postulating that the actual state of the system can be described appropriately by an element $u \in V$ of some Banach space. Moreover, we usually have *a priori* information about the modelled system reflected formally as a set of conditions or restrictions that u has to satisfy. These conditions can be made explicit by specifying the membership of u to a subset $\mathcal{K}$ of V, $u \in \mathcal{K} \subset V$, for example, by adding a regularity assumption on u. Alternatively, when modelling physical systems, we usually ask u to be the solution of a certain parametric PDE (Partial Differential Equation)

$$\mathcal{P}(u, y) = 0,$$

where $\mathcal{P}$ is a differential operator, $y \in Y \subset \mathbb{R}^d$ is the set of parameters defining $u = u(\cdot, y)$ through $\mathcal{P}$. The parameters y account for physical quantities relevant to the modelling, for example: the thermal diffusivity, electric conductivity, boundary or initial conditions, the geometry of the domain, etc. Notice also that $u(\cdot, y)$, as it is the solution of a PDE, it is actually a function taking values in a space or space-time domain $u(\cdot, y) : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$.

As guiding examples of what $\mathcal{P}$ can be, consider first the linear elliptic equation (central

for Chapter 2)

$$\begin{aligned} -\mathrm{div}(a\nabla u) - f = 0 \quad &\text{on} \quad \Omega, \\ u = 0 \quad &\text{on} \quad \partial\Omega, \end{aligned} \tag{1.1}$$

with the source term $f \in H^{-1}(\Omega)$ and

$$a = a(x, y) = \bar{a}(x) + \sum_{j=1}^{d} y_j \psi_j(x) \tag{1.2}$$

with $\bar{a}$ and ψ_j both in $L^\infty(\Omega)$. In this setting, $u(\cdot, y) \in H_0^1(\Omega)$, and consequently by specifying the vector y we are defining u through $\mathcal{P}$ and y.

As a second example, let us take the transport equation (relevant for Chapters 4 to 6)

$$\frac{\partial u}{\partial t} - a \cdot \nabla u = 0 \quad \text{on} \quad \Omega, \tag{1.3}$$

with periodic boundary conditions and $a = y = (y_1, \dots y_d)$ the velocity of the transport which, in this simplified case, will not vary neither in time nor space.

In general, the mapping, $y \mapsto u(\cdot, y)$, defines the parameterized manifold

$$\mathcal{M} = \{u(\cdot, y) : y \in Y\}$$

which, in the context of PDEs, is usually called the *solution manifold*.

In applications, however, the actual u is out of reach by the finite numerical precision of computers, which means that in practice we are forced to work with computable approximations $u^N(\cdot, y) \in V_N$ with $\dim(V_N) = N$ through a proper choice of space-time discretization and numerical solvers: Finite Element Method (FEM), Finite Volumes (FV), etc. Yet, computing u^N can become infeasible if, under tight time constraints, the requirements on accuracy are high or we need to find the solution for multiple queries of the parameters y.

The objective will then be to build spaces $V_n \subset V$ (not necessarily linear), parameterized by a small number $n \ll N$ of parameters, that approximate well the manifold $\mathcal{M}$. This should allow us to efficiently deal with the two following main problems:

- **Forward modelling**: we are asked to compute fast and accurately, the parameter-to-solution map $y \mapsto u(\cdot, y)$ for one or many parameter values $y^1, \dots, y^K$. Examples of this are found in shape optimization (*i.e.*, the optimal shape of a motor, a pipe or an air-plane wing) where one needs to solve a PDE multiple times, one for each possible parametrization of the geometry.

- **Inverse problem**: we are given $m \geqslant n$ possibly noisy measurements $z = (z_1, \dots, z_m)$ of the system, for example, pollution values at certain space locations. In this case, z is given by:

  $$z_i = \ell_i(u) + \eta_i,$$

  where η_i is the measurement noise associated to sensor i and ℓ_i is a function modelling

the measurement operation of the unknown underlying state u. In particular, if the sensor providing the measurement has a linear response with respect to the quantity represented by u, which is usually the case, and V is a Hilbert space then $\ell \in V'$ can be represented as a linear functional in V' the dual of V. Examples of such type of measurements are

- Point-wise evaluation $\ell_i(u(\cdot, y)) = u(x_i, y)$ taken in the point x_i inside the domain Ω where u is defined. Such ℓ_i is admissible when $V \subseteq \mathcal{C}(\Omega)$.
- Cell average $\ell_i(u(\cdot, y)) = \frac{1}{T}\int_T u(x, y)dx$ on an interval $T \subset \Omega$.

In this context, if one wants to know the state u of the system given the measurements z, we say it is a *state estimation* problem. If, instead, one is interested in inferring the parameters y whose associated $u(\cdot, y)$ best explains the observed measurements, we say it is a *parameter estimation* problem.

Two main questions arise in the context of building the spaces V_n to approximate the elements of the manifold $\mathcal{M}$: how to build "good" approximation spaces V_n allowing us to quickly find accurate approximations of u given y or z and how to assess the approximation capabilities of these spaces.

To answer the second question we can look at the distance between $\mathcal{M}$ and V_n defined by

$$\operatorname{dist}(\mathcal{M}, V_n)_V := \sup_{u\in\mathcal{M}} \inf_{v\in V_n} \|u - v\|_V$$

which quantifies the worst case scenario of approximating an element $u \in \mathcal{M}$ by the nearest element $v \in V_n$ (see Figure 1.1). In a setting where the parameters y are random variables sampled from a distribution ρ with support on Y and V is a Hilbert space, one may be interested instead on measuring the average error

$$\mathrm{e}_2^2(\mathcal{M}, V_n) = \mathbb{E}_{y\sim\rho}\left(\inf_{v\in V_n} \|u(\cdot, y) - v\|_V^2\right).$$

These notions do not tell us in practice how to actually find v from the knowledge of y or z. These do not inform on how to build good V_n spaces either. However, they provide us with a way of analyzing theoretically the approximation capabilities of a given space V_n. Furthermore, if we restrict our search of V_n within some family $\mathcal{G}_n$ of *n-parametric* spaces of V with some specified property, we can actually minimize $\operatorname{dist}(\mathcal{M}, V_n)$ on the choice of $V_n \subset \mathcal{G}_n$

$$\delta_n(\mathcal{M}, \mathcal{G}_n)_V = \inf_{V_n\subset\mathcal{G}_n} \operatorname{dist}(\mathcal{M}, V_n)_V$$

to get bounds on the approximation error. Note here that $\mathcal{G}_n$ can be, for example, the family of linear sub-spaces of V of dimension n. In this case we get the known *Kolmogorov n-width* (see Figure 1.1)

$$d_n(\mathcal{M})_V = \inf_{V_n \text{ linear}} \operatorname{dist}(\mathcal{M}, V_n)_V.$$

If we look instead at the problem in the randomized setting, then we have that every element of $u \in \mathcal{M}$ can be written as an infinite sum $u = \mathbb{E}(\mathcal{M}) + \sum_{i=1}^{\infty} c_i e_i$ with e_i the *Karhunen–Loève* orthonormal basis obtained from the spectral analysis of the covariance operator of the stochastic process. It is known that

$$\kappa_n^2(\mathcal{M}) = \inf_{V_n \text{ linear}} \mathrm{e}_2^2(\mathcal{M}, V_n) = \sum_{i=n+1}^{\infty} \lambda_i^2 \tag{1.4}$$

where the optimal linear space V_n^* is achieved by choosing the first n components of the *Karhunen–Loève* basis. The eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant 0$ are associated with each eigenvector e_i. The explained variance of each component e_i is quantified by λ_i^2. This decomposition is closely related to the century old concept of *Principal Component Analysis* (PCA). Finally, note that $\kappa_n \leqslant d_n$ as d_n is measuring the worst case scenario while κ_n measures the average one.

Returning to the definition of $\delta_n(\mathcal{M}, \mathcal{G}_n)_V$, let us note that we need to impose some relevant restrictions on what V_n can be via $\mathcal{G}_n$. Otherwise one may get degenerate situations where a single parameter space $V_{n=1}$ can be built such that it fills the entire manifold $\mathcal{M}$. Although it may achieve almost perfect approximation results, the notion of proximity $\|u - u'\|_V < \varepsilon$ in V is lost once we look at the distances of the approximations inside V_n as $\|P_{V_n} u - P_{V_n} u'\|_{V_n} \gg \varepsilon$. Here we take

$$P_{V_n} u \in \arg\min_{v \in V_n} \|u - v\|$$

to be the projection of u onto the manifold V_n and $\|\cdot\|_{V_n}$ the distance inside the n-dimensional manifold determined by V_n. For more details and related definitions see [66] where *non-linear n-widths* where first defined, a very similar notion to $\delta_n(\mathcal{M}, \mathcal{G}_n)_V$.

The analysis of $\delta_n(\mathcal{M}, \mathcal{G}_n)_V$ gives us a way to ponder in advance if a particular family of spaces $\mathcal{G}_n$ is suitable for approximating $\mathcal{M}$ or if it is doomed to fail no matter how well we decide to optimize our choice of V_n inside the class. This allows us to rule out beforehand some families $\mathcal{G}_n$ and divert our efforts to more promising ones. In this context we can separate the overarching process of research into four main instances:

(1) Theoretical approximation analysis of optimal spaces:

- for a given problem, defined through properties on $\mathcal{P}$, for example, by deciding to work on a specific PDE as (1.1) and (1.3),
- and a family $\mathcal{G}_n$ of *n-parameterized* spaces, for example, n-dimensional linear sub-spaces,

the objective is to find bounds on $\delta_n(\mathcal{M}, \mathcal{G}_n)_V$ to quantify the rate at which it will decrease when increasing the number n of parameters. For example, in [52] it is shown that, if $\mathcal{P}$ defines a parametric elliptic PDE, as the one of our first example (1.1), and $\mathcal{G}_n$ is the family of n-dimensional linear sub-spaces, then the *Kolmogorov n-width* has an exponential decay

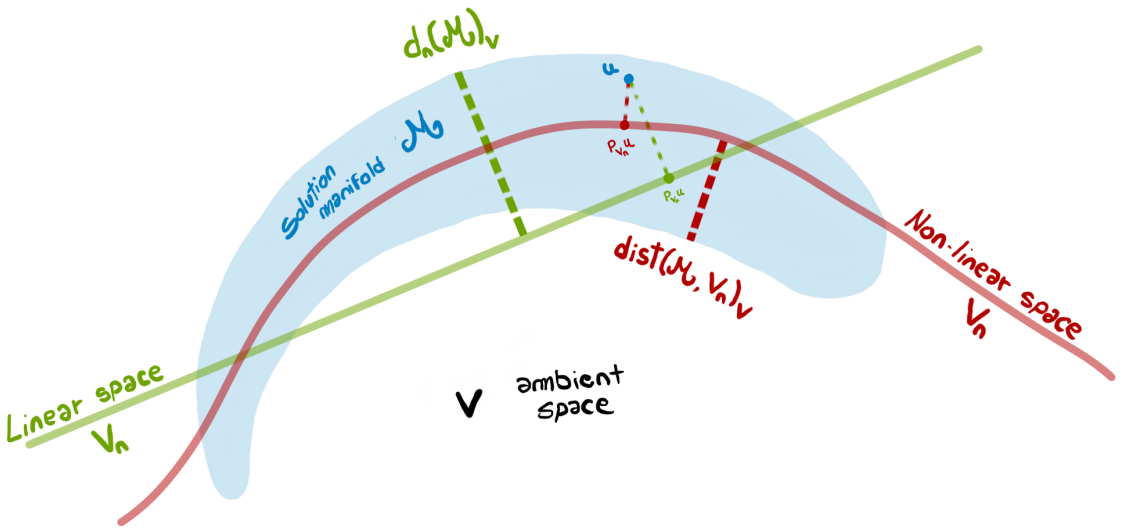


Figure 1.1: Distance between space V_n and manifold $\mathcal{M}$. The optimal linear space V_n gives the Kolmogorov n-width $d_n(\mathcal{M})_V$.

rate

$$d_n(\mathcal{M})_V \lesssim e^{-\beta n},$$

meaning that linear spaces are suitable for the approximation of elliptic PDEs as one can exponentially reduce the approximation error. In other words, for a prescribed accuracy requirement a small number of basis elements will suffice to attain it.

(2) *Offline* stage (or sub-optimal spaces construction): theoretical guarantees for optimal spaces tell us how good or bad the best choice of space $V_n \subset \mathcal{G}_n$ approximating $\mathcal{M}$ can be. Even though it can serve to discard couples of $(\mathcal{M}, \mathcal{G}_n)$, in practice the optimal space still remains unreachable. Consequently, we are forced to rely on sub-optimal spaces built under some heuristic.

In this stage there is no time constraints yet so one can use an expensive solver to generate many solutions $u_i = u(\cdot, y^i)$ and build the sample set $\mathcal{M}^K = \{u_1, \dots, u_K\}$ with $K \geqslant n$. This is done by sampling the parameter space $y^i \in Y$ in regions relevant to the problem. Then, one can build the space V_n, for example, by replacing $\text{dist}(\mathcal{M}, V_n)$, the quantity one would like to optimize in theory, by a discrete approximation of it

$$\text{dist}(\mathcal{M}^K, V_n)_V = \max_{1 \leqslant i \leqslant K} \|u_i - P_{V_n} u_i\|.$$

If we restrict ourselves to the case of linear sub-spaces we can construct V_n as a subspace of $V_K = \text{span}\{u_1, \dots, u_K\}$ in various ways (see also Figure 1.2 for a schematic representation of the methods):

- **Random**: picking randomly n elements from $\mathcal{M}^K$.

- **Greedy**: adding iteratively the element $u_{n+1} \in \mathcal{M}^K$ at farthest distance from the subspace built so far:

$$V_n = \operatorname{span}\{u_1, \dots, u_n\}$$
$$u_{n+1} = \arg\max_{1 \leqslant i \leqslant K} \|P_{V_n}^{\perp} u_i\|,$$

  taking $P_{V_n}^{\perp} u = u - P_{V_n} u$.

- **Principal components analysis (PCA)**: it can be seen also as a greedy algorithm where instead of searching for the element $u_{n+1} \in \mathcal{M}^K$ that maximizes the distance to V_n, one looks for the element $v \in V$ such that

$$u_{n+1} = \arg\max_{\substack{v \in V \\ \|v\|=1}} \sum_{1 \leqslant i \leqslant K} \|P_v P_{V_n}^{\perp} u_i\|_V^2.$$

  Contrary to the previous greedy method, this procedure can be done directly by *singular value decomposition* of the covariance matrix of the data which is the discrete version of the covariance operator used to obtain the *Karhunen–Loève* basis mentioned before.

(3) Approximation guarantees of sub-optimal spaces: the construction of sub-optimal spaces raises the question about their approximation properties. More specifically, given an algorithm $\mathcal{A}$ to build a sub-optimal V_n space, like the ones presented above, one would be satisfied if $\operatorname{dist}_{\mathcal{A}}(\mathcal{M}, V_n)_V$ also has similar rates to the optimal one. In the case of $\mathcal{G}_n$ consisting of linear spaces and $\mathcal{M}$ the class of elliptic PDEs, it was shown in [52] that the subspaces generated by the greedy algorithm still share the same convergence rates as $d_n(\mathcal{M})_V$.

(4) The ***Online* stage** corresponds to the practical application step whether it is a forward modelling or an inverse problem. As we are now tied to short-time restrictions, we need to have a fast way of finding an element $\tilde{u} \in V_n$ that is a good approximation of u. This is done through some reconstruction strategy R yielding approximations $\tilde{u}$ which ideally should be close to the projection $P_{V_n} u$. Let us recall that $P_{V_n} u$ is out of reach as we would need to know u in the first place.

In general, the reconstruction strategy R will depend on the problem of interest:

- For *forward modelling* we want to build a mapping $R : Y \to V$ such that $\tilde{u} := R(y)$,

- For *inverse problems* $R : \mathbb{R}^m \to V$ such that $\tilde{u} := R(z)$ recalling that $z = \ell(u)$ are the observations.

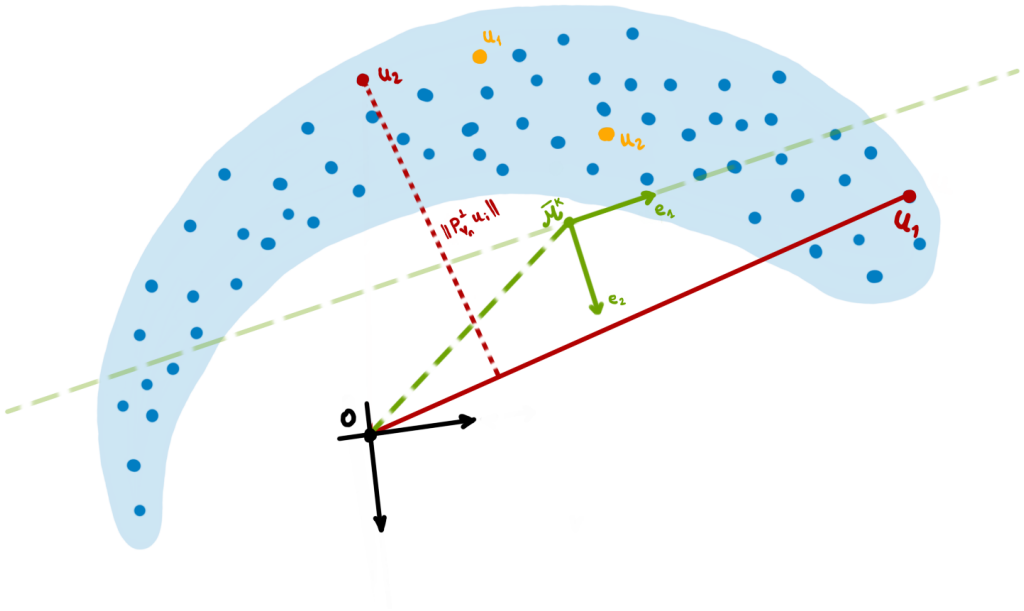


Figure 1.2: Scheme of the different criteria to build a reduced basis. The scattered blue dots represent the sample set $\mathcal{M}^K$. The red points $u_1, u_2 \in \mathcal{M}^K$ are the first two elements that one would obtain by following a greedy selection strategy. The green vectors $e_1, e_2 \in V$ represent the principal components with $\overline{\mathcal{M}^K}$ the center of mass of the discrete data $\mathcal{M}^K$. The orange points u_1 and u_2 represent a possible realization of a random selection strategy.

Asking the reconstructions $\tilde{u}$, obtained through R, to be close to the best possible approximant $P_{V_n}u$ in V_n is equivalent to requiring that R satisfies the *near optimality property*

$$\|u - \tilde{u}\| \leqslant C\|u - P_{V_n}u\|,$$

which is further analysed in Chapter 3 in the context of non-linear approximation.

In the case of linear spaces, as V_n has already been chosen, one can directly compute the projection onto the reduced sub-space by solving the corresponding linear systems. Writing $\tilde{u} = \sum_i^n c_i u_i$ with $\{u_1, \dots, u_n\}$ a basis of V_n built using, for example, some of the aforementioned strategies we can perform a

- **Galerkin projection** in the case of *forward modelling*: $P^y_{V_n}$. In the case of (1.1), relying on the usual variational formulation, we can write

$$\sum_i^n c_i \int_\Omega a(x,y)\nabla v_i(x)\nabla u_k(x)dx = \int_\Omega f u_k(x)dx, \quad 1 \leqslant k \leqslant n \tag{1.5}$$

  and obtain a reduced linear system of size $n \times n$. This is in contrast to solving the equation with a classical discretization method which would involve a space V^N with $N \gg n$ degrees of freedom.

- **Least squares minimization** in the case of *inverse state estimation*:

$$\min_{v \in V_n} \|z - \ell(v)\|^2 = \min_{c \in \mathbb{R}^n} \|z - \sum_i^n \ell(c_i u_i)\|_{\ell^2}^2 \tag{1.6}$$

  leading to an $m \times n$ linear system.

  Alternatively, in the context of noiseless measurements z and if V is a Hilbert space, one can write the minimisation problem directly in the ambient space V

$$\min_{c \in \mathbb{R}^n} \|w - \sum_i^n c_i u_i\|_V$$

  where $w = P_W u$ is the projection of u into the measurement space $W = \text{span}\{w_1, \dots, w_m\}$ which is spanned by the Riesz representers w_j of the linear functionals, $\ell_j(u) = \langle w_j, u \rangle$.

  In this last scenario, one would expect $\tilde{u}$ to exactly recover the observations. However, it will not be usually the case as the underlying physical model $\mathcal{P}$ is a simplification of the real world. In this case we can apply a correction so that $u^* = \tilde{u} + (w - P_w \tilde{u})$. This process of correction is known by the name of *parameterized background data weak* (PBDW) [108] and we will delve more into it and its non-linear extensions in Chapter 3.

There is today a good understanding of both the potential and the limitations of using linear reduced spaces as an approximation class $\mathcal{G}_n$. In particular, it is known [14, 52, 54] that the *Kolmogorov n-width* has an algebraic convergence rate n^{-s} with $s = \frac{1}{p} - 1$ regardless of the dimension d if the solution map $y \mapsto u(\cdot, y)$ has the following properties:

- Presents **anisotropy**, meaning that there is a hierarchy of decreasing importance on the variables y_j. That is, there exists an affine representation of y such that

$$y := \overline{y} + \sum_{j=1}^{d} a_j \psi_j \tag{1.7}$$

  with d possibly ∞, $a_j \in [-1, 1]$, $\overline{y}$ and ψ_j in $L^\infty(Y)$ and $(\|\psi_j\|)$ is a sequence in $\ell^p(\mathbb{N})$ when d is infinite.

- It can be **holomorphically extended** around Y.

Note also that these conditions are not restricted only to PDE solution maps.

The value of p expresses the strength of the anisotropy of the solution map and showcases the decaying importance of elements ψ_j used to parameterize the map. The finite dimensional case is anisotropic by nature as the sequence $(\|\psi_j\|)$ is finite. (1.2) is an example as it is of the same form as (1.7). Another situation where the anisotropy is observed, occurs in the randomized setting where we can associate $\psi_j = \lambda_j e_j$ with the *Karhunen-Loève* basis multiplied by its eigenvalues. Consequently, the sequence $(\|\psi_j\|)$ will be ℓ^p-summable if the

eigenvalues decrease fast enough evidencing the anisotropy in their decay rate.

1.2 Towards non linear model reduction, motivation, objective and outline of this thesis

In Chapter 2 we extend the approximation guarantees offered by linear spaces in the context of the linear elliptic Equation (1.1) when the parameter space Y is allowed to be unbounded, or equivalently, when extreme levels of contrast in the diffusivity constants are possible.

Chapters 3 to 6 present different approaches to tackle problems where linear spaces are doomed to fail. One of such situations occurs when the elements of the manifold $\mathcal{M}$ we wish to approximate are functions presenting jump discontinuities, for example, if $u = \chi_{\Omega(y)}(x)$ is the characteristic function defined on a parameterized domain $\Omega(y)$. To have an idea of the limitations faced by linear spaces when used to approximate such functions, take the Fourier basis $u = \sum_{j=1}^{\infty} c_j e_j$ and study the mean-square error when approximating u by its truncated series (which is the best linear model in L^2 sense according to (1.4)). In 1d, the coefficients c_n of the Fourier series of an indicator function are proportional to $1/n$ and consequently, $|c_n|^2 \propto n^{-2}$ and $\kappa_n^2 \propto n^{-1}$ as it is obtained after summation of the first n terms. This yields the following lower bound for the *Kolmogorov n-width* $d_n \geqslant \kappa_n \gtrsim n^{-1/2}$ which showcases the limitation of linear spaces on this circumstances pushing research towards seeking non-linear methodologies which is the central thrive of this thesis.

In this context, Chapter 3 presents a theoretical framework to analyse the effectiveness of non-linear strategies, while Chapter 4 delves more into the practical implementation of high-order algorithms able to reconstruct discontinuities out of cell average data as one could find in images or finite volume discretization of PDEs. In Chapter 5 we show how it is possible to accelerate or render more accurate the reconstruction of $1d$ characteristic functions by learning long Fourier expansions ($N \gg 1$) from the knowledge of only the first $n \ll N$ Fourier coefficients. We do this with a machine learning technique. In Chapter 6 we turn the attention to another learning technique known as Physics Informed Neural Networks (PINNs) to tackle a linear transport-diffusion equation when the diffusivity vanishes and shocks appear.

Finally, in Chapter 7 we apply a combination of linear and non-linear methods to a real case scenario where the objective is to predict the pollution level on every point in a city using heterogeneous sources of data like temporal pollution series on specified locations, the geometry of the streets and Google Maps traffic information.

We next summarize the content of each chapter.

1.3 Linear reduced order modelling for high contrast diffusivity

In Chapter 2, based on [51], we center our analysis on the particular case of the diffusion equation presented in (1.1). To ensure existence and uniqueness of solutions through Lax-Milgram theory, the Uniform Ellipticity Assumption (UEA) is usually considered which imposes bounds on the possible values that the diffusivity can take

$$r \leqslant a(x,y) \leqslant R,$$

with $0 < r \leqslant R < \infty$. In this context, [15, 150] showed that

$$d_n(\mathcal{M})_V \lesssim \exp\left(-cn^{1/d}\right),$$

that is, the *Kolmogorov n-width* has a sub-exponential decaying rate.

More specifically, we work in the situation in which the modelled system consists of a material composed of multiple disjoint sub-domains Ω_j, with constant diffusivity $a(x,y) = y_j$, $x \in \Omega_j$ inside the region. We represent this scenario by specifying $\psi_j = \chi_{\Omega_j}$ which is the characteristic function on the sub-domain Ω_j with $\cup_{j=1}^{d}\Omega_j = \Omega$ (see Figure 1.3). Consequently, (1.2) becomes the piece-wise constant function

$$a(x,y) = \sum_{j=1}^{d} y_j \chi_{\Omega_j}(x)$$

with the solution $u(y)$ now satisfying the variational formulation

$$\sum_{j=1}^{d} y_j \int_{\Omega_j} \nabla u(y) \cdot \nabla v dx = \langle f, v \rangle_{H^{-1},H_0^1}. \tag{1.8}$$

The integral over the whole domain in the left hand side of Equation (1.8) can be decoupled into the sum over regions due to the disjoint partitioning of the domain and y_j can exit the integral as it is a constant for each subdomain.

Let us note that it is not possible to approximate uniformly well all elements of $\mathcal{M}$ due to the homogeneity property

$$u(ty) = t^{-1}u(y)$$

as it implies that $\lim_{y\to 0} \|u(y)\|_{H_0^1} = \infty$ and the same for $\operatorname{dist}(u(y), V_n) = \|u(y) - P_{V_n}u(y)\|_{H_0^1}$. This leads us to work with

$$Y' = [1,\infty]^d$$

and analyse the approximation error estimates in the relative sense, when working on

$$Y =]0,\infty]^d,$$

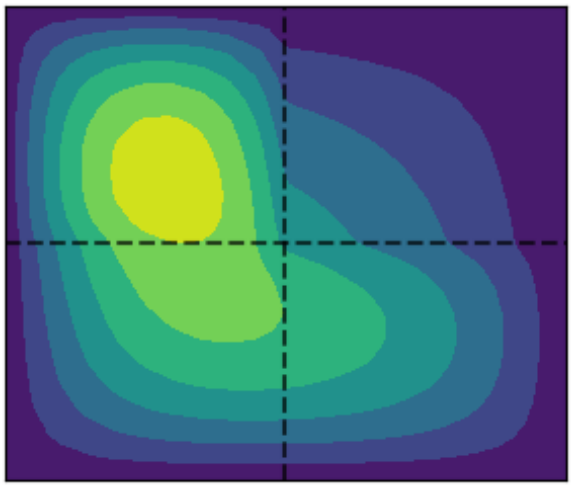


Figure 1.3: Example of a solution to Equation (1.8) for a domain composed of $d = 4$ subdomains. For this example the diffusion coefficients associated to each one of the four subdomains are: $(y)_{ij} = \begin{pmatrix} 7 & 87 \\ 17 & 16 \end{pmatrix}$. One can observe that in the solution $u(\cdot, y)$ the mass accumulates in the upper left corner due to its low value of diffusivity.

$$\operatorname{dist}(u(y), V_n)_{H_0^1} \leqslant \varepsilon_n \|u(y)\|_{H_0^1}, \tag{1.9}$$

with $\varepsilon_n \to 0$ a zero-converging sequence indexed by n.

The main conclusion of this work is that linear spaces are still a "good" approximation class for the solution manifold of the diffusion equation with piece-wise constant diffusivity as they still retain sub-exponential convergence rates when UEA does not hold as arbitrary high contrast is allowed.

1.3.1 Compactness and convergence

To arrive there, several ingredients have to be concatenated. First we prove that $\mathcal{M}' := \{u(y) : y \in Y'\}$ is a compact set of $H_0^1(\Omega)$ which allows us later to establish the following convergence theorem:

Theorem 1.3.1. *There exists a sequence of linear spaces $(V_n)_{n>1}$ such that $\dim(V_n) = n$, and a sequence $(\varepsilon_n)_{n>1}$ that converges to zero such that*

$$\|u(y) - P_{V_n} u(y)\|_{H_0^1} \leqslant \varepsilon_n \|u(y)\|_{H_0^1}$$

for all $u \in Y'$, where P_{V_n} is the $H_0^1(\Omega)$-orthogonal projector onto V_n.

The key element to prove compactness is the introduction of limiting solutions $u_S \in V_S$ with

$$V_S := \{v \in H_0^1; \nabla v_{|\Omega_j} = 0; j \in S \subset \{1, 2, \ldots, d\}\}.$$

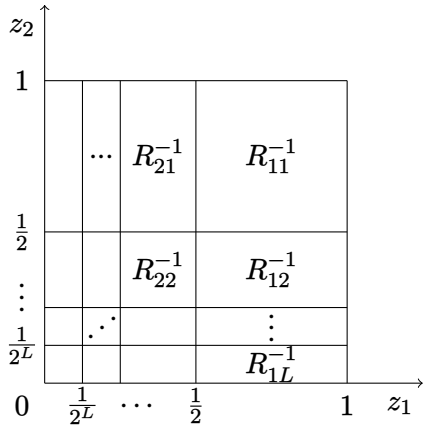


Figure 1.4: Partition of $[0,1]^d$ by the inverse rectangles R_b^{-1} in the case $d=2$. The axis variables $z=y^{-1}$ are the inverse of the parameter variables so that $y_i=\infty$ is represented by $z_i=0$.

In other words, elements of V_S are $H_0^1(\Omega)$ functions that are constant on some subdomains Ω_j given by the set S. Splitting $y=(y_{S^c},y_S)$ then u_S is the solution to the problem

$$\sum_{j\in S^c} y_j \int_{\Omega_j} \nabla u_S(y_{S^c})\cdot\nabla v\,dx = \langle f,v\rangle_{H^{-1},H_0^1} \quad v\in V_S. \tag{1.10}$$

The fact that functions u_S are the limit of $u(y)$ when $y_S\to\infty$ is proved in:

Lemma 1.3.2. *There exists a unique solution $u_S(y_{S^c})\in V_S$ to Equation (1.10), which is the limit in $H_0^1(\Omega)$ of the solution $u(y_S^c,y_S)$ as $y_j\to\infty$ for all $j\in S$.*

The convergence result of Theorem 1.3.1 does not give a quantification on the rate, for this we need to build reduced model spaces so that the relative error (1.9) retains a sub-exponential decay. For this, we adapt the strategy used in [13], under UEA, on each region of a dyadic partition of the parameter space. That is, we divide $Y'=[1,\infty]^d$ in rectangular domains such that for any $b=(b_1,\dots,b_d)\in\mathbb{N}_0^d$ we define the rectangle $R_b=[2^{b_1},2^{b_1+1}]\times\cdots\times[2^{b_d},2^{b_d+1}]$, see the scheme on Figure 1.4. Of course, we cannot infinitely divide the parameter space, consequently we join together all rectangles given a threshold L. That is, $[2^{b_j},2^{b_j+1}]$ is replaced by $[2^{b_j},\infty]$ if $b_j=L$.

The sub-exponential convergence obtained in [15, 150] relies on the three sufficient conditions, described in the introduction, which guarantee an algebraic or exponential rate of convergence for linear spaces. Here, the anisotropy is found in the fact that $d<\infty$ and the holomorphic extension is still retained for all "interior" rectangles R^I, that is, those whose bounds are finite in all directions. For those, we can use a polynomial approximation of the form

$$u(y)=\sum_{\nu\in\mathbb{N}^d} u_\nu y^\nu,$$

with $y\in R_b^I$. By holomorphy, the series converges to the target function $u(y)$ while the

approximation is done through best k-term truncation

$$u_{b,k}(y) = \sum_{|\nu| \leqslant k} u_{b,\nu} y^{\nu},$$

thus obtaining similar sub-exponential rates as in [15, 150] for each individual interior rectangle.

To find similar results on the infinite rectangles we need to introduce the additional geometrical assumption that all subdomains are disjoint inclusions. This is needed to define the trace operator on an epsilon extended domain.

The constants L (how much the parameter domain is partitioned) and k (the dimension of the reduced basis inside each rectangle) reach the optimal balance when the approximation error on interior rectangles matches the infinite ones.

This partitioning results in a family of local reduced model spaces $V_{b,k} = \text{span}\{u_{b,\nu} : |\nu| \leqslant k\}$ that can be used individually if we know the rectangle R_b where y belongs to, typically in *forward modelling* problems. Note that in this case the strategy is non-linear with the benchmark given by the notion of library width [148] instead of the *Kolmogorov n-width*. We can, otherwise, combine all the $V_{b,k}$ spaces to build a global reduced model space V_n. In this last scenario, we found that the sub-exponential rate is retained in the following form

$$d_n(\mathcal{M}')_{H_0^1} \leqslant C \exp(-cn^{\frac{1}{2d-2}})$$

1.3.2 Forward modelling

In the case of *forward modelling* the relevant norm is not H_0^1 but instead the one given by the Galerkin method as the mapping $y \mapsto v \in V_n$ is achieved solving

$$\sum_{j=1}^{d} y_j \int_{\Omega_j} \nabla v \cdot \nabla w \, dx = \langle f, w \rangle_{H^{-1}, H_0^1}$$

with $w \in V_n$ the test function on the variational formulation. The relevant norm is then

$$\|v\|_y^2 = \sum_{j=1}^{d} y_j \int_{\Omega_j} |\nabla v|^2$$

which defines the Galerkin projection $P_{V_n}^y$ onto the space V_n.

One would like then to have error estimates to uniformly bound $\|u - P_{V_n}^y u\|_{H_0^1}$ instead of $\|u - P_{V_n} u\|_{H_0^1}$ since P_{V_n} is out of reach in applications as one would require to know the function u which is the one we ignore in the first place and the reason we seek for approximations. On the contrary, $P_{V_n}^y$ is computable because in the $\|\cdot\|_y$ norm the target

function $u \in H_0^1$ and $v \in V_n$ have the same projection when tested against elements $w \in V_n$:

$$\langle u, w \rangle_y = \langle v, w \rangle_y.$$

A key observation is that we need to incorporate into the reduced spaces V_n some limiting solutions from V_S, otherwise if $V_n \cap V_S = \emptyset$ then there exists $y \in Y'$ such that

$$\|u(y) - P^y_{V_n} u(y)\|_{H_0^1} \geqslant C \|u(y)\|_{H_0^1}$$

for any $C \in]0, 1[$. One can intuitively see that this problem appears for the Galerkin projection as the norm $\|\cdot\|_y$ has $y = (y_{S^c}, y_S)$ which may contain some ∞ values forcing the contributions of elements $v \in V_n; v \notin V_S$ to be zero otherwise $\int_{\Omega_j} y_j \nabla v \cdot \nabla w \, dx = \infty$ for $j \in S$. As y does not appear in P_{V_n}, this problem did not show up before.

Taking this into consideration we obtain the same desired rates:

$$\|u(y) - P^y_{V_n} u(y)\|_{H_0^1} \leqslant C \exp(-cn^{\frac{1}{2d-2}})$$

with $y \in Y'$.

1.3.3 Inverse problem

Finally we use the previous results to show that the same reduced spaces V_n constructed so far can be used for the *inverse state estimation* problem retaining the same rates. This is expressed in the following proposition:

Proposition 1.3.3. *Let $y \in Y$ and $u = u(y)$. Then both estimators $\tilde{u} \in V_n$ and $u^* \in V_w := \{u \in V : \ell_i(u) = z_i, i = 1, \dots, m\}$ satisfy*

$$\max\{\|u - \tilde{u}\|_{H_0^1}, \|u - u^*\|_{H_0^1}\} \leqslant C\mu_n \exp\left(-cn^{\frac{1}{2d-2}}\right) \|u\|_{H_0^1}.$$

The positive constants c and C only depend on d, $\|f\|_{H^{-1}}$ and on the geometry of the partition.

Here, $\|u - \tilde{u}\|_{H_0^1}$ represents the approximation error of using the *reduced basis method* while $\|u - u^*\|_{H_0^1}$ corresponds to the modified version given by PBDW.

For the *inverse parameter estimation* problem the situation is usually more complicated due to the non linear nature of the inverse map $\ell(u(y)) \mapsto y$. However, if we take $V_n = \operatorname{span}\{u_1, \dots, u_n\}$ with $u_i = u(y^i)$ being a solution of the diffusion equation for some parameter vector y^i and use the fact that parameters y_j are associated with its piecewise constant diffusivity values over the corresponding Ω_j, then we have that on each subdomain Ω_j

$$f = -\operatorname{div}(y\tilde{u})_{|\Omega_j} = -\operatorname{div}(y_j \tilde{u}).$$

Using that $\tilde{u} = \sum_{i=1}^{n} c_i u_i$ is the approximation obtained after solving the *state estimation* problem

$$\frac{f}{y_j} = -\mathrm{div}\left(\sum_{i=1}^{n} c_i u_i\right)_{|\Omega_j} = \sum_{i=1}^{n} c_i \frac{f}{y_j^i}$$

finally yeilds the following estimator

$$y_j^* = \left(\sum_{i=1}^{n} \frac{c_i}{y_j^i}\right)^{-1}. \tag{1.11}$$

We also prove that this estimator achieves a recovery bound in relative error that also has a sub-exponential rate:

Proposition 1.3.4. *With the notation $1/y = (1/y_1, \ldots, 1/y_d)$, the estimator y^* defined by Equation (1.11) satisfies the bound*

$$\left\| \frac{1}{y^*} - \frac{1}{y} \right\|_\infty \leqslant C\mu_n \exp\left(-cn^{\frac{1}{2d-2}}\right) \left\| \frac{1}{y} \right\|_\infty.$$

1.4 Non-linear approximation spaces

In Chapter 3, based on [50], we extend to non-linear spaces the existing results on *near optimality* for *state estimation* inverse problems. We seek to approximate a function u from m measurements $\ell(u) = z \in \mathbb{R}^m$, with functionals ℓ and a reconstruction procedure $\tilde{u} = R(z)$ where neither of them is necessarily linear.

Let us first recall the linear setting introduced in [108] and known as *Parameterized Background Data Weak* (PBDW). Here, the reconstruction strategy is based on linear spaces V_n and assumes that

- V is a Hilbert space,
- $\ell_j \in V'$ are linear functionals.

In this context $\ell_j(u) = \langle w_j, u \rangle_V$ with $w_j \in V$ being the Riesz representer of ℓ_j. Then $w = \sum_{j=1}^{m} \ell(u)_j w_j = P_W u$ with $W = \mathrm{span}\{w_1, \ldots, w_m\}$. A first reconstruction strategy consists of finding $\tilde{u} \in V_n$ minimizing the discrepancy with the observed data:

$$\tilde{u} = \arg\min_{v \in V_n} \|P_W v - w\|_V.$$

However, in a noise-free scenario, one would prefer reconstructions u^* agreeing with the measurements, that is, $z = \ell(u^*)$. This won't be typically achieved by the estimator $\tilde{u}$. But we can correct it through

$$u^* = \tilde{u} + (w - P_W \tilde{u})$$

to obtain the PBDW measurement consistent estimator.

Note that if $V \subset \mathcal{C}(\Omega)$ and the ℓ represent point evaluations then $\ell_j \in V'(\Omega)$ are Dirac masses and the correction becomes useless. To make sense of this correction the ambient space V' has to have enough regularity. This is typically achieved in the context of *Reproducing Kernel Hilbert Spaces* RKHS where one asks for the point evaluation to be a continuous, bounded linear functional. In applications one may introduce kernel functions like Gaussians to get Riesz representers of the point evaluation that are smoother.

It is proved in [29, 108] that both estimators $\tilde{u}$ and u^* satisfy the recovery bound

$$\max\{\|u-\tilde{u}\|_V, \|u-u^*\|_V\} \leqslant \mu(e_n(u)+\beta\epsilon) \tag{1.12}$$

where $e_n(u) = \min_{v\in V_n}\|u-v\|_V$ is the lowest possible error that can be achieved when approximating u by an element of V_n. This means that the reconstruction achieves near optimal approximations. The error due to the presence of additive noise $z = \ell(u)+\eta$ where $\eta = (\eta_1,\dots,\eta_m)$ is denoted by $\beta\epsilon = \|w-\overline{w}\|_V$ where $\overline{w}$ is the perturbed version of w. The ℓ^p norm is bounded by ϵ

$$\|\eta\|_p \leqslant \epsilon$$

which quantifies the level of noise, and

$$\beta := \max_{v\in W}\frac{\|v\|_V}{\|\ell(v)\|_p}.$$

The constant

$$\mu = \mu(V_n, W) := \max_{v\in V_n}\frac{\|v\|_V}{\|P_W v\|_V}$$

is a measure of the stability of the reconstruction. It can be seen as the inverse cosine of the angle between both spaces. For example, if W is orthogonal to V_n, any element of v will have the same projection onto W, $P_W v = P_W v' \;\forall\, v, v' \in V_n$, consequently $\mu = \infty$ (see Figure 1.5). The same occurs if $m < n$ as we have less measurements than the dimension of the reduced space.

The question then is what properties are needed for retaining near optimal bounds as in (1.12) when considering non-linear n-parameter approximation families V_n.

The main conclusion of this work is that it is possible to get near optimal bounds for non-linear n-parameter approximation families if the two following properties are satisfied:

- *The measurement functionals ℓ_i, not necessarily linear, are **Lipschitz continuous**, that is*

$$\|\ell(v)-\ell(u)\|_Z \leqslant \alpha_Z\|v-u\|_V, \quad v,u\in V \tag{1.13}$$

  with $\|\cdot\|_Z$ any norm in $\mathbb{R}^m$

- *For the space V_n the following **inverse stability** holds:*

$$\alpha_Z\|v-u\|_V \leqslant \mu_Z\|\ell(v)-\ell(u)\|_Z, \quad v,u\in V_n. \tag{1.14}$$

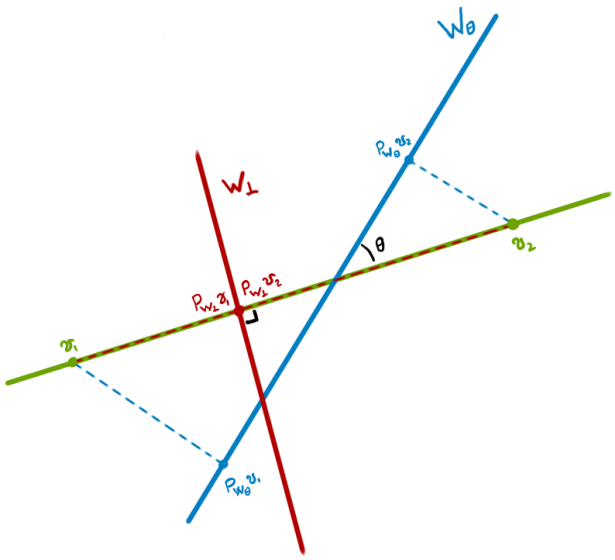


Figure 1.5: Reduced space V_n in green. In red and blue two different measurement spaces W_θ and $W_\perp$. The projections of elements $v_1, v_2 \in V_n$ into $W_\perp$ collapse into the same element $P_{W_\perp} v_1 = P_{W_\perp} v_2$.

Here, μ_Z has a similar role as μ for PBDW and we also show that it is finite only if $m \geqslant n$.

The constants α_Z and μ_Z are optimally defined as follows:

$$\alpha_Z = \sup_{v,u \in V} \frac{\|\ell(v) - \ell(u)\|_Z}{\|v - u\|_V},$$

and

$$\mu_Z = \sup_{v,u \in V_n} \frac{\|v - u\|_V}{\|\ell(v) - \ell(u)\|_Z}.$$

The framework presented here allows us to study the *best-fit estimator* issued from the minimisation of the discrepancy between observed data z and the one given by the reconstruction $\ell(\tilde{u})$ with $z \mapsto R(z) = \tilde{u} \in V_n$ such that

$$\tilde{u} := \arg\min_{v \in V_n} \|z - \ell(v)\|_Z. \tag{1.15}$$

With these definitions and conditions we arrive at similar error estimates as in (1.12) stated in the following theorem

Theorem 1.4.1. *The best fit estimator $\tilde{u}$ from (1.15) satisfies the estimate*

$$\|u - \tilde{u}\|_V \leqslant C_1 e_n(u) + C_2 \|\eta\|_p, \tag{1.16}$$

where $C_1 \coloneqq 1 + 2\alpha_Z \mu_Z$ *and* $C_2 \coloneqq 2\beta_Z \mu_Z$.

In the situation in which ℓ_i are linear functionals, while V_n still being a non-linear space, we can identify the norms that minimize C_1 or C_2. For the first one let us define the *Riesz norm*

$$\|z\|_W = \min\{\|v\|_V : \ell(v) = z\}, \tag{1.17}$$

which is a norm in $\mathbb{R}^n$ given by the minimal normed element of V compatible with the observations. If V is a Hilbert space then $\|\ell(v)\|_W = \|P_W v\|_V$.

On the one hand we have that $\|\cdot\|_W$ norm is the one minimizing the constant C_1 as it favors minimal energy approximations $v \in V_n$ consistent with the observations. This can be seen as an implicit regularisation. The constant C_2, on the other hand, is minimized by taking the ℓ^p norm.

Shape recovery from cell averages: As an application of the above general reconstruction framework, we consider the problem of recovering the shape of an interface from cell average data. This situation appears in image processing when we want to reconstruct high resolution images from lower resolution ones. Another application is in hyperbolic conservation laws when finite volume schemes are used to discretize the domain and solve the equation which is an application that we study more in depth in Chapter 4 (see Section 1.5).

In this context, V consists on characteristic functions $u = \chi_\Omega$ defined on a rectangular domain $D = [0,1]^d$ with $\Omega \subset D$. The objective is then to reconstruct the shape of Ω from local cell average information on a cartesian partition of the domain:

$$a_T(u) = \frac{1}{T}\int_T \chi_\Omega, \quad T = T_{ij} = [(i-1)h, ih] \times [(j-1)h, jh]. \tag{1.18}$$

Here $1 \leqslant i, j \leqslant 1/h$ with $h > 0$ being the cell size and $N = h^{-2}$ the number of cells. Now the measurement functionals $\ell(u)$ are $(a_{T'}(u))_{T' \in S} \in \mathbb{R}^m$, with $\#S = m$, as they represent the local cell averages on the stencil S around cell T used for the reconstruction.

First we show that the L^1 error of approximating an interface by a linear method cannot be better than the rate $N^{-\frac{1}{2}}$, or equivalently h^1, regardless of the regularity of the interface.

However, we can approximate the true interface, on a given cell T, by an element of the non-linear family $V_2 := \{\chi_{\vec{n}\cdot(x-\bar{x})\geqslant c} : \vec{n} \in \mathbb{S}^1,\, c \in \mathbb{R}\}$ composed of the indicator functions whose interface is a line (see Figure 1.6). The reconstruction operator $R(z) \mapsto v \in V_2$ is defined using (1.15) with $\ell(u) = z \in \mathbb{R}^9$ the local measurement vector obtained after using a 3×3 T-centred stencil. Note that this is a two parameter non-linear family despite the fact that the interface is defined through a line.

We show first that the space V_2 with the 9 linear measurements yields values of $\alpha = h^{-2}$, $\beta = 9^{1-\frac{1}{p}}$ and $\mu = \frac{3}{2}h^2$ ensuring that the two sufficient conditions of (1.13) and (1.14) hold and consequently also (1.16). This leads us to deduce a global approximation rate of N^{-1} or equivalently h^2 which is an order better than the linear case.

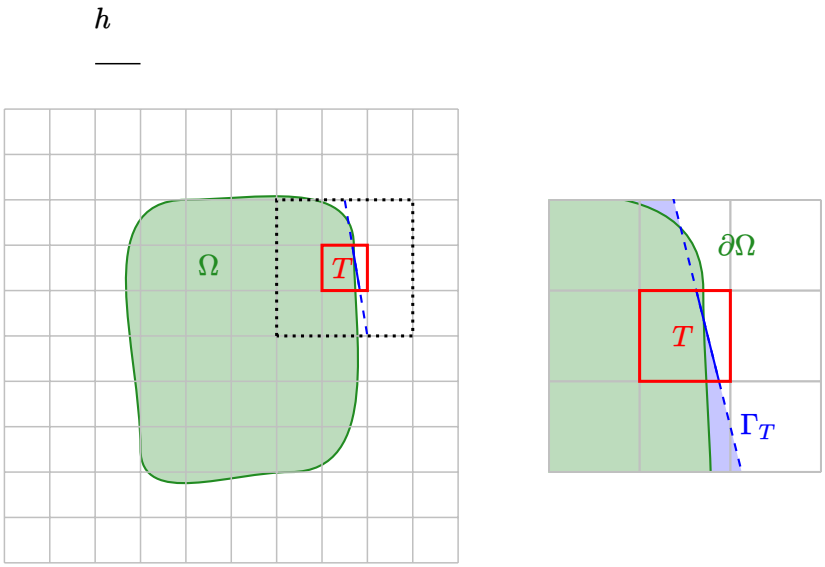


Figure 1.6: Scheme of the problem of shape recovery from cell averages. On each cell T we locally approximate the interface $\partial\Omega$ by a linear interface Γ_T.

1.5 High order non-linear interface reconstruction strategies

In Chapter 4, which is based on paper [56], we focus on the shape recovery problem. Here we seek to develop local fast non-linear interface reconstruction strategies of high order of accuracy using information from cell averages. We also work with the space V composed of piecewise constant functions $u = \chi_\Omega$ with $\partial\Omega = \Gamma$ of certain Hölder smoothness, that is, the boundary can locally be described by graphs of $\mathcal{C}^s$ functions (see [116] and also Chapter 4 of [2]).

We denote by *singular cells* $\mathcal{S}_h$ those containing part of the boundary of Ω where we will apply our reconstruction strategies. In the remaining, the *regular cells*, we have the constant value 0 or 1 given by the observed average $a_T(u)$ with T *regular*. Then, on each singular cell $T \in \mathcal{S}_h$ we will locally approximate the true interface by an element $\tilde{u}_T$ of a given non-linear family V_n. We explore non-linear families composed of indicator functions with the transition given by the following parametric curves:

- Polynomials: the curve is a polynomial of degree $n-1$.
- Circles: the curve is described by three parameters: the radius r and the coordinates of the center of the circle (x_0, y_0).
- Corners: the curve is described by the angles (θ_1, θ_2) of two lines that intersect at the point (x_0, y_0).

We propose two main families of approaches to define the reconstruction operator $R_T : \mathbb{R}^m \to V_n$.

Optimization Based Edge Reconstruction Algorithms (OBERA)

The element $\tilde{u}_T \in V_n$ is chosen by optimization in the same way as in (1.15), through the best fit of the available cell-average data on the associated stencil $S = S_T$. The minimization takes the form

$$\tilde{u}_T = R_T(a_S(u)) \in \underset{v \in V_n}{\arg\min} \|a_S(v) - a_S(u)\| \tag{1.19}$$

where $\|\cdot\|$ is a given norm on $\mathbb{R}^m$, usually ℓ^2, and $m := \#(S)$ is the size of the stencil, for example, $m = 9$ for a 3×3 square stencil. $a_S : V \to \mathbb{R}^m$ represents the local measurement operation (that we denoted by ℓ in the general context), here consisting of the cell averages of the stencil S. This method, when restricted to linear interfaces, it is known as LVIRA [132, 129].

Note that one can modify the norm under which is performed the optimization (1.19) to force the reconstruction $\tilde{u}_T$ to have area consistency on the relevant cell, that is $a_T(u) = a_T(\tilde{u}_T)$. We do this by adding to (1.19) the term $K|a_T(u) - a_T(\tilde{u}_T)|$ with $K = 100$.

Performing an optimization on each cell becomes computationally very demanding, to avoid this we propose yet another strategy.

Algorithms for Edge Reconstruction using Oriented Stencils (AEROS)

In the previous method the spatial structure of the stencil is not exploited, here we retain the matrix form of $a_S \in \mathbb{R}^{k \times l}$. We consider the sum of columns (or rows), depending on a chosen *orientation*, and we look at the resulting values $(a)_i \in \mathbb{R}^k$ as one-dimensional averages. Afterwards, we can find an approximation to the interface Γ by an $n \leqslant k$ parameterizable curve defined as the graph of a function p such that

$$y = p(x) \quad \text{or} \quad x = p(y). \tag{1.20}$$

As before, we look for the best fit of the available 1d-cell-average data which, for some families of curves, *i.e.*, polynomials, amounts at solving an $n \times n$ linear system. In addition, the use of polynomials give a relatively straightforward way to analyse the order of accuracy attainable through this strategy.

One central assumption is that Γ has to cross the sides of the stencil to guarantee that the 1d-averages can be associated to an integral on an interval. We ensure this through the following two steps:

1. First, we choose an *orientation*, that can be vertical or horizontal, using a Sobel filter yielding an approximation to the numerical gradient $G_T = (H_T, V_T)$. Consequently,

   - if $|V_T| \geqslant |H_T|$ then we will search for functions $p(x)$ with x as independent variable.
   - if $|V_T| < |H_T|$ then we will search for functions $p(y)$ with y as independent variable.

   The sign of V_T or H_T determines, in addition, if the domain Ω lies below or above the curve defined through p.

2. Second, we adaptively choose a rectangular stencil S of size $k \times l$ with l big enough so that the singular cells that are contained in the stencil are neither in the first nor in the last rows. In addition, the stencil may also be allowed to shift horizontally to avoid l becoming too big if the cell T is in a region where Γ is rapidly changing orientation.

The main conclusion of this work is that both proposed strategies, OBERA and AEROS, yield interface reconstructions of $\mathcal{O}(h^{d+1})$ order of accuracy when the interface is parametrized by a polynomial of degree d. In addition, AEROS strategy is two orders of magnitude faster than OBERA.

We prove that this orientation test based on the Sobel filter correctly finds the preferable orientation when the boundary is determined by a line. Furthermore, following a perturbation analysis, we also show that if the interface is not a line and the mesh size h is below a critical value of h^*, then one can find a stencil tall or wide enough ensuring the true interface will cross the sides of the stencil.

Based on these results we also provide a quantification of the order of convergence of AEROS, when using polynomials of degree $n-1$ to approximate the interface, as shown in the following theorem:

Theorem 1.5.1. *Let Ω be a $\mathcal{C}^s$ domain for some $s \geqslant 1$. The AEROS recovery of the interface based on polynomial of degree $n-1$ satisfies for each singular cell T a local error bound of the form*

$$\|u - R_T(a_S(u))\|_{L^1(T)} \leqslant Ch^{r+1}, \quad r := \min\{s, n\}, \tag{1.21}$$

and the global error bound of order $\mathcal{O}(h^r)$ for the same value of r.

In many applications, the interface might contain corners for which other strategies need to be developed. We give two methods which, combined together, yield visually "good" approximations of vertices.

- **AEROS Vertex**: AEROS strategy is not necessarily restricted to the case of polynomial interfaces. We propose to use the four parameter family of corner interfaces V_4, for which it is possible to find explicit algebraic equations for the inverse map $a_{S_T}(u) \mapsto y = (\theta_1, \theta_2, x_0, y_0)$ yielding the parameters y of $\tilde{u}_T \in V_4$. This strategy, as it is based on the orientability of the interface, cannot deal with right angles when they are parallel to the mesh orientation. This limitation motivates the following method.

- **Tangent Extension Method (TEM)**: For this method to work we first need to have an approximation $\tilde{u}_T$ on each singular cell $T \in S_h$. In order to propose a new corner approximation on a given cell $T \in S_h$, we properly pick two other singular cells from the neighbourhood and extend two lines issued from a Taylor expansion of order 1 at some chosen point. This procedure yields two angles (θ_1, θ_2) and the intersection (x_0, y_0) which are the four parameters needed to describe an element of V_4.

One key ingredient behind both methods is the possibility of deciding on each cell $T \in S_h$ which approximation should remain between the possibly many tested alternatives, for example, polynomial *versus* corner interfaces. To this end we compare the stencil cell averages issued from each reconstruction, $a_S(\tilde{u}_1)$ and $a_S(\tilde{u}_2)$, with the observed ones $a_S(u)$ and keep the method yielding the least discrepancy error $\|a_S(v) - a_S(u)\|$, with $v = \tilde{u}_1$ or $v = \tilde{u}_2$.

1.6 Non-linear reduced basis

Chapter 5, based on [55], also focuses on developing non-linear strategies to deal with classes of functions that are known to be poorly approximated by linear models. The starting point here is the observation that in some situations even though both $d_n(\mathcal{M})_V$ and $\kappa_n(\mathcal{M})$ decrease slowly when n grows, other complexity measures, like the *stable non-linear widths* (see [66]) or the *sensing numbers* $s_n(\mathcal{M})_V$ may decrease much faster. The latter is defined as

$$s_n(\mathcal{M})_V := \inf_{R,\ell_1,\dots,\ell_n} \max_{u\in\mathcal{M}} \|u - R(\ell_1(u),\dots,\ell_n(u))\|_V, \tag{1.22}$$

where the infimum is taken over all choices of linear functionals $\ell_1,\dots,\ell_n \in V'$ and reconstruction map R. While the *Kolmogorov n-width* imposes $\mathcal{G}_n$ to be a linear space, in the *sensing numbers* $s_n(\mathcal{M})_V$ the approximation spaces are allowed to be non-linear provided that the information used for the reconstruction are linear measurements ℓ. In absence of an explicit form for the optimal R we can write an approximation using some learning technique such as a *neural network* or *random forests* which usually are defined using many parameters θ with $\#\theta \gg n$. Note also that in the definition of $s_n(\mathcal{M})_V$ the relevant dimension n is the number of functionals and not the number of parameters needed to encode the reconstruction map[1].

Also notice that the optimal ℓ_i are usually unattainable or computationally too costly to obtain. However, one can take a *reduced basis* $V_n = \operatorname{span}\{u_1,\dots,u_n\}$ and define $\ell_i(u) = c_i$ with c_i the coefficients obtained after solving the *forward modelling* (see (1.5)) or the *inverse problem* (see (1.6)). Using this choice we cannot achieve the optimal error given by $s_n(\mathcal{M})_V$, but we can build sub-optimal strategies, hoping to beat linear methods.[2] This is given by

[1] In the linear case the reconstruction consists of $R(u) = \sum_{i=1}^n c_i u_i$ with c_i found by solving an $n \times n$ (*forward modelling*) or $m \times n$ (*inverse problem*) system. In the non-linear case considered here, there is in principle no restriction on the number of parameters $\#\theta$ one may use to write R.

[2] If we also impose the reconstruction to be linear $R(u) = \sum_{i=1}^n \ell_i(u) u_i = P_{V_n} u$ we would fall again into the linear setting.

the following:

$$\begin{aligned} R(u) &= P_{V_n} u + P_{V_{N-n}} u \\ &= \sum_{i=1}^{n} \ell_i(u) u_i + \sum_{i=n+1}^{N} \tilde{\ell}_i(u) u_i. \end{aligned} \tag{1.23}$$

Here $(u_i)_{1 \leqslant i \leqslant N}$ are the basis elements of a bigger linear space V_N and by the second term we seek to perform a correction of the linear model as if we were working on an N dimensional space even though we only have access to the first n coefficients of such decomposition.

The remaining $N - n$ coefficients $\tilde{\ell}(u)$ have to be learned from the known ones $\ell(u)$. That is, for $i > n$, $\ell_i(u) \approx \tilde{\ell}_i(u) = \psi_i(\ell_1(u), \dots, \ell_n(u))$, we need to construct $N - n$ functions $\psi_i : \mathbb{R}^n \to \mathbb{R}$ mapping the known coefficients to the unknown ones. Usually, in the *offline* stage of a reduced basis method many snapshots of the problem of interest are available $\{v_1, \dots, v_K\}$. From those snapshots we can build both spaces V_n and V_{N-n} with $n \ll N \leqslant K$ and $V_N = V_n \oplus V_{N-n}$. As a consequence, for each element $v_j \in V_{N-n}$ we can calculate $\ell_i(v_j) = \langle u_i, v_j \rangle$ with $u_i \in V_n$ and build a training dataset so that a learning technique like *random forests* can be used to learn a "good" mapping.

The main conclusion of this work is that in the context of reconstructing characteristic functions it is possible to estimate $\ell_i(u)$ for $N \geqslant i > n$ from the first $\ell_i(u)$, $i \leqslant n$ by using learning techniques.

Remark 1.6.1. *Note that the computational burden of calculating the second term of Equation (1.23) would have been of $\mathcal{O}(N^3)$ if we had used the space V_N instead of the space V_n. However, when using the learnt non-linear mapping this reduces to $\mathcal{O}(N^2)$ if the cost of performing the non-linear map is small*[3].

The rest of the work concentrates on showing by numerical experimentation that this approach yields reliable improvements for *random forests* as a learning technique. We use the eigenvectors of the *Karhunen–Loève* basis associated to the family of step functions to build the spaces V_n and V_N.

1.7 Physics Informed Neural Networks for singularly perturbed convection-diffusion equations

In Chapter 6, which is based on paper [20], we study the ability of different formulations of Physics Informed Neural Networks (PINNs) to solve singularly perturbed convection-

[3]For this estimation we supposed that the discretized dimension of the ambient space is equal to N, $\dim(V) = N$ but it could be bigger.

diffusion equations. For simplicity we focused on a 1d-simplified version which reads:

$$-\varepsilon u''(x) + Fu'(x) = f, \quad \forall x \in \Omega = (0,1), \tag{1.24}$$

with Robin boundary conditions

$$\begin{aligned} -\alpha u'(0) + \kappa u(0) &= g_0, \\ \alpha u'(1) + \kappa u(1) &= g_1. \end{aligned} \tag{1.25}$$

The vanishing ε is known to affect the performance of numerical methods, such as Finite Element Method (FEM), since the bound on the approximation error of the Galerkin method degrades proportional to ε^{-1}. We recall that the Galerkin method, based on a variational formulation, searches for approximations v to the problem in the space $H^1(\Omega)$ which is less regular than $H^2(\Omega)$ as it is a requirement for PINNs.

PINNs approach does not rely on the variational formulation. It works with a strong residual formulation and it uses a *neural network* $\mathcal{NN}_\theta : \mathbb{R} \to \mathbb{R}$ as an ansatz for the solution. Here $\theta \in \mathbb{R}^q$ are the parameters of the *neural network* that are learned by minimizing the residual of the original differential equation:

$$\mathcal{R} = \sum_{\substack{x_i \in \Omega \\ i=1,\dots,m}} |\mathcal{P}\left(\mathcal{NN}_\theta(x_i), y\right)|^2 + \sum_{\substack{x_i \in \partial\Omega \\ i=1,\dots,m_b}} |\mathcal{B}(\mathcal{NN}_\theta(x_i), y)|^2. \tag{1.26}$$

Here $\mathcal{P}$ also represents the differential operator which is given by (1.24) while $\mathcal{B}$ accounts for the boundary term of (1.25). The parameters of the equation are $y = (\varepsilon, F, f, \alpha, \kappa, g_0, g_1)$, however for our numerical test we fix all except ε.

The key feature of PINNs is that they rely heavily on the possibility offered by today's software of easily computing, via *automatic differentiation*, the exact value of the gradient of a $\mathcal{NN}_\theta$ at a given point. One can obtain the gradient with respect to both its inputs $x \in \mathbb{R}^d$ and its parameters θ. The latter is needed for the learning task as it relies on gradient descent optimization strategies. However, one can go further and calculate multiple derivatives with respect to the inputs allowing us to evaluate the differential operator $\mathcal{P}$ at any point $x \in \Omega$.

As the $\mathcal{NN}_\theta$ is trained to satisfy the equation at random points there is the hope that it will eventually converge to a reliable approximation $\tilde{u}(x) = \mathcal{NN}(x)$ of the solution $u(x)$ of the PDE.

This strategy is also expected to suffer from the vanishing ε as one would required to sample in a too small region to capture the rapidly changing shape of the true solution around the forming shock.

In this work we present alternative formulations to the classical PINN:

- **Vanilla**: as described above.

- **Vanilla with change of variables**: to define the learning task we first redefine the

equation by performing the following change of variables:

$$u(x) = e^{-cFx} z(x) \tag{1.27}$$

and train the $\mathcal{NN}$ as in vanilla PINN but on this new system with z as the input variable instead of x.

- **Weak formulation and change of variables**: Noting that the resulting new system is elliptic, we can write it in variational formulation where z is the new unkown. The $\mathcal{NN}$ is then trained using the weak formulation system.

  Notice that the summation in Equation (1.26) is due to the need of sampling Ω pointwise in order to have a coverage of the relevant regions. In the weak formulation, however, the sampling has a precise interpretation: it is the discretization of the weak integrals via a quadrature method of choice.

- **Weak formulation, change of variables and domain rescaling**: Starting from the weak formulation we also perform a domain rescaling $\tilde{\Omega} = \Omega/\varepsilon$ in the hope of taming the exponentials that appeared explicitly after the change of variables.

- **Weak formulation**: The learning task is also based on the weak formulation but going back to the original variables in order to make the optimization on u again instead of z.

The general conclusion is that all the methods degrade with vanishing ε although PINNs are an easy to implement alternative to FEM not requiring many collocation points to get reliable approximations on high ε regimes.

In particular all the methods optimized on the changed variable z blow up below $\varepsilon \lesssim \varepsilon^* \approx 0.063$. From the remaining, vanilla PINN shows the best results having almost constant accuracy for bigger values of e although it fails for smaller values of ε, it remains less dramatic compared to the rest. If FEM is implemented with sparse matrices, then one can use very dense meshes and still be competitive in terms of computing time with respect to vanilla PINN. Interestingly, though, by using just $m = 10$ collocation points one can achieve using Vanilla PINN, the same accuracy level as with $m = 10000$ for FEM.

1.8 Modelling pollution at a city scale

Chapter 7, based on paper [67], is dedicated to the state estimation inverse problem in the context of modelling air pollution at the scale of a city. One main difficulty is the few available sensors measuring NO_2 pollution $z \in \mathbb{R}^m$ which in the studied scenario, the city of Paris, consisted only of $m = 13$ spatial locations x_i^{obs}. To enhance this data with local

features we developed an automatic way of capturing Google Map's traffic screenshots for the whole city. In addition, we also took into account wind $w(t)$ and temperature series $\theta(t)$ for the three studied months. To combine all this heterogeneous sources of information we propose data-driven, physics-driven or hybrid approaches and showed that it is possible to build relatively accurate pollution cartographies.

The main conclusion of this work is that one can improve pollution estimations by incorporating traffic data processed from Google Maps screenshots.

The collected traffic data has to be preprocessed before it can be properly used for pollution prediction. For this, we decided to work on the graph representation $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ of the city streets, where the edges E represent the streets and the nodes V the intersections. Then, for each image, we first extract the pixels containing one of the four possible traffic colors and project them onto the closest edge. Finally, we aggregate this information on each node such that we have for each time t and each node v a vector $q_{\mathsf{c}}^{\mathsf{v}}(t) \in \mathbb{R}^4$ representing the neighbouring traffic information.

To compare results we use two models as baselines. The first one consists on estimating for every point $x \in \mathbb{R}^2$ the same value given by the *spatial average* of the observed pollution $\overline{z}(t) = \frac{1}{m}\sum_i^m z_i(t)$. The second model is known as the *Best Linear Unbiased Estimator* ($BLUE$) and gives the best possible accuracy for any linear method using z as the only available information. This method, can not be used as a state estimation model because it requires the knowledge of the full statistics (averages and covariances) of every point of interest. However, it serves as a lower error bound for other strategies.

By looking at the covariance between stations and their respective pairwise distances one can see that pollution concentrations are more correlated when their locations are near and less when they are far. One can fit an exponential decay to this relation and use it to get an approximate value of the covariances between locations with and without sensors. By introducing this approximate covariances one can modify $BLUE$ and make it a state estimation model known as *Kriging*.

A straightforward way to take into account the traffic data is by building a parametric model $\mathcal{T}_\alpha : \mathbb{R}^4 \to \mathbb{R}$ that transforms the given four color densities associated to a node $q_{\mathsf{c}}^{\mathsf{v}}(t)$ into a local correction with respect to the *spatial average* pollution estimation. The parameters α are found differently depending on form of $\mathcal{T}$: LASSO regression with cross validation for *linear* or *polynomial* models; gradient descent with early stopping for *neural networks*. This learning is done using the observed stations by associating each location to the nearest node in the graph.

The previous strategy, named *source* model, gives very localized corrections as for each point x the pollution prediction depends only on the traffic values of the nearest node. To generate smoothed versions of $q_{\mathsf{c}}^{\mathsf{v}}(t)$ we project $q_{\mathsf{c}}^{\mathsf{v}}(t)$ on a reduced basis that we build from the first eigenvectors of the graph Laplacian associated with G. This can be seen as solving a reaction-diffusion equation on the graph with source term given by the initial un-smoothed field $q_{\mathsf{c}}^{\mathsf{v}}(t)$. After this process one can again learn a mapping $\mathcal{T}_\alpha$ by taking the smoothed version of the traffic densities.

Finally, as the model *Kriging* is only good at estimating the pollution density at points that are close to a sensor, we take the outputs of *Kriging* and the ones given by the traffic dependant strategies and average them with varying weights that depend on the distance to the nearest sensor. This combined strategy manages to consistently improve with respect to the *spatial average* baseline.

1.9 Python package for reproducible research

The development of a unified framework to systematically define, execute, store, and present numerical experiments came from the observation that in almost all scientific projects involving coding many tasks not related to the actual problem can be abstracted and automated. To mention a few:

- File management: In every project one needs to define a location to store data, logs, and results generated during experimentation.
- Save, load, and check experiments: The inputs and results of experiments are stored in a tree structure that can be retrieved to further analysis or to avoid recomputing already done experiments.
- Parallelize experiments: It is possible to parallelize the exploration of independent runs by just writing the number of desired cores to be used.
- Export results: The results of the experiments can be exported in user-friendly formats such as .csv for further analysis outside Python.
- Fast visual exploration: One can produce basic and specifically designed visuals to intermediately get insights and explore stored results.
- Connect to LaTeX: One can insert in the text variables directly issued from the experiments. This reduces the chances of writing in a report outdated parameters and results as they will be automatically modified if any change on the code is produced.
- Measure CO_2 emissions: Using [38] package one can also track the electricity consumption and approximate CO_2 emissions generated by the experiments and analysis.

This package can be found at `https://github.com/agussomacal/PerplexityLab` and it was used for structuring the experiments of papers [50, 55, 56, 67]. Other related projects are AiiDA [95] and `https://pydoit.org/`.

Chapter 2

Reduced order modelling for elliptic problems with high contrast diffusion coefficients

2.1 Introduction

2.1.1 Reduced models for parametrized PDEs

Parametric PDE's are commonly used to describe complex physical phenomena. With $y = (y_1, \dots, y_d)$ denoting a parameter vector ranging in some domain $Y \subset \mathbb{R}^d$, and $u(y)$ the corresponding solution to the PDE of interest, assumed to be well defined in some Hilbert space V, we denote by

$$\mathcal{M} := \{u(y) \ : \ y \in Y\}, \tag{2.1}$$

the collection of all solutions, called the *solution manifold.*

There are two main ranges of problems associated to parametric PDEs:

1. Forward modelling: in applications where many queries of the parameter to solution map $y \mapsto u(y)$ are required, one needs numerical forward solvers that efficiently compute approximations $\tilde{u}(y)$ with a prescribed accuracy.

2. Inverse problems: when the exact value of the parameter y is unknown, one is interested in either recovering an approximation to $u(y)$ (state estimation) or to y (parameter estimation), from a limited number of observations $z_i = \ell_i(u(y))$, possibly corrupted by noise.

Reduced order modelling is widely used for tackling both problems. In its most common form, its aim is to construct linear spaces V_n of moderate dimension n that approximate all solutions $u(y)$ with best possible certified accuracy. The natural benchmark for measuring the performance of such linear reduced models is provided by the *Kolmogorov n-width* of the solution manifold

$$d_n(\mathcal{M})_V := \inf_{\dim(V_n)=n} \operatorname{dist}(\mathcal{M}, V_n)_V \tag{2.2}$$

that describes the performance of an optimal space. Here

$$\operatorname{dist}(\mathcal{M}, V_n)_V := \sup_{u\in\mathcal{M}} \inf_{v\in V_n} \|u-v\|_V = \sup_{u\in\mathcal{M}} \|u - P_{V_n}u\|_V,$$

where P_{V_n} is the V-orthogonal projector onto V_n. We refer the reader to [130] for a general treatment of n-widths.

While an optimal space achieving the above infimum is usually out of reach, there exist two main approaches aiming to construct "sub-optimal yet good" spaces. The first one consists in building expansions of the parameter to solution map, for example by polynomials

$$u_n(y) := \sum_{\nu\in\Lambda_n} u_\nu y^\nu, \quad y^\nu := y_1^{\nu_1}\dots y_d^{\nu_d}, \tag{2.3}$$

where $\Lambda_n \subset \mathbb{N}^d$ is a set of cardinality n. The coefficients u_ν are elements of V and therefore, for all $y \in Y$ the approximation $u_n(y)$ is picked from the space

$$V_n := \operatorname{span}\{u_\nu \ : \ \nu \in \Lambda_n\}.$$

Notice that $u_n(y)$ is not the orthogonal projection $P_{V_n}u(y)$ in this case, but $u_n(y)$ is easy to compute for a given query y once the u_ν have been constructed (usually through a high fidelity finite element solver). We refer to [12, 19, 15],[14, 52, 54, 150] for instances of this approach.

The second approach is the reduced basis method [85, 140, 144], that consists in taking

$$V_n := \operatorname{span}\{u^1, \dots, u^n\},$$

where the $u^j = u(y^j)$ are particular solution instances corresponding to a selection of parameter vectors $y^j \in Y$. A close variant is the proper orthogonal decomposition method [45, 153, 158], where the reduced spaces are obtained by principal component analysis applied to large training set of such instances. In the reduced basis method, the parameter vectors $y^1, \dots, y^n$ can be selected by a greedy algorithm, introduced in [98] and originally studied in [39]. For such a selection process, it is proved in [28, 65] that if $d_n(\mathcal{M})_V$ has a certain algebraic or exponential rate of decay with n, then a similar rate is achieved by $\operatorname{dist}(\mathcal{M}, V_n)_V$ for the reduced basis spaces.

It follows that the reduced basis spaces constructed by the greedy algorithm are close to optimal. This is in contrast to the spaces V_n spanned by the polynomial coefficients u_ν for which the approximation rate is not guaranteed to be optimal. We refer to [13] for instances where reduced basis methods can be proved to converge with a strictly higher rate than polynomial approximations. On the other hand, the polynomial constructions (2.3) have certain numerical advantages. Namely, for several relevant classes of parametrized PDEs, it can be shown that the parameter to solution mapping $y \mapsto u(y)$ has certain smoothness properties that can be used to obtain a-priori bounds on the $\|u_\nu\|_V$ without actually computing these norms. This allows an a priori selection of an appropriate set Λ_n and the proof of concrete approximation estimates for the error $\sup_{y\in Y} \|u(y) - u_n(y)\|_V$.

These estimates in turn provide an upper bound for $d_n(\mathcal{M})_V$, and therefore for reduced basis approximations.

2.1.2 Parametrized elliptic PDEs

One prototypal instance where the convergence analysis described above has been deeply studied is the parametrized second order elliptic equation

$$-\operatorname{div}(a(y)\nabla u(y)) = f \quad \text{in } \Omega, \quad u_{|\partial\Omega} = 0 \quad \text{on } \partial\Omega, \tag{2.4}$$

where $\Omega \subset \mathbb{R}^m$ is the spatial domain, $f \in H^{-1}(\Omega)$ is a source term, and $a(y)$ has the *affine* form

$$a(y) = \overline{a} + \sum_{j=1}^{d} y_j \psi_j, \tag{2.5}$$

with $\overline{a}$ and $(\psi_1, \ldots, \psi_d)$ some fixed functions in $L^\infty(\Omega)$.

The corresponding solution $u(y) \in H^1_0(\Omega)$ is defined through the standard variational formulation in $H^1_0(\Omega)$ equipped with its usual norm. Up to renormalization, it is usually assumed that the y_j range in $[-1,1]$, or equivalently $Y = [-1,1]^d$. To ensure existence and uniqueness of solutions, one typically assumes that the so-called *Uniform Ellipticity Assumption* (UEA) holds: for some fixed $0 < r \leqslant R < \infty$,

$$r \leqslant a(x,y) \leqslant R, \qquad x \in \Omega, \quad y \in Y, \tag{2.6}$$

where $a(x,y) := a(y)(x) = \overline{a}(x) + \sum_{j=1}^d y_j \psi_j(x)$, or in short $r \leqslant a(y) \leqslant R$ for all $y \in Y$. Under this assumption, Lax-Milgram theory ensures that the solution map $y \mapsto u(y)$ is well defined from Y into $H^1_0(\Omega)$, with the uniform bound

$$\|u(y)\|_{H^1_0} := \|\nabla u(y)\|_{L^2} \leqslant \frac{C_f}{r}, \quad y \in Y.$$

Here and throughout this paper

$$C_f := \|f\|_{H^{-1}}. \tag{2.7}$$

It was proved in [15, 150] that, under UEA, polynomial approximations (2.3) of given total degree converge sub-exponentially: for $\Lambda_n = \{|\nu| \leqslant k\}$ with $n = \binom{k+d}{d}$, one has

$$\sup_{y\in Y} \|u(y) - u_n(y)\|_{H^1_0} \leqslant C' \exp(-cn^{1/d}), \tag{2.8}$$

Such sub-exponential rates show that the spaces V_n based on polynomial expansions or reduced bases perform significantly better than standard finite element spaces, at least for a moderate number d of parameters. It is possible to maintain a rate of convergence as d grows, and even when $d = \infty$, when assuming some anisotropy in the variable y_j through the decay of the size of ψ_j as $j \to \infty$, see in particular [14, 52, 54] for results of this type.

2.1.3 High constrast problems

The Uniform Ellipticity Assumption (2.6) implies that there is a uniform control on the level of contrast in the diffusion function

$$\kappa(y) := \frac{\max_{x\in\Omega} a(x,y)}{\min_{x\in\Omega} a(x,y)} \leqslant \frac{R}{r}, \quad y \in Y. \tag{2.9}$$

This assumption also plays a key role in the derivation of the above approximation results, since it guarantees that the parameter to solution map has a holomorphic extension to a sufficiently large complex neighbourhood of Y. In this case, a good polynomial approximation u_n may be defined by simply truncating the power series $\sum_{\nu\in\mathbb{N}^d} u_\nu y^\nu$, leading to the estimate (2.8).

On the other hand, there exist various situations where one would like to avoid such a strong restriction on the level of contrast. Perhaps the most representative setting is when the domain Ω is partitioned into disjoint subdomains $\{\Omega_1,\dots,\Omega_d\}$, each of them admitting a constant diffusivity level that could vary strongly between subdomains. This is typically the case when modelling diffusion in materials having multiple layers or inclusions that could have very different nature, for example air or liquid versus solid. This situation can be encountered in groundwater flow applications, where certain subdomains correspond to cavities, for which the diffusion function becomes nearly infinite, as opposed to subdomains containing sediments or other porous rocks.

In such a case, we do not want to limit the contrast level. To represent this setting, we let

$$a(y)_{|\Omega_j} = y_j, \quad y_j \in]0,\infty[\tag{2.10}$$

or equivalently $a(y) = \sum_{j=1}^d y_j \chi_{\Omega_j}$, which corresponds to the affine form (2.5) with $\bar{a} = 0$ and $\psi_j = \chi_{\Omega_j}$, now with

$$Y :=]0,\infty[^d. \tag{2.11}$$

We take (2.11) as the definition of the parameter domain Y for the remainder of this paper. The solution $u(y)$ satisfies the variational formulation

$$\sum_{j=1}^d y_j \int_{\Omega_j} \nabla u(y)\cdot\nabla v\, dx = \langle f, v\rangle_{H^{-1},H_0^1}, \quad v \in H_0^1(\Omega), \tag{2.12}$$

or equivalently $-y_j\Delta u(y) = f$ as elements of $H^{-1}(\Omega_j)$ on each Ω_j, with the standard jump conditions $[a(y)\partial_{\vec{n}} u(y)] = 0$ across the boundaries between subdomains.

Let us observe that in this setting, it is hopeless to find spaces V_n that approximate all solutions $u(y)$ uniformly well. Indeed, the following homogeneity property obviously holds: for any $y \in Y$ and $t > 0$, one has

$$u(ty) = t^{-1}u(y). \tag{2.13}$$

This property implies in particular that $\|u(y)\|_{H_0^1}$ tends to infinity as $y \to 0$, and so does

$\|u(y) - P_{V_n} u(y)\|_{H^1_0}$ in general. In fact, this also shows that the solution manifold $\mathcal{M}$ is *not* relatively compact and does not have finite n-widths.

In addition to this principal difficulty, let us remind that when using the spaces V_n in forward modelling, we typically use the Galerkin method, that delivers the orthogonal projection onto V_n however for the energy norm

$$\|v\|_y^2 := \sum_{j=1}^{d} y_j \int_{\Omega_j} |\nabla v|^2 \, dx. \tag{2.14}$$

This approximation is thus optimal in $H^1_0(\Omega)$, however up to the constant $\kappa(y)^{1/2}$, which deteriorates with high contrast.

The main contribution of this paper is to treat these issues, and derive approximation estimates that are robust to high contrast, in the sense that they are independent of $y \in Y$.

Due to the main objection coming from the homogeneity property (2.13), it is natural to look for uniform approximation estimates in relative error, that is, estimates of the form

$$\|u(y) - P_{V_n} u(y)\|_{H^1_0} \leqslant \varepsilon_n \|u(y)\|_{H^1_0}, \quad y \in Y, \tag{2.15}$$

with $\lim_{n\to\infty} \varepsilon_n = 0$, and similarly for $P^y_{V_n} u(y)$. Our main results, Theorems 2.3.7 and 2.4.2, exhibit spaces V_n ensuring the validity of such uniform estimates with ε_n having sub-exponential decay with n, similar to the known results under UEA.

Remark 2.1.1. *High contrast problems have been the object of intense investigation, in particular with the objective of developing techniques for multilevel or domain decomposition preconditioning [6, 5, 78] and a-posteriori error estimation [4, 25], that are provably robust with respect to the level of contrast. We also refer to [91, 126] for the treatment of high-contrast problems by multiscale methods, in the context of hetereogeneous media, see also [11]. To our knowledge, the present work is the first in which this robustness is established for reduced modelling methods in the context of parametrized coefficients.*

2.1.4 Outline

Throughout this paper, we consider the parametrized elliptic PDE (2.4) with $a(y)$ having piecewise constant form (2.10) over a fixed partition. In view of the homogeneity property (2.13), we are led to consider the subset

$$Y' := [1, \infty[^d \tag{2.16}$$

of parameters corresponding to the coercive regime. Any result on relative approximation error that is established for Y' extends automatically to all of Y because of the homogeneity property. Accordingly, we let

$$\mathcal{B} := \{u(y) \ : \ y \in Y'\}. \tag{2.17}$$

In Section 2.2, we start by proving that $\mathcal{B}$ is a precompact set of $H_0^1(\Omega)$. One crucial ingredient for this analysis are the *limit solutions* of the so-called *stiff problem*, obtained as $y_j \to \infty$ for certain $j \in \{1, \dots, d\}$.

In Section 2.3, we construct specific reduced model spaces for which the approximation estimate (2.15) holds with ε_n decaying sub-exponentially. Our construction is based on partitioning the parametric domain Y' into rectangular regions and using a different polynomial approximations on each region. This results in global reduced model space V_n for which the accuracy bound remains sub-exponential, however in $\exp(-cn^{\frac{1}{2d-2}})$. A key ingredient for establishing these sub-exponential rates is the derivation of quantitative estimates on the convergence of $u(y)$ towards limit solutions defined in Section 2.2 as some y_j tend to infinity. These estimates are established under an additional geometrical assumption on the partition, similar results for a general partition of Ω being an open problem.

In Section 2.4, we discuss the use of these reduced model spaces in forward modelling and inverse problems. Our main result relative to forward modelling is that the estimate (2.15) also holds for the Galerkin projection with the same exponential decay e_n. We show that such a result is only possible if V_n includes functions that have constant values over some subdomains. For the state estimation problem, we follow the Parametrized Background Data Weak (PBDW) method [26, 108], and obtain recovery bounds that are uniform over $y \in Y$ in relative error. For the parameter estimation problem, we introduce an ad-hoc strategy that specifically exploits the piecewise constant structure of the diffusion coefficient and obtain similar recovery bounds for the inverse diffusivity.

We conclude in Section 2.5 by presenting some numerical illustrations revealing the effectiveness of the reduced model spaces even in the high-contrast regime, as expressed by the approximation results.

Acknowledgements: We thank the anonymous reviewers for their constructive comments. We also thank François Murat for useful discussions in the understanding of the convergence process towards limit solutions, Hamza Maimoune for leading us to this work through his remarks during his master project, and Jules Pertinand for useful discussions.

2.2 Uniform approximation in relative error

In this section we work under no particular geometric assumption on the partition $\{\Omega_1, \dots, \Omega_d\}$ of Ω, and consider the solution manifold $\mathcal{M}$ defined by (2.1), where $u(y) \in H_0^1(\Omega)$ is solution to the elliptic boundary value problem with variational formulation (2.12). Our objective is to show the existence of spaces V_n that uniformly approximate $\mathcal{M}$ in the relative error sense expressed by (2.15).

2.2.1 Limit solutions and the extended solution manifold

Our first observation is that this collection can be continuously extended when $y_j = \infty$ for some values of j, through limit solutions of stiff inclusions problems. Such limit solutions have for example been considered in the context homogeneization, see e.g. p.98 of [96].

For this purpose, to any $S \subset \{1,\dots,d\}$, we associate the space

$$V_S := \{v \in H^1_0(\Omega) \ : \ \nabla v_{|\Omega_j} = 0, \ \ j \in S\}. \tag{2.18}$$

In other words, V_S consists of the functions from $H^1_0(\Omega)$ that have constant values on the subdomains Ω_j for $j \in S$ (or on each of their connected components if these subdomains are not connected). It is a closed subspace of $H^1_0(\Omega)$. We decompose the parameter vector y according to

$$y = (y_S, y_{S^c}), \quad y_S := (y_j)_{j\in S} \quad \text{and} \quad y_{S^c} := (y_j)_{j\in S^c}. \tag{2.19}$$

For any finite and positive vector y_{S^c}, similar to the $\|\cdot\|_y$ norm (2.14), we may define

$$\|v\|^2_{y_{S^c}} := \sum_{j\in S^c} y_j \int_{\Omega_j} |\nabla v|^2 \, dx, \tag{2.20}$$

which is a semi-norm on $H^1_0(\Omega)$, and a full norm equivalent to the H^1_0-norm on V_S. Also note that when $y = (y_S, y_{S^c})$ is finite, one then has $\|v\|_{y_{S^c}} = \|v\|_y$ for any $v \in V_S$.

For any finite and positive vector y_{S^c}, we define the function $u_S(y_{S^c}) \in V_S$ solution to the following stiff inclusions problem:

$$\sum_{j\in S^c} y_j \int_{\Omega_j} \nabla u_S(y_{S^c}) \cdot \nabla v \, dx = \langle f, v\rangle_{H^{-1},H^1_0}, \quad v \in V_S. \tag{2.21}$$

The following result shows that this solution is well defined and is the limit of $u(y)$, when y_{S^c} is fixed and $y_j \to \infty$ for $j \in S$. Note that the weak convergence is established in [96] (p. 98) and so we concentrate the proof on the strong convergence.

Lemma 2.2.1. *There exists a unique $u_S(y_{S^c}) \in V_S$ solution to (2.21), which is the limit in $H^1_0(\Omega)$ of the solution $u(y_S, y_{S^c})$ as $y_j \to \infty$ for all $j \in S$.*

Proof. Using the bilinear form $(u,v) \mapsto \sum_{j\in S^c} y_j \int_{\Omega_j} \nabla u \cdot \nabla v \, dx$ in the space V_S, Lax-Milgram theory implies the existence of a unique solution $u_S(y_{S^c}) \in V_S$ to (2.21).

Consider now a sequence $(y^n)_{n\geqslant 1} \in Y^{\mathbb{N}}$, with $y^n_{S^c} = y_{S^c}$ and $y^n_j \to \infty$ for all $j \in S$. Denoting $u_n = u(y^n)$, it is readily seen that $(u_n)_{n\geqslant 1}$ is uniformly bounded in H^1_0 norm by $C = C_f \, c^{-1}$, where $c := \min_{n\geqslant 1} \min_{1\leqslant j\leqslant d} y^n_j > 0$, and that any weak limit of a sequence extraction is solution to the variational (2.21). Therefore the whole sequence $(u_n)_{n\geqslant 1}$ weakly converges to $\bar u = u_S(y_{S^c})$.

We finally prove strong convergence by writing

$$\begin{aligned} c\|u_n - \bar u\|^2_{H^1_0} &\leqslant \int_\Omega a(y^n) |\nabla(u_n - \bar u)|^2 \, dx \\ &= \langle f, u_n\rangle_{H^{-1},H^1_0} - 2\langle \bar u, u_n\rangle_{y_{S^c}} + \|\bar u\|^2_{y_{S^c}} \\ &\underset{n\to\infty}{\longrightarrow} \langle f, \bar u\rangle_{H^{-1},H^1_0} - \|\bar u\|^2_{y_{S^c}} = 0. \end{aligned}$$

□

The above lemma allows us to readily extend the solution manifold by introducing

$$\widetilde{Y} :=]0, \infty]^d,$$

and

$$\overline{\mathcal{M}} := \{u(y) \ : \ y \in \widetilde{Y}\},$$

where we have formally set

$$u(y) := u_S(y_{S^c}),$$

when $y_j = \infty$ for $j \in S$ and $y_j < \infty$ for $j \in S^c$. Note that when $S = \{1, \dots, d\}$ the space V_S is trivial and one has

$$u(\infty, \dots, \infty) = 0.$$

Remark 2.2.2. *Although we do not make explicit use of it, it can be checked that despite the fact that $y_j = 0$ is excluded in the definition of $\overline{\mathcal{M}}$, it indeed coincides with the closure of $\mathcal{M}$ in $H^1_0(\Omega)$ due to the fact that $\|u(y)\|_{H^1_0} \to \infty$ as $y \to 0$.*

Remark 2.2.3. *More precisely, when some y_j tend to zero, $u(y)$ converges to the solution of the so-called soft inclusions problem (see [96], chapter 3), outside the corresponding subdomains Ω_j. Here, due to the fact that the approximation estimates that we prove further are in relative error, these other limit solutions are of no use in our analysis.*

2.2.2 A compactness result

As already observed in the introduction, the manifold $\overline{\mathcal{M}}$ is not bounded in $H^1_0(\Omega)$ due to the homogeneity property (2.13) and therefore not compact.

In order to treat this defect, we consider

$$\widetilde{Y}' := [1, \infty]^d,$$

and the submanifold

$$\overline{\mathcal{B}} := \{u(y) \ : \ y \in \widetilde{Y}'\},$$

which is now bounded in $H^1_0(\Omega)$, from the standard a-priori estimate

$$\|u(y)\|_{H^1_0} \leqslant \frac{C_f}{\min y_j} \leqslant C_f,$$

that is obtained by taking $v = u(y)$ in the variational formulation (2.12), with $C_f = \|f\|_{H^{-1}}$ as in (2.7). This estimate trivially extends to $u_S(y_{S^c})$ when the y_j have infinite value for $j \in S$. In addition we have the following result.

Theorem 2.2.4. *The set $\overline{\mathcal{B}}$ is compact in $H^1_0(\Omega)$.*

Proof. Consider any sequence of vectors $y^n = (y_1^n, \dots, y_d^n) \in \widetilde{Y}'$ for $n \geqslant 1$. We need to prove that the corresponding sequence of solutions $(u(y^n))_{n\geqslant 1}$ admits a converging subsequence. For this purpose, we observe that there exists a subset $S \in \{1, \dots, d\}$ such that, up to subsequence extraction,

$$\lim_{n\to\infty} y_j^n = \infty, \quad j \in S,$$

and

$$\lim_{n\to\infty} y_j^n = y_j < \infty, \quad j \in S^c.$$

Note that S could be empty, for instance in the case where the y_j^n are uniformly bounded for all j.

Let $\mathrm{e} > 0$. Using the strong convergence result in Lemma 2.2.1, for all $n \geqslant 1$ there exists an auxiliary vector $\bar{y}^n$ such that $\bar{y}_j^n = y_j^n$ when $y_j^n < \infty$, $\bar{y}_j^n < \infty$ when $y_j^n = \infty$, such that by having picked $\bar{y}_j^n$ large enough in the second case

$$\|u(y^n) - u(\bar{y}^n)\|_{H_0^1} \leqslant \mathrm{e}/3.$$

In addition we may assume that $\bar{y}_j^n \to \infty$ for $j \in S$. Next we introduce the vector $\tilde{y}^n$ such that $\tilde{y}_j^n = \bar{y}_j^n$ when $j \in S$ and $\tilde{y}_j^n = y_j$ when $j \in S^c$. Applying again Lemma 2.2.1, we find that with $y_{S^c} = (y_j)_{j\in S^c}$, one has

$$\|u(\tilde{y}^n) - u_S(y_{S^c})\|_{H_0^1} \leqslant \mathrm{e}/3,$$

for n sufficiently large. Finally we argue that

$$\|u(\tilde{y}^n) - u(\bar{y}^n)\|_{H_0^1} \leqslant \mathrm{e}/3,$$

for n large enough. This is a consequence of the following variant of Strang first lemma (which proof is similar and left as an exercise to the reader) that says that for two diffusion functions $\bar{a}$ and $\tilde{a}$, the corresponding solution $\bar{u}$ and $\tilde{u}$ with the same data f satisfy

$$\|\bar{u} - \tilde{u}\|_{H_0^1} \leqslant \frac{C_f \, \|\bar{a} - \tilde{a}\|_{L^\infty}}{\min\{\bar{a}_{\min}, \tilde{a}_{\min}\}^2}.$$

We then apply this to $\bar{a} := \bar{a}_n = a(\bar{y}^n)$ and $\tilde{a} := \tilde{a}_n = a(\tilde{y}^n)$, observing that from their definition, $\|\bar{a} - \tilde{a}\|_{L^\infty} = \max_{j\in S^c} |\bar{y}_j^n - y_j| \to 0$ as $n \to \infty$. Therefore $\|u(y^n) - u_S(y_{S^c})\|_{H_0^1} \leqslant \mathrm{e}$ for n sufficiently large, which concludes the proof. □

We next observe that any $y \in Y$ can be rewritten as

$$y = t\tilde{y},$$

with $\tilde{y} \in Y'$ and normalization $\min \tilde{y}_j = 1$, for some $t > 0$, and from (2.13) one has $u(y) = t^{-1}u(\tilde{y})$. This motivates the study of the further reduced manifold

$$\mathcal{N} := \{u(y) \ : \ y \in \widetilde{Y}', \ \min y_j = 1\}, \tag{2.22}$$

which is a subset of $\overline{\mathcal{B}}$.

One important observation is that the solutions contained in $\mathcal{N}$ are also uniformly bounded from below, under mild assumptions on the data f.

Lemma 2.2.5. *The set $\mathcal{N}$ is compact in $H^1_0(\Omega)$. Moreover, one has the framing*

$$\min_{1\leqslant j\leqslant d}\|f\|_{H^{-1}(\Omega_j)} \leqslant \|u(y)\|_{H^1_0} \leqslant C_f, \tag{2.23}$$

for all $u(y)\in\mathcal{N}$.

Proof. The compactness of $\mathcal{N}$ follows from that of $\overline{\mathcal{B}}$, since $\mathcal{N}$ is a closed subset of $\overline{\mathcal{B}}$. For the framing, as $a(y)\geqslant 1$ on Ω,

$$\|u\|^2_{H^1_0} \leqslant \sum_{j\in S^c} y_j \int_{\Omega_j} |\nabla u(y)|^2\,dx = \langle f, u(y)\rangle_{H^{-1},H^1_0} \leqslant C_f\|u(y)\|_{H^1_0},$$

so $\|u(y)\|_{H^1_0}\leqslant C_f$. Now take $j\in\{1,\dots,d\}$ such that $y_j=1$, and consider $\phi\in H^1_0(\Omega_j)$. Then

$$\langle f,\phi\rangle_{H^{-1},H^1_0} = \int_{\Omega_j}\nabla u(y)\cdot\nabla\phi\,dx \leqslant \|u(y)\|_{H^1_0(\Omega)}\|\phi\|_{H^1_0(\Omega_j)},$$

which gives the result. □

In the sequel of this paper, we always work under the condition that the lower bound in (2.23) is strictly positive

$$c_f := \min_{1\leqslant j\leqslant d}\|f\|_{H^{-1}(\Omega_j)} > 0. \tag{2.24}$$

Let us observe that when f is a function in $L^2(\Omega)$, this is ensured as soon as f is not identically zero on one of the Ω_j. We thus have

$$0 < c_f \leqslant \|u(y)\|_{H^1_0} \leqslant C_f, \tag{2.25}$$

for all $u(y)\in\mathcal{N}$.

Remark 2.2.6. *The condition $c_f>0$ is in general necessary for controlling $\|u(y)\|_{H^1_0}$ from below. Indeed assume $\|f\|_{H^{-1}(\Omega_j)}=0$ for some j such that $\overline{\Omega}\setminus\overline{\Omega}_j$ is connected. Then taking $y_k=\infty$ for $k\neq j$ and $y_j=1$, we find that $u(y)\in V_S$ with $S=\{j\}^c$, which is equivalent to $u(y)\in H^1_0(\Omega_j)$ since it vanishes on the other sub-domains. As $\|f\|_{H^{-1}(\Omega_j)}=0$, we obtain $u(y)=0$.*

Remark 2.2.7. *One also has the uniform framing in the $\|\cdot\|_y$ norm since*

$$0 < c_f \leqslant \|u(y)\|_{H^1_0} \leqslant \|u(y)\|_y = \sqrt{\langle f,u\rangle_{H^{-1},H^1_0}} \leqslant C_f, \tag{2.26}$$

for all $u(y)\in\mathcal{N}$ when all y_j are finite.

The framing (2.25) has an implication on the existence of reduced model spaces that approximate uniformly well all solutions $u(y) \in \overline{\mathcal{M}}$ in relative error.

Theorem 2.2.8. *There exists a sequence of linear spaces $(V_n)_{n\geqslant 1}$ such that $\dim(V_n) = n$, and a sequence $(\varepsilon_n)_{n\geqslant 1}$ that converges to zero such that*

$$\|u(y) - P_{V_n} u(y)\|_{H_0^1} \leqslant \varepsilon_n \|u(y)\|_{H_0^1} \tag{2.27}$$

for all $y \in \widetilde{Y}$, where P_{V_n} is the $H_0^1(\Omega)$-orthogonal projector onto V_n.

Proof. Since $\mathcal{N}$ is compact, there exists a sequence of spaces $(V_n)_{n\geqslant 1}$ with $\dim(V_n) = n$ and a sequence $(\sigma_n)_{n\geqslant 1}$ that tends to 0, such that

$$\|v - P_{V_n} v\|_{H_0^1} \leqslant \sigma_n, \quad v \in \mathcal{N}.$$

Now let $y \in \widetilde{Y}$ differing from $(\infty, \dots, \infty)$, for which there is nothing to prove since $u(\infty, \dots, \infty) = 0$, and let $t^{-1} = \min_{1\leqslant j\leqslant d} y_j < \infty$. By homogeneity, $t^{-1} u(y) = u(ty) \in \mathcal{N}$, and therefore

$$\|u(y) - P_{V_n} u(y)\|_{H_0^1} = t \|u(ty) - P_{V_n} u(ty)\|_{H_0^1(\Omega)} \leqslant t\sigma_n.$$

On the other hand, $\|u(y)\|_{H_0^1(\Omega)} = t\|u(ty)\|_{H_0^1(\Omega)} \geqslant t c_f$ by framing (2.23), which proves Theorem 2.2.8 with $\varepsilon_n = \sigma_n / c_f$. □

The above theorem tells us that we can achieve contrast-independent approximation in relative error. It is however still unsatisfactory from two perspectives:

1. It does not describe the rate of decay of ε_n as the reduced dimension n grows. In practice, one would like to construct reduced spaces V_n such that this decay is fast, similar to the exponential decay obtained under UEA.

2. The approximation property is expressed in terms of the orthogonal projection P_{V_n}. In applications to forward modelling, we approximate the solution $u(y)$ in the space V_n by the Galerkin projection $P_{V_n}^y u(y)$. We thus wish for uniform estimates also for such approximations.

These two problems are treated in Section 2.3 and Section 2.4 respectively.

2.3 Approximation rates

Our construction of efficient reduced model spaces is based on a certain partitioning of the parameter domain $\widetilde{Y}'$ associated to the manifold $\overline{\mathcal{B}}$. To any $\ell = (\ell_1, \dots, \ell_d) \in \mathbb{N}_0^d$ we associate the dyadic rectangle

$$R_\ell = [2^{\ell_1}, 2^{\ell_1+1}] \times \cdots \times [2^{\ell_d}, 2^{\ell_d+1}], \tag{2.28}$$

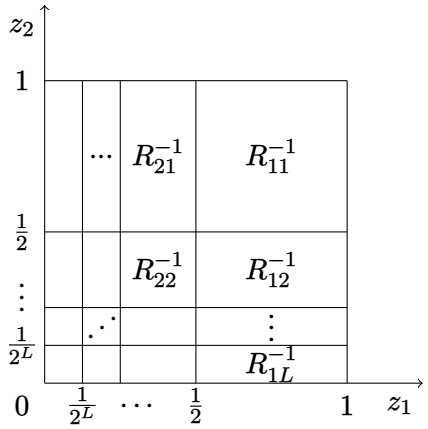


Figure 2.1: Partition of $[0,1]^d$ by the inverse rectangles R_ℓ^{-1} in the case $d=2$.

For a positive integer L to be fixed further, we modify the definition of R_ℓ by replacing the interval $[2^{\ell_j}, 2^{\ell_j+1}]$ by $[2^{\ell_j}, \infty]$ when $\ell_j = L$ for some j. This leads to the partition

$$\widetilde{Y}' = \bigcup_{\ell \in \{0,\dots,L\}^d} R_\ell. \tag{2.29}$$

This partition is best visualized in the inverse parameter domain by setting

$$z = (z_1, \dots, z_d) := (y_1^{-1}, \dots, y_d^{-1}) \in [0,1]^d. \tag{2.30}$$

Then, the inverse rectangles R_ℓ^{-1} split the unit cube, as shown on Figure 2.1. In particular, the rectangles touching the axes correspond to rectangles R_ℓ of infinite size.

We build reduced model spaces through a piecewise polynomial approximation over this partition. In other words, for each $\ell \in \{0,\dots,L\}^d$, we use different polynomials

$$u_{\ell,k}(y) = \sum_{|\nu| \leqslant k} u_{\ell,\nu} y^\nu,$$

of total degree k for approximating $u(y)$ when $y \in R_\ell$, leading to a family of local reduced model spaces

$$V_{\ell,k} = \operatorname{span}\{u_{\ell,\nu} \ : \ |\nu| \leqslant k\}, \tag{2.31}$$

that can be either used individually when approximating $u(y)$ if the rectangle R_ℓ containing y is known, or summed up in order to obtain a global reduced model space.

In this section we show that this construction yields exponential convergence rates in (2.15), similar to those obtained under a Uniform Ellipticity Assumption. This requires a proper tuning between the total polynomial degree k and the integer L that determines the size of the partition. In the study of local polynomial approximation, we treat separately the inner rectangles for which $\ell \in \{0,\cdots,L-1\}^d$ and the infinite rectangles for which one

or several ℓ_j are equal to L. The estimates obtained in the latter case rely on the additional assumption that the partition has a geometry of disjoint inclusions.

2.3.1 Polynomial approximation on inner rectangles

Inner rectangles R_ℓ are particular cases of rectangles of the form

$$R = [a_1, 2a_1] \times \cdots \times [a_d, 2a_d], \tag{2.32}$$

for some $a_j \geqslant 1$. The following lemma, adapted from [13], shows that one can approximate the parameter to solution map in the $\|\cdot\|_y$ and $\|\cdot\|_{H^1_0}$ norms on such rectangles, with a rate that decreases exponentially in the total polynomial degree.

Lemma 2.3.1. *Let R be any rectangle of the form (2.32). Then, for each $k \geqslant 0$, there exists functions $u_\nu \in H^1_0(\Omega)$ such that*

$$\Big\| u(y) - \sum_{|\nu| \leqslant k} u_\nu y^\nu \Big\|_y \leqslant C 3^{-k}, \quad y \in R, \tag{2.33}$$

where $C := \frac{1}{\sqrt{3}} C_f$, and

$$\Big\| u(y) - \sum_{|\nu| \leqslant k} u_\nu y^\nu \Big\|_{H^1_0} \leqslant C 3^{-k}, \quad y \in R, \tag{2.34}$$

where $C := \frac{1}{\sqrt{6}} C_f$.

Proof. The exponential rate is established in [13] for a single parameter domain with uniform ellipticity assumption. Here the difficulty lies in the fact that we want the same estimate for all parametric rectangles R and thus without control on the uniform ellipticity. Still the technique of proof, based on power series, is similar.

The elliptic equation $-\operatorname{div}(a(y)u(y)) = f$ may be written in operator form

$$A_y u(y) = f,$$

where the invertible operator $A_y : H^1_0(\Omega) \to H^{-1}(\Omega)$ is defined by

$$\langle A_y v, w \rangle_{H^{-1}, H^1_0} := \int a(y) \nabla v \cdot \nabla w \, dx = \langle v, w \rangle_y.$$

We introduce

$$\overline{y} := \frac{3}{2}(a_1, \ldots, a_d),$$

the center of the rectangle, and write any $y \in R$ as

$$y = \overline{y} + \tilde{y},$$

where the components $\tilde{y}_j$ of $\tilde{y}$ vary in $[-a_j/2, a_j/2]$. We may write $A_y = A_{\overline{y}} + \sum_{j=1}^d \tilde{y}_j A_j$, where the operators $A_j : H_0^1(\Omega) \to H^{-1}(\Omega)$ are defined by

$$\langle A_j v, w\rangle_{H^{-1},H_0^1} := \int_{\Omega_j} \nabla v \cdot \nabla w \, dx.$$

This allows us to rewrite the equation as

$$(I + B(\tilde{y}))u(y) = g,$$

where $g := A_{\overline{y}}^{-1} f \in H_0^1(\Omega)$ and $B(\tilde{y}) = \sum_{j=1}^d \tilde{y}_j A_{\overline{y}}^{-1} A_j$ acts in $H_0^1(\Omega)$. We then observe that

$$\langle B(\tilde{y})v, w\rangle_{\overline{y}} = \langle A_{\overline{y}} B(\tilde{y})v, w\rangle_{H^{-1},H_0^1} = \sum_{j=1}^d \tilde{y}_j \langle A_j v, w\rangle_{H^{-1},H_0^1} = \sum_{j=1}^d \tilde{y}_j \int_{\Omega_j} \nabla v \cdot \nabla w \, dx,$$

and therefore, since $|\tilde{y}_j| \leqslant \frac{1}{3}\overline{y}_j$,

$$|\langle B(\tilde{y})v, w\rangle_{\overline{y}}| \leqslant \frac{1}{3}\sum_{j=1}^d \overline{y}_j \Big| \int_{\Omega_j} \nabla v \cdot \nabla w \, dx \Big| \leqslant \frac{1}{3}\|v\|_{\overline{y}}\|w\|_{\overline{y}},$$

which shows that $\|B(\tilde{y})\|_{\overline{y}\to\overline{y}} \leqslant \frac{1}{3}$. We may thus approximate $(I + B(\tilde{y}))^{-1}$ by the partial Neumann series

$$\sum_{l=0}^k (-1)^l B(\tilde{y})^l,$$

which is a polynomial in $\tilde{y}$ of total degree k. The corresponding polynomial approximation to $u(y)$ is given by

$$N_k u(y) = \sum_{l=0}^k (-1)^l B(\tilde{y})^l g = \sum_{l=0}^k (-1)^l \left(\sum_{j=1}^d \tilde{y}_j A_{\overline{y}}^{-1} A_j \right)^l g = \sum_{|\nu|\leqslant k} v_\nu \tilde{y}^\nu,$$

and coincides with the truncated power series of $\tilde{u}(\tilde{y}) := u(\overline{y} + \tilde{y})$ at $\tilde{y} = 0$, that is,

$$v_\nu := \frac{1}{\nu!}\partial^\nu u(\overline{y}), \qquad \nu! := \prod \nu_j!.$$

It can be rewritten in the form

$$N_k u(y) = \sum_{|\nu|\leqslant k} u_\nu y^\nu.$$

One has

$$\|u(y) - N_k u(y)\|_{\overline{y}} \leqslant \sum_{l>k} \|B(\tilde{y})^l g\|_{\overline{y}} \leqslant \Big(\sum_{l>k} 3^{-l}\Big)\|A_{\overline{y}}^{-1} f\|_{\overline{y}} = \frac{3^{-k}}{2}\|A_{\overline{y}}^{-1} f\|_{\overline{y}},$$

and

$$\|A_{\overline{y}}^{-1} f\|_{\overline{y}}^2 = \langle A_{\overline{y}} A_{\overline{y}}^{-1} f, A_{\overline{y}}^{-1} f\rangle_{H^{-1},H_0^1} = \langle f, u(\overline{y})\rangle_{H^{-1},H_0^1} \leqslant C_f \|u(\overline{y})\|_{H_0^1} \leqslant C_f^2,$$

where the last inequality follows from Lax-Milgram estimate since $a(\overline{y}) \geqslant 1$. This proves the estimate

$$\Big\|u(y) - \sum_{|\nu|\leqslant k} u_\nu y^\nu\Big\|_{\overline{y}} \leqslant C3^{-k}, \quad y \in R, \tag{2.35}$$

with $C := \frac{1}{2}C_f$. Using the inequalities

$$\|v\|_y^2 \leqslant \frac{4}{3}\|v\|_{\overline{y}}^2, \quad v \in H_0^1(\Omega),\ y \in R,$$

and

$$\|v\|_{H_0^1}^2 \leqslant \frac{2}{3}\|v\|_{\overline{y}}^2, \quad v \in H_0^1(\Omega),$$

we obtain the estimate (2.33) and (2.34) with the modified multiplicative constants. □

Remark 2.3.2. *The above lemma shows that the set $\mathcal{M}_R := \{u(y) \ : \ y \in R\}$ can be approximated with accuracy $C3^{-k}$ by the space*

$$V_R := \operatorname{span}\{u_\nu \ : \ |\nu| \leqslant k\}. \tag{2.36}$$

The dimension of V_R is at most $\binom{k+d}{d}$, however, as noticed in [13], it can in fact be seen that

$$\dim(V_R) \leqslant \binom{k+d-1}{d-1}. \tag{2.37}$$

This stems from the fact that the operators defined in the above proof satisfy the dependency relation

$$A_{\overline{y}} = \sum_{j=1}^{d} \overline{y}_j A_j,$$

and therefore, one can rewrite A_y as

$$A_y := (1 + \tilde{y}_d/\overline{y}_d)A_{\overline{y}} + \sum_{j=1}^{d-1}(\tilde{y}_j - \tilde{y}_d\overline{y}_j/\overline{y}_d)A_j.$$

Using this form, the partial Neumann sum $N_k u(y)$ has at most $\binom{k+d-1}{d-1}$ independent terms.

We shall also make use of the following adaptation of the above lemma to the approxi-

mation of the limit solution map $y_{S^c} \mapsto u_S(y_{S^c})$, defined by (2.21). Its proof is an immediate adaptation of the previous one and is therefore omitted.

Lemma 2.3.3. *Let $S \subset \{1, \dots, d\}$, and for some $a_j \geqslant 1$, let R be a rectangle of the form*

$$R = \prod_{j \in S^c} [a_j, 2a_j]. \tag{2.38}$$

Then, there exists functions $u_\nu \in V_S$ such that

$$\left\| u_S(y_{S^c}) - \sum_{|\nu| \leqslant k} u_\nu y_{S^c}^\nu \right\|_{y_{S^c}} \leqslant C 3^{-k}, \quad y_{S^c} \in R, \tag{2.39}$$

where $C := \frac{1}{\sqrt{3}} C_f$, and

$$\left\| u_S(y_{S^c}) - \sum_{|\nu| \leqslant k} u_\nu y_{S^c} \right\|_{H_0^1} \leqslant C 3^{-k}, \quad y_{S^c} \in R, \tag{2.40}$$

where $C := \frac{1}{\sqrt{6}} C_f$.

2.3.2 Polynomial approximation on infinite rectangles

We now consider the infinite rectangles R_ℓ, corresponding to the ℓ such that some of the ℓ_j equal L. We define

$$S := \{j \ : \ \ell_j = L\}, \tag{2.41}$$

the set of such indices. When $y \in R_\ell$, we thus have

$$y_j \geqslant 2^L, \quad j \in S,$$

and so $u(y)$ should be close to $u_S(y_{S^c})$ as L is large. On the other hand y_{S^c} belongs to a rectangle of the form

$$R_{\ell_{S^c}} = \prod_{j \in S^c} [2^{\ell_j}, 2^{\ell_j + 1}].$$

Therefore, by Lemma 2.3.3, we can approximate $u_S(y_{S^c})$ by a polynomial of total degree k in these restricted variables.

In order to conclude that this polynomial is a good approximation to $u(y)$ on R_ℓ, we need a quantitative estimate on the convergence of $u(y)$ towards $u_S(y_{S^c})$. Let us observe that since

$$\sum_{j=1}^d y_j \int_{\Omega_j} \nabla u(y) \cdot \nabla v \, dx = \langle f, v \rangle_{H^{-1}, H_0^1} = \sum_{j \in S^c} y_j \int_{\Omega_j} \nabla u_S(y_{S^c}) \cdot \nabla v \, dx, \quad v \in V_S,$$

the function $u_S(y_{S^c})$ coincides with the orthogonal projection of $u(y)$ onto V_S for the y-norm,

as well as for the y_{S^c}-norm:

$$u_S(y_{S^c}) = P_{V_S}^{y} u(y) = P_{V_S}^{y_{S^c}} u(y). \tag{2.42}$$

In addition, with

$$\Omega_S := \bigcup_{j \in S} \Omega_j, \tag{2.43}$$

we have

$$2^L \|\nabla u(y)\|_{L^2(\Omega_S)}^2 \leqslant \sum_{j \in S} y_j \int_{\Omega_j} |\nabla u(y)|^2 \, dx \leqslant \langle f, u(y) \rangle_{H^{-1}, H_0^1} \leqslant C_f^2,$$

since $\|u(y)\|_{H_0^1} \leqslant C_f$, and therefore, since $\nabla u_S(y_{S^c}) = 0$ on Ω_S, we find that

$$\|\nabla u(y) - \nabla u_S(y_{S^c})\|_{L^2(\Omega_S)} \leqslant C_f 2^{-L/2}. \tag{2.44}$$

Our objective is to obtain a similar error bound on the remaining domains Ω_j for $j \in S^c$. This turns out to be feasible, with an even better rate 2^{-L}, when making certain geometric assumptions on the partition of the domain Ω.

Definition 2.3.4. *We say that $\{\Omega_1, \dots, \Omega_d\}$ is a* Lipschitz partition *if and only if for any subset $T \subset \{1, \dots, d\}$, the domain $\Omega_T = \bigcup_{j \in T} \Omega_j$ has Lipschitz boundaries.*

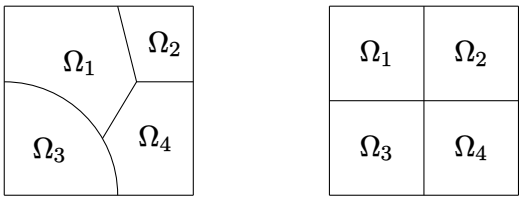


Figure 2.2: A Lipschitz partition of Ω (left) and a counter-example (right) since $\Omega_1 \cup \Omega_4$ is not Lipschitz.

Note that such a property is stronger than just saying that each domain is Lipschitz, see Figure 2.2 (right) for a counter-example. In a Lipschitz partition, all subdomains Ω_j are Lipschitz, and the common boundary between two subdomains is either empty or a $(n-1)$-dimensional surface, as illustrated on Figure 2.2 (left). In particular, it is easily checked that partitions consisting of a background domain and well separated subdomains that have Lipschitz boundaries fall in this category. Similar to the Ω_T, the individual Ω_j could have several connected components, that should then be well separated. Here by "well separated", we mean that δ-neighbourhoods of the subdomains remain disjoints for some $\delta > 0$.

For the inner domains Ω_T such that $\partial\Omega_T \cap \partial\Omega = \emptyset$, the classical Stein's extension theorem [147] guarantees the existence of continuous extension operators

$$E_T : H^1(\Omega_T) \to H^1(\Omega),$$

that satisfy $(E_T v)_{|\Omega_T} = v$ for all $v \in H^1(\Omega_T)$. We refer to chapter 5 of [2] for a relatively simple construction of the extension operator E_j by local reflection after using a partitioning of unity along the boundary of Ω_T and local transformations mapping the boundary to the hyperplane $\mathbb{R}^{n-1}$.

For the domains Ω_T touching the boundary $\partial\Omega$, these operators are modified in order to take into account the homogeneous boundary condition, and we refer to [159] for such adaptations. Here, the relevant space is

$$\tilde{H}^1(\Omega_T) := R_T(H^1_0(\Omega)), \tag{2.45}$$

where R_T is the restriction to Ω_T, over which $v \mapsto \|\nabla v\|_{L^2(\Omega_T)}$ is equivalent to the H^1 norm by Poincaré inequality. Then, there exists a continuous extension operator

$$E_T : \tilde{H}^1(\Omega_T) \to H^1_0(\Omega).$$

Note that the norm of all these operators depends on the geometry of the partition. These operators are instrumental in proving the following convergence estimate.

Lemma 2.3.5. *Assume that $\{\Omega_1, \dots, \Omega_d\}$ is a Lipschitz partition of Ω. Then there exists a constant C_0 that only depends on the geometry of the partition such that for any $S \subset \{1, \dots, d\}$ and $y = (y_S, y_{S^c}) \in Y'$, one has*

$$\|u(y) - u_S(y_{S^c})\|_{H^1_0} \leqslant C_0 C_f \max_{j \in S} y_j^{-1}. \tag{2.46}$$

In particular, for the infinite rectangle R_ℓ,

$$\|u(y) - u_S(y_{S^c})\|_{H^1_0} \leqslant C_0 C_f 2^{-L}, \quad y \in R_\ell, \tag{2.47}$$

with S defined by (2.41).

Proof. We first note that it suffices to prove (2.46) in the particular case where the largest y_j are those for which $j \in S$. Indeed, if this is not the case, we use the decomposition

$$u(y) - u_S(y_{S^c}) = (u(y) - u_{S'}(y_{S'^c})) - (u(y') - u_{S'}(y_{S'^c})) + (u(y') - u_S(y_{S^c})),$$

with $S' = \{i \ : \ y_i \geqslant \min_{j \in S} y_j\}$ and y' defined by $y'_j = \max_{i=1,\dots,d} y_i$ if $j \in S$, $y'_j = y_j$ otherwise, so that each term falls in this particular case and will be bounded in H^1_0 norm by $C_0 C_f \max_{j \in S} y_j^{-1}$. This leads to the same estimate (2.46) up to a factor 3 in constant C_0. In addition, up to reordering the subdomains Ω_j, we may assume $y_1 \geqslant \ldots \geqslant y_d$ and therefore $S = \{1, \dots, |S|\}$.

Fix $j \geqslant |S|$, and denote $u = u(y)$ and $u_S = u_S(y_{S^c})$ for simplicity. We define the Lipschitz domain $\Omega^j = \overline{\Omega}_1 \cup \cdots \cup \overline{\Omega}_j$, remarking that

$$\Omega_S = \bigcup_{j \in S} \Omega_j = \Omega^{|S|}.$$

Poincaré's inequality ensures that there exists a function c on Ω^j, constant on any connected component of Ω^j, and null on $\partial\Omega \cap \Omega^j$, such that

$$\|u - u_S - c\|_{H^1(\Omega^j)} \leqslant C_P \|\nabla(u - u_S)\|_{L^2(\Omega^j)},$$

with C_P the maximal Poincaré constant of all unions of subdomains from the partition. Moreover, there is an extension $v \in H^1_0(\Omega)$ of $u - u_S - c \in \tilde{H}^1(\Omega^j)$ such that

$$\|v\|_{H^1_0(\Omega)} \leqslant C_E \|u - u_S - c\|_{H^1(\Omega^j)} \leqslant C_E C_P \|\nabla(u - u_S)\|_{L^2(\Omega^j)},$$

with C_E the maximal norm of all extension operators E_T, $T \subset \{1, \dots, d\}$.

As $u - u_S - v = c$ on $\Omega_S \subset \Omega^j$, the function $u - u_S - v$ is in V_S, and therefore orthogonal to $u - u_S = u - P^y_{V_S} u$ for the $\|\cdot\|_y$ norm:

$$\begin{aligned}
0 &= \langle u - u_S, u - u_S - v \rangle_y \\
&= \sum_{i=1}^d y_i \int_{\Omega_i} |\nabla(u - u_S)|^2 - \sum_{i=1}^d y_i \int_{\Omega_i} \nabla(u - u_S) \cdot \nabla v \\
&= \sum_{i>j} y_i \int_{\Omega_i} |\nabla(u - u_S)|^2 - \sum_{i>j} y_i \int_{\Omega_i} \nabla(u - u_S) \cdot \nabla v
\end{aligned}$$

since $\nabla v = \nabla(u - u_S)$ on Ω^j. In particular, we obtain

$$\begin{aligned}
y_{j+1} \|\nabla(u - u_S)\|^2_{L^2(\Omega_{j+1})} &\leqslant \sum_{i>j} y_i \int_{\Omega_i} |\nabla(u - u_S)|^2 \\
&\leqslant y_{j+1} \int_{\Omega \setminus \Omega^j} |\nabla(u - u_S) \cdot \nabla v| \\
&\leqslant y_{j+1} \|u - u_S\|_{H^1_0(\Omega)} \|v\|_{H^1_0(\Omega)} \\
&\leqslant y_{j+1} \|u - u_S\|_{H^1_0(\Omega)} C_P C_E \|\nabla(u - u_S)\|_{L^2(\Omega^j)},
\end{aligned}$$

and therefore

$$\|\nabla(u - u_S)\|^2_{L^2(\Omega^{j+1})} \leqslant (1 + C_P C_E) \|\nabla(u - u_S)\|_{L^2(\Omega)} \|\nabla(u - u_S)\|_{L^2(\Omega^j)}.$$

Applying this inequality inductively for $j = d-1, \dots, d-k$, we get

$$\|\nabla(u - u_S)\|_{L^2(\Omega)} \leqslant (1 + C_P C_E)^{2^k - 1} \|\nabla(u - u_S)\|_{L^2(\Omega^{d-k})},$$

for any $k = 1, \dots, d - |S|$. For $k = d - |S|$, this results in the bound

$$\|\nabla(u - u_S)\|^2_{L^2(\Omega)} \leqslant C_0 \|\nabla(u - u_S)\|^2_{L^2(\Omega_S)} = C_0 \|\nabla u\|^2_{L^2(\Omega_S)}, \tag{2.48}$$

for any non-empty S, with $C_0 = (1 + C_P C_E)^{2^{d-1}}$.

We now write

$$\begin{aligned}(\min_{i\in S} y_i)\|\nabla(u-u_S)\|^2_{L^2(\Omega_S)} \leqslant \|u-u_S\|_y^2 &= \langle u, u-2u_S\rangle_y + \langle u_S, u_S\rangle_{y_{S^c}} \\ &= \langle f, u-u_S\rangle_{H^{-1},H^1_0} \leqslant C_f\|\nabla(u-u_S)\|_{L^2(\Omega)},\end{aligned}$$

which, combined to the previous estimate, gives

$$\|u-u_S\|_{H^1_0} = \|\nabla(u-u_S)\|_{L^2(\Omega)} \leqslant C_0 C_f \max_{i\in S} y_i^{-1},$$

therefore proving (2.46). For (2.47), we simply notice that $\max_{j\in S} y_j^{-1} \leqslant 2^{-L}$ for $y \in Y' \cap R_\ell$, and use a continuity argument when y takes infinite values. □

Combining the estimate (2.47) from the above lemma with (2.40) from Lemma 2.3.3, we obtain the following estimate for polynomial approximation on an infinite rectangle R_ℓ:

$$\Big\|u(y) - \sum_{|\nu|\leqslant k} u_\nu y_{S^c}^\nu\Big\|_{H^1_0} \leqslant \frac{C_f}{\sqrt{6}}\,3^{-k} + C_0 C_f 2^{-L}, \quad y \in R_\ell, \tag{2.49}$$

where C_0 is the constant in (2.47). This estimate hints how the level L in the partition should be tuned to the total polynomial degree k, so that the two contributions in the above estimate are of the same order.

Remark 2.3.6. *Note that the constant $C_0 = (1+C_P C_E)^{2^{d-1}}$ becomes prohibitive even for moderate values of d. However, under more restrictive geometric assumptions, for instance if the subdomains $\overline{\Omega}_2, \dots, \overline{\Omega}_d$ are disjoint inclusions in a background Ω_1, better bounds can be obtained, with a constant C_0 that does not suffer a similar curse of dimensionality, by replacing the induction in the proof by a two-step procedure, consisting of extensions first from the high-diffusivity inclusions to the background, and then to the whole domain Ω.*

2.3.3 Approximation rates and n-widths

We are now in position to establish an approximation result for the reduced model spaces. For this purpose, we fix the smallest level $L = L_k \geqslant 1$ such that

$$C_0 C_f 2^{-L} \leqslant \frac{C_f}{\sqrt{3}}\,3^{-k}.$$

In particular L scales linearly with k, with the bound $\alpha k + \beta \leqslant L_k \leqslant \alpha k + \gamma$, where

$$\alpha := \frac{\ln 3}{\ln 2}, \quad \beta := \frac{\ln(\sqrt{3}C_0)}{\ln 2}, \quad \gamma := \frac{\ln(2\sqrt{3}C_0)}{\ln 2}. \tag{2.50}$$

Then, the polynomial approximation estimates (2.34) and (2.49) show that for each $\ell \in \{0, \cdots, L_k\}^d$, there exist functions $u_{\ell,\nu} \in H_0^1(\Omega)$ such that

$$\left\| u(y) - \sum_{|\nu| \leqslant k} u_{\ell,\nu} y^\nu \right\|_{H_0^1} \leqslant \left(\frac{C_f}{\sqrt{6}} + \frac{C_f}{\sqrt{3}} \right) 3^{-k} \leqslant C_f 3^{-k}, \quad y \in R_\ell.$$

Note that in the case of an infinite rectangle R_ℓ, the $u_{\ell,\nu}$ are non trivial only for monomials of the form $y^\nu_{S^c}$ and they belong to V_S, where $S := \{j \ : \ \ell_j = L_k\}$.

Thus the solutions $u(y)$ for $y \in R_\ell$ are approximated with accuracy $C_f 3^{-k}$ in the space

$$V_{\ell,k} := \operatorname{span}\{u_{\ell,\nu} \ : \ |\nu| \leqslant k\},$$

which in view of (2.3.2) has dimension at most $\binom{k+d-1}{d-1}$.

Note also that approximating the reduced manifold $\mathcal{N}$ defined in (2.22) requires a smaller subset of rectangles, since

$$\{y \in \widetilde{Y}' \ : \ \min y_j = 1\} \subset \bigcup_{\ell \in E_k} R_\ell, \qquad E_k := \{0, \cdots, L_k\}^d \setminus \{1, \cdots, L_k\}^d.$$

We thus introduce the reduced model space

$$V_n := \bigoplus_{\ell \in E_k} V_{\ell,k}, \quad n = \dim(V_n) \leqslant \#(E_k) \binom{k+d-1}{d-1}, \tag{2.51}$$

and find that

$$\|u(y) - P_{V_n} u(y)\|_{H_0^1} \leqslant C_f 3^{-k}, \tag{2.52}$$

for all $y \in \widetilde{Y}'$ such that $\min y_j = 1$. In view of (2.50), there exists a constant C that depends on d and C_0, such that

$$n \leqslant ((L_k+1)^d - L_k^d) \binom{k+d-1}{d-1} \leqslant C(k+1)^{2d-2}. \tag{2.53}$$

This leads to the following approximation theorem.

Theorem 2.3.7. *Assume that the partition has the geometry of disjoint inclusions. The reduced basis space V_n defined in (2.51) then satisfies*

$$\|u(y) - P_{V_n} u(y)\|_{H_0^1} \leqslant C \exp\left(-cn^{\frac{1}{2d-2}} \right), \tag{2.54}$$

for all $y \in \widetilde{Y}' = [1, \infty]^d$ such that $\min y_j = 1$. The Kolmogorov n-width (2.2) of the reduced manifold $\mathcal{N}$ satisfies

$$d_n(\mathcal{N})_{H_0^1} \leqslant C \exp\left(-cn^{\frac{1}{2d-2}} \right). \tag{2.55}$$

Over the full manifold $\overline{\mathcal{M}}$, one has the estimate in relative error

$$\|u(y) - P_{V_n} u(y)\|_{H_0^1} \leqslant C \exp\left(-cn^{\frac{1}{2d-2}}\right) \|u(y)\|_{H_0^1}, \tag{2.56}$$

for all $y \in \widetilde{Y} =]0,\infty[^d$. The positive constants c and C only depend on d, C_f, and on the geometry of the partition through the constant C_0.

Proof. The estimate (2.54) follows directly by combining (2.52) and (2.53), and (2.55) is an immediate consequence. We then derive (2.56) by using the homogeneity property (2.13) and the lower inequality in (2.25), similar to the proof of (2.27) in Theorem 2.2.8. □

Remark 2.3.8. *In the above construction of V_n, the dimension n only takes the values $n_k := \#(E_k)\binom{k+d-1}{d-1}$ for $k \geqslant 0$. However it is easily seen that if we set $V_n = V_{n_k}$ for $n_k \leqslant n < n_{k+1}$, then all the estimates in the above theorem remain valid up to a change in the constants (c, C).*

Remark 2.3.9. *Note that the union of the $V_{\ell,k}$ for $\ell \in E_k$ would suffice to approximate $\mathcal{N}$ with uniform accuracy $C_f 3^{-k}$, their sum V_n is an overkill. When y is known, for example in forward modelling, it is therefore possible to first identify the proper space $V_{\ell,k}$ associated to the rectangle R_ℓ that contains y, and build the approximation to $u(y)$ from this space. This nonlinear reduced modelling strategy has been studied in [33] with similar local polynomial approximation under UEA, and in [71, 109, 161] with local reduced basis. The natural benchmark is given by the notion of library width introduced in [148], that is defined for any compact set $\mathcal{K}$ in a Banach space V as*

$$d_{n,N}(\mathcal{K})_V := \inf_{\#(\mathcal{L}_n)\leqslant N} \sup_{u\in\mathcal{K}} \min_{V_n\in\mathcal{L}_n} \min_{v\in V_n} \|u - v\|_V, \tag{2.57}$$

where the first infimum is taken over all libraries $\mathcal{L}_n$ of n-dimensional spaces with cardinality at most N. Our results thus show that

$$d_{n,N}(\mathcal{N})_{H_0^1} \leqslant C_f 3^{-k} \sim C\exp(-cn^{\frac{1}{d}}), \quad n := \binom{k+d-1}{d-1}, \; N = (L_k+1)^d - L_k^d.$$

Note that the above sub-exponential rate can be misleading due to fact that the constant c has a hidden dependence in d. As an example, up to the constant C_f, we find that taking $k = 4, 7, 9$ leads to error bounds 3^{-k} of order $10^{-2}, 10^{-3}, 10^{-4}$, with $n = 15, 36, 55$ for $d = 3$, and $n = 35, 120, 220$ for $d = 4$, which is far better than the value of $\exp(-n^{\frac{1}{d}})$.

Remark 2.3.10. *In view of the results from [28] and [65], we are ensured that a proper selection of reduced basis elements in the manifold $\mathcal{N}$ should generate spaces V_n that perform at least with the same exponential rates as those achieved by the spaces V_n in Theorem 2.3.7. As explained in the introduction, reduced basis spaces may perform significantly better than reduced model spaces based on polynomial or piecewise polynomial approximation. This occurs in particular when the polynomial coefficients have certain linear dependency, as established in [13] for the elliptic problem with piecewise constant coefficients in the low*

contrast regime, and recalled in Remark 3.2. There, it is shown that the rate $\mathcal{O}(\exp(-cn^{\frac{1}{d}}))$ *is at least improved to* $\mathcal{O}(\exp(-cn^{\frac{1}{d-1}}))$ *and that further improvements in the rate may result from certain symmetry properties of the domain partition, however not circumventing the curse of dimensionality. While we do not pursue this analysis in the present high contrast setting, we expect similar results to hold.*

2.4 Forward modelling and inverse problems

2.4.1 Galerkin projection

In the context of forward modelling, the reduced model space V_n is used to approximate the parameter to solution map, by a map

$$y \mapsto u_n(y) \in V_n,$$

computed through the Galerkin method: $u_n(y) \in V_n$ is such that

$$\sum_{j=1}^{d} y_j \int_{\Omega_j} \nabla u_n(y) \cdot \nabla v \, dx = \langle f, v \rangle_{H^{-1},H_0^1}, \quad v \in V_n.$$

Therefore $\langle u_n(y), v \rangle_y = \langle u(y), v \rangle_y$, that is

$$u_n(y) = P_{V_n}^y u(y),$$

where $P_{V_n}^y$ is the projection onto V_n with respect to norm $\|\cdot\|_y$.

Hence, one would like to derive estimates on $\|u(y) - P_{V_n}^y u(y)\|_{H_0^1}$ in place of the estimates on $\|u(y) - P_{V_n} u(y)\|_{H_0^1}$ that we have obtained so far, since $P_{V_n} u(y)$ is not practically accessible. As explained in the introduction, we cannot be satisfied with combining the latter estimates with the bound

$$\|u(y) - P_{V_n}^y u(y)\|_{H_0^1} \leqslant \kappa(y)^{1/2} \|u(y) - P_{V_n} u(y)\|_{H_0^1}$$

derived from Cea's lemma, since the multiplicative constant $\kappa(y)$ from (2.9) is not uniformly bounded over the manifolds $\mathcal{M}$, $\mathcal{B}$ or $\mathcal{N}$. Here, we shall employ another approach to derive the same rates of convergence for $\|u(y) - P_{V_n}^y u(y)\|_{H_0^1}$.

One first observation is that in order for Galerkin projection $P_{V_n}^y$ onto a reduced model space V_n to satisfy a convergence bound in relative error, it is critical that this space contains some functions from the limit spaces V_S. This is expressed by the following result.

Proposition 2.4.1. *Assume that there exists* $S \subsetneq \{1, \dots, d\}$ *such that* $V_n \cap V_S = \{0\}$. *Then for any* $C \in]0, 1[$, *there exists* $y \in Y'$ *such that*

$$\|u(y) - P_{V_n}^y u(y)\|_{H_0^1} \geqslant C \|u(y)\|_{H_0^1}. \tag{2.58}$$

Proof. Since $V_n \cap V_S = \{0\}$, the quantity $\|\nabla v\|_{L^2(\Omega_S)}$ is a norm on V_n and one can define

$$\alpha = \min_{v \in V_n} \frac{\|\nabla v\|_{L^2(\Omega_S)}}{\|v\|_{H^1_0}} > 0.$$

For any $\varepsilon > 0$, take $y_j = \varepsilon^{-2}$ for $j \in S$ and $y_j = 1$ for $j \in S^c$. Then, for $v = P^y_{V_n} u(y)$,

$$\frac{\alpha}{\varepsilon}\|v\|_{H^1_0} \leqslant \frac{1}{\varepsilon}\|\nabla v\|_{L^2(\Omega_S)} \leqslant \|v\|_y \leqslant \|u(y)\|_y \leqslant C_f \leqslant \frac{C_f}{c_f}\|u(y)\|_{H^1_0},$$

where we have used the framings (2.25) and (2.26). Therefore, taking $\varepsilon = \frac{c_f}{C_f}\alpha(1-C)$ implies $\|v\|_{H^1_0} \leqslant (1-C)\|u(y)\|_{H^1_0}$, and (2.58) follows. □

However, in the construction of V_n in Section 2.3, each space $V_{\ell,k}$ is a subset of V_S for $S = \{j \ : \ \ell_j = L_k\}$. This prevents the phenomenon described in the previous proposition from occurring. Instead, we obtain similar convergence bounds as those obtained for P_{V_n}, as expressed in the following result.

Theorem 2.4.2. *Assume that the partition of Ω has the geometry of disjoint inclusions. On the rectangles R_ℓ for $\ell \in \{0, \dots, L\}^d$, the following uniform convergence estimates hold:*

$$\|u(y) - P^y_{V_{\ell,k}} u(y)\|_{H^1_0} \leqslant \frac{C_f}{\sqrt{3}} 3^{-k}, \quad y \in R_\ell, \tag{2.59}$$

if $\|\ell\|_\infty < L$, and

$$\|u(y) - P^y_{V_{\ell,k}} u(y)\|_{H^1_0} \leqslant \frac{C_f}{\sqrt{3}} 3^{-k} + C_0 C_f 2^{-L}, \quad y \in R_\ell, \tag{2.60}$$

if $\|\ell\|_\infty = L$. As a consequence, with $L = L_k$ and V_n defined as in Section 2.3.3, one has the estimates

$$\|u(y) - P^y_{V_n} u(y)\|_{H^1_0} \leqslant C \exp\left(-cn^{\frac{1}{2d-2}}\right), \tag{2.61}$$

for all $y \in \widetilde{Y}'$ such that $\min y_j = 1$, and

$$\|u(y) - P^y_{V_{\ell,k}} u(y)\|_{H^1_0} \leqslant C \exp\left(-cn^{1/(2d-2)}\right) \|u(y)\|_{H^1_0}, \tag{2.62}$$

for all $y \in \widetilde{Y}$, with constants c and C that only depend on d, C_f, and on the geometry of the partition through the constant C_0.

Proof. For bounded rectangles R_ℓ with $\|\ell\|_\infty < L$, we know from Lemma 2.3.1, and more precisely from (2.33), that

$$\|u(y) - P^y_{V_{\ell,k}} u(y)\|_y = \min_{v \in V_{\ell,k}} \|u(y) - v\|_y \leqslant \left\|u(y) - \sum_{|\nu| \leqslant k} u_\nu y^\nu\right\|_y \leqslant \frac{C_f}{\sqrt{3}} 3^{-k}$$

for any $y \in R_\ell$. Since all the y_j are greater or equal to 1, one has $\|v\|_{H_0^1} \leqslant \|v\|_y$ for all v and therefore (2.59) follows.

For infinite rectangles R_ℓ such that $\|\ell\|_\infty = L$, we again introduce $S = \{j \ : \ \ell_j = L\}$. Then, using (2.47),

$$\begin{aligned}\|u(y) - P^y_{V_{\ell,k}} u(y)\|_{H_0^1} &\leqslant \|u(y) - u_S(y_{S^c})\|_{H_0^1} + \|u_S(y_{S^c}) - P^y_{V_{\ell,k}} u(y)\|_{H_0^1} \\ &\leqslant C_0 C_f 2^{-L} + \|u_S(y_{S^c}) - P^y_{V_{\ell,k}} u(y)\|_{H_0^1}.\end{aligned}$$

Since $V_{\ell,k} \subset V_S$, we have

$$P^y_{V_{\ell,k}} u(y) = P^y_{V_{\ell,k}} P^y_{V_S} u(y) = P^y_{V_{\ell,k}} u_S(y_{S^c}) = P^{y_{S^c}}_{V_{\ell,k}} u_S(y_{S^c}),$$

Similarly to the previous case, we apply (2.39) from Lemma 2.3.3:

$$\|u_S(y_{S^c}) - P^y_{V_{\ell,k}} u_S(y_{S^c})\|_{H_0^1} \leqslant \|u_S(y_{S^c}) - P^y_{V_{\ell,k}} u_S(y_{S^c})\|_y \leqslant \frac{C_f}{\sqrt{3}} 3^{-k},$$

and we thus obtain (2.60).

After taking $L = L_k$ and defining V_n as the sum of the $V_{\ell,k}$ for $\ell \in E_k$, the derivation of (2.61) and (2.62) is exactly the same as for (2.54) and (2.56). □

Remark 2.4.3. *As in Remark 2.3.10, it is expected that the same rate of convergence is attained if V_n is a reduced basis space generated by solutions $u(y^i)$, $i = 1, \ldots, n$, as long as there are $O\left(\binom{k+d-1}{d-1}\right)$ samples y^i in each rectangle, however with samples forced to be of the form $u_S(y^i_{S^c}) \in V_S$ in the case of infinite rectangles.*

2.4.2 State and parameter estimation

The state estimation problem consists in retrieving the solution $\overline{u} = u(\overline{y})$ when the parameter $\overline{y}$ is unknown, and one observes m linear measurements

$$w_i = \ell_i(\overline{u}), \quad i = 1, \ldots, m,$$

where the ℓ_i are continuous linear functional on the Hilbert space V that contains the solution manifold. These linear functionals may thus be written in terms of Riesz representers

$$\ell_i(v) = \langle \omega_i, v \rangle_V.$$

The Parametrized Background Data Weak (PBDW) method, introduced in [108] and further studied in [26], exploits the fact that all potential solutions are well approximated by reduced model spaces V_n. It is based on a simple recovery algorithm that consists in solving the problem

$$\min_{u \in V_w} \min_{v \in V_n} \|u - v\|_V, \tag{2.63}$$

where, for $w = (w_1, \dots, w_m) \in \mathbb{R}^m$,

$$V_w := \{u \in V \ : \ \ell_i(u) = w_i, \ i = 1, \dots, m\},$$

is the affine space of functions that agree with the measurements.

The analysis of this problem is governed by the quantity

$$\mu_n = \mu(V_n, W) := \sup_{v \in V_n} \frac{\|v\|_V}{\|P_W v\|_V}, \tag{2.64}$$

where $W := \operatorname{span}\{\omega_1, \dots, \omega_m\}$, which is finite if and only if $V_n \cap W^\perp = \{0\}$. Then, there exists a unique minimizing pair

$$(u^*, v^*) = (u^*(w), v^*(w)) \in V_w \times V_n$$

to (2.63), which satisfies the estimates

$$\|\overline{u} - v^*\|_V \leqslant \mu_n \min_{v \in V_n} \|u - v\|_V, \tag{2.65}$$

and

$$\|\overline{u} - u^*\|_V \leqslant \mu_n \min_{v \in V_n + (W \cap V_n^\perp)} \|u - v\|_V. \tag{2.66}$$

The computation of (u^*, v^*) amounts to solving finite linear systems, and both solutions depend linearly on w.

Turning to our specific elliptic problem, and assuming that the ℓ_i belong to $H^{-1}(\Omega) = V'$ for $V = H^1_0(\Omega)$, we may apply the above PBDW method using the reduced basis spaces V_n introduced in Section 2.3. As an immediate consequence of Theorem 2.3.7, we obtain a recovery estimate in relative error.

Proposition 2.4.4. *Let $\overline{y} \in \widetilde{Y}$ and $\overline{u} = u(\overline{y})$. Then both estimators $v^* \in V_n$ and $u^* \in V_w$ satisfy*

$$\max\{\|\overline{u} - v^*\|_{H^1_0}, \|\overline{u} - u^*\|_{H^1_0}\} \leqslant C\mu_n \exp\left(-cn^{\frac{1}{2d-2}}\right) \|\overline{u}\|_{H^1_0}. \tag{2.67}$$

The positive constants c and C only depend on d, C_f, and on the geometry of the partition through the constant C_0.

Proof. It follows readily by combining (2.56) applied to $y = \overline{y}$ with the recovery estimates (2.65) and (2.66). □

We next turn to the problem of parameter estimation, namely recovering an approximation y^* to $\overline{y}$ from the measurements w. In contrast to state estimation, this is a nonlinear inverse problem since the first mapping in

$$\overline{y} \mapsto \overline{u} \mapsto w$$

is typically nonlinear. One way of relaxing this problem into a linear one is by first using a recovery u^* of the state $\overline{u}$, for example obtained by the PBDW method. One then defines

y^* as the minimizer over $\widetilde{Y}$ of the residual

$$R(y) := \|\mathrm{div}(a(y)\nabla u^*) + f\|_{H^{-1}}.$$

This is a quadratic problem when $a(y)$ has an affine dependence in y, that can be solved by standard quadratic optimization methods. The rationale for this approach is the fact that

$$R(y) = \|A_y u^* - A_y u(y)\|_{H^{-1}} \sim \|u^* - u(y)\|_{H^1_0},$$

and therefore we should be close to finding the parameter y that best explains the approximation u^*. Unfortunately, this approach is not much viable in the high-contrast regime since the equivalence $\|A_y v\|_{H^{-1}} \sim \|v\|_{H^1_0}$ has constants that are not uniform in y and deteriorate with the level of contrast.

Instead, we propose a more specific approach that exploits the piecewise constant structure of $a(y)$, assuming that V_n is a reduced space of the form

$$V_n = \mathrm{span}(u^1, \dots, u^n), \quad u^i = u(y^i),$$

for some properly selected parameter vectors

$$y^i = (y^i_1, \dots, y^i_d), \quad i = 1, \dots, n.$$

As mentioned, see Remark 2.3.10, these spaces satisfy the same exponential convergence bounds as the spaces constructed in Section 2.3.

The PBDW estimator $v^* = v^*(w) \in V_n$ thus has the form

$$v^* = \sum_{i=1}^n c_i u^i \in V_n$$

and satisfies a similar bound (2.67) as in the above proposition. Then, on the particular domain Ω_j, one has

$$\frac{f}{\overline{y}_j} = -\Delta \overline{u}_{|\Omega_j} \approx -\sum_{i=1}^n c_i \Delta u^i = \sum_{i=1}^n c_i \frac{f}{y^i_j},$$

and therefore, a natural candidate for the parameter estimate is $y^* = (y^*_1, \dots, y^*_d)$ with

$$y^*_j := \left(\sum_{i=1}^n \frac{c_i}{y^i_j} \right)^{-1}. \tag{2.68}$$

The following result gives a recovery bound in relative error for the inverse diffusivity.

Proposition 2.4.5. *With the notation $1/y = (1/y_1, \dots, 1/y_d)$, the estimator y^* defined by (2.68) satisfies the bound*

$$\left\| \frac{1}{y^*} - \frac{1}{\overline{y}} \right\|_\infty \leqslant \frac{C_f}{c_f} C \mu_n \exp\left(-c n^{\frac{1}{2d-2}} \right) \left\| \frac{1}{\overline{y}} \right\|_\infty, \tag{2.69}$$

where C_f and c_f are as in (2.25), and the other constants as in (2.67).

Proof. For $1 \leqslant j \leqslant d$, take $\phi \in H^1_0(\Omega_j)$, then

$$\begin{aligned}\left|\frac{1}{y_j^*}-\frac{1}{\overline{y}_j}\right| |\langle f,\phi\rangle_{H^{-1},H^1_0}| &= \left|\sum_{i=1}^n \frac{c_i}{y_j^i}\int_{\Omega_j} y_j^i \nabla u^i\cdot\nabla\phi\,dx - \frac{1}{\overline{y}_j}\int_{\Omega_j}\overline{y}_j\nabla\overline{u}\cdot\nabla\phi\,dx\right| \\ &= \left|\int_{\Omega_j}\nabla(v^*-\overline{u})\cdot\nabla\phi\,dx\right| \\ &\leqslant \|v^*-\overline{u}\|_{H^1_0(\Omega)}\|\phi\|_{H^1_0(\Omega_j)}.\end{aligned}$$

Optimizing over ϕ gives

$$\left\|\frac{1}{y^*}-\frac{1}{\overline{y}}\right\|_\infty \leqslant c_f^{-1}\|v^*-\overline{u}\|_{H^1_0},$$

which combined with (2.67) gives

$$\left\|\frac{1}{y^*}-\frac{1}{\overline{y}}\right\|_\infty \leqslant c_f^{-1}C\mu_n\exp\left(-cn^{\frac{1}{2d-2}}\right)\|\overline{u}\|_{H^1_0}.$$

Using the Lax-Milgram estimate

$$\|\overline{u}\|_{H^1_0} \leqslant C_f\left\|\frac{1}{\overline{y}}\right\|_\infty,$$

we reach (2.69). □

Remark 2.4.6. *The bound (2.69) is not entirely satisfactory since the approximation error on $\overline{y}_j$ remains high when $\overline{y}\in\mathcal{N}$ with $\overline{y}_j \gg 1$. We do not know if a bound of the form*

$$\left|\frac{1}{y_j^*}-\frac{1}{\overline{y}_j}\right| \leqslant \frac{\varepsilon_n}{\overline{y}_j},\quad 1\leqslant j\leqslant d,$$

which would imply $|y_j^*-\overline{y}_j| \leqslant \varepsilon_n/(1-\varepsilon_n)\,\overline{y}_j$, holds uniformly over $\mathcal{N}$ with $\varepsilon_n \underset{n\to+\infty}{\longrightarrow} 0$.

2.5 Numerical illustration

The base model that will be used all along the numerical illustrations is the diffusion equation (2.4) with data $f=1$ set on the two-dimensional square $\Omega=[-1,1]^2$ with homogeneous Dirichlet boundary conditions. We consider a piece-wise constant diffusion coefficient

$$a_{|\Omega_j} = y_j,\quad 1\leqslant j\leqslant d,$$

on a partition of Ω into 16 squares of quarter side-length.

As such this partition does not satisfy the geometrical assumption of "Lipschitz partition" that was critical in our analysis for the application of Lemma 2.3.5. Therefore we

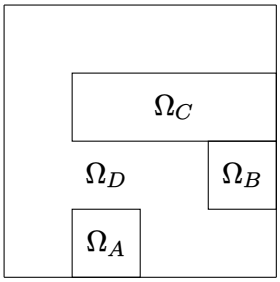


Figure 2.3: Lipschitz partition of Ω.

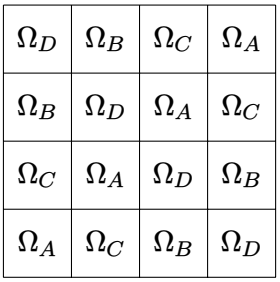


Figure 2.4: Non-lipschitz partition of Ω.

consider sub-partitions that comply to the assumptions, such as illustrated on Figure 2.3, which amounts to equate the parameters y_j of squares belonging to the same sub-domain. This way we can consider that $y = (y_A, y_B, y_C, y_D)$ consists of four parameters, one per each subdomain.

The numerical results that we next present aim to illustrate the robustness to high-contrast of the reduced basis method, and discuss in addition the effect of parameter selection, higher parametric dimensions, and inclusions that are not satisfying the geometric assumption as exemplified on Figure 2.4.

We construct different reduced bases $\{u^1, \dots, u^n\}$ of moderate dimension $1 \leqslant n \leqslant 15$, where

$$u^k = u(y^k),$$

for certain parameter selections $y^1, \dots, y^n$. Each reduced basis element u^k is numerically computed by the Galerkin method in a background finite element space V_h of dimension 6241.

The reduced basis spaces are thus subspaces of V_h, thus strictly speaking spaces $V_{n,h}$ depending on n and on the meshsize h. In our numerical computation, we always assess the error

$$P^y_{V_h} u(y) - P^y_{V_{n,h}} u(y).$$

We noticed that for the considered values of $n = 1, \dots, 15$ the error curves do not vary much when further reducing the mesh size h. In fact they are already essentially the same when the dimension of V_h is four times smaller. Therefore, for simplicity of the presentation, we still write

$$u(y) - P^y_{V_n} u(y),$$

bearing in mind that the additional finite element error $u(y) - P^y_{V_h} u(y)$ depends on h (with algebraic decay in the finite element dimension).

All the tests were done using Python 3.8. For more information and experiments not presented here we invite the reader to look into the github repository `https://github.com/agussomacal/ROMHighContrast`.

2.5.1 Parameter selection

We first study the case of a one parameter family : the diffusion coefficient y_A of Ω_A in Figure 2.3 varies from 1 to ∞, while the other subdomains are considered as background with all coefficents equal to 1. Thus the y^k are of the form $y^k = (y_A^k, 1, 1, 1)$.

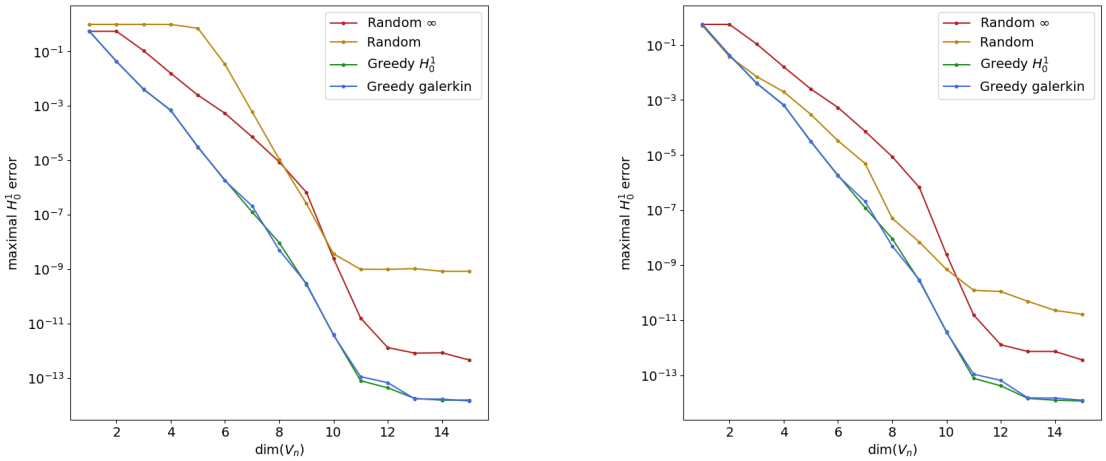


Figure 2.5: Galerkin (left) and H_0^1 (right) projection error, both measured in H_0^1 relative error, maximized over the parameter domain, for different reduced bases, case $d = 1$.

In reduced basis constructions, two approaches for parameter selection are usually considered : random or greedy. Random selection usually performs well enough in many situations, however we shall see that it fails in the high contrast regime. This is in particular due to the fact that it does not capture the limit solutions, while we have observed in Section 2.4 that robust convergence of the Galerkin method in the high-contrast regime critically requires to include limit solutions in the space V_n. Here, there is only one limit solution $u_\infty = u(y_\infty)$ where $y_\infty = (\infty, 1, 1, 1)$, and this element is picked by the greedy method if initialized at any other point.

More precisely, we compare four strategies for selecting the $y_A^k \in [1, \infty]$:

- Random: the y_A^k are drawn independently according to the uniform law for $\frac{1}{y_A} \in [0, 1]$.
- Random-∞: First the limit solution corresponding to $y_A = \infty$ is put in the basis. The rest of the elements are randomly picked as in the previous case.
- Greedy H_0^1: The y^k are picked incrementally, y^{k+1} maximizing the relative H_0^1 projection error $\|u(y) - P_{V_k} u(y)\|_{H_0^1} / \|u(y)\|_{H_0^1}$.
- Greedy Galerkin: The y^k are picked incrementally, y^{k+1} maximizing the relative H_0^1 error of the Galerkin projection $\|u(y) - P_{V_k}^y u(y)\|_{H_0^1} / \|u(y)\|_{H_0^1}$.

Figure 2.5 displays on the left the evolution of the maximal relative error of the Galerkin projection

$$\sup_{y_A \in [1,\infty]} \frac{\|u(y) - P^y_{V_n} u(y)\|_{H^1_0}}{\|u(y)\|_{H^1_0}},$$

as a function of $n = \dim(V_n)$ for these various selection strategies. It reveals the superiority of the greedy selection that reaches machine precision after picking $n = 11$ reduced basis elements, and the gain in including the limit solution in the case of a random selection. As a comparison, we display on the right the decay of the relative H^1_0-orthogonal projection error

$$\sup_{y_A \in [1,\infty]} \frac{\|u(y) - P_{V_n} u(y)\|_{H^1_0}}{\|u(y)\|_{H^1_0}}$$

for the same parameter selection strategies. Here, we notice that the inclusion of the limit solution u_∞ is not anymore critical for reaching good accuracy. Nevertheless, these errors still decay faster for the greedy strategies.

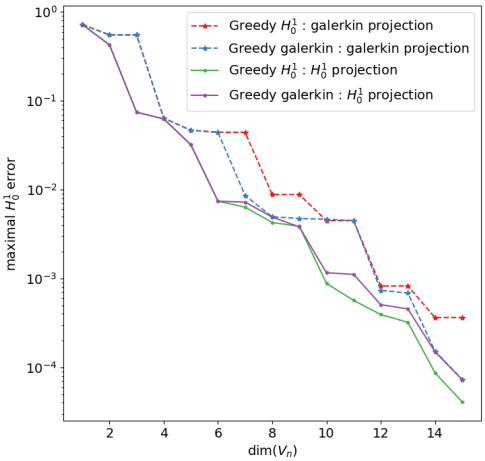


Figure 2.6: Galerkin and H^1_0 projection error (both measured in H^1_0 relative error maximized over the parameter domain) for different reduced bases, case $d = 2$.

Remark 2.5.1. *As the diffusion coefficient is piecewise constant on the partition $\Omega_A \cup \Omega_A^c$, the parameter space dimension is $d = 2$ in this numerical example. The theoretical results thus provide a bound on the error of order $\exp(-c\sqrt{n})$. However, this bound is obtained with local reduced spaces $V_{\ell,k}$ on dyadic intervals, which does not perform as well as $V_n = \bigoplus_{\ell \in E_k} V_{\ell,k}$, for which one might expect a rate closer to $\exp(-cn)$. In Figure 2.5 for $n \leqslant 11$, that is, until numerical precision issues arise, we even observe a faster than exponential convergence, that could be due to the superiority of reduced bases over polynomial approximations.*

Remark 2.5.2. *It is well known that the reduced basis can be very ill-conditioned, since u^n*

becomes extremely close to $V_{n-1} = \text{span}\{u^1, \dots, u^{n-1}\}$ *as* n *gets moderately large. In order to avoid numerical instabilities, prior to the computation of the Galerkin or* H_0^1 *projection onto* V_n*, we need to perform a change of basis, typically by some orthonormalization process. In our numerical test, we perform this orthonormalization with respect to the discrete* ℓ^2 *inner product for the nodal values in the background finite element representation, using the QR decomposition, and obtain a satisfactory stable numerical behavior. However, this process is not invariant under permutations, and we observe that it behaves better in terms of numerical stability when sorting the reduced basis elements from higher contrast to lower contrast.*

In this one parameter scenario, both greedy strategies behaved equally well. However, as we increase the dimensionality of the problem $d > 1$, Greedy Galerkin appears to be the best selection procedure, as could be expected since it optimizes the error based on the approximation which is effectively computed in forward modelling. Figure 2.6 shows this effect when $d = 2$, where y_A and y_B are allowed to vary independently while y_C and y_D are taken as background always equal to 1.

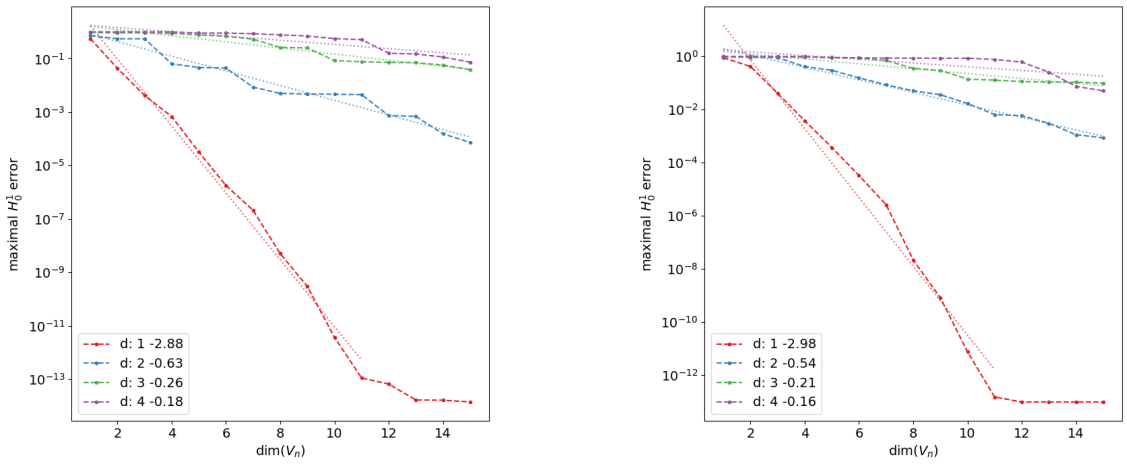


Figure 2.7: The Galerkin projection of Greedy Galerkin method for increasing dimensionality in geometries satisfying (left) or not (right) the assumptions.

2.5.2 Influence of dimensionality and geometry

In order to study the impact of dimensionality on the approximation rates, we compare the behavior of the Greedy Galerkin selection method, as we increase the number of freely varying parameters. As before, we will have for $y = (y_A, 1, 1, 1)$ when $d = 1$, then $y = (y_A, y_B, 1, 1)$ when $d = 2$, until having all four subdomains freely varying between 1 and $+\infty$.

In Figure 2.7 the degradation with respect to dimension is clearly observed as the

approximation capabilities strongly decrease. Even thought the exponential decay rate is still conserved, the decay parameter shrinks from almost 3 down to 0.22 when $d = 4$.

Secondly, we study the case where the geometrical assumptions are not satisfied. We follow the same incremental subdomains unfreezing as in the previous case but using the geometry stated in Figure 2.4. We observe that the reduced basis approach still achieves exponential approximation rates, actually higher than in the previous example. This hints that the geometric assumptions which are needed in our proofs could be artificial, and leaves open the question of achieving such results without relying on these assumptions.

Chapter 3

Non-linear approximation spaces for inverse problems

3.1 Introduction

3.1.1 The recovery problem

In this paper, we treat the following state estimation problem in a general Banach space V. We want to recover an approximation to an unknown function $u \in V$ from data given by m observations

$$z_i := \ell_i(u) + \eta_i, \quad i = 1, \dots, m, \tag{3.1}$$

where $\ell_i : V \mapsto \mathbb{R}$ are known measurement functionals, and η_i is additive noise. The functionals ℓ_i often correspond to the response of a physical measurement device but they can have a different interpretation depending on the application. Their behavior can be linear (in which case the ℓ_i are linear functionals from V', the dual of V) or nonlinear. This type of recovery problem is clearly ill-posed when the dimension of V exceeds m. It arises ubiquitously in sampling and inverse problem applications where V is infinite dimensional (to name a few, see [3, 9, 77, 94]).

One natural strategy to address this difficulty is to search for a recovery of u by an element of a low-dimensional reconstruction space $V_n \subset V$. The space V_n could be either an n-dimensional linear subspace, or more generally a nonlinear approximation space parametrized by n degrees of freedom, with $n \leqslant m$.

In order to obtain quantitative results for such recovery procedures, it is necessary to possess additional information about u, usually as an assumption that u belongs to a certain model class $\mathcal{K}$ contained in V. The approximation space V_n is chosen in order to collectively approximate the elements of $\mathcal{K}$ as well as possible, in the sense that

$$\operatorname{dist}(\mathcal{K}, V_n)_V := \max_{u \in \mathcal{K}} \min_{v \in V_n} ||u - v||_V$$

is as small as possible for moderate values of n.

Numerous theoretical results and numerical algorithms have been proposed in several

fields to study and solve the above recovery problem (below we recall some relevant results). However, to the best of our knowledge, they all involve at least one or several of the following assumptions:

- The ℓ_i are *linear* functionals,
- V_n is a *linear* (or affine) subspace of V,
- V is a *Hilbert* space,
- The model class $\mathcal{K}$ is a *ball in a smoothness space*, e.g., a unit ball in Lipschitz, Sobolev, or Besov spaces. Results involving this type of model classes have been intensively studied in the field of optimal recovery (see [32, 113, 119]).

The goal of this paper is to develop and analyze inversion procedures that do not require any of the above assumptions. Our analysis and numerical algorithms can thus be applied to virtually any recovery problem. The starting point of our development is based on algorithms introduced for inverse state estimation using reduced order models of parametrized Partial Differential Equations (PDEs). We next recall the specific framework. The presentation will also serve to explain more in depth the motivations leading to propose the present generalization.

3.1.2 State estimation with reduced models for parametrized PDE's

A relevant scenario in inverse state estimation is when the model class $\mathcal{K}$ is given by the set of solutions to some parameter-dependent PDE of the general form

$$\mathcal{P}(u, y) = 0, \tag{3.2}$$

where $\mathcal{P}$ is a differential operator, y is a vector of parameters ranging in some domain Y in $\mathbb{R}^d$, and u is the solution. If well-posedness holds in some Banach space V for each $y \in Y$, we denote by $u(y) \in V$ the corresponding solution for the given parameter value y and by

$$\mathcal{M} \coloneqq \{u(y) \ : \ y \in Y\},$$

the *solution manifold*.

In inverse state estimation, we take $\mathcal{K} = \mathcal{M}$ for the model class so the unknown u to recover belongs to $\mathcal{M}$. However, the parameter y that satisfies $u = u(y)$ is unknown, so we cannot solve the forward problem (3.2) to approximate u. Instead, we must approximate u from the partial observational data (3.1), and the knowledge of the model class $\mathcal{K} = \mathcal{M}$.

For the manifold $\mathcal{M}$, efficient approximation spaces V_n are usually obtained by reduced modelling techniques. In their most simple format, reduced models consist into linear spaces $(V_n)_{n\geqslant 0}$ with $\dim(V_n) = n$. The ideal benchmark in this linear approximation setting is provided by the Kolmogorov n-width

$$d_n(\mathcal{M})_V \coloneqq \inf_{\dim(V_n)\leqslant n} \operatorname{dist}(\mathcal{M}, V_n)_V,$$

which describes the optimal approximation performance achievable by an n-dimensional space over the set $\mathcal{M}$.

Apart from very simplified cases, the space V_n achieving the above infimum is usually out of reach. Practical model reduction techniques such as polynomial approximation in the parametrized domain [49, 54, 150] or reduced bases [71, 92, 109, 140, 161] construct spaces V_n that are "suboptimal yet good". In particular, the reduced basis method, which generates V_n by a specific selection of particular solution instances $u^1, \dots, u^n \in \mathcal{M}$, has been proved to have approximation error $\operatorname{dist}(\mathcal{M}, V_n)_V$ that decays with the same polynomial or exponential rates as $d_n(\mathcal{M})_V$, and in that sense are close to optimal [65].

3.1.3 The PBDW method

We take the *Parametrized Background Data Weak* (PBDW) method as a starting point for our analysis. The PBDW method, first introduced in [108], as well as several extensions, has been the object of a series of works [26, 27, 47, 48] on its optimality properties as a recovery algorithm. It has also been used for different practical applications, see [9, 77, 87]. We refer to [115] for an overview of the state of the art on this approach, and its connections with different fields. For our current purposes, it will suffice to recall the first version of the algorithm, which is the goal of this section.

The PBDW method uses a linear approximation space V_n of dimension $n \leqslant m$. Usually this space is a reduced model in applications. It is assumed that the ℓ_i are continuous linear functionals, that is $\ell_i \in V'$, and that V is a Hilbert space. Then, introducing the Riesz representers $\omega_i \in V$ such that $\ell_i(v) = \langle \omega_i, v \rangle_V$, the data of the noise-free observation

$$\ell(u) \coloneqq (\ell_1(u), \dots, \ell_m(u)),$$

is equivalent to that of the orthogonal projection $w = P_W u$ on the *Riesz measurement space*

$$W \coloneqq \operatorname{span}\{\omega_1, \dots, \omega_m\}.$$

Assuming linear independence of the ℓ_i, this space has dimension m. A critical quantity is the number

$$\mu = \mu(V_n, W) \coloneqq \max_{v \in V_n} \frac{\|v\|_V}{\|P_W v\|_V}, \tag{3.3}$$

that describes the "stability" of the description of an element of V_n by its projection onto W, and may be thought of as the inverse cosine of the angle between W and V_n. In particular, this quantity is finite only when $n \leqslant m$. It can be explicitly computed as the inverse of the smallest singular value of a cross-grammian matrix between orthonormal bases of V_n and W (see [26, 115]).

The PBDW method consists in solving the minimization problem

$$\min_{v \in V_w} \min_{\tilde{v} \in V_n} \|v - \tilde{v}\|_V,$$

where $V_w \coloneqq w + W^\perp$ is the set of all states v such that $P_W v = w$. We denote by $(u^*, \tilde{u}) \in$

$V_w \times V_n$ the minimizing pair, which is unique when $\mu < \infty$, and can be computed by solving an $n \times n$ linear system. The function $\tilde{u}$ may be seen as a particular best-fit estimator of u on V_n, since it is also defined by

$$\tilde{u} \coloneqq \operatorname{argmin}\{\|P_W v - w\|_V \ : \ v \in V_n\}.$$

The function u^* can be derived from $\tilde{u}$ by the correction procedure

$$u^* \coloneqq \tilde{u} + (w - P_W \tilde{u}),$$

which shows that $u^* \in V_n + W$. It may be thought of as a generalized interpolation estimator, since it agrees with the observed data ($P_W u^* = P_W u$). In the case of noise-free data, it is proved in [26, 108] that these estimators satisfy the recovery bounds

$$\|u - \tilde{u}\|_V \leqslant \mu \min_{v \in V_n} \|u - v\|_V \quad \text{and} \quad \|u - u^*\|_V \leqslant \mu \min_{v \in V_n \oplus (V_n^\perp \cap W)} \|u - v\|_V.$$

These bounds reflect a typical trade-off in the choice of the reduced basis space, since making n larger has both effect of decreasing the approximation error $\min_{v \in V_n} \|u - v\|_V$ and increasing the stability constant $\mu = \mu(V_n, W)$.

When the PBDW method is applied to noisy data, amounting in observing a perturbed version $\overline{w}$ of $w = P_W u$, the recovery bounds remain valid up to the additional term $\mu \|w - \overline{w}\|_V$. In summary, one has for both estimators

$$\max\{\|u - \tilde{u}\|_V, \|u - u^*\|_V\} \leqslant \mu(e_n(u) + \kappa), \tag{3.4}$$

where

$$e_n(u) \coloneqq \min_{v \in V_n} \|u - v\|_V$$

is the reduced model approximation error and $\kappa \coloneqq \|w - \overline{w}\|_V$ is the noise error measured in the space W. Note that since the additive perturbations η_i are applied to the data $\ell_i(u)$, a natural model for the measurement noise is to assume a bound of the form

$$\|\eta\|_p \leqslant \varepsilon, \tag{3.5}$$

for the vector $\eta = (\eta_1, \ldots, \eta_m)$, typically in the max norm $p = \infty$ or euclidean norm $p = 2$. Therefore, one has $\kappa \leqslant \beta\varepsilon$, where

$$\beta \coloneqq \max_{v \in W} \frac{\|v\|_V}{\|\ell(v)\|_p},$$

resulting in a bound of the form $\mu e_n(u) + \mu\beta\varepsilon$ for both estimators.

3.1.4 Towards nonlinear approximation spaces

The simplicity of the PBDW method and its variants comes together with a fundamental limitation on its performance: it is by essence a linear reconstruction method with recovery

bounds tied to the approximation error $e_n(u)$. When the only prior information is that the unknown function u belongs to a class $\mathcal{K}$, with $\mathcal{K} = \mathcal{M}$ the solution manifold in the case of parametric PDEs, its best performance over $\mathcal{K}$ is thus limited by the n-width $d_n(\mathcal{K})_V$ and in turn by $d_m(\mathcal{K})_V$ since $n \leqslant m$.

In several simple yet relevant settings, it is known that n-widths have poor decay with n. One instance is when the class $\mathcal{K}$ contains piecewise smooth states, with a state-dependent location of jump discontinuities. As an elementary example, one can easily check that if $V = L^2([0,1])$ and $\mathcal{K}$ is the set all indicator functions $u = \chi_{[a,b]}$ with $a, b \in [0,1]$, one has $d_n(\mathcal{K})_V \sim n^{-1/2}$. This decay is of course even slower for more general classes of piecewise smooth function in higher dimension, see in particular [21, Chapter 3, equation (3.76)]. Such functions are typical in parametrized hyperbolic PDEs, due to the presence of shocks with positions that differ when parameters entering the velocity vary. We refer to [18, 27, 72, 81, 120, 156] for other examples of parametric PDEs whose solution manifold has slow Kolmogorov n-width decay.

For such classes of functions, nonlinear approximation methods are well known to perform significantly better than their linear counterparts. Typical representatives of such methods include approximation by rational fractions, free knot splines or adaptive finite elements, best n-term approximation in a basis or dictionary, neural network or various tensor formats. In these instances the space V_n still depends on n or $\mathcal{O}(n)$ parameters but is not anymore a linear space. We refer to [63] for a general introduction on the topic of nonlinear approximation.

3.1.5 Objective and outline

The objective of this paper is to study the natural extensions of the PBDW method to such nonlinear approximation spaces and identify the basic structural properties that lead to near optimal recovery estimates similar to (3.4).

We begin in Section 3.2 by considering the most general setting where V is a Banach space, V_n a nonlinear approximation family, and the ℓ_i are functionals defined on V that are not necessarily linear, but Lipschitz continuous, that is

$$\|\ell(v) - \ell(\tilde{v})\|_Z \leqslant \alpha_Z \|v - \tilde{v}\|_V, \quad v, \tilde{v} \in V. \tag{3.6}$$

Here $\|\cdot\|_Z$ can be any given norm defined over $\mathbb{R}^m$ with the constant α_Z depending on this choice of norm. In this framework, we discuss the best-fit estimation procedure that consists in minimizing the distance to the observed data in a given norm $\|\cdot\|_Z$.

Our main structural assumption on V_n is the following *inverse stability property*: the reduced model is stable with respect to the measurement functionals if there exists a finite constant μ_Z such that

$$\|v - \tilde{v}\|_V \leqslant \mu_Z \|\ell(v) - \ell(\tilde{v})\|_Z, \quad v, \tilde{v} \in V_n. \tag{3.7}$$

The stability constant μ_Z depends on the Z norm and plays a role similar to that of μ in the linear case. In particular, we show that this constant is finite only if $n \leqslant m$. The

resulting estimator $\tilde{u}$ is then proved to satisfy a general recovery bound of the form

$$\|u - \tilde{u}\|_V \leqslant C_1 e_n(u) + C_2 \|\eta\|_p,$$

where $e_n(u) \coloneqq \min_{v \in V_n} \|u - v\|_V$ is the nonlinear reduced model approximation error, $\|\eta\|_p$ the level of measurement noise in ℓ^p norm, and the constants C_1 and C_2 depend on α_Z and μ_Z.

In Section 3.3, we consider the more particular setting where the ℓ_i are linear functionals. Then, we show that constants C_1 and C_2 are each minimized by a different choice of norm $\|\cdot\|_Z$, resulting in two different best fit estimators $\tilde{u}$, as already observed in [22] in the case of linear reduced models. This particular setting also allows us to introduce a generalized interpolation estimator u^* and establish similar recovery estimates for $\|u - u^*\|_V$.

We next apply our framework to the inverse problem that consists in recovering a general shape Ω, identified to its characteristic function χ_Ω, based on cell average data

$$a_T(\Omega) \coloneqq \frac{1}{|T|} \int_T \chi_\Omega, \quad T \in \mathcal{T},$$

where $\mathcal{T}$ is a fixed cartesian mesh. One motivation for this problem is the design of finite volume schemes for the computation of solutions to transport PDEs on such meshes.

We first discuss in Section 3.4 the best estimation rate in terms of the mesh size h that can be achieved by standard linear reconstructions, and which is essentially that of piecewise constant approximations, that is $\mathcal{O}(h^{1/q})$ regardless of the smoothness of the boundary $\partial\Omega$. This intrinsic limitation is due to the presence of the jump discontinuity that is not well resolved by the mesh.

We then discuss in Section 3.5 a local recovery strategy based on a nonlinear approximation space V_n that consists of characteristic functions of half-planes which can fit the boundary of Ω at a subcell resolution level, as already proposed in [8, 128, 129, 132]. One main result, whose proof is given in an appendix, is that this approximation space is stable in the sense of (3.7) with respect to cell average measurements on a stencil of 3×3 squares. In turn, if Ω has a C^2 boundary, the recovered shape $\tilde{\Omega}$ is proved to satisfy an estimate of the form

$$\|\chi_\Omega - \chi_{\tilde{\Omega}}\|_{L^q} \leqslant C h^{2/q},$$

where h is the mesh size, which cannot be achieved by any linear reconstruction. This paves the way to higher order reconstruction methods for smoother boundaries by using local nonlinear approximation spaces with curved boundaries and larger stencils.

Finally, we discuss in Section 3.6 the application of our results to the recovery of large vectors of size N from $m < N$ linear measurements, up to the error of best n-term approximation. This problem is well-known in compressed sensing [42, 74], and was in particular studied in [46] which discusses the importance of the recovery norm $\|\cdot\|_V$ to understand if near-optimal recovery bounds can be achieved with m not much larger than n. We show that the structural assumptions identified in our general setting are naturally related to the so-called *null space property* introduced in [46].

3.2 Nonlinear reduction of inverse problems

3.2.1 A general framework

In full generality we are interested in recovering functions u in a general Banach space V with norm $\|\cdot\|_V$, from the measurement vector $z = (z_1, \dots, z_m) \in \mathbb{R}^m$ given by (3.1). A recovery (or inversion) map

$$z \to R(z)$$

takes this vector to an approximation $R(z)$ of u. We are interested in controlling the recovery error $\|u - R(z)\|_V$.

To build the recovery map R, we use a nonlinear approximation space of dimension n is a family of functions that can be described by n parameters. Loosely speaking, this means that there exists a set $S \subset \mathbb{R}^n$ and a continuous map $\varphi : S \to V$ such that

$$V_n \coloneqq \{\varphi(x) \ : \ x \in S\}.$$

Note that this definition covers the case of an n dimensional linear subspace since we can choose $S = \mathbb{R}^n$ and φ a linear map.

Our main assumptions are the Lipschitz stability of the functionals ℓ_i over the whole space V and their inverse Lipschitz stability over the nonlinear approximation space V_n, expressed by (3.6) and (3.7), respectively. Note that since $\mathbb{R}^m$ is finite dimensional, the norm $\|\cdot\|_Z$ that is chosen in $\mathbb{R}^m$ to express these properties could be arbitrary up to a modification of the stability constants α_Z, μ_Z. These constants can be optimally defined as

$$\alpha_Z = \sup_{v_1, v_2 \in V} \frac{\|\ell(v_1) - \ell(v_2)\|_Z}{\|v_1 - v_2\|_V},$$

and

$$\mu_Z = \sup_{v_1, v_2 \in V_n} \frac{\|v_1 - v_2\|_V}{\|\ell(v_1) - \ell(v_2)\|_Z}.$$

Note that one always has $\alpha_Z \mu_Z \geqslant 1$.

Remark 3.2.1. *Note that when V_n is an n-dimensional space and the ℓ_i are linear functionals, the quantity μ_Z may be rewritten as*

$$\mu_Z = \max_{v \in V_n} \frac{\|v\|_V}{\|\ell(v)\|_Z}.$$

As discussed further, the quantity μ defined in (3.3) for the analysis of the PBDW method is an instance of μ_Z corresponding to a particular choice of norm $\|\cdot\|_Z$. Assuming the ℓ_i are independent functionals, one easily checks that finiteness of this quantity imposes that $n \leqslant m$. Indeed, if $n > m$, there exists a non-trivial $v \in V_n \cap \mathcal{N}$, where

$$\mathcal{N} \coloneqq \{v \ : \ \ell(v) = 0\}$$

is the null space of the measurement map that has codimension m, and therefore μ_Z is infinite.

Remark 3.2.2. *The restriction $n \leqslant m$ is also needed for nonlinear spaces V_n and measurement ℓ, under assumptions expressing that m and n are local dimensions. More precisely, assume that the map φ defining V_n is differentiable at some x_0 in the interior of S, that ℓ is differentiable at $v_0 = \varphi(x_0)$, and that both tangent maps have full rank at these points, that is,*

$$\dim(d\varphi_{x_0}(\mathbb{R}^n)) = n \quad \text{and} \quad \dim(d\ell_{v_0}(V)) = m.$$

Then, by taking $v_1 = v_0$ and $v_2 = \varphi(x_0 + tx)$ in the quotient that defines μ_Z, and letting $t \to 0$ for arbitrary $x \in \mathbb{R}^n$, one finds that

$$\mu_Z \geqslant \max_{v \in d\varphi_{x_0}(\mathbb{R}^n)} \frac{\|v\|_V}{\|d\ell_{v_0}(v)\|_Z},$$

and therefore it is infinite if $n \leqslant m$, by the same argument as in the previous remark.

3.2.2 The best fit estimator

We define a first recovery map $z \mapsto \tilde{u} = R(z)$ as the best fit estimator in the Z norm

$$\tilde{u} := \operatorname{argmin}\{\|z - \ell(v)\|_Z \ : \ v \in V_n\}. \tag{3.8}$$

The existence of such a minimizer is trivial if the space V_n and the measurement map ℓ are linear. It can also be ensured in the nonlinear case under additional assumptions, for example compactness of the set S defining the nonlinear space V_n, which will be the case in the application to shape recovery discussed in Section 3.5. If the minimizer does not exist, we may consider a near minimizer, that is $\tilde{u} \in V_n$ satisfying

$$\|z - \ell(\tilde{u})\|_Z \leqslant C\|z - \ell(v)\|_Z, \quad v \in V_n,$$

for some fixed $C > 1$. Inspection of the proofs of our main results below reveals that similar recovery bounds can be obtained for such a near minimizer, up to the multiplicative constant C.

Recall that our assumption (3.5) on the noise model is a control on $\|\eta\|_p$ for some $1 \leqslant p \leqslant \infty$. For this value of p, we introduce the quantity

$$\beta_Z := \max_{z \in \mathbb{R}^m} \frac{\|z\|_Z}{\|z\|_p}$$

We are now in position to state a recovery bound in this general framework.

Theorem 3.2.3. *The best fit estimator $\tilde{u}$ from (3.8) satisfies the estimate*

$$\|u - \tilde{u}\|_V \leqslant C_1 e_n(u) + C_2\|\eta\|_p, \tag{3.9}$$

where $C_1 := 1 + 2\alpha_Z\mu_Z$ and $C_2 := 2\beta_Z\mu_Z$.

Proof. Consider any $v \in V_n$ and write

$$\|u - \tilde{u}\|_V \leqslant \|u - v\|_V + \|v - \tilde{u}\|_V \leqslant \|u - v\|_V + \mu_Z \|\ell(v) - \ell(\tilde{u})\|_Z,$$

where we have used (3.7). On the other hand, the minimizing property of $\tilde{u}$ ensures that

$$\|\ell(v) - \ell(\tilde{u})\|_Z \leqslant \|z - \ell(v)\|_Z + \|z - \ell(\tilde{u})\|_Z \leqslant 2\|z - \ell(v)\|_Z.$$

Furthermore, using the stability (3.6) of ℓ and the definition of β_Z, we have

$$\|z - \ell(v)\|_Z \leqslant \|\ell(v) - \ell(u)\|_Z + \|\eta\|_Z \leqslant \alpha_Z \|u - v\| + \beta_Z \|\eta\|_p.$$

Combining the three estimates, we reach

$$\|u - \tilde{u}\|_V \leqslant (1 + 2\alpha_Z \mu_Z)\|u - v\|_V + 2\beta_Z \mu_Z \|\eta\|_p,$$

which gives (3.9) by optimizing over $v \in V_n$. □

The constants C_1 and C_2 in the above recovery estimate depend on the choice of norm $\|\cdot\|_Z$. Note that they are invariant when this norm is scaled by a factor $t > 0$, since this has the effect of multiplying α_Z and β_Z by t and dividing μ_Z by t, which is consistant with the fact that the resulting estimator $\tilde{u}$ is left unchanged by such a scaling. In the next section we show, in the particular setting of linear measurements, that specific choices of $\|\cdot\|_Z$ can be used to minimize C_1 or C_2. This setting also allows us to introduce and study a generalized interpolation estimator, which is not relevant to the present section since the nonlinear measurement map ℓ is not assumed to be surjective: in the presence of noise, there might exist no $v \in V$ that agrees with the data, in the sense that $z = \ell(u) + \eta$ does not belong to the range of ℓ.

3.3 Linear observations

In this section, we assume that the $\ell_i \in V'$ are independent linear functionals, still allowing V_n to be a general nonlinear space. In this framework, which contains the example of shape recovery discussed in Section 3.5, one has

$$\alpha_Z = \max_{v \in V} \frac{\|\ell(v)\|_Z}{\|v\|_V}$$

and

$$\mu_Z = \max_{v \in V_n^{\mathrm{diff}}} \frac{\|v\|_V}{\|\ell(v)\|_Z},$$

where

$$V_n^{\mathrm{diff}} = V_n - V_n \coloneqq \{v_1 - v_2 \ : \ v_1, v_2 \in V_n\}.$$

In this particular setting, we can identify the norms $\|\cdot\|_Z$ that minimize the constants $C_1 \coloneqq 1 + 2\alpha_Z \mu_Z$ and $C_2 \coloneqq 2\beta_Z \mu_Z$, respectively.

3.3.1 Optimal norms

As $\ell : V \to \mathbb{R}^m$ is continuous and surjective, we can define a norm on $\mathbb{R}^m$ through

$$\|z\|_W = \min\{\|v\|_V \ : \ \ell(v) = z\}. \tag{3.10}$$

Remark 3.3.1. *If V is a Hilbert space, the minimizer is unique by strict convexity of $\|\cdot\|_V$, and the m-dimensional space*

$$W := \left\{ \arg\min_{\ell(v)=z} \|v\|_V, \ z \in \mathbb{R}^m \right\}$$

is exactly the span of the Riesz representers of the observation functionals $\ell_i \in V'$. Moreover, denoting P_W the orthogonal projection on W, we have

$$\|\ell(v)\|_W = \|P_W v\|_V, \quad v \in V.$$

For this reason, we sometimes refer to $\|\cdot\|_W$ as the Riesz norm *even in the case of a more general Banach space.*

The following result shows that the choice $\|\cdot\|_Z := \|\cdot\|_W$ is the one that minimizes the constant C_1, while C_2 is minimized by simply taking the ℓ^p norm $\|\cdot\|_Z = \|\cdot\|_p$.

Theorem 3.3.2. *For any norm $\|\cdot\|_Z$, one has*

$$\alpha_W \mu_W = \mu_W \leqslant \alpha_Z \mu_Z,$$

and

$$\beta_p \mu_p = \mu_p \leqslant \beta_Z \mu_Z,$$

where $(\alpha_W, \beta_W, \mu_W)$ and $(\alpha_p, \beta_p, \mu_p)$ are the triplets $(\alpha_Z, \beta_Z, \mu_Z)$ when $\|\cdot\|_Z := \|\cdot\|_W$ and $\|\cdot\|_Z = \|\cdot\|_p$, respectively.

Proof. One has

$$\alpha_W = \max_{v \in V} \frac{\|\ell(v)\|_W}{\|v\|_V} = \max_{z \in \mathbb{R}^m} \max_{\ell(v)=z} \frac{\|z\|_W}{\|v\|_V} = 1,$$

and

$$\mu_W = \max_{v \in V_n^{\mathrm{diff}}} \frac{\|v\|_V}{\|\ell(v)\|_W} \leqslant \max_{v \in V_n^{\mathrm{diff}}} \frac{\|\ell(v)\|_Z}{\|\ell(v)\|_W} \max_{v \in V_n^{\mathrm{diff}}} \frac{\|v\|_V}{\|\ell(v)\|_Z} = \max_{v \in V_n^{\mathrm{diff}}} \frac{\|\ell(v)\|_Z}{\|\ell(v)\|_W} \mu_Z.$$

We now observe that from the definition of W, one has

$$\max_{v \in V_n^{\mathrm{diff}}} \frac{\|\ell(v)\|_Z}{\|\ell(v)\|_W} \leqslant \max_{z \in \mathbb{R}^m} \frac{\|z\|_Z}{\|z\|_W} = \max_{z \in \mathbb{R}^m} \max_{\ell(v)=z} \frac{\|z\|_Z}{\|v\|_V} = \alpha_Z.$$

We have thus obtained the first claim $\alpha_W \mu_W = \mu_W \leqslant \alpha_Z \mu_Z$. For the second claim, note

that we trivially have $\beta_p = 1$, and so

$$\beta_p \mu_p = \mu_p = \max_{v \in V_n^{\mathrm{diff}}} \frac{\|v\|_V}{\|\ell(v)\|_p} \leqslant \max_{v \in V_n^{\mathrm{diff}}} \frac{\|\ell(v)\|_Z}{\|\ell(v)\|_p} \max_{v \in V_n^{\mathrm{diff}}} \frac{\|v\|_V}{\|\ell(v)\|_Z} \leqslant \beta_Z \mu_Z.$$

□

Remark 3.3.3. *In the particular case where V is a Hilbert space, V_n a linear subspace and $p = 2$, it was already observed in [22] that the reconstruction operators based on the choice $\|\cdot\|_Z = \|\cdot\|_W$ or $\|\cdot\|_Z = \|\cdot\|_2$ are the most stable with respect to the approximation error and the noise error, respectively. The above result may thus be seen as a generalization of this state of affairs to the case of nonlinear subspaces of Banach spaces, and ℓ^p noise.*

3.3.2 The generalized interpolation estimator

Thanks to the surjectivity of ℓ, we may introduce the space

$$V_z := \{v \in V \ : \ \ell(v) = z\},$$

and consider the minimization problem

$$\min_{v \in V_z} \min_{\tilde{v} \in V_n} \|v - \tilde{v}\|_V.$$

If $(u^*, \tilde{u}) \in V_z \times V_n$ is a minimizing pair, the function u^* is given by

$$u^* = u^*(z) \in \arg\min\{\operatorname{dist}(v, V_n)_V \ : \ \ell(v) = z\},$$

and is called the generalized interpolation estimator, since it exactly matches the data.

Remark 3.3.4. *The best fit and generalized interpolation estimation may be thought of as the two extreme cases, $t \to \infty$ and $t \to 0$, of the penalized estimator*

$$u_t := \operatorname{argmin}\{\|z - \ell(v)\|_Z + t \operatorname{dist}(v, V_n)_V\}.$$

As explained earlier, the generalized interpolation operator may not be well defined in the general case where the ℓ_i are nonlinear. As opposed to the best fit, or the above penalized estimator u_t when $t > 0$, the generalized interpolation estimator does not involve the choice of a particular norm Z.

On the other hand, we see that $\tilde{u}$ is the solution to the problem

$$\min_{\tilde{v} \in V_n} \operatorname{dist}(\tilde{v}, V_z)_V.$$

Observing that

$$\operatorname{dist}(\tilde{v}, V_z)_V = \min_{\ell(v) = z} \|\tilde{v} - v\|_V = \min_{\ell(v') = \ell(\tilde{v}) - z} \|v'\|_V = \|\ell(\tilde{v}) - z\|_W,$$

we thus find that $\tilde{u}$ is precisely the best fit estimator for the Riesz norm $\|\cdot\|_Z := \|\cdot\|_W$.

In the Hilbert space setting, the generalized interpolation estimator u^* is therefore the orthogonal projection of this particular best fit estimator $\tilde{u}$ onto the affine space V_z. It may thus also be derived from $\tilde{u}$ by the correction procedure

$$u^* = \tilde{u} + w - P_W \tilde{u},$$

where $w = \underset{\ell(v)=z}{\arg\min} \|v\|_V \in W$ is the preimage by ℓ of the measurements z. In the noiseless case when $w = P_W u$, this correction can only improve the approximation since it reduces the component of $u - \tilde{u}$ in the W direction while leaving unchanged the orthogonal component, and so, in view of Theorems 3.2.3 and 3.3.2, we are ensured that

$$\|u - u^*\|_V \leqslant C_1 e_n(u),$$

where $C_1 := 1 + 2\mu_W$.

More generally, in the noisy case, and without the assumption that V is a Hilbert space, there is no guarantee that u^* performs better than $\tilde{u}$, but we still obtain an error estimate on u^* that is similar in nature to that satisfied by $\tilde{u}$.

Theorem 3.3.5. *The generalized interpolation estimator u^* satisfies the estimate*

$$\|u - u^*\|_V \leqslant C_1 e_n(u) + C_2 \|\eta\|_p, \tag{3.11}$$

where $C_1 := 2 + 2\mu_W$ and $C_2 := (1 + 2\mu_W)\beta_W$.

Proof. Take $\delta \in \underset{\ell(v)=\eta}{\arg\min} \|v\|_V$, so that $\ell(\delta) = \eta$ and $\|\eta\|_W = \|\delta\|_V$. For v and v^* in V_n, decompose

$$\|u - u^*\|_V \leqslant \|u - v\|_V + \|v - v^*\|_V + \|v^* - u^*\|_V. \tag{3.12}$$

For the middle term, using (3.7), we write

$$\begin{aligned}
\|v - v^*\|_V &\leqslant \mu_W \|\ell(v - v^*)\|_W \\
&\leqslant \mu_W (\|\ell(v - u)\|_W + \|\ell(u - u^*)\|_W + \|\ell(u^* - v^*)\|_W) \\
&\leqslant \mu_W (\|v - u\|_V + \|\eta\|_W + \|u^* - v^*\|_V)
\end{aligned}$$

since $\alpha_W = 1$, so the decomposition (3.12) becomes

$$\|u - u^*\|_V \leqslant (1 + \mu_W)\|u - v\|_V + \mu_W \|\eta\|_W + (1 + \mu_W)\|v^* - u^*\|_V.$$

To bound the last term, we optimize over the choice of $v^* \in V_n$ and use the definition of u^*

to obtain

$$\inf_{v^* \in V_n} \|v^* - u^*\|_V = \operatorname{dist}(u^*, V_n) \leqslant \operatorname{dist}(u+\delta, V_n) \leqslant \operatorname{dist}(u, V_n) + \|\delta\|_V = e_n(u) + \|\eta\|_W$$

since $\ell(u+\delta) = \ell(u) + \eta = z$. Combining the last two estimates and optimizing over $v \in V_n$ gives

$$\|u - u^*\|_V \leqslant (2 + 2\mu_W) e_n(u) + (1 + 2\mu_W)\|\eta\|_W,$$

and the result follows from the definition of β_W. □

3.4 Shape recovery from cell averages

3.4.1 The shape recovery problem

The problem of reconstructing a function u from its cell averages

$$a_T(u) \coloneqq \frac{1}{|T|} \int_T u, \quad T \in \mathcal{T},$$

where $\mathcal{T}$ is a partition of the domain $D \subset \mathbb{R}^d$ in which u is defined, appears naturally in two areas:

(i) In numerical simulation of hyperbolic conservation laws, it plays a central role when developing finite volume schemes on the computation mesh $\mathcal{T}$.

(ii) In $2d$ or $3d$ image processing, it corresponds to the so-called super-resolution problem, that is, reconstructing a high resolution image from its low resolution version defined on the coarse grid $\mathcal{T}$ of pixels or voxels.

Standard reconstruction methods are challenged when the function u exhibits jump discontinuities which are not well resolved by the partition $\mathcal{T}$. Such discontinuities correspond to edges in image processing or shocks in conservation laws. Here we may focus on the very simple case of characteristic functions of sets

$$u = \chi_\Omega,$$

that already carry the main difficulty. Therefore we are facing a problem of reconstructing a shape Ω from local averages of χ_Ω.

As a simple example we work in the domain $D = [0,1]^2$ with a uniform grid based on square cells of sidelength $h = \frac{1}{L}$ for some $L > 1$, therefore of the form

$$\mathcal{T} = \mathcal{T}_h \coloneqq \{T_{i,j} = [(i-1)h, ih] \times [(j-1)h, jh] \ : \ i,j = 1, \dots, L\}.$$

The cardinality of the grid is therefore

$$n \coloneqq \#(\mathcal{T}) = L^2 = h^{-2}.$$

We consider classes of characteristic functions χ_Ω of sets $\Omega \subset D$ with boundary of a prescribed Hölder smoothness. The definition of these classes requires some precision.

Definition 3.4.1. *For $s \geqslant 1$, $0 < R < 1/2$ and $M > 0$, we define the class $\mathcal{F}_{s,R,M}$ as consisting of all characteristic functions χ_Ω of domains $\Omega \subset [R, 1-R]^2 \subset D$ with the following property: for all $x \in D$ there exists an orthonormal system (e_1, e_2) and a function $\psi \in \mathcal{C}^s$ with $\|\psi\|_{\mathcal{C}^s} \leqslant M$, such that*

$$y \in \Omega \iff z_2 \leqslant \psi(z_1),$$

for any $y = x + z_1 e_1 + z_2 e_2$ with $|z_1|, |z_2| \leqslant R$.

Here, we have used the usual definition

$$\|\psi\|_{\mathcal{C}^s} = \sup_{0 \leqslant k \leqslant \lfloor s \rfloor} \|\psi^{(k)}\|_{L^\infty([-R,R])} + \sup_{s,t \in [-R,R]} |s-t|^{\lfloor s \rfloor - s} \left| \psi^{(\lfloor s \rfloor)}(s) - \psi^{(\lfloor s \rfloor)}(t) \right|,$$

for the Hölder norm. In the case of integer smoothness, we use the convention that $\mathcal{C}^s$ denotes functions with Lipschitz derivatives up to order $s-1$, so that in particular the case $s = 1$ corresponds to domains with Lipschitz boundaries.

Remark 3.4.2. *The condition $\Omega \subset [R, 1-R]^2$ imposing that Ω remains away from the boundary ∂D might be quite restrictive in some applications; instead, one can assume that the domains Ω and D are periodic, or symmetrize Ω with respect to ∂D.*

In what follows, we first show that all linear reconstruction methods suffer from an inherently limited rate of convergence. Then we introduce nonlinear reconstruction methods that can be analyzed based on the general principles exposed in Section 3.2 and Section 3.3, and are proved to reach better convergence rates.

We stress that nonlinear approaches in the applicative context (ii) of super-resolution have been intensively developed and studied; first by the introduction of non-quadratic regularization such as total variation or ℓ^1 norms in basis or frame expansions, nonlocal methods [73, 112, 127], and more recently by deep learning approaches such as convolution neural networks [37, 155, 160], which are empirically recognized as the current state of the art

Here, our perspective is different, closer to the applicative context (i) of numerical simulation. The goal is to locally recover on each cell an approximating function with simple analytic description, which allows to further evaluate the numerical flux at low cost by propagating this approximation. It typically elaborates on numerical techniques for subcell resolution [8] and linear interface reconstruction [128, 129, 132]. In addition, our approach comes with certified recovery bounds and convergence rates.

3.4.2 The failure of linear reconstruction methods

The most trivial linear reconstruction method consists in the piecewise constant approximation

$$\tilde{u} = \sum_{T \in \mathcal{T}} a_T(u) \chi_T. \tag{3.13}$$

The approximation rate of this reconstruction over the class $\mathcal{F}_{s,R,M}$ is as follows.

Proposition 3.4.3. *Let $u = \chi_\Omega \in \mathcal{F}_{s,R,M}$, its piecewise constant approximation $\tilde{u}$ by average values on each cell, defined in (3.13), satisfies*

$$\|\chi_\Omega - \tilde{u}\|_{L^q} \leqslant Ch^{\frac{1}{q}} = Cn^{-\frac{1}{2q}},$$

where the constant C depends on R and M.

Proof. Let $N = \lceil (\sqrt{2}R)^{-1} \rceil$, and partition the domain $D = [0,1]^2$ into N^2 squares of side $1/N$. Then each subsquare Q is contained in the set $\{x + z_1 e_1 + z_2 e_2,\ |z_1|, |z_2| \leqslant R\}$ from Definition 3.4.1, where x is the center of Q. Thus $\partial\Omega$ is the restriction of the graph of an M-Lipschitz function on Q, so its arc length is bounded by

$$|\partial\Omega \cap Q| \leqslant \operatorname{diam}(Q)\sqrt{1+M^2} \leqslant 2R\sqrt{1+M^2}.$$

As any curve of arclength h intersects at most four cells from $\mathcal{T}$, $\partial\Omega \cap Q$ intersects at most $4\lceil 2R\sqrt{1+M^2}/h \rceil$ cells, and summing over all subsquares, $\partial\Omega$ intersects at most $4N^2\lceil 2R\sqrt{1+M^2}/h \rceil$ cells. Denoting $\mathcal{T}_{\partial\Omega}$ the set of these cells, and observing that $u|_T \equiv a_T(u) \in \{0,1\}$ for $T \notin \mathcal{T}_{\partial\Omega}$, we get

$$\|\chi_\Omega - \tilde{u}\|_{L^q}^q = \sum_{T\in\mathcal{T}} \int_T |u - a_T(u)|^q \leqslant \sum_{T\in\mathcal{T}_{\partial\Omega}} |T| = h^2 |\mathcal{T}_{\partial\Omega}| \leqslant 24 \frac{\sqrt{1+M^2}}{R} h$$

for $h \leqslant R$, and this bound also holds for $h > R$ since $\|\chi_\Omega - \tilde{u}\|_{L^q}^q \leqslant 1$. □

The next result shows, for the particular case $q = 2$, that no better rate can actually be achieved by any linear method, regardless of the smoothness s of the boundary. We conjecture that a similar result holds for $1 \leqslant q \leqslant \infty$. This motivates the use of nonlinear recovery methods, which are the object of the next section.

We recall that the Kolmogorov n-width of a compact set S from some Banach space V is defined by

$$d_n(S)_V := \inf_{\dim(E)\leqslant n} \operatorname{dist}(S,E)_V,$$

where $\operatorname{dist}(S,E)_V := \max_{u\in S} \min_{v\in E} \|u - v\|_V$ and the infimum is taken over all finite dimensional spaces E of dimension at most n.

Proposition 3.4.4. *Let $s \geqslant 1$ be arbitrary. Then for R sufficiently small, and M sufficiently large, there exists $c > 0$ such that the Kolmogorov n-widths of the class $\mathcal{F}_{s,R,M}$ satisfy*

$$d_n(\mathcal{F}_{s,R,M})_{L^2} \geqslant cn^{-\frac{1}{4}}, \quad n \geqslant 1.$$

Proof. The proof of this result relies on similar lower bounds for dictionaries of d-dimensional ridge functions

$$\mathbb{P}_k^d := \{x \mapsto \sigma_k(\omega \cdot x + b) \ :\ \|\omega\|_2 = 1,\ c_1 \leqslant b \leqslant c_2\},$$

where $\sigma_k(t) := \max\{0, t\}^k$ is the so-called RELU-k function. Here, we work in the space $L^2(B)$ where B is an arbitrary ball of $\mathbb{R}^d$, and the constants (c_1, c_2) are taken as the inf and sup of $\omega \cdot x$ as $x \in B$ and $\|\omega\|_2 = 1$, respectively, that is we take all b such that the line discontinuity of the k-th derivative of $\sigma_k(\omega \cdot x + b)$ crosses the ball B. Theorem 9 from [145], which improves on earlier results from [110], shows that if

$$B_1(\mathbb{P}_k^d) := \overline{\left\{ \sum_{j=1}^n a_j g_j \ : \ n \in \mathbb{N},\ g_j \in \mathbb{P}_k^d,\ \sum_{j=1}^n |a_j| \leqslant 1 \right\}}$$

denotes the symmetrized convex hull of this dictionary (the closure being taken in $L^2(B)$), then

$$d_n(B_1(\mathbb{P}_k^d))_{L^2(B)} \geqslant c n^{-\frac{2k+1}{2d}}, \quad n \geqslant 1,$$

where c depends on k, d, and the diameter of B.

In our case of interest we work with the value $d = 2$ and $k = 0$, so that the ridge functions are simply the characteristic functions of half-planes. By convexity, we have

$$d_n(\mathbb{P}_0^2)_{L^2(B)} = d_n(B_1(\mathbb{P}_0^2))_{L^2(B)} \geqslant c n^{-\frac{1}{4}}.$$

We take for B the ball of center $(1/2, 1/2)$ and radius $1/4$, which is inside our domain $D = [0,1]^2$. It is then readily seen that for R small enough and M large enough, we can extend any ridge function $g \in \mathbb{P}_0^2$ into a characteristic function χ_Ω from $\mathcal{F}_{s,R,M}$, as illustrated in Figure 3.1.

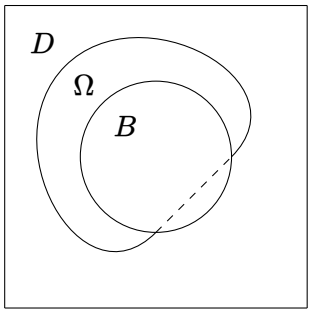


Figure 3.1: Example of extension of the indicator of a half-plane on B to the indicator of a smooth domain Ω on D

Observing that if E_D is a linear subspace of $L^2(D)$ of dimension at most n, its restriction E_B to B is a linear subspace of $L^2(B)$ of dimension at most n, and one has

$$\operatorname{dist}(\chi_\Omega, E_B)_{L^2(B)} \leqslant \operatorname{dist}(\chi_\Omega, E_D)_{L^2(D)}.$$

By infimizing, it follows that

$$d_n(\mathcal{F}_{s,R,M})_{L^2(D)} \geqslant d_n(\mathbb{P}_0^2)_{L^2(B)} \geqslant c n^{-\frac{1}{4}},$$

which concludes the proof. □

Remark 3.4.5. *The fact that we impose conditions on R and M in the above statement is natural since the class $\mathcal{F}_{s,R,M}$ becomes empty if R is not small enough and M not large enough, due to the fact that the sets Ω are assumed to be contained in the interior of D.*

Remark 3.4.6. *The above results are easily extended to higher dimension $d \geqslant 2$, with a similar definition for the class $\mathcal{F}_{s,R,M}$. The rate of approximation in L^q norm by piecewise constant functions on uniform partitions is then $n^{-\frac{1}{dq}}$, which in the case $q = 2$ is proved by a similar argument to be the best achievable by any linear reconstruction method. We conjecture that the same holds for more general $1 \leqslant q \leqslant \infty$.*

3.5 Shape recovery by nonlinear least-squares

3.5.1 Nonlinear reconstruction on a stencil

We now discuss a nonlinear reconstruction method for $u \in \mathcal{F}_{s,R,M}$, whose output $\tilde{u}$ is the indicator of a domain $\tilde{\Omega}$ with polygonal boundary : on each cell T, the domain $\tilde{\Omega}$ coincides with a certain half plane. In order to define the delimiting line we only use the average values of u on a 3×3 stencil of cells centered at T.

We assume that $h < R$, so that Ω does not intersect the boundary cells $T_{i,j}$ with i or j in $\{1, L\}$, and fix indices $1 < i, j < L$. For the cell $T = T_{i,j}$, denote $\overline{x} = ((i - \frac{1}{2})h, (j - \frac{1}{2})h)$ its center, and

$$S = [(i-2)h, (i+1)h] \times [(j-2)h, (j+1)h] = \bigcup_{i-1 \leqslant i' \leqslant i+1,\, j-1 \leqslant j' \leqslant j+1} T_{i'j'}$$

the stencil composed of T and its eight neighboring cells. We define the nonlinear approximation space

$$V_2 := \left\{ \chi_{\vec{n} \cdot (x - \overline{x}) \geqslant c} \ : \ \vec{n} \in \mathbb{S}^1, c \in \mathbb{R} \right\}, \tag{3.14}$$

which is a two-parameter family as each function is determined by $\arg \vec{n} \in [0, 2\pi)$ and $c \in \mathbb{R}$, where $\arg \vec{n}$ is the angle of $\vec{n}$ with respect to the horizontal axis.

Here, our measurements are the average values of u on the cells contained in S

$$\ell(u) = (a_{T'}(u))_{T' \subset S} \in \mathbb{R}^9.$$

In order to find a reconstruction of u in V_2 based on these measurements, we need an inverse stability property of the form (3.7). This is not possible here, since ℓ cancels on all functions $\chi_\Omega \in V_2$ with $\Omega \cap S = \emptyset$. We therefore restrict the nonlinear family V_2, and consider only indicators of half-planes whose boundary passes through the central cell T:

$$V_{2,T} := \left\{ \chi_\Omega \in V_2, \partial\Omega \cap T \neq \emptyset \right\} = \left\{ \chi_{\vec{n} \cdot (x - \overline{x}) \geqslant c},\ \vec{n} \in \mathbb{S}^1, |c| \leqslant \frac{h}{2} |\vec{n}|_1 \right\}. \tag{3.15}$$

In this setting, we prove the existence of the following stability constants for $V = L^1(S)$ and $Z = \ell^1$, which is the best norm on $\mathbb{R}^m$ in view of Theorem 3.3.2. For notational simplicity, we omit the reference to Z in these constants.

Proposition 3.5.1. *One has*

$$\|\ell(u)\|_1 \leqslant \alpha \|u\|_{L^1(S)}, \quad u \in L^1(D), \tag{3.16}$$

and

$$\|u - v\|_{L^1(S)} \leqslant \mu \|\ell(u - v)\|_1, \quad u, v \in V_{2,T}, \tag{3.17}$$

where $\alpha = h^{-2}$ and $\mu = \frac{3}{2}h^2$ are the optimal constants.

The proof of the stability property (3.16) is trivial since on each cell

$$|a_{T'}(u)| \leqslant |T'|^{-1} \|u\|_{L^1(T')} = h^{-2} \|u\|_{L^1(T')},$$

with equality in case u does not change sign. The proof of the inverse stability (3.17) is quite technical and left to the appendix.

Given the noisy observation

$$z = \ell(u) + \eta \in \mathbb{R}^9,$$

we define the estimator of u on the cell T by

$$\tilde{u}_T \in \arg\min_{v \in V_2} \|z - \ell(v)\|_1. \tag{3.18}$$

Here we minimize over all V_2, that is on all indicators of half planes, but we note that we may restrict to half-planes whose boundary passes through the stencil S.

The following result, which uses Proposition 3.5.1, shows that its distance to u in $L^1(T)$ is comparable to the error between u and its best approximation in the $L^1(S)$ norm

$$\overline{u}_S := \arg\min_{v \in V_2} \|u - v\|_{L^1(S)}.$$

Lemma 3.5.2. *For all $u \in \mathcal{F}_{s,R,M}$, one has*

$$\|u - \tilde{u}_T\|_{L^1(T)} \leqslant C_1 \|u - \overline{u}_S\|_{L^1(S)} + C_2 \|\eta\|_p,$$

where $C_1 = 1 + 2\alpha\mu = 4$ and $C_2 = 2\beta\mu = 3^{3-\frac{2}{p}} h^2$, with α, μ as in (3.5.1), and $\beta = 9^{1-\frac{1}{p}}$ the maximal ratio between ℓ^p and ℓ^1 norm in $\mathbb{R}^9$.

Proof. We distinguish two cases:

- If $\tilde{u}_T \in V_{2,T}$ and $\overline{u}_S \in V_{2,T}$, that is, both boundaries pass through the central cell T, we apply (3.9) together with Proposition 3.5.1

$$\begin{aligned}\|u-\tilde{u}_T\|_{L^1(T)} \leqslant \|u-\tilde{u}_T\|_{L^1(S)} &\leqslant C_1 \min_{v\in V_{2,T}} \|u-v\|_{L^1(S)} + C_2\|\eta\|_p \\ &= C_1\|u-\overline{u}_S\|_{L^1(S)} + C_2\|\eta\|_p\end{aligned}$$

with $C_1 = 1+2\alpha\mu$ and $C_2 = 2\beta\mu$.

- Otherwise, either $\tilde{u}_T$ or $\overline{u}_S$ has constant value 0 or 1 on T, so $\tilde{u}_T - \overline{u}_S$ has constant sign on T, and thus

$$\begin{aligned}\|\overline{u}_S-\tilde{u}_T\|_{L^1(T)} &= h^2|a_T(\tilde{u}_T-\overline{u}_S)| \leqslant h^2\|\ell(\tilde{u}_T-\overline{u}_S)\|_1 \\ &\leqslant h^2(\|\ell(\overline{u}_S)-z\|_1 + \|\ell(\tilde{u}_T)-z\|_1) \\ &\leqslant 2h^2\|\ell(\overline{u}_S)-z\|_1 \leqslant 2h^2\|\ell(\overline{u}_S-u)\|_1 + 2h^2\|\eta\|_1 \\ &\leqslant 2\|u-\overline{u}_S\|_{L^1(S)} + 2h^2\beta\|\eta\|_p.\end{aligned}$$

By triangle inequality, it follows that

$$\|u-\tilde{u}_T\|_{L^1(T)} \leqslant 3\|u-\overline{u}_S\|_{L^1(S)} + 2h^2\beta\|\eta\|_p,$$

which has better constants than in the estimate obtained in the first case, since the constant C_0 is larger than 1. □

The order of the best local approximation error $\|u-\overline{u}_S\|_{L^1(S)}$ that appears as a bound for the reconstruction error $\|u-\tilde{u}_T\|_{L^1(T)}$ depends on the smoothness of the boundary, as expressed in the following lemma.

Lemma 3.5.3. *For all $u \in \mathcal{F}_{s,R,M}$, with $R \geqslant \frac{3}{\sqrt{2}}h$, one has*

$$\|u-\overline{u}_S\|_{L^1(S)} \leqslant M(3\sqrt{2}h)^{\min(s,2)+1}.$$

Proof. We apply the definition of $\mathcal{F}_{s,R,M}$ at point $\overline{x}$: as $R \geqslant \frac{3}{\sqrt{2}}h$, the stencil S is contained in the domain

$$\{\overline{x} + z_1e_1 + z_2e_2,\ |z_1|,|z_2| \leqslant R\},$$

so $u|_S$ is the indicator of a domain delimited by a $\mathcal{C}^s$ function ψ, with $\|\psi\|_{\mathcal{C}^s} \leqslant M$. From the definition of $\mathcal{C}^s$, there exists an affine function ξ such that

$$|\psi(z_1)-\xi(z_1)| \leqslant M(3\sqrt{2}h)^{\min(s,2)}, \quad |z_1| \leqslant \frac{3}{\sqrt{2}}h.$$

Then the function $v : \overline{x} + z_1e_1 + z_2e_2 \mapsto \chi_{z_2\leqslant\xi(z_1)}$ belongs to V_2, and we have

$$\|u-\overline{u}_S\|_{L^1(S)} \leqslant \|u-v\|_{L^1(S)} \leqslant M(3\sqrt{2}h)^{\min(s,2)+1}.$$

□

3.5.2 Global nonlinear reconstruction

We now consider the process of recovering $u \in \mathcal{F}_{s,R,M}$ globally from its data

$$z = \ell(u) + \eta,$$

where now $\ell(u) := (a_T(u))_{T\in\mathcal{T}} \in \mathbb{R}^n$ and $\eta \in \mathbb{R}^n$ is the noise vector. Applying to each inner cell $T \in \mathcal{T}$ the previous reconstruction procedure based on the 3×3 stencil S centered at T, we obtain a global recovery $\tilde{u} = \tilde{u}(z)$ such that

$$\tilde{u}|_T = \tilde{u}_T|_T, \quad T = T_{i,j} \in \mathcal{T}, \quad 1 < i, j < L,$$

where $\tilde{u}_T$ is the local estimator from (3.18). On the boundary cells $T = T_{i,j}$ with i or j in $\{1, L\}$, $u|_T$ is zero by Definition 3.4.1 so we simply set $\tilde{u}|_T = 0$. Note that $\tilde{u}$ is of the form

$$\tilde{u} = \chi_{\tilde{\Omega}},$$

where $\tilde{\Omega}$ has piecewise linear boundary with respect to the mesh $\mathcal{T}$. The following result gives a global approximation bound, which confirms the improvement over linear methods when $s > 1$.

Theorem 3.5.4. *For all $u \in \mathcal{F}_{s,R,M}$, one has*

$$\|u - \tilde{u}\|_{L^q(D)} \leqslant C_1 n^{-\frac{\min(1,s/2)}{q}} + C_2 n^{-\frac{1}{pq}} \|\eta\|_p^{\frac{1}{q}}.$$

Proof. First notice that if the result is proved for $p = q = 1$, as $u - v$ has values in $\{-1, 0, 1\}$,

$$\|u - v\|_{L^q(D)}^q = \|u - v\|_{L^1(D)} \leqslant C_1 n^{-1} + C_2 n^{-1} \|\eta\|_1 \leqslant \left(C_1^{\frac{1}{q}} n^{-\frac{1}{q}} + C_2^{\frac{1}{q}} n^{-\frac{1}{pq}} \|\eta\|_p^{\frac{1}{q}} \right)^q,$$

so it suffices treat the case $p = q = 1$.

By an argument similar to the proof of Proposition 3.4.3, $\partial\Omega$ intersects at most $16N^2\lceil 2R\sqrt{1+M^2}/h \rceil$ stencils of nine cells. Using the fact that $u = \overline{u}_S$ is a constant on any other stencil, we get

$$\begin{aligned}
\|u - \tilde{u}\|_{L^1(D)} &= \sum_{T \text{ inner cell}} \|u - \tilde{u}\|_{L^1(T)} \\
&\leqslant \sum_{T \text{ inner cell}} (1 + 2\alpha\mu)\|u - \overline{u}\|_{L^1(S)} + 2\beta\mu\|\eta\|_{\ell^1(S)} \\
&\leqslant 16N^2 \left\lceil \frac{2R\sqrt{1+M^2}}{h} \right\rceil M(3\sqrt{2}h)^{\min(s,2)+1} + 18\beta\mu\|\eta\|_1 \\
&\leqslant C_1 h^{\min(s,2)} + C_2 h^2 \|\eta\|_1.
\end{aligned}$$

We conclude by recalling that $n = h^{-2}$. □

Remark 3.5.5. *Here the convergence rate for the noiseless term $n^{-\frac{\min(1,s/2)}{q}}$ is limited due to the use of polygonal domains in the reconstruction. So the best approximation rate $h^{\frac{2}{q}} = n^{-\frac{1}{q}}$ is already attained for $\mathcal{C}^2$ boundaries. When the smoothness parameter s is larger than 2, better rates $n^{-\frac{s}{2q}}$ should be reachable if we use non-linear approximation spaces that are richer than the space V_2, for example indicator functions of domains with boundary that have a higher order polynomial description rather than straight lines. Of course, the stable identification of these approximants in the sense of (3.7) might require stencils that are of larger size than 3×3.*

Remark 3.5.6. *If $\|\eta\|_\infty \leqslant \frac{1}{9}$, then $\tilde{u}$ is exactly equal to u on any cell whose corresponding stencil does not intersect $\partial\Omega$, so the error is concentrated on $\mathcal{O}(\sqrt{n})$ cells, leading to an improved rate $n^{-\frac{p+1}{2pq}}$ instead of $n^{-\frac{1}{pq}}$ for the noise term.*

3.5.3 Numerical illustration

We study the behavior of the above discussed linear and non-linear recovery methods from cell averages for the particular target function $u = \chi_\Omega$, with Ω a slightly decentered disk of radius $r = 0.325$.

The linear method consists of the piecewise constant approximation (3.13), referred to as *PiecewiseConstant*. As to the nonlinear method, for the local best fit problem, we use the ℓ^2 norm on $\mathbb{R}^9$ instead of the ℓ^1 norm. By norm equivalence on $\mathbb{R}^9$, the same convergence results can be proved to hold with different constants. This method, which we refer to as *LinearInterface*, does not ensure consistency of the reconstruction in the sense that $a_T(\tilde{u}) = a_T(u)$. One way to approach this consistency property is to modify the ℓ^2 norm by putting a large weight on the central cell. We refer to this variant as *LinearInterfaceCC*, here taking the weight 100.

In the implementation, a function $v \in V_2$ is parametrized by the pair (r, θ) where $r \geqslant 0$ is the offset distance between the center $\overline{x}$ of the central cell T and the linear interface and $\theta \in [0, 2\pi[$ is the angle such that the unit normal to the interface is $e_\theta = (\cos(\theta), \sin(\theta))$. In other words, v is of the form

$$v = v_{r,\theta} := \chi_{|\langle x - \overline{x}, e_\theta \rangle| \leqslant r}.$$

As we have seen that only intefaces passing through the stencil S should be considered, we may restrict r to $[0, \overline{r}]$ where $\overline{r} := \sqrt{3/2}h$. Then the *LinearInterface* and *LinearInterfaceCC* procedures read as follows.

Figure 3.2 shows the convergence rates of the three methods in the L^1 norm. The expected h^{-2} decay is observed in both non-linear methods while the linear method lays behind with a decay rate of h^{-1}. It is relevant to note that although both non-linear methods benefit from the same rate, the associated constants differ by an order of magnitude, showing the practical improvement gained by imposing consistency. This improvement is also visible on Figure 3.3 which shows that in the LinearInterface method, the interfaces that minimize the l_2 error on the nine surrounding cells lay always inside the circle as the curvature of the

Algorithme 1 : LinearInterface and LinearInterfaceCC

Input : $\ell(u) = (a_{T'}(u))_{T' \subset S} \in \mathbb{R}^9$ `// The nine cell averages`
Output : (r^*, θ^*) `// The estimated parameters of the line interface`
$= \operatorname{argmin}\Big\{ \sum_{T' \subset S} |a_{T'}(v_{r,\theta}) - a_{T'}|^2 + c\,|a_T(v_{r,\theta}) - a_T|^2 \ : \ (r,\theta) \in [0, \bar{r}] \times [0, 2\pi[\Big\}$
`//` $c = 0$ `in LinearInterface,` $c = 100$ `in LinearInterfaceCC`
`//` T `is the central cell of the stencil` S

boundary pushes them towards the center. On the contrary, LinearInterfaceCC seems to find the right compromise between sticking to the cell average while capturing at the same time the curvature trend hinted by the surrounding cell averages.

For more details on the implementation: https://github.com/agussomacal/SubCellResolution

3.6 Relation to compressed sensing

3.6.1 Compressed sensing and best n-term approximation

In this section we discuss the application of our setting to the sparse recovery of large vectors from a few linear observations. We thus take

$$V = \mathbb{R}^N,$$

equipped with some given norm $\|\cdot\|_V$ of interest. The linear measurements of $u = (u_1, \dots, u_N)^\top \in \mathbb{R}^N$ are given by

$$(\ell_1(u), \dots, \ell_m(u))^\top = \Phi u,$$

where Φ is an $m \times N$ measurement matrix, with typically $m \ll N$.

The topic of compressed sensing deals with sparse recovery of u from such measurements, that is, searching to recover an accurate approximation to u by a vector with only a few non-zero components. We refer to [42] for some first highly celebrated breakthrough results and to [74] for a general treatment.

We define the nonlinear space of n-sparse vectors as

$$V_n := \Big\{ u \in \mathbb{R}^N \ : \ \|u\|_0 := \#\{i \ : \ u_i \neq 0\} \leqslant n \Big\},$$

and the best n-term approximation error in the V norm as

$$e_n(u)_V := \min_{v \in V_n} \|u - v\|_V.$$

One natural question is to understand for which type of measurement matrices Φ does the noise-free measurement $y = \Phi u$ contain enough information, in order to recover any u up to an error $e_n(u)_V$. In other words, one asks if there exists a recovery map $R : \mathbb{R}^m \to \mathbb{R}^N$

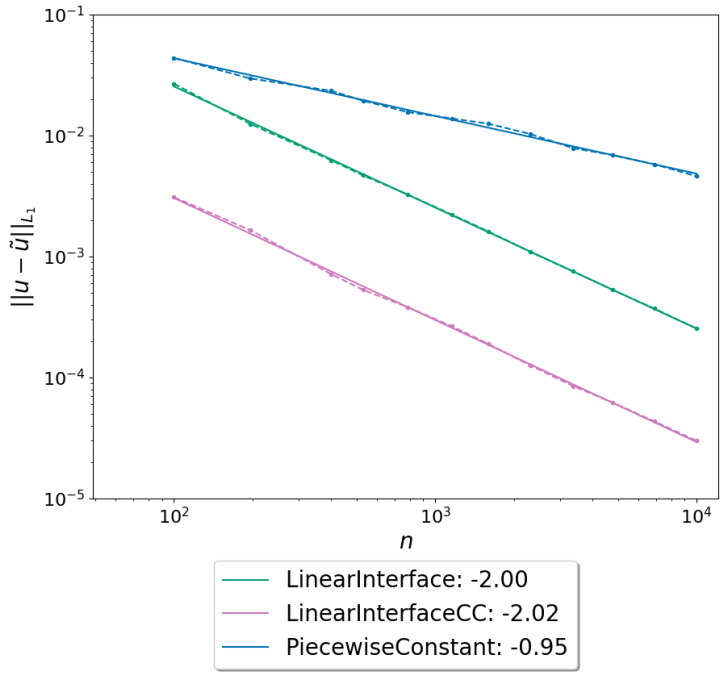


Figure 3.2: Convergence curves for the linear and nonlinear recovery methods

such that one has the *instance optimality property* at order n

$$\|u - R(\Phi u)\|_V \leqslant C_0 e_n(u)_V, \quad u \in \mathbb{R}^N, \tag{3.19}$$

with C_0 a fixed constant, which we denote by $IOP(n, C_0)$. This question has been answered in [46] in terms of the null space $\mathcal{N} := \{v \in \mathbb{R}^N \ : \ \Phi v = 0\}$. We say that Φ satisfies the *null space property* at order k with constant C_1, denoted by $NSP(k, C_1)$ if and only if

$$\|v\|_V \leqslant C_1 e_k(v)_V, \quad v \in \mathcal{N}. \tag{3.20}$$

This property quantifies how much vectors from the null space can be concentrated on a few coordinates. One main result of [46] is the equivalence between IOP at order n and NSP at order $2n$ in the following sense.

Theorem 3.6.1. *One has* $IOP(n, C_0) \Rightarrow NSP(2n, C_0)$, *and conversely* $NSP(2n, C_1) \Rightarrow IOP(n, 2C_1)$.

One natural question is whether matrices Φ with such properties can be constructed with a number of rows/measurements m barely larger than n. As we recall further the answer to this question is strongly tied to the norm V used on $\mathbb{R}^N$.

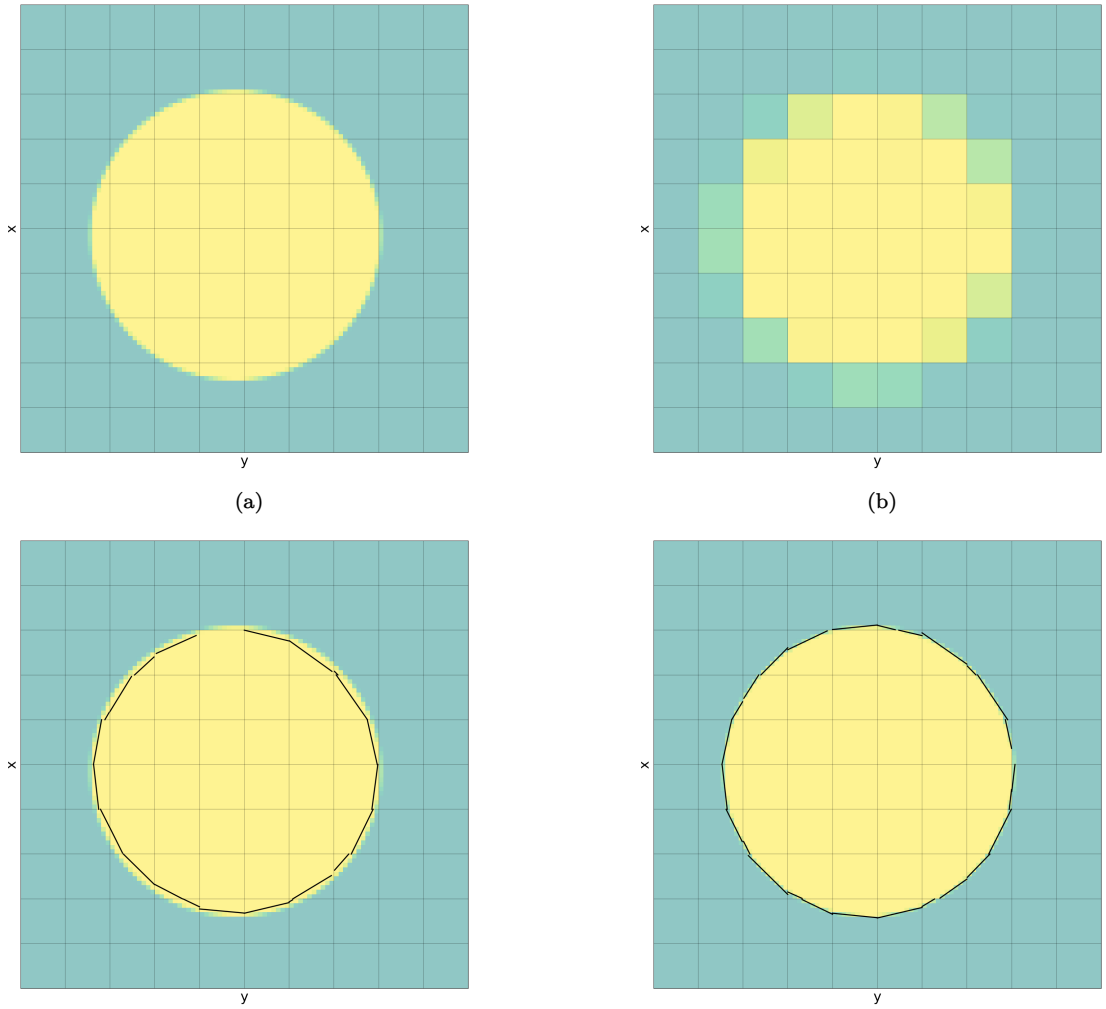


Figure 3.3: (a) The target function, (b) its recovery by PiecewiseConstant showing the cell-average data, and the recovered boundaries by (c) LinearInterface and (d) LinearInterfaceCC methods

3.6.2 Stability and the null space property

The nonlinear estimation results that we have obtained in Section 3.2 and Section 3.3 can be applied to the setting of sparse recovery, offering us a different vehicle than the null space property to establish instance optimality.

In the present setting, for a given norm $\|\cdot\|_Z$, the stability property (3.6) takes the form

$$\|\Phi u\|_Z \leqslant \alpha_Z \|u\|_V, \quad u \in \mathbb{R}^N \tag{3.21}$$

and the inverse stability property (3.7) takes the form

$$\|v\|_V \leqslant \mu_Z \|\Phi v\|_Z, \quad v \in V_{2n}, \tag{3.22}$$

since for sparse vectors we have $V_n^{\mathrm{diff}} = V_n - V_n = V_{2n}$. We refer to these properties as $S(\alpha_Z)$ and $IS(2n, \mu_Z)$, respectively.

Application of Theorem 3.2.3 in the noiseless case immediately gives us that the nonlinear best fit recovery $R(\Phi u) = \tilde{u}$ satisfies the instance optimality bound (3.19) with constant $C_0 = 1 + 2\alpha_Z \mu_Z$. In other words

$$S(\alpha_Z) \text{ and } IS(2n, \mu_Z) \Rightarrow IOP(n, C_0), \qquad C_0 = 1 + 2\alpha_Z \mu_Z. \tag{3.23}$$

The following result shows that (S, IS) is actually equivalent to NSP, and thus to IOS, in the sense that a converse result holds when $\|\cdot\|_Z$ is chosen to be the Riesz norm (3.10).

Theorem 3.6.2. *For any norm $\|\cdot\|_Z$, one has*

$$S(\alpha_Z) \text{ and } IS(2n, \mu_Z) \Rightarrow NSP(2n, C_1), \qquad C_1 = 1 + \alpha_Z \mu_Z. \tag{3.24}$$

Conversely, let $\|\cdot\|_W$ be the Riesz norm so that $\|\Phi u\|_W = \min_{\Phi v = \Phi u} \|v\|_V$, then

$$NSP(2n, C_1) \Rightarrow S(\alpha_W) \text{ and } IS(2n, \mu_W), \qquad \alpha_W = 1 \text{ and } \mu_W = 1 + C_1. \tag{3.25}$$

Proof. Assume that $S(\alpha_Z)$ and $IS(2n, \mu_Z)$ hold. Let $v \in \mathcal{N}$ and $\tilde{v}$ its best approximation in V_{2n}, then

$$\begin{aligned} \|v\|_V &\leqslant \|v - \tilde{v}\|_V + \|\tilde{v}\|_V \\ &\leqslant e_{2n}(v)_V + \mu_Z \|\Phi \tilde{v}\|_W \\ &= e_{2n}(v)_V + \mu_Z \|\Phi(v - \tilde{v})\|_W \leqslant (1 + \alpha_Z \mu_Z) e_{2n}(x)_V. \end{aligned}$$

This shows that $NSP(2n, C_1)$ holds with $C_1 = 1 + \alpha_Z \mu_Z$.

Conversely, assume that $NSP(2n, C_1)$ holds. From the definition of the Riesz norm, it is immediate that $S(\alpha_W)$ holds with $\alpha_W = 1$. For $v \in V_{2n}$, let $\tilde{v}$ be the minimizer of $\min_{\Phi \tilde{v} = \Phi v} \|\tilde{v}\|_V$. Then, one has

$$\|v\|_V \leqslant \|\tilde{v}\|_V + \|v - \tilde{v}\|_V \leqslant \|\tilde{v}\|_V + C_1 \sigma_{2n}(v - \tilde{v})_V \leqslant (1 + C_1) \|\tilde{v}\|_V,$$

by using v as a sparse approximation to $v - \tilde{v}$. Since $\|\tilde{v}\|_V = \|\Phi v\|_W$, this shows that $S(2n, \mu_W)$ holds with $\mu_W = 1 + C_1$. □

3.6.3 The case of ℓ^p norms

The range of m allowing the properties to be fulfilled is best understood in the case of the ℓ^p norms, that is $\|\cdot\|_V = \|\cdot\|_p$, as discussed in [46] which points out a striking difference between the case $p = 2$ and $p = 1$:

1. In the case $p = 2$, it is proved that $NSP(2, C_1)$ cannot hold unless $N \leqslant C_1^2 m$. In other words, instance optimality in ℓ^2 even at order $n = 1$ requires a number of measurements that is proportional to the full space dimension.

2. In the more favorable case $p = 1$, it is proved that for matrices which satisfy the ℓ^2-RIP property of order $3n$

$$(1 - \delta)\|v\|_2^2 \leqslant \|\Phi v\|_2^2 \leqslant (1 + \delta)\|v\|_2^2, \quad v \in V_{3n},$$

   with parameter $0 < \delta < \frac{(\sqrt{2}-1)^2}{3}$, the $NSP(2n, C_1)$ holds with C_1 depending on δ. Such matrices are known to exists with $m \sim n \log(N/n)$ rows.

Our setting based on the stability properties S and IS applies more naturally to a different class of matrices built from graphs, which is also known to be well adapted for sparse recovery in the ℓ^1 norm. A bipartite graph with (N, m) left and right vertices, and of left degree d, is an (l, ε)-graph expander if

$$|X| \leqslant l \Rightarrow |N(X)| \geqslant d(1 - \varepsilon)|X|, \quad X \subset \{1, \dots, N\},$$

where $N(X) \subset \{1, \dots, m\}$ is the set of vertices connected to X. We necessarily have $|N(X)| \leqslant d|X|$, and $(1 - \varepsilon)dl \geqslant m$. From [43], it is known that there exists a $(2n, \frac{1}{2})$-graph expander with $d \sim \log \frac{N}{n}$ and $m \sim nd \sim n \log(N/n)$.

Now denote $\Phi \in \{0, 1\}^{m \times N}$ the adjacency matrix of this graph, so that each column of Φ has d nonzero entries. Then

$$\|\Phi x\|_1 \leqslant d\|x\|_1, \quad x \in \mathbb{R}^N,$$

and

$$\|\Phi x\|_1 \geqslant d(1 - \varepsilon)\|x\|_1, \quad x \in V_{2n}.$$

Therefore $S(\alpha_1)$ and $IS(2n, \mu_1)$, hold with $\alpha_1 = d$ and $\mu_1 = \frac{1}{d(1-\varepsilon)} = \frac{2}{d}$, which by (3.24) and (3.23) gives $NSP(2n, C_1)$ with $C_1 = 3$ and $IOP(n, C_0)$ with $C_0 = 5$.

3.A Proof of Proposition 3.5.1

The proof contains 15 cases, represented on a tree in Figure 3.4. These cases correspond to different geometric situations, up to certain symmetries that leave the final relevant quantities $\|\ell(w)\|_1$ and $\|w\|_{L^1(S)}$ unchanged.

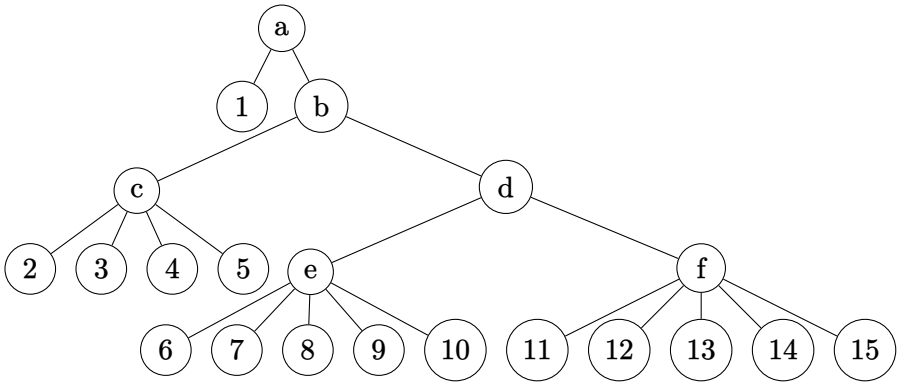


Figure 3.4: Structure of the proof, each leaf corresponds to a different case, and each node contains a general treatment valid for all its sons

Node a: Take $w = u - v \in V_{2,T}^{\text{diff}}$, with $u, v \in V_{2,T}$, and denote $\vec{n}_u$, $\vec{n}_v$ and c_u, c_v the corresponding unit vectors and offsets from (3.15) of $V_{2,T}$. Recalling that $\overline{x} = (\overline{x}_1, \overline{x}_2)$ is the center of S, we also denote

$$\Delta_u = \{x \in \mathbb{R}^2, (x - \overline{x}) \cdot \vec{n}_u = c_u\}$$

the delimiting line between $\{u = 0\}$ and $\{u = 1\}$, and define Δ_v in a similar way.

Case 1: If $\vec{n}_u = \vec{n}_v = \vec{n}$, we have

$$w = \begin{cases} \chi_{c_u \leqslant \vec{n} \cdot (x - \overline{x}) < c_v} & \text{if } c_u \leqslant c_v \\ -\chi_{c_v \leqslant \vec{n} \cdot (x - \overline{x}) < c_u} & \text{otherwise} \end{cases}$$

so w has constant sign, which implies $\|w\|_{L^1(S)} = h^2 \|\ell(w)\|_1$.

Node b: In all other cases, the cones

$$\mathcal{C}_+ = \{x \in \mathbb{R}^2,\, w(x) = 1\} \quad \text{and} \quad \mathcal{C}_- = \{x \in \mathbb{R}^2,\, w(x) = -1\}$$

are non-empty, and we can define the external bisector

$$\Delta = \{x \in \mathbb{R}^2,\, (\vec{n}_u - \vec{n}_v) \cdot (x - \overline{x}) = c_u - c_v\},$$

which is the line of symmetry between $\mathcal{C}_+$ and $\mathcal{C}_-$. We also denote

$$\mathcal{C} = \mathcal{C}_+ \cup \mathcal{C}_- = \{x \in \mathbb{R}^2, |w(x)| = 1\}.$$

Observing that

$$\|w\|_{L^1(S)} = |S \cap \mathcal{C}| \tag{3.26}$$

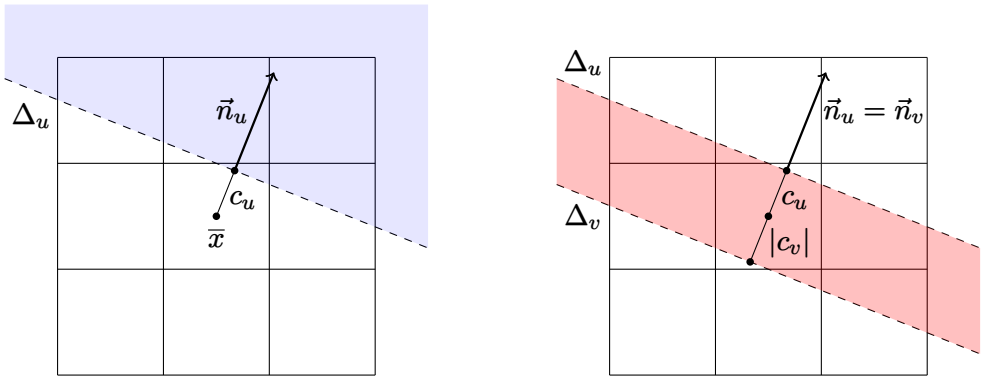


Figure 3.5: Left: 3×3 stencil S, with $\overline{x}$ its center, and an example of function $u \in V_{2,T}$ with directing vector $\vec{n}_u$ and offset $c_u > 0$. Here the dotted line corresponds to Δ_u, and the shaded region to $u = 1$, while $u = 0$ elsewhere. Right: Representation of Case 1 ($\vec{n}_u = \vec{n}_v$), here $c_v < 0 < c_u$ so $w = -1$ on the shaded region and $w = 0$ elsewhere

and

$$\|\ell(w)\|_1 = h^{-2} \sum_{T \subset S} \Big| |T \cap \mathcal{C}_+| - |T \cap \mathcal{C}_-| \Big|, \tag{3.27}$$

the stability property (3.17) can be rewritten as

$$|S \cap \mathcal{C}| \leqslant \frac{3}{2} \sum_{T \subset S} \Big| |T \cap \mathcal{C}_+| - |T \cap \mathcal{C}_-| \Big| = \frac{3}{2} \left(|S \cap \mathcal{C}| - 2 \sum_{T \subset S} \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|) \right),$$

or equivalently

$$|S \cap \mathcal{C}| \geqslant 6 \sum_{T \subset S} \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|). \tag{3.28}$$

Up to a rotation of S by a multiple of $\frac{\pi}{2}$, we may assume without loss of generality that

$$\arg(\vec{n}_u - \vec{n}_v) \in \left[\frac{\pi}{4}, \frac{3\pi}{4} \right],$$

that is, Δ is at an angle of at most $\frac{\pi}{4}$ with the horizontal axis, and $\mathcal{C}_+$ lies above Δ. Take $(\vec{e}_1, \vec{e}_2)$ the canonical basis of $\mathbb{R}^2$.

Node c: Consider the situation where $(\vec{n}_u \cdot \vec{e}_2)(\vec{n}_v \cdot \vec{e}_2) > 0$. As $\vec{n}_u \neq \vec{n}_v$ and $\vec{n}_u \neq -\vec{n}_v$, the lines Δ_u and Δ_v intersect at one point $X \in \mathbb{R}^2$. Moreover, the above condition implies $X + \vec{e}_2 \notin \mathcal{C}$. Using the fact that $|\arg(\Delta)| \leqslant \frac{\pi}{4}$, we also get $X + \vec{e}_1 \notin \mathcal{C}$.

Up to a symmetry with respect to the vertical axis, we can assume that $\mathcal{C}_+$ is included in the quadrant $X + \mathbb{R}^2_+$. Now consider a cell $T \subset S$ such that $\min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|) \neq 0$, and take points $x \in T \cap \mathcal{C}_-$ and $y \in T \cap \mathcal{C}_+$. As $x_1 \leqslant X_1 \leqslant y_1$ and $x_2 \leqslant X_2 \leqslant y_2$, we get

$X \in T$, so there is at most one such cell T, and inequality (3.28) reduces to

$$|S \cap \mathcal{C}| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

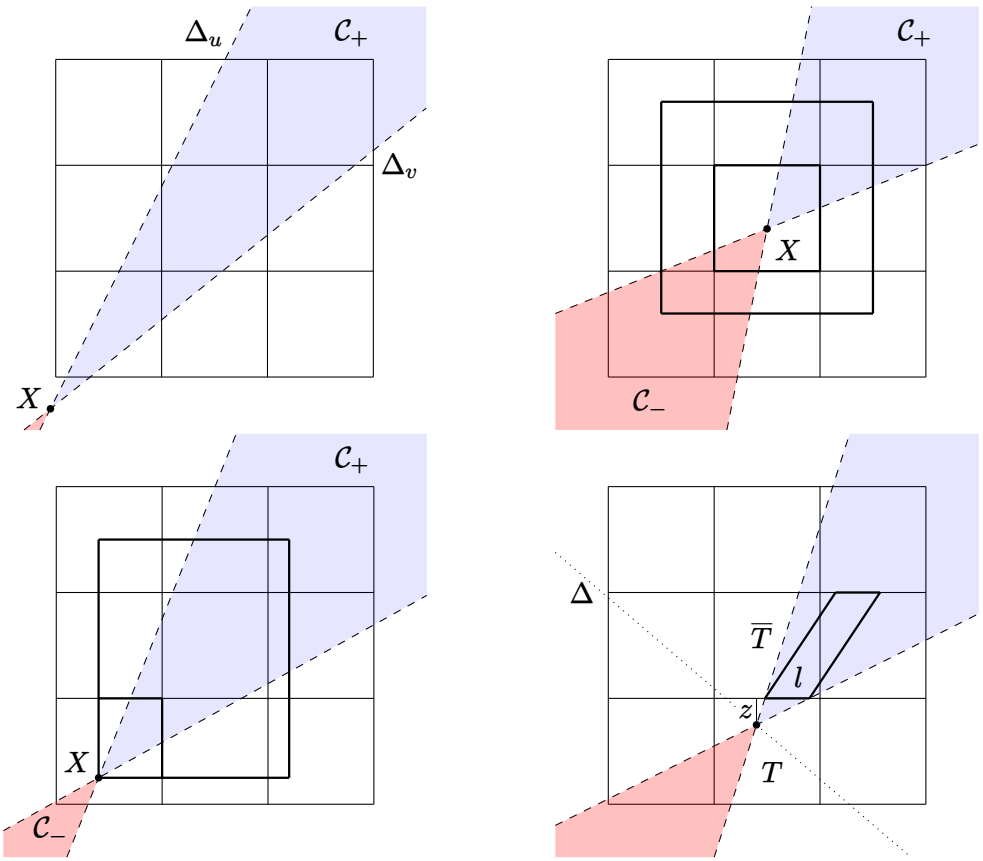


Figure 3.6: Cases 2, 3, 4, and 5

Case 2: If $X \notin S$, then w has constant sign on S, so $\|w\|_{L^1(S)} = h^2 \|\ell(w)\|_1$.

Case 3: If X is in the central cell T, the dilation of T with respect to X by a factor 2 is a subset of S, and the image of $\mathcal{C} \cap T$ is in $\mathcal{C} \cap S$, so

$$|S \cap \mathcal{C}| \geqslant 4|T \cap \mathcal{C}| \geqslant 8 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

Case 4: If X is in the lower left cell T, the dilation of $T \cap \mathcal{C}_+$ with respect to X by a factor 3 is in $S \cap \mathcal{C}_+$, so

$$|S \cap \mathcal{C}| \geqslant |S \cap \mathcal{C}_+| \geqslant 9|T \cap \mathcal{C}_+| \geqslant 9 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

The same argument holds with $\mathcal{C}_-$ instead of $\mathcal{C}_+$ when X is in the upper right cell. Moreover,

as Δ_u and Δ_v go through the central cell, X may not be in the upper left or lower right cells.

Case 5: If X is in the lower central cell T, denote $l = |\partial T \cap \mathcal{C}_+| \in (0, h)$ the distance between Δ_u and Δ_v when they pass from T to the central cell $\overline{T}$, and $z = \operatorname{dist}(X, \overline{T}) \in (0, h)$ the depth of the point of intersection. Then

$$|T \cap \mathcal{C}_+| = \frac{zl}{2} \quad \text{and} \quad |T \cap \mathcal{C}_-| \leqslant \frac{zl}{2} \left(\frac{h-z}{z} \right)^2,$$

so $\min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|) \leqslant \frac{hl}{4}$. On the other hand, the parallelogram of base $\partial T \cap \mathcal{C}_+$, of height h, and with sides orthogonal to Δ belongs to $(S \setminus T) \cap \mathcal{C}_+$ (it does not escape to the right of S because Δ is close to the horizontal axis, so the sides of the parallelogram are at an angle at most $\frac{\pi}{4}$ with the vertical axis), and has an area hl, which proves that

$$|\mathcal{C} \cap S| \geqslant hl + |\mathcal{C}_+ \cap T| + |\mathcal{C}_- \cap T| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

A similar construction can be applied to the remaining cases where X is in the upper central, central left or central right cell, which concludes the proof for Node c.

Node d: If now $(\vec{n}_u \cdot \vec{e}_2)(\vec{n}_v \cdot \vec{e}_2) \leqslant 0$, as $\arg(\vec{n}_u - \vec{n}_v) \in \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$, we get $\vec{n}_u \cdot \vec{e}_2 \geqslant 0 \geqslant \vec{n}_v \cdot \vec{e}_2$. Observe that $\mathcal{C}_+ + \vec{e}_2 \subset \mathcal{C}_+$ since for all $x \in \mathcal{C}_+$,

$$(x + \vec{e}_2 - \overline{x}) \cdot \vec{n}_u \geqslant (x - \overline{x}) \cdot \vec{n}_u \geqslant c_u \quad \text{and} \quad (x + \vec{e}_2 - \overline{x}) \cdot \vec{n}_v \leqslant (x - \overline{x}) \cdot \vec{n}_v < c_v.$$

In the same way, $\mathcal{C}_- - \vec{e}_2 \subset \mathcal{C}_-$. We now divide S into columns separated by the vertical boundaries between cells, and in addition by vertical lines where Δ intersects the two horizontal lines separating cells of S, as illustrated in Figure 3.7.

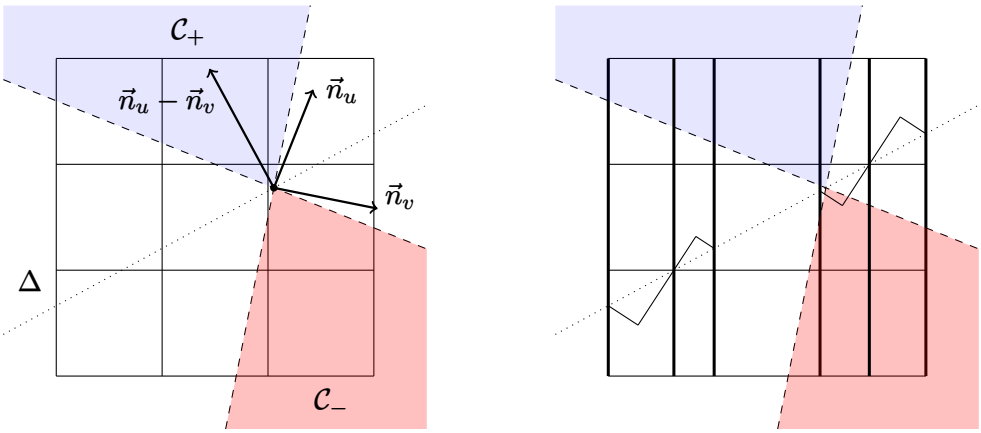


Figure 3.7: Generic situation for Node d, and partition of S into five columns: here, in addition to the four vertical lines delimiting the cells of S, we added two vertical lines passing through the intersections of Δ with the two horizontal cell delimiters

Let U be such a column, and T a cell intersecting U. If $T \cap U \neq T$, Δ intersects either the upper or lower boundary of T, but not both since Δ is at an angle of at most $\frac{\pi}{4}$ with the horizontal axis. If it is the upper boundary, the symmetric of the part of $T \cap U$ above Δ with respect to Δ is in $T \cap U$. If it is the lower boundary, the symmetric of the part of $T \cap U$ below Δ with respect to Δ is in $T \cap U$. Using the fact that $\mathcal{C}_+$ and $\mathcal{C}_-$ are symmetric with respect to Δ, we obtain

$$\begin{aligned}\min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|) = \min(|T \cap U \cap \mathcal{C}_+|, |T \cap U \cap \mathcal{C}_-|) \\ + \min(|T \cap U^c \cap \mathcal{C}_+|, |T \cap U^c \cap \mathcal{C}_-|).\end{aligned}$$

Thanks to this observation, instead of (3.28) we only have to prove the inequality

$$|U \cap \mathcal{C}| \geqslant 6 \sum_{T \subset U} \min(|T \cap U \cap \mathcal{C}_+|, |T \cap U \cap \mathcal{C}_-|) \tag{3.29}$$

on each column U separately. We thus consider only one column U in the sequel, and assume up to a horizontal dilation (which preserves the condition $|\arg(\Delta)| \leqslant \frac{\pi}{4}$) that U has width h and is composed of three full cells.

According to the definition of the columns, there is at most one cell $T \subset U$ such that $T \cap \Delta \neq \emptyset$, and as Δ separates $\mathcal{C}_+$ and $\mathcal{C}_-$, it is only for this cell that we may have $\min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|) \neq 0$. If there is no such cell, (3.29) trivially holds. Otherwise, similar to Node c, we only need to prove

$$|U \cap \mathcal{C}| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|),$$

where $T \subset U$ is the cell containing $\Delta \cap U$. Denoting P_1, P_2, P_3 and P_4 the upper left, upper right, lower left and lower right corner points of T, we observe that the assumptions on Δ and U imply $P_1, P_2 \notin \mathring{\mathcal{C}}_-$ and $P_3, P_4 \notin \mathring{\mathcal{C}}_+$.

Node e: If $U \cap \Delta_u \cap \Delta_v = \emptyset$, that is, if U contains no intersection point between Δ_u and Δ_v, we match five cases depending on the position of T in U, and of its corners with respect to $\mathcal{C}$. They are illustrated in Figure 3.8.

Case 6: If T is the bottom cell and $P_1, P_2 \in \mathcal{C}_+$, then the two other cells are included in $\mathcal{C}_+$, so

$$|U \cap \mathcal{C}| \geqslant 2h^2 + |T \cap \mathcal{C}| \geqslant 3|T \cap \mathcal{C}| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

Case 7: If T is the bottom cell and $P_1 \in \mathcal{C}_+$ but $P_2 \notin \mathcal{C}_+$, $T \cap \mathcal{C}_+$ is a triangle of width and height at most h, so there is a rectangle $R \subset (U \setminus T) \cap \mathcal{C}_+$ of same width and twice as high, and thus

$$|U \cap \mathcal{C}| \geqslant |R| + |T \cap \mathcal{C}| = 4|T \cap \mathcal{C}_+| + |T \cap \mathcal{C}| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

The same argument holds when $P_2 \in \mathcal{C}_+$ but $P_1 \notin \mathcal{C}_+$, and we necessarily have P_1 or P_2 in $\mathcal{C}_+$ since $T \cap \mathcal{C}_+ \neq \emptyset$. If T is the top cell, applying a symmetry with respect to the

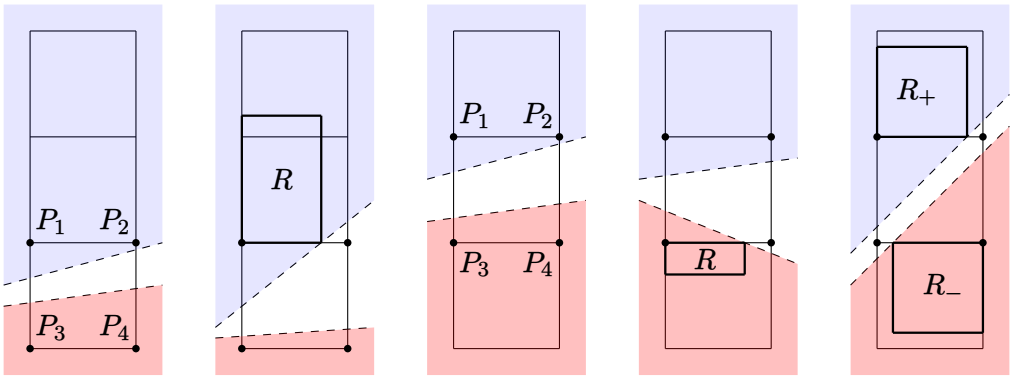


Figure 3.8: Cases 6, 7, 8, 9 and 10

horizontal axis and exchanging $\mathcal{C}_+$ with $\mathcal{C}_-$ brings us back to Cases 6 and 7.

Case 8: If T is the central cell, $P_1, P_2 \in \mathcal{C}_+$ and $P_3, P_4 \in \mathcal{C}_-$ the two other cells are included in $\mathcal{C}_+$ and $\mathcal{C}_-$, and we conclude as in Case 6.

Case 9: If T is the central cell, $P_1, P_2 \in \mathcal{C}_+$, $P_3 \in \mathcal{C}_-$ but $P_4 \notin \mathcal{C}_-$, the top cell is included in $\mathcal{C}_+$, and there is a rectangle $R \subset \mathcal{C}_-$ of same width and height as $T \cap \mathcal{C}_-$ in the bottom cell, so

$$|U \cap \mathcal{C}| \geqslant h^2 + |T \cap \mathcal{C}| + |R| \geqslant 2|T \cap \mathcal{C}| + 2|T \cap \mathcal{C}_-| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

The same situation occurs when only three points among $P_1, \dots, P_4$ are in $\mathcal{C}$.

Case 10: If T is the central cell, only one vertex among P_1, P_2 is in $\mathcal{C}_+$, and only one among P_3, P_4 is in $\mathcal{C}_-$, both $T \cap \mathcal{C}_+$ and $T \cap \mathcal{C}_-$ are triangles, and there exist rectangles R_+ and R_- of same widths and heights, so

$$|U \cap \mathcal{C}| \geqslant |R_+| + |T \cap \mathcal{C}| + |R_-| \geqslant 3|T \cap \mathcal{C}_+| + 3|T \cap \mathcal{C}_-| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

As $\mathcal{C}_+$ and $\mathcal{C}_-$ each contain at least one corner of T, we treated all cases for Node e.

Node f: Finally, we consider the situation where there is an intersection point $X \in \Delta_u \cap \Delta_v$ in U, and therefore in T. We again match five cases, illustrated in Figure 3.9, depending on the position of T in U, and of its corners with respect to $\mathcal{C}$.

Case 11: If T is the bottom cell, as Δ_u and Δ_v pass through the central cell of S, U is included in the central column of S, and no corner of T can be in $\mathring{\mathcal{C}}_+$, since otherwise Δ would have to pass through that corner, according to the definition of the columns. As a consequence, Δ_u and Δ_v necessarily pass through the central cell of U, so $T \cap \mathcal{C}_+$ is a triangle, and we proceed as in Case 7. The same happens if T is the top cell, so in the rest of the proof we only consider situations where T is the central cell.

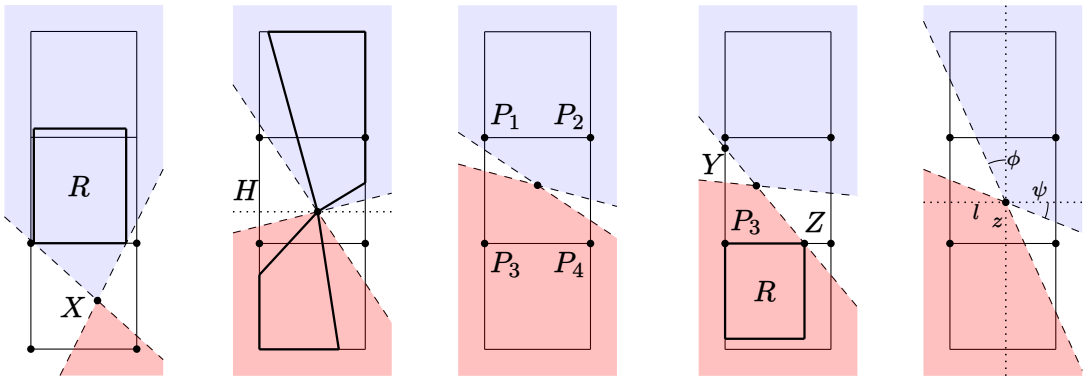


Figure 3.9: Cases 11, 12, 13, 14 and 15

Case 12: If the horizontal line H passing through X does not intersect $\mathcal{C}$ at any other point, $\mathcal{C}_+$ is entirely above H and $\mathcal{C}_-$ entirely below. Denoting $z = X_2 - \overline{x}_2 + \frac{h}{2} \in (0,h)$, the vertical dilation with respect to H by a factor $\frac{2h-z}{h-z}$ sends $T \cap \mathcal{C}_+$ in $U \cap \mathcal{C}_+$, and the vertical dilation with respect to H by a factor $\frac{h+z}{z}$ sends $T \cap \mathcal{C}_-$ in $U \cap \mathcal{C}_-$, so

$$|U \cap \mathcal{C}| \geqslant \frac{2h-z}{h-z}|T \cap \mathcal{C}_+| + \frac{h+z}{z}|T \cap \mathcal{C}_-| \geqslant 6\min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|)$$

because $\frac{2h-z}{h-z} + \frac{h+z}{z} = 2 + \frac{h^2}{z(h-z)} \geqslant 6$ for $z \in (0,h)$.

In the remaining cases, up to a symmetry with respect to the vertical axis, we can assume that $X + \mathbb{R}^2_+ \subset \mathcal{C}_+$ and $X + \mathbb{R}^2_- \subset \mathcal{C}_-$, and in particular $P_2 \in \mathcal{C}_+$ and $P_3 \in \mathcal{C}_-$.

Case 13: If $P_1 \in \mathcal{C}_+$ and $P_4 \in \mathcal{C}_-$, the situation is similar to Case 8.

Case 14: If $P_1 \in \mathcal{C}_+$ and $P_4 \notin \mathcal{C}_-$, the top cell is included in $\mathcal{C}_+$, and one of the lines Δ_u or Δ_v intersects the line segments $[P_1, P_3]$ and $[P_3, P_4]$ at points Y and Z. Then the triangle YP_3Z is included in T and contains $T \cap \mathcal{C}_-$, so there is a rectangle R of same width and height in $(U \setminus T) \cap \mathcal{C}_-$. In the end

$$|U \cap \mathcal{C}| \geqslant h^2 + |T \cap \mathcal{C}| + |R| \geqslant 2|T \cap \mathcal{C}| + 2|T \cap \mathcal{C}_-| \geqslant 6\min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|).$$

The same approach treats the symmetric case $P_1 \notin \mathcal{C}_+$ and $P_4 \in \mathcal{C}_-$,

Case 15: Finally, if $P_1 \notin \mathcal{C}_+$ and $P_4 \notin \mathcal{C}_-$, denote $l = X_1 - \overline{x}_1 + \frac{h}{2} \in (0,h)$, $z = X_2 - \overline{x}_2 + \frac{h}{2} \in (0,h)$, $\phi \in (0, \frac{\pi}{4})$ the angle between the vertical axis and the line among Δ_u and Δ_v that intersects $[P_1, P_2]$, and $\psi \in (0, \frac{\pi}{4})$ the angle between the line among Δ_u and Δ_v that intersects $[P_1, P_3]$ and the horizontal axis. As $|\arg(\Delta)| \leqslant \frac{\pi}{4}$, $\phi \geqslant \psi$ so $\tan(\psi) \leqslant \tan(\phi) =: t \leqslant 1$.

We can now compute

$$|T \cap \mathcal{C}_+| = (h-l)(h-z) + \frac{1}{2}(h-l)^2 \tan\psi + \frac{1}{2}(h-z)^2 \tan\phi,$$
$$|T \cap \mathcal{C}_-| = lz + \frac{1}{2}l^2 \tan\psi + \frac{1}{2}z^2 \tan\phi,$$

and

$$|(U \setminus T) \cap \mathcal{C}| \geqslant (h-l)h + (h-z)th + lh + zth = (1+t)h^2.$$

If $l + z \leqslant h$, we get

$$|(U \setminus T) \cap \mathcal{C}| \geqslant (1+t)(l+z)^2 - (1-t)(l-z)^2 = 4lz + 2t(l^2+z^2) \geqslant 4|T \cap \mathcal{C}_-|.$$

Similarly, $l + z \geqslant h$ implies $|(U \setminus T) \cap \mathcal{C}| \geqslant 4|T \cap \mathcal{C}_+|$. In any case, we found

$$|U \cap \mathcal{C}| = |T \cap \mathcal{C}| + |(U \setminus T) \cap \mathcal{C}| \geqslant 6 \min(|T \cap \mathcal{C}_+|, |T \cap \mathcal{C}_-|),$$

which concludes the proof. □

As a last remark, note that the constants $\alpha = h^{-2}$ and $\mu = \frac{3}{2}h^2$ in Proposition 3.5.1 are sharp, since equality is attained by functions of constant sign on each cell for α, and by $w = u - v$ with $\arg(\vec{n}_u) \in \frac{\pi}{4}\mathbb{Z}$, $c_u = 0$ and $v = 1 - u$ for μ.

Chapter 4

High order recovery of geometric interfaces from cell-average data

4.1 Introduction

4.1.1 Reconstruction from cell-averages

We consider the problem of reconstructing a function $u : D \to \mathbb{R}$ defined on a multivariate domain $D \subset \mathbb{R}^d$ from cell averages

$$a_T(u) := \frac{1}{|T|} \int_T u(x)dx, \quad T \in \mathcal{T}, \tag{4.1}$$

over a partition $\mathcal{T}$ of D. This task occurs in various contexts, the most notable ones being:

1. **Image processing:** here u is the light intensity of an image and $\mathcal{T}$ represents a grid of pixels in dimension $d = 2$ or voxels in dimension $d = 3$. Various processing tasks are facilitated by the reconstruction of the image at the continuous level, for example when applying operations that are not naturally compatible with the pixel grid such as rotations, or when changing the format of the pixel grid such as in super-resolution.

2. **Hyperbolic transport PDE's:** here u is a solution to such an equation and $\mathcal{T}$ is a computational grid, typically in dimension $d = 1, 2$ or 3. Finite volume schemes evolve the cell average data by computing at each time step the numerical fluxes at the interfaces between each cell. Several such schemes are based on an intermediate step that reconstructs simple approximations to u on each cell and compute the numerical fluxes by applying the transport operator to these approximations.

3. **Inverse Problems:** numerous inversion tasks can be formulated as the recovery of a function from observational data, and this data could typically come in the form of local averages of the type (4.1).

In this paper, we consider functions defined on the unit cube

$$D := [0,1]^d,$$

and partitions $\mathcal{T}_h$ of D based on uniform cartesian meshes, that is, consisting of cells of the form

$$T = h(k + D), \quad k := (k_1, \dots, k_d) \in \{0, \dots, l-1\}^d,$$

where $h := \frac{1}{l} > 0$ is the side-length of each cell in $\mathcal{T}_h$, for some $l > 1$. The cardinality of the partition is therefore

$$N := \#(\mathcal{T}_h) = l^d = h^{-d}.$$

We are thus interested in reconstruction operators R that return an approximation $\tilde{u} = R(a)$ to u from the N-dimensional vector $a = a(u) = (a_T(u))_{T \in \mathcal{T}_h} \in \mathbb{R}^N$. The most trivial one is the piecewise constant function

$$\tilde{u} := \sum_{T \in \mathcal{T}_h} a_T(u) \chi_T, \tag{4.2}$$

which is for example used in the Godunov finite volume scheme. Elementary arguments show that this reconstruction is first order accurate: if $u \in W^{1,p}(D)$, one has

$$\|u - \tilde{u}\|_{L^p(D)} \leqslant Ch \|\nabla u\|_{L^p(D)} \sim N^{-\frac{1}{d}},$$

where C is a fixed constant, and the exponent in this estimate cannot be improved for smoother functions.

A simple way to raise the order of accuracy is by reconstructing on each cell polynomials of higher degree using neighbouring cell averages. For example, in the univariate case $d = 1$ and for some fixed $m \geqslant 1$, we associate to each interval $T_k := [kh, (k+1)h]$ the centered stencil consisting of the cells T_l for $l = k - m, \dots, k + m$. Then, there exists a unique polynomial $p_k \in \mathbb{P}_{2m}$ such that

$$a_{T_l}(p_k) = a_{T_l}(u), \quad l = k - m, \dots, k + m.$$

We then define a piecewise polynomial reconstruction by

$$\tilde{u} := \sum_{k=0,\dots,N-1} p_k \chi_{T_k}.$$

This strategy can be generalized to higher dimension $d > 1$ in a straightforward manner: for each cell T we consider the stencil $S = S_T$ of $(2m+1)^d$ cells centered around T and define the piecewise polynomial reconstruction

$$\tilde{u} := \sum_{T \in \mathcal{T}_h} p_T \chi_T,$$

where p_T is the unique polynomial of degree $2m$ in each variable such that

$$a_{\tilde{T}}(p_T) = a_{\tilde{T}}(u), \quad \tilde{T} \in S_T.$$

For example in the bivariate case $d = 2$, we may use 3×3 stencils to reconstruct bi-quadratic polynomials on each cell. Standard approximation theory arguments show that these local reconstruction operators now satisfy accuracy estimates of the form

$$\|u - \tilde{u}\|_{L^p(D)} \leqslant Ch^r |u|_{W^{r,p}(D)} \sim N^{-\frac{r}{d}},$$

for $r \leqslant 2m + 1$.

Remark 4.1.1. *Polynomials of odd degree can also be constructed by using non-centered stencils. Also note that non-centered stencils need to be used when approaching the boundary of D, but this does not affect the above estimate.*

These classical methods are therefore efficient to reconstruct smooth functions with a rate of accuracy that optimally reflects their amount of smoothness. Unfortunately they are doomed to perform poorly in the case of functions u that are piecewise smooth with jump discontinuities accross hypersurfaces. For example, a piecewise constant reconstruction will have $\mathcal{O}(1)$ error on each cell that is crossed by the interface. Since the amount of such cells is of order $N^{d-1} = h^{1-d}$, we cannot expect a reconstruction error better than

$$\|u - \tilde{u}\|_{L^p} \gtrsim (h^d h^{1-d})^{\frac{1}{p}} = h^{\frac{1}{p}} = N^{-\frac{1}{dp}}. \tag{4.3}$$

In particular, this reconstruction has first order accuracy $\mathcal{O}(h)$ in the L^1 norm.

The use of higher order polynomial cannot improve this rate. In fact, a fundamental obstruction is the fact that all the above methods produce approximations $\tilde{u}$ that depend linearly on $a(u)$, and therefore belong to a linear space of dimension N. The bottleneck of such methods for a given class of functions $\mathcal{K}$ is therefore given by the so-called Kolmogorov N-width defined by

$$d_N(\mathcal{K})_{L^p} := \inf_{\dim(V_N)=N} \sup_{u \in \mathcal{K}} \min_{v \in V_N} \|u - v\|_{L^p}.$$

Then, it can be shown that for very simple classes $\mathcal{K}$ of discontinuous functions such as those of the form $u = \chi_H$, where H is any half-space passing through D, the N-width in L^p precisely behaves like $N^{-\frac{1}{dp}}$, see [**NonLinearReduced**]. In summary, any linear method cannot do much better than the low order piecewise constant method, even for interfaces that are infinitely smooth.

Improving the accuracy in the reconstruction of piecewise smooth functions from cell averages therefore motivates the development and study of nonlinear reconstruction strategies, which is at the heart of this work. We first recall the main existing approaches.

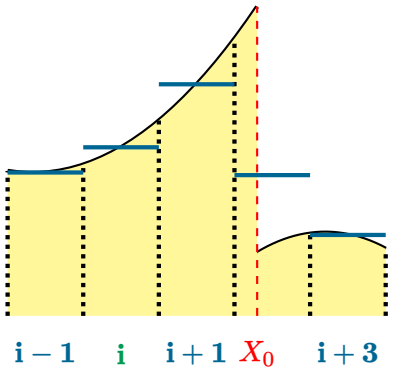


Figure 4.1: ENO-SR in one dimension: the jump point X_0 is identified by matching the average on the singular cell with the piecewise polynomial reconstruction.

4.1.2 Reconstruction of discontinuous interfaces

One first approach aiming to tackle jump discontinuities while maintaining high order approximation in the smooth regions was proposed for the univariate case $d = 1$ by Ami Harten in terms of ENO (Essentially Non Oscillatory) and ENO-SR (Subcell Resolution). The ENO strategy [89] is based on selecting for each cell T a stencil S_T that should not include the cell T^* which contains the point of jump discontinuity. This is achieved by choosing among the stencils that contains T the one where the cell average values have the least numerical variation. For $T \neq T^*$ such adaptively selected stencils will tend to avoid T^*.

As a consequence high order reconstruction can be preserved in all cells where u is smooth, and in addition this yields for free a singularity detection mechanism which identifies the singular cell T^* that is avoided from both side by the stencil selection. The ENO-SR strategy [88] then consists in reconstructing in this singular cell by extending the polynomials fitted on both sides until a point for which the resulting average match the observed average. The position of this point can therefore be identified by solving a simple algebraic equation. This strategy is very effective in the univariate case as illustrated in Figure 4.1.

While the ENO stencil selection can be generalized to higher dimension, the ENO-SR strategy does not have a straightforward multivariate version. This is due to the fact that, instead of a single point, the jump discontinuity to be identified is now a hypersurface (curve in 2d, surface in 3d, ...) that cannot be described by finitely many parameters. The approximate recovery of geometric interfaces from cell-average has been the object of continuous investigation, with the particular focus on functions of the form

$$u = \chi_\Omega, \tag{4.4}$$

where $\Omega \subset D$ is a set with boundary $\Gamma = \partial\Omega$ having a certain smoothness. For such characteristic functions, that could for example represent a two phase flow without any mixing or the evolution of a front, the whole difficulty is concentrated in the recovery of Γ since the smooth parts are the trivial constant functions 0 or 1.

In this paper, we denote by

$$S_h := \{T \in \mathcal{T}_h \ : \ |T \cap \Omega| \neq 0 \text{ and } |T \cap \Omega^c| \neq 0\},$$

the set of cells that are crossed by the interface Γ in a non trivial way. Such cells are termed as *singular*, the other as *regular*. Let us note that, for functions u of the form (4.4), singular cells are characterized by the property

$$0 < a_T(u) < 1,$$

and can thus be identified from the cell-average data.

Practical computational strategies, often termed as *volume of fluid* methods, consist in using the cell-average data to reconstruct a local approximation of the interface that can be described by finitely many parameters, such as lines in 2d or planes in 3d. This idea was introduced in [118], and was significantly improved in [132] with the LVIRA algorithm that consists in reconstructing in each singular cell a linear interface whose parameters are found by least-square minimization of the difference between exact and reconstructed cell averages on centered 3×3 stencil, in the 2d case.

Note that this continuous least-square minimization is performed over a nonlinear set. This induces a substantial computational burden that could be avoided through a variant, the ELVIRA algorithm, in which the line selection is made between 6 possible configurations explicitly computed from the cell averages. One main result is the fact that this reconstruction returns precisely the true interface if this one is indeed a straight line. We refer to [129] for a comparative survey on these reconstruction algorithms and to [93, 141, 136, 143, 84, 157, 68, 135] for improvements and applications in the domain of 2d and 3d fluid mechanics.

Figure 4.2: Local approximation of a smooth interface by a line interface.

Intuitively the advantage of locally fitting a line or plane interface is that it has the ability to better approximate the interface Γ if it is smooth, so that it is expected to improve the low order of accuracy (4.3) of linear methods. More precisely, if Γ has $\mathcal{C}^2$ regularity, on each cell of side-length h, it can be approximated with Hausdorff distance $\mathcal{O}(h^2)$ by a line in 2d, a plane in 3d, etc. Therefore, if the locally reconstructed linear interface is optimally fitted, the $\mathcal{O}(1)$ error is observed on a strip of volume $\mathcal{O}(h^{2+d-1})$, see Figure 4.2. Since the

amount of singular cell is of order $\#(\mathcal{S}_h) \sim h^{1-d}$, we may hope for a reconstruction error with improved order,

$$\|u - \tilde{u}\|_{L^p} \sim (h^{d+1} h^{1-d})^{\frac{1}{p}} = h^{\frac{2}{p}} = N^{-\frac{2}{dp}}, \tag{4.5}$$

that is, we double the rate compared to (4.3). In particular, we obtain second order accuracy $\mathcal{O}(h^2)$ in the L^1 norm.

However, to our knowledge such estimates have never been rigourously proved for the aforementionned method. In addition the approximation power of linear interface is also limited and we cannot hope to improve the above rate for interfaces that are smoother than $\mathcal{C}^2$. In this context, the objective of this paper is twofold:

1. introduce a theoretical framework for the rigourous convergence analysis of local interface reconstructions from cell averages,

2. within this framework develop reconstruction methods going beyond linear interfaces and provably achieving higher order of accuracy.

4.1.3 Outline

In this paper, we will essentially work in the bivariate case $d = 2$ which makes the exposition simpler while most of our discussion can be carried over to higher dimension.

The recovery methods that we study are local: on each cell T identified as singular, the unknown function $u = \chi_\Omega$ is approximated by a simpler $\tilde{u} = \chi_{\Omega_T}$ picked from a family V_n that can be described by n parameters and that enjoy certain approximation properties for interfaces having prescribed smoothness. This approximation is computed from the cell averages of a rectangular stencil S_T of $m \geqslant n$ cells centered around T.

We begin Section 4.2 by giving examples of such families and discussing their approximation properties for prior classes $\mathcal{K}_s$ of χ_Ω associated to sets Ω with $\mathcal{C}^s$ boundaries for $s > 1$. Our ultimate goal is to develop recovery schemes such that the error between u and $\tilde{u}$ is near optimal in the sense of being bounded (up to a multiplicative constant) by the error of best approximation of u by elements from V_n. This, in particular means that the recovery should be exact if the true function u belongs to V_n.

We introduce in Section 4.3 a first class of recovery strategies which we call Optimization Based Edge Reconstruction Algorithms (OBERA). Similar to LVIRA it is based on least-square fitting of simpler interfaces such as lines, but also circles or polynomials that allow to raise the order of accuracy. We show that near optimality of this recovery is ensured by an inverse stability inequality which can in particular be established for line edges and 3×3 stencils. Unfortunately, this property seems more difficult to prove when raising the order of accuracy, which also leads to more difficult nonlinear optimization procedures.

As a more manageable alternative, we consider recovery methods that are based on the identification of a certain preferred orientation - vertical or horizontal - for describing the interface by a function in the vicinity of each such cell. This leads us to the second class of recovery methods, termed as Algorithms for Edge Reconstruction using Oriented Stencils (AEROS) which is discussed in Section 4.4. It avoids continuous optimization by finding

an edge described by a polynomial function $y = p_T(x)$ or $x = p_T(y)$ after having used the previously discussed orientation mechanism. The polynomial p_T is identified by simple linear equations. We show that this process satisfies an exactness and stability property that leads to the near optimal recovery bound. In addition, the rate of convergence can be raised to an arbitrarily high order by raising the degree of p_T, however at the price of using larger stencils. The analysis of the orientation selection mechanism, based on the Sobel filter, is postponed to the Appendix, where it is shown that it correctly classifies the cells when h is sufficiently small.

All these methods are numerically tested and compared in Section 4.5. The differences in terms of convergence rates are confirmed, and we also compare the computational costs, which are by far less for AEROS than OBERA. We also discuss and test the specific recovery of corner singularities in the interface. Finally, in the case of linear transport, we illustrate the propagation of error when using finite volume schemes based on these local recovery strategies.

An open-source python framework[1] is made available to show the methods presented here but specially to allow an easy way of creating, testing, and comparing new subcell resolution and interface reconstruction methods without the need to re-implement everything from scratch.

4.2 Numerical analysis of local recovery methods

4.2.1 Local approximation by nonlinear families

The methods that we study in this paper for the recovery of the unknown $u = \chi_\Omega$ are based on local approximations of u on each cell $T \in \mathcal{T}_h$ that is identified as singular by simpler characteristic functions picked from an n parameter family V_n.

Let us give three examples that will be used further. We stress that in all such examples V_n is not an n-dimensional linear space, but instead should be thought as an n-dimensional nonlinear manifold.

Example 1: linear interfaces. These are functions of the form $v = \chi_H$ where H is a half-plane with a line interface $L = \partial H$. Such functions are described by $n = 2$ parameters. One convenient description is by the pair (r, θ), where $r \geqslant 0$ is the offset distance between the center z_T of the cell T of interest and the linear interface and $\theta \in [0, 2\pi[$ is the angle between this line and the horizontal axis. In other words, in this case we use

$$V_2 := \{v_{r,\theta} := \chi_{\{\langle z - z_T, e_\theta\rangle \leqslant r\}}\ ,\ \theta \in [0, 2\pi[;\ r > 0\},$$

where $e_\theta = (-\sin(\theta), \cos(\theta))$. Of course, the d-dimensional generalization by half-spaces is a d-parameter family where the unit normal vector e_θ lives on the $d-1$-dimensional unit sphere.

[1] https://github.com/agussomacal/SubCellResolution

Example 2: circular interfaces. These are functions of the form $v = \chi_D$ or $v = \chi_{D^c}$ where D is a disc with circular interface, and D^c its complement. It is easily seen that the corresponding space V_3 is now a 3 parameter family, and its d-dimensional generalization by characteristic function of balls and their complements is a $d+1$ parameter family. The idea of using circles instead of lines is to increase the approximation capability. However, as we will see, the next family turns out to be more effective both from the point of view of theoretical analysis and computational simplicity.

Example 3: oriented graphs. These are functions of the form $v = \chi_P$ where P is the subgraph or the epigraph of a function $p \in W_n$, either applied to the coordinate x or y, where W_n is a linear space of univariate functions. In other words, P is given by one among the four equations

$$y \leqslant p(x), \quad y \geqslant p(x), \quad x \leqslant p(y), \quad x \geqslant p(y). \tag{4.6}$$

Of course, the corresponding space V_n is an n-parameter family that is not a linear space, while W_n is. Note that the linear interfaces of Example 1 are a particular case where W_2 is the set of affine functions. Raising the order of accuracy will be achieved by taking for W_n the space of polynomials of degree $n-1$. The d dimensional generalisation is obtained by taking for W_n a linear space of functions of $d-1$ variables and considering P to be defined by one of the 2d equations

$$x_i \leqslant p(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d), \quad x_i \geqslant p(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d), \quad i = 1, \dots d,$$

for some $p \in W_n$. In particular W_n could be a space of multivariate polynomials.

Example 4: piecewise linear interface. These are functions of the form $u = \chi_{H_1 \cap H_2}$ where H_1 and H_2 are two half planes. Therefore the interface consists of two half lines that touch at a corner point x_0. The corresponding space V_4 is a 4 parameter family, for example by considering the coordinates $x_0 = (x_1, x_2)$ and the angles θ_1 and θ_2 of the normal vectors to the two lines. The goal of this family is to better approximate piecewise smooth interfaces that have corner singularities. Note that Example 1 may be viewed as a particular case where the two half-planes coincide and there is no corner point.

Given such a family V_n and a set $S \subset D$, we denote by

$$e_n(u)_S = \min_{v \in V_n} \|u - v\|_{L^1(S)},$$

the error of best approximation on S measured in the L^1-norm.

Remark 4.2.1. *Throughout this paper, we shall systematically measure error in L^1 norm which is the most natural since in the case of $u = \chi_\Omega$ and $\tilde{u} = \chi_{\tilde{\Omega}}$, this error is simply the area of the symmetric difference between domains, that is,*

$$\|u - \tilde{u}\|_{L^1(D)} = |\Omega \,\Delta\, \tilde{\Omega}| = |\Omega \cup \tilde{\Omega} - \Omega \cap \tilde{\Omega}|.$$

Note however that estimates in L^p norms can be derived from L^1 estimates in a straightforward manner since

$$\|u-\tilde{u}\|_{L^p(D)} = |\Omega \,\Delta\, \tilde{\Omega}|^{1/p} = \|u-\tilde{u}\|^{1/p}_{L^1(D)}.$$

In our analysis of the reconstruction error on a singular cell T, we will need to estimate the local approximation error on a stencil $S = S_T$ that consists of finitely many cells surrounding T. We shall systematically consider rectangular stencils of symmetric shape centered around T, so in the 2d case they are of size

$$m = (2k+1) \times (2l+1),$$

for some fixed $k, l \geqslant 1$.

The order of magnitude $e_n(u)_S$ both depends on the type of family V_n that one uses and on the smoothness property of the boundary $\Gamma = \partial\Omega$. We describe these properties by introducing prior classes of characteristic functions χ_Ω of sets $\Omega \subset D$ with boundary of a prescribed Hölder smoothness. There exists several equivalent definitions of a $\mathcal{C}^s$ domain. We follow the approach from [116], that expresses the fact that the boundary can locally be described by graphs of $\mathcal{C}^s$ functions (see also Chapter 4 of [2]).

Definition 4.2.2. *Let $s > 0$. A domain $\Omega \subset \mathbb{R}^2$ is of class $\mathcal{C}^s$ if and only if there exists an $R > 0$, $P > 1$ and $M > 0$, such that for any point $z_0 \in \partial\Omega$, the following holds: there exists an orthonormal system (e_1, e_2) and a function $\psi \in \mathcal{C}^s([-R, R])$ with $\|\psi\|_{\mathcal{C}^s} \leqslant M$, taking its value in $[-PR, PR]$ and such that*

$$z \in \Omega \iff z_2 \leqslant \psi(z_1),$$

for any $z = z_0 + z_1 e_1 + z_2 e_2$ with $|z_1| \leqslant R$ and $|z_2| \leqslant PR$.

Here, we have used the usual definition

$$\|\psi\|_{\mathcal{C}^s} = \sup_{0 \leqslant k \leqslant \lfloor s \rfloor} \|\psi^{(k)}\|_{L^\infty([-R,R])} + \sup_{s,t \in [-R,R]} |s-t|^{\lfloor s \rfloor - s} \left| \psi^{(\lfloor s \rfloor)}(s) - \psi^{(\lfloor s \rfloor)}(t) \right|,$$

for the Hölder norm. In the case of integer smoothness, we use the convention that $\mathcal{C}^s$ denotes functions with Lipschitz derivatives up to order $s-1$, so that in particular the case $s = 1$ corresponds to domains with Lipschitz boundaries. This definition naturally extends to domains of $\mathbb{R}^d$ with $d > 2$ with ψ now being a $\mathcal{C}^s$ function of $d-1$ variables.

We can immediately derive a first local approximation error estimate for the the space V_2 of linear interfaces from the above Example 1: let $u = \chi_\Omega$ with Ω a domain of class $\mathcal{C}^s$. Then, if S is a $2k+1 \times 2l+1$ stencil centered around a cell T that is crossed by the interface, we apply the above Definition 4.2.2 taking $z_0 = z_T$ the center of T. We assume that the sidelength h is small enough so that the stencil S is contained in the rectangle $\{z = z_0 + z_1 e_1 + z_2 e_2 \, : , |z_1| \leqslant R, \, |z_2| \leqslant PR\}$, in which $\partial\Omega$ is described by the graph of the $\mathcal{C}^s$ function ψ. For $z \in S$, we have in addition that $|z_1| \leqslant C_0 h \leqslant R$ where C_0 depends only on l and k. Using a Taylor expansion and the smoothness of ψ we find that there exists an

affine function a such that

$$\|\psi - a\|_{L^\infty([-C_0 h, C_0 h])} \leqslant C_1 h^r, \quad r := \min\{s, 2\}, \tag{4.7}$$

where C_1 depends on C_0 and the bound M on the $\mathcal{C}^s$ norm of ψ. For example, we can take for a the Taylor polynomials of order 1 at $z_1 = 0$ when $s > 1$, which corresponds to match the tangent of the interface at the point $z_0 + \psi(0)e_2$, or of order 0 when $s \leqslant 1$. Then, taking $v = \chi_H \in V_2$, where

$$H := \{z_2 \leqslant a(z_1)\},$$

is the corresponding half-space, it follows that

$$\|u - v\|_{L^1(S)} = |S \cap (\Omega \,\Delta\, H)| \leqslant C h^{r+1},$$

where C depends on (M, l, k). In summary, for the local approximation error of $\mathcal{C}^s$ domains by a linear interface we have

$$e_n(u)_S \leqslant C h^{r+1}, \quad r := \min\{s, 2\}. \tag{4.8}$$

The same reasoning in d dimensions delivers a local approximation estimate of order h^{r+d-1}.

One way to raise this order of accuracy for smoother domains is to use approximation by circular interfaces from Example 2 since this allows us to locally match the curvature in addition to the tangent. In turn we reach a similar estimate of order h^{r+1} however with $r := \min\{s, 3\}$. One more systematic way of raising the order arbitrarily high is to use approximation by oriented subgraphs from Example 3. This approach is central to the AEROS strategies discussed in Section 4.5 and we thus discuss it below in more detail.

We begin with the observation that when $s > 1$, the unit tangent vector varies continuously on $\partial\Omega$ if Ω is of class $\mathcal{C}^s$. It follows that locally around any point z_0, this vector remains away either from the horizontal vector $(1, 0)$ or from the vertical vector $(1, 0)$. This allows us to locally describe the boundary by graphs of functions of the standard cartesian coordinates, as expressed by the following alternate definition of $\mathcal{C}^s$ domains.

Definition 4.2.3. *Let $s > 1$. A domain $\Omega \subset \mathbb{R}^2$ is of class $\mathcal{C}^s$ if and only if there exists an $R > 0$, $P > 1$ and $M > 0$, such that for any point $z_0 = (x_0, y_0) \in \partial\Omega$, the following holds: there exists a function $\psi \in \mathcal{C}^s([-R, R])$ with $\|\psi\|_{\mathcal{C}^s} \leqslant M$, taking its value in $[-PR, PR]$ and such that the membership in Ω of a point $z = (x, y)$ is equivalent to one of the two equations*

$$y \leqslant \psi(x), \quad y \geqslant \psi(x), \tag{4.9}$$

when $|x - x_0| \leqslant R$ and $|y - y_0| \leqslant PR$, or one of the two equations

$$x \leqslant \psi(y), \quad x \geqslant \psi(y), \tag{4.10}$$

when $|y - y_0| \leqslant R$ and $|x - x_0| \leqslant PR$.

The generalization of this alternate definition to higher dimension is straighforward by

considering equations one of the form

$$x_i \leqslant \psi(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d) \quad \text{or} \quad x_i \geqslant \psi(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d), \quad i = 1, \dots, d,$$

with ψ a $\mathcal{C}^s$ function of $d-1$ variables defined on $[-R, R]^{d-1}$.

Remark 4.2.4. *We stress that this definition is only valid for $s > 1$ and not for less smooth domains such as Lipschitz domains. For example if Ω is a rectangle with side oriented along principal axes, then no such local parametrization can be derived if z_0 is a corner point.*

Consider now the family V_n of oriented subgraphs from Example 3, associated with the linear space $W_n = \mathbb{P}_{n-1}$ of univariate polynomials of degree $n-1$. Let $u = \chi_\Omega$ with Ω a domain of class $\mathcal{C}^s$ for some $s > 1$. Then, if S is a $(2k+1) \times (2l+1)$ stencil centered around a cell T that is crossed by the interface, we apply the above Definition 4.2.3 taking $z_0 = z_T$ the center of T. Without loss of generality, assume for example that the description of Ω near z_0 is by the equation

$$y \leqslant \psi(x),$$

for $z = (x, y)$ in the rectangle $\{|x - x_0| \leqslant R,\ |y - y_0| \leqslant PR\}$. We assume that the sidelength h is small enough so that the stencil S is contained in this rectangle. For $z = (x, y) \in S$ we thus have

$$|x - x_0| \leqslant C_0 h \leqslant R, \quad C_0 = k + \frac{1}{2}.$$

Using Taylor formula and the smoothness of ψ we find that there exists a polynomial $p \in \mathbb{P}_{n-1}$ such that

$$\|\psi - p\|_{L^\infty([-C_0 h, C_0 h])} \leqslant C_1 h^r, \quad r := \min\{s, n\}, \tag{4.11}$$

where C_1 depends on C_0 and the bound M on the $\mathcal{C}^s$ norm of ψ. For example, we can take for p the Taylor polynomials of order $\tilde{n} = \min\{\lceil s \rceil - 1, n-1\}$ at $x = x_0$. Therefore, taking v, where

$$v := \chi_{\{y \leqslant p(x)\}} \in V_n,$$

the corresponding subgraph, it follows that

$$\|u - v\|_{L^1(S)} = 2 C_0 h C_1 h^r = C h^{r+1}.$$

where C depends on (M, k). We treat the other cases $y \geqslant \psi(x)$, $x \leqslant \psi(y)$ and $x \geqslant \psi(y)$ in a similar manner. In summary, for the local approximation error of $\mathcal{C}^s$ domains by polynomial oriented subgraphs, we have

$$e_n(u)_S \leqslant C h^{r+1}, \quad r := \min\{s, n\}. \tag{4.12}$$

The same reasoning in d dimensions delivers a local approximation estimate of order h^{r+d-1}.

4.2.2 Near optimal recovery from cell averages

The recovery methods that we develop in Section 4.3 and Section 4.4 are based on recovering on each singular cell T an element $\tilde{u}_T \in V_n$ where V_n is a given nonlinear family, based on the data of the cell-averages

$$a_S(u) = (a_{\tilde{T}}(u))_{\tilde{T}\in S},$$

where $S = S_T$ is a rectangular stencil centered around T. It can therefore be summarized by a local nonlinear recovery operator

$$R_T : \mathbb{R}^m \to V_n$$

where $m = (2k+1)\times(2l+1)$ is the size of the stencil, such that

$$\tilde{u}_T = R_T(a_S(u)).$$

We are interested in deriving a favorable comparison between the local recovery error $\|u - \tilde{u}_T\|_{L^1(T)}$ and the error of best approximation by V_n whose magnitude can be estimated depending on the amount of smoothness of the boundary, as previously discussed.

Definition 4.2.5. *The local recovery procedure is said to be near-optimal over a class of function $\mathcal{K}$ if there exists a fixed constant C so that one has*

$$\|u - R_T(a_S(u))\|_{L^1(T)} \leqslant C e_n(u)_S, \tag{4.13}$$

for all u in this class. In particular C should be independent of the considered singular cell T and mesh size h.

Remark 4.2.6. *In the above definition, the recovery error on T is bounded by the approximation error on the larger stencil S. This is due to the fact that the recovery operator R_T acts on the cell averages $a_{\tilde{T}}(u)$ for all cells $\tilde{T} \subset S$.*

On regular cells $T \in \mathcal{T}_h \setminus \mathcal{S}_h$, that is, such that $a_T(u) = 0$ or $a_T(u) = 1$, we simply define the reconstruction by the constant value

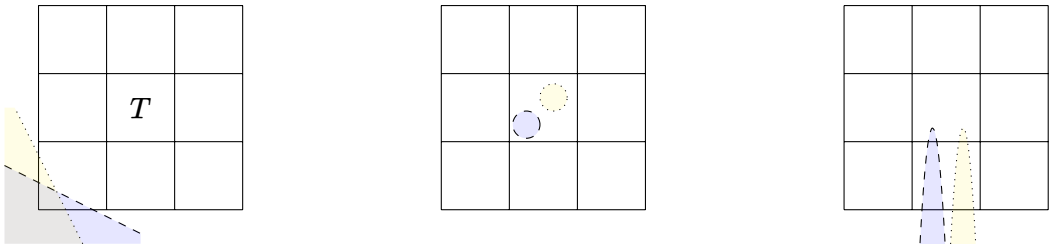
$$\tilde{u}_T = a_T(u),$$

which is then the exact value of u. The global reconstruction of u from the cell-averages $(a_T(u))_{T\in\mathcal{T}_h}$ is given by the function

$$\tilde{u} = \sum_{T\in\mathcal{T}_h} \tilde{u}_T \chi_T.$$

The global L^1 error can be estimated by aggregating all local error estimates, which thus gives

$$\|u - \tilde{u}\|_{L^1(D)} = \sum_{T\in\mathcal{S}_h} \|u - \tilde{u}_T\|_{L^1(T)} \leqslant C \sum_{T\in\mathcal{S}_h} e_n(u)_{S_T}. \tag{4.14}$$



(a) Line interfaces (b) Circular interfaces (c) Quadratic subgraphs

Figure 4.3: Cases of non-injectivity: two elements of V_n having same averages on a 3×3 stencil.

where C is the stability constant in (4.13) and where S_T denotes the stencil centered at T which is used in the reconstruction of $\tilde{u}_T$.

If Ω is a $\mathcal{C}^s$ domain with $s \geqslant 1$, that is, at least a Lipschitz domain, one has the cardinality estimate

$$\#(\mathcal{S}_h) \leqslant Ch^{-1} \tag{4.15}$$

where C depends on the length of Γ. We may thus derive a global error estimate by combining (4.14) and (4.15) and the local approximation estimates (4.8) and (4.12): is Ω is a $\mathcal{C}^s$ domain with $s \geqslant 1$, we obtain

$$\|u - \tilde{u}\|_{L^1(D)} \leqslant Ch^r, \tag{4.16}$$

with $r := \min\{s, 2\}$ when using local recovery by linear interfaces and $r := \min\{s, n\}$ when using local recovery by polynomial subgraphs of degree $n-1$. This estimate generalizes to the higher dimensional case, combining local approximation estimates with the cardinality estimate $\#(\mathcal{S}_h) \leqslant Ch^{1-d}$.

Our central objective is now to propose local recovery methods that provably satisfy the near optimal recovery bound (4.13). For this, we start by remarking that this bound implies the property

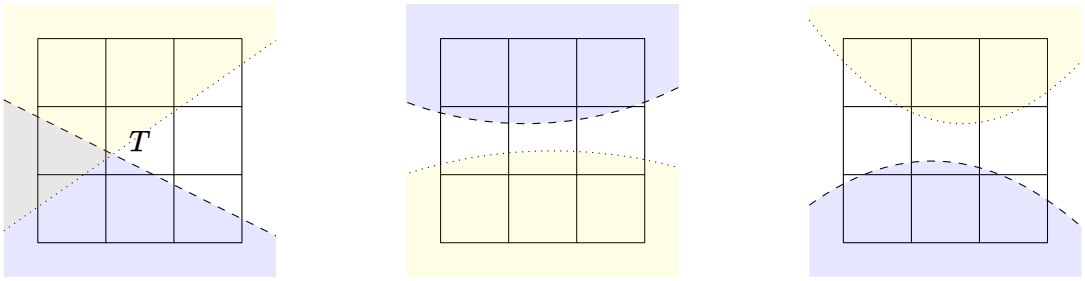
$$R_T(a_S(v)) = v, \quad \forall v \in V_n, \tag{4.17}$$

that is, the recovery is exact for elements from V_n. This property itself implies that any element from V_n should be exactly characterized by its cell-average on the stencil S. In other words, the averaging operator

$$v \mapsto a_S(v) = (a_{\tilde{T}}(v))_{\tilde{T} \in S},$$

should be injective from V_n to $\mathbb{R}^m$. For this to hold, the classes V_n from Examples 1, 2, 3, 4 need to be restricted.

In the case of linear interfaces, if H and $\tilde{H}$ are two half-spaces that contain the stencil S, the functions $v = \chi_H$ and $\tilde{v} = \chi_{\tilde{H}}$ obviously have the same cell-average vector a_S with component identically equal to 1. Asking that the linear interface passes through the stencil is thus necessary but not sufficient as illustrated on Figure 4.3a: two lines passing



(a) Line interfaces (b) Circular interfaces (c) Quadratic subgraphs

Figure 4.4: Illustration of the restrictions ensuring injectivity

only through a corner cell may result in identical cell averages. The correct restriction on V_n in this case is obtained by imposing that the linear interface passes through the central cell, which in the case of a 3×3 stencil suffices to ensure injectivity as recalled in Section 4.4 and illustrated on Figure 4.4a. An important observation is that this type of restriction does not affect the local error estimates (4.8) since we are precisely considering a stencil S centered at a cell T that contains the interface. Therefore, the tangent of the interface at the point $z_0 + \psi(0)e_2$ delivering this estimate satisfies this restriction.

In the case of circular interface, asking that the disk intercepts the central cell is not sufficient as illustrated on Figure 4.3b: two discs D and $\tilde{D}$ of equal size and contained in the central cell will result in identical cell averages for $v = \chi_D$ and $\tilde{v} = \chi_{\tilde{D}}$. In this case, an additional restriction should be that the radius of the disc is sufficiently large compared to the size of the stencil, for example by imposing that the disc center is not contained in S, as illustrated on Figure 4.4b.

In the case of subgraphs of polynomial functions, again two subgraphs passing through several cells of the stencil might have the same cell averages. This typically occurs when the polynomials are too peaky, as illustrated on Figure 4.3c. In this case, the additional restriction should be that the range of p remains inside the stencil. By this we mean, say for a subgraph of the type $y \leqslant p(x)$ and a rectangular stencil $S = [a,b]\times[c,d]$, one has that $p([a,b]) \subset [c,d]$, as illustrated in Figure 4.4c. Similarly to linear interfaces, the local error estimate (4.12) is not affected by such a restriction, as we discuss in Section 4.5.

A similar type of observation shows that there is no hope to uniquely characterize a piecewise linear interface from the cell averages on a given stencil if it is too peaky, that is, the opening angle of the cone embraced by the two lines cannot be arbitrarily close to 0 or 2π. In other words, corners cannot be arbitraritly acute or obtuse.

4.3 Reconstruction by optimization (OBERA)

4.3.1 Presentation of the method

Optimization-Based Edge Reconstruction Algorithms (OBERA) consists in recovering on each singular cell $T \in \mathcal{S}_h$ a recovery $\tilde{u}_T \in V_n$ by a best fit of the available cell-average data

on the stencil $S = S_T$. For this purpose, we solve a minimization problem of the form

$$\tilde{u}_T = R_T(a_S(u)) \in \arg\min_{v \in V_n} \|a_S(v) - a_S(u)\|$$

where $\|\cdot\|$ is a given norm on $\mathbb{R}^m$ where $m := \#(S)$ is the size of the stencil. In a practical implementation, one first simple choice is to use the Euclidean ℓ^2 norm, that is, minimize the loss function

$$\mathcal{L}(u, v) := \sum_{\tilde{T} \in S} |a_{\tilde{T}}(u) - a_{\tilde{T}}(v)|^2,$$

over all $v \in V_n$. The case of linear interface corresponds to the LVIRA method [132]. Note that $v \in V_n$ is defined through an appropriate parametrization as in Example 1, 2, 3 and 4,

$$\mu \in \mathcal{M} \subset \mathbb{R}^n \mapsto v_\mu \in V_n,$$

where $\mathcal{M}$ is the restricted range of the parameter μ that defines the family V_n. Therefore the optimization is done in practice by searching for

$$\mu^* \in \arg\min_{\mu \in \mathcal{M}} \mathcal{L}(u, v_\mu)$$

and taking $\tilde{u}_T = v_{\mu^*}$.

It is interesting to note that this recovery method is not consistent in the sense that it does not guarantee that

$$a_T(\tilde{u}_T) = a_T(u), \tag{4.18}$$

a property that is required, typically in finite volume methods since it reflects the conservation of mass. In order to restore this property, one possibility is to define the recovery by solving the constrained optimization problem

$$\tilde{u}_T \in \arg\min_{v \in V_n} \{\mathcal{L}(u, v) \,:\, a_T(v) = a_T(u)\} \tag{4.19}$$

In practice, this can be emulated by modifying the loss function into

$$\mathcal{L}(u, v) := K|a_T(u) - a_T(v)|^2 + \sum_{\tilde{T} \in S, \tilde{T} \neq T} |a_{\tilde{T}}(u) - a_{\tilde{T}}(v)|^2, \tag{4.20}$$

and taking $K \gg 1$ (in our numerical tests we took $K = 100$). As explained below, this constrained recovery satisfies similar error bounds as its unconstrained counterpart.

The practical difficulty of the OBERA lies in the quick and accurate computation of the cell averages $a_{S_T}(v)$ for any given cell T and $v \in V_n$, that is, have a fast evaluation procedure for the parameter to average map

$$\mu \in \mathcal{M} \subset \mathbb{R}^n \mapsto a_{S_T}(v_\mu) \in \mathbb{R}^m,$$

as it is needed to calculate $\mathcal{L}(u, v_\mu)$ at each iteration of the optimization algorithm. While

analytic expressions are easily available for linear interfaces, they become more difficult to derive for more general curves like polynomials or circle interfaces. In such cases, the exact computation of $a_S(v_\mu)$ is possible if for each cell $\tilde{T} \in S_T$ one is able identify the points where the curved interface crosses its boundary. We followed this approach in our numerical implementation. Another option relies on quadrature methods as used in [62], however at the expense of potentially many evaluations of v. Finally, another perspective is to use machine learning methods in order to derive a cheaply computable surrogate of the parameter to average map.

Even with such tools in hand, the computation of $a_S(v_\mu)$ for the many parameter values μ that are explored through the optimization process results in a time overhead that one would like to avoid. For linear interfaces, this was achieved by the ELVIRA method, as it only requires 6 calculations of $a_S(v_\mu)$ in order to decide which μ^* should be retained (see Figure 4.9). This can also be avoided for more general interfaces having higher order geometric approximations by the AEROS approach that we present in Section 4.4.

4.3.2 Analysis of the recovery error

In order to prove that the recovery error is near optimal in the $L^1(S)$ norm, we follow a general strategy introduced in [50] which is based on comparing the continuous $L^1(S)$ norm of functions and the discrete $\ell^1(\mathbb{R}^m)$ norm of their cell-averages.

In the 2d case, one obviously has on the one hand the inequality

$$h^2 \|a_S(v)\|_{\ell^1} \leqslant \|v\|_{L^1(S)}, \quad v \in L^1(S), \tag{4.21}$$

which is obtained by summing up

$$h^2 |a_{\tilde{T}}(v)| = \Big| \int_{\tilde{T}} v \Big| \leqslant \|v\|_{L^1(\tilde{T})},$$

over all $\tilde{T} \in S$. This property reflects the stability of the averaging operator, between the continuous and the conveniently normalized discrete norm. Note that in more general dimension d the normalizing factor is h^d.

Conversely, we say that the family V_n satisfies an *inverse stability property* if there exists a constant C independent of h such that

$$\|v - \tilde{v}\|_{L^1(S)} \leqslant C h^2 \|a_S(v) - a_S(\tilde{v})\|_{\ell^1}, \quad \forall\, v, \tilde{v} \in V_n. \tag{4.22}$$

We stress that such a property cannot hold for general pairs of integrable functions, their membership in V_n is critical. Note that this property is a more quantitative version of the injectivity of the map $v \mapsto a_S(v)$ from V_n to $\mathbb{R}^m$. Its validity is thus conditioned to a proper restriction of the classes V_n from the various Examples 1, 2, 3, and 4, as already explained in Section 4.2.2. In the particular case of linear edges the following result was proved in [50].

Theorem 4.3.1. *Let S be the 3×3 stencil centered at T and let V_2 be the family of linear*

interfaces from Example 1, with the restriction that the linear interfaces passes through T. Then (4.22) holds and the best constant is $C = \frac{3}{2}$.

The above stability and inverse stability property allow us to assess the recovery error in the following way. We first write that for any $v \in V_n$

$$\|u - \tilde{u}_T\|_{L^1(T)} \leqslant \|u - v\|_{L^1(T)} + \|v - \tilde{u}_T\|_{L^1(T)} \leqslant \|u - v\|_{L^1(T)} + Ch^2 \|a_S(v) - a_S(\tilde{u}_T)\|_{\ell^1},$$

where we have used (4.22). We then have

$$\|a_S(v) - a_S(\tilde{u}_T)\|_{\ell^1} \leqslant \sqrt{m}\|a_S(v) - a_S(\tilde{u}_T)\|_{\ell^2} \leqslant 2\sqrt{m}\|a_S(v) - a_S(u)\|_{\ell^2},$$

where the first inequality is Cauchy-Schwartz and the second comes by triangle inequality and the ℓ^2 minimization property of $\tilde{u}_T$. Finally, we have

$$\|a_S(v) - a_S(u)\|_{\ell^2}, \leqslant \|a_S(v) - a_S(u)\|_{\ell^1} \leqslant h^{-1}\|u - v\|_{L^1(S)},$$

by using (4.21). Combining all these, and using that $v \in V_n$ is arbitrary, we obtain that

$$\|u - \tilde{u}_T\|_{L^1(T)} \leqslant (1 + 2C\sqrt{m}) e_n(u)_S,$$

which is summarized in the following.

Theorem 4.3.2. *Under (4.21) and (4.22), the recovery by ℓ^2 minimization is near optimal in L^1 norm with multiplicative constant $1 + 2C\sqrt{m}$. In the case of linear interfaces using 3×3 stencils (LVIRA), this constant is $1 + 2\frac{3}{2}\sqrt{9} = 10$.*

As observed in Section 4.2.2, near optimal local recovery allows us to derive convergence rates for smooth domains in terms of the global error estimate (4.16). This gives the following.

Corollary 4.3.3. *If Ω is a $\mathcal{C}^s$ domain with $s \geqslant 1$, the LVIRA method which is OBERA recovery based on linear interfaces converges in L^1 norm at rate $\mathcal{O}(h^r)$ with $r = \min\{s, 2\}$.*

Remark 4.3.4. *Note that if we were using a more general ℓ^p norm for the data fitting, we would obtain a similar result with constant $1 + C2m^{1-\frac{1}{p}}$. The fact that we do not need to restrict ourselves just to $p = 2$ had been mentioned in [129] concerning the LVIRA method.*

As previously remarked, we can modify the OBERA approach in order to impose the consistency condition (4.18), by solving the constrained optimization problem (4.19), that is optimizing inside the subfamily

$$\tilde{V}_n := \{v \in V_n \ : \ a_T(v) = a_T(u)\} \subset V_n$$

It is obvious that if the inverse stability property (4.22) is valid for V_n, it is also valid for the smaller set $\tilde{V}_n$. We thus reach a similar estimate

$$\|u - \tilde{u}_T\|_{L^1(T)} \leqslant (1 + 2C\sqrt{m}) \tilde{e}_n(u)_S,$$

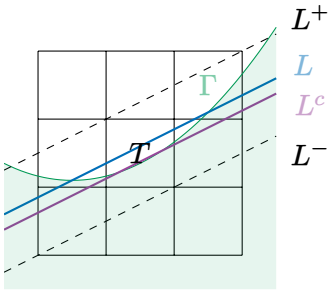


Figure 4.5: By shifting the linear interface L one achieves average consistency on T by a linear interface L^c having the same order of accuracy.

where $\tilde{e}_n(u)_S$ is the error of best approximation of u in $L^1(S)$ norm from the element of $\tilde{V}_n$, that is best approximation from V_n under the consistency constraint.

We thus need to understand if $\tilde{e}_n(u)_S$ satisfies similar size estimates as $e_n(u)_S$. While this cannot be ensured in full generality (for example the set $\tilde{V}_n$ could be empty), the following simple argument shows that this indeed holds in the particular case of linear interface : the estimate (4.7) shows that the function ψ describing the interface in the stencil S satisfies

$$a^- \leqslant \psi \leqslant a^+,$$

where $a^- = a - C_1 h^r$ and $a^+ = a + C_1 h^r$ are two affine functions that parametrize two linear interfaces L^- and L^+ that circumscribe Γ in S, as illustrated on Figure 4.5. For the corresponding halfplanes H^- and H^+ thus satisfy

$$a_T(\chi_{H^-}) \leqslant a_T(u) \quad \text{and} \quad a_T(\chi_{H^+}) \geqslant a_T(u).$$

Therefore, by sliding continuously a linear interface between L^- and L^+, which corresponds to the affine function $a + t$ when t varies in $[-C_1 h^r, C_1 h^r]$, there exists an intermediate interface L^c corresponding to a particular t^c and halfplane H^c for which one has

$$a_T(\chi_{H^c}) = a_T(u).$$

Therefore $v = \chi_{H^c} \in \tilde{V}_n$, and since one also has

$$\|\psi - a^c\|_{L^\infty([-C_0 h, C_0 h])} \leqslant C_1 h^r, \quad r := \min\{s, 2\}, \tag{4.23}$$

we reach the same estimate for $\tilde{e}_n(u)_S$ as the one obtained for $e_n(u)_S$.

From the theoretical perspective, one principle open problem is to establish the inverse stability bound (4.22) for nonlinear families V_n offering higher order approximation properties than the linear interface, for which the proof of Theorem 4.3.1 is already quite involved. This, together with the already mentioned computational complexity of the optimization process, leads us to give up on OBERA for higher order geometrical approximation in favor of the AEROS approach that we next discuss.

4.4 Reconstruction on oriented stencils (AEROS)

4.4.1 Presentation of the method

Algorithms for Edge Reconstruction using Oriented Stencils (AEROS) are based on recovering an element from the family V_n presented in Example 3. We thus recover on each singular cell $T \in \mathcal{S}_h$ a domain having one of the four subgraph or epigraph forms (4.6), that is, an interface having one of the two Cartesian forms $y = p(x)$ or $x = p(y)$, where $p \in W_n$ a linear space of dimension n.

More specifically we consider for some fixed $k \geqslant 1$ the space

$$W_{2k+1} = \mathbb{P}_{2k},$$

of polynomials of even degree $2k$. We then use stencils S_T containing T of the form $(2k+1) \times L$ or $L \times (2k+1)$, when reconstructing an interface of the form $y = p(x)$ or $x = p(y)$ respectively, for some $L > 0$. As we further explain, the value of L and the exact positioning of T inside S_T may depend on the considered cell T.

As a first step we need to identify for each $T \in \mathcal{S}_h$ the exact orientation of the subgraph or epigraph that we decide to use. The decision must be based on the available data of the cell averages . We have already noticed that if Ω is a $\mathcal{C}^s$ domain for $s > 1$, then it can itself be locally described by (at least) one of the four forms with a function $\psi \in \mathcal{C}^s$ describing the interface, as expressed by (4.9) and (4.10) in Definition 4.2.3. Our objective is that our choice of form in the recovery is consistent with the form of the exact interface over the stencil S_T for each cell T.

We thus need to identify for each T an orientation $y = \psi(x)$ or $x = \psi(y)$ for the exact interface over the stencil S_T. Let us immediately observe that this is only possible if h is below a certain critical resolution $h^* = h^*(\Omega)$ that depends on the amount of variation of the tangent to the interface, as shown on Figure 4.6 for the case $k = 1$ (stencils of widths 3).

More precisely, when Ω is a $\mathcal{C}^s$ domain with $s > 1$, the variation of the slope of the tangent to the interface between two points z and z' is controlled by a bound of the form $M|z - z'|^r$ where $r := \min\{1, s-1\}$ and M the bound on the $\mathcal{C}^s$ norm of the functions ψ that describe the interface. Therefore a given orientation, say $y \leqslant \psi(x)$, can be maintained on a stencil S_T of width $2k+1$ in the x direction provided that $h \leqslant h^* \sim k^{-1} M^{-1/r}$.

For identifying the orientation, we introduce a selection mechanism based on a numerical gradient computed by the Sobel filter. We denote by T_{e} with $\mathrm{e} = (\mathrm{e}_x, \mathrm{e}_y) \in \{-1, 0, 1\}^2$ the cells in the 3×3 stencil centered around $T = T_{0,0}$ where e_x and e_y indicate the amount of displacement by h from T in the x and y direction, respectively. We then define the numerical gradient

$$G_T = (H_T, V_T),$$

with horizontal component

$$H_T := 2a_{T_{1,0}} + a_{T_{1,1}} + a_{T_{1,-1}} - (2a_{T_{-1,0}} + a_{T_{-1,1}} + a_{T_{-1,-1}})$$

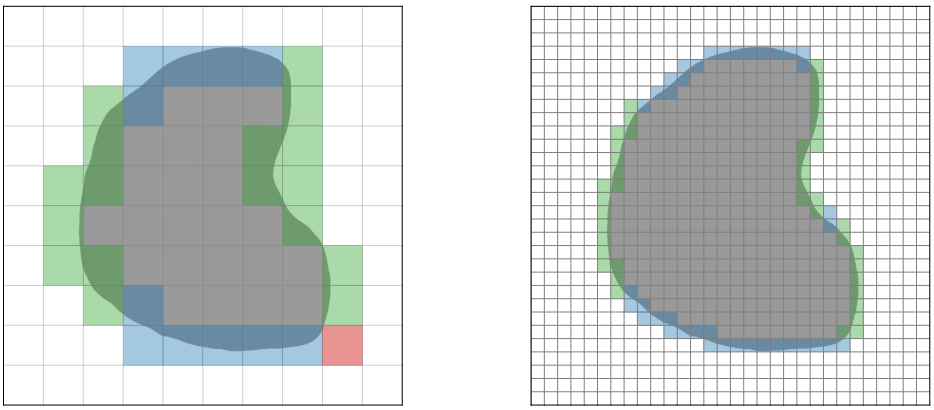


Figure 4.6: On the left $h = 1/10$ and on the right $h = 1/30$. The singular cells idenfied as horizontally and vertically oriented are pictured in blue and green. For $h = 1/10$, there exists a singular cell T (indicated in red) for which the interface cannot be described by a graph $y = \psi(x)$ or $x = \psi(y)$ in any stencil S_T of width 3 that contains T. This is no more the case for $h = 1/30$ when using an adaptive selection of the stencil.

obtained by convolution between the cell averages $a_{T_e} = a_{T_e}(u)$ and the horizontal Sobel kernel (see Figure 4.7). Similarly, the vertical component is defined as

$$V_T := 2a_{T_{0,1}} + a_{T_{1,1}} + a_{T_{-1,1}} - (2a_{T_{0,-1}} + a_{T_{1,-1}} + a_{T_{-1,-1}}).$$

The selection mechanism is based on comparing the absolute size of H_T and V_T and examining their sign. More precisely:

1. If $|V_T| \geqslant |H_T|$ and if $V_T \leqslant 0$, we search for a subgraph of the form $y \leqslant p(x)$.
2. If $|V_T| \geqslant |H_T|$ and if $V_T > 0$, we search for an epigraph of the form $y \geqslant p(x)$.
3. If $|V_T| < |H_T|$ and if $H_T \leqslant 0$, we search for a subgraph of the form $x \leqslant p(y)$.
4. If $|V_T| < |H_T|$ and if $H_T > 0$, we search for an epigraph of the form $x \geqslant p(y)$.

One important result is that this selection mechanism correctly detects the orientation of the exact interface for h sufficiently small, as also illustrated on Figure 4.6.

Theorem 4.4.1. *Let Ω be a $\mathcal{C}^s$ domain for some $s > 1$, then there exists $h^* = h^*(\Omega)$ such that under the assumption $h < h^*$, the following holds for any $T \in \mathcal{S}_h$: in each of the above cases $(1, 2, 3, 4)$ of the selection mechanism, the exact domain Ω can be described by an equation of the same form with p replaced by a function $\psi \in \mathcal{C}^s$ over the stencil S_T centered at T and of size $(2k+1) \times (2l+1)$ in case $(1, 2)$ or $(2l+1) \times (2k+1)$ in case $(2, 3)$ with $l = k + 2$. The graph of ψ remains confined in S_T in the following sense: denoting by $I \times J := \bigcup\{\tilde{T} \ : \ \tilde{T} \in S_T\}$ the total support of S_T, one has $\psi(I) \subset J$ in case $(1, 2)$ and $\psi(J) \subset I$ in case $(3, 4)$.*

We postpone the proof of this result to the Appendix, and proceed with the description and error analysis of the recovery method. We place ourselves in case 1 without loss of

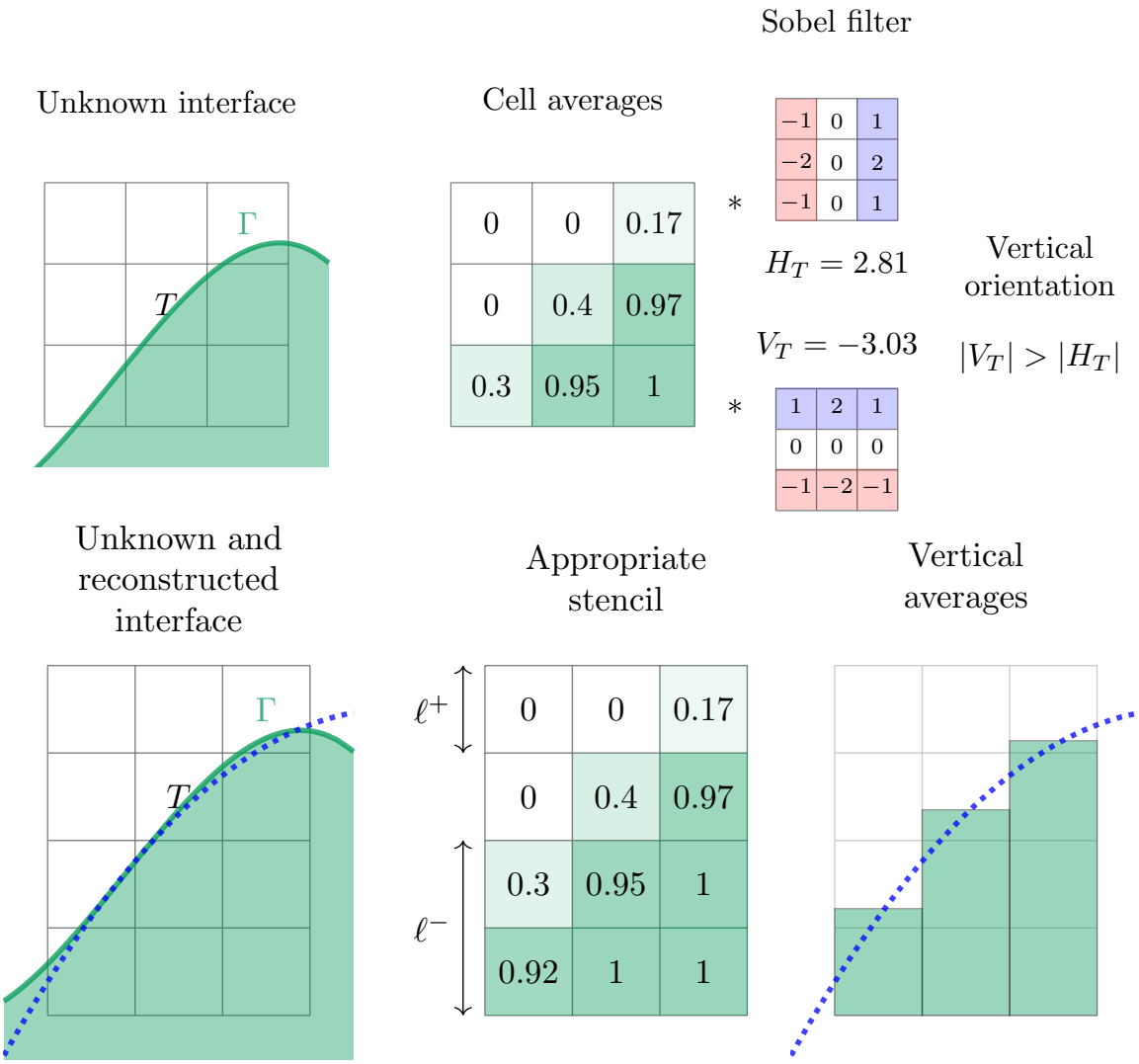


Figure 4.7: Steps for computing AEROS method. First the preferable orientation is extracted from a 3×3 stencil centered around T by applying the Sobel filter (a convolution with the horizontal and vertical Sobel kernels). Then an appropriate stencil is found so that the interface crosses the sides of the stencil. Finally, the column averages are calculated and a polynomial is used to approximate the interface.

generality since all other cases are dealt with similarly up to an obvious exchange of x and y or change of sign in one of these variable.

According to the above theorem, we are ensured that Ω is characterized by the equation $y \leqslant \psi(x)$ when $(x, y) \in S_T$ where S_T is a stencil of size $(2k+1) \times (2l+1)$ with $l = k+2$. The choice $l = k + 2$ is conservative and our numerical experiments show that it can sometimes be reduced while maintaining the property that the graph of ψ remains confined in S_T. In practice we use the following adaptive strategy to use a stencil of minimal vertical side.

By convention, we denote by (i, j) the coordinates of a generic cell $\tilde{T}$ when the lower left corner of $\tilde{T}$ is (ih, jh). Let (i_T, j_T) be the coordinates of the singular cell T. We explore the neighboring cells by defining

$$l^- := \min\{l > 0 \ : \ a_{\tilde{T}}(u) = 1, \ i = i_T - k, \dots, i_T + k, \ j = j_T - l - 1\},$$

which is the smallest lower shift below which we find a row of non-singular cells. Likewise, we define

$$l^+ := \min\{l > 0 \ : \ a_{\tilde{T}}(u) = 0, \ i = i_T - k, \dots, i_T + k, \ j = j_T + l + 1\}.$$

Then, we take for S_T the stencil that consists of cells $\tilde{T}$ of coordinates (i, j) for $i = i_T - k, \dots, i_T + k$ and $j = j_T - l^-, \dots, j_T + l^+$. This stencil has size $(2k+1) \times (1 + l^- + l^+)$ and is centered around the cell T horizontally but not vertically. From the definition of l^- and l^+ we have the guarantee that the graph of ψ remains confined in S_T (see Figure 4.7). One option to further adapt the stencil S_T is to allow that it is also not centered horizontally but still contains T. This leads to $(2k+1) \times (1 + l^- + l^+)$ stencils corresponding to values $i = i_T - k^-, \dots, i_T + k^+$ and $j = j_T - l^-, \dots, j_T + l^+$, where $k^-, k^+ \geqslant 0$ are such that $k^- + k^+ = 2k$ and are selected so to minimize the vertical size $1 + l^- + l^+$.

Once the stencil S_T has been selected, the polynomial $p_T \in \mathbb{P}_{2k}$ is constructed as follows. For each $i = i_T - k^-, \dots, i_T + k^+$, we denote by R_i the column that consists of the cells $\tilde{T} \in S_T$ with first coordinate equal to i and define the corresponding column average

$$a_i(u) = h \left(\sum_{\tilde{T} \in R_i} a_{\tilde{T}}(u) \right).$$

Since the graph of ψ remains confined in S_T, it follows that $a_i(u)$ can be identified to the univariate cell average of ψ on the interval $[ih, (i+1)h]$ after having subtracted the base elevation of the stencil (see Figure Figure 4.7), that is,

$$a_i(u) + b_T = \frac{1}{h} \int_{ih}^{(i+1)h} \psi(x) dx, \quad i = i_T - k^-, \dots, i_T + k^+, \quad b_T := (j_T - l^-)h.$$

We then define $p_T \in \mathbb{P}_{2k}$ as the unique polynomial of degree at most $2k$ that agrees with the observed averages of ψ, that is, such that

$$\frac{1}{h} \int_{ih}^{(i+1)h} p_T(x) dx = a_i(u) + b_T \quad i = i_T - k^-, \dots, i_T + k^+.$$

The polynomial p_T is sometimes called the interpolant of averages, and its existence and uniqueness is standard, similar to the more usual Lagrange interpolant of point values. In particular, it is easily checked that the $\mathbb{P}_{2k}$ interpolant of the averages of a function v on $2k+1$ adjacent intervals is the derivative of the $\mathbb{P}_{2k+1}$ Lagrange interpolant at the $2k+2$ interval endpoints for the primitive of v.

Quite remarkably, although we are locally approximating u by a nonlinear family, we observe that the recovery map

$$L_T : \psi \mapsto p_T,$$

is linear, and that the AEROS recovery approach amounts to solving a simple $(2k+1)\times(2k+1)$ linear system, resulting in substantial computational saving compared to the OBERA approach.

4.4.2 Analysis of the recovery error

In order to assess the recovery error, we first observe that the above described strategy has the property of exact recovery for polynomials

$$\psi \in \mathbb{P}_{2k} \implies p_T = \psi, \tag{4.24}$$

due to the uniqueness of the interpolant of averages. In other words, AEROS recovers on T the true interface if it is described by a polynomial of degree $2k$ on S_T.

Our next observation is that the linear application L_T is stable in the max norm over the relevant interval $I_T = [(i_T - k^-)h, \dots, (i_T + k^+)h]$, with stability constant that does not depend on h. This can be proved by making the affine change of variable

$$x = \varphi(\hat{x}) = h(i_T + \hat{x}),$$

that maps the reference interval $\hat{I} := [-k^-, \dots, k^+]$ onto I_T. Then, it is readily checked that

$$L\psi \circ \varphi = \hat{L}(\psi \circ \varphi),$$

where $\hat{L}$ is the average interpolant for the intervals of size 1 contained in $\hat{I}$. Therefore, we may write

$$\|L\psi\|_{L^\infty(I_T)} = \|L\psi \circ \varphi\|_{L^\infty(\hat{I})} = \|\hat{L}(\psi \circ \varphi)\|_{L^\infty(\hat{I})} \leqslant C\|\psi \circ \varphi\|_{L^\infty(\hat{I})} = \hat{C}\|\psi\|_{L^\infty(I_T)}$$

where the constant $\hat{C}$ is the norm of $\hat{L}$ acting on $L^\infty(\hat{I})$. This constant only depends on k.

We are now in position to obtain an error estimate by writing for all $p \in \mathbb{P}_{2k}$,

$$\|\psi - p_T\|_{L^\infty(I_T)} \leqslant \|\psi - p\|_{L^\infty(I_T)} + \|p_T - p\|_{L^\infty(I_T)} = \|\psi - p\|_{L^\infty(I_T)} + \|L(\psi - p)\|_{L^\infty(I_T)} \leqslant (1+\hat{C})\|\psi - p\|_{L^\infty(I_T)},$$

where we have combined exact recovery of polynomials and uniform stability. Since p is arbitrary we have obtained the following result.

Theorem 4.4.2. *The AEROS recovery of the interface based on polynomials of degree $2k$ satisfies for each singular cell the near optimality property*

$$\|\psi - p_T\|_{L^\infty(I_T)} \leqslant (1 + \hat{C}) \min_{p \in \mathbb{P}_{2k}} \|\psi - p\|_{L^\infty(I_T)}. \tag{4.25}$$

This error bound of ψ in L^∞ readily induces an $L^1(T)$ error bound between the recovery

$$R_T(a_S(u)) = u_T := \chi_{\{y \leqslant p_T(x)\}}$$

and u by multiplying by the width of the cell. This gives

$$\|u - R_T(a_S(u))\|_{L^1(T)} \leqslant (1 + \hat{C})h \min_{p \in \mathbb{P}_{2k}} \|\psi - p\|_{L^\infty(I_T)}. \tag{4.26}$$

Note that the right-side is not $e_n(u)_S$ and we thus have not obtained the near-optimal recovery property in the form (4.13). Nevertheless we can derive from (4.26) the same convergence rates since these are based on the Taylor polynomial approximation error (4.11), which thus yields the following result.

Theorem 4.4.3. *Let Ω be a $\mathcal{C}^s$ domain for some $s \geqslant 1$. The AEROS recovery of the interface based on polynomial of degree $2k$ satisfies for each singular cell T a local error bound of the form*

$$\|u - R_T(a_S(u))\|_{L^1(T)} \leqslant Ch^{r+1}, \quad r := \min\{s, 2k+1\}, \tag{4.27}$$

and the global error bound (4.16) of order $\mathcal{O}(h^r)$ for the same value of r.

4.5 Numerical experiments

4.5.1 Recovery of smooth domains

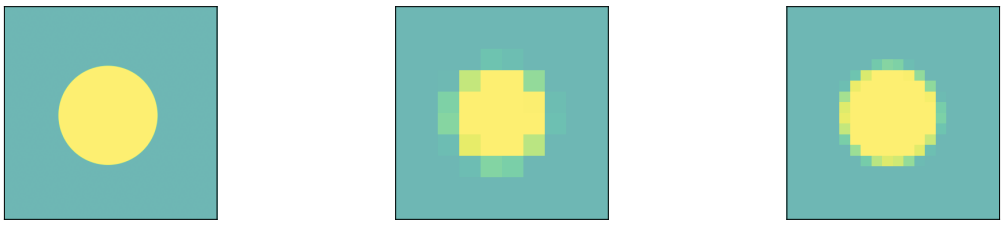
In this section, we compare various recovery strategies in terms of:

1. visual aspects,
2. quantitative rate of convergence,
3. computational time.

In order to draw the second comparison, we first consider the simple case of a domain Ω with circular boundary (see Figure 4.8) for which the recovery error can be computed within machine precision.

The following eight reconstruction strategies are compared:

1. **Piecewise Constant**: as mentioned in the introduction, the simplest linear method that one can come up with is $\tilde{u} := \sum_{T \in \mathcal{T}_h} a_T(u)\chi_T$, that is, on each cell T the value given by the observed cell average $a_T(u)$.
2. **OBERA Linear**: we apply the minimization strategy described in Section 4.3 for *linear interfaces* as in Example 1, with loss function $\mathcal{L}(u, v) = \|a_S(u) - a_S(v)\|$ using ℓ^1 norm and a 3×3 stencil S.
3. **OBERA-W Linear**: we apply the same approach but enforcing area consistency on T through the weighted loss function (4.20) with $K = 100$.



(a) Circle. (b) $h = 1/10$. (c) $h = 1/20$.

Figure 4.8: Circular domain and its cell average data at different scales of refinement.

4. **ELVIRA**: following [129], the ELVIRA method consists on replacing, still for *linear interfaces*, the OBERA continuous optimization strategy by providing only 6 combinations of parameters μ from which to choose μ^*, the minimizer of $\mathcal{L}$ with ℓ^2 norm instead of ℓ^1. The 6 alternatives are obtained by proposing, for the interface's slope, the 6 possible finite differences estimations of a 3×3 stencil (see Figure 4.9).

5. **ELVIRA-WO**: we apply ELVIRA but choosing first an *orientation*, as in AEROS, which allows to reduce the choice to 3 alternatives. In addition, we work with the modified *weighted* loss function, with $K = 100$, to favor area consistency on T.

6. **OBERA Quadratic**: we apply the minimization strategy for *quadratic interfaces* after choosing an orientation to have a Cartesian parametrization of the interface, that is, $v_\mu = \chi_P$, as in Example 3, and $p \in \mathbb{P}_2$ the space of univariate polynomials of degree 2. Here, we also use a 3×3 fixed stencil and enforce area consistency on T through the same modified loss function (4.20) with $K = 100$.

7. **AEROS Quadratic**: we apply the AEROS reconstruction strategy for *quadratic interfaces* with the adaptive method, described in Section 4.4.1, to build stencils of width 3 minimal height.

8. **AEROS Quartic**: we apply the AEROS strategy now with polynomials of degree 4, therefore with stencils of width 5.

Figure 4.10 and Figure 4.11. display a detail of the recovery with $h = 1/10$ and $h = 1/20$ respectively in order to compare the visual quality.

As to linear interfaces, we clearly notice the relevance of enforcing area consistency on the cell T of interest by appropriately modifying the loss function $\mathcal{L}$. Although the four methods benefit from similar convergence rates of $\mathcal{O}(h^2)$ as expected from Corollary 4.3.3 and Theorem 4.4.3, see Figure 4.12, the reconstruction error improvement is of an order of magnitude by this simple change. When area consistency is not imposed, the linear interfaces are pushed towards the interior of the circle as the curvature of the original domain points inwardly.

As to the AEROS strategies, we notice that for $h = 1/10$, they have difficulties to reconstruct the interface on cells where Γ is rapidly passing from a situation where an

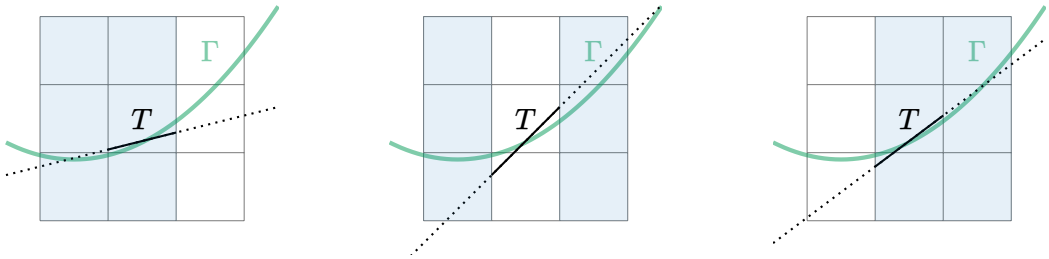


Figure 4.9: ELVIRA 3 cases for the vertical orientation. The slope is estimated using the differences between the two first column averages (left), the first and third (center) and the last two columns (right).

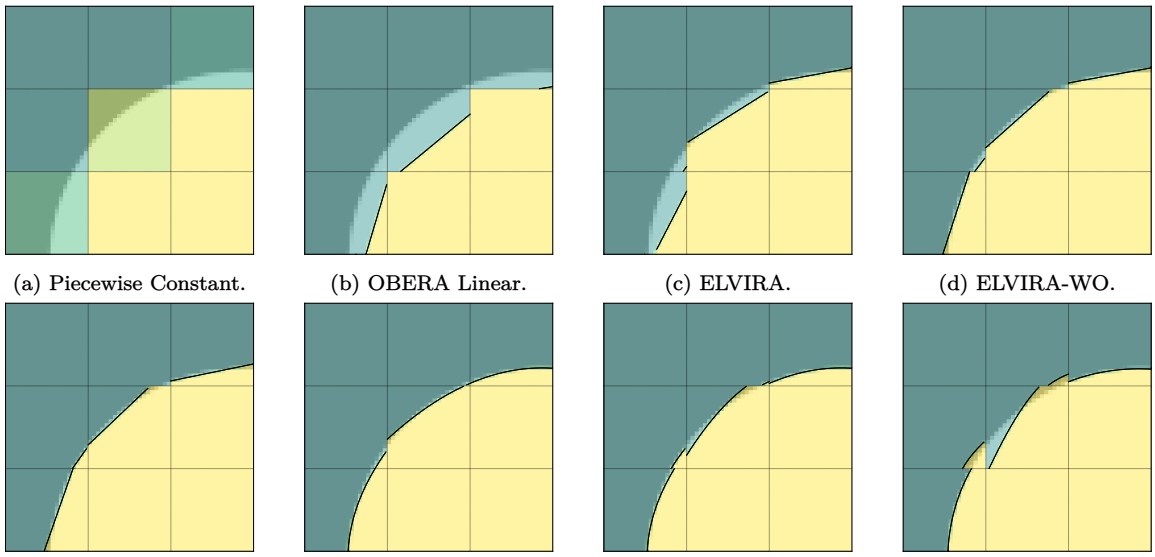
horizontal orientation is preferred to another in which a vertical one is better. This effect if particulary evident for the case of AEROS Quartic as the method needs a stencil with 5 columns. At this scale we are above the critical scale (see Figure 4.6) in which for some cells there is no stencil of the needed width allowing the curve to be described as a graph. This problem disappears for $h = 1/20$, for which these methods have the best visual quality.

On Figure Figure 4.12, we see that in terms of convergence we obtain for both AEROS the expected rates from Theorem 4.4.3: for quadratics we get $\mathcal{O}(h^3)$ and we almost get $\mathcal{O}(h^5)$ for polynomials of degree 4. As mentioned before and graphically shown in Figure 4.10, AEROS Quartic breaks down when the scale of the discretization is above the critical scale which, for this particular, example is around $h = 1/20$.

Regarding the computational time per cell taken by each algorithm we observe on Table 4.1 that OBERA strategies are two orders of magnitude slower than any AEROS approach. At the same time, although ELVIRA methods are faster than OBERA, they still are an order slower than AEROS due to the bottleneck of having to compute the stencil cell averages to compare the 6 or 3 alternatives under evaluation. This limit is justified by the fact that choosing an orientation, as in ELVIRA-WO, cuts by half the computing time of the overall algorithm, while the improvement in accuracy is achieved by modifying the loss function.

In summary, all three comparisons in terms of visual aspect, order of convergence and computational time are in favor of the AEROS strategy provided that h is below the critical scale.

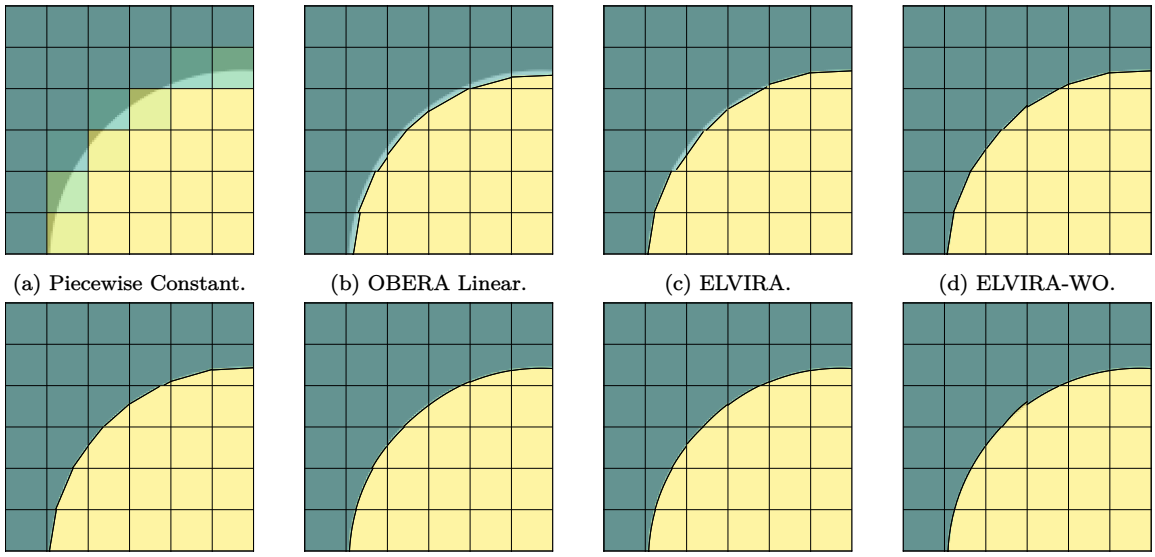
Finally, we show in Figure 4.13 how the best linear interface method, OBERA-W Linear, and the three higher order methods compare when used to reconstruct an arbitrary, still smooth, domain. We see that by passing from the linear interface method to AEROS Quadratic the reconstruction becomes smoother while still suffering from some imperfections, notably in regions where there is a stronger change in the orientation of the curve. This is slightly improved by the optimization done in OBERA Quadratic method. An even smoother result is obtained with AEROS Quartic at the expense of some small deviations again in regions where the orientation is rapidly changing.



(a) Piecewise Constant. (b) OBERA Linear. (c) ELVIRA. (d) ELVIRA-WO.

(e) OBERA-W Linear. (f) AEROS Quadratic. (g) OBERA Quadratic. (h) AEROS Quartic.

Figure 4.10: Reconstruction of a portion of the circle by different methods for a scale of $h = 1/10$.



(a) Piecewise Constant. (b) OBERA Linear. (c) ELVIRA. (d) ELVIRA-WO.

(e) OBERA-W Linear. (f) AEROS Quadratic. (g) OBERA Quadratic. (h) AEROS Quartic.

Figure 4.11: Reconstruction of a portion of the circle by different methods for a scale of $h = 1/20$.

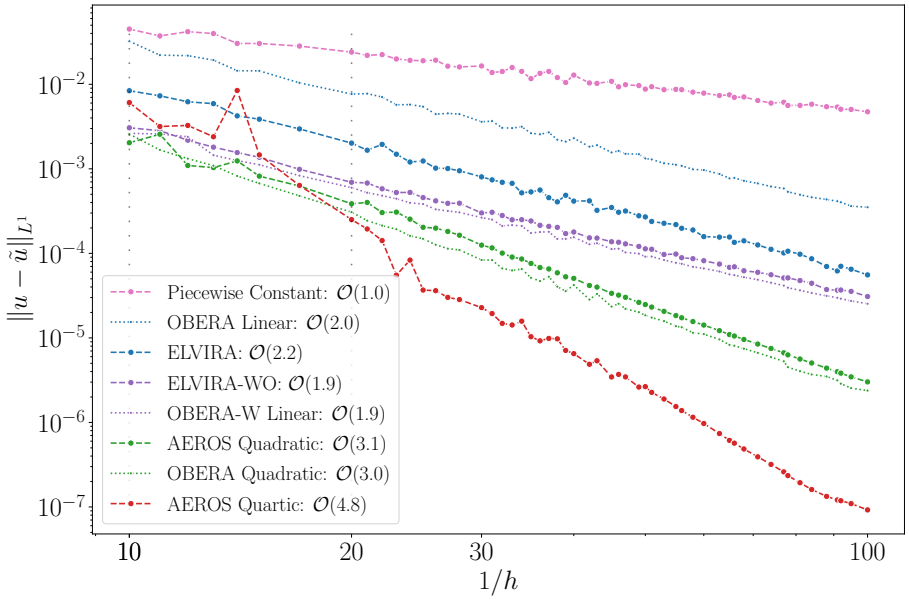
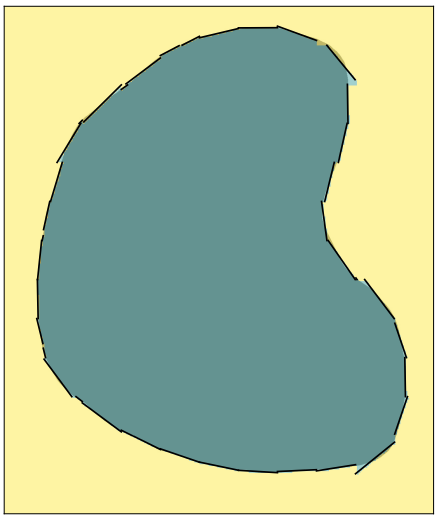


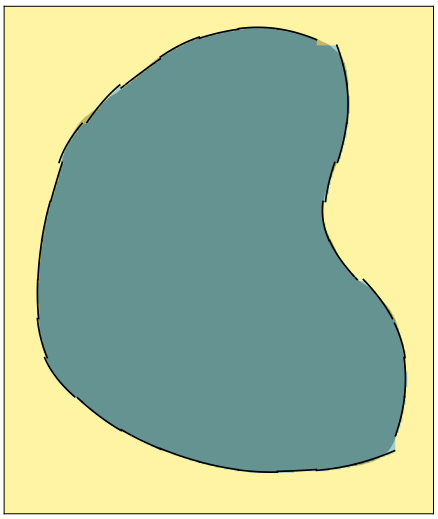
Figure 4.12: Convergence for different reconstruction models. The convergence rates (in parenthesis) are estimated using values $1/h > 30$.

OBERA Linear	**0.8**
OBERA-W Linear	**0.7**
OBERA Quadratic	**0.3**
ELVIRA	**0.04**
ELVIRA-WO	**0.02**
AEROS Quadratic	**0.003**
AEROS Quartic	**0.003**

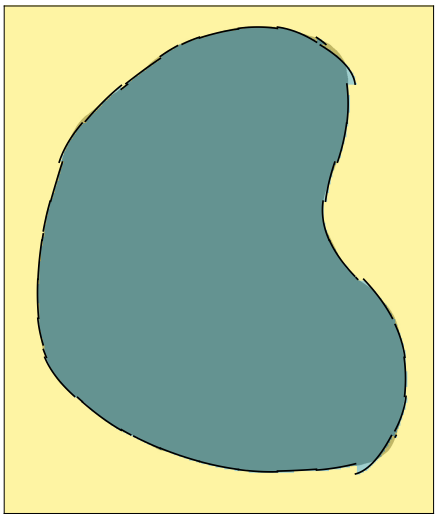
Table 4.1: Average time (in seconds) taken to find the parameters of the interface by the different tested models. The average is taken over all instances in which each algorithm was called (to perform a local approximation) to produce Figure 4.12 (which is in the order of the 4000 per method).



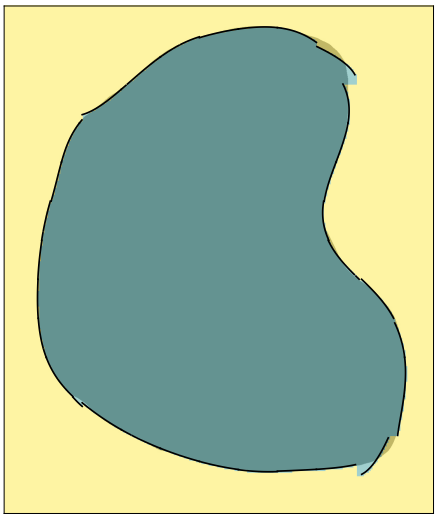
(a) OBERA-W Linear.



(b) AEROS Quadratic.



(c) OBERA Quadratic.



(d) AEROS Quartic.

Figure 4.13: Reconstruction of a smooth domain by different methods for a scale of $h = 1/15$. The higher the order the smoother the reconstruction except in the transitions of orientation where higher order methods like AEROS Quartic struggle because there is not enough information or the stencil is too much de-centered.

4.5.2 The treatment of corner domains

The previously tested strategies achieve to reconstruct, at their corresponding orders, different smooth domains with Hölder smoothness $s > 1$. This excludes the case of domains with piecewise smooth boundaries for which the presence of corners call for a specific treatment. The simplest option that we study here is to use local recovery by the approximation family V_4 of piecewise linear interface from Example 4. We propose, in what follows, two methods to deal with vertices of any angle, bearing in mind the limitations expressed in Figure 4.4c.

AEROS Vertex: The first method applies the AEROS strategy for $v \in \hat{V}_4$ a restriction of V_4 where $\pi/2 < \theta_1 < 3\pi/2$ and $-\pi/2 < \theta_2 < \pi/2$, *i.e.* elements v whose interface can be written as an oriented graph, which is a particular instance of Example 3 where instead of searching p in a space of univariate polynomials, we pick it into W_4 the space of piecewise functions with one breakpoint contained in T or its immediate neighbours. This excludes the possibility of reconstructing a rectangle whose sides are parallel to the mesh but it applies to the case of the same rectangle slightly rotated. Under this restrictions it is possible, although lengthy, to extract explicit equations that allow us to derive a finite set of admissible parameters $\mu = (x_1, x_2, \theta_1, \theta_2)$ from the observed cell averaged vector a_S. We compare each proposed local approximation $v \in \hat{V}_4$ as in ELVIRA or OBERA, that is, by means of their associated loss $\mathcal{L}(u, v)$ while retaining at the end the one achieving the minimal value between the many possibilities. This same model selection strategy can be used to aggregate other competing models, like quadratic interfaces. This has the effect of keeping higher order models when the interface is locally smooth, while taking corners into account as illustrated on Figure 4.14. By this approach we avoid defining a vertex detection mechanism at the expense of computational overhead as we now need to compute for each cell many losses, which was already the time bottleneck for ELVIRA method.

Tangent Extension Method (TEM): The above restriction that the interface with a vertex needs to be an oriented graph could be limiting in some applications but it can be removed at the expense of complexifying the reconstruction procedure. Our second proposed method deals with this aspect and consists in the following steps:

1. Associate to each singular cell $T \in \mathcal{S}_h$ some reconstruction v_T stemming from any of the local interface reconstruction methods discussed discussed so far.

2. For each cell where the presence of a vertex is suspected (eventually for all $T \in \mathcal{S}_h$) we search for two singular cells $T_1, T_2 \in \mathcal{S}_h$ satisfying the following

   - $T \neq T_1 \neq T_2 \neq T$
   - $S_{T_1} \cap \{T\} = \emptyset$
   - $S_{T_2} \cap \{T\} = \emptyset$

   where $S_{v_{T_i}}$ denotes the stencil S used by a given smooth interface reconstruction method (for example linear or quadratic) to produce local approximations v_{T_i}.

3. Take the parameterized interfaces Γ_1 and Γ_2, associated to v_{T_1} and v_{T_2} respectively, and do an order 1 Taylor expansion at an intermediate point between cells (T, T_1) and (T, T_2) respectively. This yields the parameters of the two half planes H_1 and H_2 of Example 4 needed to define $v \in V_4$.

4. Finally, we compare the new local approximation $v \in V_4$ with the existing one v_T, as explained above, retaining only the one whose associated loss, $\mathcal{L}(u, v)$ or $\mathcal{L}(u, v_T)$, is minimal.

This procedure will reconstruct exactly corners when the interface is a line along both directions, but it will not produce area-consistent reconstructions on cell T otherwise. In this regard, AEROS Vertex, being based on AEROS strategy, will yield interfaces that are, though not cell-consistent as one could get with OBERA, at least column-consistent as long as we remain in the interpolation case. This is ensured if the stencil width equals the number of parameters of the approximating class which in the case of $\hat{V}_4$ is guaranteed by using 4-width stencils.

Figure 4.14 displays the successive improvements in the reconstruction when combining the different strategies described so far:

- On Figure 4.14 (up-left), we use the ELVIRA-WO method. We observe that it recovers the interface in a satisfactory manner only far enough from corners.
- On Figure 4.14 (up-right), we use the AEROS Quadratic method. We observe that it recovers the interface in a satisfactory manner only far enough from corners.
- On Figure 4.14 (center-left), we first find a curve for each singular cell using AEROS Quadratic and then TEM. In this case, some of the problems are solved, in particular corners with a 90° angle and parallel to the grid.
- On Figure 4.14 (center-right), we add to the previous method the first proposed approach, based on AEROS for vertices. We obtain almost perfect results except for some cells where the quadratic approximation of the interface given by AEROS Quadratic was not replaced by a better one.
- On Figure 4.14 (down), this last issue is addressed by aggregating, before applying any of the vertex mechanisms described before, an ELVIRA-WO strategy to offer an alternative when AEROS Quadratic is too much affected by the presence of a nearby corner.

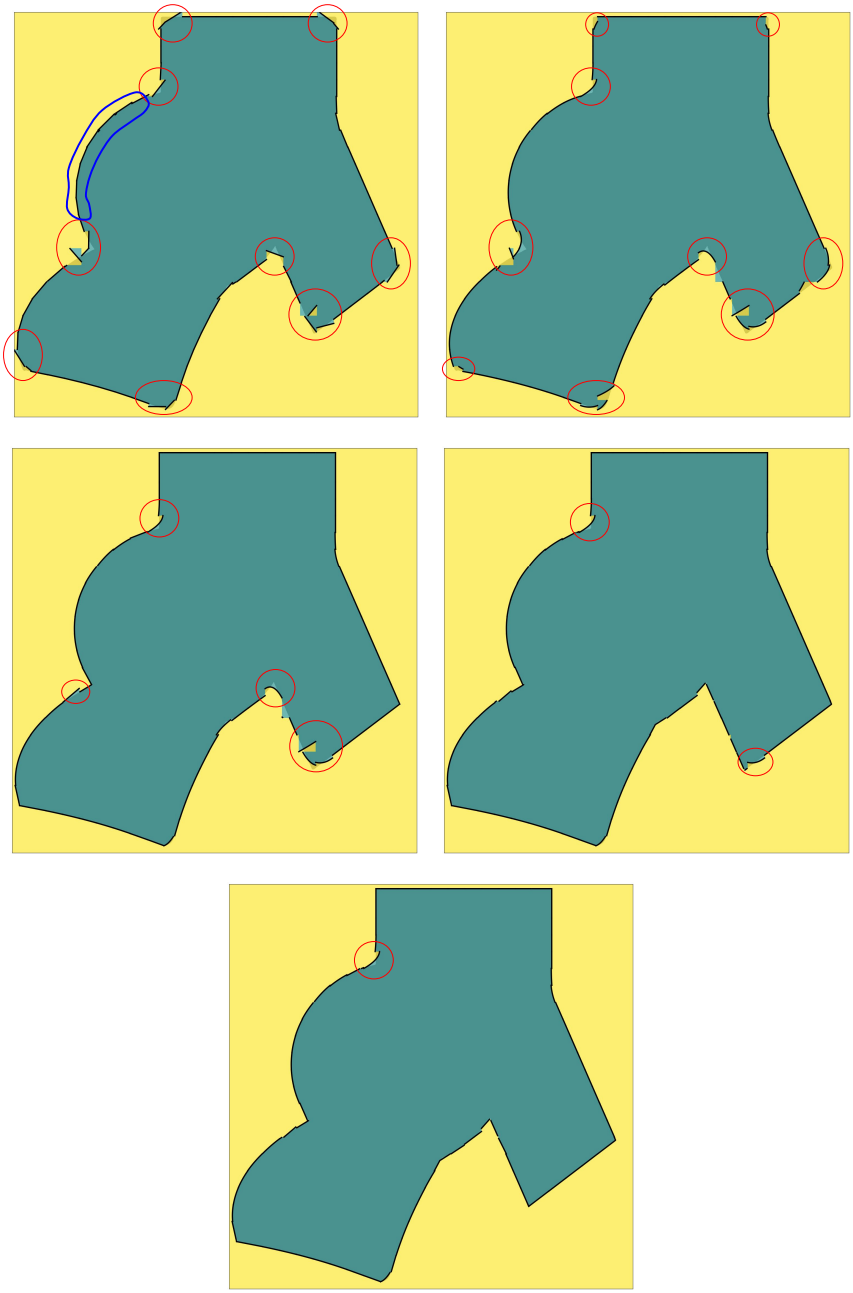


Figure 4.14: Reconstruction of a domain with corners from cell-averages of scale $h = 1/30$. Reconstructions made with ELVIRA-WO (up-left), AEROS Quadratic (up-right), AEROS Quadratic + TEM (center-left), AEROS Quadratic + TEM + AEROS Vertex (center-right) and finally ELVIRA-WO + AEROS Quadratic + TEM + AEROS Vertex (down). The red markings show the problems around vertices and how they are progressively resolved when one puts all the methods to work together.

4.5.3 Finite volume evolution in time

Finally, we use the interface recovery methods presented so far as a constituent part of a finite volume solver with the objective of reducing numerical dissipation. We study the particular case of a simple linear transport PDE:

$$\frac{\partial u}{\partial t} + b \cdot \nabla u = 0,$$

in the unit square domain $D = [0,1]^2$ with periodic boundary conditions and initial condition being a piecewise constant function $u^0 : D \to \{0,1\}$ with Ω limited by a smooth interface as in Figure 4.13 or an interface having corners as in Figure 4.14.

As in a large class finite volume schemes, the reconstruction is used at each step to compute the flux that updates the averages at the next step. For simplicity of the presentation we have set a constant (both in space and time) velocity field $b = (h/4, 0)$ and worked with unit time steps $\Delta t = 1$ and coarse grids of size $h = 1/30$, so that the CFL condition is maintained. In this case, the numerical flux induced by a local reconstruction $u^k_{i,j}$ on a cell T of coordinate (i,j) at time step k takes the form

$$\mathcal{F}(u^k_{i,j}) := \frac{1}{|R_T|} \int_{R_T} u^k_{i,j}(x) dx$$

where $R_T = [(i+1)h - b, (i+1)h] \times [jh, (j+1)h]$. The finite volume approximation at the next time step $k+1$ is then given by the updated cell-average

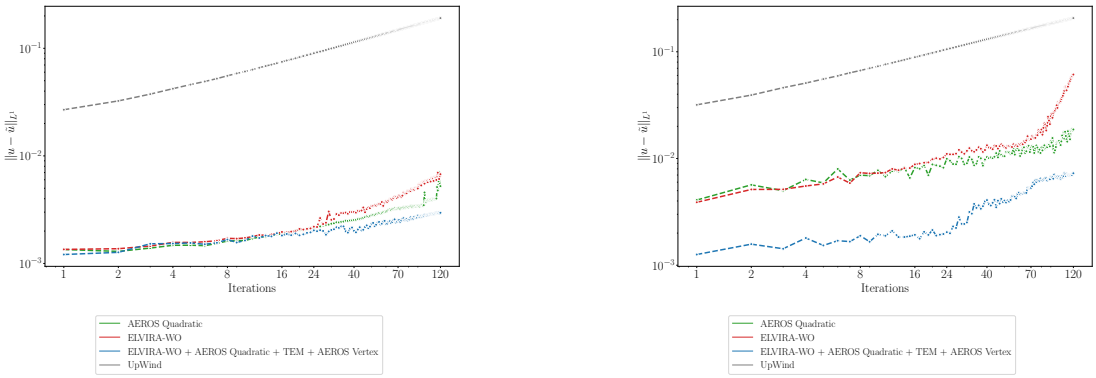
$$a^{k+1}_{i,j} = a^k_{i,j} + \mathcal{F}(\tilde{u}^k_{i-1,j}) - \mathcal{F}(\tilde{u}^k_{i,j}).$$

Figure 4.15a displays the evolution of the L^1 error between the exact solution and the reconstruction, for the time evolution of a smooth domain. In this case, all three methods ELVIRA-WO, AEROS Quadratic and ELVIRA-WO + AEROS Quadratic + TEM + AEROS Vertex behave similarly with an error that is maintained at the same level for all tested times showing that numerical dissipation has been avoided on the three cases. This contrasts with the piecewise constant reconstruction that corresponds to the standard upwind scheme.

Figure 4.15b shows the effect of the presence of corners in the interface of Ω in terms of a slow but accumulative deterioration in both methods ELVIRA-WO, AEROS Quadratic as they are not design to treat vertices. In contrast, ELVIRA-WO + AEROS Quadratic + TEM + AEROS Vertex, as before, keeps its error on the same level for all the times tested.

4.6 Conclusion and perspectives

In this work, we have presented several interface recovery methods. For the two main classes OBERA and AEROS, we have provided general analysis strategies for establishing convergence rates that depend on the geometric smoothness of the interface. From a practical perspective, the methods can be combined with the aggregation strategy outlined in the



(a) Smooth domain.

(b) Domain with corners.

Figure 4.15: Time evolution of the finite volume scheme error for $h = 1/30$.

previous section, and we have made them available in the open-source python package [2].

Several natural perspectives are foreseen (and we have explored some of them already in the open-source package):

1. Address the reconstruction of more general piecewise smooth functions with jump discontinuities across geometrically smooth or piecewise smooth interfaces. This requires a proper adaptation of the interface recovery strategies, combined with a high-order treatment of the smooth part of the function corresponding to the cell $T \notin \mathcal{S}_h$. The latter can be done by using polynomial reconstructions on stencils not containing cells of $\mathcal{S}_h$ following the standard ENO strategy.

2. Study the use of machine learning techniques trained on sufficiently rich sets of interfaces for performing certain tasks in an automated and hopefully more efficient manner. Such tasks include, for example, the fast reconstruction of the parameter μ from the cell averages, the identification of cells that may contain vertices, or the direct access to the numerical flux in the case of finite volume scheme, as proposed for example in [62] for vertices forming angles of $\phi = 90°$. It should however be noted that, as opposed to the approaches that we developped in this paper, the machine learning-based approach does not offer rigorous convergence guarantees. Also, our attempts to beat the AEROS method with machine learning strategies were so far unsuccessful, both from the accuracy point of view, and the runtime point of view. We provide our implementation of these strategies in the Python package.

4.A The orientation test

In this apprendix we give the proof of Theorem 4.4.1, which is based on:

1. First studying the case where the interface Γ is a line over the 3×3 stencil where the numerical gradient $G_T = (H_T, V_T)$ is computed.

[2] https://github.com/agussomacal/SubCellResolution

2. Second applying a perturbation argument in the case of a general $\mathcal{C}^s$ interface which locally deviates from a line in a quantitatively controlled manner.

4.A.1 The case of a linear interface

We assume here that, over the 3×3 stencil S centered at T, the interface Γ is a line crossing T. Therefore, the restriction of u to S is of the form

$$u_{|S} = v_{r,\theta} := \chi_{\{\langle z - z_T, e_\theta^\perp \rangle \leqslant r\}}.$$

where z_T is the center of T and $e_\theta = (-\sin(\theta), \cos(\theta))$ with θ the angle between Γ and the horizontal line, that is, e_θ is the unit normal vector to Γ pointing to the outward direction where $u_{|S} = 0$. The following result shows that the orientation test discriminates exactly if the direction of Γ is closer to horizontal or vertical. Its proof uses elementary geometrical arguments, which are only sketched using pictures in order to avoid cumbersome analytic developments.

Theorem 4.A.1. *If $G_T = (H_T, V_T)$ is the numerical gradient based on the Sobel filter for the above function $v_{r,\theta}$, then the following holds:*

- $|V_T| > |H_T|$ *if and only if* $\theta \in [0, \pi/4[\cup]3\pi/4, 5\pi/4[\cup]7\pi/4, 2\pi[$
- $|H_T| > |V_T|$ *if and only if* $\theta \in]\pi/4, 3\pi/4[\cup]5\pi/4, 7\pi/4[$
- $|H_T| = |V_T|$ *if and only if* $\theta \in \{\pi/4, 3\pi/4, 5\pi/4, 7\pi/4\}$

In addition

- $V_T > 0$ *if and only if* $\theta \in]\pi/2, 3\pi/2[$ *and* $V_T < 0$ *if and only if* $\theta \in [0, \pi/2[\cup]3\pi/2, 2\pi[$
- $H_T > 0$ *if and only if* $\theta \in]0, \pi[$ *and* $H_T < 0$ *if and only if* $\theta \in]\pi, 2\pi[$.

Proof: Without loss of generality, we only consider the case where $\theta \in [0, \pi/4]$ since all other cases $[k\pi/4, (k+1)\pi/4]$ for $k = 1, \ldots, 7$ are treated in a similar way. In order to understand the effect of θ on the values of H_T and V_T, we parametrize the function $v_{r,\theta}$ differently: we fix $\overline{z}_T$ to be the point crossed by Γ on the descending diagonal of T (which exists and is unique when $\theta \in [0, \pi/4]$) and study H_T and V_T for the function

$$v_\theta := \chi_{\{\langle z - \overline{z}_T, e_\theta^\perp \rangle \leqslant 0\}},$$

as we let θ vary. By scale invariance, we may assume that we work with cells of side-length equal to 1 without affecting H_T and V_T.

Figure 4.16 (left) pictures the value of V_T as the difference between areas of the portions of cells from the upper and lower rows crossed by the half-plane below Γ with weight 2 for central cells and 1 for left and right cells. This difference is strictly negative for all $\theta \in [0, \pi/4]$. Its value at $\theta = 0$ is equal -4. As θ grows towards $\pi/4$ it first stays equal to -4 until it starts strictly increasing for some value $\theta^* \in [0, \pi/4[$ that depends on the

position of $\overline{z}_T$ on the diagonal. This monotonic growth can be checked by observing that for $0 \leqslant \theta_1 < \theta_2 \leqslant \pi/4$, one has $V_T(v_{\theta_2}) - V_T(v_{\theta_1}) = V_T(v_{\theta_2} - v_{\theta_1})$ and the function $v_{\theta_2} - v_{\theta_1}$ is supported in a symmetric cone K_{θ_1,θ_2} centered at $\overline{z}_T$ and has value 1 on the right and -1 on the left. Thus $V_T(v_{\theta_2}) - V_T(v_{\theta_1})$ is the sum of the areas of the portions of cells from the upper and lower rows intersected by K_{θ_1,θ_2} with weight 2 for central cells and 1 for for left and right cells, which is strictly positive if $\theta_2 > \theta^*$.

Figure 4.16 (center) pictures the value of H_T as the difference between areas of the portions of cells from the right and left columns crossed by the half-plane below Γ with weight 2 for central cells and 1 for lower and upper cells. This difference is null when $\theta = 0$ and increases strictly as θ grows from 0 to $\pi/4$. Once again, the strictly monotonic growth is due to the fact that $V_T(v_{\theta_2}) - V_T(v_{\theta_1})$ is the sum of the areas of the portions of cells from the left and right columns crossed by K_{θ_1,θ_2} with weight 2 for central cells and 1 for for left and right cells, which is strictly positive whenever $0 \leqslant \theta_1 < \theta_2 \leqslant \pi/4$.

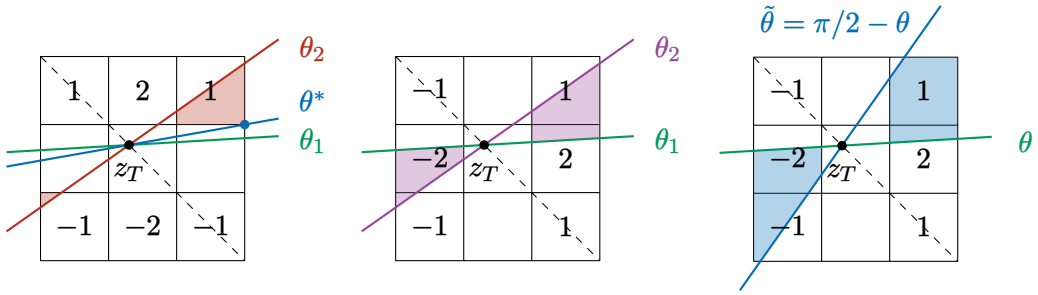


Figure 4.16: Dependence of V_T (left), H_T (center), $|V_T| - |H_T|$ (right) as θ varie in $[0, \pi/4]$.

This demonstrates the second statement regarding signs of V_T and H_T.

On the other hand, by symmetry, we note that $V_T(v_\theta) = -H_T(v_{\tilde{\theta}})$ where $\tilde{\theta} = \pi/2 - \theta$ with same base point z_T. Therefore $|V_T| - |H_T| = -V_T - H_T = H_T(v_{\tilde{\theta}} - v_\theta)$ and $v_{\tilde{\theta}} - v_\theta$ is supported in a symmetric cone $K_{\theta,\tilde{\theta}}$ centered at $\overline{z}_T$ with value 1 on the right and -1 on the left. As pictured on Figure 4.16 (right), this quantity is the sum of the areas of the portions of cells from the right and left column crossed by $K_{\theta,\tilde{\theta}}$ with weight 2 for central cells and 1 for lower and upper cells. These quantities are null when $\theta = \pi/4$ since the cone is then restricted to a line, and increases strictly as θ decreases from $\theta/4$ to 0 since the cone is opening.

This demonstrates the first statement regarding comparison between $|V_T|$ and $|H_T|$. □.

We will use the following direct consequence of this result, which is obtained by compactness since V_T and H_T are continuous with respect to θ and $\overline{z}_T$: for any $0 < \delta < \pi/4$, there exists a $\gamma = \gamma(\delta) > 0$ such that

$$\theta \in [\pi/4 + \delta, 3\pi/4 - \delta] \cup [5\pi/4 + \delta, 7\pi/4 - \delta] \implies |H_T| \geqslant |V_T| + \gamma \tag{4.28}$$

and

$$\theta \in [3\pi/4 - \delta, 5\pi/4 + \delta] \implies V_T \geqslant \gamma, \tag{4.29}$$

4.A.2 A perturbation analysis

We next turn to the proof of Theorem 4.4.1, focusing without loss of generality on case 1. Let us assume that the interface $\Gamma = \partial\Omega$ has $\mathcal{C}^s$ smoothness for some $s > 1$, and let T be a singular cell crossed by Γ and S the 3×3 stencil centered at T.

We now fix $\tilde{z}_T$ to be one point of $\Gamma \cap T$ and $e_\theta^\perp = (-\sin(\theta), \cos(\theta))$ be the outer unit normal to Ω at $\tilde{z}_T$. The function

$$v_\theta := \chi_{\{\langle z - \tilde{z}_T, e_\theta^\perp \rangle \leqslant 0\}},$$

is a perturbation of u as pictured on Figure 4.17 where the line interface L is the tangent to Ω at $\tilde{z}_T$. Since Ω has $\mathcal{C}^s$ smoothness, the deviation between the curved interface Γ and its tangent L has area of order $\mathcal{O}(h^{s+1})$ over S, and therefore

$$|a_{\tilde{T}}(u) - a_{\tilde{T}}(v_\theta)| \leqslant C h^{s-1}, \quad \tilde{T} \in S,$$

for some fixed constant C that only depends on the $\mathcal{C}^s$ norm of the graph that locally characterizes Γ. In turn, up to multiplying the constant C by 8, one has

$$|H_T(u) - H_T(v_\theta)| \leqslant C h^{s-1} \quad \text{and} \quad |V_T(u) - V_T(v_\theta)| \leqslant C h^{s-1} \tag{4.30}$$

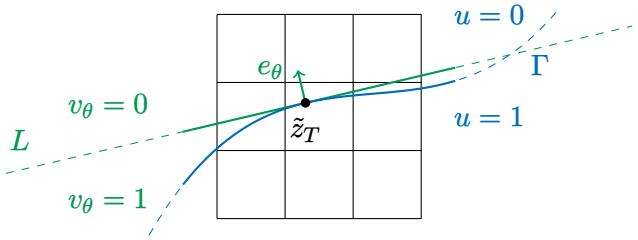


Figure 4.17: Approximation of a smooth interface by its tangent

We now take any $0 < \delta < \pi/4$ arbitrarily small and consider the quantity $\gamma = \gamma(\delta)$ such that (4.28) and (4.29) are valid. For $h \leqslant h_0$ small enough, we are ensured that

$$C h^{s-1} \leqslant \frac{\gamma}{3},$$

where C is the constant in (4.30). Therefore, if $\theta \notin [0, \pi/4+\delta] \cup [7\pi/4-\delta, 2\pi[$, or equivalently $\theta \notin [-\pi/4 - \delta, \pi/4 + \delta]$, we obtain either by (4.28) that

$$|H_T(u)| \geqslant |H_T(v_\theta)| - \frac{\gamma}{3} \geqslant |V_T(v_\theta| + \frac{2\gamma}{3} \geqslant |V_T(u)| + \frac{\gamma}{3} > |V_T(u)|,$$

or by (4.29) that

$$V_T(u) \geqslant V_T(v_\theta) - \frac{\gamma}{3} \geqslant \frac{2\gamma}{3} > 0.$$

Therefore, we conclude for case 1 that if $|H_T(u)| \leqslant |V_T(u)|$ and $V_T(u) \leqslant 0$, the angle θ of the tangent line L necessarily lies in $[-\pi/4 - \delta, \pi/4 + \delta]$. In other words, the points $z = (x, y)$ in the half plane $\{\langle z - \tilde{z}_T, e_\theta^\perp \rangle \leqslant 0\}$ can be characterized by an equation of the form

$$y \leqslant a(x),$$

where a is affine and $|a'(x)| \leqslant 1 + \mathrm{e}$ such that $1 + \mathrm{e} = \tan(\pi/4 + \delta)$, with $0 \leqslant \mathrm{e} \leqslant \delta$ and $\mathrm{e} \sim \delta$ when δ is small. On the other hand the interface Γ is described on T by an equation of the form

$$y \leqslant \psi(x),$$

where due to the $\mathcal{C}^s$ smoothness of Γ, one has an estimate of the form

$$|\psi'(x) - a'(x)| \leqslant Ch^{s-1},$$

and the same will hold on a stencil S_T of width $2k + 1$ up to enlarging the value of C, so that

$$|\psi'(x)| \leqslant 1 + \mathrm{e} + Ch^{s-1}, \quad x \in I,$$

where I is the horizontal support of S_T. Therefore, we can find $h^* = h^*(\Omega)$ such that if $h \leqslant h^*$ we are thus ensured that

$$|\psi'(x)| \leqslant \frac{k+2}{k+1}, \quad x \in I.$$

This implies that the graph of ψ remains confined in S_T if we choose it to be of size $(2k + 1) \times (2l + 1)$ with $l = k + 2$, which concludes the proof of the theorem.

Chapter 5

Nonlinear compressive reduced basis approximation for PDE's

5.1 Introduction

The approximation of the solution of a parameterized partial differential equation (PDE) : given μ, find u solution to

$$\mathcal{D}(u;\mu) = 0$$

can benefit from the *a priori* analysis of the set of all generated solutions when the parameter μ is varied, that is,

$$\mathcal{K} := \{u_\mu \ : \ \mu \in \mathcal{P}\},$$

where u_μ is the solution for the given value $\mu = (\mu_1, \ldots, \mu_d)$ of the parameter vector that ranges in some set $\mathcal{P} \subset \mathbb{R}^d$. The set $\mathcal{K}$ is also referred to as the *solution manifold*, since it may be thought of as a parameterized d-dimensional manifold typically immersed in a Hilbert space X, where the solution to the PDE is well defined. In what follows, the norm in X and the scalar product are respectively denoted by $\|.\|_X$ and $\langle .,. \rangle_X$.

Assuming $\mathcal{K}$ to be compact in X, its *Kolmogorov m-width* defined as

$$d_m(\mathcal{K})_X = \inf_{\dim(X_m)\leqslant m} \max_{v\in\mathcal{K}} \min_{w\in X_m} \|v - w\|_X, \tag{5.1}$$

describes how well the set can be approximated by an ideally selected (and usually out of reach) m-dimensional space. If d_m has a certain rate of decay as $m \to \infty$, it is possible to practically construct low-dimensional spaces X_m that perform with the same approximation rate, by pre-computing *offline* a *reduced basis* consisting of m solutions associated with a well-chosen set of parameters. If d_m has a fast rate of decay, the RB method yields an approximation of the solution for any parameter based on an algebraic system involving very few unknowns. We refer to [133], [92] or [107] for a presentation of RB methods.

If the parameters μ_i are considered as random variables and thus u_μ is an X-valued random variable, a stochastic counterpart to these concepts is described by the principal component analysis in the Hilbert space X, that is, the spectral analysis of the covariance

operator

$$v \mapsto \mathbb{E}(\langle v, u_\mu \rangle_X u_\mu),$$

once u_μ has been recentered so that $\mathbb{E}(u_\mu) = 0$. Denoting by $\sigma_1 \geqslant \sigma_2 \geqslant \cdots$ the sequence of positive eigenvalues of this compact self-adjoint operator, and by $e_1, e_2, \ldots$ the Karhunen-Loève orthonormal basis of eigenfunctions, it is well known that

$$\kappa_m^2 := \min_{\dim(X_m) \leqslant m} \mathbb{E}(\min_{w \in X_m} \|u_\mu - w\|_X^2) = \sum_{n>m} \sigma_n,$$

and that the minimizing space is spanned by $e_1, \ldots, e_m$. This is the starting point to the *Proper Orthogonal Decomposition* (POD) method which amounts to replacing the aforementioned eigenfunctions by approximations computed offline, based on a sufficiently large set of training solutions.

These *linear reduced modelling* methods have penetrated industrial applications, a guarantee of their success. However, there are still cases where these approaches have difficulties to overcome, namely, when the Kolmogorov width d_m or eigenvalues σ_m do not decay fast.

This is in particular what happens for transport type problems. Even in the conceptually simple case of constant speed translation of an initial condition given by a step function, where the only parameter is the position of the discontinuity, it is well known that with $X = L^2$, the numbers d_m and κ_m decay slowly like $\mathcal{O}(m^{-1/2})$. In other words, for a target precision of ε, the basis is of prohibitive size $\mathcal{O}(\varepsilon^{-2})$.

For families of such functions, substantial gain can be expected when searching for *nonlinear reduced models*. Prominent examples of nonlinear approximation include rational fractions, finite elements on adaptive grids of fixed cardinality, n-term approximations in a basis or dictionary, and neural networks, see [63, 64] for a general treatment. In these methods, the "coordinates" describing the approximation to a function u are typically nonlinear functionals applied to u, and the reconstruction map from such parameters is also nonlinear. In the frame of model reduction, we refer to [7, 82, 33] that considers libraries of affine reduced models, [16] that uses quadratic manifolds, and [104, 76, 83, 121, 125, 17] for neural network based approaches, see also [24, 40, 30, 86], and [124] for an overview on these nonlinear approaches.

Interestingly, it appears that an efficient approach to nonlinear model reduction is to maintain linear functionals for computing the coordinates while performing reconstruction in a well-chosen nonlinear way. This state of affair is in particular illustrated by the development of *compressed sensing* in the last two decades, where signals are reconstructed from linear measurements by nonlinear methods promoting sparsity, such as ℓ^1 minimization.

In this note, we begin by substantiating this idea more precisely in §2, by recalling and comparing certain notions of linear and nonlinear m-widths. We present in §3 a general approach that consists in taking as linear functionals the first components in a linear reduced model (RB or POD) that has been learned offline; and also use the offline stage to learn a computationally tractable nonlinear map that reconstructs the missing components from these first ones to reach a better accuracy. One key aspect lies in the type of nonlinear maps that is allowed. This approach is analyzed in §4, in the case of a simple univariate

model of step functions; it is illustrated by numerical tests for this model in §5.

5.2 Linear and nonlinear notions of m-widths

Generally speaking, the process of dimensionality reduction can be described by a pair of continuous mappings, the encoder

$$E : X \to \mathbb{R}^m,$$

and the decoder

$$D : \mathbb{R}^m \to X.$$

The maximum distorsion of the encoding procedure over $\mathcal{K}$ is given by the quantity

$$\max_{v \in \mathcal{K}} \|v - D(E(v))\|_X.$$

Then, for a general Banach space X and a compact set $\mathcal{K} \subset X$, we can define various notion of widths

$$\inf_{D,E} \max_{v \in \mathcal{K}} \|v - D(E(v))\|_X,$$

by optimizing the choice of E and D, under specific restrictions:

- If D and E are both assumed to be linear, one obtains the *approximation numbers*

  $$a_m(\mathcal{K})_X := \inf_{L} \max_{v \in \mathcal{K}} \|v - Lv\|_X,$$

  where the infimum is taken over operators L of rank at most m.

- If only D is assumed to be linear, one obtains the already mentionned Kolmogorov width $d_m(\mathcal{K})_X$. When X is a general Banach space, the inequality

  $$d_m(\mathcal{K})_X \leqslant a_m(\mathcal{K})_X,$$

  can be strict. Equality obviously holds in the case when X is a Hilbert space since best approximation in a subspace of X of dimension m is achieved by linear orthogonal projection.

- The *sensing numbers* $s_m(\mathcal{K})_X$ correspond to the reciprocal situation, where E is assumed to be linear and D is assumed to be nonlinear. In other words, they can be defined as

  $$s_m(\mathcal{K})_X := \inf_{D,\lambda_1,\ldots,\lambda_m} \max_{v \in \mathcal{K}} \|v - D(\lambda_1(v), \ldots, \lambda_m(v))\|_X,$$

  where the infimum is taken over all choice of linear functionals $\lambda_1, \ldots, \lambda_m \in X'$ and decoding map D. These number are closely related to the *Gelfand width* classically

defined as

$$d^m(\mathcal{K})_X := \inf_{\lambda_1,\dots,\lambda_m} \max\{\|v\|_X \ : \ v \in \mathcal{K},\ \lambda_1(v) = \cdots = \lambda_m(v) = 0\}.$$

It is easily checked that $s_m(\mathcal{K})_X = d^m(\mathcal{K})_X$ in the case where $\mathcal{K}$ is convex and centrally symmetric; and that

$$s_m(\mathcal{K})_X \leqslant d^m(\mathcal{K}-\mathcal{K})_X \leqslant 2s_m(\mathcal{K})_X,$$

for a general compact set $\mathcal{K}$ and $\mathcal{K}-\mathcal{K}$ is a notation for the set $\{u : u = v - w, v \in \mathcal{K}, w \in \mathcal{K}\}$.

- Finally, the *nonlinear width* or *manifold width* $\delta_m(\mathcal{K})_X$ is defined when no other assumption but continuity is made on the operators E and D. For numerical stability purpose, it is interesting to tame this notion by imposing that D and E are both Lipschitz continuous, that is

$$\|D(a)-D(b)\|_X \leqslant \gamma\|a-b\|_m \quad \text{and} \quad \|E(v)-E(w)\|_m \leqslant \gamma\|v-w\|_X, \quad a,b \in \mathbb{R}^m,\ v,w \in X,$$

for some fixed $\gamma > 1$, with $\|\cdot\|_m$ an arbitrary norm on $\mathbb{R}^m$. The resulting infimized quantity $\delta_m^\gamma(\mathcal{K})_X$ is referred to as the *stable width.*

The last two notions of width s_m and δ_m (or δ_m^γ) are natural to describe the expected performance of optimal *nonlinear model reduction*, since the manifold is approximated by the set $D(\mathbb{R}^m)$ – which is no longer a linear space. However, the sensing numbers take the view that encoding can be restricted to simple linear measurements.

As already mentionned, the quantities d_m and a_m typically decay slowly for families of piecewise smooth functions, which reflects the fact that they cannot be well approximated efficiently by linear spaces. A substantial gain in the rate of decay can be expected however when considering the nonlinear widths δ_m and δ_m^γ. Interestingly, it appears that this substantial optimal gain is already present when considering the sensing numbers s_m.

As a basic example, consider the two-parameter family of univariate step functions

$$\mathcal{K} := \{u := \chi_{[a,a+\ell]} \ : \ a \in \mathbb{R},\ \ell > 0\}.$$

Clearly, the parameters (a,ℓ) are not linear functionals of u. However, any $u \in \mathcal{K}$ can be exactly reconstructed from two linear functionals, namely, the first moments

$$\lambda_k(u) = \int x^k u(x)dx, \quad k = 0,1.$$

Indeed, $\lambda_0 = \ell$ and $\lambda_1 = \frac{1}{2}\ell(2a+\ell)$, so that a and ℓ can be exactly recovered from such data. Therefore, one has $s_m(\mathcal{K})_X = 0$ for any $m > 2$ and for any Banach space X.

At a more general level, it was proved in [53] that when X is a Hilbert space, then both $s_m(\mathcal{K})_X$ and $\delta_m^\gamma(\mathcal{K})_X$ are tied to the so-called *entropy numbers* $\mathrm{e}_m(\mathcal{K})_X$ defined as the smallest value of $\mathrm{e} > 0$ such that $\mathcal{K}$ can be covered by 2^m balls of radius e. More precisely,

it was shown that, on the one hand, for any $s > 0$, one has a Carl-type inequality

$$\sup_{m>0} m^s \mathrm{e}_m(\mathcal{K})_X \leqslant C_s \sup_{m>0} m^s \delta_m^\gamma(\mathcal{K})_X,$$

where C_s depends on (s,γ), and that on the other hand, there exists $c > 0$ depending on $\gamma > 1$ such that

$$\delta_{cm}^\gamma(\mathcal{K})_X \leqslant 3\mathrm{e}_m(\mathcal{K})_X, \quad m \geqslant 1.$$

In the proof of this last inequality, the γ-stable encoding-decoding pair (E,D) which is constructed has actually a linear E. In turn, one also has

$$s_{cm}(\mathcal{K})_X \leqslant 3\mathrm{e}_m(\mathcal{K})_X, \quad m \geqslant 1.$$

One consequence of these results is that $s_m(\mathcal{K})_X$, $\delta_m^\gamma(\mathcal{K})_X$ and $\mathrm{e}_m(\mathcal{K})_X$ share the same algebraic rates of decay.

Remark 5.2.1. *An additional aspect of nonlinear dimensionality reduction is the notion of adaptivity, which means that the measurements $E(u) = (E_1(u), \ldots, E_m(u))$ are chosen incrementally, that is, the functional E_m is picked depending on the value of $E_1(u), \ldots, E_{m-1}(u)$. This allows the definition of similar notions of adaptive sensing numbers and nonlinear widths. Our next described approach is not of this form, since we use linear functionals that are pre-defined through the standard POD or RB analysis.*

5.3 Nonlinear compressive Reduced Basis approximation

In this contribution, we thus intend to deal with situations where:

- The Kolmogorov widths $d_m(\mathcal{K})_X$, or the singular values σ_n, decay slowly.
- The sensing numbers $s_m(\mathcal{K})_X$, and stable nonlinear widths $\delta_m^\gamma(\mathcal{K})_X$, decay much faster.

In other words, a target accuracy $\mathrm{e} > 0$ can be reached by $d_N(\mathcal{K})_X$ or κ_N, however with a dimension $N = N(\mathrm{e})$ much larger than the value of $n = n(\mathrm{e})$, such that $s_n(\mathcal{K})_X$ reaches the same accuracy.

Since the optimal linear functionals in the definition of $s_n(\mathcal{K})_X$ are usually out of reach and could be computationally unpractical to apply, **we take the view of fixing these measurements to be a small number n of components in the offline computed (orthonormalized) RB or POD basis** $(e_j)_{j=1,\ldots,N}$ for some $N >> n$. Typically, we choose the n first ones, that is,

$$\lambda_j(v) = \langle v, e_j \rangle_X, \quad j = 1, \ldots, n.$$

Intuitively, it is expected that in the situation where $s_n(\mathcal{K})_X$ is very small, then the unknown component $(\lambda_j(v))_{j=n+1,\ldots,N}$ should be somehow dependent, up to a small error, of the n first ones that carry most of the relevant information. This idea was at first presented in

[17]. Here, we formalize it and study its validity in detail on a simple step function model, and propose a general numerical strategy that we test on this model.

Our objective is thus to predict from these first components the extra components $\lambda_k(v)$ for $k = n+1, \dots, N$ that are needed to approximate the functions $v \in \mathcal{K}$ with target accuracy e. We are thus interested to construct $N - n$ functions $\psi_k : \mathbb{R}^n \to \mathbb{R}$ so that

$$\tilde{\lambda}_k(v) := \psi_k(\lambda_1(v), \dots, \lambda_n(v)),$$

is a very accurate approximation to $\lambda_k(v)$ for $k = n+1, \dots, N$ and can be fastly computed.

Let us stress that ψ_k should typically be a nonlinear function. Indeed consider the ideal case of the PCA basis computed after having recentered the variable u_μ. Then the variables

$$z_j = \lambda_j(u_\mu),$$

are uncorrelated and centered, such that, for any $k > n$,

$$\min_{\alpha_1, \dots, \alpha_n} \mathbb{E}(|z_k - \sum_{j=1}^{n} \alpha_j z_j|^2) = \mathbb{E}(|z_k|^2).$$

Thus, the best choice of a linear function would be the null one that does not deliver any information.

On the other hand, the best choice of a nonlinear function in this mean square sense, that is, minimizing $\mathbb{E}(|z_k - \psi(z_1, \dots, z_n)|^2)$ over all functions ψ, is given by the conditional expectation

$$\psi_k^*(z_1, \dots, z_n) = \mathbb{E}(z_k | z_1, \dots, z_n),$$

which is out of reach and should be approximated by a computationally tractable function.

Our practical approach to the construction of ψ_k is by *learning* it in a second step of the offline stage, after the basis $(e_j)_{j=1,\dots,N}$ has been identified. Having in mind the above mean square loss, one typical approach is to select ψ_k within a sufficiently rich class $\mathcal{F}$ of nonlinear functions by empirical risk minimization : with $(u^i)_{i=1,\dots,M}$ a training set of random snapshots $u^i = u_{\mu^i}$, we define

$$\psi_k := \operatorname{argmin}\Big\{ \sum_{i=1}^{m} |\lambda_k(u^i) - \psi(\lambda_1(u^i), \dots, \lambda_n(u^i))|^2 \ : \ \psi \in \mathcal{F} \Big\}.$$

A critical aspect in this approach lies in the choice of the class $\mathcal{F}$, which could be, for example, the set of:

- Quadratic functions, as in [16] or [79].
- Polynomials of some higher degree $d > 2$.
- Neural networks with a given architecture, as proposed in [17] (see also [104] where an autoencoder-based approximation was proposed, which was in a way a pioneering idea but unfortunately not computationally tractable one).

This class should be able to approximate correctly the ideal but out of reach ψ_k^* by a computationally tractable function $\psi_k \in \mathcal{F}$. Another difficulty with this approach is the fact that when the number n of informative components is chosen to be not very small, one faces a regressing problem in large dimension, for which classical methods such as splines or polynomials are known to suffer from the curse of dimensionality.

For these reasons, we have also considered in our numerical tests regression methods based on trees (CART) and random forests, that are both universally consistent and able to tackle large-dimensional problems. These methods seem to deliver the best numerical results for the considered problems.

5.4 Analysis of a model framework : periodic step functions

In order to investigate the aforementioned questions, we place ourselves in a framework where the Karhunen-Loève basis is explicitely known. Specifically, we work in the Hilbert space

$$X = L^2(0,1),$$

and consider a randomly parameterized family such that

$$\mathbb{E}(u_\mu(x)) = \bar{u},$$

independently of $x \in [0,1]$ and such that

$$\mathbb{E}((u_\mu(x) - \bar{u})(u_\mu(y) - \bar{u})) = R(x-y),$$

where R is an even and 1-periodic function. In other words, u_μ is a periodic stationary process, its covariance operator coincides with the convolution operator by R, and therefore its Karhunen-Loève basis is exactly given by the basis of the Fourier series on $[0,1]$ (see e.g. [122]).

More specifically, we consider a simple model of periodic stationary step functions by introducing the three-parameter family

$$u_\mu(x) := \begin{cases} b & \text{for } x \in (a, a+\ell) \pmod 1 \\ 0 & \text{for } x \in (a+\ell, a) \pmod 1 \end{cases}, \qquad \mu = (a, \ell, b), \tag{5.2}$$

that is, $u_\mu = b\chi_{[a,a+\ell]}$ in a 1-periodic sense.

Here a, ℓ, and b are assumed to have independent uniform distributions. Taking the base point a to be uniformly distributed over $[0,1]$, it is easily checked that the process is periodic stationnary. In addition, we take the height b to be uniformly distributed in $[0,1]$ and the length ℓ to be uniformly distributed in $[\ell_{\min}, 1-\ell_{\min}]$ for some $0 < \ell_{\min} < \frac{1}{2}$.

The best linear approximation of dimension $m = 2n+1$ is thus given by the truncation up to $k \leqslant n$ of the Fourier expansion

$$u_\mu = \sum_{k \in \mathbb{N}} \alpha_k \cos(2\pi k x) + \sum_{k \in \mathbb{N}^*} \beta_k \sin(2\pi k x) \tag{5.3}$$

where

$$\left\{\begin{array}{ll} \alpha_0 = \alpha_0(a,\ell,b) & = b\ell \\ \alpha_k = \alpha_k(a,\ell,b) & = b\frac{\sin(2\pi k(a+\ell))-\sin(2\pi ka)}{2\pi k} = b\frac{\sin(\pi k\ell))\cos(\pi k(2a+\ell))}{\pi k} \\ \beta_k = \beta_k(a,\ell,b) & = b\frac{\cos(2\pi k(a+\ell))-\cos(2\pi ka)}{2\pi k} = -b\frac{\sin(\pi k\ell))\sin(\pi k(2a+\ell))}{\pi k} \end{array}\right. \tag{5.4}$$

Clearly $\sigma_0 = \mathbb{E}(|\alpha_0|^2) = \mathbb{E}(\ell^2)\mathbb{E}(b^2) = \frac{1}{9}((1-\ell_{\min})^3 - \ell_{\min}^3)$. It is also easily checked that the eigenvalues associated to the functions $x \mapsto \cos(2\pi kx)$ and $x \mapsto \sin(2\pi kx)$ are the same and are given by

$$\sigma_k = \mathbb{E}(|\alpha_k|^2) = \mathbb{E}(|\beta_k|^2) = \frac{c}{k^2},$$

for some $c = c(\ell_{\min}) > 0$. It follows that the best linear approximation has a mean-square error κ_m^2 behaving like m^{-1}. Note that, for the corresponding manifold

$$\mathcal{K} := \{u_\mu \ : \ a \in [0,1],\ \ell \in [\ell_{\min}, 1-\ell_{\min}],\ b \in [0,1]\},$$

one obviously has $d_m(\mathcal{K})_X \geqslant \kappa_m$, since a worst case error dominates the average error. On the other hand, it is also readily seen that the worst case approximation by Fourier series behaves like $m^{-1/2}$, and therefore

$$d_m(\mathcal{K})_X \sim \kappa_m \sim m^{-1/2}.$$

We also consider the two-parameter family $\mathcal{K}_{\ell_0}$ obtained by freezing the value $\ell = \ell_0$ and the one-parameter family $\mathcal{K}_{b_0,\ell_0}$ obtained by freezing in addition the value $b = b_0$. It is easily checked that one has the same behaviour $m^{-1/2}$ for κ_m and d_m after such restrictions.

The one-parameter family $\mathcal{K}_{b_0,\ell_0}$ can be encoded by the data of a, so that its nonlinear width satifies

$$\delta_m(\mathcal{K}_{b_0,\ell_0})_X = 0, \quad m > 1.$$

It is easily seen that the data of only one Fourier coefficient is not sufficient to characterize the elements of this family. Indeed, $\alpha_0(a,\ell,b)$ is independent of a, $\alpha_k(a,\ell,b) = \alpha_k(1/2 - a - \ell, \ell, b)$, and $\beta_k(a,\ell,b) = \beta_k(1/4 - a - \ell, \ell, b)$ for $k \neq 0$.

On the other hand, the recovery of a can be done through the data of the two coefficients α_1 and β_1 since

$$a = -\frac{\ell}{2} - \frac{1}{2\pi}\arctan\left(\frac{\beta_1}{\alpha_1}\right) \pmod 1 \tag{5.5}$$

Similarly, any element in the two-parameter family $\mathcal{K}_{\ell_0}$, parametrized by a and b, can be recovered from the data of these two coefficients, since one also has

$$b = \frac{\pi}{\sin(\pi\ell_0)}(\alpha_1^2 + \beta_1^2)^{1/2}. \tag{5.6}$$

When more coefficients are available, we note that there is not a unique reconstruction map : for example from the three coefficients α_0 , α_1, and β_1, we can also recover b according

to

$$b = \frac{\alpha_0}{\ell_0}.$$

Finally, in the case of the three parameter family $\mathcal{K}$, exact recovery of (a, b, ℓ) can be obtained by solving the nonlinear system

$$\left\{ \begin{array}{ll} a + \frac{\ell}{2} & = -\frac{1}{\pi}\arctan(\frac{\beta_1}{\alpha_1}) \\ \pi b \sin(\pi \ell) & = (\alpha_1^2 + \beta_1^2)^{1/2} \\ b\ell & = \alpha_0 \end{array} \right. \tag{5.7}$$

however the exact recovery map does not anymore have an explicit form.

These exact recovery procedures induce for all $k > 1$ an exact recovery map ψ_k^* such that

$$\alpha_k = \psi_k^*(\alpha_0, \alpha_1, \beta_1).$$

and similarly an exact recovery map $\tilde{\psi}_k^*$ such that

$$\beta_k = \tilde{\psi}_k^*(\alpha_0, \alpha_1, \beta_1),$$

In this very simple case, the success of the learning strategy outlined in the previous section therefore depends on how these maps can be approximated by the family $\mathcal{F}$.

A simple intuition can be given when looking at the particular case of $\tilde{\psi}_k^*$ for the one-parameter family $\mathcal{K}_{b_0,\ell_0}$, when $b_0 = 1$ and $\ell_0 = \frac{1}{2}$. Then, we find that

$$\beta_k = b\frac{\sin(\pi k/2)\cos(2\pi k a)}{\pi k},$$

which is null for even values of k, and for odd values $k = 2j + 1$ satisfies

$$\beta_k = \frac{b}{k\pi}(-1)^j T_k\Big(\frac{\pi}{b}\beta_1\Big),$$

where T_k is the Chebychev polynomial of degree k. Therefore for such values, the optimal reconstruction is exact and given by

$$\tilde{\psi}_k^*(x, y, z) = \frac{b}{k\pi}(-1)^j T_k\Big(\frac{\pi}{b}z\Big).$$

Clearly, the function $\tilde{\psi}_k^*$ cannot be well approximated by polynomials of moderate dimensions for large values of k. On the other hand, it is well known that the derivative of T_k has maximal norm of order k over $[-1, 1]$, and this implies that the functions $\tilde{\psi}_k^*$ are Lipschitz continuous with Lipschitz constant bounded independently of $k > 0$.

This property holds in more generality from the following argument: the derivative of the arctan function being upper bounded by 1, the recovery of a from $\alpha_1(a, \ell_0, b_0)$ and

$\beta_1(a, \ell_0, b_0)$, still in the case of the one-parameter family $\mathcal{K}_{b_0,\ell_0}$, is stable as

$$\frac{da}{d\alpha_1} = \frac{1}{2\pi} \frac{1}{1 + \left[\frac{\beta_1}{\alpha_1}\right]^2} \frac{\beta_1}{\alpha_1^2} = -\frac{1}{2\pi} \frac{\beta_1}{\alpha_1^2 + \beta_1^2} \tag{5.8}$$

and

$$\frac{da}{d\beta_1} = -\frac{1}{2\pi} \frac{1}{1 + \left[\frac{\beta_1}{\alpha_1}\right]^2} \frac{1}{\alpha_1} = -\frac{1}{2\pi} \frac{\alpha_1}{\alpha_1^2 + \beta_1^2} \tag{5.9}$$

are both bounded since, by construction (see (5.6), with $b = b_0$ fixed),

$$[\alpha_1]^2 + [\beta_1]^2 = b_0^2 \frac{\sin^2(\pi \ell_0)}{\pi^2}.$$

Hence, an error in the values of α_1 or β_1 will induce an error of comparable size on a. The same holds for the determination of b in the case of $\mathcal{K}_{\ell_0}$.

On the other hand, it is readily seen from the definition of Fourier coefficients that the two maps

$$\mu \mapsto \alpha_k(u_\mu) \quad \text{and} \quad \mu \mapsto \beta_k(u_\mu),$$

are Lipschitz continuous with Lipschitz constants bounded independently of $k > 0$. In turn, the stable recovery of a and b from α_0, α_1 and β_1, induces recovery maps

$$(\alpha_1(a, \ell_0, b_0), \beta_1(a, \ell_0, b_0)) \longmapsto (\alpha_k(a, \ell_0, b_0), \beta_k(a, \ell_0, b_0)) \tag{5.10}$$

and

$$(\alpha_0(a, \ell_0, b), \alpha_1(a, \ell_0, b), \beta_1(a, \ell_0, b)) \longmapsto (\alpha_k(a, \ell_0, b), \beta_k(a, \ell_0, b)) \tag{5.11}$$

that are Lipschitz continuous with Lipschitz constants bounded independently of $k > 0$.

This state of affairs explains that universally consistent methods such as random forests are well adapted for the joint approximation of ψ_k^* and $\tilde{\psi}_k^*$, while approaches based on low order polynomials are doomed to fail. This is confirmed by the numerical tests presented in the next section.

In the perspective of recovering more general piecewise smooth functions, we expect that the low-order components are affected by the smooth pieces in addition to the jumps, while the high-order components are only affected by the jumps. Thus it is interesting to adress the question of the recovery of the parameters of the step function from a few components α_k and β_k for larger values of k.

This task appears to be more involved and requiring more coefficients. For example, when recovering a as in (5.5), we find

$$a = -\frac{\ell}{2} - \frac{1}{2\pi} \arctan\left[\frac{\beta_k}{\alpha_k}\right] \pmod{1/k}. \tag{5.12}$$

One possiblity to lift the indeterminacy $\pmod{1/k}$ is to combine the information coming

from (5.12) and

$$a = -\frac{\ell}{2} - \frac{1}{2\pi}\arctan\left[\frac{\beta_{k+1}}{\alpha_{k+1}}\right] \pmod{1/(k+1)} \tag{5.13}$$

since $a = a',\ (\mathrm{mod}\ 1/k)$, $a = a',\ (\mathrm{mod}\ 1/(k+1))$, and $a, a' \in (0,1)$ imply $a = a'$.

Thus we can in principle recover the parameters a and b out of 4 coefficients of arbitrary high frequencies $(k, k+1)$. However we also observe that the stability of this recovery is deteriorated since $\frac{da}{d\alpha_k}$ and $\frac{da}{d\beta_k}$ increase linearly with k. We may hope to improve the stability by using a larger number of coefficient values with indexes $(k, k+1, \ldots, k+d)$ for some $d > 1$.

5.5 Numerical illustrations

In this section, we investigate the ability of different methods to learn mappings that use different amount m of components, namely

$$\begin{array}{lll} m = 2, & (\alpha_1, \beta_1) & \longmapsto (\alpha_k, \beta_k) \\ m = 3, & (\alpha_0, \alpha_1, \beta_1) & \longmapsto (\alpha_k, \beta_k) \\ m = 5, & (\alpha_0, \alpha_1, \beta_1, \alpha_2, \beta_2) & \longmapsto (\alpha_k, \beta_k) \end{array}$$

for each of the three families $\mathcal{K}_{b_0,\ell_0}$ (Figures 5.1 and 5.2), $\mathcal{K}_{\ell_0}$ (Figures 5.3 and 5.4), $\mathcal{K}$ (Figures 5.5 and 5.6), for all $2 \leqslant k \leq 500$. In these Figures, the average recovery error for the α_k and β_k are presented in a symmetric manner, on the left and right side of the x-axis respectively.

The learning methods are

- linear regression: $\mathcal{F}$ is the set of linear functions,
- quadratic regression: $\mathcal{F}$ is the set of polynomials of total degree 2,
- quartic regression: $\mathcal{F}$ is the set of polynomials of total degree 4,
- decision tree [90],
- random forest [35] [80].

For all the numerical illustrations we used Python 3.8 and scikit-learn 1.2.2 [123] for the implementation of each of the regression methods described above. For more information, the code can be found at `https://github.com/agussomacal/NonLinearRBA4PDEs`.

As can be expected, linear regression give the same results as the null forecast, and quadratic and quartic regression give the same (bad) result here, with some improvement over the null forecast only for very small value of k in certain cases (see Figures 5.3 and 5.4).

In contrast, decision tree and random forest are well suited as can be seen for the one parameter family on Figure 5.1, with improved results on Figure 5.2 obtained from a larger training samples (10 000 rather than 1 000). This reflects the universal consistency of these methods that are guaranteed to converge towards the regression function as the number of samples tends to $+\infty$.

The same also holds for the two parameter family, as seen on Figures 5.3 and 5.4 : the problem is slightly more involved but nevetheless decison tree and random forest manage to obtain a fair (resp. good) approximation after a learning phase of 1 000 (resp. 10 000) training samples.

The numerical results for the three parameter family are displayed on Figure 5.5 for the range $\ell \in [0.4, 0.6]$, and on Figure 5.6 for the range $\ell \in [0.01, 0.99]$, that is $\ell_{\min} = 0.4$ and 0.01 respectively. One first observation is that all methods fail in the case of $m = 2$ known component since they are insufficient to characterize an element of $\mathcal{K}$. Secondly we observe that the performance deteriorates as ℓ is allowed to be very small in which case the exact recovery becomes less stable in view of the multiplicative coupling between b and ℓ in the last two equations of the system (5.7). These last results reveal difficulties for these too simple nonlinear recovery methods to achieve a satisfactory recovery. We expect that more involved approaches such as deep neural networks can improve this state of affair.

Finally, in the case of the two parameter family (ℓ fixed), we study the recovery error of (α_k, β_k) for $k > 10 + d$ when using information from high frequency coefficients (α_j, β_j) for $j = 10, 11, \ldots, 10 + d$. As explained in the end of the last section, the exact recovery is feasible for $d = 1$, yet less stable and thus more difficult to learn. This is confirmed by Figure 5.7, where we see an improvement when using a larger value of d and a larger training sample.

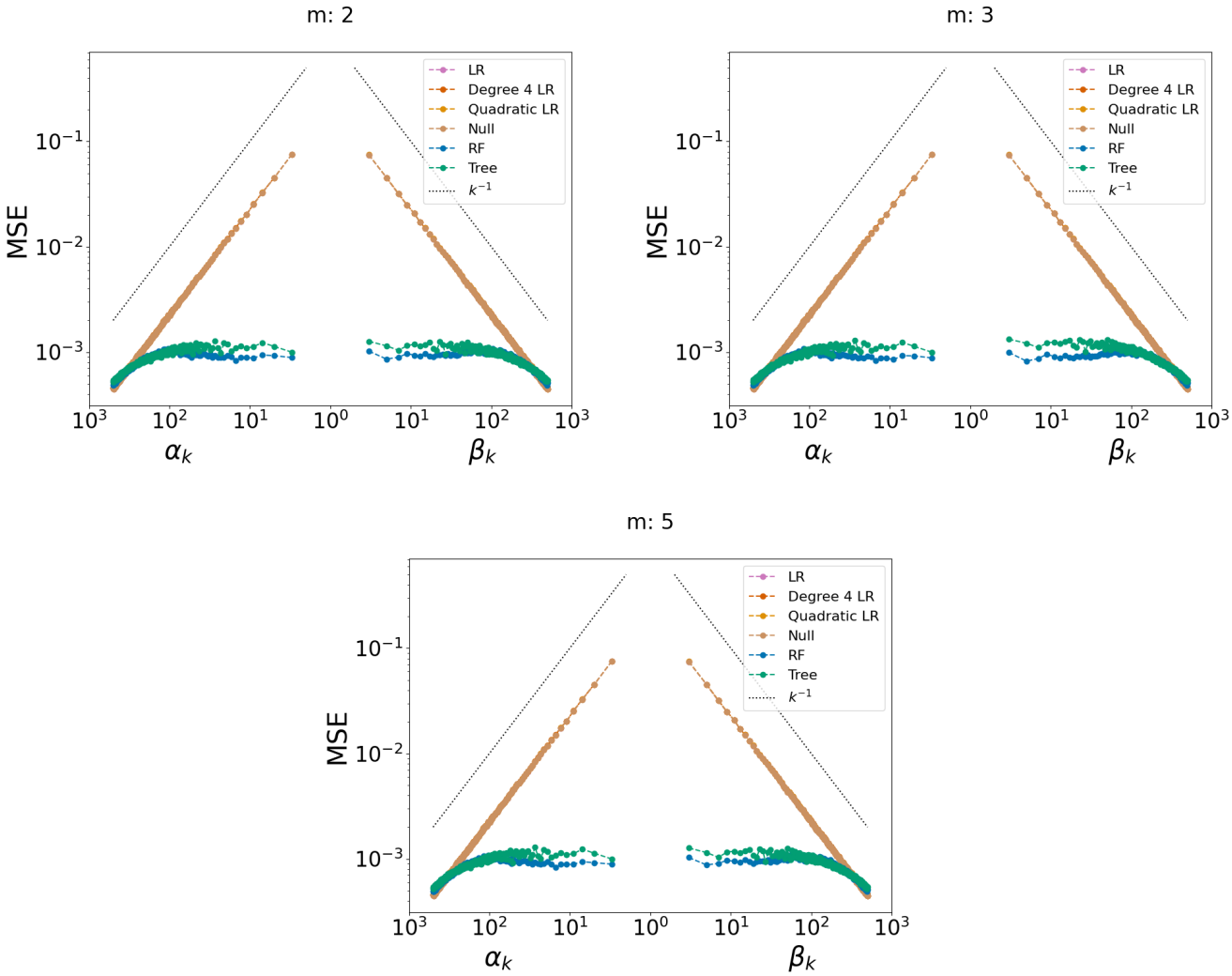


Figure 5.1: In this figure we plot the error obtained from different recovery methods for the family $\mathcal{K}_{b=1,\ell=.5}$ where we recover all coefficients α_k and β_k in (5.3) for $2 \leqslant k \leq 500$ from 2 (left) 3 (center) and 5 (right) Fourier coefficients with different approaches: linear, quadratic, quartic, tree and random forests. Note that linear, quadratic, quartic are superposed and do not improve over the trivial recovery of the missing modes by value 0. The learning phase is based on 1 000 training samples. The x-axis represents (in a log scale) the index k of α_k and the index k of β_k and the y-axis the mean-square reconstruction error on the mode.

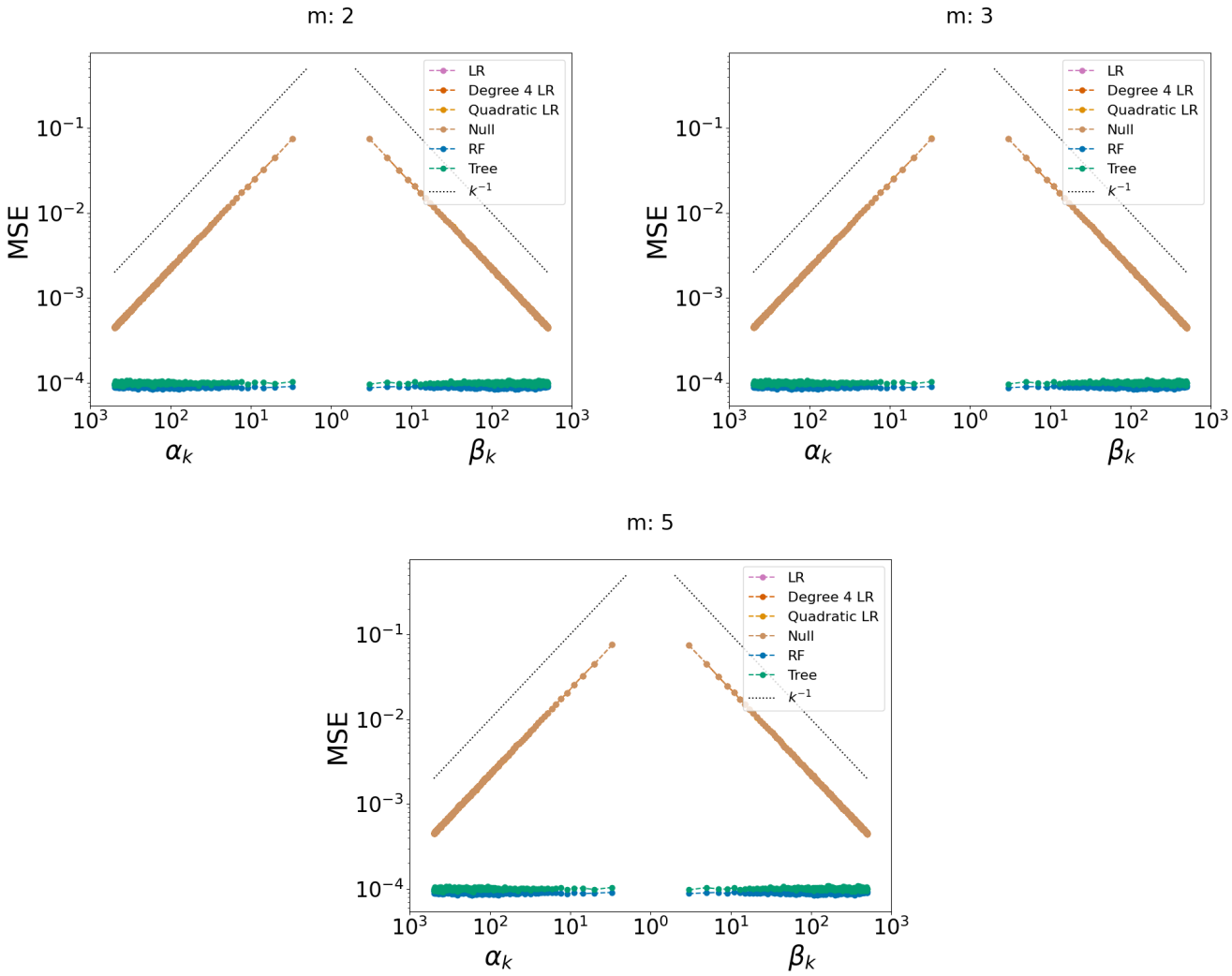


Figure 5.2: Sames test as Figure 1 with 10 000 training samples.

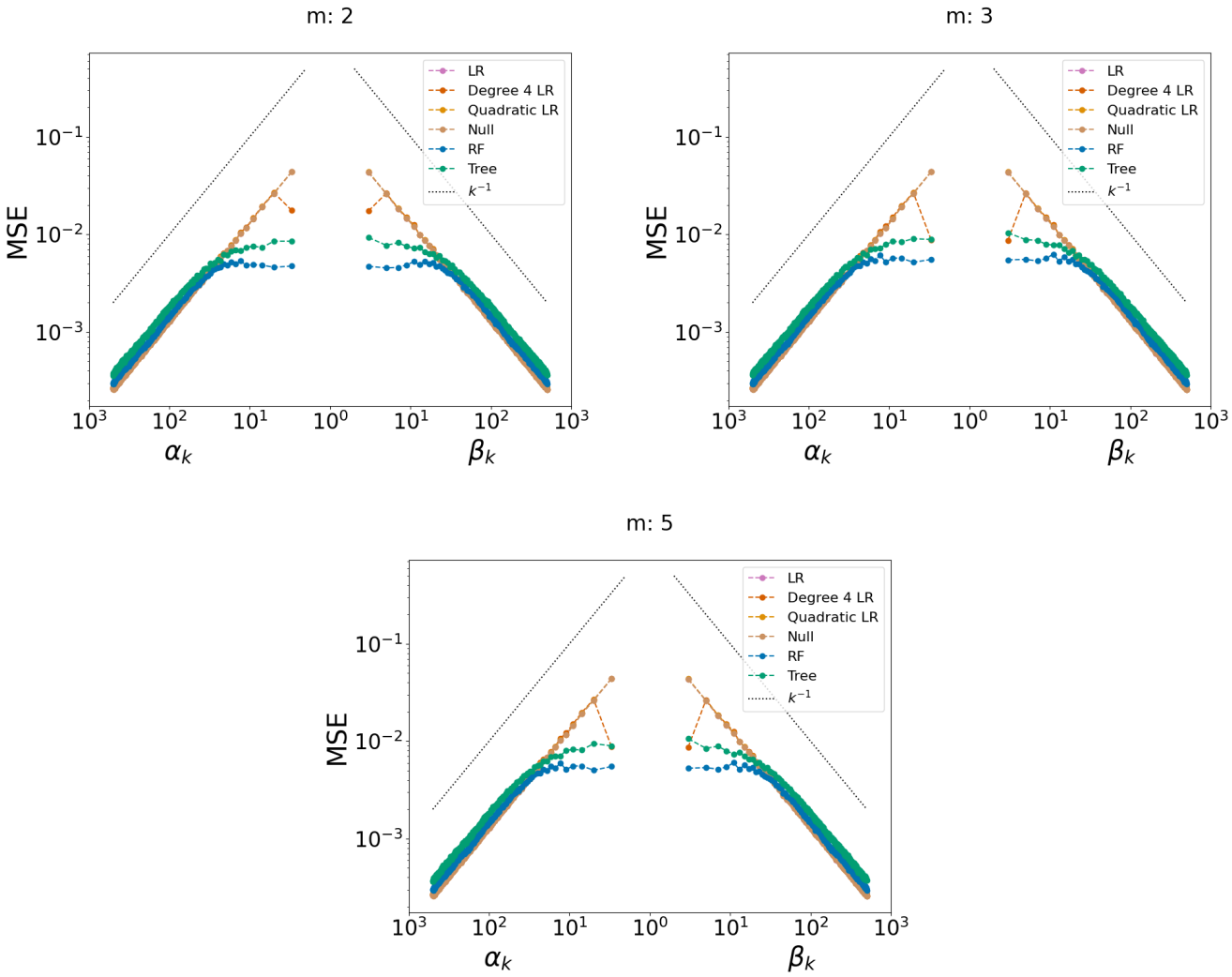


Figure 5.3: Same test as Figure 1 for the two parameter family $\mathcal{K}_{\ell=.5}$, using 1 000 training samples.

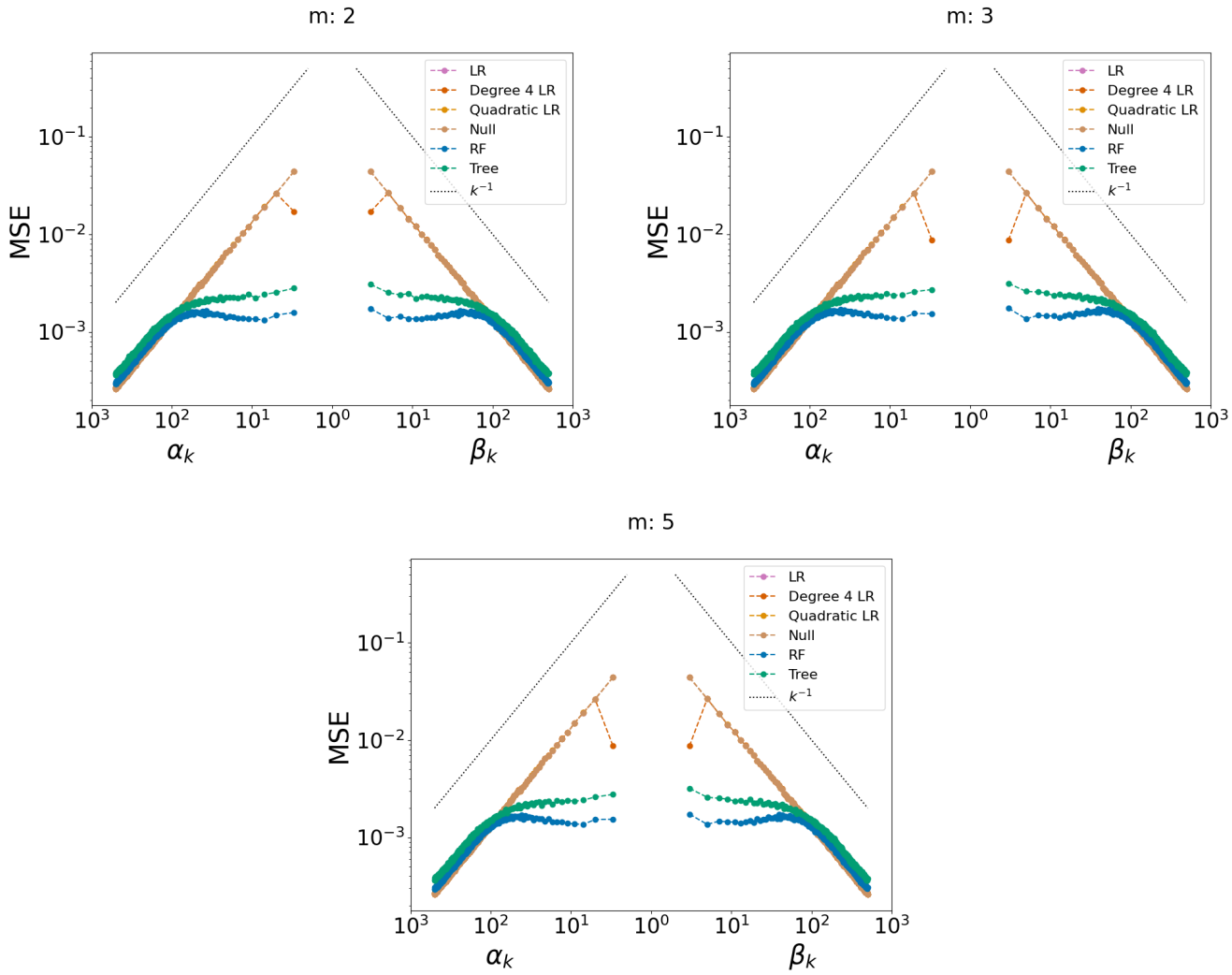


Figure 5.4: Same test as Figure 3 for the two parameter family $\mathcal{K}_{\ell=.5}$, using 10 000 training samples.

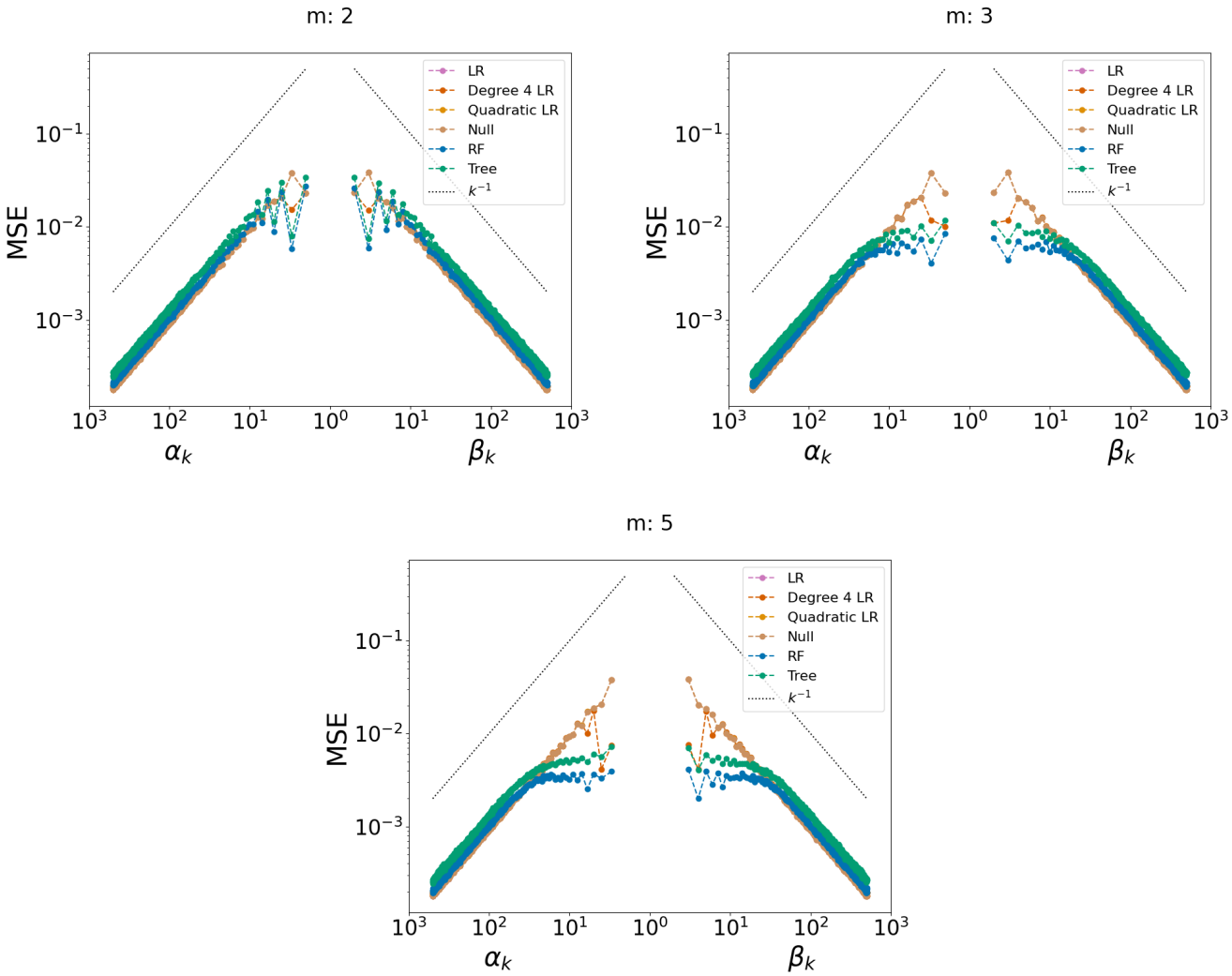


Figure 5.5: Same test as Figure 1 for the three parameter family $\mathcal{K}$, using 10 000 training and $\ell_{\min} = 0.4$.

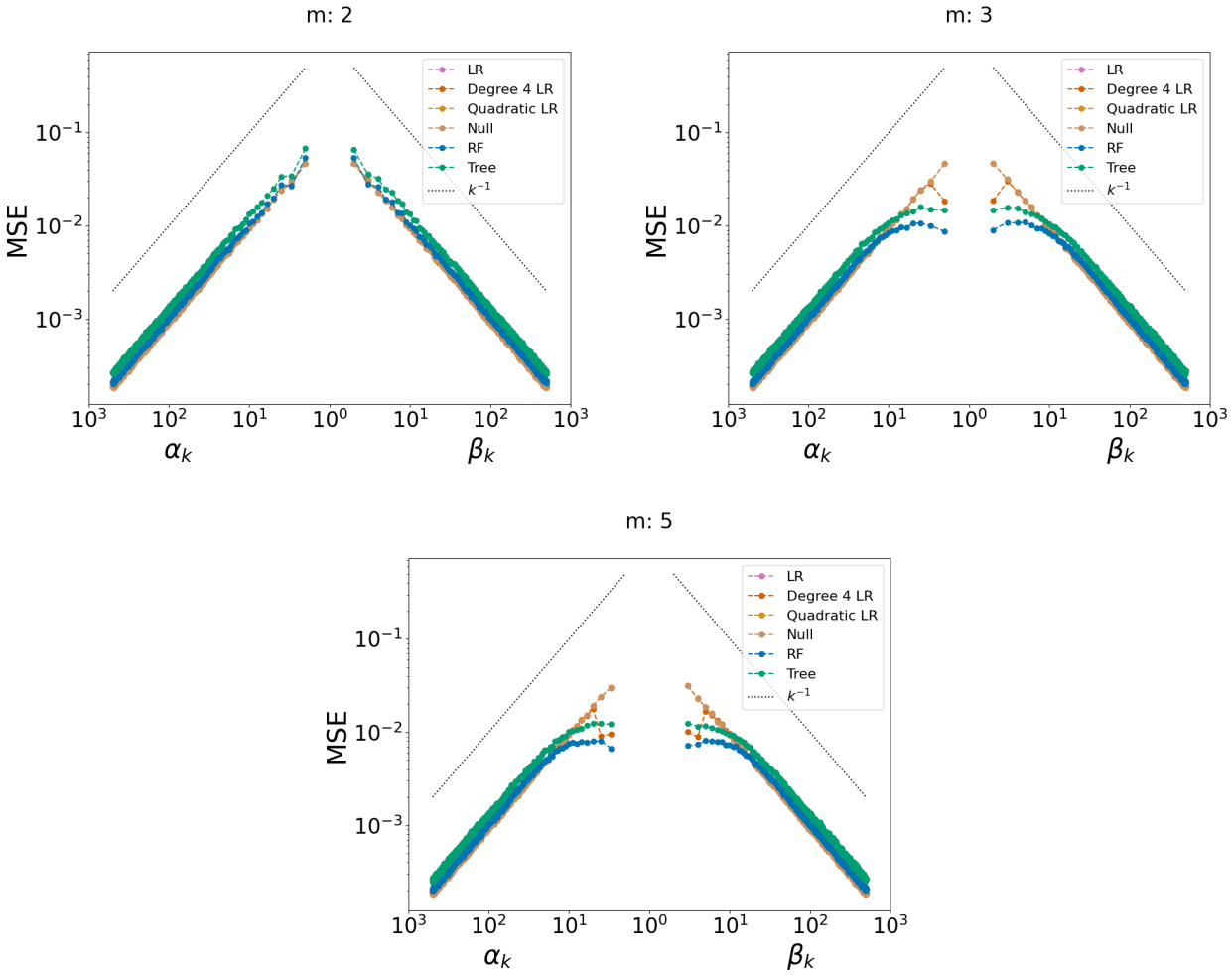


Figure 5.6: Same test as Figure 1 for the three parameter family $\mathcal{K}$, using 10 000 samples and $\ell_{\min} = 0.01$.

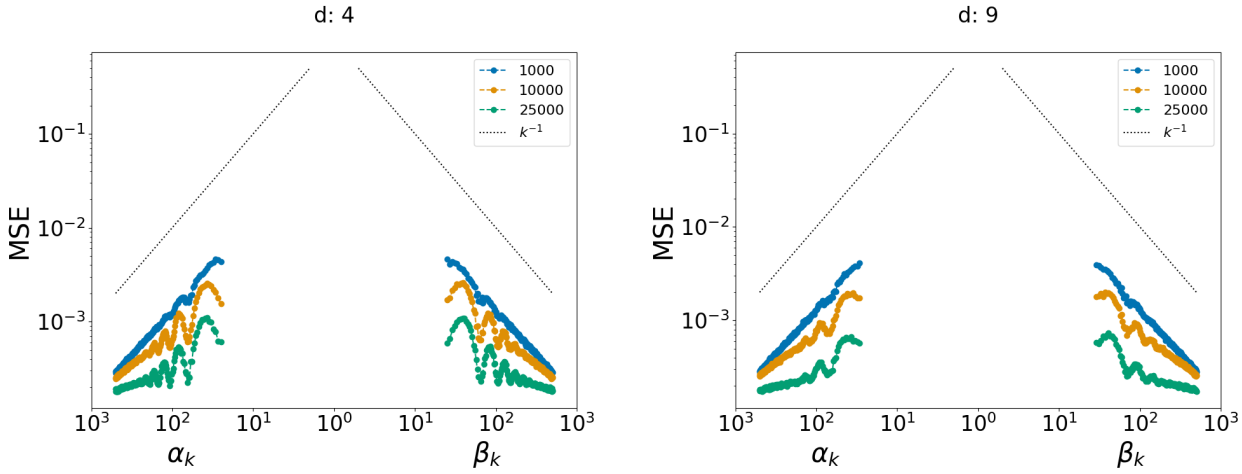


Figure 5.7: Recovery of components k for $k > 10 + d$ for the two parameter family $\mathcal{K}_{\ell=.5}$ using random forest and components j for $j = 10, 11, \ldots, 10 + d$ with $d = 4$ (left) and $d = 9$ (right), and various numbers of training samples.

Chapter 6

Deep learning-based schemes for singularly perturbed convection-diffusion problems

6.1 Introduction

6.1.1 Scientific context and goals

Singularly perturbed differential equations are typically characterized by a small parameter $\varepsilon > 0$ multiplying some of the highest order terms in the differential equation. In general, the solutions to such equations exhibit multiscale phenomena, and this raises significant challenges to classical numerical methods such as finite elements or finite volumes. To build accurate and robust approximations with these methods as ε decreases, it is necessary to develop elaborate numerical discretizations. In addition to the mathematical difficulties of the formulation, the resulting numerical schemes are often not entirely trivial to implement: they often require mesh adaptation, and working on complicated geometries is challenging. These difficulties motivate the search for new discretization schemes, hopefully mesh-free, with potential to deliver good quality approximations with easier implementation techniques. In this work, we explore this research direction, and consider strategies based on deep learning techniques. Our main goal is to test various neural network-based schemes, so as to design a strategy which should be robust when $\varepsilon \to 0$, easily implementable even for complicated geometries, and with potential to scale in high dimension.

The idea of working with neural network functions to solve PDEs is by far not novel, and countless contributions have been proposed on this front in recent years. The strategies can be roughly classified into two categories:

1. In the first category, deep neural networks are employed to assist classical numerical methods by improving some limitations, or accelerating certain steps (see, e.g., [103, 154]). It has notably been used to assist in the construction of numerical fluxes adapted to Finite Volumes (see, e.g., [62]). They can also be used in order to compute

reduced-order models of parametric problems: for each value of the parameters, the solution (or other quantities of interest) of the model of interest is computed by means of a standard numerical scheme, and the values of the solutions are interpolated by mean of a deep neural network over the whole range of parameter values [151, 75].

2. In the second category, neural networks are used to directly approximate the solution of PDEs. The solution schemes become in this case an optimization problem where it is crucial to design appropriate loss functions. The loss functions are mostly based on residuals of the equations, and yield to different methods depending on the specific choice:

   (a) Physics-informed neural networks (PINNs, [134]) is a collocation-based method. One finds the coefficients of the neural network solution by minimizing a discretized version of the L^2 norm of the strong form of the residual of the PDE. This method is very easily implementable but it implicitly assumes that solutions are very regular.

   (b) Other strategies leverage weak variational formulations where less regular solutions are allowed. On this front, most of the classical methods originally formulated for piecewise polynomial functions have by now been tested with trial and test spaces of neural network functions. In this respect, the deep Galerkin method (DGM, [146]) is based on a least-squares formulation, and the variational PINNS (VPINNs, [99, 100]) is based on the Galerkin method. The main drawback of this approach is that the approximation quality depends on the architecture of both the trial *and* the test neural network classes. In addition, numerous evaluations for multiple test functions need to be performed. Also, strategies involving the minimization of weak-form residuals are usually not trivial to implement because they involve the computation of norms in very weak spaces which necessitate extra discretization steps.

   (c) Another approach based on weak variational formulations is the so-called deep Ritz method (DRM, [70]). It leverages the fact that the solution of certain PDEs is the unique minimizer to a certain energy functional. When possible, this approach seems the most appealing: the loss function is naturally given by the problem, it can accommodate low regular solutions, and the computational cost is moderate in the sense that it only requires to handle test functions (no trial functions). It also carries potential to address high dimensional problem as illustrated in [69, 70, 139].

6.1.2 Contribution

The goal of this work is to compare and develop several neural network schemes for singularly perturbed problems when $\varepsilon \to 0$. We focus more particularly on convection-diffusion (or stationary Fokker-Planck) problems with vanishing diffusion for which we explore schemes from the second category according to the above distinction. In other words, we approximate solutions of singular PDEs with feed-forward neural network functions.

When $\varepsilon \to 0$, the regularity of the solutions is deteriorated because of local or boundary thin layers. The classical formulation commonly used in neural network based schemes is constructed from the strong formulation of the problem, where the analytical solution is approximated by the one generated by evaluations on the sampling points. It is referred in this paper as vanilla PINNs, which has been introduced in the (2)-(a) subcategory above. More details could be found in Section 6.2.3. Therefore, it is expected to perform poorly for small values of ε because it commits a "variational crime" (and this is actually confirmed in our numerical experiments). By "variational crime", we mean that the norm of the residual which is traditionally used in vanilla PINNs methods requires that the solution of the PDE has to be more regular than what would naturally be expected from the theory. In our case (convection-diffusion problems), the Vanilla-PINN method requires the solution to belong to an H^2 space, whereas it is more natural from a theoretical point of view to look for an approximation of the solution in a set of H^1 functions. Methods based on weak variational formulations seem better adapted, and on that front, it is desirable to work with the deep Ritz method. However, finding energy formulations is not straightforward due to the non-symmetric nature of convective effects. We show how this method can be applied in this context thanks to a change of variable. We compare its numerical robustness with respect to the PINNs method, and a naive finite element discretization with a uniform grid. In the present study, our tests are performed on a 1D example. Despite its simplicity, the example exhibits all the features that are challenging for numerical schemes. For our purposes, the example also presents the important advantage of having analytic solutions which we can leverage in our error analysis, and our validations. Higher-dimensional tests involving also more elaborate sampling strategies are left for future work.

The paper is organized as follows. In Section 6.2, various formulations of the convection-diffusion problem we are interested in are introduced. In Section 6.3, we introduce various neural networks-based schemes which are inspired from the various formulations introduced in Section 6.2. The reader is encouraged to observe that an expert mathematical insight is required in order to build formulations that do not incur in variational crimes. In Section 6.4, these various schemes are compared for one-dimensional problem. To conclude, in Section 6.4.4, after the presentation of the numerical results, it is summarized how each one of the PINN methods behave for small values of ε and its comparison with the FEM method.

6.2 A singularly perturbed convection-diffusion equation

The aim of this section is to introduce the singularly perturbed convection-diffusion equation we consider in this work, and various formulations of the problem which will be used in Section 6.3 so as to design various neural networks-based schemes for its numerical solution.

6.2.1 Problem definition

As a prototypical example, we consider the following singularly perturbed convection-diffusion equation on a given domain $\Omega \subset \mathbb{R}^d$, with $d \in \mathbb{N}^*$. Let $F : \Omega \to \mathbb{R}^d$ be a

given force field, $0 < \varepsilon \ll 1$ a small parameter, and $f : \Omega \to \mathbb{R}$ be a given right-hand side function. Our goal is to find a solution $u : \Omega \to \mathbb{R}$ to

$$-\varepsilon(\Delta u)(x) + \nabla \cdot (Fu)(x) = f(x), \quad \forall x \in \Omega, \tag{6.1}$$

together with Robin boundary conditions

$$\alpha(\nabla u \cdot n)(x) + \kappa u(x) = g(x) \quad \forall x \in \partial\Omega, \tag{6.2}$$

where n refers to the outward unit vector of $\partial\Omega$, α, $\kappa \geqslant 0$ and g is a real-valued function defined on $\partial\Omega$. In the following, we assume that the force field F derives from a potential function $V : \Omega \to \mathbb{R}$, in the sense that

$$F(x) = -\nabla V(x), \quad \forall x \in \Omega.$$

Under appropriate assumptions on F (or V), f and g, which are assumed to be smooth functions for the sake of simplicity, (6.1) and (6.2) can be proved to have a unique solution [137, 97, 31]. Note that, more generally, α and κ could also be given as real-valued functions defined on $\partial\Omega$, instead of constants, and our subsequent developments could be easily adapted.

The equation represents the change in the concentration u of a quantity in a given medium, and in presence of convective and diffusive effects. The force field F represents the drag force while the singular perturbation parameter ε represents the diffusivity of the medium. In the limit of an inviscid medium as $\varepsilon \to 0$, the equation changes from elliptic to hyperbolic nature, and from second to first order. For Dirichlet boundary conditions $u = 0$ on $\partial\Omega$, the solution can develop sharp boundary layers of width ε near the outflow. We refer the reader to [138] for general references on this equation regarding its analysis and numerical methods.

Classical numerical methods are challenged by problem (6.1) when ε is small. In the case of the Galerkin finite element method, the poor performance for this problem is reflected in the bound on the error in the finite element solution. For $\Omega = (0, 1)$ and Dirichlet boundary conditions, a standard Galerkin method with a uniform grid of size h delivers a solution u_h on a finite element space $\mathbb{P}_h$ that satisfies

$$\|u - u_h\|_{H^1(0,1)} \leqslant C(\varepsilon) \inf_{w_h \in \mathbb{P}_h} \|u - w_h\|_{H^1(0,1)}, \tag{6.3}$$

where $C(\varepsilon) \sim \varepsilon^{-1}$, so that the constant blows up as $\varepsilon \to 0$ (see [138, Theorem 2.49]). The dependence of C on ε is usually referred to as a *loss of robustness* in the sense that, as ε decreases, the Galerkin method is bounded more and more loosely by the best approximation error. As a consequence, on a coarse mesh and for small values ε, the Galerkin approximation develops spurious oscillations everywhere in the domain. This very well-known behaviour will actually be observed later on in our numerical tests.

Numerous methods have been proposed in order to address this loss of robustness in finite element methods. An important family of methods is based on using residual-based

stabilization techniques. Given some variational form, the problem is modified by adding to the bilinear form the strong form of the residual, weighted by a test function and scaled by a stabilization constant τ. The most well-known example of this technique is the streamline upwind Petrov-Galerkin (SUPG) method (see [36]). The addition of the residual-based stabilization term, can be interpreted as a modification of the test functions which means that these methods seek stabilization by changing the test space, and motivates to search for optimal test spaces in the spirit of [61, 44].

Other classical discretization methods such as finite volumes suffer from similar issues, and strategies involving layer-adaptive grids such as Shishkin meshes have been proposed (see, e.g., [101]).

The aim of this work is to explore the potential of approximating solutions of such problems with neural network functions, and the next section presents several options for this, with a discussion on their merits and limitations.

6.2.2 General formulation

Any neural-network based numerical scheme for the solution of (6.1) and (6.2) relies on the use of a variational formulation of this problem which enables to write u (or another function defined from u) as a minimizer of a problem of the form

$$\min_{v \in \mathcal{V}} \mathcal{J}(v), \tag{6.4}$$

where $\mathcal{V}$ is a particular set of real-valued functions defined on Ω. The loss function $\mathcal{J} : \mathcal{V} \to \mathbb{R}$ is usually of the form

$$\mathcal{J}(v) := \int_\Omega \mathcal{R}(v)(x) \mathrm{d}\rho(x) + \int_{\partial\Omega} \mathcal{S}(v)(r) \mathrm{d}\tau(r), \quad \forall v \in \mathcal{V}, \tag{6.5}$$

where for every $v \in \mathcal{V}$, $\mathcal{R}(v)$ and $\mathcal{S}(v)$ are real-valued functions defined on Ω and $\partial\Omega$ respectively. They are assumed to be measurable with respect to the measures ρ and τ, which are defined on Ω and $\partial\Omega$ respectively. Note that the measures ρ and τ have to be chosen a priori, and they can greatly affect the final result. The question of discovering the optimal weights is beyond the scope of our present investigation.

The aim of the next sections is to introduce various formulations of (6.1) and (6.2) under the form given by (6.4) and (6.5). This requires appropriate definitions of the set $\mathcal{V}$, the functions $\mathcal{R}(v)$ and $\mathcal{S}(v)$ for any $v \in \mathcal{V}$ and the unknown function solution of (6.4). Unless otherwise stated, the measures ρ and τ will be defined as the Lebesgue bulk measure and Lebesgue surfacic measure respectively.

6.2.3 Vanilla (V) formulation

We begin by introducing the most classical formulation used in neural network-based numerical schemes such as PINNs. For the reasons that we outline next, different aspects of this formulation can be improved, therefore we refer to it as *vanilla* (V) formulation in the following.

The formulation consists in interpreting the solution u of (6.1) and (6.2) as the unique solution of a minimization problem of the form (6.4) with $\mathcal{V} = H^2(\Omega)$ and to define for all $v \in \mathcal{V}$,

$$\begin{cases} \mathcal{R}(v)(x) := \lambda \left| -\varepsilon(\Delta v)(x) + \nabla \cdot (Fv)(x) - f(x) \right|^2, & \text{for all } x \in \Omega, \\ \mathcal{S}(v)(x) := (1-\lambda) \left| \alpha(\nabla v \cdot n)(x) + \kappa v(x) - g(x) \right|^2, & \text{for all } x \in \partial\Omega, \end{cases} \tag{6.6}$$

for some $\lambda \in (0,1)$. In this approach, the parameter λ enables to tune the respective weight of the contributions of the bulk and boundary terms in the total functional $\mathcal{J}$ to be minimized. In practice, in the numerical tests presented in Section 6.4, λ will always be chosen to be equal to $\frac{1}{2}$.

Note that such an approach requires the solution u to belong to $H^2(\Omega)$, which implies that the solution has to be sufficiently regular. When $\varepsilon \to 0$, this assumption becomes less and less realistic due to the formation of boundary layers. This raises the question as to whether it is possible to introduce another formulation of (6.1) and (6.2) which would allow for less regular solutions. The goal of the next section is to introduce such an alternative formulation.

6.2.4 Weak variational (W) formulation

In this section we develop an avenue based on an energy minimization approach which requires less regularity in the solutions than the vanilla formulation. To this aim, we introduce the change of variable

$$u(x) = e^{cV(x)} z(x), \tag{6.7}$$

where $c \in \mathbb{R}$ is a constant yet to be determined. Taking first and second derivatives in (6.7) yield that for all $x \in \Omega$,

$$\begin{aligned} \nabla u(x) &= e^{cV(x)} \left(c\nabla V(x) z(x) + \nabla z(x) \right) \\ \Delta u(x) &= e^{cV(x)} \left(c\Delta V(x) z(x) + |c\nabla V(x)|^2 z(x) + 2c\nabla V(x) \cdot \nabla z(x) + \Delta z(x) \right). \end{aligned}$$

Now, setting the value of c to be

$$c = \frac{1}{2\varepsilon},$$

and inserting the change of variable into (6.1), we conclude that u is a solution to (6.1) if and only if z is a solution to the elliptic problem

$$-\Delta z(x) + \left(\frac{\Delta V(x)}{2\varepsilon} + \frac{|\nabla V(x)|^2}{4\varepsilon^2} \right) z(x) = f(x) \frac{e^{-\frac{V(x)}{2\varepsilon}}}{\varepsilon}, \quad \forall x \in \Omega, \tag{6.8}$$

with Robin boundary conditions

$$\alpha(\nabla z(x) \cdot n(x)) + \left(\kappa + \frac{\alpha}{2\varepsilon} \nabla V(x) \cdot n(x) \right) z(x) = e^{\frac{-V(x)}{2\varepsilon}} g(x), \quad \forall x \in \partial\Omega. \tag{6.9}$$

At this stage, one could of course apply the vanilla formulation to solve (6.8) and (6.9) and compute z solution of a minimization problem of the form (6.4) with $\mathcal{V} = H^2(\Omega)$ and the functionals $\mathcal{R}$ and $\mathcal{S}$ defined by

$$\begin{cases} \mathcal{R}(v)(x) := \lambda \left| -\Delta v(x) + \left(\frac{\Delta V(x)}{2\varepsilon} + \frac{|\nabla V(x)|^2}{4\varepsilon^2} \right) v(x) - f(x) \frac{e^{-\frac{V(x)}{2\varepsilon}}}{\varepsilon} \right|^2, \quad \forall x \in \Omega, \\ \mathcal{S}(v)(x) := (1-\lambda) \left| \alpha(\nabla v(x) \cdot n(x)) + \left(\kappa + \frac{\alpha}{2\varepsilon} \nabla V(x) \cdot n(x) \right) v(x) - e^{\frac{-V(x)}{2\varepsilon}} g(x) \right|^2, \quad \forall x \in \partial\Omega, \end{cases} \tag{6.10}$$

for all $v \in \mathcal{V} = H^2(\Omega)$ and some $\lambda \in (0,1)$. The value of λ chosen in our numerical tests is $\lambda = 0.5$. We will refer to this approach as the *vanilla-z* (Vz) formulation.

Note that this method does not fully exploit the change of variables since the elliptic nature of (6.8) allows us to easily build a weak formulation of this equation. Testing against a smooth test function v and integrating by parts we obtain the weak formulation

$$\int_\Omega \nabla z \cdot \nabla v - \int_{\partial\Omega} v \nabla z \cdot n \mathrm{d}x + \int_\Omega \left(\frac{\Delta V}{2\varepsilon} + \frac{|\nabla V|^2}{4\varepsilon^2} \right) zv = \int_\Omega f \frac{e^{-\frac{V}{2\varepsilon}}}{\varepsilon} v.$$

Using equality (6.9), we get

$$\begin{aligned} \int_\Omega \nabla z \cdot \nabla v + \int_\Omega \left(\frac{\Delta V}{2\varepsilon} + \frac{|\nabla V|^2}{4\varepsilon^2} \right) zv + \int_{\partial\Omega} \left(\frac{\kappa}{\alpha} + \frac{1}{2\varepsilon} \nabla V \cdot n \right) zv \\ = \int_\Omega f \frac{e^{-\frac{V}{2\varepsilon}}}{\varepsilon} v + \int_{\partial\Omega} \frac{1}{\alpha} e^{\frac{-V}{2\varepsilon}} gv. \end{aligned}$$

Therefore the weak formulation of (6.8) is to find $z \in H^1(\Omega)$ such that

$$a(z,v) = \ell(v), \quad \forall v \in H^1(\Omega) \tag{6.11}$$

with

$$\begin{aligned} a(z,v) &:= \int_\Omega \nabla z \cdot \nabla v + \int_\Omega \left(\frac{\Delta V}{2\varepsilon} + \frac{|\nabla V|^2}{4\varepsilon^2} \right) zv \mathrm{d}x + \int_{\partial\Omega} \left(\frac{\kappa}{\alpha} + \frac{1}{2\varepsilon} \nabla V \cdot n \right) zv \\ l(v) &:= \int_\Omega f \frac{e^{-\frac{V}{2\varepsilon}}}{\varepsilon} v + \int_{\partial\Omega} \frac{1}{\alpha} e^{\frac{-V}{2\varepsilon}} gv. \end{aligned}$$

To ensure that the symmetric bilinear form a is continuous and coercive, we assume in the sequel that the following conditions are satisfied:

$$\begin{cases} \left(\frac{\Delta V(x)}{2\varepsilon} + \frac{|\nabla V(x)|^2}{4\varepsilon^2} \right) \geqslant a_0 > 0, & \forall x \in \Omega \\ \left(\frac{\kappa}{\alpha} + \frac{1}{2\varepsilon} \nabla V(x) \cdot n(x) \right) \geqslant 0, & \forall x \in \partial\Omega \\ f \in L^2(\Omega),\ g \in L^2(\partial\Omega) \end{cases} \tag{6.12}$$

If conditions (6.12) are satisfied, z can be equivalently rewritten as the unique solution

of a minimization problem of the form

$$z = \underset{v \in H^1(\Omega)}{\operatorname{argmin}} \frac{1}{2} a(v,v) - \ell(v). \tag{6.13}$$

This implies that z can equivalently be recast as the unique solution of a minimization problem of the form (6.4) with $\mathcal{V} = H^1(\Omega)$ and

$$\begin{cases} \mathcal{R}(v)(x) & := \frac{1}{2}\left[|\nabla v(x)|^2 + \left(\frac{\Delta V(x)}{2\varepsilon} + \frac{|\nabla V(x)|^2}{4\varepsilon^2}\right)|v(x)|^2\right] - f(x)\frac{e^{-\frac{V(x)}{2\varepsilon}}}{\varepsilon} v(x), \quad \forall x \in \Omega, \\ \mathcal{S}(v)(x) & := \frac{1}{2}\left[\left(\frac{\kappa}{\alpha} + \frac{1}{2\varepsilon}\nabla V(x)\cdot n(x)\right)|v(x)|^2\right] - \frac{1}{\alpha} e^{\frac{-V(x)}{2\varepsilon}} g(x) v(x), \quad \forall x \in \partial\Omega. \end{cases} \tag{6.14}$$

We will refer to this approach as the *weak-z* (Wz) formulation.

Moreover, using (6.7), we can equivalently express u as a solution of a minimization problem of the form (6.4) with

$$\mathcal{V} := \left\{ v = e^{\frac{V}{2\varepsilon}} \overline{v},\ \overline{v} \in H^1(\Omega) \right\}, \tag{6.15}$$

and rewriting the Equation (6.14)

$$\begin{cases} \mathcal{R}(v)(x) & := \frac{1}{2}\left[|\nabla \overline{v}(x)|^2 + \left(\frac{\Delta V(x)}{2\varepsilon} + \frac{|\nabla V(x)|^2}{4\varepsilon^2}\right)|\overline{v}(x)|^2\right] - f(x)\frac{e^{-\frac{V(x)}{2\varepsilon}}}{\varepsilon} \overline{v}(x), \quad \forall x \in \Omega, \\ \mathcal{S}(v)(x) & := \frac{1}{2}\left[\left(\frac{\kappa}{\alpha} + \frac{1}{2\varepsilon}\nabla V(x)\cdot n(x)\right)|\overline{v}(x)|^2\right] - \frac{1}{\alpha} e^{\frac{-V(x)}{2\varepsilon}} g(x) \overline{v}(x), \quad \forall x \in \partial\Omega, \end{cases} \tag{6.16}$$

for all $v \in \mathcal{V}$ with $\overline{v} := v e^{-\frac{V}{2\varepsilon}}$ the change of variable suggested above. We will refer to this formulation as the *weak* (W) formulation.

Note that in the non-discretized case, formulations (W) and (Wz) are equivalent up to the exponential change of variable. However, when the minimizer of both formulations is computed by means of a neural network, the corresponding approximations will be different and have different accuracies. The (W) formulation has the advantage to avoid potential machine precision issues linked to the presence of the exponential term when ε becomes small.

6.2.5 Rescaled formulation

In this section, we introduce another formulation based on a change of scale in the original problem. More precisely, introducing $\Omega_\varepsilon := \frac{1}{\varepsilon}\Omega$, we introduce auxiliary functions $\widetilde{u} : \Omega_\varepsilon \to \mathbb{R}$, $\widetilde{z} : \Omega_\varepsilon \to \mathbb{R}$ and $\widetilde{V} : \Omega_\varepsilon \to \mathbb{R}$ defined so that for all $x \in \Omega$,

$$u(x) = \varepsilon \widetilde{u}\left(\frac{x}{\varepsilon}\right), \quad z(x) = \varepsilon \widetilde{z}\left(\frac{x}{\varepsilon}\right), \quad V(x) = \varepsilon \widetilde{V}\left(\frac{x}{\varepsilon}\right). \tag{6.17}$$

Notice that if u and z satisfy (6.7), then

$$\widetilde{z}(y) = \widetilde{u}(y) e^{\frac{1}{2}\widetilde{V}(y)}, \quad \forall y \in \Omega_\varepsilon.$$

Denoting by $\widetilde{F}(y) := -\nabla\widetilde{V}(y) = F(\varepsilon y)$ for all $y \in \Omega_\varepsilon$, it holds that u is solution to (6.1) and (6.2) if and only if $\widetilde{u}$ is solution to

$$-\Delta\widetilde{u}(y) + \nabla\cdot\left(\widetilde{F}\widetilde{u}\right)(y) = \widetilde{f}(y), \quad \forall y \in \Omega_\varepsilon \tag{6.18}$$

where $\widetilde{f}(y) := f(\varepsilon y)$ for all $y \in \Omega_\varepsilon$ with boundary conditions

$$\alpha(\nabla\widetilde{u}\cdot n)(y) + \varepsilon\kappa\widetilde{u}(y) = \widetilde{g}(y), \quad \forall y \in \partial\Omega_\varepsilon, \tag{6.19}$$

with $\widetilde{g}(y) := g(\varepsilon y)$ for all $y \in \Omega_\varepsilon$.

Using similar calculations to the ones done in Section 6.2.4, the Lax-Milgram theorem guarantees that $\widetilde{z}$ is the unique solution in $H^1(\Omega_\varepsilon)$ of the following variational problem: for all $\widetilde{v} \in H^1(\Omega_\varepsilon)$,

$$\begin{aligned}\int_{\Omega_\varepsilon}\nabla\widetilde{z}(y)\cdot\nabla\widetilde{v}(y)\,dy + \int_{\partial\Omega_\varepsilon}\left(\frac{\varepsilon\kappa}{\alpha} + \frac{1}{2}\nabla\tilde{V}(y)\cdot n(y)\right)\widetilde{z}(y)\widetilde{v}(y)\,dy \\ + \int_{\Omega_\varepsilon}\left(\frac{\Delta\tilde{V}(y)}{2} + \frac{|\nabla\tilde{V}(y)|^2}{4}\right)\widetilde{z}(y)\,\widetilde{v}(y)\,dy \\ = \int_{\Omega_\varepsilon}\widetilde{f}(y)e^{-\frac{1}{2}\tilde{V}(y)}\widetilde{v}(y)\,dy + \int_{\partial\Omega_\varepsilon}\frac{1}{\alpha}e^{-\frac{1}{2}\tilde{V}(y)}\widetilde{g}(y)\widetilde{v}(y)\,dy.\end{aligned} \tag{6.20}$$

The result is valid provided that the following assumptions on the coefficients hold

$$\begin{cases}\Delta\tilde{V}(y) + |\nabla\tilde{V}|^2(y) \geqslant 0, \ \forall y \in \Omega_\varepsilon, \\ \frac{\kappa\varepsilon}{\alpha} + \frac{1}{2}\nabla\tilde{V}(y)\cdot n(y) \geqslant 0, \ \forall y \in \partial\Omega_\varepsilon, \\ \widetilde{f} \in L^2(\Omega_\varepsilon), \ \widetilde{g} \in L^2(\partial\Omega_\varepsilon).\end{cases}$$

The above conditions are equivalent to the assumptions (6.12) stated in Section 6.2.4.

As in the previous section, the function $\widetilde{z}$ is then the unique solution of a minimization problem of the form (6.4) with $\mathcal{V} = H^1(\Omega_\varepsilon)$ with

$$\begin{cases}\mathcal{R}(v)(y) &:= \frac{1}{2}\left[|\nabla v(y)|^2 + \left(\frac{\Delta\tilde{V}(y)}{2} + \frac{|\nabla\tilde{V}(y)|^2}{4}\right)|v(y)|^2\right] - \widetilde{f}(y)e^{-\frac{\tilde{V}(y)}{2}}v(y), \quad \forall y \in \Omega_\varepsilon, \\ \mathcal{S}(v)(y) &:= \frac{1}{2}\left[\left(\frac{\varepsilon\kappa}{\alpha} + \frac{1}{2}\nabla\tilde{V}(y)\cdot n(y)\right)|v(y)|^2\right] - \frac{1}{\alpha}e^{\frac{-\tilde{V}(y)}{2}}\widetilde{g}(y)v(y), \quad \forall y \in \partial\Omega_\varepsilon.\end{cases} \tag{6.21}$$

We will refer to this approach as the *rescaled-weak-z* (RWz) formulation.

6.2.6 Summary of the methods

For the sake of clarity, we summarize here the main features of each method.

Method	Acronym	Unknown	$\mathcal{V}$	$\mathcal{R}$ and $\mathcal{S}$
vanilla	V	u	$H^2(\Omega)$	Equation (6.6)
vanilla-z	Vz	z	$H^2(\Omega)$	Equation (6.10)
weak-z	Wz	z	$H^1(\Omega)$	Equation (6.14)
weak	W	u	(6.15)	Equation (6.16)
rescaled-weak-z	RWz	$\widetilde{z}$	$H^1(\Omega_\varepsilon)$	Equation (6.21)

6.3 Neural networks based numerical schemes

In this section we describe the numerical approach used in order to compute an approximation of the solution of a minimization problem of the form (6.4) by means of a neural-network based method. We first present in Section 6.3.1 the general principle of such approaches. The main ingredients to design a neural-network based method consist in the choice of a class of neural network functions and of sampling schemes in order to approximate the integrals involved in the definition of the loss function $\mathcal{J}$ defined by (6.5). These two ingredients are detailed respectively in Sections 6.3.2 and 6.3.3. Finally, some details on the numerical implementation are given in Section 6.4.2.

6.3.1 General principle

The numerical solution of a minimization problem of the form (6.4) usually requires to consider alternatives to $\mathcal{V}$ and $\mathcal{J}$ that are amenable for practical implementation. The strategy thus consists in formulating a related problem of the form

$$\min_{v \in \mathcal{K}} \widehat{\mathcal{J}}(v), \tag{6.22}$$

where

- $\mathcal{K} \subset \mathcal{V}$ is a set of functions parametrized by a finite number of scalar coefficients. A classical class of functions are finite elements. Here, we consider neural networks (see Section 6.3.2 below);
- $\widehat{\mathcal{J}}$ is an approximation of the loss function $\mathcal{J}$ where the integrals are approximated using some particular quadrature or sampling schemes.

More precisely, for given integers $K, M \in \mathbb{N}^*$, given sets of points $(x_k)_{1 \leqslant k \leqslant K} \subset \Omega$, $(y_m)_{1 \leqslant m \leqslant M} \subset \partial\Omega$, and given sets of weights $(\rho_k)_{1 \leqslant k \leqslant K} \subset \mathbb{R}$ and $(\tau_m)_{1 \leqslant m \leqslant M} \subset \mathbb{R}$, for all $v \in \mathcal{K}$, the functional $\widehat{\mathcal{J}}(v)$ is defined by

$$\widehat{\mathcal{J}}(v) := \sum_{k=1}^{K} \rho_k \mathcal{R}(v)(x_k) + \sum_{m=1}^{M} \tau_m \mathcal{S}(v)(y_m). \tag{6.23}$$

As a consequence, the definition of a neural-network based numerical scheme for the approximation of a problem of the form (6.4) requires the definition of two ingredients:

- the class $\mathcal{K} \subset \mathcal{V}$ of neural network functions;
- the sampling scheme, i.e. the choice of K, M, $(x_k)_{1\leqslant k\leqslant K}$, $(y_m)_{1\leqslant m\leqslant M}$, $(\rho_k)_{1\leqslant k\leqslant K}$ and $(\tau_m)_{1\leqslant m\leqslant M}$ in order to define the approximate functional $\widehat{\mathcal{J}}$ given by (6.23).

The set of neural network functions $\mathcal{K}$ we consider in our numerical experiments is presented in Section 6.3.2. The various sampling schemes tested here are given in Section 6.3.3.

6.3.2 Neural Network classes of functions

In this work, we only consider classes of functions defined by means of feed-forward neural networks whose definition we recall next (see [106] for general references).

Let $\mathcal{X} \subset \mathbb{R}^{d_{\mathcal{X}}}$ and $\mathcal{Y} \subset \mathbb{R}^{d_{\mathcal{Y}}}$ be some input and output sets of finite dimensions $d_{\mathcal{X}}, d_{\mathcal{Y}} \in \mathbb{N}^*$. A feed-forward neural network is a function

$$\psi : \mathcal{X} \to \mathcal{Y}$$

which reads as

$$\psi(x) = T_L(\sigma(T_{L-1}(\sigma(\dots\sigma(T_0(x))))), \quad \forall x \in \mathcal{X}. \tag{6.24}$$

For every $\ell \in \{0, \dots, L\}$,

$$T_\ell : \left\{ \begin{array}{ccc} \mathbb{R}^{p_\ell} & \to & \mathbb{R}^{p_{\ell+1}} \\ x_\ell & \mapsto & T_\ell(x_\ell) \coloneqq A_\ell x_\ell + b_\ell \end{array} \right. \tag{6.25}$$

is an affine function which can be expressed through a matrix $A_\ell \in \mathbb{R}^{p_{\ell+1}\times p_\ell}$, and an offset vector $b_\ell \in \mathbb{R}^{p_{\ell+1}}$, and $\sigma : \mathbb{R} \to \mathbb{R}$ is called the (nonlinear) activation function. By a slight abuse of notation, for all $p \in \mathbb{N}^*$ and for any vector $w \coloneqq (w_i)_{1\leqslant i\leqslant p} \subset \mathbb{R}^p$, the notation $\sigma(w)$ actually denotes the vector of $\mathbb{R}^p$ with entries $\sigma(w_i)$, that is, $\sigma(w) = (\sigma(w_i))_{i=1}^p$. Note that since ψ maps $\mathcal{X}$ onto $\mathcal{Y}$, it is necessary that $p_0 = d_{\mathcal{X}}$ and $p_{L+1} = d_{\mathcal{Y}}$. The layers numbered from 1 to L are usually called the hidden layers of the neural network.

To define a class of feed-forward neural networks, we fix an architecture by prescribing a given activation function σ, depth $L \in \mathbb{N}$, and layer widths $\boldsymbol{p} = (p_0, \dots, p_{L+1}) \in (\mathbb{N}^*)^{L+2}$. Once the values of σ, L and $\boldsymbol{p}$ have been chosen, we view the coefficients $(A_\ell, b_\ell)_{0\leqslant\ell\leqslant L}$ of the affine mappings $T_0, \cdots, T_L$ as parameters. We gather these coefficients in the vector of parameters

$$\theta \coloneqq \{(A_\ell, b_\ell)\}_{\ell=0}^L,$$

and assume that θ takes values in a set

$$\Theta \subseteq \bigtimes_{\ell=0}^{L} \left(\mathbb{R}^{p_\ell\times p_{\ell+1}} \times \mathbb{R}^{p_{\ell+1}}\right).$$

For any $\theta \in \Theta$, we define by $\psi_\theta : \mathcal{X} \to \mathcal{Y}$ the function ψ defined by (6.24) with $\theta = \{(A_\ell, b_\ell)\}_{\ell=0}^L \in \Theta$.

The class of neural network functions with architecture $(\sigma, L, \boldsymbol{p})$ and coefficient sets Θ

is then defined as

$$\mathcal{N}(\sigma, L, \boldsymbol{p}, \Theta) := \{\psi_\theta \ : \ \theta \in \Theta\}.$$

In our context, the input and output sets $\mathcal{X}$ and $\mathcal{Y}$ are respectively given by

$$\mathcal{X} = \Omega \text{ (or } \Omega_\varepsilon) \quad \text{and} \quad \mathcal{Y} = \mathbb{R},$$

so that $d_\mathcal{X} = d$ and $d_\mathcal{Y} = 1$. In all the numerical tests presented below, the class $\mathcal{K}$ is chosen as

$$\mathcal{K} := \mathcal{N}(\sigma, L, \boldsymbol{p}, \Theta),$$

with

$$\sigma = \tanh,\ L = 2,\ \boldsymbol{p} = (d, 10, 10, 1) \text{ and } \Theta = \bigtimes_{\ell=0}^{L} \left(\mathbb{R}^{p_\ell \times p_{\ell+1}} \times \mathbb{R}^{p_{\ell+1}}\right).$$

Note that the set $\mathcal{K}$ is then a subset of $\mathcal{V}$ for all the formulations of the convection-diffusion problem we introduced in Section 6.2. Moreover, the solution of the approximate (6.22) is equivalent to finding a minimizer $\theta^* \in \Theta$ solution to

$$\min_{\theta \in \Theta} \widehat{\mathcal{J}}(\psi_\theta). \tag{6.26}$$

Remark: In many machine learning applications, the choice of relu activation functions is very common due to its low computational cost when performing evaluation or first order differentiation. However, in our problem, second order derivatives are needed to calculate the loss function. If relu activation functions were used, then the second order derivative terms would be 0, and no good approximation could be learned. This reason motivates our choice of tanh as the activation function.

6.3.3 Sampling schemes

We detail in this section the various sampling schemes we considered in our numerical tests in order to define the approximate loss function $\widehat{\mathcal{J}}$.

Since we work with one-dimensional examples, we carry the discussion for dimension one. In fact, we consider (6.1) and (6.2) with $\Omega = (0, 1)$ so that $\partial\Omega = \{0\} \cup \{1\}$ (and $\partial\Omega_\varepsilon = \{0\} \cup \{1/\varepsilon\}$). Thus, for all our tests, the domain boundary has $M = 2$ points $y_1 = 0$ and $y_2 = 1$ (or $y_2 = \frac{1}{\varepsilon}$ for the RWz method). Taking $\tau_1 = \tau_2 = 1$ for the surface weights, the surface term in (6.23) takes the simple form

$$\sum_{m=1}^{M=2} \tau_m \mathcal{S}(v)(y_m) = \int_{\partial\Omega} \mathcal{S}(v)\, d\tau \quad \forall v \in \mathcal{V} \text{ (or } \int_{\partial\Omega_\varepsilon} \mathcal{S}(v)\, d\tau \text{ for the RWz formulation)}.$$

We consider three different sampling schemes for the approximation of the bulk term $\int_\Omega \mathcal{R}(v)\, d\rho$:

1. The first choice is a simple *uniform* sampling scheme (labeled $-u$ in our tests). For a given $K \in \mathbb{N}^*$, we set $\rho_k = \frac{1}{K}$ and $(x_k)_{1 \leqslant k \leqslant K}$ as the centers of the intervals given by

a uniform discretization grid of the interval $(0,1)$.

2. The second sampling scheme, called *random* $(-r)$ scheme, consists in choosing $\rho_k = \frac{1}{K}$ and the points $(x_k)_{1\leqslant k\leqslant K}$ as a collection of random points, identically independently distributed according to the uniform distribution on $(0,1)$.

3. We lastly consider a third sampling scheme, called *exponential* $(-e)$ scheme, which is specific to the Wz formulation. Recall that in this case, for all $v \in \mathcal{K}$, the expression of $\mathcal{R}(v)$ is given by (6.14), namely

$$\mathcal{R}(v)(x) = \mathcal{R}^{(1)}(v)(x) + \mathcal{R}^{(2)}(v)(x), \quad \forall x \in \Omega$$

with

$$\begin{cases} \mathcal{R}^{(1)}(v)(x) & := \frac{1}{2}\left[|\nabla v(x)|^2 + \left(\frac{\Delta V(x)}{2\varepsilon} + \frac{|\nabla V(x)|^2}{4\varepsilon^2}\right)|v(x)|^2\right] \\ \mathcal{R}^{(2)}(v)(x) & := -f(x)\frac{e^{-\frac{V(x)}{2\varepsilon}}}{\varepsilon}v(x), \quad \forall x \in \Omega. \end{cases}$$

Thus, the bulk integral term:

$$\int_\Omega \mathcal{R}(v)(x)\mathrm{d}\rho(x) = \int_\Omega \mathcal{R}^{(1)}(v)(x)\mathrm{d}\rho(x) + \int_\Omega \mathcal{R}^{(2)}(v)(x)\mathrm{d}\rho(x),$$

and we approximate each component separately as follows. For the first term, we draw a collection of $K_1 \in \mathbb{N}^*$ independent identically distributed (iid) random points $(x_k^{(1)})_{1\leqslant k\leqslant K_1}$ from the uniform distribution on $(0,1)$ and for all $1 \leqslant k \leqslant K_1$, the weights $\rho_k^{(1)}$ are chosen to be equal to $\frac{1}{K_1}$. For the second term, we draw $K_2 \in \mathbb{N}^*$ iid random points $(x_k^{(2)})_{1\leqslant k\leqslant K_2}$ following the probability density

$$\rho^{(2)}(x) := \frac{e^{-\frac{V(x)}{2\varepsilon}}}{Z_\varepsilon}, \quad x \in \Omega,$$

with

$$Z_\varepsilon := \int_\Omega e^{-\frac{V(x)}{2\varepsilon}}\,dx.$$

Setting now $\rho_k^{(2)} = \frac{Z_\varepsilon}{K_2}$ for all $1 \leqslant k \leqslant K_2$, the integral $\int_\Omega \mathcal{R}(v)$ is then approximated by

$$\sum_{k=1}^{K_1} \rho_k^{(1)} \left(\frac{1}{2}\left[|\nabla v(x_k^{(1)})|^2 + \left(\frac{\Delta V(x_k^{(1)})}{2\varepsilon} + \frac{|\nabla V(x_k^{(1)})|^2}{4\varepsilon^2}\right)|v(x_k^{(1)})|^2\right]\right) - \sum_{k=1}^{K_2} \rho_k^{(2)} \left(f(x_k^{(2)})\frac{1}{\varepsilon}v(x_k^{(2)})\right). \quad (6.27)$$

In the following, we use the notation $-u$ (respectively $-r$ and $-e$), after the name of

a formulation, in order to refer to the numerical method obtained by using this formulation, together with a uniform (respectively random or exponential) sampling scheme. For instance, the $V - u$ method refers to the vanilla formulation used in conjunction with a uniform sampling scheme.

Note that to tackle higher-dimensional problems, special sampling techniques such as adaptive Markov-Chain Monte-Carlo or Quasi Monte Carlo would be required.

6.3.4 Comparison with finite element schemes

One important point in the investigation of the merits and limitations of deep learning-based numerical schemes is to understand how they compare with respect to other existing schemes. In our tests, we provide a numerical comparison with a vanilla finite element Galerkin scheme involving a uniform mesh. For the sake of completeness, we briefly recall the main steps of our finite element Galerkin approach.

Integrating the original Equation (6.1) against a sufficiently smooth function $v \in \mathcal{C}^\infty(\Omega)$, and integrating by parts, it follows that a weak formulation of problem (6.1) is to find $u \in H^1(\Omega)$ such that

$$a(u, v) = l(v), \quad \forall v \in H^1(\Omega)$$

with

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v + \varepsilon^{-1} \int_\Omega \nabla \cdot (Fu)v + \frac{\kappa}{\alpha} \int_{\partial\Omega} uv \tag{6.28}$$

$$l(v) = \varepsilon^{-1} \int_\Omega fv - \int_{\partial\Omega} gv. \tag{6.29}$$

We numerically solve this problem by Galerkin projection. For this, we consider a mesh $(T_n)_{n=1}^N$ of Ω and define the associated $\mathbb{P}_1$ finite element space

$$\mathcal{V}_N \coloneqq \{v \in \mathscr{C}^0(\mathbb{R}) \ : \ \forall 0 \leqslant s \leqslant N-1, \ v|_{[x_s, x_{s+1})} \in \mathbb{P}^1([x_s, x_{s+1}])\}$$

with

$$\mathbb{P}^1([x_s, x_{s+1}]) \coloneqq \{v : [x_s, \ x_{s+1}] \to \mathbb{R}, \ v(x) = ax + b, \ (a, b) \in \mathbb{R}^2\}.$$

We then search for a solution $u_N \in \mathcal{V}_N \subset H^1(\Omega)$ by Galerkin projection, that is, we search for $u_N \in \mathcal{V}_N$ such that

$$a(u_N, v) = l(v), \quad \forall v \in \mathcal{V}_N.$$

We next take as a basis of $\mathcal{V}_N$ the set of tent functions defined as

$$\varphi_i(x_j) = \delta_{ij}, \ \text{for } 1 \leqslant i, j \leqslant N,$$

and we express the solution as $u_N = \sum_{i=1}^N c_i \varphi_i$. Gathering the expansion coefficients in the vector $c = (c_i)_{i=1}^N$, and injecting the expansion of u_N in the variational formulation, we are led to the system of equations

$$Mc = q$$

where

$$M = (M_{i,j})_{1\leqslant i,j\leqslant N}, \quad M_{i,j} \coloneqq a(\varphi_i, \varphi_j),$$
$$q = (q_i)_{i=1}^N, \quad q_i \coloneqq l(\varphi_i).$$

6.4 Numerical Results

6.4.1 Test case and comparison criteria

In this section we show the results obtained by approximating the exact solution of the problem described in (6.1) using the methods introduced above.

Here, we work in the case when $d = 1$, $\Omega = (0,1)$, and F, f are assumed to be equal to some constant real numbers. Then, the solution of (6.1) and (6.2) has an analytic expression which is given hereafter. Let us also introduce $g_0, g_1 \in \mathbb{R}$ so that $g(0) = g_0$ and $g(1) = g_1$. The problem then reads as follows: find $u : (0,1) \to \mathbb{R}$ solution to

$$\begin{cases} -\varepsilon u''(x) + Fu'(x) = f, & \forall x \in (0,1), \\ -\alpha u'(0) + \kappa u(0) = g_0, \\ \alpha u'(1) + \kappa u(1) = g_1. \end{cases} \tag{6.30}$$

Then, it can be easily checked that the solution to this equation reads as

$$u(x) = C_1 + C_2 e^{\frac{Fx}{\varepsilon}} + \frac{f}{F}x$$

where C_1 and C_2 are constants that are determined with the Robin boundary conditions. They satisfy the system

$$\begin{pmatrix} \kappa & \kappa - \frac{\alpha F}{\varepsilon} \\ \kappa & \kappa e^{\frac{F}{\varepsilon}} + \alpha \frac{F}{\varepsilon} e^{\frac{F}{\varepsilon}} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} g_0 + \alpha \frac{f}{F} \\ g_1 - \frac{f}{F}(\kappa + \alpha) \end{pmatrix}$$

which is invertible except for

$$\kappa = 0, \quad \text{or} \quad \frac{\alpha}{\kappa} = \frac{\varepsilon(1 - e^{\frac{F}{\varepsilon}})}{F(1 + e^{\frac{F}{\varepsilon}})}.$$

In the following, the values of κ and α are always chosen so that the above system is invertible.

In the numerical tests presented below, we fix $F = 1$, $f = 1$. We choose Robin boundary conditions that mimic Dirichlet conditions and we set $\alpha = 10^{-3}$, $\kappa = 1$, $g_0 = g_1 = 0$. Note that we cannot take $\alpha = 0$ since all variational methods are not well defined for pure

Dirichlet boundary conditions. With these choices, the equation reads

$$\begin{cases} -\varepsilon u''(x) + u'(x) = 1, & \forall x \in (0,1), \\ -10^{-3}u'(0) + u(0) = 0, \\ 10^{-3}u'(1) + u(1) = 0. \end{cases} \tag{6.31}$$

Since the exact solution has an analytic form, we can thus easily compare the approximation quality of the output functions $\hat{u}$ from our methods by computing a discrete version of their $L^2(\Omega)$ error norm with respect to the exact solution:

$$e_{L^2}^2 := \|u - \hat{u}\|_{L^2(\Omega)}^2 \approx \frac{1}{\tilde{K}} \sum_{k=0}^{\tilde{K}-1} (u(x_k) - \hat{u}(x_k))^2 =: e_{\ell^2}^2.$$

The points x_k are sampled uniformly as defined in Section 6.3.3. We use 10 times more points than the ones used for approximating the integral, so $\tilde{K} = 10K$. Similarly, we also compute the error with respect to the $H^1(\Omega)$ semi-norm:

$$e_{H^1}^2 := \|u' - \hat{u}'\|_{L^2(\Omega)}^2 \approx \frac{1}{\tilde{K}} \sum_{k=0}^{\tilde{K}-1} (u'(x_k) - \hat{u}'(x_k))^2 =: e_{h^1}^2.$$

Note that one can obtain the H^1 error by adding the above error components.

We study the impact on the errors of the following parameters:

- The values of ε. They range from $5 \cdot 10^{-3}$ to 10.0 with a logarithmic spacing.
- The number K of training points (or collocation points). We consider $K = 10$, 10^2, 10^3, 10^4.
- The choice of the sampling method for the training points (uniformly spaced or uniformly random, labelled as $-u$ and $-r$).
- The impact of the machine precision (Float16, Float32, Float64).

Due to the randomness in the initialization of weights on the neural networks, for each combination of parameters (ε, K, sampling type, and machine precision), we perform 10 repetitions with different initializations. Since we didn't notice a big difference between the l^2 error and the h^1 error, we keep just the second one for clarity and put in Section 6.A the plots in l^2 error.

6.4.2 Our code and practical implementation details

All our neural network based numerical tests were performed in Python 3.6 and using the TensorFlow 1.13.1 library [1]. The code provided in the original paper on PINNs [99] was used as the starting point for our own code developments, and we have followed similar guidelines to generalize and enlarge it where needed. In the same way, for each numerical method, derivatives of functions $v \in \mathcal{K}$ are computed using automatic differentiation. The

numerical optimization procedure used in order to compute an approximation of θ^* a minimizer of (6.26) is given by the quasi-Newton L-BFGS algorithm [59]. The code used to generate the examples shown here is available at

https://github.com/agussomacal/ConDiPINN

The interested reader can reproduce our results and test the impact of the variations of certain parameters such as ε, K, the sampling method, and the machine precision.

6.4.3 Discussion

6.4.3.1 Impact of the number K of training points

In this section we discuss the impact of the number K of training points. We fix the machine precision to Float32, and the uniform sampling $-u$.

Figure 6.1 shows the best result obtained in the tests, i.e., the minimum value of the h^1 norm obtained in the 10 different simulations, plotted against the values of ε. In Figure 6.2, we fix $\epsilon = 10$, and plot statistics on the accuracy e_{h^1} (left plot) and computation runtimes for different K (right plot), and for the different methods.

From these figures, we first notice that the approximation of FEM degrades when ε decreases. However, the accuracy improves when the number of discretization points increases (see, e.g., Figure 6.2 - left plot). The rate of improvement is linear as we can see from the right plot in Figure 6.2, as expected. In addition, when looking at the runtimes (Figure 6.2 - right) we observe the expected linear increase with respect to the number K of discretization points.

We can next study the behavior of Vanilla PINN and compare to FEM. We observe that it performs at an almost constant accuracy for any number of training points until around $\varepsilon = 0.027$ where it stops producing reliable approximations (see Figure 6.1). One remarkable observation is that the Vanilla PINN error for large values of ε and small number of training points $K = 10$ is comparable to the FEM errors with a much larger number of degrees of freedom $K > 10^3$ (see left plot in Figure 6.2). However, we observe that the running times of FEM computations remain much lower than the ones of PINN-based methods (right plot in Figure 6.2). Note that the low runtime of the FEM approach is due to the fact that the associated resulting linear system is tridiagonal, and this allows to solve with a linear cost w.r.t. the number of degrees of freedom.

We next comment on the other PINN-based variational methods. For ε large enough, we observe that all the variational based methods follow the same error trend as FEM both with respect to ε and K and for $K < 10^4$ they even perform marginally better. With respect to the computing time, all the methods perform with almost constant time with respect to K and similarly to a FEM method with $K = 100$ degrees of freedom. However, for $\varepsilon < 0.63$, the methods Wz, $Wz - e$ and Vz blow up and lose completely their approximation capabilities. We conjecture that this is due to the fact that the neural network is used to approximate the solution z from the transformed problem, and there is an exponential term to go back from z to u (see (6.7)). This may lead to machine precision overflows (in the exponential computation) and underflows (the neural network has to learn very small

values of z which also are in the limits of precision). To address this issue, we have explored two possible strategies: one was by directly minimizing over u while maintaining the weak formulation which accounts for the method W. The second approach is to perform the re-scaling of the domain RWz. In both cases the blow up caused by the exponential is solved although the re-scaling method RWz doesn't perform as good as others in the region with large ε values.

We finish this section by plotting in Figure 6.3 the best approximated solution for each model, and different values of ε. The interested reader may experiment other configurations in our provided code. The most striking observation is that only FEM and the vanilla PINN method recover the final shape of the exact solution when ε is small. The other variational PINN methods fail despite that some of them exhibit comparable values to FEM in the generalization errors as Figure 6.1 illustrates. This observations suggests that perhaps other types of error metrics should be introduced in order to be able to better distinguish between "good solution shapes" and "bad ones".

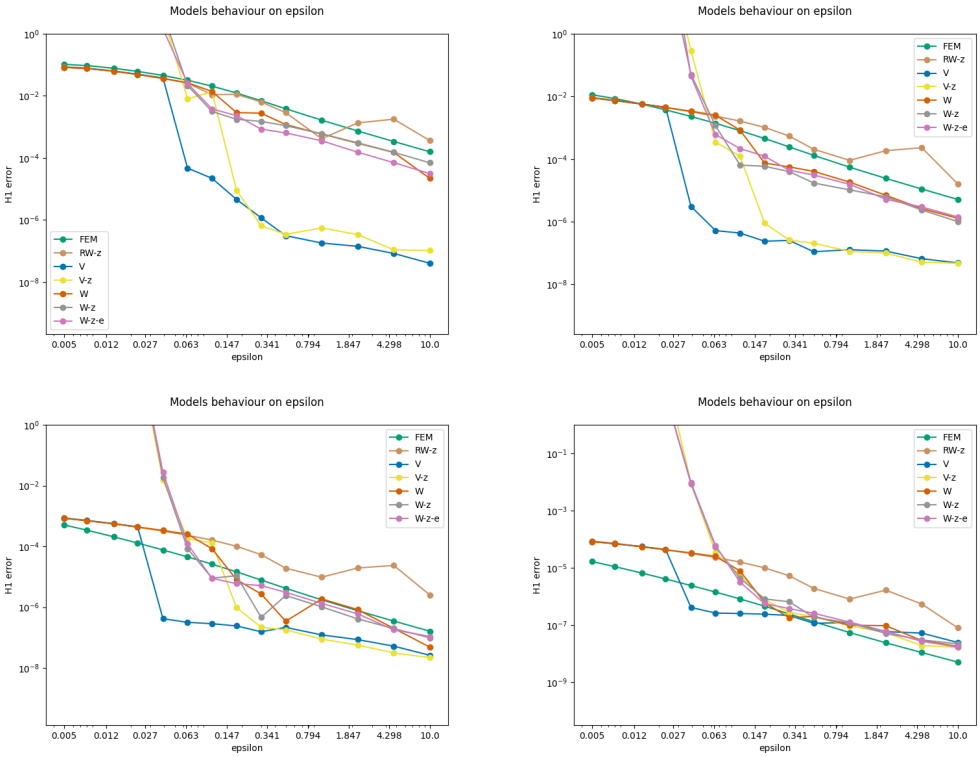


Figure 6.1: Comparison of the behavior of the h^1 error for the different methods and different number of sampling points in training. From top to bottom and from left to right, the first figure is produced for $K = 10$, the second for $K = 100$, the third for $K = 1000$ and the last one for $K = 10000$. The set of points to train and test have been chosen with the *uniform* sampling method. The precision has been fixed to Float32 for all the tests.

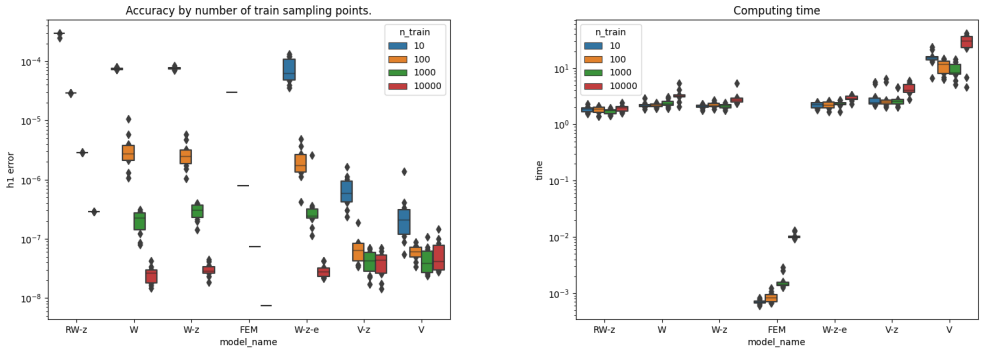


Figure 6.2: For $\varepsilon = 10$ (region where all methods work well), we look at the comparison between methods and the difference with respect to the number of training points K. The h^1 error (left) and the computation times (right). The set of points to train and test have been chosen with the *uniform* sampling method. The precision has been chosen as Float32 for all the tests.

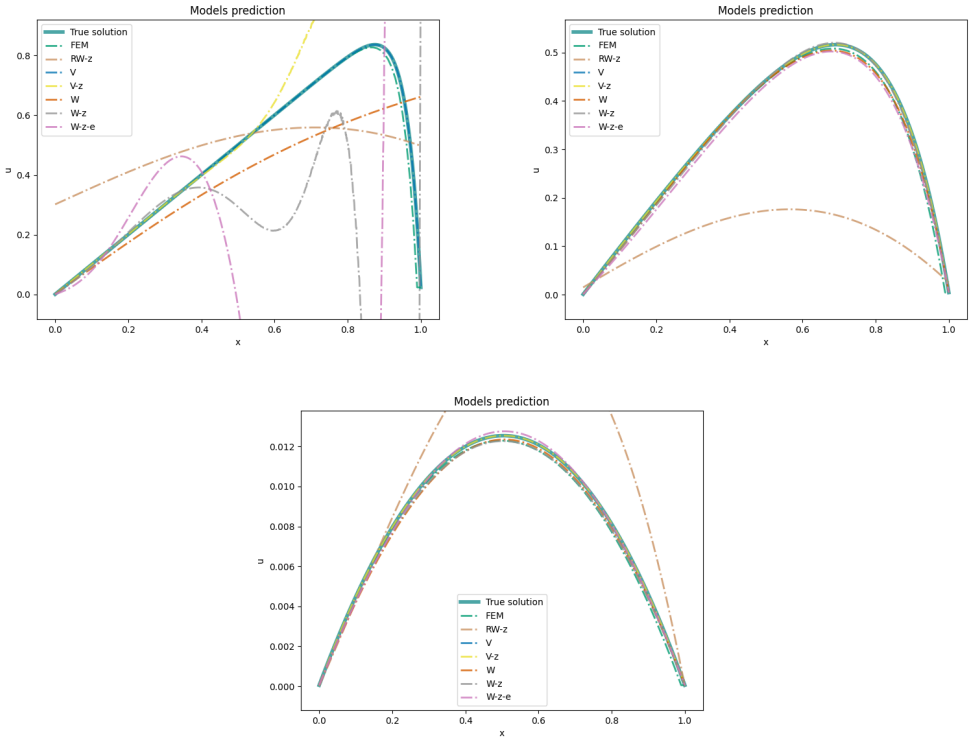


Figure 6.3: The best approximated solution out of 10 repetitions, for each model, and with $K = 100$ training samples. From left to right: $\varepsilon = 0.039, 0.18, 10$. The interested reader may experiment other configurations in our provided code.

6.4.3.2 Impact of Machine Precision

Figure 6.4 shows the h^1-error of the different approximated solution by changing the machine precision in the parameters of the neural networks for the different values of ε: Float16, Float32 and Float64. There is an improvement when going from Float16 to Float32 in all methods. Interestingly, we did not obtain very satisfactory results when working with Float64 precision. This precision seems to difficult the convergence to good quality minima: even after 10 repetitions, we failed to find good results. However, as the plots show, when a good minimum is found, it delivers slightly better approximation than lower machine precisions. For these reasons we have performed our experiments using the Float32 which seemed the most stable choice.

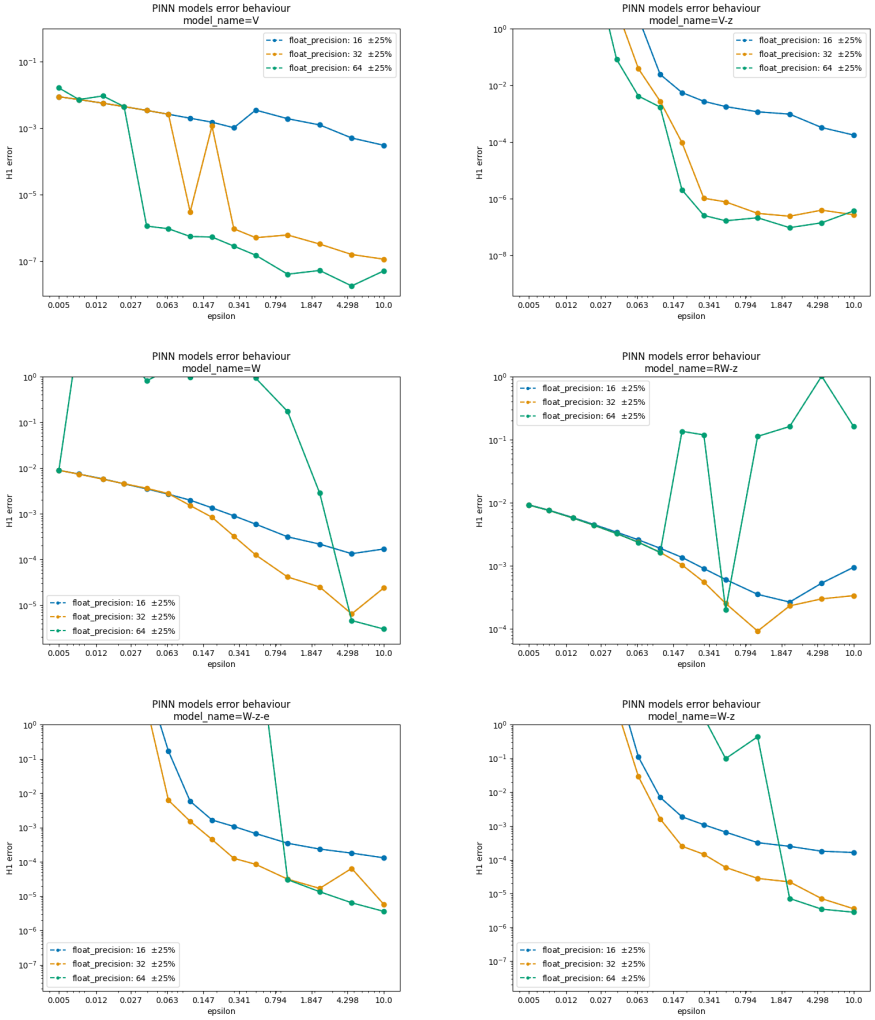


Figure 6.4: Here, the comparison of the behavior of the model for different float precision. The tests have been performed for $K = 100$ and uniform sampling.

6.4.3.3 Impact of Sampling Strategy

Figure 6.5 shows the h^1-error of the different approximated solution by changing the sampling strategy. For all models, the $uniform$ strategy is found to be either as good as the $random$ or slightly better. For this reason we performed all the experiments using the $uniform$ strategy.

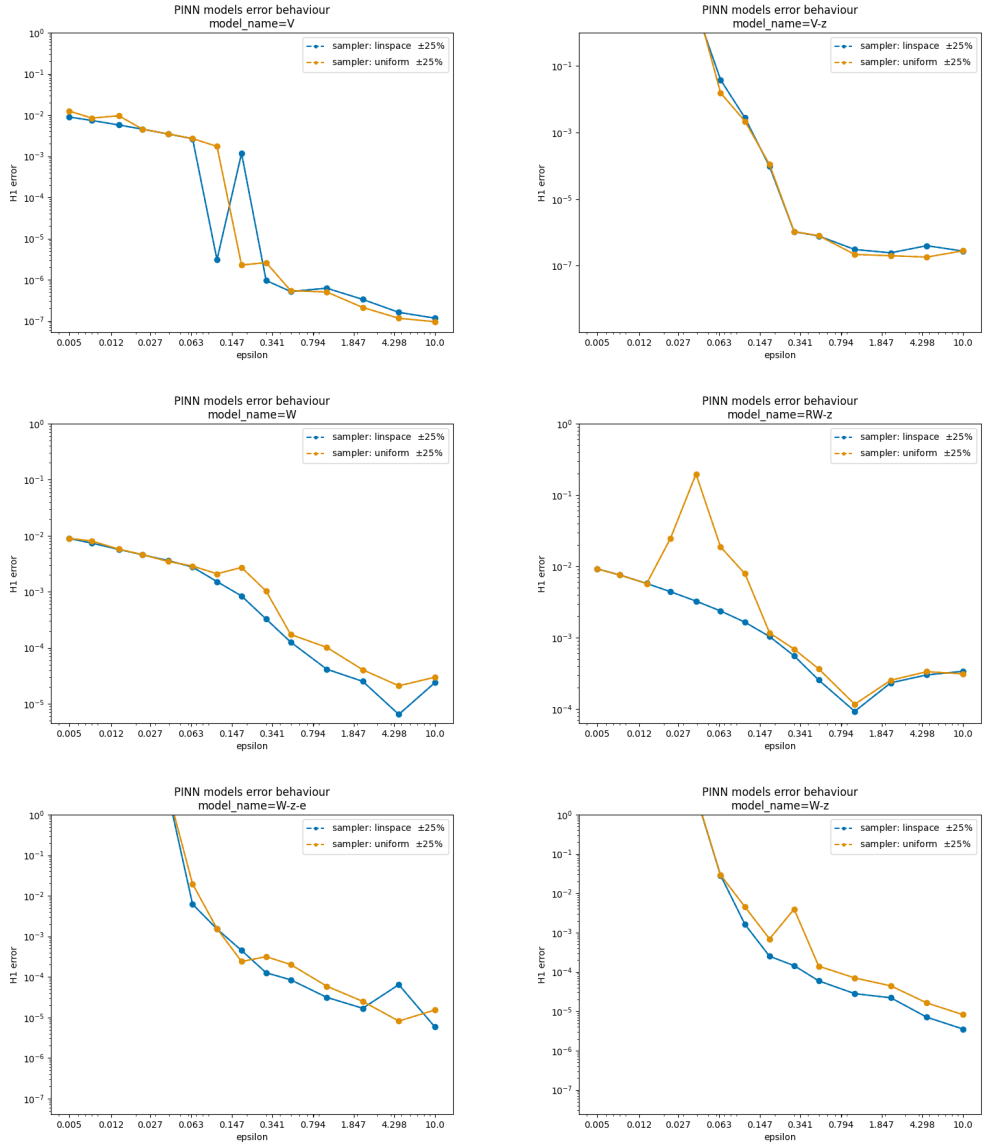


Figure 6.5: Here, the comparison of the behavior of the model for the two different sampling strategies. The tests have been performed for $K = 100$ and the float precision equal to Float32.

6.4.4 Conclusions from the numerical experiments

The above numerical experiments depict a contrasted landscape concerning the merits and limitations of deep learning-based approaches when the solutions become low regular:

- For large values of ε when solutions are rather regular, some PINNs perform clearly better than FEM regarding the generalization errors. The superiority is particularly remarkable for very small number K of training points. However, the shapes of PINN solutions are sometimes not as satisfactory as the ones given by FEM.
- For the challenging case where ε becomes small and solutions become less regular (which was the main motivation of our study), the accuracy of the variational neural-network methods is essentially comparable or worse to the one given by FEM in terms of generalization errors. Some PINN variational approaches become too unstable and the errors blow up. Only FEM and the vanilla PINN approach seem to be able to recover the correct shape of the exact function. The latter one has however the risk of sometimes falling into local minima with bad shapes.
- The run-times are clearly in favor of FEM, as Figure 6.2 illustrates, provided one uses sparse representations of the system matrices. However, the simplicity of implementation is in favor to all PINN methods.

6.5 Future research directions and extensions

One important point to explore in future works concerns the choice of the loss function for the training, and also the metric to evaluate generalization errors. It will also be interesting to explore if adaptive sampling strategies during the training could help to recover good solutions in a more stable manner. Finally, the impact of the machine precision in some steps involving exponential transformations seems also to be an important obstacle to retrieving stable solutions. It would be interesting to develop strategies that circumvent this issue. All these developments will play a crucial role in order to address higher dimensional problems with similar characteristics as the one considered here.

6.A l2 error plots

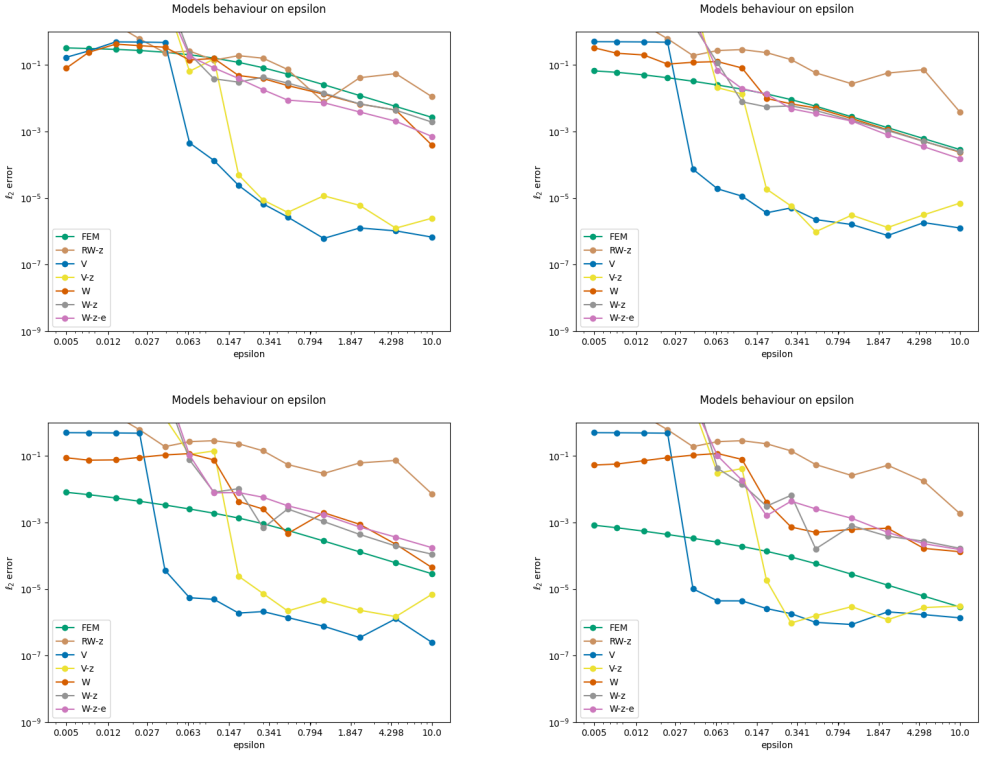


Figure 6.6: The comparison of the behavior of the l^2 error for the different methods and different number of sampling points in training. From up to down and from left to right, the first figure is produced for $K = 10$, the second for $K = 100$, the third for $K = 1000$ and the last one for $K = 10000$. The set of points to train and test have been chosen with the *uniform* sampling method. The precision has been chosen as Float32 for all the tests.

Chapter 7

State estimation of urban air pollution with statistical, physical, and super-learning graph models

7.1 Introduction

7.1.1 Background and motivation

Data-driven estimations are becoming increasingly relevant and widespread as the volume and heterogeneity of available data increases. A fundamental challenge is to build numerical methods for which one can estimate how optimally they exploit the given information. The present paper addresses some essential computational aspects connected to this question. More specifically, our goal is to reconstruct a state u of a physical process, for which we have at hand very heterogeneous sources of data coming from direct partial observations of u, from quantities related to u, and from the knowledge that the physics can be modelled by Partial Differential Equations (PDEs).

Assume that u belongs to some Banach space $\mathcal{U}$ of potentially infinite dimension, with associated norm $\|\cdot\|_{\mathcal{U}}$, and that all the available information is given by an element x_u from some abstract metric space $\mathcal{X}$. Our goal is thus to build a mapping $A : \mathcal{X} \to \mathcal{U}$ such that $A(x_u)$ approximates u at best, in the sense that the approximation error

$$e(A, u) = \|u - A(x_u)\|_{\mathcal{U}} \tag{7.1}$$

is as small as possible, for any configuration (u, x_u) of the system. In practice, finding the optimal map A is not feasible, and various suboptimal reconstruction techniques have been proposed, each of them having its own virtues and drawbacks: statistical approaches such as BLUE [142] and kriging [57], model order reduction of parametric PDEs [92, 115], or more recently approximations by neural networks and machine learning strategies [114, 60]. Since all of these strategies are sub-optimal, and each one is based on different a priori assumptions, one should not make the methods compete against each other, but rather

collaborate with each other to enhance their respective strengths. This leads naturally to explore approaches based on ensemble super-learning [34, 152, 131] as we consider in the present work.

7.1.2 Urban air pollution modelling

There are numerous applications in which one is confronted with the above state estimation problem. As a guiding example, we consider in this paper the real-time reconstruction of urban pollution fields. Beyond the relevance of such a task to limit environmental and health risks in the city, pollution state estimation is an excellent example where collaborative, super-learning methods are required. This is because the problem accumulates several difficulties that make the reconstruction challenging for most common reconstruction methods. Among the issues, we may mention the following:

- *Scarcity of pollution measurements:* The amount of reliable sensor devices measuring pollutant concentrations is often limited, and the measurements are usually taken at fixed locations. As a result, reconstruction methods based solely on these measurements lack spatial resolution, and exhibit huge uncertainties in regions without sensors.

- *Heterogeneous data:* In addition to the pollutant measurements, other sources of relevant information are available such as traffic estimations in each street, wind speed, topography, temperature, etc. However, it is not obvious how to meaningfully combine this data to enhance the estimation. Some attempts have been tried in [58] through the use of Gradient Boosting Machines and Universal Krigging, with positive results for the estimation of PM10 particle levels in the city of Barcelona.

- *Lack of training data:* Even when incorporating other sources of information, the available data may be insufficient, noisy or hardly correlated to the pollution levels we wish to caracterize. Purely data-driven models greatly suffer from these impediments in their training phase.

- *Complexity of the physical problem:* The equations governing the dispersion of pollutants in the atmosphere are nonlinear, with turbulent effects at the street scale, thus imposing a fine spatial resolution, at least near the sensor stations [58]. On the other hand, the computational domain is of the size of a city, making it prohibitively expensive to solve a full model like 3D Navier-Stokes equations.

- *Parameter calibration:* Reduced models use effective parameters, which account for large-scale averages of local effects, in order to alleviate the requirements on the resolution. However these parameters must be calibrated based on the available data or preliminary simulations, which is a hard task given the above issues.

The above obstructions advocate for collaborative strategies combining physics-driven and data-driven approaches such as the one that we develop in this paper. A similar idea has

been explored in [117], but for forecasting temporal series of pollutant, instead of performing state estimation on a large spatial domain.

It should be noted that the limited number of reliable measurements will still pose problems for validating and assessing the quality of each model, which is a crucial part in collaborative strategies. We will mitigate this defect by operating multiple leave-one out cross validations, which preserve as much data as possible for the training part of each model, while testing them on many instances.

7.1.3 Contributions and layout of the paper

Our main contributions are:

1. the construction of numerous physics-based and data-driven models for state estimation;
2. the construction of a very general ensemble super-learning method combining the above models;
3. its application to the task of recovering urban pollution maps at a city scale, together with a comparison of its constitutive submodels;
4. the development of a routine extracting car emissions in each street from traffic maps. Moreover, we have created a dataset comprised of processed traffic data from Google Maps screenshots, which can be used for future research.

In our numerical experiments, we work with the inner city of Paris, which covers a surface of about $140\,\mathrm{km}^2$. The pollutant we consider is NO_2, which is monitored for its respiratory effects, while being mainly produced by vehicle emissions. We use concentration measurements from Airparif sensors[1], and real-time traffic data from Google Maps[2]. Compared to previous contributions and other existing reconstruction methods (see, e.g., [111, 149]), the use of such online traffic data is rather novel. It gives a rough estimation of the spatial density of street traffic, benefits from a very fine spatial resolution, and can be freely updated as frequently as desired, in contrast to many existing approaches which only use time averages of traffic data.

Another distinctive aspect of our approach is the representation of the city by a graph, where nodes and edges correspond to crossroads and street segments, instead of considering an open subset of $\mathbb{R}^2$ or $\mathbb{R}^3$ as the spatial domain. This description immediately includes geometric specificities of the agglomeration under study, such as the orientation of each street or the configuration of each neighborhood. It is a natural framework for taking into account pollutant emissions caused by traffic, which are located on the graph. Moreover, it is in adequacy with our goal of estimating local variations in the concentration of pollutants close to the ground, since the streets are isolated from each other at this height.

[1] We extracted data from the Airparif database, which can be found at https://data-airparif-asso.opendata.arcgis.com

[2] The permission to use Google Maps data for non-profit research is stated here: https://about.google/brand-resource-center/products-and-services/geo-guidelines/#google-maps

Physical models can be solved on such domains thanks to the theory of *quantum graphs*, that is, metric graphs endowed with a differential operator acting on functions defined on the graph (see [23] for details and references). The metric graph structure leads to the definition of suitable and natural function spaces to pose the problem. Of course, several physical models of different complexity could be considered. In this paper, we work with simple elliptic operators, obtained by assuming that the emission and diffusion of pollutant reached a steady state, thus allowing us to treat each time step independently. The model could however be refined by considering, for instance, advection-reaction-diffusion operators in a time-dependent setting.

The rest of the paper is organized as follows. In Section 7.2 we present our guiding numerical example of the Parisian area, and the available data. Section 7.3 explains the different reconstruction methods we have used for our numerical experiments, including ensemble super-learning methods combining the previous ones. By construction, the super-learner has higher approximation power than each individual model. Section 7.4 discusses how to theoretically quantify performance and optimality of the numerical algorithms, and why leave-one-out is a good way to estimate this performance in practice. We summarize our numerical experiments and provide some illustrations in Section 7.5. Finally, Section 7.A details the mathematical setting for the problem of pollution state estimation on graphs.

7.2 Available data and pre-processing

7.2.1 Pollution sensors

The main information we use consists of direct measurements of the NO_2 concentration field u at Airparif sensor stations. There are $m = 13$ such stations, which are placed at fixed locations

$$r^{\mathrm{obs}} \coloneqq \{r_1^{\mathrm{obs}}, \dots, r_m^{\mathrm{obs}}\} \in (\mathbb{R}^2)^m, \tag{7.2}$$

see Figure 7.1. Each of the stations provides hourly averages of the concentration of nitrogen dioxide, in $\mu\mathrm{g}/\mathrm{m}^3$, of the form

$$z_i = u(r_i^{\mathrm{obs}}) + \eta_i, \quad i = 1, \dots, m,$$

where η_i is some noise in the measurements, with nominal relative error $|\eta_i|/u(r_i^{\mathrm{obs}}) \leqslant 15\%$.

Note that, in principle, the equations governing the dispersion of pollutants are time dependent. However, as we only have measurements every hour, we opt for a static model, where the state at a given time is computed based on the data available at this time only. This essentially amounts to assuming that the emissions vary slowly over time, and that the system reaches an equilibrium state in less than one hour.

7.2.2 Meteorological conditions

Wind, as well as stratification effects in the atmosphere due to variations in temperature, play a major role in the dispersion of pollutants [41]. Moreover, the chemical equilibrium

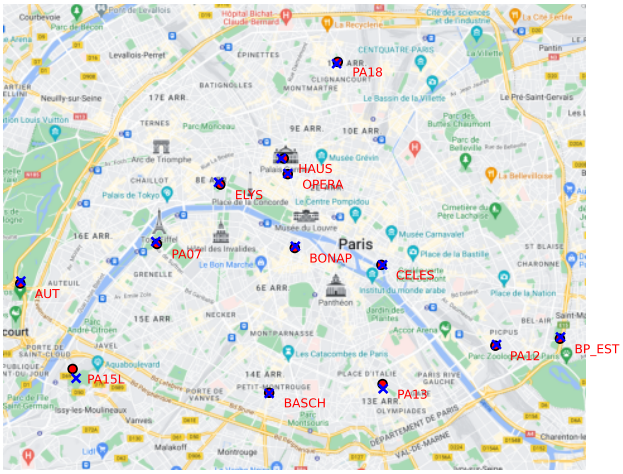


Figure 7.1: Cropped Google Map screenshot of Paris and the $m = 13$ available stations in the study: red dots represent the projection of the station locations to the nearest vertex in the graph of streets, while the blue crosses correspond to the exact position of the station.

between NO and NO_2 depends on the cloud cover [105]. Therefore, we collect the temperature $\theta \in \mathbb{R}$ and the wind speed $w \in \mathbb{R}^2$ at every hour from a weather archive [3]. These two quantities are measured at one point above the Seine river, near the Eiffel tower, and treated as global, that is, they are assumed to be constant over the spatial domain, apart from possible local effects at the level of each street.

7.2.3 Traffic

Car traffic is responsible for more than half the emissions of NO_2 in urban environments [102]. There is an increasing number of available sources that give access to traffic data. In our case, we work with traffic information extracted from Google Maps. We have designed a script using the Python library Selenium to automatically take screenshots of Paris every 15 minutes over an area of 1253×1253 pixels with zoom level 13. An example of resulting *raw image* can be seen in Figure 7.2. Note that city landmarks could not be removed before taking the screenshot, nor even by substracting a background image, since each screenshot has slight color variations, rendering this approach impractical. Another issue is the absence of traffic data in the smallest streets of the city. In addition, linking it to the pollution field requires some calibration.

On the one hand, this kind of information is very rich because of its availability in real time, and its spatial coverage of the whole city at high resolution. On the other hand, it is

[3] See `https://www.windguru.cz`. For the wind, we combined the absolute wind speed with the wind direction to obtain a vector in $\mathbb{R}^2$.

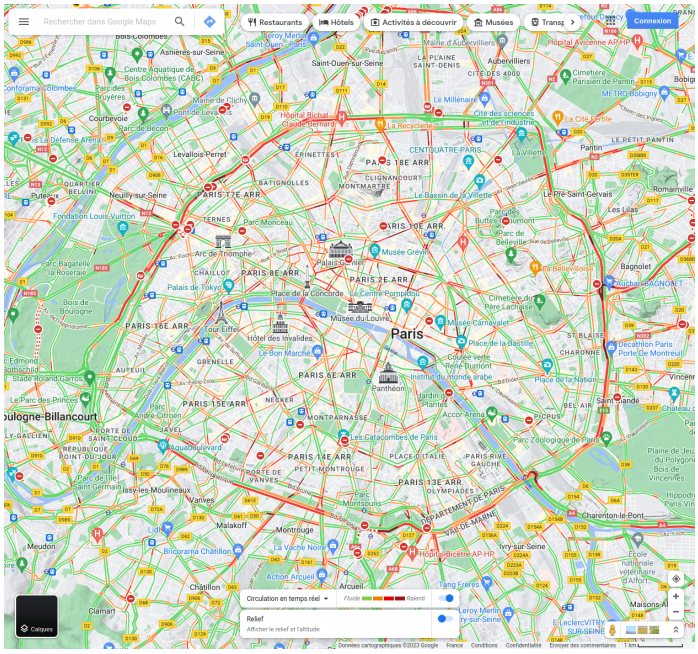


Figure 7.2: Raw data from Google Maps: the image contains the city with its main landmarks, and some streets are highlighted with one of the four colors corresponding to traffic.

very partial: it comes in the form of four colors, each representing a certain traffic intensity, which gives a qualitative estimate of the number of cars in each street: a street marked in red is, for instance, more congested than one marked in green. One main goal in our work is to examine the potential of incorporating such low-quality information for state estimation tasks.

7.2.4 Graph of Paris

In order to locate our sources and to express the spatial dependence of a state, we consider a graph domain with streets as edges and intersections as vertices. We use a metric graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ provided by Open Street Maps, together with the Python library osmnx. For the mathematical definition of metric graphs and their associated function spaces, we refer to Section 7.A. The graph G covers the whole inner ring of the city, as shown in Figure 7.3. The full graph has $|\mathsf{V}| = 12963$ vertices and $|\mathsf{E}| = 25476$ edges, but we restrict it to the biggest connected component of the subgraph that remains after filtering out all the edges which have never been colored with traffic information. After this operation, our actual graph has $|\mathsf{V}| = 10116$ vertices and $|\mathsf{E}| = 18713$ edges. The street network is relatively dense, most nodes having 3 to 6 edges.

The vertices $\mathsf{v} \in \mathsf{V}$ come with precise geographical coordinates. In the following, we assume that the graph is embedded in the two-dimensional plane, and do not take altitude into account. Each edge $\mathsf{e} \in \mathsf{E}$ is a street or a portion of it, and we have access to its length ℓ_{e} as well as its shape, number of lanes and speed limit. The information is so detailed

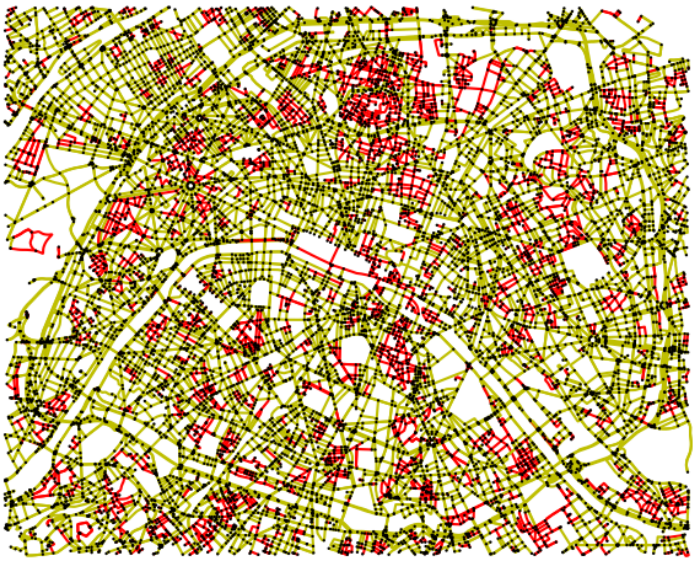


Figure 7.3: The metric graph downloaded from Open Street Maps, with the edges that never had Google Traffic activation in red, and the edges remaining after filtration in yellow.

that the streets are represented by paths that are not necessarily straight lines. However, in the following, we replace every street path by a straight line between its endpoints, for simplicity.

The location of the sensor stations is known, but does not exactly match with vertices of the graph. We therefore project the positions r^{obs} from (7.2) onto the nearest vertices, which yields observational nodes

$$\mathsf{v}_i^{\text{obs}} := \arg\min_{\mathsf{v}\in\mathsf{V}} |r_i^{\text{obs}} - \mathsf{v}|, \quad i = 1, \dots, m.$$

As Figure 7.1 illustrates, the projected locations are very close to the exact locations, with a maximal discrepancy of 165m, to be compared with the width of the domain, of about 12km. As a consequence, we will assume that the observations z_i correspond to the values $u(\mathsf{v}_i^{\text{obs}})$, up to a slight increase in the measurement errors, since η_i is replaced by $\eta_i + u(r_i^{\text{obs}}) - u(\mathsf{v}_i^{\text{obs}})$.

7.2.5 Pre-processing of traffic data

We also map the traffic information onto pollutant emissions on the graph edges, by implementing the following pipeline:

- **Cropping:** Starting from a raw image like Figure 7.2, we first crop it to the shape 800×1000, in order to eliminate toolbars and adapt it to the size of the graph. The background of Figure 7.1 is obtained by the same procedure.

- **Traffic colors extraction:** The colors associated with the four levels of traffic

$$\texttt{colors} := \{\texttt{green}, \texttt{orange}, \texttt{red}, \texttt{dark-red}\}$$

  are easily identified[4]. They seem to be used exclusively for that purpose, hence it suffices to extract the pixels having one of these colors.

- **Projection on graph edges:** These pixels, once expressed in their geographical coordinates, almost perfectly overlap the metric graph from Open Street Maps. For each edge $\mathsf{e} \in \mathsf{E}$ and each color $\mathtt{c} \in \texttt{colors}$, we count the number $p_{\mathtt{c}}^{\mathsf{e}}$ of pixels of color $\mathtt{c}$ that are closest to edge e. Note that the traffic color might change along an edge, in which case we give up on some local information by only considering the total traffic on the edge.

- **Edge normalization:** We then transform these pixel counts into proportions of traffic colors on each edge. As the edges may remain blank at times where there is no traffic, we take as a normalizing constant the maximal amount of pixels encountered over all times T for which we collect traffic data:

$$q_{\mathtt{c}}^{\mathsf{e}}(t) = \frac{p_{\mathtt{c}}^{\mathsf{e}}(t)}{\max_{t' \in T} \sum_{\mathtt{c}' \in \texttt{colors}} p_{\mathtt{c}'}^{\mathsf{e}}(t')}, \quad t \in T.$$

  In this way, $q_{\mathtt{c}}^{\mathsf{e}}(t) \in [0,1]$ indicates the proportion of edge e colored with $\mathtt{c}$ at time t, but remains null if no traffic is reported.

- **Hourly averaging:** As the pollution information is only available every hour, we take the average of the four values of $q_{\mathtt{c}}^{\mathsf{e}}$ encountered every fifteen minutes, which we still denote $q_{\mathtt{c}}^{\mathsf{e}}$ in the sequel.

- **Projection on graph nodes:** In our models, it is in fact simpler to localize emissions on the nodes of the graph. For this reason, we calculate the density of each traffic color $\mathtt{c}$ around a vertex $\mathsf{v} \in \mathsf{V}$ as a weighted average on its neighboring edges $\mathsf{E}(\mathsf{v})$

$$q_{\mathtt{c}}^{\mathsf{v}}(t) = \frac{\sum_{\mathsf{e} \in \mathsf{E}(\mathsf{v})} a_{\mathsf{e}} q_{\mathtt{c}}^{\mathsf{e}}(t)}{\sum_{\mathsf{e} \in \mathsf{E}(\mathsf{v})} a_{\mathsf{e}}},$$

[4]The RGB value of each color is given by: $\texttt{green} = (99, 214, 104)$, $\texttt{orange} = (255, 151, 77)$, $\texttt{red} = (242, 60, 50)$, $\texttt{dark-red} = (129, 31, 31)$

where a_{e} stands for the area of the road associated to edge e, given by the product of its length ℓ_{e} with the number of lanes.

7.2.6 Summary

While the history of sensor and weather data can be found on archives, our script for capturing traffic images only runs in real time, since Google Traffic only provides current information. We collected all types of data on an hourly basis for a period of time comprised between December 9, 2022 and March 19, 2023. After removing time stamps for which some data was missing, we end up with a set of acquisition times T, of cardinality $|T| = 1712$, which we divide into a set T_{train} of 1338 training times, and a set T_{test} of 374 testing times.

In the end, given the graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, the available information at any time $t \in T$ is of the form

$$x = (\mathsf{v}^{\text{obs}}, z, \theta, w, (q_{\mathsf{c}}^{\mathsf{v}})_{\mathsf{c},\mathsf{v}}) \in \mathcal{X} = \mathsf{V}^m \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^{4|\mathsf{V}|}. \tag{7.3}$$

In the next section, we present various models to estimate the pollution field from this data, using either statistical inference, linear mappings based on expert knowledge, or neural networks.

7.3 Reconstruction methods

We have implemented several methods of state estimation by leveraging the different information sources. Our methods give reconstructions on the metric graph G, that is, we consider mappings $A : \mathcal{X} \to \mathcal{U}$ where $\mathcal{U}$ is a space of functions defined on G. Typical examples are $\mathcal{U} = \mathcal{C}(\mathsf{G})$, $L^2(\mathsf{G})$ or $H^1(\mathsf{G})$, as defined in Section 7.A. As our main interest is in assessing the effect of incorporating indirect information like the real-time traffic data, we first consider models that take only a portion of $x \in \mathcal{X}$ as input.

7.3.1 Spatial average

If we give up on all the spatially-dependent data v^{obs} and $(q_{\mathsf{c}}^{\mathsf{e}})_{\mathsf{c},\mathsf{e}}$, the reconstruction is necessarily constant over the whole domain G, which yields no better choice than the average of the observed concentration values

$$A_{\text{avg}}(x)(r) = \bar{z} = \frac{1}{m}\sum_{i=1}^{m} z_i, \quad \forall r \in \mathsf{G}.$$

This extremely simple reconstruction will serve as our baseline to compare more sophisticated reconstructions. In the sequel, we will add the spatially-dependent data and view the other models as corrections to the spatial average above. This will result in spatially unbiased estimators, provided that the locations of the sensors are representative of the whole pollution field. More precisely, assuming that the stations are randomly drawn according to the uniform probability distribution μ on G, the expectation over $z_1, \dots, z_m$ of

the spatially-averaged error is

$$\mathbb{E}_z\left(\int_{\mathsf{G}}\big(u(r)-A_{\mathrm{avg}}(x)(r)\big)\,d\mu(r)\right)=\int_{\mathsf{G}}u\,d\mu-\mathbb{E}_z\left(\frac{1}{m}\sum_{i=1}^{m}z_i\right)=0,$$

and this remains true when adding to A_{avg} a correction of vanishing expectation.

7.3.2 Best unbiased linear estimator

If we only want to estimate a missing measurement z_i at a given station $i\in\{1,\dots,m\}$, we may also use statistical information stemming from the history (z_i^t) of the station at previous times t, as well as the observations from other stations $j\neq i$ in the present and the past, denoted respectively z_j and (z_j^t). For T_{train} the set of training times, define the empirical average

$$\langle z_i\rangle:=\frac{1}{|T_{\mathrm{train}}|}\sum_{t\in T_{\mathrm{train}}}z_i^t$$

and empirical covariance matrix $K\in\mathbb{R}^{m\times m}$ with entries

$$K_{i,j}:=\Big\langle\big(z_i-\langle z_i\rangle\big)\big(z_j-\langle z_j\rangle\big)\Big\rangle=\langle z_iz_j\rangle-\langle z_i\rangle\langle z_j\rangle.$$

Any unbiased linear estimator $\tilde{z}_i$ of z_i is of the form

$$\tilde{z}_i=\langle z_i\rangle+\sum_{j\neq i}c_j\big(z_j-\langle z_j\rangle\big),$$

for some coefficients $(c_j)_{j\neq i}$. Let $c\in\mathbb{R}^m$ be the vector with coordinates c_j for $j\neq i$ and $c_i=-1$. Then the best linear unbiased estimator (BLUE) is obtained by optimizing the averaged squared error

$$\operatorname*{arg\,min}_{(c_j)_{j\neq i}}\Big\langle\big(\tilde{z}_i-z_i\big)^2\Big\rangle=\operatorname*{arg\,min}_{(c_j)_{j\neq i}}c^{\top}Kc=\big[(K_{j,k})_{j,k\neq i}\big]^{-1}(K_{j,i})_{j\neq i},$$

where $(K_{j,k})_{j,k\neq i}$ and $(K_{j,i})_{j\neq i}$ are seen as a matrix in $\mathbb{R}^{(m-1)\times(m-1)}$ and a vector in $\mathbb{R}^{m-1}$.

If the set of training times T_{train} is large enough, we expect an ergodicity property of the form $\langle z_i\rangle\approx\mathsf{E}(z_i)$ to hold. For this reason, BLUE should be a near minimizer of the expected squared error, given the available data. Therefore, in the numerical experiments, we will evaluate the different methods A by comparing $A(x\setminus\{z_i\})(z_i)$ and z_i, and the error $|\tilde{z}_i-z_i|^2$ will act as an optimality benchmark.

It should be emphasized that BLUE itself is not a valid reconstruction method, since it requires statistical information which is accessible only at the locations of the stations $\mathsf{v}_i^{\mathrm{obs}}$. Hence this estimator cannot be computed at any point $r\in\mathsf{G}$ of the graph domain.

7.3.3 Kriging

In order to transform BLUE into a reconstruction method, one needs to propose a surrogate for the correlation between any two points in the graph. Moreover, as we don't know the average pollution at all points of the graph, we proceed without subtracting spatial averages $\langle z_i \rangle$ in this subsection, in contrast to the previous one. Therefore, we consider the Gram matrix of normalized second-order moments

$$G_{i,j} = \frac{\langle z_i z_j \rangle}{\sqrt{\langle z_i^2 \rangle \langle z_j^2 \rangle}}.$$

Taking the positions $\mathsf{v}^{\mathrm{obs}}$ of the stations into account, we observe that each entry $G_{i,j}$ partly depends on the distance $|\mathsf{v}_i^{\mathrm{obs}} - \mathsf{v}_j^{\mathrm{obs}}|$ between the stations, see Figure 7.4. A typical choice of approximant is the Gaussian kernel

$$G_{i,j} \approx \hat{G}_{i,j} := C \exp\left(-\frac{|\mathsf{v}_i^{\mathrm{obs}} - \mathsf{v}_j^{\mathrm{obs}}|^2}{2\sigma^2} \right) + (1 - C)\delta_{i,j},$$

with parameter values $C = 0.968$ and $\sigma = 33.4\mathrm{km}$ obtained by fitting the station data in our case.

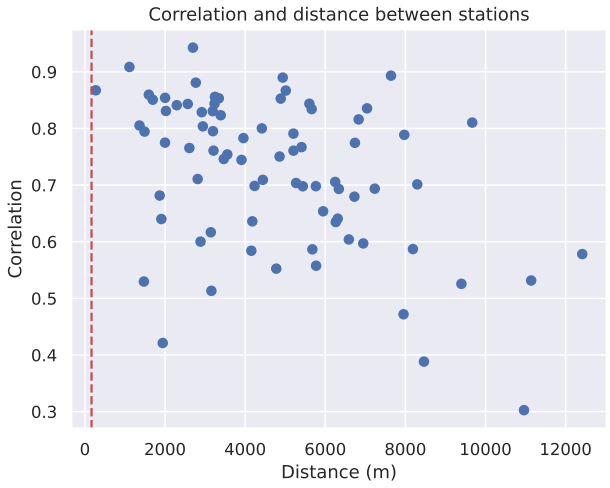


Figure 7.4: Correlation between stations as a function of the distance. The vertical slashed red line marks the maximal separation between vertex and station (165m) which still lays in the zone of high correlation.

Remark 7.3.1. *The fact that $C < 1$ can be interpreted as the presence of random noise η_i on the measurements $z_i = u(\mathsf{v}_i^{obs}) + \eta_i$. As a safety check, one may notice that the average relative error*

$$\frac{\langle \eta_i^2 \rangle}{\langle u(\mathsf{v}_i^{obs})^2 \rangle} = \frac{\langle z_i^2 \rangle}{\langle u(\mathsf{v}_i^{obs})^2 \rangle} - 1 \approx \frac{1}{C} - 1 = 3.31\%$$

is effectively much smaller than the uniform error guarantee $\|\eta_i/|u(r_i^{obs})\|_{L^\infty} \leqslant 15\%$ *that we discussed in Section 7.2.1. Adding the matrix* $(1-C)I$ *ensures that* $\hat{G}$ *has ones on its diagonal, as expected of a correlation matrix, and regularizes the system, by making the inversion of* $\hat{G}$ *stable. More practically, the reconstructed value* $A(x)(v_i^{obs})$ *will not be exactly* z_i*, but rather an average of the measurements close to* v_i^{obs}*.*

Let $r \in \mathsf{G}$, we again examine a linear model

$$A_{\text{krig}}(x)(r) = \sum_{i=1}^{m} c_i^r z_i,$$

with coefficients $c^r \in \mathbb{R}^m$ to be determined. This estimator is unbiased if and only if $\sum_{i=1}^m c_i^r = 1$, and in that case we can write it as a correction to the temporal or spatial average

$$A_{\text{krig}}(x)(r) = \langle A_{\text{krig}}(x)(r)\rangle + \sum_{i=1}^{m} c_i^r (z_i - \langle z_i\rangle) = \bar{z} + \sum_{i=1}^{m} \left(c_i^r - \frac{1}{m}\right)(z_i - \bar{z}).$$

By analogy with BLUE, we thus define the weights as the renormalized solution of a system of correlation equation

$$c^r = \frac{\hat{c}^r}{\sum_{i=1}^m \hat{c}_i^r}, \qquad \hat{c}^r = \hat{G}^{-1} g^r, \qquad g_i^r = C \exp\left(-\frac{|v_i^{\text{obs}} - r|^2}{2\sigma^2}\right).$$

Although there are no optimality guarantees, we expect kriging to have intermediate performance when compared to the spatial average baseline, and to the ideal BLUE reconstruction. However, due to the important spacing between peripheral stations, we only observe an improvement in the central part of Paris. The insufficient density of pollution measurements calls for models involving other sources of information, such as traffic data. This is the objective of the next two subsections.

7.3.4 Source model

The simplest way to incorporate traffic data consists in using only local values $q_{\mathsf{c}}^{\mathsf{e}}$ for estimating the pollution on an edge $\mathsf{e} \in \mathsf{E}$, or $q_{\mathsf{c}}^{\mathsf{v}}$ for a node $\mathsf{v} \in \mathsf{V}$. As we projected the station locations on V, we focus on the latter case here. We call such a method a source model, since it directly maps the sources of emission to pollution values.

We opt for a linear model acting as a correction on the spatial average baseline:

$$A_{\text{src}}(x)(\mathsf{v}) = \bar{z} + \sum_{\mathsf{c} \in \texttt{colors}} \alpha_{\mathsf{c}} (q_{\mathsf{c}}^{\mathsf{v}} - \bar{q}), \tag{7.4}$$

where we subtracted the spatial average of traffic $\bar{q}$ for unbiasedness. The vector of coeffi-

cients $\alpha \in \mathbb{R}^4$ is found by solving a LASSO problem

$$\min_{\alpha \in \mathbb{R}^4} \sum_{t \in T_{\text{train}}} \sum_{i=1}^{m} |z_i^t - A_{\text{src}}(x(t))(\mathsf{v}_i^{\text{obs}})|^2 + \lambda \|\alpha\|_1,$$

and we perform a cross-validation to estimate the optimal parameter λ, in order to prevent overfitting.

Alternatively, we can also write nonlinear variants, of the form

$$A_{\text{src}}(x)(\mathsf{v}) = \bar{z} + \mathcal{T}_\alpha\left((q_{\mathsf{c}}^{\mathsf{v}}), \theta, w\right), \tag{7.5}$$

which may take into account other sources of information like temperature θ and wind w. Here, $\mathcal{T} : \mathbb{R}^{\#\alpha} \times \mathbb{R}^4 \times \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}$ can be a *polynomial* combination of the inputs $((q_{\mathsf{c}}^{\mathsf{v}}), \theta, w)$, a *neural network*, or any other nonlinear mapping. The set of parameters α is no longer associated to the four traffic colors, but still needs to be learned via a LASSO regression.

All such models rely on the assumption that pollution depends on its sources in a very localized manner. However, the traffic charts $(q_{\mathsf{c}}^{\mathsf{v}})_{\mathsf{v} \in \mathsf{V}}$ exhibit sharp variations from one node to its neighbors, which incites to smooth the emissions before applying the above methods. This is the purpose of the following section, which attempts to model such a diffusive behavior.

7.3.5 Physical modelling

An important inspiration for reconstruction methods comes from the physical modelling of pollution dispersion. In our setting, we resort to building a *quantum graph*, that is, we endow our metric graph G with a differential operator acting on functions from functional spaces such as $L^2(\mathsf{G})$ or $H^1(\mathsf{G})$, as defined in Section 7.A (we also refer to [23] for more details and references). Our approach can be summarized as follows:

Elliptic equation: One can first model pollution with a time-independent elliptic equation, by assuming that all time-dependent parameters have sufficiently slow variations, here over the course of an hour, for the pollution field to reach a steady state. For any point $r \in \mathsf{G}$, the pollutant concentration is modelled by a function $u : \mathsf{G} \to \mathbb{R}$ solution to the diffusion-reaction equation

$$\mathcal{P}(u) := -\frac{\mathrm{d}}{\mathrm{d}r} \cdot \left(a(r)\frac{\mathrm{d}}{\mathrm{d}r}u(t,r)\right) + h(r)u(r) = q(r), \qquad r \in \mathsf{G}, \tag{7.6}$$

which we choose to complement with "Newmann-Kirchoff" conditions on the vertices, that is,

$$\sum_{\mathsf{e} \in \mathsf{E}(\mathsf{v})} \frac{\mathrm{d}u}{\mathrm{d}r}\Big|_{\mathsf{e}}(\mathsf{v}) = 0, \qquad \mathsf{v} \in \mathsf{V}, \tag{7.7}$$

expressing the conservation of the quantity of pollutant at every crossroad $\mathsf{v} \in \mathsf{V}$. Here, $\mathsf{E}(\mathsf{v})$ denotes the edges having v as an endpoint, and the derivatives are assumed to be taken

in the directions away from the vertex.

In (7.6), the function $a \in L^\infty(\mathsf{G})$ is an effective diffusion coefficient, which takes into account turbulent dissipative effects. The absorption coefficient $h \in L^\infty(\mathsf{G})$ models the leakage of pollutants from the streets to the higher atmosphere, as well as chemical reaction, in particular between NO_2 and other nitrogen oxides, which are not measured by the sensors. Lastly, the source term $q \in L^2(\mathsf{G})$ models all possible emissions of pollutant, which in the case of Paris essentially come from traffic, local heating, and industrial and urban activity outside the city. As we only have access to local traffic data, we assume that the other sources are spatially constant, and average them out by solving $\mathcal{P}(u - \bar{u}) = q - \bar{q}$, where $\bar{u}$ is the spatial average of the pollutant concentration, estimated by $A_{\text{avg}}(x) = \bar{z}$, and $q - \bar{q}$ corresponds to the variations of traffic around its spatial average, computed through the procedure from Sections 7.2.5 and 7.3.4.

Remark 7.3.2. *Equation (7.7) has a similar effect as Newmann conditions at the borders of the spatial region under consideration, here the rectangle contour of Figure 7.1. Therefore the only exchanges with the exterior of this region are contained in the source term q. As this no-flux condition only give a very rough approximation of the solution close to the border, in the numerical experiments of Section 7.5, we will concentrate on the accurate prediction of the pollution in the central part of the city.*

Variational formulation: The operator $\mathcal{P}(u)$ in (7.6) is defined for functions $u \in H^2(\mathsf{G})$, but the equation can be stated in a weak form, which only requires that $u \in H^1(\mathsf{G})$. Multiplying (7.6) by a sufficiently smooth test function $v \in H^1(\mathsf{G})$, and using the Kirchoff-Neumann boundary conditions, it follows that the corresponding weak formulation of the problem is to find $u \in H^1(\mathsf{G})$ such that

$$\mathfrak{b}(u, v) = \mathfrak{f}(v), \qquad v \in H^1(\mathsf{G}) \tag{7.8}$$

where $\mathfrak{b}$ is the symmetric bilinear form defined as

$$\begin{array}{rccl} & H^1(\mathsf{G})^2 & \to & \mathbb{R} \\ \mathfrak{b}: & (u,v) & \mapsto & \sum_{\mathsf{e}\in\mathsf{E}} \left\{ \int_{\mathsf{e}} a(r) \dfrac{\mathrm{d}u}{\mathrm{d}r}(r) \dfrac{\mathrm{d}v}{\mathrm{d}r}(r) \mathrm{d}r + \int_{\mathsf{e}} h(r) u(r) v(r) \mathrm{d}r \right\} \end{array}$$

and $\mathfrak{f} : v \in H^1(\mathsf{G}) \mapsto \sum_{\mathsf{e}\in\mathsf{E}} \int_{\mathsf{e}} q(r)v(r)\mathrm{d}r$ is a continuous linear form.

Assuming that $a(r) \geqslant a_0 > 0$ and $h(r) \geqslant h_0 > 0$ for $r \in \mathsf{G}$ a.e., we see that $\mathfrak{b}$ is continuous and coercive in $H^1(\mathsf{G})$ with coercivity constant $\min(a_0, h_0)$, and continuity constant $\max(\|a\|_{L^\infty(\mathsf{G})}, \|h\|_{L^\infty(\mathsf{G})})$. By the Lax-Milgram theorem, problem (7.8) admits a unique solution $u \in H^1(\mathsf{G})$.

Discretization: In our numerical tests, we discretize the equation with $\mathbb{P}_1$ finite elements, that is, continuous functions whose restriction to any edge is affine. We describe below the main guidelines, and refer to [10] for further details and a complete analysis.

We define the set of hat functions $\{\varphi_{\mathsf{v}}\}_{\mathsf{v}\in\mathsf{V}}$ by $\varphi_{\mathsf{v}}(\mathsf{v}') = \delta_{\mathsf{v},\mathsf{v}'}$ for any vertices $\mathsf{v},\mathsf{v}' \in \mathsf{V}$, and

$$\forall x_{\mathsf{e}} \in [0,\ell_{\mathsf{e}}], \quad \varphi_{\mathsf{v}}(x_{\mathsf{e}}) = \begin{cases} 1 - \frac{x_{\mathsf{e}}}{\ell_{\mathsf{e}}}, & \text{if } \mathsf{e} \in \mathsf{E}(\mathsf{v}), \\ 0, & \text{if } \mathsf{e} \notin \mathsf{E}(\mathsf{v}), \end{cases}$$

for any edge $\mathsf{e} \in \mathsf{E}$. Fixing our finite element space $\mathbb{P}_1 = \operatorname{span}\{\varphi_{\mathsf{v}}\}_{\mathsf{v}\in\mathsf{V}} \subset H^1(\mathsf{G})$, we search for the Galerkin solution $\hat{u} = \sum_{\mathsf{v}\in\mathsf{V}} c_{\mathsf{v}}\varphi_{\mathsf{v}} \in \mathbb{P}_1$ such that

$$\mathfrak{b}(\hat{u},\hat{v}) = \mathfrak{f}(\hat{v}), \qquad \hat{v} \in \mathbb{P}_1.$$

Gathering the expansion coefficients of the solution in the vector $\mathbf{c} = \{c_{\mathsf{v}}\}_{\mathsf{v}\in\mathsf{V}}$, we obtain the linear system of equations

$$\mathbb{B}\,\mathbf{c} = \mathbf{f} \tag{7.9}$$

with $\mathbb{B} = (\mathfrak{b}(\varphi_{\mathsf{v}},\varphi_{\mathsf{v}'}))_{\mathsf{v},\mathsf{v}'\in\mathsf{V}}$ and $\mathbf{f} = (\mathfrak{f}(\varphi_{\mathsf{v}}))_{\mathsf{v}\in\mathsf{V}}$. Again by Lax-Milgram theory, this system is invertible, which allows to compute the solution $\hat{u}$.

Reduced models: Unfortunately, solving Equation (7.9) is expensive, given the size $|\mathsf{V}| \approx 10^4$ of the graph, so we cannot afford to find $\hat{u}$ at each time step. In order to mitigate the computational cost, we rely on model order reduction techniques, which have received much attention in the context of parametrized elliptic PDEs [52, 54, 71, 92, 140, 161]. Here, the parameters would be the diffusion a, the reaction h, and the right-hand side q. We consider three reconstructions methods.

1. **Eigenstates of the graph Laplacian:** One option consists in taking as a reduced model the subspace $\mathcal{V}_n \subset H^1(\mathsf{G})$ spanned by the n first eigenfunctions of the Laplacian in $\mathbb{P}_1$. As this operator is self-adjoint and coercive, it admits a spectral decomposition with positive eigenvalues, and the coefficients of the eigenstates in the basis $\{\varphi_{\mathsf{v}}\}_{\mathsf{v}\in\mathsf{V}}$ are the eigenvectors of $\mathbb{B}$. Assuming that the diffusion and reaction coefficients a and h are constants calibrated in a pre-processing phase, we define the reconstruction mapping $A : \mathcal{X} \to H^1(\mathsf{G})$ by taking $\hat{u} = A(q)$ the Galerkin projection of u onto $\mathcal{V}_n$, that is, by searching $\hat{u} \in \mathcal{V}_n$ solution to

   $$\mathfrak{b}(\hat{u},\hat{v}) = \mathfrak{f}(\hat{v}), \qquad \hat{v} \in \mathcal{V}_n,$$

   which is simply a diagonal system in the eigenstate basis. We then plug $\hat{u}$ instead of q in equation (7.4) or (7.5), and learn the coefficients associated to each color, or the more general parameters α.

2. **Principal components of traffic data:** Starting with the whole history of traffic data $(q(t))_{t\in T} \in \mathbb{R}^{|T|\times|\mathsf{V}|}$, we can also perform a singular value decomposition to find the n first modes $q_1,\dots,q_n$, compute the solutions to $\mathcal{P}(u_k) = q_k$, and assemble them in a reduced space $\mathcal{V}_n = \operatorname{span}\{u_1,\dots,u_n\}$. In this way, we expect $\mathcal{V}_n$ to better capture physical properties of the pollution field, such as strong correlations along a large avenue.

As full-order solves remain costly, we resort to a convolution with a gaussian kernel:

$$(u_k)_c^v = \frac{\sum_{v' \in V} e^{-d_{vv'}^2/2\delta^2} (q_k)_c^{v'} \phi_{v'}}{\sum_{v' \in V} e^{-d_{vv'}^2/2\delta^2}}$$

where $d_{vv'} = |v - v'|$ is the distance in $\mathbb{R}^2$ (which is equivalent, up to constants, to the distance on the graph). We set $\delta = 400\,\mathrm{m}$, after observing that pollution data is optimally correlated to regularized traffic information for δ close to this value.

In our experiments, we perform the smoothed projections $q \mapsto \hat{u} = \sum_{k=1}^n \langle q, q_k \rangle u_k$ into a different reduced space for each of the four traffic colors. After this operation, we can apply any of the strategies described in Section 7.3.4 to $\hat{u}$ instead of q.

These two methods regularize the traffic data, but they do not exploit the information from the pollution sensors, apart from the average value $\bar{z}$. In order to assimilate data of both types, it is possible to use a combined least-squares fit of the form

$$A(x) = \arg\min_{\hat{v} \in \mathcal{V}_n} \|z - \hat{v}(v^{\mathrm{obs}})\|_2^2 + \lambda' \|q - \hat{q}\|_{\ell^2(V)}^2,$$

where $\lambda' > 0$ balances the contributions of z and q, and $\hat{q} = \mathcal{P}(\hat{v}) = \sum_{k=1}^n \hat{c}_k q_k$ for the coefficients $\hat{c} \in \mathbb{R}^n$ such that $\hat{v} = \sum_{k=1}^n \hat{c}_k u_k$. However, this approach requires tuning an additional parameter λ', and creates spatial correlations in the pollution output based on the traffic history, without taking distances into account. These non-local effects are amplified by the noise in the data. As the method did not perform well in practice, we did not include it in the numerical experiments.

In the last method, if $m \geqslant n$ and λ' tends to 0, the prediction $\hat{u}$ does a least squares fit of u at the available measurement points v_i^{obs}. In general, it is possible to enforce $\hat{u}(v^{\mathrm{obs}}) = u(v^{\mathrm{obs}})$ by applying a correction to the prediction. This post-process, called *Parameterized Background Data-Weak* method, was originally introduced in [108] and has been analyzed and extended in a series of papers such as [26, 47, 48, 50]. The whole approach has found numerous applications, including pollution dispersion [87].

It would of course be possible to gain in accuracy, by considering more refined equations for pollution dispersion, which capture additional physical properties, and thus by encoding these properties into the reduced space $\mathcal{V}_n$. One could for instance think of advection by wind, vertical fluxes or stratification of the atmosphere depending on the temperatures, changes in the chemical equilibrium between NO and NO_2 caused by cloud coverage and precipitations [105], as well as local turbulent effects near the sensor stations. Note that, if nonlinear equations are involved, $\mathcal{V}_n$ can be a nonlinear approximation space defined through a chart of n parameters, and approximation guarantees are more difficult to obtain [50].

7.3.6 Super-Learning as a collaborative approach

To gain in accuracy over each individual model, one can combine a set of p available mappings $A_1, \dots, A_p$ coming from the previous methods, and build a super-learner

$$\begin{array}{rccc} & \mathcal{F}(\mathcal{X},\mathcal{U})^p & \longrightarrow & \mathcal{F}(\mathcal{X},\mathcal{U}) \\ \mathcal{S}: & (A_1,\dots,A_p) & \longmapsto & \mathcal{S}(A_1,\dots,A_p), \end{array}$$

where $\mathcal{F}(\mathcal{X},\mathcal{U})$ denotes the set of functions from $\mathcal{X}$ to $\mathcal{U}$. The most simple merger, usually called aggregator in statistics, amounts to taking a linear combination

$$\mathcal{S}_\omega(A_1,\dots,A_p) = \sum_{i=1}^{p} \omega_i A_i,$$

for some weights $\omega = (\omega_1, \dots, \omega_p)$ expressing the confidence in each individual model.

More sophisticated strategies involve nonlinear combinations and compositional structure. One could think of using a first model to obtain a rough estimation, and compose it with a second model performing refinements based on its output. This is already an underlying idea in our constructions, where we start with the spatial average, and add spatially-dependent corrections. The physical models involve one more compositional step, since they are of the form $A_{\text{src}} \circ \hat{u}(q)$.

Neural networks constitute another prominent example of nonlinear super-learners: one could treat $A_1(x), \dots, A_p(x)$ as inputs, and train the parameters ω by minimizing an empirical loss. We would like to emphasize here that properly training the super-learner requires to implement a nested leave-one-out strategy: one should first train each parametrized submodel by leave-one-out, and then optimize the neural network with another leave-one-out step, in order to avoid overfitting. As a consequence, at least two observation points are removed from the training set of the submodels, which may cause a loss of accuracy, especially when the number m of observations is small.

In our application, the advantage provided by nonlinear approaches was limited, and the neural network super-learner performed slightly worse than its linear counterpart, which should come as no surprise in view of the above observation.

7.4 Reconstruction benchmarks and Leave-One-Out

There are several ways to quantify the quality of a reconstruction map $A : \mathcal{X} \to \mathcal{U}$. Ideally, given a state $u \in \mathcal{U}$ and the associated observations $x \in \mathcal{X}$, one would like to find A such that the error $\|u - A(x)\|_{\mathcal{U}}$ is as small as possible. Assuming that (u, x) is a random variable with distribution $\pi \in \text{Prob}(\mathcal{U} \times \mathcal{X})$, we define the performance of A as the L^2 norm of the error

$$e(A)^2 := \int_{\mathcal{U}\times\mathcal{X}} \|u - A(x)\|_{\mathcal{U}}^2 \, d\pi(u, x),$$

which acts as a good compromise between the L^∞ worst-case error and the L^1 average error. In addition, although the state u has in principle H^2 regularity, we asses the spatial error

also in L^2, that is, we take $\mathcal{U} = L^2(\mathsf{G})$. Unfortunately, finding A minimizing $e(A)$ is out of reach for several reasons. First, we don't know the distribution π, nor even its support, which is the set of all possible states and observations. Second, given $u \in \mathcal{U}$, we cannot evaluate $u(r)$ at any point $r \in \mathsf{G}$, making the computation of $\|u - A(x_u)\|_{\mathcal{U}}$ intractable.

Concerning the first issue, as we have access to hourly data on a large period of time, we can replace the integral over π by an empirical average

$$e(A)^2 \approx \frac{1}{T_{\text{test}}} \sum_{t \in T_{\text{test}}} \|u(t) - A(x)(t)\|_{\mathcal{U}}^2$$

over the set T_{test} of 374 test times. Assuming that these states are independent, this approximation induces an error of order

$$\frac{\mathbb{E}(\|u\|_{\mathcal{U}}^2)^{1/2}}{\sqrt{|T_{\text{test}}|}} \approx \frac{41}{19.3} \approx 2.1\,\mu\text{g/m}^3,$$

which is totally acceptable in view of the noise level on the measurements.

The second obstacle is more tricky, because we only know u at a very limited number m of fixed positions, and because these observations are also needed for constructing A. Ignoring the last issue leads to an systematic underestimation of $e(A)$, as we detail below.

Assume that the observation points $\mathsf{v}_i^{\text{obs}}$ are distributed uniformly at random on G, define the discrete semi-norm

$$\|u\|_m^2 = \frac{1}{m} \sum_{i=1}^{m} |u(\mathsf{v}_i^{\text{obs}})|^2$$

corresponding to an empirical version of $\|u\|_{\mathcal{U}}^2$, and consider map A solution to

$$\min_{A : \mathcal{X} \to \mathcal{V}_n \text{ linear}} \frac{1}{T_{\text{train}}} \sum_{t \in T_{\text{train}}} \|u - A(x)\|_m^2$$

for some linear space $\mathcal{V}_n \subset \mathcal{U}$ of dimension n. This setting is valid for most of our methods, with $\mathcal{V}_n$ the set of constant functions in the case of A_{avg} (of dimension $n = 1$), but also $\mathcal{V}_n = \text{span}\{r \mapsto c_i^r\}_{1 \leqslant i \leqslant m}$ in the case of A_{krig} (of dimension $n = m$), and $\mathcal{V}_n$ the reduced basis in the methods based on physical modeling.

Then, for A^* the optimal map taking values in $\mathcal{V}_n$

$$A^* = \operatorname*{arg\,min}_{A' : \mathcal{X} \to \mathcal{V}_n \text{ linear}} \mathbb{E}(\|u - A'(x)\|_{\mathcal{U}}^2) = \mathbb{E}_\pi(u|x),$$

we obtain, by applying Pythagoras theorem both for $\|\cdot\|_{\mathcal{U}}$ and $\|\cdot\|_m$,

$$e(A)^2 = \mathbb{E}(\|u - A(x)\|_{\mathcal{U}}^2) \geqslant \mathbb{E}(\|u - A^*(x)\|_{\mathcal{U}}^2) = \mathbb{E}(\|u - A^*(x)\|_m^2) \geqslant \mathbb{E}(\|u - A(x)\|_m^2),$$

where the central equality comes from the assumption that the $\mathsf{v}_i^{\text{obs}}$ are random. This proves that $\|u - A(x)\|_m^2$ is a biased estimator for $e(A)^2$ as soon as one of the inequalities

is strict, that is, as soon as $A(x) \neq A^*(x)$. Note that separating the training and test data by splitting the set of time indices T is not sufficient, since the algorithm will then fit its prediction to the station locations, without generalization guarantees to the rest of the domain G.

As a consequence, we must separate the stations into a training set and test points. In order to compute an unbiased estimator of $e(A)^2$ with minimal variance, while keeping the maximal number of stations in the training set, we proceed to leave-one-out cross-validation. This procedure is very standard and has been used in other works on pollution reconstruction, such as [58]. For $1 \leqslant i \leqslant m$, denote

$$\|u\|_{m\setminus i}^2 = \frac{1}{m-1}\sum_{j\neq i} |u(\mathsf{v}_j^{\mathrm{obs}})|^2 \quad \text{and} \quad A_i = \operatorname*{arg\,min}_{A:\mathcal{X}\to\mathcal{V}_n \text{ linear}} \frac{1}{T_{\mathrm{train}}} \sum_{t\in T_{\mathrm{train}}} \|u - A(x)\|_{m\setminus i}^2.$$

Assuming that the station locations are independent random variables, the cross-validation estimator of the error

$$e_{\mathrm{CV}}(A)^2 := \frac{1}{T_{\mathrm{test}}} \sum_{t\in T_{\mathrm{test}}} \frac{1}{m}\sum_{i=1}^{m} |u(\mathsf{v}_i^{\mathrm{obs}}) - A_i(\mathsf{v}_i^{\mathrm{obs}})|^2$$

is unbiased, since

$$\mathbb{E}(e_{\mathrm{CV}}(A)^2) = \mathbb{E}(|u(\mathsf{v}_i^{\mathrm{obs}}) - A_i(\mathsf{v}_i^{\mathrm{obs}})|^2) = \mathbb{E}(\|u - A_i(x)\|_{\mathcal{U}}^2) = e(A),$$

with the difference that x contains only $m-1$ direct evaluations of u this time.

7.5 Numerical results

We have implemented and tested numerous variants and combinations of the models from Section 7.3. This was done thanks to a Python code we have developed, which can be found at `https://github.com/agussomacal/CityPollutionModeling`. The interested user could add its own models for further testing. In this section, we summarize the most important results that emerge from our tests. We report on the performance of the following reconstruction strategies:

- **Spatial average**: We take a simple spatial average, as in Section 7.3.1. The resulting error serves as a baseline, which we expect to beat with the other more sophisticated models.
- **BLUE**: As explained in Section 7.3.2, BLUE can be seen as an estimate of the optimal linear reconstruction method. It can be used as a benchmark of the best performance that we can expect of linear methods. Note that, in principle, nonlinear strategies could be more accurate than BLUE. However, we will see in our experiments that none of our methods achieves such accuracy.
- **Kriging**: We apply the kriging method depicted in Section 7.3.3 with an exponential

kernel. The parameters σ and C are obtained by fitting an exponential to the correlation between training stations (that is, we do not into account correlations with the station that is set aside for testing) as a function of their distance (see Figure 7.4).

- **Source**: We apply a linear source model, as described in Section 7.3.4, with temperature θ and wind w as extra regressor variables.
- **Physical-PCA**: We apply the second physical model from Section 7.3.5, using a gaussian kernel to smooth the node traffic data. The four reduced spaces $\mathcal{V}_n$, associated to the four traffic colors, each consist of the first 10 principal components of the corresponding traffic data, as observed in the training set. After the smoothing and projection operations, we assemble the variables θ, w and the $q_{\mathsf{c}}^{\mathsf{v}}$ on each node v into a vector $s = (q_{\texttt{green}}^{\mathsf{v}}, q_{\texttt{orange}}^{\mathsf{v}}, q_{\texttt{red}}^{\mathsf{v}}, q_{\texttt{dark-red}}^{\mathsf{v}}, \theta, w) \in \mathbb{R}^6$ and build a *polynomial* model of degree 2:
$$\mathcal{T}_\alpha(s) := \sum_{j=1}^{6} \alpha_j s_j + \sum_{j,k=1}^{6} \alpha_{jk} s_j s_k,$$
where $\alpha \in \mathbb{R}^{42}$ is computed following the lines of Section 7.3.4.
- **Physical-Laplacian**: We apply the first physical model from Section 7.3.5, with the reduced space consisting of the first 5 eigenvectors of the graph laplacian. After projection into the subspace, we take the 4-colour traffic values on each node $(q_{\mathsf{c}}^n)_{\cdot,i}$ and obtain their *degree*-3 *polynomial* combinations. Finally we apply a *neural network* consisting of two hidden layers of 20 neurons each and a ReLU activation function. The neural network is trained with ADAM optimizer with early stopping to prevent overfitting.
- **Ensemble**: We apply an ensemble model, as described in Section 7.3.6, combining the Kriging method A_{krig}, the Source model A_{src} and the Physical-Laplacian model A_{lapl}. We train each of them separately and compute the following linear combination:
$$A_{\text{ens}}(x)(r) = \omega(r) A_{\text{krig}}(x)(r) + \frac{1-\omega(r)}{2} A_{\text{src}}(x)(r) + \frac{1-\omega(r)}{2} A_{\text{lapl}}(x)(r),$$
with a weight function $\omega(r) = \exp(\min_{1 \leqslant i \leqslant m} |r - \mathsf{v}_i^{\text{obs}}|/\delta)$, where $\delta = 800\,\text{m}$. Essentially, we favour Kriging when r is close to one of the sensor stations, and average the predictions of models using local or global traffic information otherwise.

In Figure 7.5, we show the root mean square error (in $\mu\text{g}/\text{m}^3$) for each model's predictions on the test times T_{test} and tested stations i:
$$e_{\text{RMSE}}(A, i) := \left(\frac{1}{T_{\text{test}}} \sum_{t \in T_{\text{test}}} |u(\mathsf{v}_i^{\text{obs}}) - A_i(\mathsf{v}_i^{\text{obs}})|^2 \right)^{1/2}.$$

Note that the cross-validation error $e_{\text{CV}}(A)$ from Section 7.4 is just the ℓ^2-average of these errors over all stations. For the tests, we only keep the 10 stations located in the interior of

Paris. The remaining 3 are set aside because they have a significant proportion of missing values (in average 10% of the data is lacking in each of these stations), and because they lay close to the border of the image, making the surrounding traffic information incomplete.

The shaded blue area is the region corresponding to errors smaller than the reference *BLUE* model. On the opposite side, the shaded red area marks situations in which the prediction is worse that the *spatial average* baseline. The white margin in between indicates the region where we expect feasible improvements.

We first notice that using a linear *source model* already yields reliable improvements with respect to the *spatial average* baseline. It fails, however, in HAUS and OPERA stations due to the absence of traffic information around the former, as the Google Maps symbol for the Paris Opera is drawn over the location of the sensor. This affects the predictions on both stations but most prominently on HAUS. This problem can be alleviated if we average traffic information over a bigger region, as done in *Pysical-PCA* thanks to the Gaussian smoothing, at the expense of losing precision on other stations like PA18.

The *Kriging* model manages to produce enhanced predictions in both HAUS and OPERA stations because of their proximity and correspondingly high correlation in pollution values. However, for other stations, especially those further from the center, the performance highly deteriorates.

With the *Physical-Laplacian* model, we get further improvements in 6 stations compared to the linear *source model*, while losing some advantage in the remaining 4. Finally, the *ensemble* method, by combining two traffic models and the *Kriging* method, manages to exploit the advantages of each in a pretty decent way. It yields predictions that beat or equal the *spatial average* baseline on all stations, and that reach the best average error among all our tested methods. In Figure 7.6, we show an example of pollution maps generated with this last model.

7.6 Conclusion and future works

In this work, we have shown that it is possible to leverage pollution sensor data, meteorological information and Google Traffic images to create pollution maps in real time. In particular, we explained how to build statistical, physics-based and ensemble reconstruction strategies by posing the problem of pollution state estimation on metric and quantum graphs. Furthermore, the right combination of these techniques produced systematic improvements over the proposed baselines, namely the *Spatial average* and *Kriging*.

Neither our linear reconstruction strategies nor our nonlinear ones could beat the BLUE benchmark that indicates the accuracy of the best linear estimator. We conjecture that this is due to the limited amount of stations giving us spatial information on the pollution field, and to the indirect nature of traffic data. Regarding the last point, even though the volume of traffic information is large, it still remains of reduced utility. This is due to the fact that we only measure the fluidity of traffic, instead of the actual amount of passing vehicles, which is the relevant variable directly impacting emissions. One can then hope to obtain further improvements by following a similar approach with better suited data.

Another limitation is the unavailability of local pollution averages for the city of Paris.

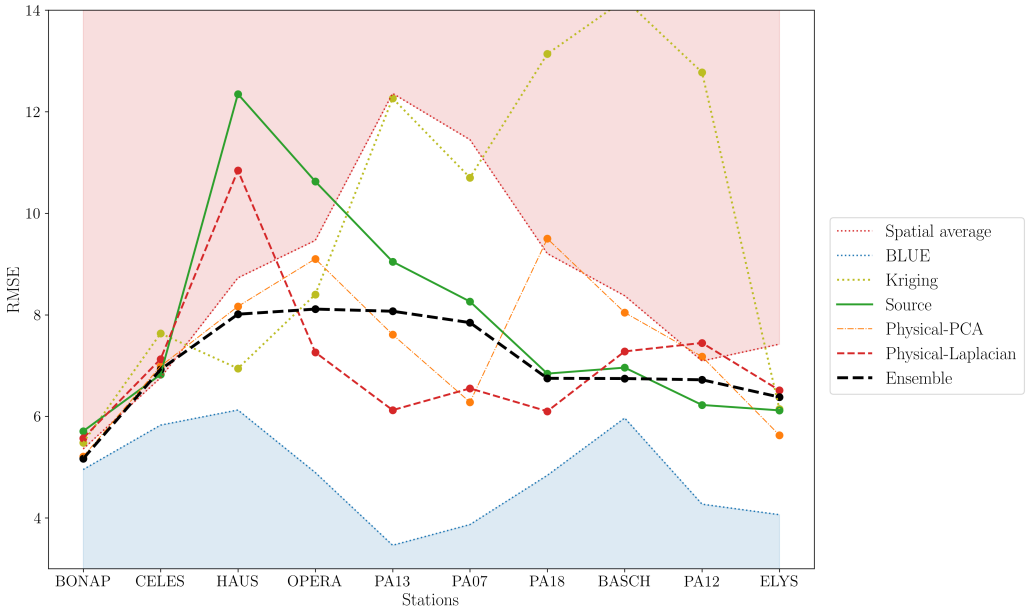


Figure 7.5: Root mean square error on tested stations for the different proposed methods

Having access to this kind of data through intensive measurement campaigns lasting a few weeks, but employing hundreds to thousands of sensors, as done in [58] for the city of Barcelona, would give a much more precise baseline, and allow to learn corrections to the local average instead of the global one.

Finally, in setting the problem on the graph, we did not take into account the vicinity of open spaces like parks or rivers, nor the topology and variations in altitude. This is particularly visible in Figure 7.6, where the parks of Boulogne and Vincennes are colored in red because of the surrounding highways, and the absence of small internal streets. We leave the inclusion of such relevant features to future studies.

Acknowledgments and disclosure of funding: The authors would like to thank Prof. Albert Cohen, Prof. Joubine Aghili, Dr. Rachida Chakir, Dr. Vivien Mallet, and Dr. Fabien Brocheton for fruitful discussions, and for preliminary work on the extraction of applicable data.

This work was done in the framework of the research project "Models and Measures" funded by the Parisian City Council (Emergences grant program). In addition, Matthieu Dolbeault acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project number 442047500 through the Collaborative Research Center "Sparsity and Singular Structures" (SFB 1481).

7.A Metric graphs

Here we recall several notions about graphs that are necessary in our developments. The presentation is based on the book [23], which provides a comprehensive introduction to

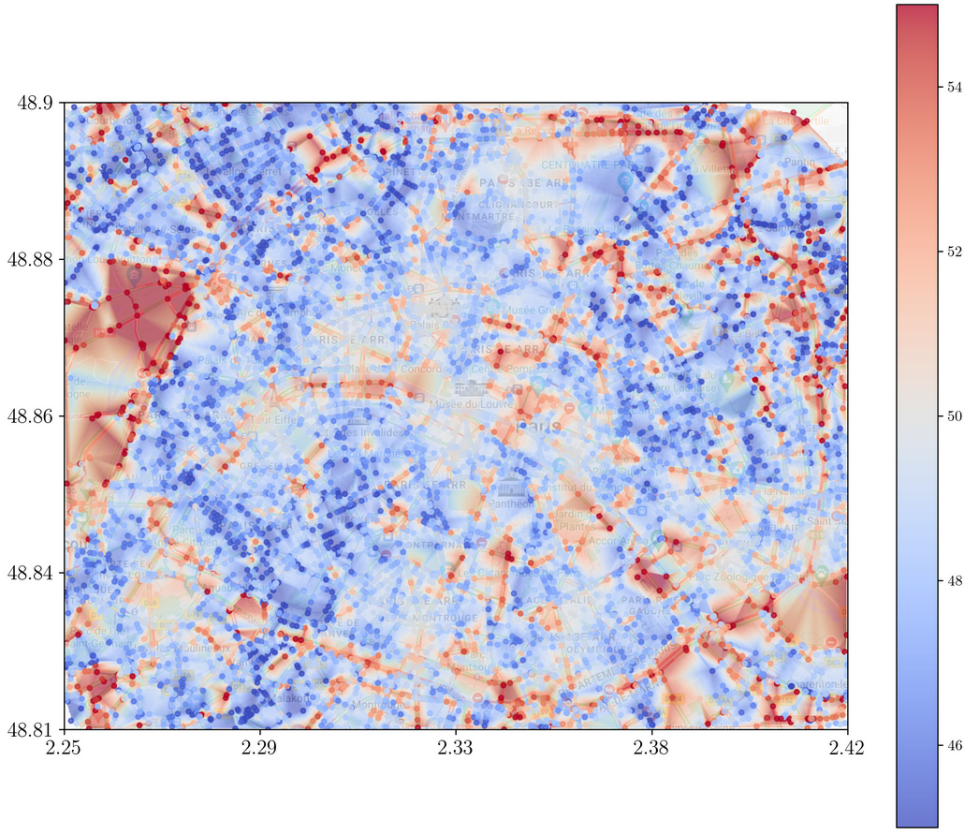


Figure 7.6: Pollution map for the *ensemble* model at 8am on March 1st, 2023. Based on the predictions on the node, one can linearly extrapolate pollution values even outside of the graph edges. Note that the fine variations in pollutant concentration (between 45 and 55 $\mu g/m^3$) seem to trace the main circulation axes.

quantum graphs, and on the paper [10], which develops finite element discretizations of elliptic operators in quantum graphs. We sometimes narrow down the generality of certain notions for the purposes of the present paper.

A *combinatorial graph* $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is a collection of a finite number of vertices V and of edges $\mathsf{E} \subset \mathsf{V} \times \mathsf{V}$ connecting pairs of vertices. We restrict our attention to *undirected* graphs where no orientation is assigned to the edges, and denote $|\mathsf{V}|$ and $|\mathsf{E}|$ the number of vertices and edges, respectively.

We will work with *connected* graphs, where any two vertices $\mathsf{v}, \mathsf{w} \in \mathsf{V}$ are connected by at least one path $(\mathsf{v}, \mathsf{v}_1), (\mathsf{v}_1, \mathsf{v}_2), \dots, (\mathsf{v}_k, \mathsf{w})$ made by consecutive adjacent edges in E. A connected graph becomes a *metric graph* if we assign a length $\ell_\mathsf{e} > 0$ and a local coordinate $r_\mathsf{e}(x)$, for $x \in [0, \ell_\mathsf{e}]$, to each edge $\mathsf{e} = (\mathsf{v}, \mathsf{w}) \in \mathsf{E}$, in such a way that $r_\mathsf{e}(0) = \mathsf{v}$ and $r_\mathsf{e}(\ell_\mathsf{e}) = \mathsf{w}$.

In our case, the crossroads V are embedded in $\mathbb{R}^2$ through their geographical coordinates, and the streets $r_\mathsf{e}([0, \ell_\mathsf{e}]) \subset \mathbb{R}^2$ are differentiable curves with no loops. However, as done very often, we redefine them as simple straight lines joining the two vertices. Regardless of the choice of the edge curves, the points r in a metric graph G are thus not only its vertices

but also all intermediate points on the edges as well, parametrized by the local coordinates r_{e}:

$$\mathsf{V} \subsetneq \mathsf{G} = \bigcup_{\mathsf{e}\in\mathsf{E}} r_{\mathsf{e}}([0,\ell_{\mathsf{e}}]).$$

As the name suggests, any metric graph can be endowed with a natural metric as follows. The distance between two vertices $\mathsf{v}, \mathsf{w} \in \mathsf{V}$ is usually defined as the length of the shortest path connecting them. This notion of distance between vertices is then extended in a natural way to any two points possibly lying on different edges, by further adding the local coordinates along these edges.

We may now introduce function spaces and linear differential operators on a metric graph G. The space of continuous functions $\mathcal{C}(\mathsf{G})$ contains the functions $u : \mathsf{G} \to \mathbb{R}$ such that $u \circ r_{\mathsf{e}}$ is continuous on $[0,\ell_{\mathsf{e}}]$ for any edge $\mathsf{e} \in \mathsf{E}$, which implies in particular the continuity of u along any path in G. The space of square-integrable functions

$$L^2(\mathsf{G}) = \bigoplus_{\mathsf{e}\in\mathsf{E}} L^2(r_{\mathsf{e}}([0,\ell_{\mathsf{e}}]))$$

is a Hilbert space when endowed with the inner product

$$\langle u, v\rangle_{L^2(\mathsf{G})} \coloneqq \int_{\mathsf{G}} u(r)v(r)\mathrm{d}r = \sum_{\mathsf{e}\in\mathsf{E}} \int_0^{\ell_{\mathsf{e}}} u(r_{\mathsf{e}}(x))v(r_{\mathsf{e}}(x))dx.$$

Finally, the Sobolev space

$$H^1(\mathsf{G}) = \mathcal{C}(\mathsf{G}) \cap \bigoplus_{\mathsf{e}\in\mathsf{E}} H^1(r_{\mathsf{e}}([0,\ell_{\mathsf{e}}]))$$

is also a Hilbert space, for the norm

$$\|u\|^2_{H^1(\mathsf{G})} \coloneqq \int_{\mathsf{G}} u^2\mathrm{d}r + \int_{\mathsf{G}} \left(\frac{\mathrm{d}u}{\mathrm{d}r}\right)^2 \mathrm{d}r = \sum_{\mathsf{e}\in\mathsf{E}} \int_0^{\ell_{\mathsf{e}}} u(r_{\mathsf{e}}(x))^2 dx + \int_{\mathsf{G}} \left(\frac{\mathrm{d}(u\circ r_{\mathsf{e}})}{dx}\right)^2 dx.$$

The restriction to $\mathcal{C}(\mathsf{G})$ in the definition of $H^1(\mathsf{G})$ stems from the fact that functions in $H^1(r_{\mathsf{e}}([0,\ell_{\mathsf{e}}]))$ are continuous (because their domain is one dimensional), which automatically implies that functions in $H^1(\mathsf{G})$ must be continuous also at the vertices. In the same vein, one has to impose restrictions on the derivatives of u, such as Newmann-Kirchoff boundary conditions, for functions $u \in H^2(\mathsf{G})$.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems. *arXiv preprint*, 2016. DOI: arXiv:1603.04467 (cited on page 170).

[2] R. A. Adams and J. J. F. Fournier. *Sobolev spaces*. eng, number 140 in Pure and applied mathematics. Academic Press, Amsterdam Heidelberg, 2. ed., reprinted edition, 2008 (cited on pages 19, 46, 105).

[3] B. Adcock, A. C. Hansen, and C. Poon. Beyond consistent reconstructions: optimality and sharp bounds for generalized sampling, and application to the uniform resampling problem. *SIAM Journal on Mathematical Analysis*, 45(5):3132–3167, 2013 (cited on page 63).

[4] M. Ainsworth. Robust a posteriori error estimation for nonconforming finite element approximation. *SIAM J. Numer. Anal.*, 42(6):2320–2341, 2005. DOI: 10.1137/S0036142903425112 (cited on page 33).

[5] B. Aksoylu, I. G. Graham, H. Klie, and R. Scheichl. Towards a rigorously justified algebraic preconditioner for high-contrast diffusion problems. *Comput. Vis. Sci.*, 11(4-6):319–331, 2008. DOI: 10.1007/s00791-008-0105-1 (cited on page 33).

[6] B. Aksoylu and Z. Yeter. Robust multigrid preconditioners for cell-centered finite volume discretization of the high-contrast diffusion equation. *Comput. Vis. Sci.*, 13(5):229–245, 2010. DOI: 10.1007/s00791-010-0140-6 (cited on page 33).

[7] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012. DOI: 10.1002/nme.4371 (cited on page 136).

[8] F. Arandiga, A. Cohen, R. Donat, and N. Dyn. Interpolation and approximation of piecewise smooth functions. *SIAM Journal on Numerical Analysis*, 43(1):41–57, 2005 (cited on pages 68, 76).

[9] 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. *Journal of Computational Physics*, 363:354–370, 2018. DOI: 10.1016/j.jcp.2018.02.050 (cited on pages 63, 65).

[10] M. Arioli and M. Benzi. A finite element method for quantum graphs. *IMA Journal of Numerical Analysis*, 38(3):1119–1163, 2018 (cited on pages 192, 201).

[11] I. Babuska and R. Lipton. Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. *Multiscale Modeling & Simulation*, 9(1):373–406, 2011 (cited on page 33).

[12] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007. DOI: 10.1137/050645142 (cited on page 30).

[13] M. Bachmayr and A. Cohen. Kolmogorov widths and low-rank approximations of parametric elliptic pdes. *Mathematics of Computation*, 86(304):701–724, 2017 (cited on pages 12, 30, 41, 43, 50).

[14] M. Bachmayr, A. Cohen, and G. Migliorati. Sparse polynomial approximation of parametric elliptic PDEs. Part I: Affine coefficients. *ESAIM Math. Model. Numer. Anal.*, 51(1):321–339, 2017. DOI: 10.1051/m2an/2016045 (cited on pages 8, 30, 31).

[15] J. Bäck, F. Nobile, L. Tamellini, and R. Tempone. Stochastic Spectral Galerkin and Collocation Methods for PDEs with Random Coefficients: A Numerical Comparison. en. In *Spectral and High Order Methods for Partial Differential Equations*. Volume 76, pages 43–62. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. DOI: 10.1007/978-3-642-15337-2_3. Series Title: Lecture Notes in Computational Science and Engineering (cited on pages 10, 12, 13, 30, 31).

[16] J. Barnett and C. Farhat. Quadratic approximation manifold for mitigating the kolmogorov barrier in nonlinear projection-based model order reduction. *Journal of Computational Physics*, 464:111348, 2022. DOI: 10.1016/j.jcp.2022.111348 (cited on pages 136, 140).

[17] J. L. Barnett, C. Farhat, and Y. Maday. Mitigating the kolmogorov barrier for the reduction of aerodynamic models using neural-network-augmented reduced-order models. In *AIAA SCITECH 2023 Forum*, page 0535, 2023. DOI: 10.2514/6.2023-0535 (cited on pages 136, 140).

[18] B. Battisti, T. Blickhan, G. Enchery, V. Ehrlacher, D. Lombardi, and O. Mula. Wasserstein model reduction approach for parametrized flow problems in porous media. *ESAIM: Proceedings and Surveys*, 73:28–47, 2023 (cited on page 67).

[19] J. Beck, F. Nobile, L. Tamellini, and R. Tempone. Implementation of optimal galerkin and collocation approximations of pdes with random coefficients. In *ESAIM: Proceedings*, volume 33, pages 10–21. EDP Sciences, 2011 (cited on page 30).

[20] A. Beguinet, V. Ehrlacher, R. Flenghi, M. Fuente, O. Mula, and A. Somacal. Deep learning-based schemes for singularly perturbed convection-diffusion problems. *ESAIM: Proceedings and Surveys*, 73:48–67, 2023. DOI: 10.1051/proc/202373048 (cited on pages v, vi, 23, 218, 219).

[21] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox. *Model Reduction and Approximation: Theory and Algorithms*, volume 15. SIAM, 2017 (cited on page 67).

[22] P. Berger, K. Gröchenig, and G. Matz. Sampling and reconstruction in distinct subspaces using oblique projections. *J. Fourier Anal. Appl.*, 25(3):1080–1112, 2019. DOI: 10.1007/s00041-018-9620-8 (cited on pages 68, 73).

[23] G. Berkolaiko and P. Kuchment. *Introduction to quantum graphs*, volume 186. American Mathematical Soc., 2013 (cited on pages 182, 191, 200).

[24] F. Bernard, A. Iollo, and S. Riffaud. Reduced-order model for the bgk equation based on pod and optimal transport. *Journal of Computational Physics*, 373:545–570, 2018. DOI: 10.1016/j.jcp.2018.07.001 (cited on page 136).

[25] C. Bernardi and R. Verfürth. Adaptive finite element methods for elliptic equations with non-smooth coefficients. *Numer. Math.*, 85(4):579–608, 2000. DOI: 10.1007/PL00005393 (cited on page 33).

[26] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Data assimilation in reduced modeling. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):1–29, 2017. DOI: 10.1137/15M1025384 (cited on pages 34, 53, 65, 66, 194).

[27] P. Binev, A. Cohen, O. Mula, and J. Nichols. Greedy algorithms for optimal measurements selection in state estimation using reduced models. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1101–1126, 2018. DOI: 10.1137/17M1157635 (cited on pages 65, 67).

[28] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM Journal on Mathematical Analysis*, 43(3):1457–1472, 2011. DOI: 10.1137/100795772 (cited on pages 30, 50).

[29] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Data Assimilation in Reduced Modeling. en. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):1–29, January 2017. DOI: 10.1137/15M1025384 (cited on page 16).

[30] F. Black, P. Schulze, and B. Unger. Projection-based model reduction with dynamically transformed modes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 54(6):2011–2043, 2020. DOI: 10.48550/arXiv.1912.11138 (cited on page 136).

[31] V. Bogachev, N. Krylov, M. Röckner, and S. Shaposhnikov. *Fokker-Planck-Kolmogorov Equations*, volume 207. American Mathematical Soc., 2015 (cited on page 158).

[32] B. Bojanov. Optimal recovery of functions and integrals. In *First European Congress of Mathematics*, pages 371–390. Springer, 1994 (cited on page 64).

[33] A. Bonito, A. Cohen, R. DeVore, D. Guignard, P. Jantsch, and G. Petrova. Nonlinear methods for model reduction. *ESAIM Math. Model. Numer. Anal.*, 55(2):507–531, 2021. DOI: 10.1051/m2an/2020057 (cited on pages 50, 136).

[34] L. Breiman. Stacked regressions. *Machine learning*, 24:49–64, 1996 (cited on page 180).

[35] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001 (cited on page 145).

[36] A. N. Brooks and T. J. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 32(1-3):199–259, 1982 (cited on page 159).

[37] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In *4th International Conference on Learning Representations, ICLR 2016*, 2016 (cited on page 76).

[38] S. Budennyy, V. Lazarev, N. Zakharenko, A. Korovin, O. Plosskaya, D. Dimitrov, V. Akhripkin, I. Pavlov, I. Oseledets, I. Barsola, et al. Eco2AI: carbon emissions tracking of machine learning models as the first step towards sustainable AI. In *Doklady Mathematics*, pages 1–11. Springer, 2023 (cited on page 27).

[39] 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: Mathematical modelling and numerical analysis*, 46(3):595–603, 2012 (cited on page 30).

[40] N. Cagniart, Y. Maday, and B. Stamm. Model order reduction for problems with large convection effects. *Contributions to partial differential equations and applications*:131–150, 2019. DOI: 10.1007/978-3-319-78325-3\textunderscore10 (cited on page 136).

[41] J. Cai, J. Chen, H. Cheng, S. Zi, J. Xiao, F. Xia, and J. Zhao. The effects of thermal stratification on airborne transport within the urban roughness sublayer. *International Journal of Heat and Mass Transfer*, 184:122289, 2022 (cited on page 182).

[42] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006. DOI: 10.1002/cpa.20124 (cited on pages 68, 84).

[43] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 659–668, 2002 (cited on page 88).

[44] J. Chan, N. Heuer, T. Bui-Thanh, and L. Demkowicz. A robust DPG method for convection-dominated diffusion problems ii: adjoint boundary conditions and mesh-dependent test norms. *Computers & Mathematics with Applications*, 67(4):771–795, 2014 (cited on page 159).

[45] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*:808–817, 2000 (cited on page 30).

[46] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009 (cited on pages 68, 85, 88).

[47] A. Cohen, W. Dahmen, R. DeVore, J. Fadili, O. Mula, and J. Nichols. Optimal reduced model algorithms for data-based state estimation. *SIAM Journal on Numerical Analysis*, 58(6):3355–3381, 2020. DOI: 10.1137/19m1255185 (cited on pages 65, 194).

[48] A. Cohen, W. Dahmen, O. Mula, and J. Nichols. Nonlinear reduced models for state and parameter estimation. *SIAM/ASA Journal on Uncertainty Quantification*, 10(1):227–267, 2022 (cited on pages 65, 194).

[49] A. Cohen and R. DeVore. Approximation of high-dimensional parametric PDEs. *Acta Numer.*, 24:1–159, 2015. DOI: 10.1017/S0962492915000033 (cited on page 65).

[50] A. Cohen, M. Dolbeault, O. Mula, and A. Somacal. Nonlinear approximation spaces for inverse problems. *Anal. Appl. (Singap.)*, 21(1):217–253, 2023. DOI: 10.1142/S0219530522400140 (cited on pages v, vi, 15, 27, 112, 194, 218, 219).

[51] A. Cohen, W. Dahmen, M. Dolbeault, and A. Somacal. Reduced order modeling for elliptic problems with high contrast diffusion coefficients. en. *ESAIM: Mathematical Modelling and Numerical Analysis*, 57(5):2775–2802, September 2023. DOI: 10.1051/m2an/2023013 (cited on pages v, vi, 10, 218, 219).

[52] A. Cohen and R. DeVore. Approximation of high-dimensional parametric PDEs. en. *Acta Numerica*, 24:1–159, May 2015. DOI: 10.1017/S0962492915000033 (cited on pages 4, 6, 8, 30, 31, 193).

[53] A. Cohen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Optimal stable nonlinear approximation. *Foundations of Computational Mathematics*, 22(3):607–648, 2022. DOI: 10.1007/s10208-021-09494-z (cited on page 138).

[54] A. Cohen, R. Devore, and C. Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's. *Anal. Appl. (Singap.)*, 9(1):11–47, 2011. DOI: 10.1142/S0219530511001728 (cited on pages 8, 30, 31, 65, 193).

[55] A. Cohen, C. Farhat, Y. Maday, and A. Somacal. Nonlinear compressive reduced basis approximation for PDEs. en. *Comptes Rendus. Mécanique*, 351(S1):1–18, September 2023. DOI: 10.5802/crmeca.191 (cited on pages v, vi, 22, 27, 218, 219).

[56] A. Cohen, O. Mula, and A. Somacal. High order recovery of geometric interfaces from cell-average data, 2024. DOI: arXiv:2402.00946 (cited on pages v, vi, 19, 27, 218, 219).

[57] N. Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015 (cited on page 179).

[58] A. Criado, J. M. Armengol, H. Petetin, D. Rodriguez-Rey, J. Benavides, M. Guevara, C. Pérez Garcıa-Pando, A. Soret, and O. Jorba. Data fusion uncertainty-enabled methods to map street-scale hourly NO_2 in Barcelona: a case study with CALIOPE-Urban v1.0. en. *Geoscientific Model Development*, 16(8):2193–2213, April 2023. DOI: 10.5194/gmd-16-2193-2023 (cited on pages 180, 197, 200).

[59] J. N. D. C. Liu. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45:503–528, 1989 (cited on page 171).

[60] W. Dahmen, M. Wang, and Z. Wang. Nonlinear reduced DNN models for state estimation. *arXiv preprint*, 2021. DOI: arXiv:2110.08951 (cited on page 179).

[61] L. Demkowicz and N. Heuer. Robust DPG method for convection-dominated diffusion problems. *SIAM Journal on Numerical Analysis*, 51(5):2514–2537, 2013. DOI: 10.1137/120862065 (cited on page 159).

[62] B. Després and H. Jourdren. Machine learning design of volume of fluid schemes for compressible flows. *Journal of Computational Physics*, 408:109275, 2020 (cited on pages 112, 130, 155).

[63] R. DeVore. Nonlinear approximation. *Acta numerica*, 7:51–150, 1998. DOI: 10.1017/S0962492900002816 (cited on pages 67, 136).

[64] R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021. DOI: 10.1017/S0962492921000052 (cited on page 136).

[65] R. DeVore, G. Petrova, and P. Wojtaszczyk. Greedy algorithms for reduced bases in Banach spaces. *Constr. Approx.*, 37(3):455–466, 2013. DOI: 10.1007/s00365-013-9186-2 (cited on pages 30, 50, 65).

[66] R. A. DeVore, R. Howard, and C. Micchelli. Optimal nonlinear approximation. *Manuscripta mathematica*, 63:469–478, 1989 (cited on pages 4, 22).

[67] M. Dolbeault, O. Mula, and A. Somacal. State estimation of urban air pollution with statistical, physical, and super-learning graph models, 2024. DOI: arXiv:2402.02812 (cited on pages v, vi, 25, 27, 218, 219).

[68] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):361–384, 2008. DOI: doi:10.1016/j.jcp.2007.12.029 (cited on page 101).

[69] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017 (cited on page 156).

[70] W. E and B. Yu. The Deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018 (cited on page 156).

[71] J. L. Eftang, A. T. Patera, and E. M. Ronquist. An "hp" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 32(6):3170–3200, 2010 (cited on pages 50, 65, 193).

[72] V. Ehrlacher, D. Lombardi, O. Mula, and F.-X. Vialard. Nonlinear model reduction on metric spaces. application to one-dimensional conservative pdes in wasserstein spaces. *ESAIM M2AN*, 54(6):2159–2197, 2020. DOI: 10.1051/m2an/2020013 (cited on page 67).

[73] M.-J. Fadili, J.-L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. *The Computer Journal*, 52(1):64–79, 2009 (cited on page 76).

[74] S. Foucart and H. Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013 (cited on pages 68, 84).

[75] B. A. Freno and K. T. Carlberg. Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations. *Computer Methods in Applied Mechanics and Engineering*, 348:250–296, 2019 (cited on page 156).

[76] S. Fresca, L. Dede, and A. Manzoni. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. *Journal of Scientific Computing*, 87(2):1–36, 2021. DOI: 10.1007/s10915-021-01462-7 (cited on page 136).

[77] F. Galarce, D. Lombardi, and O. Mula. State estimation with model reduction and shape variability. application to biomedical problems. *SIAM Journal on Scientific Computing*, 44(3):B805–B833, 2022 (cited on pages 63, 65).

[78] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Model. Simul.*, 8(5):1621–1644, 2010. DOI: 10.1137/100790112 (cited on page 33).

[79] R. Geelen, S. Wright, and K. Willcox. Operator inference for non-intrusive model reduction with quadratic manifolds. *Computer Methods in Applied Mechanics and Engineering*, 403:115717, 2023. DOI: 10.1016/j.cma.2022.115717 (cited on page 140).

[80] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, April 2006. DOI: 10.1007/s10994-006-6226-1 (cited on page 145).

[81] C. Greif and K. Urban. Decay of the Kolmogorov n-width for wave problems. *Applied Mathematics Letters*, 96:216–222, 2019 (cited on page 67).

[82] S. Grimberg, C. Farhat, R. Tezaur, and C. Bou-Mosleh. Mesh sampling and weighting for the hyperreduction of nonlinear petrov–galerkin reduced-order models with local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 122(7):1846–1874, 2021. DOI: 10.1002/nme.6603 (cited on page 136).

[83] A. Gruber, M. Gunzburger, L. Ju, and Z. Wang. A comparison of neural network architectures for data-driven reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 393:114764, 2022. DOI: 10.1016/j.cma.2022.114764 (cited on page 136).

[84] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152(2):423–456, 1999. DOI: https://doi.org/10.1006/jcph.1998.6168 (cited on page 101).

[85] B. Haasdonk. Reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems. In *Model reduction and approximation*. Volume 15, Comput. Sci. Eng. Pages 65–136. SIAM, Philadelphia, PA, 2017. DOI: 10.1137/1.9781611974829.ch2 (cited on page 30).

[86] W. Haik, Y. Maday, and L. Chamoin. Assimilation de données variationnelle et hybride pour le contrôle thermique en temps réel du procédé de fabrication additive. In *CSMA: 15ème Colloque National en Calcul des Structure*, 2022 (cited on page 136).

[87] J. K. Hammond, R. Chakir, F. Bourquin, and Y. Maday. PBDW: a non-intrusive reduced basis data assimilation method and its application to an urban dispersion modeling framework. *Applied Mathematical Modelling*, 76:1–25, 2019 (cited on pages 65, 194).

[88] A. Harten. ENO schemes with subcell resolution. *Journal of Computational Physics*, 83(1):148–184, 1989 (cited on page 100).

[89] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. *Uniformly high order accurate essentially non-oscillatory schemes, III*. Springer, 1997 (cited on page 100).

[90] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction, Second Edition*, volume 2. Springer, 2009 (cited on page 145).

[91] P. Henning, A. Målqvist, and D. Peterseim. A localized orthogonal decomposition method for semi-linear elliptic problems. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 48(5):1331–1349, 2014 (cited on page 33).

[92] J. S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2015 (cited on pages 65, 135, 179, 193).

[93] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981 (cited on page 101).

[94] T. Hrycak and K. Gröchenig. Pseudospectral fourier reconstruction with the modified inverse polynomial reconstruction method. *Journal of Computational Physics*, 229(3):933–946, 2010 (cited on page 63).

[95] S. P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. H auselmann, D. Gresch, T. M uller, A. V. Yakutovich, C. W. Andersen, F. F. Ramirez, C. S. Adorf, F. Gargiulo, S. Kumbhar, E. Passaro, C. Johnston, A. Merkys, A. Cepellotti, N. Mounet, N. Marzari, B. Kozinsky, and G. Pizzi. AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. en. *Scientific Data*, 7(1):300, September 2020. DOI: 10.1038/s41597-020-00638-4 (cited on page 27).

[96] V. V. Jikov, S. M. Kozlov, and O. A. Oleinik. *Homogenization of differential operators and integral functionals*. Springer Science & Business Media, 2012 (cited on pages 34–36).

[97] R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998 (cited on page 158).

[98] D. R. K. Veroy C. Prudhomme and T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. *Proc. 16th AIAA Computational Fluid Dynamics Conference Orlando*, 2003 (cited on page 30).

[99] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint*, 2019. DOI: arXiv:1912.00873 (cited on pages 156, 170).

[100] E. Kharazmi, Z. Zhang, and G. Karniadakis. Hp-vpinns: variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021 (cited on page 156).

[101] N. Kopteva and E. O'Riordan. Shishkin meshes in the numerical solution of singularly perturbed differential equations. *Int. J. Numer. Anal. Model*, 7(3):393–415, 2010 (cited on page 159).

[102] R. Kurtenbach, J. Kleffmann, A. Niedojadlo, and P. Wiesen. Primary NO_2 emissions and their impact on air quality in traffic environments in germany. *Environmental Sciences Europe*, 24:1–8, 2012 (cited on page 183).

[103] H. Lee and I. S. Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91(1):110–131, 1990 (cited on page 155).

[104] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020. DOI: 10.48550/arXiv.1812.08373 (cited on pages 136, 140).

[105] J. Li, X. Zhang, J. Orlando, G. Tyndall, and G. Michalski. Quantifying the nitrogen isotope effects during photochemical equilibrium between NO and NO_2: implications for $\delta^{15}N$ in tropospheric reactive nitrogen. *Atmospheric Chemistry and Physics*, 20(16):9805–9819, 2020 (cited on pages 183, 194).

[106] C. Ma, S. Wojtowytsch, L. Wu, et al. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint*, 2020. DOI: 10.48550/arXiv.2009.10713 (cited on page 165).

[107] Y. Maday and A. Patera. Reduced basis methods. *Model Order Reduction; Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., Silveira, L., Eds*:139–179, 2020. DOI: 10.1515/9783110671490-004 (cited on page 135).

[108] 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. en. *International Journal for Numerical Methods in Engineering*, 102(5):933–965, May 2015. DOI: 10.1002/nme.4747 (cited on pages 8, 15, 16, 34, 53, 65, 66, 194).

[109] Y. Maday and B. Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM J. Sci. Comput.*, 35(6):A2417–A2441, 2013. DOI: 10.1137/120873868 (cited on pages 50, 65).

[110] Y. Makovoz. Random approximants and neural networks. *Journal of Approximation Theory*, 85(1):98–109, 1996 (cited on page 78).

[111] V. Mallet, G. Stoltz, and B. Mauricette. Ozone ensemble forecast with machine learning algorithms. *Journal of Geophysical Research: Atmospheres*, 114(D5), 2009 (cited on page 181).

[112] A. Marquina and S. J. Osher. Image super-resolution by tv-regularization and bregman iteration. *Journal of Scientific Computing*, 37(3):367–382, 2008 (cited on page 76).

[113] C. A. Micchelli and T. J. Rivlin. *A survey of optimal recovery*. Springer, 1977 (cited on page 64).

[114] R. Molinaro, Y. Yang, B. Engquist, and S. Mishra. Neural inverse operators for solving PDE inverse problems. *arXiv preprint*, 2023. DOI: arXiv:2301.11167 (cited on page 179).

[115] O. Mula. Inverse problems: a deterministic approach using physics-based reduced models. In *Model Order Reduction and Applications: Cetraro, Italy 2021*, pages 73–124. Springer Nature Switzerland, Cham, 2023. DOI: 10.1007/978-3-031-29563-8_2 (cited on pages 65, 179).

[116] J. Necas, J. Dhombres, M. Maillé, and P. Robba. *Les méthodes directes en théorie des équations elliptiques*. fre. Masson Academia, Paris Prague, 1967 (cited on pages 19, 105).

[117] M. Niu, Y. Wang, S. Sun, and Y. Li. A novel hybrid decomposition-and-ensemble model based on CEEMD and GWO for short-term PM2.5 concentration forecasting. en. *Atmospheric Environment*, 134:168–180, June 2016. DOI: 10.1016/j.atmosenv.2016.03.056 (cited on page 181).

[118] W. F. Noh and P. Woodward. Slic (simple line interface calculation). In *Proceedings of the fifth international conference on numerical methods in fluid dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340. Springer, 2005 (cited on page 101).

[119] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems: Standard information for functionals*, volume 2. European Mathematical Society, 2008 (cited on page 64).

[120] M. Ohlberger and S. Rave. Reduced basis methods: success, limitations and future challenges. In *Proceedings of the Conference Algoritmy*, pages 1–12, 2016 (cited on page 67).

[121] D. Papapicco, N. Demo, M. Girfoglio, G. Stabile, and G. Rozza. The neural network shifted-proper orthogonal decomposition: a machine learning approach for non-linear reduction of hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, 392:114687, 2022. DOI: 10.1016/j.cma.2022.114687 (cited on page 136).

[122] A. Papoulis and S. Unnikrishna Pillai. *Probability, random variables and stochastic processes*. McGraw-Hill, 2002 (cited on page 141).

[123] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011 (cited on page 145).

[124] B. Peherstorfer. Breaking the kolmogorov barrier with nonlinear model reduction. *Notices of the American Mathematical Society*, 69(5):725–733, 2022. DOI: 10.1090/noti2475 (cited on page 136).

[125] Z. Peng, M. Wang, and F. Li. A learning-based projection method for model order reduction of transport problems. *Journal of Computational and Applied Mathematics*, 418:114560, 2023. DOI: 10.1016/j.cam.2022.114560 (cited on page 136).

[126] D. Peterseim and R. Scheichl. Robust numerical upscaling of elliptic multiscale problems at high contrast. *Computational Methods in Applied Mathematics*, 16(4):579–603, 2016 (cited on page 33).

[127] G. Peyré, S. Bougleux, and L. D. Cohen. Non-local regularization of inverse problems. *Inverse Problems and Imaging*, 5(2):511–530, 2011 (cited on page 76).

[128] J. E. Pilliod. *An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods*. U. of Calif., Davis, 1992 (cited on pages 68, 76).

[129] J. E. Pilliod Jr and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004. DOI: 10.1016/j.jcp.2003.12.023 (cited on pages 20, 68, 76, 101, 113, 121).

[130] A. Pinkus. *n-Widths in Approximation Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985. DOI: 10.1007/978-3-642-69894-1 (cited on page 30).

[131] E. C. Polley and M. J. Van der Laan. *Super learner in prediction*. bepress, 2010 (cited on page 180).

[132] E. G. Puckett. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In *proceedings of the fourth international symposium on Computational Fluid Dynamics*, pages 933–938, 1991 (cited on pages 20, 68, 76, 101, 111).

[133] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2016. DOI: 10.1007/978-3-319-15431-2 (cited on page 135).

[134] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019 (cited on page 156).

[135] R. A. Remmerswaal and A. E. Veldman. Parabolic interface reconstruction for 2d volume of fluid methods. *Journal of Computational Physics*, 469:111473, 2022. DOI: https://doi.org/10.1016/j.jcp.2022.111473 (cited on page 101).

[136] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *Journal of computational physics*, 141(2):112–152, 1998 (cited on page 101).

[137] H. Risken. Fokker-planck equation. In *The Fokker-Planck Equation*, pages 63–95. Springer, 1996 (cited on page 158).

[138] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*, volume 24. Springer Science & Business Media, 2008 (cited on page 158).

[139] G. M. Rotskoff, A. R. Mitchell, and E. Vanden-Eijnden. Active importance sampling for variational objectives dominated by rare events: consequences for optimization and generalization. *arXiv preprint*, 2020. DOI: arXiv:2008.06334 (cited on page 156).

[140] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008. DOI: 10.1007/s11831-008-9019-9 (cited on pages 30, 65, 193).

[141] M. Rudman. Volume-tracking methods for interfacial flow calculations. *International journal for numerical methods in fluids*, 24(7):671–691, 1997 (cited on page 101).

[142] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams. *The design and analysis of computer experiments*, volume 1. Springer, 2003 (cited on page 179).

[143] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual review of fluid mechanics*, 31(1):567–603, 1999 (cited on page 101).

[144] S. Sen. Reduced-basis approximation and a posteriori error estimation for many-parameter heat conduction problems. *Numerical Heat Transfer*, B-Fund 54:369–389, 2008. DOI: 10.1080/10407790802424204 (cited on page 30).

[145] J. W. Siegel and J. Xu. Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks. *Found. Comput. Math.*:1–57, 2022 (cited on page 78).

[146] J. Sirignano and K. Spiliopoulos. DGM: a deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018 (cited on page 156).

[147] E. M. Stein. *Singular integrals and differentiability properties of functions*. Princeton university press, 1970 (cited on page 45).

[148] V. N. Temlyakov. Nonlinear Kolmogorov widths. en. *Mathematical Notes*, 63(6):785–795, June 1998. DOI: 10.1007/BF02312773 (cited on pages 13, 50).

[149] A. Tilloy, V. Mallet, D. Poulet, C. Pesin, and F. Brocheton. BLUE-based NO_2 data assimilation at urban scale. *Journal of Geophysical Research*, 118(4):2, 031–2, 040, 2013. DOI: 10.1002/jgrd.50233 (cited on page 181).

[150] H. Tran, C. G. Webster, and G. Zhang. Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients. en. *Numerische Mathematik*, 137(2):451–493, October 2017. DOI: 10.1007/s00211-017-0878-6 (cited on pages 10, 12, 13, 30, 31, 65).

[151] S. Trehan, K. T. Carlberg, and L. J. Durlofsky. Error modeling for surrogates of dynamical systems using machine learning. *International Journal for Numerical Methods in Engineering*, 112(12):1801–1827, 2017 (cited on page 156).

[152] M. J. Van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007 (cited on page 180).

[153] S. Volkwein. Proper orthogonal decomposition: theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, January 2012 (cited on page 30).

[154] L. Wang and J. M. Mendel. Structured trainable networks for matrix algebra. In *1990 IJCNN International Joint Conference on Neural Networks*, pages 125–132. IEEE, 1990 (cited on page 155).

[155] Z. Wang, J. Chen, and S. C. Hoi. Deep learning for image super-resolution: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020 (cited on page 76).

[156] G. Welper. Transformed snapshot interpolation with high resolution transforms. *SIAM J. Sci. Comput.*, 42(4):A2037–A2061, 2020. DOI: 10.1137/19M126356X (cited on page 67).

[157] G. Weymouth and D. K.-P. Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *Journal of Computational Physics*, 229(8):2853–2865, 2010. DOI: doi:10.1016/j.jcp.2009.12.018 (cited on page 101).

[158] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA journal*, 40(11):2323–2330, 2002 (cited on page 30).

[159] A. Ženíšek. Extensions from the Sobolev spaces H^1 satisfying prescribed Dirichlet boundary conditions. *Appl. Math.*, 49(5):405–413, 2004. DOI: 10.1023/B:APOM.0000048120.75291.a5 (cited on page 46).

[160] K. Zhang, D. Tao, X. Gao, X. Li, and J. Li. Coarse-to-fine learning for single-image super-resolution. *IEEE transactions on neural networks and learning systems*, 28(5):1109–1122, 2016 (cited on page 76).

[161] Z. Zou, D. Kouri, and W. Aquino. An adaptive local reduced basis method for solving pdes with uncertain inputs and evaluating risk. *Computer Methods in Applied Mechanics and Engineering*, 345:302–322, 2019 (cited on pages 50, 65, 193).

Model reduction for forward simulation and inverse problems: towards non-linear approaches

Abstract

Model reduction is a technique used to compute fast and accurate approximations of physical systems' states when they are described through parametric *Partial Differential Equations* (PDEs). In the classical setting a linear subspace is carefully built, in an *offline* stage, using a set of high resolution descriptions of possible states of the system of interest. Afterwards the subspace is used to quickly and accurately solve *forward* or *inverse* problems. It is known that these strategies can approximate well the solution of elliptic PDEs but they fail on hyperbolic PDEs or when states present jump discontinuities. In this context, this thesis focuses on developing efficient non-linear strategies to tackle the limitations of linear approximation spaces.

Chapter 2 extends the approximation guarantees offered by linear spaces for the stationary diffusion equation when extreme levels of contrast in the diffusivity constants are possible.

Chapter 3 presents a theoretical framework to analyse the effectiveness of non-linear strategies for *inverse* problems while Chapter 4 focuses on the practical implementation of high-order techniques to locally reconstruct interfaces from cell average data. In Chapter 5, we show a method to accelerate the reconstruction of 1d characteristic functions by a machine learning strategy trained to learn a mapping from lower order Fourier coefficient values to higher order ones. In Chapter 6, we turn the attention to another learning technique, known as Physics Informed Neural Networks (PINNs), to tackle a linear advection-diffusion equation when the diffusivity vanishes and shocks appear.

Finally, in Chapter 7, we apply a combination of linear and non-linear methods to a real case scenario in which the objective is to predict the pollution on every point in a city using heterogeneous sources of data like temporal pollution series on specified locations, the geometry of the streets, and Google Maps traffic information.

Chapters 2, 3, 5 and 6 are based on the published articles [51, 50, 55, 20] respectively while Chapters 4 and 7 are based on the submitted articles [56, 67].

Keywords: Non-linear approximation, Reduced order modelling, Numerical approximation

---

Laboratoire Jacques-Louis Lions

Sorbonne Université – Campus Pierre et Marie Curie – 4 place Jussieu – 75005 Paris – France

Réduction de modèle pour des problèmes directs et inverses: vers des approches non linéaires

Résumé

La réduction de modèle est une technique utilisée pour calculer des approximations rapides et précises des états de systèmes physiques, décrits par des *Équations* aux *Dérivées Partielles* (EDP) paramétriques. Dans le cadre classique, un sous-espace linéaire est construit dans une étape *offline* en utilisant un ensemble de descriptions à haute résolution des états possibles du système d'intérêt. Ensuite, le sous-espace est utilisé pour résoudre rapidement et avec précision des problèmes *directes* ou *inverses*. Il est connu que ces stratégies peuvent bien approximer la solution des EDP elliptiques avec peu d'éléments de base mais échouent sur les EDP hyperboliques ou lorsque les états présentent des discontinuités. Dans ce contexte, cette thèse se concentre sur le développement de stratégies non linéaires efficaces pour aborder les limitations des espaces linéaires.
Le Chapitre 2 étend les garanties d'approximation offertes par les espaces linéaires pour l'équation de diffusion stationnaire pour des niveaux extrêmes de contraste dans les champs de diffusion.
Le Chapitre 3 présente un cadre théorique pour analyser l'efficacité des stratégies non linéaires pour les problèmes *inverses* tandis que le Chapitre 4 se concentre sur la mise en œuvre pratique des techniques d'ordre élevé pour reconstruire localement des interfaces à partir des moyennes. Dans le Chapitre 5, nous montrons une méthode pour accélérer la reconstruction de fonctions caractéristiques en 1d par une stratégie d'apprentissage automatique entraînée à fournir une correspondance entre les valeurs des coefficients de Fourier d'ordre inférieur et celles d'ordre supérieur. Dans le Chapitre 6, nous portons notre attention sur une autre technique d'apprentissage connue sous le nom de réseaux neuronaux informés par la physique (PINN) pour traiter une équation de transport-diffusion linéaire lorsque la diffusivité tend vers zéro et que des chocs apparaissent.
Enfin, dans le Chapitre 7, nous appliquons une combinaison de méthodes linéaires et non linéaires à un scénario réel dans lequel l'objectif est de prédire la pollution en tout point d'une ville en utilisant des sources de données hétérogènes telles que des séries temporelles de pollution sur des emplacements spécifiques, la géométrie des rues et les informations de traffic de Google Maps.
Les Chapitres 2, 3, 5 et 6 sont basés sur les articles publiés [51, 50, 55, 20] tandis que les Chapitres 4 et 7 sont basés sur les articles soumis [56, 67].

Mots clés : Approximation non-lineaire, Modeles reduites, Approximation numérique

---