université PARIS-SACLAY

Natural gradients and kernel methods for Physics Informed Neural Networks (PINNs)

Gradients naturels et méthodes à noyaux pour les réseaux de neurones informés par la physique (PINNs)

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire Interdisciplinaire des Sciences du Numérique (Université Paris-Saclay, CNRS)**
sous la direction de **Cyril FURTLEHNER**, chargé de recherche, INRIA Saclay

Thèse soutenue à Paris-Saclay, le 08 décembre 2025, par

Nilo SCHWENCKE

Composition du jury

Membres du jury avec voix délibérative

Claire BOYER Professeure, Université Paris-Saclay	Présidente
Emmanuel FRANCK Chargé de recherche, HDR, INRIA Nancy-Grand Est	Rapporteur & Examinateur
Olga MULA Professeure, Universität Wien	Rapporteure & Examinatrice
Francis BACH Directeur de recherche, INRIA Paris	Examinateur
Victor MICHEL-DANSAC Chargé de recherche, INRIA Nancy-Grand Est	Examinateur

THESE DE DOCTORAT
NNT : 2025UPASG087

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Gradients naturels et méthodes à noyaux pour les réseaux de neurones informés par la physique (PINNs).....
Mots clés: PINNs, NTK, EDPs, Gradient Naturel

Résumé: Les Physics-Informed Neural Networks (PINNs) ont émergé ces dernières années comme un paradigme prometteur pour la résolution d'équations aux dérivées partielles (EDP), en intégrant directement les contraintes physiques dans l'entraînement des réseaux de neurones. Malgré leur attrait conceptuel et leur adoption rapide dans de nombreux domaines scientifiques et industriels, les PINNs souffrent souvent d'un manque de précision et de robustesse comparés aux méthodes numériques classiques. Ces limitations ont suscité un large corpus de travaux visant à améliorer les algorithmes, à optimiser les stratégies d'apprentissage, et à analyser plus finement leurs comportements théoriques.

L'objectif de cette thèse est de faire progresser ces recherches selon deux directions complémentaires. D'une part, sur le plan algorithmique, nous visons à concevoir des schémas d'entraînement plus efficaces pour les PINNs, en hybridant des outils issus des méthodes à noyaux et des gradients naturels. D'autre part, sur le plan théorique, nous cherchons à ancrer les PINNs dans un cadre mathématique plus rigoureux. En les inscrivant dans le langage des espaces de Hilbert à noyaux reproduisants (RKHS), de la théorie des opérateurs et de l'analyse spectrale, nous clarifions leur structure, les relions aux méthodes variationnelles, et contribuons à une meilleure compréhension mathématique des PINNs.

Après avoir introduit les objets fondamentaux (réseaux de neurones et PINNs et RKHS), nous présentons nos contributions, regroupées en deux articles.

1. **ANaGRAM.** Nous établissons une nouvelle connexion entre méthodes à noyaux et gradient naturel, menant à la notion de *gradient naturel empirique*. Nous introduisons *l'algorithme ANaGRAM*, un algorithme d'entraînement exploitant systématiquement cette relation. Cette approche améliore les performances numériques tout en fournissant un lien conceptuel entre gradient naturel et fonctions de Green.
2. **AMStraMGRAM.** Notre deuxième contribution étudie la dynamique d'ANaGRAM. Nous analysons le rôle de la régularisation par cutoffs spectraux, que nous réinterprétons en termes d'imposition d'isométries et que nous relions à la théorie des fonctions de Green et des noyaux reproduisants. Ce point de vue éclaire en quoi les cutoffs renforcent stabilité et précision, et motive un nouvel algorithme à *cutoffs adaptatifs*, ajustant dynamiquement la régularisation au cours de l'entraînement : *AMStraMGRAM*.

Notons enfin que nous ouvrons au cours de notre exposé, une perspective vers le formalisme plus général des noyaux de Schwartz, qui englobe à la fois les noyaux reproduisants et les Riggings de Hilbert. Ce cadre offre des pistes prometteuses pour inscrire les PINNs et les méthodes de gradient naturel dans un contexte d'analyse fonctionnelle encore plus riche.

Ces contributions visent à la fois à améliorer l'efficacité pratique et à renforcer les fondements théoriques des PINNs. Elles soulignent que les points de vue géométriques et fondés sur les noyaux ne sont pas seulement des outils pour accroître les performances, mais constituent aussi un langage adéquat pour intégrer les PINNs dans le paysage plus large de l'analyse numérique et fonctionnelle.

Title: Natural gradients and kernel methods for Physics Informed Neural Networks (PINNs).....
Keywords: PINNs, NTK, PDEs, Natural Gradient

Abstract: Physics-Informed Neural Networks (PINNs) have emerged in recent years as a promising paradigm for solving partial differential equations (PDEs) by embedding physical constraints directly into the training of neural networks. Despite their conceptual appeal and rapid adoption across scientific and engineering domains, PINNs often suffer from limited accuracy and robustness compared to classical numerical methods. These limitations have motivated a rich body of research on algorithmic refinements, improved training strategies, and theoretical analyses of their behavior. The objective of this dissertation is to advance this line of research along two complementary directions. On the algorithmic side, our goal is to design more efficient and accurate training schemes for PINNs by hybridizing tools from kernel methods and natural gradient optimization. On the theoretical side, we aim to anchor PINNs within a rigorous mathematical framework. By situating them in the language of reproducing kernel Hilbert spaces (RKHS), operator theory, and spectral analysis, we seek to clarify their structure, relate them to existing approximation and variational methods, and foster a deeper mathematical understanding of PINNs.

After introducing the fundamental objects (neural networks and RKHS) in preliminary chapters, we present our contributions across two papers.

1. ANaGRAM. We first establish a novel connection between kernel methods and natural gradient optimization, leading to the notion of the *empirical natural gradient*. Building upon this framework, we introduce the *ANaGRAM algorithm*, a new PINN training scheme that systematically exploits this connection. This yields improved numerical performance while also providing a principled link between natural gradient methods and Green's functions.

2. AMStraMGRAM. Our second contribution studies the dynamics of ANaGRAM. We analyze the role of regularization by spectral cutoffs, which we reinterpret in terms of enforcing isometries and relate to the theory of Green's functions and reproducing kernels. This perspective clarifies why cutoffs improve stability and accuracy, and motivates a new algorithm with *adaptive cutoffs*, which adjusts the degree of regularization dynamically during training. We call this refined method *AMStraMGRAM*.

Throughout the manuscript, we also develop a broader perspective on the general formalism of Schwartz kernels, which encompasses both reproducing kernels and Hilbert Riggings. This framework offers promising directions for situating PINNs and natural gradient methods within an even richer functional-analytic setting. Taken together, these contributions advance both the practical efficiency and the theoretical foundations of PINNs. They underscore that kernel-based and geometric viewpoints are not only effective tools for improving performance, but also provide the right language for integrating PINNs into the broader landscape of numerical and functional analysis.

Remerciements

Au moment de mettre un point final à ce manuscrit, je souhaite exprimer ma gratitude aux nombreuses personnes qui ont rendu cette thèse possible.

* * *

Aux membres de mon jury d'abord, en particulier à Olga Mula et Emmanuel Franck, qui ont accepté la laborieuse tâche de relire mon manuscrit, mais aussi à Claire Boyer, qui a accepté d'en assurer la présidence, ainsi qu'à Francis Bach et Victor-Michel Dansac, qui ont accepté de venir jusqu'au plateau de Saclay pour y participer.

À mon directeur de thèse, Cyril, ensuite, qui a accepté de se lancer dans l'aventure en m'accordant sa confiance et en me laissant une grande autonomie, à la fois dans la conduite de mes recherches, mais également dans mon organisation au jour le jour, ce dont je lui suis infiniment redevable !

To Alena and Roland, for your enthusiasm, tenacity, and trust, which made it a real pleasure to work with you both!

À Cyriaque pour ton travail acharné et les heures passées ensemble à échanger et à s'arracher les cheveux, mais également pour ton amitié et ta grande générosité !

À Julien et Sheng pour votre enthousiasme, votre travail rigoureux et votre bonne humeur ! C'est un plaisir de travailler et de séjourner avec vous !

* * *

À Michèle et Marc pour m'avoir mis le pied à l'étrier dans le monde passionnant de la recherche et m'avoir appris à y faire mes premières chevauchées.

À Sylvain et Guillaume pour votre soutien, vos encouragements et vos conseils précieux, mais également à Sergio, Lionel, Onofrio, François et Cécile pour les discussions scientifiques, mais aussi politiques, toujours enrichissantes !

À l'ensemble des doctorant-es et postdoctorant-es de l'équipe TAU pour leur bonne humeur, leur appétit du jeu et leur franche camaraderie, en particulier Théo, Stella, Manon et Stéphane pour les aventures viennoises, Nicolas Béreux (et Stéphane encore !) pour les aventures singapouriennes, Anaclara, Solal, Alex, Trang et Barbara pour les séances d'escalade à Arkose, Sébastien et Bruno pour avoir partagé avec moi le bureau des années durant, et Théo parfois, le temps d'une sieste, Adrien et Éléonore pour leurs soirées jacuzzi mémorables, mais aussi Thomas, Théofanis, Victor, Emmanuel, Alice, Matthieu,

Luca, Dylan, Jose, Joachim, Audrey, Marine, Roman, Théophile, Lucas, Romain, Victor et Théo. Grâce à vous toustes, TAU est devenue en quelque sorte ma deuxième maison !

An Willy Dörfler, der mich so freundlich und oft in Karlsruhe aufgenommen und mich während meiner gesamten Laufbahn unterstützt hat, an Mehdi, der mich in die Feinheiten der PINNs eingeführt hat, an Tim für seine Begeisterung, seine unermüdliche Arbeit und seine stets fruchtbare Neugier, aber auch an die gesamte Arbeitsgruppe *Numerik partieller Differentialgleichungen*, die mich so herzlich aufgenommen hat : Raphael, Stefan, Joshua, Dustin und Tino, und natürlich Roland !

* * *

À Caroline Appert, Anne Vilnat, Emmanuelle Frenoux et Alain Denise, et plus généralement aux membres de l'ED STIC et du LISN, dont les efforts combinés m'ont permis de mener à bien ce travail malgré les embûches et les imprévus nombreux !

À Elisabeth, pour avoir aidé à me remettre en selle.

À Chantal pour ta sollicitude et ton soutien constant, qui m'ont été d'une aide plus que précieuse dans les moments cruciaux.

* * *

À Armand, Margarita, Louis et Sylvain, membres fondateurs et réguliers du très distingué club Tan Do.

Aux camarades de Melun pour le travail patient de terrain, mais aussi pour tous les moments chaleureux partagés ; Julie, pour deux ans et demi de colocation et ton amitié ; Arnaud, Rémy, Fatiha, Cécile, Éric, Cédric, Mehdi, Véronique, et toustes les autres !

Aux camarade de Paris-Saclay, pour les heures sur le terrain : Francis, Jacques-Olivier, Nadia, Julien, Samuel, Anne, Clara, Cédric, et les autres !

Aux camarades du Printemps Giffois, pour contribuer à faire souffler ce vent printanier : Anne-Marie, Frank, Christophe, Michel, Jean, Annick, Isabelle, Yoann, Roger, Marie-Hélène, Claude, Francis, Louis et les autres !

Aux camarades de X-Alternative pour les idées portées ensemble : Samuel, Régis, Zako, Virginie, Alexandre, Michel, Yves, Daniel, et toustes les autres !

Aux camarades de la SphinX pour les combats menés ensemble : Denis, Matthieu, Clara, Lucas, Quentin, Rubing, Léonard, Amaury, Thomas, et toustes les autres !

Aux camarades de Coll·doc, pour cette aventure vivifiante et exaltante, bien que trop courte : Raphaël, Anatole, Axelle, Léa, Lug, Nicolas, et toustes les autres !

Aux doctorant-es du LJLL, qui m'ont accueilli avec bonne humeur et bienveillance, et avec qui j'ai eu grand plaisir à échanger, scientifiquement ou non : Matthieu, Yvone, Elena, Pierre, Antoine, et toustes les autres.

* * *

À Martin et Christina pour leur écoute et leur soutien constants dans les moments difficiles, ainsi que pour notre inénarrable séjour à Brassac.

À Jean-Philippe, Jade et Yann, qui me font l'honneur de leur amitié depuis plus de trente ans maintenant !

À l'inaltérable groupe des geeks : Swann *aka* armabon, Laure *aka* elre, Mathieu *aka* machin300, Margaux *aka* gomar, Quentin *aka* wc3pro, Adrien *aka* gothics13, Manon *aka* man, Stéphane *aka* tétane, Nico *aka* xxx(x)fr, Marion *aka* mainlinh, Étienne, Thibault *aka* drokabelly, Antoine *aka* blueonion, Pierre *aka* pir, Marc *aka* skorm et Colin *aka* agone. Vous êtes ce roc inamovible sur lequel toujours reprendre pied, quelle que soit l'intensité de la tempête !

À Quentin et Blanche pour votre accueil, et bien sûr à Charlotte *aka* KP pour sa mignonitude si attendrissante !

À Diego et Alicia pour votre amitié, les heures de discussions et de jeux, vos conseils et vos interventions, fermes quand il le fallait !

A María, por tu amistad, tu apoyo y tu gran corazón, que me han ayudado mucho y me han hecho más feliz !

To Shorouk and Agustin, for your friendship, positivity, scientific conversations, vacations, and all the great times we had together!

* * *

Aux citoyen-nes français-es qui ont financé l'ensemble de mes études, dont ma thèse. Vivre dans une société disposant (encore) d'un service public fort est une véritable source de progrès, d'égalité et d'émancipation, qu'il faut choyer et cultiver encore et encore !

* * *

À Audrey, rencontrée au cours de cette aventure. Tu contribues grandement à mon bonheur et à mon complet épanouissement, et j'espère que tu resteras à mes côtés avec Lila pour de très nombreuses années encore !

Un enorme agradecimiento a mi familia, por su apoyo grandísimo y su amor, en particular a mi mami Alheli, a mi hermano Guillen, a mi abuela Nani, a Angela, a mis tías Ximena, Rocío, Rodrigo, Coral e Irene, a mis primos Milena, Alice, Esteban, Lisa, Gabriel, Celeste, Mara e Ismael.

A mi abuelo Jaime, también conocido como Yaye, finalmente, a quien debo en gran parte mi curiosidad científica, y que desgraciadamente no llegó a poder ver el final de esta tesis que le quiero dedicar.

* * *
* *
*

À mon grand-père, Yaye

Introduction détaillée

Les équations aux dérivées partielles (EDP) sont au cœur des sciences et de l'ingénierie modernes. Elles décrivent une grande variété de phénomènes physiques, allant de la mécanique des fluides (équations de Navier–Stokes) à la mécanique quantique (équation de Schrödinger), en passant par l'électromagnétisme (équations de Maxwell), l'élasticité ou encore la propagation des ondes (équations de Helmholtz et des ondes). D'un point de vue mathématique, une EDP modélise l'équilibre de quantités fondamentales telles que la masse, la quantité de mouvement ou l'énergie, dans l'espace et dans le temps, ce qui en fait le langage universel de la modélisation physique.

Néanmoins, les EDP ne sont en général *pas solubles par des moyens algébriques*. Cette impossibilité peut être justifiée de manière rigoureuse par des arguments du même type que ceux qui apparaissent en théorie de Galois (Singer, 2009), laquelle classe les équations selon leur solubilité par radicaux : de la même manière que la plupart des équations polynomiales de degré supérieur n'admettent pas de solution en forme close, la plupart des EDP non linéaires ne possèdent pas de solution analytique explicite.

Par conséquent, l'étude des EDP repose le plus souvent sur des *méthodes indirectes*. Plutôt que de chercher à exprimer des solutions explicites, on s'intéresse à leurs propriétés qualitatives : existence, unicité, régularité et stabilité. L'analyse fonctionnelle (Brezis, 2011), la théorie spectrale (Edmunds and Evans, 2018) et les méthodes variationnelles (Gelfand et al., 2000) sont ainsi devenues des outils centraux pour caractériser les espaces de solutions et comprendre le comportement des EDP, même en l'absence de formules explicites.

En pratique, cependant, la seule manière d'obtenir des solutions concrètes pour des EDP générales est de recourir à des *approximations numériques* (Iserles, 2009). Les approches classiques incluent les méthodes aux différences finies (Strikwerda, 2004), aux éléments finis (Zienkiewicz and Taylor, 2013) et aux volumes finis (Moukalled et al., 2016), qui reposent toutes sur la discrétisation du domaine continu par un maillage et sur le remplacement des dérivées par des opérateurs algébriques manipulables par des outils d'algèbre linéaire ou des schémas itératifs. Une classe d'approches légèrement différente est constituée par les méthodes spectrales (Shen et al., 2011), qui n'approximent pas la fonction inconnue par une discrétisation locale, mais en la projetant sur un sous-espace de dimension finie engendré par des fonctions de base globales, telles que des fonctions trigonométriques ou des polynômes orthogonaux (Legendre, Chebyshev, etc.). Dans ce cadre, la différentiation et l'intégration peuvent être représentées exactement dans la base choisie.

Malgré leur succès remarquable et leur maturité théorique, toutes ces méthodes partagent néanmoins plusieurs limitations communes :

- **Malédiction de la dimension :** le coût de calcul croît exponentiellement avec la dimension d du domaine.
- **Géométries complexes :** les méthodes basées sur des maillages nécessitent une construction soigneuse de la grille, ce qui devient difficile pour des domaines irréguliers ou évoluant dans le temps.
- **Non-linéarités et rigidité :** les EDP fortement non linéaires ou présentant des phénomènes multi-échelles (par exemple les écoulements turbulents) requièrent souvent des discrétisations extrêmement fines pour rester stables et précises, ce qui entraîne des coûts de calcul prohibitifs.
- **Manque de flexibilité :** les solveurs classiques sont étroitement liés à des discrétisations spécifiques, ce qui rend difficile la réutilisation d'informations entre différentes EDP, conditions aux limites ou régimes paramétriques.
- **Intégration limitée des données :** les solveurs traditionnels supposent des modèles physiques entièrement spécifiés et ne permettent pas d'incorporer naturellement des données partielles ou bruitées. Bien que des techniques spécialisées telles que l'assimilation de données (Asch et al., 2016) existent, elles sont en général spécifiques à un problème donné et consistent à corriger la sortie du solveur plutôt qu'à intégrer données et physique dans un cadre unifié.

Dans de nombreuses applications modernes, telles que la modélisation climatique (Stocker, 2011), l'imagerie médicale (Paulsen and Moeslund, 2020) ou la mécanique des fluides (Wendt, 2009), les équations gouvernantes ne sont cependant que partiellement connues, ou doivent être inférées à partir d'observations. De plus, les géométries sous-jacentes sont souvent complexes, multi-échelles, voire seulement partiellement spécifiées, tandis que les modèles impliquent un grand nombre de variables aux interactions non linéaires complexes, conduisant à des problèmes de très grande dimension.

Ces difficultés ont motivé le développement d'approches plus flexibles, sans maillage, et adaptatives aux données, inspirées par les avancées récentes en apprentissage automatique (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016; Bach, 2024). Cela a donné naissance à un domaine en pleine expansion, connu sous le nom de *Machine Learning Scientifique* (SciML).

Cette discipline émergente vise à combiner la flexibilité et la puissance expressive de l'apprentissage automatique moderne avec la rigueur et la structure

de la modélisation physique, afin de relier données et physique de manière principielle. De manière générale, ces approches peuvent être regroupées en trois grandes catégories :

- **Modélisation guidée par les données.** Cette ligne de recherche vise à découvrir directement à partir des données les équations gouvernantes ou les lois dynamiques, sans supposer de connaissance préalable de la physique sous-jacente. Parmi les exemples typiques figurent l'identification parcimonieuse de dynamiques non linéaires (SINDy ; Brunton et al., 2016; Fiorini et al., 2025) et les équations différentielles ordinaires neuronales (Neural ODEs ; Chen et al., 2018), dont l'objectif est d'apprendre l'opérateur d'évolution expliquant au mieux les trajectoires observées.

- **Apprentissage d'opérateurs.** Une seconde branche, en plein essor, du SciML consiste à apprendre des applications entre espaces de fonctions entiers plutôt que des valeurs ponctuelles (Boullé and Townsend, 2024; Kovachki et al., 2024), dans le but d'approximer l'opérateur solution associé à une famille d'EDP. Une approche particulièrement influente est celle des *Neural Operators* (Kovachki et al., 2021), parmi lesquels le DeepONet (Lu et al., 2021a) et le Fourier Neural Operator (FNO ; Li et al., 2020) ont acquis une grande popularité. Une fois entraînés, ces modèles permettent une inférence rapide pour de nouvelles données d'entrée, conditions aux limites ou paramètres, et peuvent être vus comme des opérateurs substituts appris.

- **Apprentissage automatique informé par la physique (PIML).** Enfin, cette famille de méthodes intègre des lois physiques connues, typiquement des équations différentielles ou des principes de conservation, directement dans le processus d'apprentissage, sous forme de contraintes souples ou fortes. L'approche la plus représentative de cette catégorie est celle des *Physics-Informed Neural Networks* (PINNs ; Raissi et al., 2019b), qui approximent les solutions d'EDP en minimisant à la fois le résidu des équations gouvernantes et l'erreur sur les conditions aux limites et initiales. De nombreuses extensions et raffinements ont été proposés ces dernières années, constituant un champ de recherche actif et en pleine croissance (Cuomo et al., 2022; Meng et al., 2025).

Ces trois paradigmes — découverte guidée par les données, apprentissage d'opérateurs et régularisation informée par la physique — constituent les fondements du SciML moderne. Pris ensemble, ils offrent un éventail de stratégies pour intégrer données, physique et calcul dans un cadre mathématique et algorithmique unifié.

Notons que ces trois perspectives ne sont en aucun cas disjointes. Ces dernières années, un nombre croissant de travaux ont cherché à hybrider ces paradigmes, en combinant leurs forces complémentaires. Par exemple, plusieurs méthodes associent l'apprentissage d'opérateurs à une modélisation informée par la physique, telles que le *Physics-Informed Neural Operator* (PINO ; Li et al., 2021), qui incorpore directement les résidus d'EDP dans la fonction de coût de l'apprentissage d'opérateur, ou encore le *Physics-Enhanced Neural Operator* (PENO ; Chen et al., 2024), qui intègre des connaissances physiques directement dans l'architecture de l'opérateur neuronal.

Des stratégies hybrides ont également émergé à l'interface entre l'apprentissage informé par la physique et la modélisation guidée par les données. Des extensions des PINNs ont ainsi été développées pour traiter des systèmes partiellement connus ou des données observationnelles bruitées, en les couplant à des techniques d'assimilation de données ou d'identification de systèmes (Raissi et al., 2020; Kissas et al., 2020; Nguyen et al., 2022a). Dans une approche probabiliste, Yang et al. (2021) ont proposé un cadre bayésien unifié combinant données expérimentales et a priori physiques, permettant ainsi une quantification de l'incertitude à la fois sur le modèle et sur les données.

Enfin, les travaux récents de Doumèche et al. (2024b,a, 2025) introduisent le cadre du *Physics-Informed Kernel Learning* (PIKL), qui s'appuie sur des solveurs à noyaux afin de fournir à la fois une formulation algorithmique opérationnelle et une analyse statistique rigoureuse des bénéfices apportés par l'incorporation d'informations physiques par rapport à des approches purement guidées par les données.

De manière similaire, le SciML n'est pas isolé de l'analyse numérique classique. L'un des ponts les plus notables entre ces deux domaines est fourni par les méthodes de *Model Order Reduction* (MOR ; Schilders et al., 2008), qui entretiennent des liens conceptuels profonds avec l'apprentissage d'opérateurs. Des travaux récents ont cherché à enrichir ces approches classiques à l'aide de techniques d'apprentissage profond ; par exemple, Menier (2024) étendent les cadres traditionnels de MOR à l'aide d'architectures neuronales, tandis que Somacal (2024) établissent des connexions entre les méthodes de bases réduites (Cohen and DeVore, 2015; Benner et al., 2017) et le SciML, y compris dans des extensions informées par la physique.

Un effort de recherche considérable a également été consacré aux stratégies d'échantillonnage pertinentes à la fois pour le *Scientific Machine Learning* (SciML) et pour l'analyse numérique classique (Bon et al., 2025; Haberstich et al., 2022; Gruhlke et al., 2024; Nouy and Michel, 2024; Dolbeault and Cohen, 2022a; Cohen and Migliorati, 2017; Dolbeault and Cohen, 2022b).

Dans une direction différente, Nastorg (2024) proposent une approche guidée par les données fondée sur les *Graph Neural Networks* (GNNs ; voir Wu et al., 2020) pour la décomposition de domaines (Chan and Mathew, 1994), appliquée

à des équations elliptiques.

D'autres stratégies hybrides combinent des solveurs neuronaux avec des schémas numériques traditionnels. Par exemple, la *Neural Semi-Lagrangian Method* (Franck et al., 2025) introduit un schéma discrétisé en temps dans lequel les réseaux neuronaux approximent la composante spatiale par régression du flot des caractéristiques, calculé via une intégration de type Runge–Kutta (voir, par exemple, Iserles, 2009). Une autre stratégie mixte consiste à approximer dans un premier temps une solution grossière à l'aide d'un PINN, puis à raffiner l'erreur résiduelle à l'aide d'une méthode aux éléments finis (FEM), comme proposé dans Barucq et al. (2025).

Dans cette thèse, nous nous concentrons plus particulièrement sur le cadre des *Physics-Informed Neural Networks* (PINNs). S'appuyant sur les travaux pionniers de Dissanayake and Phan-Thien (1994); Lagaris et al. (1998), cette approche a été revisitée par Berg and Nyström (2018) à la suite de l'essor de l'apprentissage profond (LeCun et al., 2015), puis largement popularisée par Raissi et al. (2019b), qui ont introduit l'utilisation décisive de la différentiation automatique (Baydin et al., 2018) pour le calcul des opérateurs différentiels requis.

L'idée centrale consiste à représenter la solution inconnue $u : \Omega \to \mathbb{R}$ par un réseau de neurones $u_{\boldsymbol{\theta}}$, paramétré par $\boldsymbol{\theta} \in \mathbb{R}^P$, et à entraîner ce réseau de sorte qu'il satisfasse à la fois l'EDP gouvernante et les conditions aux limites associées. Concrètement, on définit une fonction de perte mesurant l'écart entre le résidu de l'EDP et les conditions aux limites, évaluée en des points de collocation échantillonnés aléatoirement :

$$\ell(\boldsymbol{\theta}) = \frac{1}{2S_D} \sum_{i=1}^{S_D} \left| D[u_{\boldsymbol{\theta}}](x_i^D) - f(x_i^D) \right|^2 + \frac{1}{2S_B} \sum_{j=1}^{S_B} \left| B[u_{\boldsymbol{\theta}}](x_j^B) - g(x_j^B) \right|^2, \quad (0.1)$$

où D et B désignent respectivement les opérateurs différentiel et de bord. Les paramètres $\boldsymbol{\theta}$ du réseau sont ensuite optimisés afin de minimiser $\ell(\boldsymbol{\theta})$, fournissant ainsi une approximation de la solution qui satisfait l'EDP sur les points de collocation choisis.

Au fil des années, de nombreuses variantes des PINNs ont été proposées, se distinguant principalement par la manière dont les contraintes physiques sous-jacentes sont imposées. Parmi les premières alternatives figure la *Deep Ritz Method* (DRM ; E and Yu, 2017), dans laquelle la fonction de perte est définie à partir de la formulation variationnelle (énergétique) de l'EDP plutôt que de son résidu. D'autres approches reposent sur des formulations de type Galerkin ou Petrov–Galerkin, où le résidu est multiplié par une fonction test avant intégration. Selon le choix de cette fonction test, on obtient différentes formulations, qui sont résumées dans la Table 1.1 ci-dessous.

Table 1: Variantes des PINNs fondées sur différents choix de fonctions test (adapté de Kharazmi et al., 2021).

Type de fonction test	**Méthode**
Dirac	PINNs (Raissi et al., 2019b)
Résidu de l'EDP	DGM (Sirignano and Spiliopoulos, 2018)
Polynômes globaux	VPINNs (Kharazmi et al., 2019)
Polynômes par morceaux	VarNet (Khodayi-Mehr and Zavlanos, 2020)
Polynômes non recouvrants	*hp*-VPINNs (Kharazmi et al., 2021)

Plusieurs autres extensions ont été proposées dans ce cadre général, notamment les *Conservative PINNs* (CPINNs ; Jagtap et al., 2020) et les *Extended PINNs* (XPINNs ; Jagtap and Karniadakis, 2021), qui visent à subdiviser le domaine global en sous-domaines plus petits et à entraîner des PINNs indépendants sur chacun d'eux, la cohérence globale n'étant assurée que par des conditions d'interface entre sous-domaines. Une autre ligne de recherche, connue sous le nom de *greedy training* (Siegel et al., 2023; Wang and Lai, 2024), consiste à entraîner les PINNs de manière séquentielle, chaque nouveau réseau étant optimisé pour minimiser le résidu laissé par le précédent. L'approximation finale est alors obtenue comme la somme de tous les réseaux entraînés individuellement, affinant progressivement la solution globale.

L'ensemble de ces techniques a depuis été appliqué à un large éventail de domaines et a donné lieu à des succès remarquables dans des champs aussi variés que la mécanique des fluides (Raissi et al., 2019c,a; Sun et al., 2020; Raissi et al., 2020; Jin et al., 2021; de Wolff et al., 2021), le bio-ingénierie (Sahli Costabal et al., 2020; Kissas et al., 2020), les problèmes à frontière libre (Wang and Perdikaris, 2021), l'optimisation de formes (Bélieres–Frendo et al., 2024) ou encore l'apprentissage par renforcement (Mukherjee and Liu, 2023; Shilova et al., 2024). À titre d'illustration paradigmatique de leur potentiel, les PINNs ont même été appliqués avec succès à des problèmes de dimension allant jusqu'à un million (Shi et al., 2024), un travail qui a reçu le *Best Paper Award* à NeurIPS 2024[1].

Cela étant dit, ces résultats doivent être nuancés, car de nombreux défis subsistent. Une limitation majeure est que les PINNs standards peinent souvent à atteindre les niveaux de précision requis en calcul scientifique, en particulier dans les régimes où les solveurs reposant sur un maillage excellent,

[1] https://blog.neurips.cc/2024/12/10/announcing-the-neurips-2024-best-paper-awards/

qu'il s'agisse des méthodes aux éléments finis (FEMs ; Grossmann et al., 2024) ou des schémas fondés sur les différences finies (Karnakov et al., 2022).

De nombreuses difficultés et modes d'échec ont été identifiés, permettant d'éclairer ces limitations :

- **Paysages d'optimisation fortement non convexes.** La fonction de perte des PINNs implique généralement les résidus d'opérateurs différentiels non linéaires, ce qui conduit à des problèmes d'optimisation fortement non convexes et mal conditionnés. En conséquence, l'optimisation stagne fréquemment dans des minima locaux médiocres ou sur des plateaux, en particulier pour des systèmes de grande dimension (Rathore et al., 2024; Krishnapriyan et al., 2021).

- **Déséquilibre des gradients entre les composantes de la fonction de perte.** Les termes correspondant au résidu de l'EDP, aux conditions aux limites et à l'ajustement aux données peuvent présenter des gradients de grandeurs très différentes, ce qui conduit certaines contraintes à dominer l'optimisation tandis que d'autres sont de facto négligées. Ce déséquilibre, parfois qualifié de *gradient pathology* (*pathologie du gradient*) des PINNs, peut entraîner une convergence lente ou des solutions biaisées (Wang et al., 2021a, 2025a).

- **Biais spectral des réseaux neuronaux.** Il est bien connu que les réseaux neuronaux apprennent beaucoup plus rapidement les composantes basses fréquences d'une fonction cible que ses composantes hautes fréquences, un phénomène désigné sous le nom de biais spectral (Rahaman et al., 2019; Cao et al., 2020). Dans le contexte des PINNs, ce biais se manifeste par une incapacité à capturer des structures de solution de haute fréquence ou fortement localisées, même lorsque la capacité du réseau est suffisante. Cet effet a été abondamment documenté et analysé dans (Wang et al., 2021b; Deshpande et al., 2022).

Un large éventail de stratégies a été proposé pour répondre à ces défis et améliorer la robustesse ainsi que la précision des PINNs. Ces approches couvrent plusieurs directions complémentaires :

- **Apprentissage par curriculum.** Des stratégies d'entraînement progressif ou par étapes ont été proposées afin de faciliter l'optimisation en augmentant graduellement la difficulté de la tâche d'apprentissage, par exemple en raffinant progressivement les termes de l'EDP (Krishnapriyan et al., 2021).

- **Pondération adaptative des pertes.** Afin de corriger le déséquilibre entre les différentes composantes de la fonction de perte, des mécanismes

de repondération automatique ont été développés, soit entre les termes EDP, bord et données (Wang et al., 2021a), soit de manière adaptative au niveau des points de collocation (McClenny and Braga-Neto, 2022).

- **Architectures alternatives et enrichissement des représentations.** D'importants efforts ont été consacrés à l'amélioration de la capacité d'expression et du biais inductif des architectures neuronales utilisées dans les PINNs. Parmi les exemples notables figurent les réseaux hybrides ou enrichis en Fourier (Wang et al., 2024, 2023; Urbán et al., 2025), les architectures fondées sur les transformeurs (Zhao et al., 2024), ainsi que les plus récents *Kolmogorov–Arnold Networks* (KANs ; Liu et al., 2025; Wang et al., 2025b), visant tous à mieux capturer des structures de solution complexes ou multi-échelles.

- **Imposition forte des contraintes.** Plutôt que de recourir à des termes de pénalisation, plusieurs travaux proposent d'intégrer directement les contraintes physiques dans l'architecture ou la paramétrisation du réseau, garantissant ainsi la satisfaction exacte des conditions initiales et aux limites (Zhu et al., 2019; Sun et al., 2020; Liu and Wang, 2021; Wang et al., 2023).

- **Échantillonnage adaptatif et raffinement des points de collocation.** Au lieu de fixer les points de collocation *a priori* , des schémas adaptatifs concentrent dynamiquement les échantillons dans les régions où le résidu de l'EDP est le plus élevé, améliorant ainsi la convergence et la précision (Lu et al., 2021b; Lau et al., 2024).

- **Apprentissage respectant la causalité.** Pour les problèmes dépendant du temps, l'imposition d'un ordre causal dans l'entraînement, progressant le long de la dimension temporelle, s'est révélée bénéfique en termes de stabilité et de précision (Krishnapriyan et al., 2021; Wang et al., 2022b; Nguyen et al., 2022b, 2023).

- **Optimisation de second ordre et quasi-Newton.** Étant donné que les méthodes du premier ordre comme Adam (Kingma and Ba, 2014) peuvent rencontrer des difficultés face au mauvais conditionnement des fonctions de perte des PINNs, plusieurs solveurs de second ordre ou quasi-Newton ont été étudiés, permettant une convergence plus rapide et plus stable (Rathore et al., 2024; Urbán et al., 2025; Wang et al., 2025a).

Pour une synthèse étendue et à jour des avancées récentes en matière de méthodologies PINNs, nous renvoyons le lecteur à la synthèse de Luo et al. (2025).

Une autre ligne de recherche importante (Wang et al., 2021b; Deshpande et al., 2022) a revisité ces problématiques à travers le prisme de la théorie du *Neural Tangent Kernel* (NTK ; Jacot et al., 2018). En analysant le spectre du NTK sous l'hypothèse dite de *lazy training* (Chizat et al., 2019), ces travaux ont mis en évidence un fort mauvais conditionnement, conduisant à l'identification d'un biais spectral. Ce phénomène explique en partie la lenteur de convergence des PINNs ainsi que leur incapacité à capturer les composantes de haute fréquence, ce qui se traduit *in fine* par une précision limitée. Afin d'atténuer cet effet, plusieurs stratégies de préconditionnement ont été proposées pour remodeler le spectre du NTK, améliorant ainsi l'apprentissage des modes de haute fréquence et renforçant à la fois la vitesse de convergence et la stabilité numérique (De Ryck et al., 2024; Liu et al., 2024). Par ailleurs, la théorie du NTK a également inspiré de nouvelles stratégies d'échantillonnage des points de collocation (Lau et al., 2024), établissant des liens avec les méthodes classiques de Nyström (Sun et al., 2015).

Enfin, une ligne de recherche distincte a revisité les PINNs à travers le prisme de la *géométrie fonctionnelle* (Müller and Zeinhofer, 2023, 2024; Jnini et al., 2024), atteignant des niveaux de précision comparables à ceux des solveurs numériques traditionnels. Dans cette perspective, la dynamique d'entraînement des PINNs est interprétée comme une approximation de flots de gradient dans un espace fonctionnel de dimension infinie. Cette formulation géométrique établit un lien naturel entre les PINNs et les *méthodes de gradient naturel* (Amari, 1998; Ollivier, 2015b,a; Bai et al., 2022), offrant ainsi une compréhension de principe de leur comportement en optimisation.

Notre travail se situe précisément à l'intersection de ces deux approches complémentaires : la perspective de la *géométrie fonctionnelle* et le cadre du *Neural Tangent Kernel* (NTK). Plus précisément, cette thèse vise à combler le fossé entre la formulation géométrique différentielle riemannienne (Gallot et al., 1990) sous-jacente à la géométrie fonctionnelle, et la famille des *méthodes à noyaux* (Hofmann et al., 2008). Il est rapidement apparu que le cadre théorique des *Reproducing Kernel Hilbert Spaces* (RKHS ; Paulsen and Raghupathi, 2016) constitue l'environnement le plus naturel pour formaliser cette connexion, donnant lieu à plusieurs développements algorithmiques fructueux.

Nous présentons ainsi deux contributions principales dans cette thèse :

- **Dans le Chapter 4**, nous présentons notre première contribution, l'algorithme *ANaGRAM* (Schwencke and Furtlehner, 2025). Nous commençons par établir un cadre formel unifiant le Neural Tangent Kernel (NTK) et les méthodes de gradient naturel, et introduisons la notion de *gradient naturel empirique*, qui permet de réduire la complexité computationnelle du gradient naturel de $\mathcal{O}(P^3)$ à $\mathcal{O}(\min(P^2S, S^2P))$,

où $P = \#\{\text{paramètres}\}$ et $S = \#\{\text{échantillons du batch}\}$. Nous montrons ensuite comment cet algorithme s'étend naturellement aux PINNs, en produisant des résultats numériques dont la précision dépasse de plusieurs ordres de grandeur celle obtenue avec les optimiseurs standards. Enfin, nous mettons en évidence, dans le cadre théorique développé, un lien entre le gradient naturel appliqué aux PINNs et la fonction de Green de l'opérateur différentiel sous-jacent.

- **Dans le Chapter 5**, nous présentons notre seconde contribution, l'algorithme *AMStramGRAM* (Schwencke et al., 2025). Cette méthode s'appuie sur une analyse empirique détaillée du schéma de régularisation introduit dans ANaGRAM et en déduit une nouvelle stratégie de régularisation adaptative, permettant d'atteindre une précision de niveau machine sur des benchmarks simples et des performances compétitives sur une large classe de problèmes aux dérivées partielles. Nous fournissons également des interprétations à la fois statistiques et géométriques de ce mécanisme de régularisation, et montrons, à l'aide d'outils issus de la théorie spectrale, qu'une telle régularisation constitue une condition nécessaire à la solvabilité du problème d'inversion étudié dans le Chapter 3 sur les RKHS. Comme sous-produit, nous étendons les liens théoriques entre ce cadre et la théorie des fonctions de Green.

Bien que cette orientation n'ait pas été envisagée *a priori* , elle a progressivement conduit le présent travail vers des investigations plus approfondies en analyse fonctionnelle, qui occupent désormais une place centrale dans ce manuscrit. À ce titre, notre étude s'inscrit dans l'effort théorique croissant consacré ces dernières années à la compréhension de la *physics-informed machine learning*, en complément notamment des analyses statistiques de Doumèche (2025), qui établissent des garanties de convergence à l'aide de méthodes à noyaux, ainsi que des travaux de De Ryck (2024), centrés sur la convergence numérique et la stabilité.

Nous espérons que notre contribution, complémentaire à ces différentes lignes de recherche, contribuera à ouvrir la voie à de nouveaux développements, en particulier en construisant de nouveaux ponts entre des outils mathématiques et des techniques algorithmiques qui, jusqu'à présent, ont largement évolué en parallèle sans pleinement se rencontrer.

Contents

Remerciements **v**

Introduction détaillée **ix**

Table of Contents **xix**

I Preliminaries 1

1 Introduction **3**

2 Neural Networks and Physics-Informed Neural Networks **13**
2.1 A Differential-Geometric Viewpoint on Neural Networks . . . 14
2.1.1 Neural Networks . . . 14
2.1.2 Parametric models . . . 18
2.1.3 Notion of Loss . . . 21
2.1.4 Natural and classical gradient flow . . . 23
2.1.5 Natural gradient flow induced by a Hilbertian gradient flow . . . 29
2.1.6 PINNs theoretical loss . . . 33
2.2 Approximations to the theoretical framework . . . 35
2.2.1 Automatic differentiation . . . 35
2.2.2 Loss approximation . . . 40
2.2.3 (Natural) Neural Tangent Kernel (N)NTK . . . 44
2.2.4 Gradient flow approximation . . . 47

3 Reproducing Kernel Hilbert Spaces (RKHS) **55**
3.1 Motivation . . . 55
3.2 Fundamental setting . . . 56
3.2.1 Kernels, Integral Operators, and Green's Functions . . . 56
3.2.2 Reproducing property . . . 58
3.2.3 Fondamental examples . . . 62
3.3 Characterization of $\mathcal{H}$ by its reproducing kernel . . . 66
3.3.1 Uniqueness and Representations of the Reproducing Kernel . . . 66
3.3.2 Aronszajn–Moore construction . . . 71
3.3.3 Regularity of $\mathcal{H}(k)$ inherited from its kernel . . . 76
3.4 Effects of Norm Modifications on the Reproducing Kernel . . . 80
3.4.1 First Positive Results . . . 80
3.4.2 Norm induced by a bicontinuous operator D . . . 82
3.4.3 Norm induced by a non-bijective operator D . . . 86
3.4.4 Combined norm modifications between RKHSs . . . 89

3.5 Interpolation and Approximation in RKHSs . . . 91
3.5.1 Formal problem statement . . . 92
3.5.2 Interpolation: the case of a surjective ι_F . . . 95
3.5.3 Approximation: the case of a non-surjective ι_F . . . 98
3.6 Beyond RKHS: Schwartz's Generalization of Reproducing Kernels . . . 101
3.6.1 Schwartz's Framework . . . 101
3.6.2 Recovering Moore–Aronszajn construction from Schwartz's Framework . . . 102
3.6.3 Recovering Hilbert Rigging from Schwartz's Framework . . . 103

II Contributions 107

4 ANaGRAM: A Natural Gradient Relative to Adapted Model for efficient PINNs learning 109
4.1 Position of the problem . . . 109
4.1.1 Parametric models and Neural Networks . . . 110
4.1.2 Physics Informed Neural Networks (PINNs) . . . 110
4.1.3 Natural Gradient . . . 110
4.2 Empirical Natural Gradient and ANaGRAM algorithm . . . 111
4.2.1 Empirical Natural Gradient . . . 111
4.2.2 ANaGRAM approximation theorem . . . 112
4.2.3 ANaGRAM algorithm . . . 118
4.3 Adaptation of empirical Natural Gradient and ANaGRAM to PINNs . . . 119
4.3.1 PINNs as a least-squares regression problem . . . 119
4.3.2 Natural Neural Tangent Kernel and empirical Tangent Space of PINNs . . . 121
4.3.3 NNTK *vs* NTK in PINNs context . . . 122
4.4 PINNs Natural Gradient is a Green's Function . . . 131
4.4.1 Statement of the connection with Green's functions and interpretation . . . 132
4.4.2 Proof of Theorem 4.8 . . . 133
4.4.3 Illustration of estimated Green's function for PINNs . . . 135
4.5 Experiments . . . 138
4.5.1 2 D Laplace equation . . . 139
4.5.2 1+1 D Heat equation . . . 140
4.5.3 5 D Laplace equation . . . 141
4.5.4 1+1 D Allen-Cahn equation . . . 142
4.6 Conclusion and Perspectives . . . 143

5 AMStraMGRAM: Adaptive Multi-cutoff Strategy Modification for ANaGRAM 145
5.1 Position of the Problem . . . 145
5.1.1 Parametric Models and Neural Networks . . . 145
5.1.2 Physics-Informed Neural Networks (PINNs) . . . 146
5.1.3 Natural Gradient . . . 146
5.1.4 Empirical Natural Gradient . . . 147

5.2 Regularization . . . 147
5.2.1 Why Regularization is Necessary . . . 148
5.2.2 Ridge-regression . . . 150
5.2.3 Cutoff regression . . . 154
5.2.4 Connection to Green's Function . . . 157
5.3 Reconstruction Error of the Functional Gradient . . . 159
5.3.1 Definition . . . 160
5.3.2 Properties of the Reconstruction Error . . . 160
5.4 Insights on ANaGRAM's Training Dynamics . . . 165
5.4.1 Empirical Observations: Flattening . . . 165
5.4.2 Incomplete Flattening and Adaptive Strategies . . . 169
5.5 Algorithmic Design: Exploiting Flattening . . . 170
5.5.1 Adaptive Cutoff Strategy . . . 170
5.5.2 Outline of the AMStraMGRAM Algorithm . . . 171
5.6 Experimental Results . . . 173
5.7 Limitations . . . 174
5.8 Conclusion . . . 176

6 Conclusion and Perspectives **177**

A Complements to ANaGRAM (Chapter 4) **199**
A.1 Complementary material for the Experiments . . . 199
A.1.1 Description of the experimental setting . . . 199
A.1.2 Figures relative to computation time . . . 199
A.1.3 Statistical tables of results . . . 201
A.1.4 Additional experiment : Burgers equation . . . 207
A.1.5 A practical example : Derivation of generalized Green's function for Laplacian operator, based on PINN's natural gradient formulation on a Fourier's basis . . . 209
A.2 Mathematical equivalence of ANaGRAM and Energy Natural Gradient implementation (Müller and Zeinhofer, 2023) for linear operators . . . 211

B Complements to AMStraMGRAM (Chapter 5) **215**
B.1 Our vocabulary . . . 215
B.2 Practical Implementation Considerations . . . 217
B.2.1 The Dual Cutoff Strategy: Addressing Empirical Challenges . . . 217
B.2.2 Three-Phase Training Dynamics . . . 217
B.2.3 Complete Practical Algorithm . . . 218
B.2.4 Empirical Justification for Design Choices . . . 221
B.3 Deep dive on selected experiments . . . 221
B.3.1 One Dimensional Burgers Equation . . . 221
B.3.2 Heat Equation . . . 222
B.3.3 Laplace Equations (L2D and L5D) . . . 222
B.3.4 Non Linear Poisson Equation . . . 223

B.3.5 Allen-Cahn Equation 224

C Supplementary Mathematical Results **227**
C.1 Notations 227
C.2 Adjoints 228
C.2.1 Adjoint of a Bounded Operator 228
C.2.2 Adjoints for densely defined closed operators 231
C.3 Elements of General Spectral Theory 234

Part I

Preliminaries

1 - Introduction

Partial Differential Equations (PDEs) lie at the heart of modern science and engineering. They describe a wide variety of physical phenomena, ranging from fluid dynamics (Navier–Stokes equations) to quantum mechanics (Schrödinger equation), electromagnetism (Maxwell equations), and elasticity or wave propagation (Helmholtz and wave equations). Mathematically, a PDE models the balance of fundamental quantities such as mass, momentum, or energy over space and time, making them the universal language of physical modeling.

Nevertheless, PDEs are generally *not solvable by algebraic means*. This impossibility can be rigorously justified by the same kind of arguments that arise in Galois theory (Singer, 2009), which classifies equations based on their solvability by radicals: just as most high-degree polynomial equations cannot be solved in closed form, most nonlinear or even mildly coupled PDEs lack analytic solutions.

Consequently, the study of PDEs often proceeds through *indirect methods*. Rather than attempting to express explicit solutions, one investigates their qualitative properties: existence, uniqueness, regularity, and stability. Functional analysis (Brezis, 2011), spectral theory (Edmunds and Evans, 2018), and variational methods (Gelfand et al., 2000) have thus become central tools for characterizing solution spaces and understanding the behavior of PDEs, even in the absence of explicit formulas.

In practice, however, the only way to obtain concrete solutions for general PDEs is through *numerical approximation* (Iserles, 2009). Classical approaches include the finite-difference (Strikwerda, 2004), finite-element (Zienkiewicz and Taylor, 2013), and finite-volume methods (Moukalled et al., 2016), each relying on the discretization of the continuous domain by a mesh and the replacement of derivatives with algebraic operators that can be handled computationally using linear algebra or iterative schemes. A slightly different class of approaches is provided by spectral methods (Shen et al., 2011), which approximate the unknown function not through local discretization but by projecting it onto a finite-dimensional subspace spanned by global basis functions, such as trigonometric or orthogonal polynomials (Legendre, Chebyshev, etc.). In this setting, differentiation and integration can be represented exactly within the chosen basis.

Despite their remarkable success and theoretical maturity, all these methods share common limitations:

- **Curse of dimensionality:** the computational cost grows exponentially with the dimension d of the domain.

- **Complex geometries:** mesh-based methods require careful grid construction, which becomes difficult for irregular domains or domains evolving in time.

- **Nonlinearities and stiffness:** strongly nonlinear PDEs or PDEs with multiscale features (e.g., turbulent flows) often require extremely fine discretizations to remain stable and accurate, leading to prohibitive computational cost.

- **Lack of flexibility:** classical solvers are tied to specific discretizations, making it difficult to reuse information across different PDEs, boundary conditions, or parameter regimes.

- **Limited data integration:** traditional solvers assume fully specified physical models and cannot naturally incorporate partial or noisy data. While specialized techniques such as data assimilation (Asch et al., 2016) exist, they are typically problem-specific and update the solver's output rather than integrating data and physics within a unified framework.

In many modern applications, however, such as climate modeling (Stocker, 2011), medical imaging (Paulsen and Moeslund, 2020), or fluid mechanics (Wendt, 2009), the governing equations are only partially known or must be inferred from observations. Moreover, the underlying geometries are often complex, multiscale, or even only partially specified, while the models involve numerous variables with intricate nonlinear interactions, resulting in extremely high-dimensional problems.

These challenges have motivated the development of more flexible, mesh-free, and data-adaptive approaches, inspired by recent advances in machine learning (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016; Bach, 2024). This has given rise to a rapidly expanding field known as *Scientific Machine Learning* (SciML).

This emerging discipline seeks to combine the flexibility and expressive power of modern machine learning with the rigor and structure of physical modeling, in order to bridge data and physics in a principled way. Broadly speaking, these approaches can be grouped into three main categories:

- **Data-driven modeling.** This line of research focuses on discovering governing equations or dynamical laws directly from data, without assuming prior knowledge of the underlying physics. Typical examples include the sparse identification of nonlinear dynamics (SINDy; Brunton et al., 2016; Fiorini et al., 2025) and neural ordinary differential equations (Neural ODEs; Chen et al., 2018), where the objective is to learn the evolution operator that best explains the observed trajectories.

- **Operator learning.** A second, rapidly developing branch of SciML focuses on learning mappings between entire function spaces rather than pointwise values (Boullé and Townsend, 2024; Kovachki et al., 2024), with the goal of approximating the solution operator associated with a family of PDEs. A particularly influential approach is that of Neural Operators (Kovachki et al., 2021), among which the DeepONet (Lu et al., 2021a) and the Fourier Neural Operator (FNO; Li et al., 2020) have gained notable popularity. Once trained, these models enable rapid inference for new inputs, boundary conditions, or parameters, effectively serving as learned surrogate operators.

- **Physics-informed machine-learning (PIML).** Finally, this family of methods incorporates known physical laws, typically differential equations or conservation principles, into the learning process as soft or hard constraints. The most representative approach in this category is that of *Physics-Informed Neural Networks* (PINNs; Raissi et al., 2019b), which approximate PDE solutions by minimizing both the residual of the governing equations and the mismatch on boundary and initial conditions. Numerous extensions and refinements have been proposed in recent years, forming an active and rapidly growing field of research (Cuomo et al., 2022; Meng et al., 2025).

These three paradigms—data-driven discovery, operator learning, and physics-informed regularization—constitute the foundations of modern SciML. Together, they offer a spectrum of strategies for integrating data, physics, and computation into a unified mathematical and algorithmic framework.

Note that these three perspectives are by no means disjoint. In recent years, a growing body of research has sought to hybridize these paradigms, combining their complementary strengths. For example, several methods merge operator learning with physics-informed modeling, such as the Physics-Informed Neural Operator (PINO; Li et al., 2021), which incorporates PDE residuals directly into the operator-learning loss, or the Physics-Enhanced Neural Operator (PENO; Chen et al., 2024), which integrates physical knowledge into the neural operator architecture.

Hybrid strategies have also emerged at the interface between physics-informed learning and data-driven modeling. Extensions of PINNs have been developed to handle partially known systems or noisy observational data by coupling them with data-assimilation or system-identification techniques (Raissi et al., 2020; Kissas et al., 2020; Nguyen et al., 2022a). In a probabilistic vein, Yang et al. (2021) proposed a unified Bayesian framework that combines experimental data with physics-based priors, thereby enabling uncertainty quantification for both model and data errors.

Finally, the recent work of Doumèche et al. (2024b,a, 2025) introduces the Physics-Informed Kernel Learning (PIKL) framework, which leverages kernel-based solvers to provide both a practical algorithmic formulation and a rigorous statistical analysis of the benefits of incorporating physical information compared to purely data-driven methods.

Similarly, SciML is not isolated from classical numerical analysis. One of the most notable bridges is through Model Order Reduction (MOR) methods (Schilders et al., 2008), which share deep conceptual links with operator learning. Recent works have sought to enrich these classical approaches with deep learning techniques, for instance, Menier (2024) extend traditional MOR frameworks using neural architectures, while Somacal (2024) establish connections between Reduced Basis Methods (Cohen and DeVore, 2015; Benner et al., 2017) and SciML, including extensions toward physics-informed formulations.

Extensive research has also been devoted to sampling strategies that are relevant to both Scientific Machine Learning (SciML) and classical numerical analysis (Bon et al., 2025; Haberstich et al., 2022; Gruhlke et al., 2024; Nouy and Michel, 2024; Dolbeault and Cohen, 2022a; Cohen and Migliorati, 2017; Dolbeault and Cohen, 2022b).

In a different vein, Nastorg (2024) propose a data-driven approach using Graph Neural Networks (GNNs; see Wu et al., 2020) for domain decomposition (Chan and Mathew, 1994) applied to elliptic equations.

Other hybrid strategies combine neural solvers with traditional numerical schemes. For example, the Neural Semi-Lagrangian Method (Franck et al., 2025) introduces a time-discretized scheme in which neural networks approximate the spatial component through regression of the characteristic flow, computed via Runge–Kutta integration (see, *e.g.* Iserles, 2009). Another mixed strategy consists in first approximating a coarse solution using a PINN, then refining the residual error using a finite-element method (FEM), as in Barucq et al. (2025).

In this dissertation, we focus in particular on the framework of *Physics-Informed Neural Networks* (PINNs). Building upon the pioneering work of Dissanayake and Phan-Thien (1994); Lagaris et al. (1998), this approach was later revisited by Berg and Nyström (2018) following the rise of deep learning (LeCun et al., 2015), and subsequently popularized by Raissi et al. (2019b), who introduced the decisive use of automatic differentiation (Baydin et al., 2018) to compute the required differential operators.

The key idea is to represent the unknown solution $u : \Omega \to \mathbb{R}$ by a neural network $u_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^P$, and to train this network so that it satisfies both the governing PDE and the associated boundary conditions. Concretely, one defines a loss functional measuring the mismatch between the PDE residual and the boundary conditions, evaluated at randomly sampled

collocation points:

$$\ell(\boldsymbol{\theta}) = \frac{1}{2S_D}\sum_{i=1}^{S_D}\left|D[u_{\boldsymbol{\theta}}](x_i^D) - f(x_i^D)\right|^2 + \frac{1}{2S_B}\sum_{j=1}^{S_B}\left|B[u_{\boldsymbol{\theta}}](x_j^B) - g(x_j^B)\right|^2, \quad (1.1)$$

where D and B respectively denote the differential and boundary operators. The network parameters $\boldsymbol{\theta}$ are then optimized to minimize $\ell(\boldsymbol{\theta})$, yielding an approximation of the solution that satisfies the PDE over the chosen collocation points.

Over the years, numerous variants of PINNs have been proposed, differing mainly in how the underlying physical constraints are enforced. Among the earliest alternatives is the Deep Ritz Method (DRM; E and Yu, 2017), where the loss is defined from the variational (energy) formulation of the PDE rather than its residual. Other approaches rely on Galerkin or Petrov–Galerkin formulations, where the residual is multiplied by a test function before integration. Depending on the choice of test function, this leads to different formulations, which are summarized in Table 1.1 below.

Table 1.1: Variants of PINNs based on different choices of test functions (adapted from Kharazmi et al., 2021).

Test function type	**Method**
Dirac delta	PINNs (Raissi et al., 2019b)
PDE residual	DGM (Sirignano and Spiliopoulos, 2018)
Global polynomials	VPINNs (Kharazmi et al., 2019)
Piecewise polynomials	VarNet (Khodayi-Mehr and Zavlanos, 2020)
Non-overlapping polynomials	*hp*-VPINNs (Kharazmi et al., 2021)

Several other extensions have been proposed within this general framework, including *Conservative PINNs* (CPINNs; Jagtap et al., 2020) and *Extended PINNs* (XPINNs; Jagtap and Karniadakis, 2021), which aim to subdivide the global domain into smaller subdomains and train independent PINNs on each, enforcing consistency only through interface conditions between subdomains. Another line of research, known as greedy training (Siegel et al., 2023; Wang and Lai, 2024), consists in training PINNs sequentially, where each new network is optimized to minimize the residual left by the previous one. The final approximation is then obtained as the sum of all individually trained networks, progressively refining the overall solution.

All these techniques have since been applied across a wide range of fields and have achieved remarkable successes in areas as diverse as fluid mechanics

(Raissi et al., 2019c,a; Sun et al., 2020; Raissi et al., 2020; Jin et al., 2021; de Wolff et al., 2021), bioengineering (Sahli Costabal et al., 2020; Kissas et al., 2020), free-boundary problems (Wang and Perdikaris, 2021), shape optimization (Bélieres–Frendo et al., 2024) or reinforcement learning (Mukherjee and Liu, 2023; Shilova et al., 2024). As a paradigmatic illustration of their potential, PINNs have even been successfully applied to problems in up to one million dimensions (Shi et al., 2024), a work that received the *Best Paper Award* at NeurIPS 2024[1].

All that being said, these achievements must be put into perspective, as many challenges still remain. A major limitation is that standard PINNs often struggle to reach the precision levels required in scientific computing—particularly in regimes where mesh-based solvers excel, whether through finite element methods (FEMs; Grossmann et al., 2024) or finite-difference-based schemes (Karnakov et al., 2022).

Numerous difficulties and failure modes have been identified that help explain these shortcomings:

- **Highly nonconvex optimization landscapes.** The training loss of PINNs typically involves the residuals of nonlinear differential operators, leading to highly nonconvex and ill-conditioned optimization problems. As a result, optimization often stagnates in poor local minima or plateaus, especially in high-dimensional or stiff systems (Rathore et al., 2024; Krishnapriyan et al., 2021).

- **Gradient imbalance between loss components.** The PDE residual, boundary conditions, and data-fitting terms in the composite loss can have gradients of vastly different magnitudes, causing certain constraints to dominate the optimization while others are effectively ignored. This imbalance, sometimes referred to as the gradient pathology of PINNs, can lead to slow convergence or biased solutions (Wang et al., 2021a, 2025a).

- **Spectral bias of neural networks.** Neural networks are known to learn low-frequency components of a target function much faster than high-frequency ones—a phenomenon known as spectral bias (Rahaman et al., 2019; Cao et al., 2020). In the context of PINNs, this bias manifests as a failure to capture high-frequency or localized solution features, even when the network capacity is sufficient. This effect has been extensively documented and analyzed in (Wang et al., 2021b; Deshpande et al., 2022).

[1] https://blog.neurips.cc/2024/12/10/announcing-the-neurips-2024-best-paper-awards/

A wide range of strategies have been proposed to address these challenges and improve the robustness and accuracy of PINNs. These approaches span several complementary directions:

- **Curriculum learning.** Progressive or staged training strategies have been proposed to ease optimization by gradually increasing the difficulty of the learning task, for example by refining the PDE terms over time (Krishnapriyan et al., 2021).

- **Adaptive loss weighting.** To mitigate the imbalance between different loss components, automatic reweighting mechanisms have been developed, either across the PDE, boundary, and data terms (Wang et al., 2021a), or adaptively across collocation samples (McClenny and Braga-Neto, 2022).

- **Alternative architectures and feature embeddings.** Significant efforts have been made to improve the expressive power and inductive bias of neural architectures used in PINNs. Examples include hybrid or Fourier-enhanced networks (Wang et al., 2024, 2023; Urbán et al., 2025), transformer-based architectures (Zhao et al., 2024), and the more recent Kolmogorov–Arnold Networks (KANs; Liu et al., 2025; Wang et al., 2025b), all aiming to better capture complex or multiscale solution structures.

- **Hard enforcement of constraints.** Instead of relying on soft penalty terms, several works have proposed to embed the physical constraints directly into the network architecture or parametrization, ensuring that boundary and initial conditions are exactly satisfied (Zhu et al., 2019; Sun et al., 2020; Liu and Wang, 2021; Wang et al., 2023).

- **Adaptive sampling and collocation refinement.** Rather than fixing the collocation points a priori, adaptive schemes dynamically concentrate samples in regions where the PDE residual is largest, improving convergence and accuracy (Lu et al., 2021b; Lau et al., 2024).

- **Causality-aware training.** For time-dependent problems, enforcing a causal training order—progressing along the temporal dimension—has been shown to improve stability and accuracy (Krishnapriyan et al., 2021; Wang et al., 2022b; Nguyen et al., 2022b, 2023).

- **Second-order and quasi-Newton optimization.** Since first-order methods like Adam (Kingma and Ba, 2014) can struggle with ill-conditioned PINN losses, several second-order or quasi-Newton solvers have been investigated, yielding faster and more stable convergence (Rathore et al., 2024; Urbán et al., 2025; Wang et al., 2025a).

For an extensive and up-to-date overview of recent advances in PINN methodologies, we refer the reader to Luo et al. (2025).

Another important line of research (Wang et al., 2021b; Deshpande et al., 2022) has revisited these issues through the lens of the *Neural Tangent Kernel* (NTK; Jacot et al., 2018) theory. By analyzing the NTK spectrum under the lazy training assumption (Chizat et al., 2019), these studies have revealed a severe ill-conditioning that leads to the identification of a spectral bias. This phenomenon partially explains the slow convergence of PINNs and their inability to capture high-frequency components, ultimately resulting in poor accuracy. To mitigate this effect, several preconditioning strategies have been proposed to reshape the NTK spectrum, thereby improving the learning of high-frequency modes and enhancing both convergence speed and numerical stability (De Ryck et al., 2024; Liu et al., 2024). Furthermore, NTK theory has also inspired new collocation sampling strategies (Lau et al., 2024), establishing connections with classical Nyström methods (Sun et al., 2015).

Finally, a distinct line of research has revisited PINNs through the lens of *functional geometry* (Müller and Zeinhofer, 2023, 2024; Jnini et al., 2024), achieving accuracy levels comparable to those of traditional numerical solvers. In this view, the training dynamics of PINNs are interpreted as approximate gradient flows in an infinite-dimensional function space. This geometric formulation naturally connects PINNs with *natural gradient methods* (Amari, 1998; Ollivier, 2015b,a; Bai et al., 2022), offering a principled understanding of their optimization behavior.

Our work stands at the crossroads of those two complementary approaches: the *functional geometry* viewpoint and the *Neural Tangent Kernel* (NTK) framework. More precisely, this dissertation aims to bridge the gap between the Riemannian differential geometric formulation (Gallot et al., 1990) underlying the functional-geometric perspective, and the family of *kernel methods* (Hofmann et al., 2008). It soon became apparent that the theoretical framework of *Reproducing Kernel Hilbert Spaces* (RKHS; Paulsen and Raghupathi, 2016) provides the most natural setting in which to formalize this connection, leading to several fruitful algorithmic developments.

We thus make two main contributions in this dissertation:

- **In Chapter 4**, we present our first contribution, the *ANaGRAM* algorithm (Schwencke and Furtlehner, 2025). We begin by establishing a formal framework that unifies the Neural Tangent Kernel (NTK) and natural gradient methods, and introduce the notion of an *empirical natural gradient*, which reduces the computational complexity of the natural gradient from $\mathcal{O}(P^3)$ to $\mathcal{O}(\min(P^2S, S^2P))$, where $P = \#\{\text{parameters}\}$

and $S = \#\{\text{batch samples}\}$. We then show how this algorithm naturally extends to PINNs, achieving numerical results with accuracies several orders of magnitude higher than those obtained with standard optimizers. Finally, we demonstrate, within the theoretical framework we developed, a connection between the natural gradient in PINNs and the Green's function of the underlying differential operator.

- **In Chapter 5**, we present our second contribution, the *AMStramGRAM* algorithm (Schwencke et al., 2025). This method builds upon a detailed empirical analysis of the regularization scheme introduced in ANaGRAM and derives a new adaptive regularization strategy that achieves machine-level precision on simple benchmarks and competitive results on a broad class of PDE problems. We also provide both statistical and geometric interpretations of this regularization mechanism and show, using tools from spectral theory, that such regularization is a necessary condition for ensuring the solvability of the inversion problem discussed in Chapter 3 on RKHSs. As a byproduct, we extend the theoretical links between this framework and Green's function theory.

Although this direction was not initially planned *a priori*, it progressively steered the present work toward deeper functional-analytic investigations, which now occupy a central place in this manuscript. In this sense, our study contributes to the growing theoretical effort devoted in recent years to understanding *physics-informed machine learning*, complementing in particular the statistical analyses of Doumèche (2025), who established convergence guarantees using kernel-based methods, and the work of De Ryck (2024), focused on numerical convergence and stability.

We hope that our contribution, complementary to these lines of research, will help open the way to further developments, above all by building new bridges between mathematical tools and algorithmic techniques that, until now, have evolved in parallel without fully intersecting.

The remainder of this dissertation is structured as follows:

- **In Chapter 2**, we introduce the fundamental algorithmic and mathematical concepts underlying neural networks and physics-informed neural networks (PINNs), from the standpoint of functional geometry. We pay particular attention to the formalism of the *Neural Tangent Kernel* (NTK) and to optimization methods, with an emphasis on the *natural gradient*.

  The concepts introduced in this chapter will be instrumental for the developments presented in Chapters 4 and 5.

- **In Chapter 3**, we present the theory of *Reproducing Kernel Hilbert Spaces* (RKHSs) from the perspective of functional analysis, highlighting the fundamental link between the inversion of linear operators and the modification of the topological properties of a space. This mechanism should be regarded as the theoretical backbone of the entire dissertation. As an extension, we also discuss the generalization proposed by Schwartz (1964), which connects RKHSs to the theory of distributions (Schwartz, 1966).

  The concepts introduced in this chapter will be instrumental for the developments presented in Chapters 4 and 5.

- **In Chapter 4**, we present our first contribution, corresponding to the *ANaGRAM* algorithm (Schwencke and Furtlehner, 2025).

- **In Chapter 5**, we describe our second contribution, the *AMStramGRAM* algorithm (Schwencke et al., 2025).

2 - Neural Networks and Physics-Informed Neural Networks

In this chapter, we introduce the two main numerical components of this dissertation.

Neural Networks (NNs). Also referred to as *Artificial Neural Networks* (ANNs), they were historically introduced under the name "Perceptron" by Rosenblatt (1958) as mathematical models designed to emulate the behavior of biological neurons (hence their name). In a biological neural network, multiple dendrites receive signals from other neurons, which are aggregated by the soma before being transmitted through the axon and relayed to other neurons via synapses.

Drawing inspiration from this mechanism, artificial neural networks combine "input signals" linearly (through an affine transformation) and process them using an *activation function* (such as the sign function) before passing the resulting signal to the next neuron. Each neuron thus performs an elementary computation. The key idea is that the weights applied to the inputs, together with the affine coefficient called the *bias*, are considered *learnable parameters*—that is, parameters adjusted during training so that the neuron learns to perform a useful primitive operation. When stacked together, as in a *Multi-Layer Perceptron* (MLP), these simple units can collectively represent extremely complex mappings.

Another cornerstone of modern neural networks is the training method known as *Stochastic Gradient Descent* (SGD), introduced by Amari (1967). This method provides an efficient way to minimize loss functions over large datasets and will serve as a foundation for most of our optimization discussions later in this chapter.

A third fundamental idea that led to the success of modern neural networks is the combination of *backpropagation* and *automatic differentiation*. Initially implemented by Linnainmaa (1976) and later formalized by Werbos (1982), these techniques were popularized by LeCun et al. (1998), establishing the computational basis for large-scale neural network training. Since then, neural networks have flourished across a broad spectrum of applications, including computer vision (Voulodimos et al., 2018) and natural language processing (Goldberg, 2022). In this work, we will focus on the classical *Multi-Layer Perceptron* (MLP), which is sufficient for our theoretical and numerical developments.

Physics-Informed Neural Networks (PINNs). First introduced by Dissanayake and Phan-Thien (1994) and Lagaris et al. (1998), *Physics-Informed Neural Networks* (PINNs) extend neural networks to approximate solutions of Partial Differential Equations (PDEs). The core idea is to compute the derivatives of a neural network's outputs with respect to its inputs—using automatic differentiation—and combine them into the residual of a PDE, which the network is then trained to minimize.

This concept experienced a major revival with the work of Raissi et al. (2019b), who leveraged modern automatic differentiation frameworks (Baydin et al., 2018) to compute derivatives efficiently and demonstrated the effectiveness of PINNs on a wide range of forward and inverse problems. As reviewed in Introduction, numerous extensions and applications have since been developed, establishing PINNs as one of the most active research directions in scientific machine learning.

Throughout this chapter, and indeed the entire manuscript, we will adopt a functional–geometric viewpoint, consistent with the perspective that underlies all our subsequent developments.

This choice of viewpoint, however, will not prevent us from addressing the algorithmic foundations required for the practical implementation of these methods. In particular, we will review the main optimization techniques used to train neural networks, highlighting their respective similarities and differences, as well as their underlying assumptions and objectives.

Let us therefore begin by introducing neural networks from the standpoint of differential geometry.

2.1 . A Differential-Geometric Viewpoint on Neural Networks

In this section, we introduce Neural Networks and Physics-Informed Neural Networks (PINNs) using the language of functional analysis and differential geometry.

2.1.1 . Neural Networks

We begin by providing the formal definition of a *Multi-Layer Perceptron* (MLP).

Definition 2.1 (Multiple Layers Perceptron (MLP))**.** Let us fix:

- $d \in \mathbb{N}_1$, called the *depth* of the MLP. When $d \geqslant 2$, we say that the MLP is *deep*.
- $(l_i)_{0 \leqslant i \leqslant d} \in \mathbb{N}_1^{d+1}$, called the *widths*.

- $(W_i)_{1 \leqslant i \leqslant d} \in \prod_{i=1}^{d} \mathcal{M}_{l_i, l_{i-1}}(\mathbb{R})$, called the *weights*.
- $(b_i)_{1 \leqslant i \leqslant d} \in \prod_{i=1}^{d} \mathbb{R}^{l_i}$, called the *biases*.
- $(\alpha_i)_{1 \leqslant i \leqslant d} \in \prod_{i=1}^{d} \mathcal{F}(\mathbb{R}^{l_i})$, called the *activation functions*.

The associated MLP is the mapping

$$\mathrm{MLP} : \left\{ \begin{array}{ccc} \mathbb{R}^{l_0} & \longrightarrow & \mathbb{R}^{l_d} \\ x & \longmapsto & \big(\circ_{i=1}^{d} \alpha_i(W_i \cdot + b_i) \big)(x) \, . \end{array} \right. \tag{2.1}$$

***Remark* 2.2.** In practice, the domain of a Multiple Layer Perceptron (MLP) is seldom taken to be the whole space $\mathbb{R}^{l_0}$. Instead, one usually considers a subdomain $\Omega \subset \mathbb{R}^{l_0}$ of dimension l_0, corresponding, for instance, to the computational domain of interest.

In such cases, we naturally define the restriction

$$\mathrm{MLP}_{|\Omega} : \left\{ \begin{array}{ccc} \Omega & \longrightarrow & \mathbb{R}^{l_d} \\ x & \longmapsto & \big(\circ_{i=1}^{d} \alpha_i(W_i \cdot + b_i) \big)(x) \, . \end{array} \right. \tag{2.2}$$

However, this restriction is generally clear from context and need not be stated explicitly.

***Remark* 2.3.** In the context of neural networks, activation functions are typically of the form "a scalar function applied component-wise". Formally, for a given $k \in \mathbb{N}_1$ and an activation $\alpha \in \mathcal{F}(\mathbb{R}^k)$, there exists $g : \mathbb{R} \to \mathbb{R}$ such that $\alpha : x \in \mathbb{R}^k \mapsto \big(g(x_i)\big)_{1 \leqslant i \leqslant k} \in \mathbb{R}^k$. It is therefore common to abuse notation and refer to α simply by the scalar function g.

Popular choices for activations, in the sense of the scalar function g of Remark 2.3, are listed in Table 2.1. Identifying the most effective activation functions remains an active research area (Dubey et al., 2022).

***Remark* 2.4.** Since differentiability is crucial for neural network training (see Section 2.1.3), we will primarily consider activations that are differentiable almost everywhere. This includes all Lipschitz-continuous functions, as guaranteed by Rademacher's theorem (Evans 2018, Section 3.1).

We now introduce the notion of *architecture*, which lies at the heart of both the practical use of neural networks and our theoretical framework. An architecture specifies the structural design of a neural network: it determines its size (*i.e.* its depth and the width of each layer), the symmetries imposed on the weights (for instance, some coefficients may be constrained to be equal, as in convolutional networks; see *e.g.* Albawi et al. 2017), and the choice of activation functions. Only a subset of the weights and biases are left free—these will be *learned* during training, in the sense that their values will be adjusted by an optimization algorithm (see Section 2.1.3), while others may remain fixed (for instance, to enforce sparsity or symmetry constraints). We now formalize this notion in the language of functional analysis.

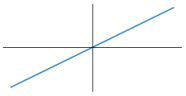
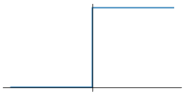
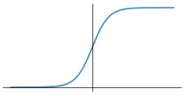
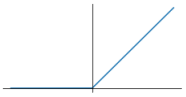
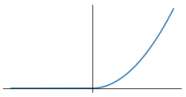
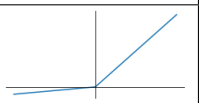
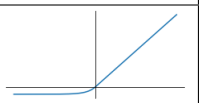
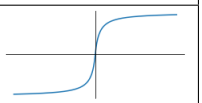
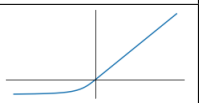
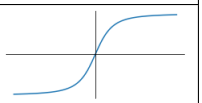
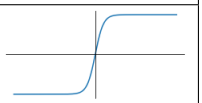
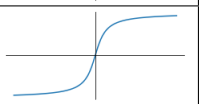
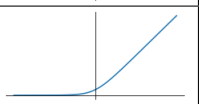
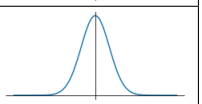
Name	Definition ($x \in \mathbb{R}$)	Plot
linear	x	
Heaviside / step	$\mathbb{1}_{(0,\infty)}(x)$	
logistic (sigmoid)	$\frac{1}{1+e^{-x}}$	
ReLU	$\max\{0, x\}$	
Power ReLU	$\max\{0, x\}^k$ for $k \in \mathbb{N}_1$	
Parametric ReLU	$\max\{ax, x\}$ for $a \geqslant 0$, $a \neq 1$	
ELU	$x\mathbb{1}_{[0,\infty)}(x) + (e^x - 1)\mathbb{1}_{(-\infty,0)}(x)$	
softsign	$\frac{x}{1+\lvert x\rvert}$	
inverse square root linear unit	$x\mathbb{1}_{[0,\infty)}(x) + \frac{x}{\sqrt{1+ax^2}}\mathbb{1}_{(-\infty,0)}(x)$	
inverse square root unit	$\frac{x}{\sqrt{1+ax^2}}$ for $a > 0$	
tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	
arctan	$\arctan(x)$	
SoftPlus	$\ln(1 + e^x)$	
Gaussian	$e^{-x^2/2}$	

Table 2.1: Commonly used activation functions (adapted from Berner et al. 2021).

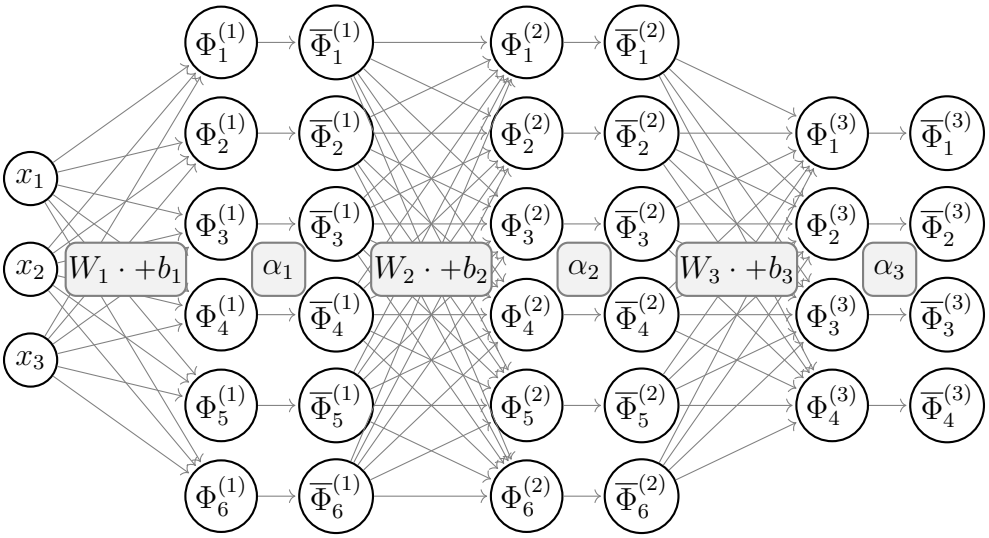


Figure 2.1: Illustration of a MLP : $\mathbb{R}^3 \to \mathbb{R}^4$ of depth 3 with widths $(3, 6, 6, 4)$ and activations $(\alpha_i)_{i=1}^3$ (adapted from Berner et al. 2021).
Reading: the input $x \in \mathbb{R}^3$ is first multiplied by matrix $W_1 \in \mathcal{M}_{6,3}(\mathbb{R})$ before adding $b_1 \in \mathbb{R}^6$. Then α_1 is applied to the result. The same process is repeated layer by layer until the final output in $\mathbb{R}^4$ is obtained after α_3.
Formally, for all $1 \leqslant i \leqslant 3$ and $1 \leqslant j \leqslant l_i$, $\Phi_j^{(i)} : x \mapsto \big(W_j(\circ_{k=1}^{j-1}\alpha_k(W_k \cdot + b_k))(x) + b_j\big)_i$ and $\overline{\Phi}_j^{(i)} : x \mapsto \big((\circ_{k=1}^{j}\alpha_k(W_k \cdot + b_k))(x)\big)_i$.

Definition 2.5 (Architecture)**.** Let us fix:

- $d \in \mathbb{N}_1$, called the **depth** of the network;
- $(l_i)_{0 \leqslant i \leqslant d} \in \mathbb{N}_1^{d+1}$, called the **widths** of each layer;
- $(\alpha_i)_{1 \leqslant i \leqslant d} \in \prod_{i=1}^d \mathcal{D}^1(\mathbb{R}^{l_i})$, called the **activation functions**;
- $P \in \mathbb{N}_1$, with $1 \leqslant P \leqslant \sum_{i=1}^d l_i(l_{i-1} + 1)$, called the **number of parameters**;
- a differentiable map $\psi \in \mathcal{C}^1\Big(\mathbb{R}^P \to \prod_{i=1}^d \mathcal{M}_{l_i,l_{i-1}}(\mathbb{R}) \times \prod_{i=1}^d \mathbb{R}^{l_i}\Big)$, called the **parameterization map**.

Then, an *architecture* for an MLP is a mapping

$$u : \left\{ \begin{array}{ccc} \mathbb{R}^P & \longrightarrow & \mathcal{F}(\mathbb{R}^{l_0} \to \mathbb{R}^{l_d}) \\ \theta & \longmapsto & \Big(\circ_{k=1}^d \alpha_k\big(\psi_k(\theta) \cdot + \psi_{d+k}(\theta)\big)\Big), \end{array} \right. \tag{2.3}$$

where $\psi_k(\theta)$ and $\psi_{d+k}(\theta)$ respectively denote the weight matrices and bias vectors corresponding to each layer.

***Remark* 2.6.** The parameterization map ψ encodes both the *symmetries* among weights and biases (such as those arising in convolutional or tied-weight architectures) and the distinction between *learnable* and *fixed* parameters. In this dissertation, we will mainly view ψ as a convenient reparameterization of a real vector of dimension

$$P = \sum_{i=1}^{d} l_i(l_{i-1} + 1), \tag{2.4}$$

into the corresponding collection of weight matrices $(W_i)_{i=1}^d \in \prod_{i=1}^d \mathcal{M}_{l_i,l_{i-1}}(\mathbb{R})$ and bias vectors $(b_i)_{i=1}^d \in \prod_{i=1}^d \mathbb{R}^{l_i}$. This interpretation allows us to rigorously describe the geometry of the parameter space while maintaining the flexibility to incorporate structural constraints or prior knowledge directly into the architecture.

***Remark* 2.7.** A wide variety of neural network architectures have been developed beyond the classical Multiple Layer Perceptron (MLP), each tailored to specific data structures or invariances. Among the most prominent are *Convolutional Neural Networks* (CNNs; LeCun et al., 1998; Goodfellow et al., 2016), particularly suited for spatially structured data such as images; *Recurrent Neural Networks* (RNNs; Elman, 1990) and their variants such as LSTMs and GRUs (Hochreiter and Schmidhuber, 1997; Cho et al., 2014), designed for sequential data; and more recently, *Transformers* (Vaswani et al., 2017), which rely on attention mechanisms and have revolutionized natural language processing and beyond.

Nevertheless, many of these architectures can be viewed as specific instances or constrained variants of MLPs, where certain weights are shared, frozen, or structured according to the problem at hand (for instance, convolutional filters in CNNs or attention weights in Transformers). In the context of this dissertation, we will therefore restrict our analysis to standard MLPs, as this class already captures the essential mathematical and geometrical features relevant to our study. Accordingly, throughout this work, we shall use the terms *neural network* and *MLP* interchangeably.

A natural question that arises is to understand the structure of the image of a neural network architecture.

2.1.2 . Parametric models

Let us fix an architecture u. A purely algebraic point of view is insufficiently rich to discuss essential notions such as continuity or differentiability, concepts that are crucial for defining optimization algorithms. We therefore need to endow both $\mathbb{R}^P$ and $\operatorname{Im} u$ with appropriate norm topologies. Since all norms on $\mathbb{R}^P$ are equivalent, we may simply consider the one induced by its canonical inner product. For $\operatorname{Im} u$, however, several choices are possible. This topic is studied in detail in Berezansky et al. 1996b, Chapter 12.

Since this correspondence between algebraic and topological structures encompasses many cases beyond neural networks and will form the foundation of the connections we establish with other numerical solution techniques, we include them in the following abstract and general definition.

Definition 2.8 (Parametric model)**.** Given a domain Ω of $\mathbb{R}^n$, and a Hilbert space $\mathcal{H}$ made of functions $\Omega \to \mathbb{K}^m$, $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, a parametric model is a differentiable functional:

$$u : \mathbb{R}^P \longrightarrow \mathcal{H}. \tag{2.5}$$

Notation 2.9. Since, for a given $\theta \in \mathbb{R}^P$, the evaluation $u(\theta)$ is itself a function, we shall use the notation u_θ instead of $u(\theta)$ to avoid any possible confusion.

Since a parametric model is differentiable by definition, we can define its differential:

Definition 2.10 (Differential of a parametric model)**.** Let $u : \mathbb{R}^P \to \mathcal{H}$ be a parametric model and $\theta \in \mathbb{R}^P$. Then the differential of the parametric model u in the parameter θ is:

$$\mathrm{d}u_\theta : \left\{ \begin{array}{ccc} \mathbb{R}^P & \to & \mathcal{H} \\ h & \mapsto & \sum_{p=1}^P h_p \frac{\partial u}{\partial \theta_p} = \lim\limits_{s \to 0} \frac{u_{\theta + sh} - u_\theta}{s} \end{array} \right. . \tag{2.6}$$

Notation 2.11. To simplify notations, we will write for all $1 \leqslant p \leqslant P$ and for all $\theta \in \mathbb{R}^P$, $\partial_p u_\theta$, instead of $\frac{\partial u}{\partial \theta_p}$.

Let us now assume that, in a neighborhood of some point $\boldsymbol{\theta_0} \in \mathbb{R}^P$, the map u is continuously differentiable and that its differential $\mathrm{d}u_{\boldsymbol{\theta_0}}$ at $\boldsymbol{\theta_0}$ is invertible, *i.e.* injective (continuity of the inverse automatically holds since the space is finite-dimensional). Then, by the Inverse Function Theorem (local version), there exists a neighborhood $V_{\boldsymbol{\theta_0}}$ of $\boldsymbol{\theta_0}$ in $\mathbb{R}^P$ such that the restriction $u_{|V_{\boldsymbol{\theta_0}}}$ is a diffeomorphism onto its image. In particular, $\mathcal{M}_{\boldsymbol{\theta_0}} := \operatorname{Im} u_{|V_{\boldsymbol{\theta_0}}}$ inherits the structure of a $\mathcal{C}^1$ manifold. Hence, we can define the *tangent space* $T_\theta \mathcal{M}_{\boldsymbol{\theta_0}}$ at any point $\theta \in V_{\boldsymbol{\theta_0}}$ as

$$T_\theta \mathcal{M}_{\boldsymbol{\theta_0}} := \operatorname{Im} \mathrm{d}u_\theta. \tag{2.7}$$

***Example* 2.12** (Partial Fourier's series)**.** Let us fix a dimension $d \in \mathbb{N}_1$. We then define the N-partial Fourier's Serie in $[0,1]^d$ as:

$$S_N : \left\{ \begin{array}{ccc} \mathbb{R}^{[\![-N,N]\!]^d} & \to & \mathrm{L}^2([0,1]^d \to \mathbb{C}) \\ (\alpha_{k_1,\ldots,k_d}) & \mapsto & \left(\underset{[0,1]^d}{x} \mapsto \sum\limits_{k_1=-N}^{N} \cdots \sum\limits_{k_d=-N}^{N} \alpha_{k_1,\ldots,k_d} e^{2i\pi \left(\sum_{l=1}^d k_l x_l\right)} \right) \end{array} \right. . \tag{2.8}$$

We see that for all $k \in [\![-N, N]\!]^d$ and $\theta \in \mathbb{R}^{[\![-N,N]\!]^d}$,

$$\partial_k S_{N|\theta} = \left(x \in [0,1]^d \mapsto e^{2i\pi\left(\sum_{l=1}^d k_l x_l\right)}\right). \tag{2.9}$$

As a consequence: for all $\theta \in \mathbb{R}^{[\![-N,N]\!]^d}$

$$\mathrm{d}S_{N|\theta} = S_N,$$

an thus: for all $\theta \in \mathbb{R}^{[\![-N,N]\!]^d}$

$$\mathcal{M} = T_\theta \mathcal{M} = \mathrm{Span}\left(x \in [0,1]^d \mapsto e^{2i\pi\left(\sum_{l=1}^d k_l x_l\right)} \ : \ k \in [\![-N, N]\!]^d\right) \tag{2.10}$$

This precisely means that S_N is a linear parametric model.

However, while it is always possible to restrict attention to parametric models u that are differentiable everywhere (for instance, by choosing $\mathcal{C}^1$ activation functions in an MLP), the differentials $\mathrm{d}u_\theta$ are not necessarily injective. For example, in an MLP with $d \geqslant 2$, if $\boldsymbol{\theta_0}$ corresponds to a configuration in which all the weights of the last layer vanish, then all partial derivatives with respect to the weights of the first layer are zero, and consequently $\mathrm{d}u_{\boldsymbol{\theta_0}}$ fails to be injective.

Strictly speaking, this means that the image $\mathrm{Im}\, u$ does not form a smooth manifold in the classical sense, but rather a *stratified space* (also called a *pseudomanifold*; see Pflaum, 2001). Nevertheless, the precise study of such objects lies beyond the scope of this dissertation. Since all our theoretical and algorithmic developments rely only on the existence of a differential and the identification of the tangent space at $\theta \in \mathbb{R}^P$ with $\mathrm{Im}\, \mathrm{d}u_\theta$, we will, by a mild abuse of language, assume that $\mathcal{M} := \mathrm{Im}\, u$ possesses a $\mathcal{C}^1$-manifold structure.

We now turn to a fundamental density result that justifies the expressive power of neural networks.

Theorem 2.13 (Universal Approximation Theorem)**.** *Let $f \in \mathcal{C}^0_c(\mathbb{R}^n \to \mathbb{R}^m)$, $d \geqslant 2$, and let $\|\cdot\|_{\mathbb{R}^m}$ be a norm on $\mathbb{R}^m$. Assume $(g_i)_{1 \leqslant i \leqslant d-1} \in \mathcal{F}(\mathbb{R} \to \mathbb{R})^{d-1}$ are locally bounded, piecewise continuous, and non-polynomial activation functions.*

Then, for any $\varepsilon > 0$, there exist integers $(l_i)_{0 \leqslant i \leqslant d} \in \mathbb{N}_1^{d+1}$ with $l_0 = n$ and $l_d = m$, together with weights $(W_i)_{1 \leqslant i \leqslant d} \in \prod_{i=1}^d \mathcal{M}_{l_i, l_{i-1}}(\mathbb{R})$ and biases $(b_i)_{1 \leqslant i \leqslant d} \in \prod_{i=1}^d \mathbb{R}^{l_i}$ such that the associated neural network

$$\mathrm{MLP} : x \in \mathbb{R}^n \longmapsto W_d\left(\circ_{i=1}^{d-1} g_i(W_i \cdot + b_i)\right)(x) + b_d \in \mathbb{R}^m \tag{2.11}$$

satisfies

$$\sup_{x \in \mathbb{R}^n} \|\mathrm{MLP}(x) - f(x)\|_{\mathbb{R}^m} < \varepsilon. \tag{2.12}$$

Proof. See Leshno et al. 1993. □

Theorem 2.13 is a cornerstone of neural network theory. It asserts that the set $\mathcal{NN}_{n,m}$ of all neural networks with input dimension n and output dimension m, allowing arbitrary architectures and weights, is dense in $\mathcal{C}_c^0(\mathbb{R}^n \to \mathbb{R}^m)$, the space of continuous, compactly supported functions. This result is particularly powerful because $\mathcal{C}_c^0(\mathbb{R}^n \to \mathbb{R}^m)$ is dense in most commonly used functional spaces (such as L_p spaces for $1 \leqslant p < \infty$, and Sobolev spaces), implying that MLPs can approximate virtually any function of practical interest.

More refined results have since been obtained, providing explicit error bounds in terms of network size for specific activation functions; see, for example, De Ryck et al. (2021) and Daubechies et al. (2022). This remains an active area of research.

A complementary line of investigation explores the combinatorial and computational aspects of approximation, namely, how the number of weights or neurons scales with the desired precision. These studies highlight the expressive benefits of depth, showing that deep architectures can approximate certain functions exponentially more efficiently than shallow ones; see Telgarsky (2016) for a detailed discussion.

While these theoretical results establish the expressive universality of neural networks, they do not provide a constructive way to find the required weights and biases. In practice, this challenge is addressed through the process of *training*, which provides heuristic but effective optimization algorithms to approximate the ideal parameters. Training, therefore, constitutes the practical cornerstone of neural network implementation and will be the subject of the rest of this section.

Note, however, that other lines of research explore architectures that evolve jointly with the adaptation of their weights. See in particular the recent work of Verbockhaven (2025).

2.1.3 . Notion of Loss

The *training* of a neural network refers, in general, to any procedure that, given an architecture in the sense of Definition 2.5, seeks to determine the optimal weights and biases that best fit a prescribed objective, for instance, approximating a target function. The most common way to formalize such an objective is through a *loss function*, which encodes the structure of the problem and expresses the objective as a minimization task. Designing an appropriate loss for a given objective constitutes a central part of modern deep learning methodology (Wang et al., 2022a). Physics-Informed Neural Networks (PINNs) can be viewed as a prominent example of this principle.

Following our general framework, we now introduce a rigorous definition of loss functions within the language of functional analysis and differential

geometry, extending beyond neural networks to any parametric model in the sense of Definition 2.8.

Definition 2.14 (Loss)**.** Let $\big(\mathcal{M}, \langle \cdot\,,\,\cdot \rangle_{\mathcal{M},\cdot}\big)$ be a Riemannian manifold. A **loss** is a differentiable function $\mathcal{L} : \mathcal{M} \to \mathbb{R}$ that is bounded from below. Without loss of generality, we may assume that $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$.

***Remark* 2.15.** In particular, a loss can be defined on any Hilbert space $\big(\mathcal{H}, \|\cdot\|_{\mathcal{H}}\big)$.

***Remark* 2.16.** Additional properties may be required depending on the problem, such as convexity, which guarantees the existence of a unique minimum.

We now present a fundamental example of a theoretical loss function, which serves as a cornerstone for both theory and practice in neural network training.

***Example* 2.17** (Theoretical mean square error)**.** Let $\mathcal{M} = \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$, and fix $f \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$. We define the loss

$$\mathcal{L}_2 : \left\{ \begin{array}{ccc} \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu) & \longrightarrow & \mathbb{R}_+ \\ v & \longmapsto & \dfrac{1}{2} \|v - f\|^2_{\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)} \end{array} \right. . \tag{2.13}$$

This *theoretical mean square error* (MSE) is differentiable on $\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$, with differential at v given by

$$\mathrm{d}\mathcal{L}_{2_{|v}} : \left\{ \begin{array}{ccc} \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu) & \longrightarrow & \mathbb{R} \\ h & \longmapsto & \langle v - f\,,\, h \rangle_{\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)} \end{array} \right. . \tag{2.14}$$

The MSE is one of the most widely used losses in both theory and applications of neural networks.

Let now $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$ be a loss. Since each tangent space $T_p\mathcal{M}$ (for $p \in \mathcal{M}$) is a Hilbert space, the Riesz–Fréchet representation theorem ensures the existence of a unique element that represents the differential of $\mathcal{L}$ at each point.

Definition-Proposition 2.18 (Gradient of a loss)**.** *Let $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$ be a differentiable loss. Then, for every $v \in \mathcal{M}$, there exists a unique $\nabla\mathcal{L}_v \in T_v\mathcal{M}$ such that for all $h \in T_v\mathcal{M}$,*

$$d\mathcal{L}_v(h) = \langle \nabla\mathcal{L}_v\,,\, h \rangle_{\mathcal{M},v}\,. \tag{2.15}$$

The element $\nabla\mathcal{L}_v$ is called the gradient *of $\mathcal{L}$ at v.*

Proof. Since $\mathcal{L}$ is continuously differentiable, for every $v \in \mathcal{M}$, the differential $\mathrm{d}\mathcal{L}_v : T_v\mathcal{M} \to \mathbb{R}$ is a continuous linear form. The result follows directly from the Riesz–Fréchet representation theorem. □

***Remark* 2.19.** Fix $v \in \mathcal{M}$ and a chart $\varphi : \mathbb{R}^P \to \mathcal{M}$ such that $v \in \operatorname{Im}\varphi$. Without loss of generality, assume $\varphi(0) = v$. By the definition of the tangent space $T_v\mathcal{M}$, for every $h \in T_v\mathcal{M}$ there exists $\delta \in \mathbb{R}^P$ such that $\mathrm{d}\varphi_0(\delta) = h$. We then obtain the identity

$$\mathrm{d}\mathcal{L}_v(h) = \mathrm{d}\mathcal{L}_{\varphi(0)}\big(\mathrm{d}\varphi_0(\delta)\big) = \big\langle \nabla\mathcal{L}_{\varphi(0)} \,,\, \mathrm{d}\varphi_0(\delta)\big\rangle_{\mathcal{M},\,\varphi(0)} \,. \tag{2.16}$$

***Remark* 2.20.** In particular, if $\mathcal{M} = \mathcal{H}$ is a Hilbert space, then for every $v \in \mathcal{H}$, the tangent space satisfies $T_v\mathcal{H} = \mathcal{H}$. The identity map $\mathcal{I}_\mathcal{H}$ thus provides a global chart, and the gradient of $\mathcal{L}$ exists directly in the ambient space. Hence, Equation (2.16) rewrites as

$$\mathrm{d}\mathcal{L}_v(h) = \mathrm{d}\mathcal{L}_{\mathcal{I}_\mathcal{H}(v)}\big(\mathrm{d}(\mathcal{I}_\mathcal{H})_v(h)\big) = \big\langle \nabla\mathcal{L}_{\mathcal{I}_\mathcal{H}(v)} \,,\, \mathrm{d}(\mathcal{I}_\mathcal{H})_v(h)\big\rangle_\mathcal{H} = \langle \nabla\mathcal{L}_v \,,\, h\rangle_\mathcal{H} \,. \tag{2.17}$$

***Example* 2.21.** For the loss $\mathcal{L}_2$ defined in Example 2.17, Equation (2.14) immediately shows that, for every $v \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$, the gradient is given by $\nabla\mathcal{L}_{2_v} = v - f$.

We now have all the necessary ingredients to define the *theoretical gradient flow* associated with a loss function on a manifold.

We will successively adopt two complementary perspectives: first, the one from the functional (or ambient) space of the manifold, which leads to the notion of the *natural gradient flow*; and second, the one from the parameter space of the manifold, which leads to the notion of the *classical gradient flow*.

2.1.4 . Natural and classical gradient flow

Let us begin with the natural gradient flow, which is the most natural starting point in our context.

Definition 2.22 (Functional natural gradient flow)**.** Let $\boldsymbol{v_0} \in \mathcal{M}$ and $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$ be a loss. The theoretical gradient flow associated with $\mathcal{L}$, starting at $\boldsymbol{v_0}$, is the following ordinary differential equation (ODE) on $v : \mathbb{R}_+ \to \mathcal{M}$:

$$\begin{cases} v(0) & = \boldsymbol{v_0}, \\ \dfrac{d}{dt}v(t) & = -\nabla\mathcal{L}_{v(t)} \end{cases} . \tag{2.18}$$

***Remark* 2.23.** If, for some $t_0 \geqslant 0$, one has $\nabla\mathcal{L}_{v(t_0)} = 0$, then $\forall t \geqslant t_0$, $v(t) = v(t_0)$. In other words, once the flow reaches a critical point of $\mathcal{L}$, it remains there. This justifies the term "training": the objective of gradient descent is precisely to reach a local minimum of $\mathcal{L}$ in as little time as possible.

***Remark* 2.24.** The gradient $\nabla\mathcal{L}_{v(t)}$ depends crucially on the Riemannian structure of $\mathcal{M}$. Different choices of metric lead to different gradient dynamics. This distinction lies at the heart of the difference between *classical* and *natural* gradients.

***Remark* 2.25.** Note that Definition 2.22 also applies in particular to Hilbert spaces, since a Hilbert space is itself a flat Riemannian manifold.

The most important case for us is when $\mathcal{M}$ is the image of a parametric model u. In this setting, the definition still holds formally, but a key simplification arises: since u provides a global chart, the gradient flow can be expressed directly in the parameter space. Note, however, that in the general case of an abstract manifold, this is no longer true: a parametric gradient flow can only be defined locally on the image of a chart. One must then transition between charts, change coordinate systems, and perform appropriate gluing operations to recover the global flow.

We first consider the case where the differentials of u are invertible for all $\theta \in \mathbb{R}^P$.

Definition-Proposition 2.26 (Invertible parametric natural gradient flow)**.** *Let $u : \mathbb{R}^P \to \mathcal{M}$ be a parametric model whose differentials du_θ are invertible for all θ, and let $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$ be a loss. Then, applying the theoretical gradient flow of Definition 2.22 to $\upsilon := u \circ \theta$ yields the equivalent parametric ODE for $\theta : \mathbb{R}_+ \to \mathbb{R}^P$:*

$$\begin{cases} \theta(0) & = u^{-1}(\boldsymbol{\upsilon_0}), \\ \dfrac{d}{dt}\theta(t) & = du^{-1}_{\theta(t)}\big(-\nabla\mathcal{L}_{u(\theta(t))}\big) \end{cases}. \tag{2.19}$$

Proof. Clearly, $u(\theta(0)) = u\big(u^{-1}(\boldsymbol{\upsilon_0})\big) = \boldsymbol{\upsilon_0}$. The result then follows directly from the chain rule:

$$\begin{aligned} \frac{d}{dt}(u \circ \theta)(t) &= \mathrm{d}u_{\theta(t)}\left(\frac{d}{dt}\theta(t)\right) \overset{(2.19)}{=} \mathrm{d}u_{\theta(t)}\Big(\mathrm{d}u^{-1}_{\theta(t)}\big(-\nabla\mathcal{L}_{u(\theta(t))}\big)\Big) \\ &= -\nabla\mathcal{L}_{u(\theta(t))}. \end{aligned} \tag{2.20}$$

□

When $\mathrm{d}u_\theta$ is not invertible for all $\theta \in \mathbb{R}^P$, note that since u parameterizes $\operatorname{Im} u$, we still have $T_{u(\theta)}\mathcal{M} = \operatorname{Im}\mathrm{d}u_\theta$, *i.e.* $\mathrm{d}u_\theta$ is merely non-injective. In this case, we can still define a gradient flow on any supplementary subspace to $\operatorname{Ker}\mathrm{d}u_\theta$. A natural and practical choice is to take $(\operatorname{Ker}\mathrm{d}u_\theta)^\perp$, where $\perp$ denotes the orthogonal complement with respect to the Euclidean inner product.

Definition-Proposition 2.27 (Parametric natural gradient flow)**.** *Let $u : \mathbb{R}^P \to \mathcal{M}$ be a parametric model and $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$ a loss. Then, applying the theoretical gradient flow of Definition 2.22 to $\upsilon := u \circ \theta$ yields the equivalent parametric ODE for $\theta : \mathbb{R}_+ \to \mathbb{R}^P$:*

$$\begin{cases} \theta(0) & = u^{-1}(\boldsymbol{\upsilon_0}), \\ \dfrac{d}{dt}\theta(t) & = du^{\dagger}_{\theta(t)}\big(-\nabla\mathcal{L}_{u(\theta(t))}\big) \end{cases}. \tag{2.21}$$

Proof. The only difference with the proof of Definition-Proposition 2.26 lies in justifying the use of the pseudo-inverse of $\mathrm{d}u_{\theta(t)}$.

Fix $t \in \mathbb{R}_+$ and consider the SVD of $\mathrm{d}u_{\theta(t)}$ given for all $\delta \in \mathbb{R}^P$ by,

$$\mathrm{d}u_{\theta(t)} = \sum_{p=1}^{P} V_{\theta(t),p} \Delta_{\theta(t),p} \left\langle U_{\theta(t),p} \,,\, \delta \right\rangle_{\mathbb{R}^P} = \sum_{p=1}^{r_\theta} V_{\theta(t),p} \Delta_{\theta(t),p} \left\langle U_{\theta(t),p} \,,\, \delta \right\rangle_{\mathbb{R}^P} , \tag{2.22}$$

where $r_\theta \leqslant P$ is the rank of $\mathrm{d}u_{\theta(t)}$. In particular, $\big(V_{\theta(t),p}\big)_{1 \leqslant p \leqslant r_\theta}$ forms a basis of $\operatorname{Im} u$. Then, for all $h \in T_{u(\theta)} \operatorname{Im} u$,

$$\mathrm{d}u^{\dagger}_{\theta(t)}(h) = \sum_{p=1}^{r_\theta} U_{\theta(t),p} \Delta^{-1}_{\theta(t),p} \left\langle V_{\theta(t),p} \,,\, h \right\rangle_{T_{u(\theta)} \operatorname{Im} u} . \tag{2.23}$$

Therefore,

$$\begin{aligned} \mathrm{d}u_{\theta(t)} \Big(\mathrm{d}u^{\dagger}_{\theta(t)}(-\nabla \mathcal{L}_{u(\theta(t))}) \Big) &= \sum_{p=1}^{r_\theta} V_{\theta(t),p} \left\langle V_{\theta(t),p} \,,\, -\nabla \mathcal{L}_{u(\theta(t))} \right\rangle_{T_{u(\theta)} \operatorname{Im} u} \\ &= -\nabla \mathcal{L}_{u(\theta(t))} . \end{aligned} \tag{2.24}$$

The conclusion then follows by the same reasoning as in the proof of Definition-Proposition 2.26. □

We now move on to the *classical theoretical gradient flow*. Let us consider a parametric model $u : \mathbb{R}^P \to \mathcal{M}$ and a loss $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$. We begin with the following definition.

Definition-Proposition 2.28 (Parametric loss)**.** *The function*

$$\ell := \mathcal{L} \circ u : \mathbb{R}^P \to \mathbb{R}_+ \tag{2.25}$$

is a loss in the sense of Definition 2.14, called the parametric loss *associated with the loss $\mathcal{L}$ and the parametric model u.*

Proof. To verify that this definition fits within the framework of Definition 2.14, simply note that $\mathbb{R}^P$, equipped with its canonical Euclidean inner product, is a flat Riemannian manifold. In particular, for all $\theta \in \mathbb{R}^P$, we have $T_\theta \mathbb{R}^P = \mathbb{R}^P$, so the composition $\mathcal{L} \circ u$ indeed defines a valid differentiable loss on $\mathbb{R}^P$. □

Notation 2.29 (Functional loss)**.** To avoid ambiguity in this context, we shall refer to $\mathcal{L}$ as the *functional loss*.

For completeness, let us recall the standard expression of the gradient of ℓ with respect to the canonical Euclidean inner product.

Proposition 2.30 (Euclidean gradient formula)**.** *Let $\ell : \mathbb{R}^P \to \mathbb{R}_+$ be a parametric loss as defined in Definition-Proposition 2.28. Then, for all $\theta \in \mathbb{R}^P$, the Euclidean gradient $\nabla \ell_\theta$ is given by*

$$\nabla \ell_\theta = \sum_{p=1}^{P} \partial_p \ell_\theta \, \boldsymbol{e}_p, \tag{2.26}$$

where $(\boldsymbol{e}_p)_{1 \leqslant p \leqslant P}$ denotes the canonical basis of $\mathbb{R}^P$, and $\partial_p \ell_\theta$ is the partial derivative of ℓ with respect to $\boldsymbol{e}_p$, that is,

$$\partial_p \ell_\theta = d\ell_\theta(\boldsymbol{e}_p). \tag{2.27}$$

Proof. For any $\delta \in \mathbb{R}^P$, we have

$$\begin{aligned} \left\langle \sum_{p=1}^{P} \partial_p \ell_\theta \, \boldsymbol{e}_p \, , \, \delta \right\rangle_{\mathbb{R}^P} &= \sum_{p=1}^{P} \partial_p \ell_\theta \, \delta_p = \sum_{p=1}^{P} \mathrm{d}\ell_\theta(\boldsymbol{e}_p) \, \delta_p = \mathrm{d}\ell_\theta \left(\sum_{p=1}^{P} \delta_p \, \boldsymbol{e}_p \right) \\ &= \mathrm{d}\ell_\theta(\delta), \end{aligned} \tag{2.28}$$

where the penultimate equality follows directly from the linearity of the differential $\mathrm{d}\ell_\theta$. □

We may now directly define a gradient flow with respect to the parametric loss ℓ, as formalized in the following definition.

Definition 2.31 (Parametric classical gradient flow)**.** Let $\boldsymbol{\theta_0} \in \mathbb{R}^P$, and let $\ell : \mathbb{R}^P \to \mathbb{R}_+$ be a parametric loss. The classical gradient flow associated with ℓ, initialized at $\boldsymbol{\theta_0}$, is defined by the following ordinary differential equation (ODE) on $\theta : \mathbb{R}_+ \to \mathbb{R}^P$:

$$\begin{cases} \theta(0) & = \boldsymbol{\theta_0} \\ \dfrac{d}{dt}\, \theta(t) & = -\nabla \ell_{\theta(t)} \end{cases}. \tag{2.29}$$

More generally, we may endow $\mathbb{R}^P$ with a non-Euclidean Riemannian structure by allowing its inner product to depend on the point $\theta \in \mathbb{R}^P$. In this setting, we still have $T_\theta \mathbb{R}^P = \mathbb{R}^P$ for all θ, but the inner product now varies smoothly with θ, defining the metric

$$\langle h_1 \, , \, h_2 \rangle_{\rho(\theta)} := h_1^\top \rho(\theta) \, h_2, \tag{2.30}$$

where $\rho : \mathbb{R}^P \to \mathcal{M}_P^+(\mathbb{R})$ is a Riemannian metric mapping taking values in the cone of symmetric positive definite matrices on $\mathbb{R}^P$. This construction therefore defines a smoothly varying family of inner products $\langle \cdot \, , \, \cdot \rangle_{\rho(\theta)}$ parametrized by θ.

This observation naturally leads to a generalization of Definition 2.31.

Definition 2.32 (Parametric Riemannian gradient flow)**.** Let $\rho : \mathbb{R}^P \to \mathcal{M}_P^+(\mathbb{R})$ be a Riemannian metric, and let $\ell : \mathbb{R}^P \to \mathbb{R}_+$ be a parametric loss. The *parametric gradient flow* of ℓ, initialized at $\boldsymbol{\theta_0}$ and defined with respect to the metric ρ, is given by the following ordinary differential equation (ODE) on $\theta : \mathbb{R}_+ \to \mathbb{R}^P$:

$$\begin{cases} \theta(0) & = \boldsymbol{\theta_0}, \\ \dfrac{d}{dt}\theta(t) & = -\nabla \ell_{\rho(\theta(t))}, \end{cases} . \tag{2.31}$$

Here, $\nabla \ell_{\rho(\theta(t))}$ denotes the gradient of ℓ with respect to the inner product $\langle \cdot\,,\,\cdot \rangle_{\rho(\theta(t))}$; that is, the unique vector $\nabla \ell_{\rho(\theta(t))} \in \mathbb{R}^P$ satisfying, for all $\delta \in \mathbb{R}^P$,

$$\mathrm{d}\ell_{\theta(t)}(\delta) = \left\langle \nabla \ell_{\rho(\theta(t))}\,,\,\delta \right\rangle_{\rho(\theta(t))} . \tag{2.32}$$

Note in particular that the parametric gradient $\nabla \ell_{\rho(\theta(t))}$ can be expressed in terms of the Euclidean parametric gradient

Proposition 2.33 (Relation between Riemannian and Euclidean gradients)**.** *Let $\ell : \mathbb{R}^P \to \mathbb{R}_+$ be a parametric loss, and let $\rho : \mathbb{R}^P \to \mathcal{M}_P^+(\mathbb{R})$ be a Riemannian metric. Then, for all $\theta \in \mathbb{R}^P$, the gradient of ℓ with respect to the metric $\rho(\theta)$ can be expressed in terms of its Euclidean gradient as*

$$\nabla \ell_{\rho(\theta)} = \rho(\theta)^{-1} \nabla \ell_\theta . \tag{2.33}$$

Proof. By definition of the Euclidean gradient, for all $h \in \mathbb{R}^P$, we have

$$\mathrm{d}\ell_\theta(h) = \langle \nabla \ell_\theta\,,\,h \rangle_{\mathbb{R}^P} = (\nabla \ell_\theta)^\top h. \tag{2.34}$$

We now express the same differential in terms of the Riemannian metric $\rho(\theta)$:

$$\begin{aligned} \langle \nabla \ell_\theta\,,\,h \rangle_{\mathbb{R}^P} &= (\nabla \ell_\theta)^\top h = (\nabla \ell_\theta)^\top \rho(\theta)^{-1} \rho(\theta) h \\ &= \left(\rho(\theta)^{-1} \nabla \ell_\theta\right)^\top \rho(\theta) h = \left\langle \rho(\theta)^{-1} \nabla \ell_\theta\,,\,h \right\rangle_{\rho(\theta)} . \end{aligned} \tag{2.35}$$

Comparing with the definition of the Riemannian gradient, namely

$$\mathrm{d}\ell_\theta(h) = \left\langle \nabla \ell_{\rho(\theta)}\,,\,h \right\rangle_{\rho(\theta)} , \tag{2.36}$$

we obtain that $\nabla \ell_{\rho(\theta)} = \rho(\theta)^{-1} \nabla \ell_\theta$, which establishes Equation (2.33). □

The next key question is to determine under which conditions Definition 2.32 and Definition-Proposition 2.27 are equivalent.

We provide a first answer to this question with the following proposition.

Proposition 2.34. *Let $u : \mathbb{R}^P \to \mathcal{M}$ be a parametric model and $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$ a functional loss. Consider the associated parametric loss $\ell := \mathcal{L} \circ u$. If, for all $\theta \in \mathbb{R}^P$, the differential $\mathrm{d}u_\theta$ is an isometry, then, provided that $u_{\boldsymbol{\theta_0}} = \boldsymbol{\upsilon_0}$, Definition-Proposition 2.27 and Definition 2.32 are equivalent, i.e. they yield the same gradient flow on $\theta : \mathbb{R}_+ \to \mathbb{R}^P$.*

Proof. Since $\mathrm{d}u_\theta$ is an isometry for all $\theta \in \mathbb{R}^P$, we have, whenever $\mathrm{d}u_\theta$ is invertible,

$$\mathrm{d}u_\theta^{-1} = \mathrm{d}u_\theta^*, \tag{2.37}$$

and more generally,

$$\mathrm{d}u_\theta^\dagger = \mathrm{d}u_\theta^*. \tag{2.38}$$

Therefore, for all $t \in \mathbb{R}_+$ and $h \in \mathbb{R}^P$, we obtain

$$\begin{aligned}
\left\langle \mathrm{d}u_{\theta(t)}^\dagger\big(-\nabla\mathcal{L}_{u(\theta(t))}\big)\ ,\ h\right\rangle_{\rho(\theta(t))} &= \left\langle \mathrm{d}u_{\theta(t)}^*\big(-\nabla\mathcal{L}_{u(\theta(t))}\big)\ ,\ h\right\rangle_{\rho(\theta(t))} \\
&= \left\langle -\nabla\mathcal{L}_{u(\theta(t))}\ ,\ \mathrm{d}u_{\theta(t)}(h)\right\rangle_{\mathcal{M},u(\theta(t))} \\
&\overset{\text{gradient definition}}{=} -\mathrm{d}\mathcal{L}_{u(\theta(t))}\left(\mathrm{d}u_{\theta(t)}(h)\right) \\
&\overset{\text{chain rule}}{=} -\mathrm{d}\left(\mathcal{L}\circ u\right)_{\theta(t)}(h) \\
&= -\mathrm{d}\ell_{\theta(t)}(h) \\
&\overset{\text{gradient definition}}{=} \left\langle -\nabla\ell_{\rho(\theta(t))}\ ,\ h\right\rangle_{\rho(\theta(t))}.
\end{aligned} \tag{2.39}$$

This implies that

$$\Big(\mathrm{d}u_{\theta(t)}^\dagger\big(-\nabla\mathcal{L}_{u(\theta(t))}\big) + \nabla\ell_{\rho(\theta(t))}\Big) \in \big(\mathbb{R}^P\big)^{\perp_{\rho(\theta(t))}} = \{0\}, \tag{2.40}$$

i.e.

$$\mathrm{d}u_{\theta(t)}^\dagger\big(-\nabla\mathcal{L}_{u(\theta(t))}\big) = -\nabla\ell_{\rho(\theta(t))}. \tag{2.41}$$

Hence, Equation (2.21) and Equation (2.31) describe the same dynamics, which concludes the proof. □

The next question is to determine under which conditions $\mathrm{d}u_\theta$ is an isometry. The key observation is that this can be reformulated as the problem of defining suitable Riemannian structures such that $\mathrm{d}u_{\theta(t)}$ becomes an isometry. Two natural constructions arise from this perspective:

- **Pullback (parameter-space) metric.** One may define a Riemannian structure ρ on $\mathbb{R}^P$ and transport it to $T_\theta\mathcal{M}$ through the differential $\mathrm{d}u_\theta$. More precisely, for all $\theta \in \mathbb{R}^P$ and $\delta_1, \delta_2 \in \mathbb{R}^P$, we define

  $$\langle \mathrm{d}u_\theta(\delta_1)\ ,\ \mathrm{d}u_\theta(\delta_2)\rangle_{T_\theta\mathcal{M}} := \langle \delta_1\ ,\ \delta_2\rangle_{\rho(\theta)}. \tag{2.42}$$

  This construction is well defined only if $\mathrm{d}u_\theta$ is injective. Otherwise, one can still define a valid Riemannian metric by restricting δ_1 and δ_2 to $\operatorname{Ker}\mathrm{d}u_\theta^{\perp_{\rho(\theta)}}$. In particular, if $\mathbb{R}^P$ is endowed with its canonical Euclidean structure and, for simplicity, $\mathrm{d}u_\theta$ is injective, this yields the metric defined for all $1 \leqslant p, q \leqslant P$ by

  $$\langle \partial_p u_\theta\ ,\ \partial_q u_\theta\rangle_{\mathrm{d}u_\theta} = \delta_{p,q}. \tag{2.43}$$

  This corresponds precisely to the setting of the classical gradient flow defined in Definition 2.31.

- **Ambient (function-space) metric.** Alternatively, one may define a Riemannian structure $\big(\langle \cdot\,,\,\cdot\rangle_{T_\theta\mathcal{M}}\big)_{\theta\in\mathbb{R}^P}$ directly on $\big(T_\theta\mathcal{M}\big)_{\theta\in\mathbb{R}^P}$ and transport it back to $\mathbb{R}^P$ by setting, for all $h_1, h_2 \in \mathbb{R}^P$,

$$\langle h_1\,,\,h_2\rangle_\theta := \langle \mathrm{d}u_\theta(h_1)\,,\,\mathrm{d}u_\theta(h_2)\rangle_{T_\theta\mathcal{M}}\,. \tag{2.44}$$

  As before, this construction is well defined only if $\mathrm{d}u_\theta$ is injective. Otherwise, it can be restricted to any supplementary space to $\operatorname{Ker}\mathrm{d}u_\theta$, for instance $\operatorname{Ker}\mathrm{d}u_\theta^\perp$, where $\perp$ denotes the orthogonal complement with respect to the canonical Euclidean inner product.

  Equivalently, this amounts to defining the Riemannian metric mapping

$$\rho : \begin{cases} \mathbb{R}^P & \longrightarrow & \mathcal{M}_P^+(\mathbb{R}) \\ \theta & \longmapsto & G_\theta := \Big(\langle \partial_p u_\theta\,,\,\partial_q u_\theta\rangle_{T_\theta\mathcal{M}}\Big)_{1\leqslant p,q\leqslant P} \end{cases}, \tag{2.45}$$

  where G_θ is called the *Gram matrix* at θ. If $\mathrm{d}u_\theta$ is not injective, G_θ becomes positive semidefinite, defining an inner product only on $\operatorname{Im} G_\theta = \operatorname{Ker}\mathrm{d}u_\theta^\perp$, where $\perp$ is again taken with respect to the canonical Euclidean inner product.

  In particular, following Proposition 2.33, we have in this case, for all $\theta \in \mathbb{R}^P$,

$$\nabla(\mathcal{L}\circ u)_{G_\theta} = {G_\theta}_{|\operatorname{Ker}\mathrm{d}u_\theta^\perp}{}^{-1}\nabla(\mathcal{L}\circ u)_\theta = G_\theta^\dagger\nabla(\mathcal{L}\circ u)_\theta, \tag{2.46}$$

  where $\nabla(\mathcal{L}\circ u)_{G_\theta}$ denotes the gradient with respect to the metric induced by G_θ, and $\nabla(\mathcal{L}\circ u)_\theta$ is the Euclidean gradient introduced in Proposition 2.30. This corresponds precisely to the setting of the natural gradient flow defined in Definition-Proposition 2.27.

We now turn to the most relevant framework for this dissertation: the case where the loss $\mathcal{L}$ is defined on a Hilbert space $\mathcal{H}$.

2.1.5 . Natural gradient flow induced by a Hilbertian gradient flow

Let $\big(\mathcal{H}, \langle\cdot\,,\,\cdot\rangle_{\mathcal{H}}\big)$ be a Hilbert space and let $\mathcal{L} : \mathcal{H} \to \mathbb{R}_+$ be a (Fréchet) differentiable functional loss. Let $u : \mathbb{R}^P \to \mathcal{H}$ be a parametric model whose image $\mathcal{M} := \operatorname{Im} u$ is endowed with the Riemannian structure induced by $\mathcal{H}$, *i.e.* for every $v \in \mathcal{M}$ the tangent space $T_v\mathcal{M}$ is a Hilbert subspace of $\mathcal{H}$ with the inner product inherited from $\langle\cdot\,,\,\cdot\rangle_{\mathcal{H}}$.

By restricting the loss to the submanifold, $\mathcal{L}_{|\mathcal{M}} : \mathcal{M} \to \mathbb{R}_+$, we can define the (functional) natural gradient flow as in Definition 2.22.

The subtility is that we then also have for all $v \in \mathcal{H}$ an hilbertian gradient $\nabla\mathcal{L}_v \in \mathcal{H}$ in the ambiant space $\mathcal{H}$.

The following proposition makes explicit the link between the hilbertian gradient and the natural gradient on the submanifold.

Proposition 2.35 (Natural gradient and orthogonal projection)**.** *For every $v \in \mathcal{M}$, let $\nabla\mathcal{L}_v \in \mathcal{H}$ denote the Riesz-Fréchet gradient of $\mathcal{L}$ at v in the ambient Hilbert space $\mathcal{H}$, i.e. the unique $\nabla\mathcal{L}_v \in \mathcal{H}$, such that for all $h \in \mathcal{H}$,*

$$d\mathcal{L}_v(h) = \langle \nabla\mathcal{L}_v \, , \, h \rangle_{\mathcal{H}} \, . \tag{2.47}$$

Then the gradient of the restricted loss on the manifold, $\nabla\mathcal{L}_{|\mathcal{M},\,v} \in T_v\mathcal{M}$, is the orthogonal projection of the ambient gradient onto the tangent space:

$$\nabla\mathcal{L}_{|\mathcal{M},\,v} \;=\; \Pi_{T_v\mathcal{M}}\big(\nabla\mathcal{L}_v\big), \tag{2.48}$$

where $\Pi_{T_v\mathcal{M}} : \mathcal{H} \to T_v\mathcal{M}$ denotes the orthogonal projector (in $\mathcal{H}$) onto $T_v\mathcal{M}$.

Proof. Fix $v \in \mathcal{M}$. Since $T_v\mathcal{M}$ is a closed subspace of $\mathcal{H}$, we have the orthogonal decomposition $\mathcal{H} = T_v\mathcal{M} \oplus^{\perp} (T_v\mathcal{M})^{\perp}$, and the orthogonal projector $\Pi_{T_v\mathcal{M}}$ is well defined and continuous. For any $h \in T_v\mathcal{M}$, using (2.47) and the fact that $h \in T_v\mathcal{M}$ while $\big(\nabla_{\mathcal{H}}\mathcal{L}_v - \Pi_{T_v\mathcal{M}}\nabla_{\mathcal{H}}\mathcal{L}_v\big) \in (T_v\mathcal{M})^{\perp}$, we obtain

$$\mathrm{d}\mathcal{L}_v(h) \;=\; \langle \nabla_{\mathcal{H}}\mathcal{L}_v \, , \, h \rangle_{\mathcal{H}} \;=\; \langle \Pi_{T_v\mathcal{M}}\nabla_{\mathcal{H}}\mathcal{L}_v \, , \, h \rangle_{\mathcal{H}} \, . \tag{2.49}$$

Since $T_v\mathcal{M}$ is a Hilbert space with the inner product inherited from $\mathcal{H}$, the Riesz representation theorem within $T_v\mathcal{M}$ asserts the existence and uniqueness of a vector $g_v \in T_v\mathcal{M}$ such that $\mathrm{d}\mathcal{L}_v(h) = \langle g_v \, , \, h \rangle_{\mathcal{H}}$ for all $h \in T_v\mathcal{M}$. Comparing with (2.49) yields

$$g_v \;=\; \Pi_{T_v\mathcal{M}}\nabla_{\mathcal{H}}\mathcal{L}_v. \tag{2.50}$$

By definition, $g_v = \nabla\mathcal{L}_{|\mathcal{M},\,v}$, which proves (2.48). □

***Remark* 2.36.** Proposition 2.35 immediately implies the following two gradient flows:

1. *Ambient (Hilbertian) gradient flow in $\mathcal{H}$:*

$$\frac{d}{dt}v(t) \;=\; -\nabla\mathcal{L}_{v(t)}. \tag{2.51}$$

2. *Natural (projected) gradient flow on the submanifold $\mathcal{M}$:*

$$\frac{d}{dt}v(t) \;=\; -\Pi_{T_{v(t)}\mathcal{M}}\big(\nabla\mathcal{L}_{v(t)}\big). \tag{2.52}$$

   In words: the natural gradient at v is the best feasible descent direction in $\mathcal{M}$, obtained by orthogonally projecting the ambient gradient onto $T_v\mathcal{M}$.

In particular, we obtain the following proposition.

Proposition 2.37. *Let $u : \mathbb{R}^P \to \mathcal{H}$ be a parametric model, and let $\mathcal{L} : \mathcal{H} \to \mathbb{R}_+$ be a loss function. Disregarding the initial condition, the parametric natural gradient flow associated with $\mathcal{L}$ can be written as follows:*

$$\frac{d}{dt}\theta(t) = du^{\dagger}_{\theta(t)}\Big(-\Pi_{T_{u_{\theta(t)}}\mathcal{M}}\big(\nabla\mathcal{L}_{u_{\theta(t)}}\big)\Big) \tag{2.53}$$

$$= -G^{\dagger}_{\theta(t)}\nabla_{\theta(t)}\ell_{\theta(t)}, \tag{2.54}$$

where ℓ is the parametric loss $\ell := \mathcal{L} \circ u$, $\nabla_{\theta(t)}\ell_{\theta(t)}$ denotes the gradient with respect to the canonical Euclidean inner product on $\mathbb{R}^P$, and $G_\theta(t)$ is the Gram matrix,

$$G_{\theta(t)} := \Big(\big\langle \partial_p u_{\theta(t)}\,,\,\partial_q u_{\theta(t)}\big\rangle_{T_{\theta(t)}\mathcal{M}}\Big)_{1\leqslant p,q\leqslant P}. \tag{2.55}$$

Proof. Equation (2.53) follows directly from Definition-Proposition 2.27 applied to Proposition 2.35. Equation (2.54) then results immediately from Proposition 2.33, as detailed in Equation (2.46). □

***Example* 2.38** (Explicit illustration of Remark 2.36 on the theoretical mean square error of Example 2.17)**.** Following Example 2.17, let $\mathcal{H} = \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$, and fix $f \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$. We consider the theoretical mean square error defined by

$$\mathcal{L}_2(v) = \frac{1}{2}\,\|v - f\|^2_{\mathrm{L}^2(\Omega\to\mathbb{R}^m,\mu)}. \tag{2.56}$$

As shown in Example 2.21, for every $v \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$, the (Hilbertian) gradient is given by

$$\nabla\mathcal{L}_{2_v} = v - f. \tag{2.57}$$

Ambient gradient flow. The *ambient* (Hilbertian) gradient flow in $\mathcal{H}$ satisfies

$$\frac{d}{dt}v(t) = -\nabla\mathcal{L}_{2_{v(t)}} = f - v(t), \tag{2.58}$$

whose solution is

$$v(t) = f + \big(v(0) - f\big)e^{-t}. \tag{2.59}$$

Hence, $v(t)$ converges exponentially fast to f in $\mathrm{L}^2(\Omega)$.

Natural (projected) gradient flow. Now suppose that we have a parametric model $u : \mathbb{R}^P \to \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$. Following Remark 2.36, the natural gradient flow of $\mathcal{L}_2$ on $\operatorname{Im} u$ is given by

$$\frac{d}{dt}u_{\theta(t)} = -\,\Pi_{T_{u_{\theta(t)}}\mathcal{M}}\left(\nabla\mathcal{L}_{2_{u_{\theta(t)}}}\right), \tag{2.60}$$

where $\Pi_{T_{u_{\theta(t)}}\mathcal{M}}$ denotes the orthogonal projection in $\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$ onto the tangent space

$$T_{u_{\theta(t)}}\mathcal{M} = \operatorname{Span}\big(\partial_p u_{\theta(t)}\ ;\ 1 \leqslant p \leqslant P\big). \tag{2.61}$$

Hence, the natural gradient flow of Equation (2.60) becomes

$$\frac{d}{dt}u_{\theta(t)} = -\Pi_{T_{u_{\theta(t)}}\mathcal{M}}\big(u_{\theta(t)} - f\big) = \Pi_{T_{u_{\theta(t)}}\mathcal{M}}\big(f - u_{\theta(t)}\big). \tag{2.62}$$

Comparison of the two flows.

- The *ambient gradient flow* $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = f - v(t)$ moves directly toward f in $\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$ and reaches it exponentially fast;
- The *natural gradient flow* restricts this motion to feasible directions $\partial_p u_\theta$ in the tangent space, ensuring that the trajectory remains within $\operatorname{Im} u$ at all times.

Hence, the natural flow identifies the steepest descent direction compatible with the representational capacity of the parametric model.

We illustrate Example 2.38 in Figure 2.2 below.

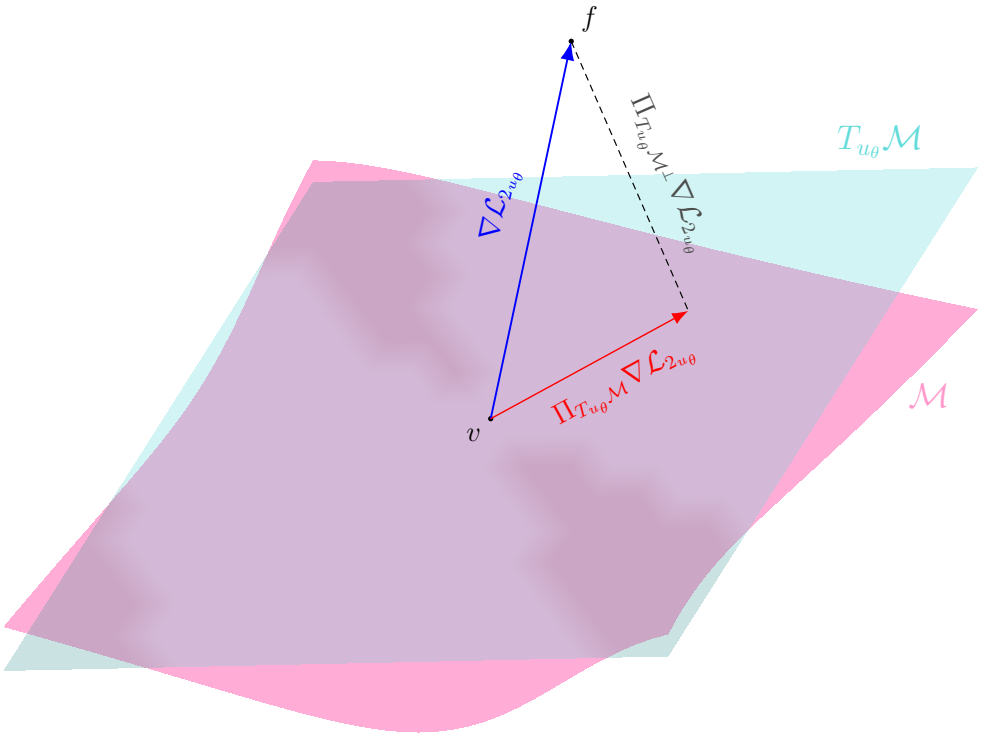


Figure 2.2: Illustration of the orthogonal projection of the functional gradient onto the tangent space. The ideal update direction is the functional gradient $\nabla \mathcal{L}_{2_{u_\theta}}$ (blue), but the parametric model u restricts updates to directions lying in the tangent space $T_{u_\theta}\mathcal{M}$ (green plane). The optimal feasible update is therefore the orthogonal projection $\Pi_{T_{u_\theta}\mathcal{M}}\Big(\nabla \mathcal{L}_{2_{u_\theta}}\Big)$ (red).

***Remark* 2.39.** From a certain perspective, the difference between the two gradient flows discussed in Example 2.38 can be interpreted as the consequence of replacing the functional space $\mathcal{H} = \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu)$ with the submanifold $\operatorname{Im} u$.

While the loss function $\mathrm{d}\mathcal{L}_2$ enjoys desirable properties in $\mathrm{L}^2(\Omega \to \mathbb{R})$, such as convexity in the present example, its elements cannot, in general, be computed on a computer. We therefore rely on the functions in $\operatorname{Im} u$, which are, by construction, computable and known to possess strong approximation properties in $\mathrm{L}^2(\Omega \to \mathbb{R})$ (see Theorem 2.13). However, this substitution comes at the cost of losing some of the "nice" theoretical properties of $\mathcal{L}_2$, due to the highly nonlinear nature of u.

It is important to emphasize that this nonlinearity is not an undesirable artifact but rather the very feature that grants neural networks their universal approximation capabilities in $\mathrm{L}^2(\Omega \to \mathbb{R})$.

***Remark* 2.40.** Example 2.38 highlights an important aspect of gradient flows defined above : neither the convergence to a critical point nor the optimality of that critical point (whether it is global, local, or even a minimum) is guaranteed in general. Indeed, this remains an open and active topic of research in the theory of neural networks. While general convergence results and bounds exist for *linear* neural networks trained by gradient flow or gradient descent (Arora et al., 2018), such guarantees under broad conditions are still lacking for deep architectures.

Recent advances by Boursier et al. (2022) established partial convergence results under strong regularity assumptions on the loss. Other works, such as Yang (2019); Chizat et al. (2019), have identified different asymptotic regimes, namely the *lazy*, *mean-field*, and *active* regimes, depending on the initialization $\boldsymbol{\theta_0}$ and the network width, while Jacot et al. (2022) analyzed in depth the dynamics in the active regime. Finally, Chizat and Bach (2018) obtained asymptotic global convergence results using a gradient flow structure inspired by Wasserstein geometry and optimal transport (Santambrogio, 2017), and Bach and Chizat (2021) further studied the asymptotic dynamics of gradient flows with respect to network width.

To conclude this section, let us formalize the formulation of the PINNs problem by defining its theoretical loss.

2.1.6 . PINNs theoretical loss

Let Ω be an open domain of $\mathbb{R}^n$, $n \in \mathbb{N}_1$, with compact closure $\overline{\Omega}$ and boundary $\partial\Omega$. We consider a differential operator

$$D : \mathcal{C}^\infty(\Omega) \longrightarrow \mathrm{L}^2(\Omega \to \mathbb{R}^{m_D}, \mu_n), \tag{2.63}$$

and a boundary operator

$$B : \mathcal{C}^\infty(\Omega) \longrightarrow \mathrm{L}^2(\partial\Omega \to \mathbb{R}^{m_B}, \sigma_n), \tag{2.64}$$

where μ_n and σ_n denote respectively a finite measure on Ω and a finite surface measure on $\partial\Omega$, with $m_D, m_B \in \mathbb{N}_1$.

Berezansky et al. (1996b, Chapter 12) shows how to construct a Hilbert space $\mathcal{H}_{D,B}$ such that both D and B can be extended to it while remaining differentiable. For now, we simply fix such a space without further specification.

Given $f \in \mathrm{L}^2(\Omega \to \mathbb{R}^{m_D}, \mu_n)$ and $g \in \mathrm{L}^2(\partial\Omega \to \mathbb{R}^{m_B}, \sigma_n)$, the PDE of interest can be written as

$$\begin{cases} D[u] = f & \text{in } \Omega, \\ B[u] = g & \text{on } \partial\Omega. \end{cases} \tag{2.65}$$

We can now define the corresponding PINNs loss.

Definition 2.41 (Theoretical loss of PINNs)**.** For all $v \in \mathcal{H}_{D,B}$, the *theoretical loss of PINNs* is defined as

$$\mathcal{L}_{D,B}(v) := \frac{1}{2} \left\| D[v] - f \right\|^2_{\mathrm{L}^2(\Omega \to \mathbb{R}^{m_D}, \mu_n)} + \frac{1}{2} \left\| B[v] - g \right\|^2_{\mathrm{L}^2(\partial\Omega \to \mathbb{R}^{m_B}, \sigma_n)}. \tag{2.66}$$

***Remark* 2.42.** It is straightforward to verify that, for all $v \in \mathcal{H}_{D,B}$,

$$\mathcal{L}_{D,B}(v) = 0 \quad \iff \quad v \text{ is a solution of Equation (2.65).}$$

This observation motivates the interpretation of minimizing $\mathcal{L}_{D,B}$ as approximating a solution of the PDE. Finding a "good minimizer" v for $\mathcal{L}_{D,B}$ therefore gives reasonable hope of obtaining a good approximation of a true solution to Equation (2.65). We will make this intuition rigorous in Chapter 5.

Let us now compute the differential of $\mathcal{L}_{D,B}$. For all $v \in \mathcal{H}_{D,B}$ and $h \in \mathcal{H}_{D,B}$, we have

$$\begin{aligned} \mathrm{d}\mathcal{L}_{D,B|v}(h) &= \left\langle D[v] - f \,,\, \mathrm{d}D_v(h) \right\rangle_{\mathrm{L}^2(\Omega \to \mathbb{R}^{m_D}, \mu_n)} + \left\langle B[v] - g \,,\, \mathrm{d}B_v(h) \right\rangle_{\mathrm{L}^2(\partial\Omega \to \mathbb{R}^{m_B}, \sigma_n)} \\ &= \left\langle \mathrm{d}D_v^* \left[D[v] - f \right] \,,\, h \right\rangle_{\mathcal{H}_{D,B}} + \left\langle \mathrm{d}B_v^* \left[B[v] - g \right] \,,\, h \right\rangle_{\mathcal{H}_{D,B}} \\ &= \left\langle \mathrm{d}D_v^* \left[D[v] - f \right] + \mathrm{d}B_v^* \left[B[v] - g \right] \,,\, h \right\rangle_{\mathcal{H}_{D,B}}. \end{aligned} \tag{2.67}$$

Hence, the gradient $\nabla\mathcal{L}_{D,B|v}$ of $\mathcal{L}_{D,B}$ at v in $\mathcal{H}_{D,B}$ is given by

$$\nabla\mathcal{L}_{D,B|v} = \mathrm{d}D_v^* \left[D[v] - f \right] + \mathrm{d}B_v^* \left[B[v] - g \right]. \tag{2.68}$$

Note that computing this gradient requires evaluating the adjoints $\mathrm{d}D_v^*$ and $\mathrm{d}B_v^*$, which is generally not straightforward in practice. In Chapter 4, we will show that it is possible to circumvent this difficulty by introducing an alternative notion of natural gradient specifically adapted to the PINNs framework.

While the formulation above provides a sound theoretical foundation for the analysis of PINNs, it cannot be implemented directly in a practical algorithmic setting. To bridge this gap, we will develop in the following section an approximate framework that remains faithful to the theoretical principles presented here.

2.2 . Approximations to the theoretical framework

While it is valuable to have a mathematically coherent framework, the theoretical constructions presented in Section 2.1, and in particular those related to PINNs in Section 2.1.6, cannot be directly implemented on a computer, as they involve non-computable operations.

Without delving into the formal theory of computation (Sipser, 2012), which lies beyond the scope of this work, we shall simply say that a function is computable if it can be evaluated using only a finite number of discrete operations.

From this perspective, the framework introduced above faces three main limitations preventing its direct implementation:

- The L^2-norms used in the definition of the theoretical PINN loss in Definition 2.41 cannot be computed exactly, as they involve integrals over continuous domains. We must therefore approximate them using suitable quadrature rules. This will be addressed in Section 2.2.2.
- The gradient flows introduced in Definition-Proposition 2.27 and Definition 2.31 cannot be solved exactly either. We must thus replace them with discrete iterative schemes, which will be discussed in Section 2.2.4.
- In the case of PINNs, it may also be necessary to approximate the operators D and B in Equation (2.65), as well as their differentials. However, we will see that for differential operators, these computations can be carried out exactly by leveraging *automatic differentiation*. A careful presentation of this method will be given in Section 2.2.1.

In the following section, we begin by introducing the framework of automatic differentiation.

2.2.1 . Automatic differentiation

In this section, we introduce the principle of *automatic differentiation* (often abbreviated as *autodiff*), also referred to as *algorithmic differentiation*, following the clear exposition of Baydin et al. (2018). Automatic differentiation is a collection of techniques designed to compute numerical derivatives with near-exact precision[1].

[1]Up to round-off errors due to finite-bit arithmetic, which inherently limits accuracy and performs poorly on strongly multi-scale quantities.

Broadly speaking, methods for computing derivatives can be grouped into four families: manual differentiation, finite-difference approximations, symbolic differentiation, and automatic differentiation.

Computing derivatives by hand is extremely time-consuming and error-prone, and should be avoided whenever possible.

Symbolic differentiation automates this process using a small set of differentiation rules[2], guaranteeing exact results. However, it often leads to exponentially growing and opaque symbolic expressions, a phenomenon known as *expression swell* (Corliss, 1988). Moreover, it requires explicit closed-form expressions of the functions to be differentiated, which is impractical for algorithms that are defined only as computational procedures.

Finite-difference approximations, on the other hand, estimate a directional derivative of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ along a direction $r \in \mathbb{R}^n$, at a point $x \in \mathbb{R}^n$, by returning to the definition of the derivative as a limit:

$$\partial_r f(x) \simeq \frac{f(x + hr) - f(x)}{h} \quad \text{or} \quad \partial_r f(x) \simeq \frac{f(x + hr) - f(x - hr)}{2h}, \tag{2.69}$$

where $h \in (0, 1]$ is a small scalar step. This method scales linearly with the size of the Jacobian (here nm) and is therefore relatively inexpensive. It also requires only evaluations of f, which makes it highly flexible. However, it suffers from ill-conditioning and numerical instability due to the competing effects of truncation errors[3] and round-off errors, the latter growing as the former decreases (Jerrell, 1997).

Automatic differentiation seeks to combine the best features of symbolic and finite-difference differentiation: the precision of the former and the practicality of the latter. Its central insight is that any numerical algorithm can ultimately be decomposed into a finite sequence of elementary operations whose derivatives are analytically known (Griewank and Walther, 2008). The key idea is to perform the algorithm's original computation while simultaneously propagating derivative information through it, by applying simple differentiation rules[2] at each step. In contrast to symbolic differentiation, these rules are applied not to abstract expressions but to concrete numerical values, as in finite-difference methods, thus allowing autodiff to compute derivatives of algorithms that involve loops or conditionals, while maintaining near-exact numerical accuracy[1].

To make this more concrete, consider two differentiable functions $g : \mathbb{R} \to \mathbb{R}^m$ and $\rho : \mathbb{R} \to \mathbb{R}$, and define

$$f : x \in \mathbb{R}^n \longmapsto \left(\sum_{i=1}^{n} \rho(x_i) \right) g(x_n) \in \mathbb{R}^m. \tag{2.70}$$

[2]For example, the Leibniz product rule, the chain rule, and the derivatives of polynomials and elementary functions.

[3]*i.e.* inaccuracies due to h being finite rather than infinitesimal.

Given an evaluation point $x \in \mathbb{R}^n$ and a *differentiation direction* $r \in \mathbb{R}^n$, or a *partial differentiation codirection* $b \in \mathbb{R}^m$, automatic differentiation can be carried out in two distinct ways:

- in *forward mode*, to compute $\partial_r f(x)$;
- in *reverse mode* (or *backward mode*), to compute $Jf(x)^\top b$, where $Jf(x)$ denotes the Jacobian matrix of f evaluated at x, and $^\top$ denotes transposition.

Forward mode. In forward mode, derivatives are propagated alongside the original computation by initializing a differentiation variable $\dot{x} = r$ and applying the chain rule to each intermediate operation. The resulting sequence of computations is summarized in Table 2.2 below.

Sequence of operations to compute $f(x)$

v_{-n+i}	$= x_i$	$1 \leqslant i \leqslant n$
v_i	$= \rho(v_{-n+i})$	$1 \leqslant i \leqslant n$
v_{n+1}	$= \sum_{i=1}^n v_i$	
v_{n+1+j}	$= g(v_0)_j$	$1 \leqslant j \leqslant m$
$v_{n+1+m+j}$	$= v_{n+1+j} v_{n+1}$	$1 \leqslant j \leqslant m$
$\boldsymbol{f(x)_j}$	$= \boldsymbol{v_{n+1+m+j}}$	$1 \leqslant j \leqslant m$

Sequence of operations to compute $\partial_r f(x)$

$\dot{v}_{-n+i}$	$= r_i$	$1 \leqslant i \leqslant n$
$\dot{v}_i$	$= \rho'(v_{-n+i}) \dot{v}_{-n+i}$	$1 \leqslant i \leqslant n$
$\dot{v}_{n+1}$	$= \sum_{i=1}^n \dot{v}_i$	
$\dot{v}_{n+1+j}$	$= g'(v_0)_j \dot{v}_0$	$1 \leqslant j \leqslant m$
$\dot{v}_{n+1+m+j}$	$= \dot{v}_{n+1+j} v_{n+1} + v_{n+1+j} \dot{v}_{n+1}$	$1 \leqslant j \leqslant m$
$\boldsymbol{\partial_r f(x)_j}$	$= \boldsymbol{\dot{v}_{n+1+m+j}}$	$1 \leqslant j \leqslant m$

Table 2.2: Forward mode of automatic differentiation applied to $f : x \in \mathbb{R}^n \mapsto \left(\sum_{i=1}^n \rho(x_i)\right) g(x_n) \in \mathbb{R}^m$, where $g : \mathbb{R} \to \mathbb{R}^m$ and $\rho : \mathbb{R} \to \mathbb{R}$ are differentiable functions with known derivatives g' and ρ' (adapted from Baydin et al. 2018).
The left table records the sequence of intermediate operations required to compute $f(x)$, using the variables $(v_k)_{k=1-n}^{n+1+2m}$. The right table illustrates how the directional derivative $\partial_r f(x)$ for a given $r \in \mathbb{R}^n$ can be efficiently computed by initializing the variables $(\dot{v}_k)_{k=1-n}^{n+1+2m}$ and updating them iteratively according to the differentiation rules. These updates are performed in parallel with the forward evaluation of f, using both the stored intermediate values $(v_k)_{k=1-n}^{n+1+2m}$ and the known derivatives g' and ρ'.

As a consequence, this mode is particularly convenient when $n \ll m$, since in the extreme case where $n = 1$, the derivative of f can be obtained after applying the above procedure[4] only once. In contrast, when $n \gg m$, obtaining the full Jacobian of f requires applying the same procedure n times (by iterating r over the canonical basis vectors $(e_i)_{1 \leqslant i \leqslant n}$ of $\mathbb{R}^n$), which quickly becomes computationally inefficient, especially in the extreme case where $m = 1$. Unfortunately, this latter situation is precisely the one encountered when computing

[4]Often referred to as the *forward pass*, which gives its name to the mode.

the Euclidean gradient (see Proposition 2.30) of a parametric loss $\ell : \mathbb{R}^P \to \mathbb{R}_+$ (see Definition-Proposition 2.28). It is therefore the typical setting in the context of the gradient descent algorithm introduced in Section 2.2.4 below. Fortunately, the reverse mode presented below efficiently bridges this gap.

Reverse (or backward) mode. The reverse mode is less intuitive than the forward mode, as differentiation is not performed simultaneously with the evaluation of g, ρ, and ultimately f. Instead, it records the sequence of elementary operations performed, namely, the evaluations of g and ρ at x_n and $(x_i)_{1 \leqslant i \leqslant n}$, respectively, the summation of the $\big(\rho(x_i)\big)_{1 \leqslant i \leqslant n}$, and the final multiplication of $g(x_n)$ by $\sum_{i=1}^n \rho(x_i)$. During this process, it also stores the partial derivatives of each elementary operation with respect to the subsequent ones to which they are connected, using the backward chain rule and the derivatives of these elementary operations.

Once the evaluation of f at x has been completed, the derivatives are then computed in the reverse (or *backward*) direction[5], by propagating through the recorded dependencies between the elementary operations, starting from the so-called "partial differentiation codirection" b. This procedure is illustrated in Table 2.3 below.

[5]Hence the mode's name

Sequence of operations to compute $f(x)$

v_{-n+i}	$= x_i$	$1 \leqslant i \leqslant n$
v_i	$= \rho(v_{-n+i})$	$1 \leqslant i \leqslant n$
v_{n+1}	$= \sum_{i=1}^{n} v_i$	
v_{n+1+j}	$= g(v_0)_j$	$1 \leqslant j \leqslant m$
$v_{n+1+m+j}$	$= v_{n+1+j} v_{n+1}$	$1 \leqslant j \leqslant m$
$\boldsymbol{f(x)_j}$	$= \boldsymbol{v_{n+1+m+j}}$	$1 \leqslant j \leqslant m$

Sequence of operations to compute $Jf(x)^\top b$

$\boldsymbol{Jf(x)^\top b_i}$	$= \boldsymbol{\overline{v}_{-n+i}}$	$1 \leqslant i \leqslant n$
$\overline{v}_{-n+i}$	$= \overline{v}_i \frac{\partial v_i}{\partial v_{-n+i}}$	$1 \leqslant i \leqslant n$
	$= \overline{v}_i \, \rho'(v_{-n+i})$	
$\overline{v}_0$	$= \sum_{j=1}^{m} \overline{v}_{n+1+j} \frac{\partial v_{n+1+j}}{\partial v_0} + \overline{v}_n \frac{\partial v_n}{\partial v_0}$	
	$= \sum_{j=1}^{m} \overline{v}_{n+1+j} \, g'(v_0)_n + \overline{v}_n \rho'(v_0)$	
$\overline{v}_i$	$= \overline{v}_{n+1} \frac{\partial v_{n+1}}{\partial v_i}$	$1 \leqslant i \leqslant n$
	$= \overline{v}_{n+1}$	
$\overline{v}_{n+1}$	$= \sum_{j=1}^{m} \overline{v}_{n+1+m+j} \frac{\partial v_{n+1+j}}{\partial v_{n+1}}$	
	$= \sum_{j=1}^{m} \overline{v}_{n+1+m+j} \, v_{n+1+j}$	
$\overline{v}_{n+1+j}$	$= \overline{v}_{n+1+m+j} \frac{\partial v_{n+1+m+j}}{\partial v_{n+1+j}}$	$1 \leqslant j \leqslant m$
	$= \overline{v}_{n+1+m+j} \, v_{n+1}$	
$\boldsymbol{\overline{v}_{n+1+m+j}}$	$= \boldsymbol{b_j}$	$1 \leqslant j \leqslant m$

Table 2.3: Reverse mode of automatic differentiation applied to $f : x \in \mathbb{R}^n \mapsto \left(\sum_{i=1}^{n} \rho(x_i)\right) g(x_n) \in \mathbb{R}^m$, with given functions $g : \mathbb{R} \to \mathbb{R}^m$ and $\rho : \mathbb{R} \to \mathbb{R}$, whose respective derivatives g' and ρ' are known (adapted from Baydin et al. 2018).
The left table records the sequence of intermediate operations performed to compute $f(x)$, storing their corresponding values in the variables $(v_k)_{k=1-n}^{n+1+2m}$. The right table illustrates how the computation of $Jf(x)^\top b$ for a given $b \in \mathbb{R}^m$ can be efficiently carried out by tracking the dependencies between the variables $(v_k)_{k=1-n}^{n+1+2m}$ from the left table and computing their relative derivatives $\frac{\partial v_{i+k}}{\partial v_i}$ for $1-n \leqslant i < i+k \leqslant n+1+2m$, whenever v_{i+k} depends on v_i. These derivatives are computed using the standard differentiation rules, together with the recorded values of $(v_k)_{k=1-n}^{n+1+2m}$ and the derivative functions g' and ρ'.
Once the forward computation (left table) has been completed, one can iteratively compute the adjoint variables $(\overline{v}_k)_{k=1-n}^{n+1+2m}$ by propagating backward from the partial differentiation codirection $b \in \mathbb{R}^m$, using the stored partial derivatives $\frac{\partial v_{i+k}}{\partial v_i}$ for $1-n \leqslant i < i+k \leqslant n+1+2m$ whenever v_{i+k} depends on v_i.

As discussed above, this mode is particularly efficient when $n \gg m$, since, in the extreme case where $m = 1$, the gradient of f can be obtained by applying the procedure described previously only once, simply by setting $b = 1$. Conversely, when $n \ll m$, computing the entire Jacobian of f requires repeating the same procedure m times (iterating b over the canonical basis vectors $(\ell_i)_{1 \leqslant i \leqslant m}$ of $\mathbb{R}^m$), which makes the method inefficient—especially in the limiting case where $n = 1$.

Beyond gradient descent, presented in Section 2.2.4, for which it was originally introduced in deep learning[6], automatic differentiation can also be used

[6]The combination of gradient descent and automatic differentiation has become so

to compute the derivatives of neural network outputs with respect to their inputs. Moreover, since this operation itself constitutes a numerical computation, it can be differentiated again using the same procedure. This recursive capability forms the practical foundation of *Physics-Informed Neural Networks (PINNs)*.

It is worth emphasizing that, beyond numerical improvements, this is precisely the conceptual innovation introduced by Raissi et al. (2019b) compared to the earlier work of Lagaris et al. (1998), which largely explains why interest in PINNs truly surged only after the former's publication. More concretely, if D or B in Definition 2.41 are differential operators, they can be evaluated on the image of a parametric model $u : \mathbb{R}^P \to \mathcal{F}(\mathbb{R}^n \to \mathbb{R}^m)$, itself evaluated at a given $\theta \in \mathbb{R}^P$ and point $x \in \Omega$. The result of this operation can, in turn, be differentiated with respect to θ. In particular, one could then apply backpropagation to $\mathcal{L}_{D,B} \circ u$, where $\mathcal{L}_{D,B}$ denotes the theoretical PINNs loss of Definition 2.41. However, this cannot be done directly, since $\mathcal{L}_{D,B}$ involves computing integrals, which are not tractable in practice.

Approximating these integrals is precisely the objective of the next section. This discussion will also naturally introduce the concept of the *Neural Tangent Kernel (NTK)*, originally proposed by Jacot et al. (2018).

2.2.2 . Loss approximation

In this section, we discuss how to approximate the loss defined in Equation (2.66) so that it can be effectively implemented on a computer. For clarity, let us first recall the expression of Equation (2.66):

$$\mathcal{L}_{D,B}(v) := \frac{1}{2} \|D[v] - f\|^2_{\mathrm{L}^2(\Omega \to \mathbb{R}^{m_D}, \mu_n)} + \frac{1}{2} \|B[v] - g\|^2_{\mathrm{L}^2(\partial\Omega \to \mathbb{R}^{m_B}, \sigma_n)} .$$

By definition, this loss can be equivalently written as:

$$\mathcal{L}_{\mathrm{PINN}}(v) := \int_\Omega \big\| \big(D[v] - f\big)(x) \big\|^2_{\mathbb{R}^{m_D}} \, \mu_n(\mathrm{d}x) + \int_{\partial\Omega} \big\| \big(B[v] - g\big)(x) \big\|^2_{\mathbb{R}^{m_B}} \, \sigma_n(\mathrm{d}x). \tag{2.71}$$

As seen in Section 2.2.1, both D and B can be computed efficiently through automatic differentiation. The norms $\|\cdot\|_{\mathbb{R}^{m_D}}$ and $\|\cdot\|_{\mathbb{R}^{m_B}}$ are standard Euclidean norms on $\mathbb{R}^{m_D}$ and $\mathbb{R}^{m_B}$, and are therefore directly computable. Hence, the only non-computable elements in Equation (2.71) are the two integrals themselves.

While numerical integration is a well-established topic (Davis and Rabinowitz, 2007), the vast majority of loss approximations used in deep learning rely on Monte Carlo or quasi-Monte Carlo methods (Lemieux, 2009). In essence, given a measurable space A endowed with a finite measure μ and an

central to deep learning that it is now commonly referred to as *backpropagation*.

integrable function $f : A \to \mathbb{R}$, (quasi-)Monte Carlo methods approximate the integral

$$\int_A f(x)\,\mu(\mathrm{d}x)$$

by sampling $k \in \mathbb{N}_1$ points $(x_i)_{1 \leqslant i \leqslant k}$ according to the normalized probability distribution $\mathbb{P}_\mu := \frac{\mu}{\mu(A)}$, and setting

$$\int_A f(x)\,\mu(\mathrm{d}x) \;\simeq\; \frac{\mu(A)}{k} \sum_{i=1}^{k} f(x_i).$$

This approximation is justified by the law of large numbers. Indeed, if X is a random variable following the law $\mathbb{P}_\mu$, and $(x_i)_{i \in \mathbb{N}_1}$ are independent and identically distributed copies of X, then

$$\mathbb{P}\left(\lim_{k \to +\infty} \frac{1}{k} \sum_{i=1}^{k} f(x_i) = \mathbb{E}[f(X)] = \int_A f(x)\,\mathbb{P}_\mu(\mathrm{d}x) = \frac{1}{\mu(A)} \int_A f(x)\,\mu(\mathrm{d}x) \right) = 1.$$

There are three compelling reasons for using (quasi-)Monte Carlo methods in deep learning:

1. **Empirical data availability.** In most deep learning scenarios, only a finite set of samples drawn from an underlying data distribution is available, typically originating from experiments or observations. Monte Carlo methods naturally approximate the theoretical loss integral using this empirical data, providing a practical and statistically grounded estimation.

2. **Computational efficiency.** Monte Carlo estimators are extremely cheap to compute. Apart from evaluating f, one only needs to perform a simple summation, followed by a scalar multiplication. These operations are trivially parallelizable and thus highly efficient on modern hardware.

3. **Scalability with dimension.** Although (quasi-)Monte Carlo methods converge more slowly than classical quadrature techniques, their convergence rate does not deteriorate with the dimension of the space. This property makes them particularly well suited to deep learning applications, where data often lie in very high-dimensional spaces (for instance, images may have dimensions on the order of one million).

Since $\overline{\Omega}$ is compact, both $\mu_n(\Omega)$ and $\sigma_n(\partial\Omega)$ are finite. We can therefore apply a Monte Carlo approximation to the loss defined in Equation (2.71), leading to the following definition of the empirical loss for PINNs.

Definition 2.43 (Empirical loss of PINNs)**.** Let $v \in \mathcal{H}_{D,B}$, where $\mathcal{H}_{D,B}$ is defined as in Section 2.1.6, and let $S_D, S_B \in \mathbb{N}_1$. Given samples $(x_{D,i})_{1 \leqslant i \leqslant S_D}$ and $(x_{B,i})_{1 \leqslant i \leqslant S_B}$ drawn respectively from the normalized probability distributions $\mathbb{P}_{\mu_n} := \frac{\mu_n}{\mu_n(\Omega)}$ and $\mathbb{P}_{\sigma_n} := \frac{\sigma_n}{\sigma_n(\partial\Omega)}$, the empirical loss of PINNs is defined by

$$\begin{aligned} \mathcal{L}_{X_D,X_B}(v) :=& \frac{\mu_n(\Omega)}{2\,S_D} \sum_{i=1}^{S_D} \big\| \big(D[v] - f\big)(x_{D,i}) \big\|_{\mathbb{R}^{m_D}}^2 \\ &+ \frac{\sigma_n(\partial\Omega)}{2\,S_B} \sum_{i=1}^{S_B} \big\| \big(B[v] - g\big)(x_{B,i}) \big\|_{\mathbb{R}^{m_B}}^2 . \end{aligned} \tag{2.72}$$

***Remark* 2.44.** The samples X_D and X_B need not be fixed throughout training. In practice, and particularly in the context of PINNs, one may employ resampling strategies, as mentioned in the Introduction. More generally, in deep learning, it is common to evaluate the empirical loss Equation (2.72) on randomly selected subsets of the available samples, called *mini-batches*, and to update these subsets at each iteration of the gradient descent procedure introduced in Section 2.2.4. Over successive iterations, this ensures that the entire dataset is visited, giving rise to the well-known *stochastic gradient descent* (SGD) method.

Let us now determine under which conditions Definition 2.43 indeed defines a valid loss function in the sense of Definition 2.14. The key point is to establish the differentiability of $\mathcal{L}$. To simplify the discussion, let us consider the theoretical Mean Square Error (MSE) introduced in Example 2.17. The construction of its empirical counterpart follows the same reasoning as in Definition 2.43.

Definition 2.45 (Empirical MSE)**.** Let $S \in \mathbb{N}_1$, and let $(x_i)_{1 \leqslant i \leqslant S}$ be samples drawn from the normalized probability distribution $\mathbb{P}_{\mu_n} := \frac{\mu_n}{\mu_n(\Omega)}$. For a fixed $f \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)$, the empirical Mean Square Error is defined for all $v \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)$ by

$$\mathcal{L}_{2,X}(v) := \frac{\mu_n(\Omega)}{2\,S} \sum_{i=1}^{S} \|v(x_i) - f(x_i)\|_{\mathbb{R}^m}^2 . \tag{2.73}$$

Note that Definition 2.45 implicitly assumes that point evaluations $v(x_i)$ are well defined for all $1 \leqslant i \leqslant S$, which is not generally the case in $\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)$. However, let us set this issue aside for now and try to write formally the differential of the empirical MSE.

For any $h \in \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)$, we have

$$\begin{aligned}\mathcal{L}_{2,X}(v+h) - \mathcal{L}_{2,X}(v) =& \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle v(x_i) - f(x_i)\,,\, h(x_i)\rangle_{\mathbb{R}^m} \\ &+ \frac{\mu_n(\Omega)}{2\,S} \sum_{i=1}^{S} \|h(x_i)\|^2_{\mathbb{R}^m}\,.\end{aligned} \tag{2.74}$$

We therefore see that $\mathcal{L}_{2,X}$ is differentiable if and only if

$$\frac{\mu_n(\Omega)}{2\,S} \sum_{i=1}^{S} \|h(x_i)\|^2_{\mathbb{R}^m} = o\Big(\|h\|_{\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)}\Big)\,. \tag{2.75}$$

This condition holds in particular when the point evaluations $(v \mapsto v(x_i))_{1 \leqslant i \leqslant S}$ are bounded linear functionals with respect to the L^2-norm.

In that case, the differential of $\mathcal{L}_{2,X}$ at v is given by

$$\mathrm{d}\mathcal{L}_{2,X_v}(h) = \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle v(x_i) - f(x_i)\,,\, h(x_i)\rangle_{\mathbb{R}^m}\,. \tag{2.76}$$

Assuming this to hold, let us denote by δ_{x_i} the point evaluation at x_i. Then, formally, we can write

$$\begin{aligned}\mathrm{d}\mathcal{L}_{2,X_v}(h) &= \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \Big\langle v(x_i) - f(x_i)\,,\, \langle \delta_{x_i}\,,\, h\rangle_{\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)} \Big\rangle_{\mathbb{R}^m} \\ &= \left\langle \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle v(x_i) - f(x_i)\,,\, \delta_{x_i}\rangle_{\mathbb{R}^m}\,,\, h \right\rangle_{\mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)}\,.\end{aligned} \tag{2.77}$$

Hence, in this formal setting, the gradient of $\mathcal{L}_{2,X}$ at v is given by

$$\nabla \mathcal{L}_{2,X_v} = \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle v(x_i) - f(x_i)\,,\, \delta_{x_i}\rangle_{\mathbb{R}^m}\,. \tag{2.78}$$

This observation naturally motivates the introduction of Reproducing Kernel Hilbert Spaces (RKHS), whose framework allows point evaluations to be represented as continuous linear functionals. A detailed overview of this theory will be provided in Chapter 3.

For now, we adopt a different perspective that will naturally lead us to introduce the notion of the *Neural Tangent Kernel (NTK)*, originally proposed by Jacot et al. (2018). The precise connection between these two viewpoints is thoroughly developed in our first contribution, presented in Chapter 4.

2.2.3 . (Natural) Neural Tangent Kernel (N)NTK

Let $u : \mathbb{R}^P \to \mathrm{L}^2(\Omega \to \mathbb{R}^m, \mu_n)$ be a parametric model in the sense of Definition 2.8. We can then define the parametric loss $\widehat{\ell}_2 := \mathcal{L}_{2,X} \circ u$ (see Definition-Proposition 2.28) associated with the empirical functional loss $\mathcal{L}_{2,X}$ of Equation (2.73) and the parametric model u, for all $\theta \in \mathbb{R}^P$.

If, for all $\theta \in \mathbb{R}^P$, the evaluations of u_θ at the samples $(x_i)_{1 \leqslant i \leqslant S}$ are bounded, and provided that the evaluations of $f \in \mathrm{L}^2$ at these points are also well defined, then $\widehat{\ell}_2$ defines a continuously differentiable loss.

Given $\boldsymbol{\theta_0} \in \mathbb{R}^P$, we can therefore define the classical gradient flow associated with $\widehat{\ell}_2$ (see Definition 2.31), namely

$$\begin{cases} \theta(0) & = \boldsymbol{\theta_0} \\ \dfrac{d}{dt}\,\theta(t) & = -\nabla \widehat{\ell}_{2,X_{\theta(t)}}, \end{cases} \tag{2.79}$$

where $\nabla \widehat{\ell}_{2,X_{\theta(t)}}$ denotes the Euclidean gradient of $\widehat{\ell}_{2,X}$.

By Proposition 2.30, we know that for all $\theta \in \mathbb{R}^P$ and $1 \leqslant p \leqslant P$,

$$\begin{aligned} \left(\nabla \widehat{\ell}_{2,X_\theta}\right)_p = \partial_p \widehat{\ell}_{2,X_\theta} = \mathrm{d}(\mathcal{L}_{2,X} \circ u)_\theta\,(\boldsymbol{e}_p) &\overset{\text{chain rule}}{=} \mathrm{d}\mathcal{L}_{2,X_{u(\theta)}}(\mathrm{d}u_\theta(\boldsymbol{e}_p)) \\ &\overset{(2.76)}{=} \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle v(x_i) - f(x_i)\,,\, \partial_p u_\theta(x_i) \rangle_{\mathbb{R}^m}\,. \end{aligned} \tag{2.80}$$

We can then study the induced gradient flow on $v := u \circ \theta$, which, by the chain rule, satisfies

$$\begin{cases} v(0) & = u_{\boldsymbol{\theta_0}} \\ \dfrac{d}{dt}\,v(t) & = \mathrm{d}u_{\theta(t)}\left(-\nabla \widehat{\ell}_{2,X_{\theta(t)}}\right). \end{cases} \tag{2.81}$$

In particular, for all $t \in \mathbb{R}_+$, we have

$$\begin{aligned} \frac{d}{dt}\,v(t) &= \mathrm{d}u_{\theta(t)}\left(-\frac{\mu_n(\Omega)}{S} \sum_{p=1}^{P} \sum_{i=1}^{S} \left\langle v(x_i) - f(x_i)\,,\, \partial_p u_{\theta(t)}(x_i) \right\rangle_{\mathbb{R}^m} \boldsymbol{e}_p\right) \\ &= \frac{\mu_n(\Omega)}{S} \sum_{p=1}^{P} \sum_{i=1}^{S} \left\langle f(x_i) - v(x_i)\,,\, \partial_p u_{\theta(t)}(x_i) \right\rangle_{\mathbb{R}^m} \mathrm{d}u_{\theta(t)}(\boldsymbol{e}_p) \\ &= \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \left\langle f(x_i) - v(x_i)\,,\, \sum_{p=1}^{P} \partial_p u_{\theta(t)}(x_i) \otimes \partial_p u_{\theta(t)} \right\rangle_{\mathbb{R}^m}. \end{aligned} \tag{2.82}$$

We can now identify the *Neural Tangent Kernel* (NTK; Jacot et al., 2018) at $\theta \in \mathbb{R}^P$, defined for all $x, y \in \Omega$ as

$$\mathrm{NTK}_\theta(x, y) := \sum_{p=1}^{P} \partial_p u_\theta(x) \otimes \partial_p u_\theta(y). \tag{2.83}$$

***Remark* 2.46.** The NTK can be interpreted as a measure of how a perturbation at a point y influences the output at another point x. In particular, Equation (2.82) shows that the infinitesimal evolution of u_θ at a point $x \in \Omega$ is driven by the average of the local errors $(f(x_i) - u_\theta(x_i))$, each weighted by the kernel $\mathrm{NTK}_\theta(x_i, x)$, which quantifies their relative influence.

As noted earlier in Section 2.1.4, choosing the Euclidean gradient implicitly endows $\operatorname{Im} u$ with the Riemannian structure given by the pullback metric of the canonical Euclidean metric on $\mathbb{R}^P$ through the differential $\mathrm{d}u_\theta$. In this setting, Proposition 2.34 shows that the expression in Equation (2.82) corresponds to the functional gradient $\nabla_{\theta(t)}\mathcal{L}_{2,X}$ of $\mathcal{L}_{2,X}$ at $\theta(t)$ for the Euclidean pullback metric, that is, for any $\theta \in \mathbb{R}^P$:

$$\nabla_\theta \mathcal{L}_{2,X} = \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle f(x_i) - v(x_i)\,,\, \mathrm{NTK}_\theta(x_i, \cdot)\rangle_{\mathbb{R}^m}\,. \tag{2.84}$$

In particular, by comparing this expression with Equation (2.78), we see that they coincide formally up to the identification

$$\delta_{x_i} = \mathrm{NTK}_\theta(x_i, \cdot). \tag{2.85}$$

Making this correspondence rigorous and well-defined is precisely one of the main motivations for introducing the framework of Reproducing Kernel Hilbert Spaces (RKHS), which will be developed in Chapter 3.

Suppose now, for simplicity, that $m = 1$. In this case, the NTK defined in Equation (2.83) reduces, for all $\theta \in \mathbb{R}^P$ and $x, y \in \Omega$, to

$$\mathrm{NTK}_\theta(x, y) := \sum_{p=1}^{P} \partial_p u_\theta(x)\, \partial_p u_\theta(y), \tag{2.86}$$

and therefore, for all $1 \leqslant i \leqslant S$,

$$\mathrm{NTK}_\theta(x_i, \cdot) \in \operatorname{Im} \mathrm{d}u_\theta. \tag{2.87}$$

In particular, from Equation (2.84), we observe that

$$\nabla_\theta \mathcal{L}_{2,X} \in \operatorname{Span}\big(\mathrm{NTK}_\theta(x_i, \cdot) \,:\, 1 \leqslant i \leqslant S\big) \subset \operatorname{Im} \mathrm{d}u_\theta. \tag{2.88}$$

The space $\operatorname{Span}\big(\mathrm{NTK}_\theta(x_i, \cdot) \,:\, 1 \leqslant i \leqslant S\big)$ plays a central role in our first contribution, presented in Chapter 4.

More generally, if $m > 1$, we can introduce, for all $1 \leqslant i \leqslant S$ and $1 \leqslant j \leqslant m$,

$$\mathrm{NTK}_\theta(x_i, \cdot)_j = \big(\mathrm{NTK}_\theta(x_i, \cdot)\big)_{j,\cdot} = \sum_{p=1}^{P} \big(\partial_p u_\theta(x_i)\big)_j\, \partial_p u_\theta. \tag{2.89}$$

We can then replace $\mathrm{Span}\left(\mathrm{NTK}_\theta(x_i,\cdot) \,:\, 1 \leqslant i \leqslant S\right)$ with

$$\mathrm{Span}\left(\mathrm{NTK}_\theta(x_i,\cdot)_j \,:\, 1 \leqslant i \leqslant S,\ 1 \leqslant j \leqslant m\right), \tag{2.90}$$

so that, by Equation (2.84),

$$\nabla_\theta \mathcal{L}_{2,X} \in \mathrm{Span}\left(\mathrm{NTK}_\theta(x_i,\cdot)_j \,:\, 1 \leqslant i \leqslant S,\ 1 \leqslant j \leqslant m\right) \subset \mathrm{Im}\,\mathrm{d}u_\theta. \tag{2.91}$$

More generally, if we consider a non-Euclidean Riemannian metric on $\mathbb{R}^P$, that is, a mapping $\rho : \mathbb{R}^P \to \mathcal{M}_P^+(\mathbb{R})$ as introduced in Section 2.1.4, then, following the same line of reasoning, and given $\boldsymbol{\theta_0}$, the induced gradient flow on $v := u \circ \theta$ satisfies, by the chain rule,

$$\begin{cases} v(0) & = u_{\boldsymbol{\theta_0}} \\ \dfrac{d}{dt}\, v(t) & = \mathrm{d}u_{\theta(t)}\Big(-\nabla\widehat{\ell}_{2,X_{\rho(\theta(t))}}\Big)\,, \end{cases} \tag{2.92}$$

where $\nabla\widehat{\ell}_{2,X_{\rho(\theta(t))}}$ denotes the gradient of $\widehat{\ell}_{2,X}$ with respect to the inner product $\langle\cdot\,,\,\cdot\rangle_{\rho(\theta(t))}$.

By Proposition 2.33 and its generalization in Equation (2.46), we have for all $\theta \in \mathbb{R}^P$,

$$\nabla\widehat{\ell}_{2,X_{\rho(\theta)}} = \rho(\theta)^\dagger \nabla\widehat{\ell}_{2,X_\theta}, \tag{2.93}$$

where $\nabla\widehat{\ell}_{2,X_\theta}$ is the Euclidean gradient. Injecting this into Equation (2.92) and expanding, we obtain

$$\begin{aligned} \frac{d}{dt}\, v(t) &= \mathrm{d}u_{\theta(t)}\left(-\frac{\mu_n(\Omega)}{S}\sum_{p=1}^{P}\sum_{i=1}^{S}\big\langle v(x_i) - f(x_i)\,,\, \partial_p u_{\theta(t)}(x_i)\big\rangle_{\mathbb{R}^m}\, \rho(\theta)^\dagger \boldsymbol{e}_p\right) \\ &= \frac{\mu_n(\Omega)}{S}\sum_{p=1}^{P}\sum_{i=1}^{S}\big\langle f(x_i) - v(x_i)\,,\, \partial_p u_{\theta(t)}(x_i)\big\rangle_{\mathbb{R}^m}\, \mathrm{d}u_{\theta(t)}\big(\rho(\theta)^\dagger \boldsymbol{e}_p\big) \\ &= \frac{\mu_n(\Omega)}{S}\sum_{i=1}^{S}\left\langle f(x_i) - v(x_i)\,,\, \sum_{1\leqslant p,q\leqslant P} \partial_p u_{\theta(t)}(x_i)\rho(\theta)^\dagger_{p,q}\partial_q u_{\theta(t)}\right\rangle_{\mathbb{R}^m}. \end{aligned} \tag{2.94}$$

We thus see that in this setting, the NTK is replaced by the *Natural Neural Tangent Kernel (NNTK)* (Rudner et al., 2019; Bai et al., 2022), defined at $\theta \in \mathbb{R}^P$, for all $x, y \in \Omega$, by

$$\mathrm{NNTK}_\theta(x,y) := \sum_{1\leqslant p,q\leqslant P} \partial_p u_\theta(x)\, \rho(\theta)^\dagger_{p,q}\, \partial_q u_\theta(y). \tag{2.95}$$

Note that this construction generalizes the NTK, which is recovered as the special case where ρ is the constant metric $\rho(\theta) = \mathrm{I}_{\mathbb{R}^P}$, which corresponds indeed to the canonical Euclidean metric.

As in Equation (2.84), the metric ρ on $\mathbb{R}^P$ implicitly endows $\operatorname{Im} u$ with the Riemannian structure given by the pullback of ρ through the differential $\mathrm{d}u_\theta$. Proposition 2.34 then shows that the expression in Equation (2.94) corresponds to the functional gradient of $\mathcal{L}_{2,X}$ for all $\theta \in \mathbb{R}^P$, namely:

$$\nabla_{\rho(\theta)} \mathcal{L}_{2,X} = \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \langle f(x_i) - v(x_i)\,,\, \mathrm{NNTK}_\theta(x_i, \cdot) \rangle_{\mathbb{R}^m}\,. \tag{2.96}$$

By comparing this expression with Equation (2.78), we see that they coincide formally under the identification

$$\delta_{x_i} = \mathrm{NNTK}_\theta(x_i, \cdot), \tag{2.97}$$

a correspondence that, again, will be made rigorous in Chapter 3.

Following the same reasoning, we find that when $m = 1$, for all $\theta \in \mathbb{R}^P$,

$$\nabla_{\rho(\theta)} \mathcal{L}_{2,X} \in \operatorname{Span}\left(\mathrm{NNTK}_\theta(x_i, \cdot) \,:\, 1 \leqslant i \leqslant S\right) \subset \operatorname{Im} \mathrm{d}u_\theta. \tag{2.98}$$

More generally, for $m > 1$, introducing for all $1 \leqslant i \leqslant S$ and $1 \leqslant j \leqslant m$,

$$\mathrm{NNTK}_\theta(x_i, \cdot)_j := \left(\mathrm{NNTK}_\theta(x_i, \cdot)\right)_{j,\cdot} = \sum_{1 \leqslant p,q \leqslant P} \left(\partial_p u_\theta(x_i)\right)_j \rho(\theta)^\dagger_{p,q} \partial_q u_\theta, \tag{2.99}$$

we have

$$\nabla_{\rho(\theta)} \mathcal{L}_{2,X} \in \operatorname{Span}\left(\mathrm{NNTK}_\theta(x_i, \cdot)_j \,:\, 1 \leqslant i \leqslant S,\ 1 \leqslant j \leqslant m\right) \subset \operatorname{Im} \mathrm{d}u_\theta. \tag{2.100}$$

We now turn to the approximation of the empirical theoretical gradient flows introduced above. In general, these flows cannot be solved analytically, and must therefore be approximated using discrete iterative schemes.

2.2.4 . Gradient flow approximation

In this section, we explain how the theoretical gradient flows introduced in Section 2.1.4 and Section 2.2.2 can be approximated for numerical implementation.

For simplicity, let us consider an abstract parametric loss $\ell : \mathbb{R}^P \to \mathbb{R}_+$ in the sense of Definition-Proposition 2.28, and recall that the corresponding classical gradient flow is defined in Definition 2.31 by

$$\begin{cases} \theta(0) & = \boldsymbol{\theta_0} \\ \dfrac{d}{dt}\,\theta(t) & = -\nabla \ell_{\theta(t)} \end{cases}. \tag{2.29}$$

***Remark* 2.47.** The same reasoning applies to the natural gradient case, where the Riemannian metric $\rho : \mathbb{R}^P \to \mathcal{M}_P^+(\mathbb{R})$ modifies the gradient according to Equation (2.46) (and its generalization therein). In this case, the

Euclidean gradient at θ is simply multiplied by $\rho(\theta)^\dagger$. However, computing $\rho(\theta)^\dagger$ has a cubic complexity $O(P^3)$, which becomes prohibitive for large-scale applications. Different approaches have been proposed to circumvent these limitations. The most prominent one is *K-FAC*, introduced by Heskes (2000) and further developed by Martens and Grosse (2015); Grosse and Martens (2016), which approximates the Gram matrix $\rho(\theta)$ by block-diagonal matrices. This approximation can be interpreted as making the ansatz that the partial derivatives of the weights belonging to different layers are mutually orthogonal. A refinement of this method was proposed by George et al. (2018), who carefully accounted for the eigenstructure of the block-diagonal matrices in order to obtain a more accurate approximation of the diagonal rescaling induced by the inversion of the Gram matrix. This approach was later extended to PINNs by Dangel et al. (2025). In a completely different vein, Ollivier (2017) proposed a statistical approach that was proven to converge to the natural gradient update in the zero–learning-rate limit.

One of the main goals of our first contribution, presented in Chapter 4, is precisely to provide an efficient workaround for this computational bottleneck.

Unless ℓ admits a particularly convenient analytical form, the ODE above generally has no closed-form solution. We must therefore approximate this equation.

A simple approach is to consider the first-order Taylor expansion of the function $\theta : \mathbb{R} \to \mathbb{R}^P$ at time $t \in [0, +\infty)$: for all $s > 0$,

$$\theta(t+s) = \theta(t) + \frac{d\theta}{dt}(t)\, s + \varepsilon_t(s)\, s,$$

where $\varepsilon_t : [0, +\infty) \to \mathbb{R}^P$ satisfies $\lim_{s \to 0} \|\varepsilon_t(s)\|_{\mathbb{R}^P} = 0$. Substituting Equation (2.29), we obtain for all $s > 0$,

$$\theta(t+s) = \theta(t) - \nabla \ell_{\theta(t)}\, s + \varepsilon_t(s)\, s. \tag{2.101}$$

For sufficiently small $s > 0$, we can approximate

$$\theta(t+s) \simeq \theta(t) - \nabla \ell_{\theta(t)}\, s,$$

neglecting higher-order terms $\varepsilon_t(s)\, s$. Iterating this procedure allows us to approximate any $\theta(t_1)$ for $t_1 > 0$ by constructing a discrete sequence: $\forall 0 \leqslant n \leqslant N_1 - 1$,

$$\tilde{\theta}_n^{t_1, N_1} := \begin{cases} \boldsymbol{\theta_0}, & \text{if } n = 0 \\ \tilde{\theta}_{n-1}^{t_1, N_1} - \nabla \ell_{\tilde{\theta}_{n-1}^{t_1, N_1}} \dfrac{t_1}{N_1}, & \text{if } 1 \leqslant n < N_1. \end{cases}$$

This simple discretization scheme, dating back to Euler, is now known as the *Euler method*, and forms the basis of the Runge–Kutta family of numerical ODE solvers (Butcher, 2016, Chapter 3).

Surprisingly,[7] this approximation works remarkably well in practice for training neural networks. Indeed, all major advances in deep learning, from early convolutional models for image recognition (LeCun et al., 1998) to modern architectures such as transformers (Vaswani et al., 2017), including the celebrated Generative Pre-trained Transformers (GPT)[8], rely on variants of this method, which we now formalize.

Definition 2.48 (Gradient descent (GD))**.** Let $\boldsymbol{\theta_0} \in \mathbb{R}^P$ be an initialization, and let $\ell : \mathbb{R}^P \to \mathbb{R}_+$ be a parametric loss in the sense of Definition-Proposition 2.28. Given a learning rate $\eta > 0$, the *gradient descent* algorithm (LeCun et al., 1998, Eq. 2) defines an iterative sequence $(\tilde{\theta}_n)_{n\in\mathbb{N}_0} \subset \mathbb{R}^P$ by

$$\tilde{\theta}_n := \begin{cases} \boldsymbol{\theta_0}, & \text{if } n = 0 \\ \tilde{\theta}_{n-1} - \eta\, \nabla\ell(\tilde{\theta}_{n-1}), & \text{if } n \geqslant 1. \end{cases} \tag{2.102}$$

***Remark* 2.49.** As already mentioned in seminal works such as (LeCun et al., 1998, Eq. 3), the method most commonly used in deep learning is the *Stochastic Gradient Descent (SGD)*. As discussed in Section 2.2.2, this variant incorporates a resampling strategy that modifies the empirical loss at each iteration of Equation (2.102). Hence, the modification lies in the definition of the loss ℓ, while the update rule itself remains unchanged once ℓ is specified.

The main difficulty in applying the algorithm of Definition 2.48 in practice, namely, ensuring its convergence to at least a local minimum of ℓ, lies in selecting an appropriate learning rate η. Choosing this parameter is often a crucial part of the experimental process when training neural networks. Nonetheless, two major adaptations of this algorithm have been proposed, both providing significant empirical performance improvements:

(S)GD with momentum. This variant was first introduced by Rumelhart et al. (1985), although similar ideas appear to have been developed independently by Nesterov 1983[9]. The key idea is to incorporate information from past updates of $(\tilde{\theta}_n)_{n\in\mathbb{N}_0}$, that is, to retain a trace of the sequence of gradients $\left(\nabla\ell(\tilde{\theta}_i)\right)_{i=0}^{n-1}$, so as to obtain a more stable and informed update direction for $\tilde{\theta}_{n-1}$.

This is achieved by fixing parameters $0 < \beta < 1$ and $\eta > 0$, and defining

[7] A complete theoretical understanding of this phenomenon remains an open problem; see Remark 2.42 for a brief survey of existing results.

[8] https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

[9] The paper is written in Russian.

an auxiliary sequence $(\overline{m}_n)_{n\in\mathbb{N}_0}$ as follows: for all $n \in \mathbb{N}_0$,

$$\tilde{\theta}_n := \begin{cases} \boldsymbol{\theta_0} & \text{if } n = 0 \\ \tilde{\theta}_{n-1} - \eta\, \overline{m}_{n-1} & \text{if } n \geqslant 1 \end{cases}, \tag{2.103a}$$

$$\overline{m}_n := \begin{cases} \nabla \ell(\boldsymbol{\theta_0}) & \text{if } n = 0 \\ \beta\, \overline{m}_{n-1} + (1-\beta)\, \nabla \ell(\tilde{\theta}_{n-1}) & \text{if } n \geqslant 1 \end{cases}. \tag{2.103b}$$

Qian (1999) showed that, in the limit $\eta \to 0$, the sequence $(\overline{m}_n)_{n\in\mathbb{N}_0}$ behaves analogously to the momentum of a Newtonian particle moving in a viscous medium, hence the term *momentum*. This interpretation connects the method to ideas from evolutionary strategies (Beyer, 2001; Emmerich et al., 2018), see also Hansen (2016, Section 2).

RMSProp. Proposed by Geoffrey Hinton[10], RMSProp addresses the fact that the components of the gradient $\nabla \ell(\tilde{\theta}_n)$ may have vastly different magnitudes and may vary significantly during training, often impeding convergence. The algorithm mitigates this issue by maintaining an exponentially weighted moving average of the squared gradients and scaling each coordinate of the update step accordingly.

More precisely, let $0 < \gamma < 1$, $\epsilon > 0$, and $\eta > 0$, and define a sequence $(\varsigma_n)_{n\in\mathbb{N}_0}$ by:

$$\tilde{\theta}_n := \begin{cases} \boldsymbol{\theta_0} & \text{if } n = 0 \\ \tilde{\theta}_{n-1} - \eta\, \operatorname{diag}\left(\frac{1}{\sqrt{\varsigma_{n-1}+\epsilon}}\right) \nabla \ell(\tilde{\theta}_{n-1}) & \text{if } n \geqslant 1 \end{cases}, \tag{2.104a}$$

$$\varsigma_n := \begin{cases} (\nabla \ell(\boldsymbol{\theta_0}))^2 & \text{if } n = 0 \\ \gamma\, \varsigma_{n-1} + (1-\gamma) \left(\nabla \ell(\tilde{\theta}_{n-1})\right)^2 & \text{if } n \geqslant 1 \end{cases}. \tag{2.104b}$$

where the square, square root, and inverse operations are applied componentwise, and $\operatorname{diag} : \mathbb{R}^P \to \mathcal{M}_P(\mathbb{R})$ maps a vector $\omega \in \mathbb{R}^P$ to the diagonal matrix $\operatorname{diag}(\omega)$ with diagonal entries $\operatorname{diag}(\omega)_{ii} = \omega_i$ for all $1 \leqslant i \leqslant P$. RMSProp, too, bears close connections with evolutionary strategy principles (Hansen, 2016, Section 3).

Both of these ideas were subsequently combined, along with minor refinements, into the celebrated **Adam**[11] algorithm introduced by Kingma and Ba (2014), which remains the most widely used optimization method in deep learning today. For completeness, we recall its formulation below.

[10] Although never formally published, this algorithm was first presented in a lecture at the University of Toronto. The slides are available at `www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

[11] An abbreviation for "adaptive moment estimation", though not a strict acronym.

Definition 2.50 (Adam)**.** Let $\boldsymbol{\theta_0} \in \mathbb{R}^P$, and let $\ell : \mathbb{R}^P \to \mathbb{R}$ be a differentiable loss function as in Definition 2.48. Given $\epsilon, \eta > 0$ and $0 < \gamma, \beta < 1$, the Adam algorithm defines three sequences $(\tilde{\theta}_n)_{n\in\mathbb{N}_0}$, $(\overline{m}_n)_{n\in\mathbb{N}_0}$, and $(\varsigma_n)_{n\in\mathbb{N}_0}$ as follows: for all $n \in \mathbb{N}_0$,

$$\tilde{\theta}_n := \begin{cases} \boldsymbol{\theta_0} & \text{if } n = 0 \\ \tilde{\theta}_{n-1} - \dfrac{\eta}{(1-\gamma^n)(1-\beta^n)} \operatorname{diag}\Big(\frac{1}{\sqrt{\varsigma_{n-1}}+\epsilon}\Big) \overline{m}_{n-1} & \text{if } n \geqslant 1 \end{cases}, \tag{2.105a}$$

$$\overline{m}_n := \begin{cases} \nabla\ell(\boldsymbol{\theta_0}) & \text{if } n = 0 \\ \beta\, \overline{m}_{n-1} + (1-\beta)\, \nabla\ell(\tilde{\theta}_{n-1}) & \text{if } n \geqslant 1 \end{cases}, \tag{2.105b}$$

$$\varsigma_n := \begin{cases} (\nabla\ell(\boldsymbol{\theta_0}))^2 & \text{if } n = 0 \\ \gamma\, \varsigma_{n-1} + (1-\gamma) \Big(\nabla\ell(\tilde{\theta}_{n-1})\Big)^2 & \text{if } n \geqslant 1 \end{cases}. \tag{2.105c}$$

As in RMSProp, the square, square root, and inverse are applied componentwise, and diag maps $\omega \in \mathbb{R}^P$ to the diagonal matrix whose (i,i)-th entry equals ω_i for $1 \leqslant i \leqslant P$.

Finally, it is worth noting that the hyperparameters η, β, and γ in the above definition can themselves be made adaptive, that is, replaced by sequences rather than fixed constants. This process, known as *scheduling*, allows the learning rate and momentum parameters to evolve dynamically during training. Many heuristics have been developed in this direction, including strategies that make (η_n), (β_n), and (γ_n) depend explicitly on $\tilde{\theta}_n$, $(\overline{m}_n)$, or (ς_n); see, for example, Li and Arora (2019); Kim et al. (2021); Lewkowycz (2021).

One may also wonder whether extending the Taylor expansion in Equation (2.101) to higher order terms could provide a more accurate approximation of the theoretical gradient flow. The answer is, of course, affirmative. However, the main limitation lies in the exponential growth of the number of parameters required to compute higher-order differentials.

Indeed, each k-th order differential of ℓ, for $k \geqslant 1$, evaluated at a point $\omega \in \mathbb{R}^P$, can be viewed as a k-linear map from $\mathbb{R}^P$ to $\mathbb{R}$, that is, an element of a space of dimension P^k. This quickly becomes intractable, since P itself is already very large, typically of the order of 10^3 for the simplest architectures, and up to 10^{10} or even 10^{11} for large-scale models such as transformers.

However, in cases where P remains moderate (typically $P \leqslant 10^4$), second-order information, namely, the Hessian, can still be approximated effectively. This is precisely the idea behind the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) algorithm (Fletcher, 2000, Section 3.4, p. 62), which is itself inspired by Newton's method (Fletcher, 2000, Section 3.1, p. 44). Liu and Nocedal (1989) later

proposed a modification, known as the *Limited-memory BFGS* (L-BFGS) algorithm, that makes the method scalable to high-dimensional problems and thus suitable for training neural networks. For completeness, we recall their formulations below.

Definition 2.51 (Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm)**.** Let $\ell : \mathbb{R}^P \to \mathbb{R}$ be a twice-differentiable loss function, and let $\boldsymbol{\theta_0} \in \mathbb{R}^P$ be an initialization. The BFGS algorithm approximates the inverse Hessian of ℓ iteratively through a sequence $(H_n)_{n\in\mathbb{N}_0}$ of symmetric positive-definite matrices, while updating the parameters $(\tilde{\theta}_n)_{n\in\mathbb{N}_0}$ according to:

$$\tilde{\theta}_n := \begin{cases} \boldsymbol{\theta_0}, & \text{if } n = 0, \\ \tilde{\theta}_{n-1} - \eta\, H_{n-1}\, \nabla\ell(\tilde{\theta}_{n-1}), & \text{if } n \geqslant 1, \end{cases} \tag{2.106a}$$

$$H_n := \begin{cases} \mathrm{I}_P, & \text{if } n = 0, \\ \Big(\mathrm{I}_P - \dfrac{s_{n-1}y_{n-1}^\top}{y_{n-1}^\top s_{n-1}}\Big) H_{n-1} \Big(\mathrm{I}_P - \dfrac{y_{n-1}s_{n-1}^\top}{y_{n-1}^\top s_{n-1}}\Big) + \dfrac{s_{n-1}s_{n-1}^\top}{y_{n-1}^\top s_{n-1}}, & \text{if } n \geqslant 1, \end{cases} \tag{2.106b}$$

where $s_{n-1} = \tilde{\theta}_n - \tilde{\theta}_{n-1}$ and $y_{n-1} = \nabla\ell(\tilde{\theta}_n) - \nabla\ell(\tilde{\theta}_{n-1})$. The parameter $\eta > 0$ denotes the learning rate, and I_P is the $P \times P$ identity matrix.

Definition 2.52 (Limited-memory BFGS (L-BFGS))**.** The *Limited-memory BFGS* (L-BFGS) algorithm is a memory-efficient variant of BFGS, specifically designed for large-scale optimization problems. Instead of storing the full inverse Hessian matrix $H_n \in \mathcal{M}_P(\mathbb{R})$, which requires $O(P^2)$ memory, L-BFGS maintains only the M most recent curvature pairs (s_i, y_i), where $s_i = \tilde{\theta}_{i+1} - \tilde{\theta}_i$ and $y_i = \nabla\ell(\tilde{\theta}_{i+1}) - \nabla\ell(\tilde{\theta}_i)$, for $i \in \{n - M, \ldots, n - 1\}$ and $M \ll P$ a fixed memory parameter.

At each iteration, the product $H_n\nabla\ell(\tilde{\theta}_{n-1})$ is computed implicitly through a *two-loop recursion*, without ever forming H_n explicitly, thereby yielding an update direction d_{n-1}. The full procedure is detailed in Algorithm 1.

Finally, the parameters are updated according to

$$\tilde{\theta}_n = \tilde{\theta}_{n-1} + \eta\, d_{n-1}, \tag{2.107}$$

where $\eta > 0$ denotes the learning rate.

This implicit computation of the inverse Hessian action reduces both memory and computational costs from $O(P^2)$ to $O(MP)$, making L-BFGS highly scalable for high-dimensional optimization problems, such as those arising in neural network training.

Algorithme 1 : Two-loop recursion for L-BFGS

Input : Current gradient $g_{n-1} = \nabla \ell(\tilde{\theta}_{n-1})$; curvature pairs (s_i, y_i) for $i = n-M, \ldots, n-1$; memory size M.

Output : Descent direction $d_{n-1} = -H_n g_{n-1}$.

Step 1: Backward pass. Initialize $q := g_{n-1}$.

for $i = n-1$ ***down to*** $n-M$ **do**

  $\alpha_i := \dfrac{s_i^\top q}{y_i^\top s_i}$;

  $q := q - \alpha_i \, y_i$;

end for

Step 2: Initial scaling. Set $H_0 := \dfrac{y_{n-1}^\top s_{n-1}}{y_{n-1}^\top y_{n-1}} \, \mathrm{I}_P$, and compute $r := H_0 q$.

Step 3: Forward pass.

for $i = n-M$ ***to*** $n-1$ **do**

  $\beta_i := \dfrac{y_i^\top r}{y_i^\top s_i}$;

  $r := r + s_i \, (\alpha_i - \beta_i)$;

end for

Step 4: Descent direction. Set $d_{n-1} := -r$.

We conclude this chapter with a few remarks on the choice of the initialization point $\boldsymbol{\theta_0} \in \mathbb{R}^P$. An extensive literature exists on this topic and its implications; see, for instance, Kumar (2017); Narkhede et al. (2022). In particular, the relationship between the statistical distribution chosen to initialize the coordinates of $\boldsymbol{\theta_0}$ and the asymptotic convergence of the neural network function $u_{\boldsymbol{\theta_0}}$ toward a Gaussian process as $P \to +\infty$ has been known since the seminal works of Neal (1996) and Williams (1996).

More recently, Jacot et al. (2018) demonstrated that, under commonly satisfied assumptions on the statistical distribution of the coordinates of $\boldsymbol{\theta_0}$, the training of a neural network via (stochastic) gradient descent (SGD) asymptotically converges, as $P \to +\infty$, to a kernel regression governed by the *Neural Tangent Kernel (NTK)* introduced in Section 2.2.2. Chizat et al. (2019) later extended these results to more general settings beyond neural networks.

In the context of this work, however, we shall adopt the so-called *Glorot-normal* and *Glorot-uniform* initializations (sometimes also referred to as *Xavier-normal* and *Xavier-uniform*, both in reference to Xavier Glorot), introduced by Glorot and Bengio (2010), and now considered as the standard approach.

Using the terminology of Definition 2.5, let us fix an architecture u built upon a mapping ψ, assumed to be a simple reparametrization. Each coordinate θ_p for $1 \leqslant p \leqslant P$ is then unambiguously mapped either to a weight in $\mathcal{M}_{l_{i-1},l_i}$ or to a bias in $\mathbb{R}^{l_i}$ for some $1 \leqslant i \leqslant d$. Then, for all $1 \leqslant p \leqslant P$, the initialization proceeds as follows:

Glorot-normal. Each $\boldsymbol{\theta_0}_p$ corresponding to a weight in $\mathcal{M}_{l_{i-1},l_i}$ is sampled from a normal distribution:

$$\boldsymbol{\theta_0}_p \sim \mathcal{N}\left(0, \frac{2}{l_{i-1}+l_i}\right).$$

If $\boldsymbol{\theta_0}_p$ corresponds to a bias in $\mathbb{R}^{l_i}$, it is initialized as:

$$\boldsymbol{\theta_0}_p = 0.$$

Glorot-uniform. Each $\boldsymbol{\theta_0}_p$ corresponding to a weight in $\mathcal{M}_{l_{i-1},l_i}$ is sampled from a uniform distribution:

$$\boldsymbol{\theta_0}_p \sim \mathcal{U}\left(-\sqrt{\frac{6}{l_{i-1}+l_i}}, \sqrt{\frac{6}{l_{i-1}+l_i}}\right).$$

As before, if $\boldsymbol{\theta_0}_p$ corresponds to a bias in $\mathbb{R}^{l_i}$, it is initialized as:

$$\boldsymbol{\theta_0}_p = 0.$$

These initializations are designed to preserve, as much as possible, the variance of activations and gradients across layers during the forward and backward passes, thereby mitigating vanishing or exploding gradient effects during training.

3 - Reproducing Kernel Hilbert Spaces (RKHS)

This chapter is largely inspired by Paulsen and Raghupathi (2016).

3.1 . Motivation

In this chapter, we lay the foundations of the theory of *Reproducing Kernel Hilbert Spaces* (RKHS). Historically, two complementary perspectives have shaped this theory: the *functional-analytic* viewpoint and the more recent *machine-learning* interpretation. Let us first describe the second perspective, which is more intuitive.

Imagine that we only know the values of a function f at a finite number of points of a set X, *i.e.* through a dataset. Intuitively, an RKHS endows us with a kernel k that quantifies the *influence* of each point $x \in X$ on all others. This influence captures several aspects at once: the regularity of the functions in the space, the diversity of behaviors they can express, and the correlations we expect between different points of X. In the case of neural networks, this notion directly echoes the *Neural Tangent Kernel* (NTK) introduced in Section 2.2.3 of Chapter 2, which will be central to our analysis in the following chapters.

Given the samples (x_i), the function f can then be expressed as a weighted combination:

$$f(x) = \sum_{i=1}^{n} f(x_i)\, k(x_i, x), \tag{3.1}$$

which can be interpreted as a weighted average of the known values $f(x_i)$, each weight $k(x_i, x)$ reflecting how strongly x_i influences the point x. If the points (x_i) were completely uncorrelated, this expression would yield an exact interpolant. In general, however, their correlations are encoded in the kernel itself, and the coefficients $f(x_i)$ must satisfy the linear system defined by the Gram matrix $K_F = \big(k(x_i, x_j)\big)_{1 \leqslant i,j \leqslant n}$.

This simple geometric idea, expressing interpolation as an inner product structure between function evaluations—lies at the very heart of RKHS theory and of many modern learning algorithms.

The second viewpoint, rooted in functional analysis, begins from an abstract Hilbert space of functions in which evaluation is a bounded linear functional. This leads to the classical characterization of RKHSs and naturally to the celebrated construction of Aronszajn and Moore, which shows how to build an RKHS directly from a positive definite kernel. This construction not only provides a concrete recipe but also unveils a deep correspondence between the geometry of functions and the algebra of kernels: a kernel can be regarded as an abstract inner product between evaluations.

In this chapter, we choose to begin from the second perspective in order to reach a less commonly emphasized, yet central, aspect of the theory, one that lies at the very core of our dissertation: *what happens when we modify the norm of an RKHS?* This question will enable us to build a bridge with the framework of *Hilbert riggings* developed in the previous chapter and to interpret RKHSs as natural tools for the implicit inversion of operators.

As an illustration, we will show that the classical results on interpolation and approximation, and, consequently, the machine-learning viewpoint, can be recovered as special cases of this implicit inversion framework, thereby providing a unified perspective that connects functional geometry, operator theory, and machine learning.

Throughout the chapter, X denotes a nonempty set.

3.2 . Fundamental setting

We begin with the general definition of a kernel (not necessarily reproducing) and the associated notions of an integral operator.

3.2.1 . Kernels, Integral Operators, and Green's Functions

Let us start by introducing the general notion of a kernel function.

Definition 3.1 (Kernel function)**.** Let X and Y be two non-empty sets. A *kernel* on $X \times Y$ is a function $k : X \times Y \to \mathbb{K}$, where $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$.

The main interest of a kernel lies in the construction of the associated *integral operator*.

Definition 3.2 (Integral operator)**.** Let $\big(\mathcal{H}_X, \langle \cdot\,, \cdot \rangle_{\mathcal{H}_X}\big)$ and $\big(\mathcal{H}_Y, \langle \cdot\,, \cdot \rangle_{\mathcal{H}_Y}\big)$ be two Hilbert spaces, respectively contained in the spaces of functions $\mathcal{F}(X \to \mathbb{R})$ and $\mathcal{F}(Y \to \mathbb{R})$. A linear operator $A : \mathcal{H}_Y \to \mathcal{H}_X$ is said to be an *integral operator* if there exists a kernel $k : X \times Y \to \mathbb{K}$, $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, such that for all $f \in \mathcal{H}_Y$ and all $x \in X$,

$$A(f)(x) = \langle k(x, \cdot)\,,\, f \rangle_{\mathcal{H}_Y}\,. \tag{3.2}$$

***Example* 3.3.** The most classical case occurs when the inner product is itself defined by an integral, *i.e.* when $\mathcal{H}_Y = \mathrm{L}^2(Y, \mu_Y)$ for some measure μ_Y on Y. In that case, we have

$$A(f)(x) = \int_Y k(x, y)\, f(y)\, \mu_Y(\mathrm{d}y), \tag{3.3}$$

hence the name *integral operator*.

One of the most important subclasses for this work is that of *Green's functions*. Let us fix a linear differential operator $D : \mathcal{H}_D \to \mathrm{L}^2(X \to \mathbb{R}, \mu)$, where $\mathcal{H}_D$ denotes the Hilbert space on which D is continuous (see Berezansky et al. 1996b, Chapter 12).

Definition 3.4 (Green's function)**.** A *Green's function* of D is any kernel $g : X \times X \to \mathbb{R}$ such that the operator

$$R : \left\{ \begin{array}{lll} D[\mathcal{H}_D] & \longrightarrow & \mathcal{H}_D \\ f & \longmapsto & \left(x \mapsto \int_X g(x,s)\, f(s)\, \mu(\mathrm{d}s)\right) \end{array} \right. \tag{3.4}$$

is a right-inverse of D, that is, $D \circ R = \mathcal{I}_{\mathcal{H}_D}$.

***Remark* 3.5.** Equivalently, g is a Green's function if, for all $x, s \in X$,

$$D[g(\cdot, s)](x) = \delta_x(s),$$

where δ_x denotes the Dirac distribution centered at x. This property will play a central role in connecting Green's functions with RKHS theory.

***Remark* 3.6.** From Definition 3.4, we immediately deduce that if $D[u] = f \in D[\mathcal{H}_D]$, then the solution is given by

$$u(x) = \int_X g(x,s)\, f(s)\, \mu(\mathrm{d}s).$$

To obtain more general results, we introduce the following extension.

Definition 3.7 (Solution in the least-squares sense)**.** Let $\mathcal{H}_0 \subset \mathcal{H}_D$ be a subspace isometrically embedded in $\mathcal{H}_D$, and $f \in \mathrm{L}^2(X \to \mathbb{R}, \mu)$. We call $u_0 \in \mathcal{H}_0$ a *solution of* $D[u] = f$ *in the least-squares sense* if

$$\|D[u_0] - f\|^2_{\mathrm{L}^2(X \to \mathbb{R}, \mu)} = \inf_{u \in \mathcal{H}_0} \|D[u] - f\|^2_{\mathrm{L}^2(X \to \mathbb{R}, \mu)}\,. \tag{3.5}$$

***Remark* 3.8.** If $f \in D[\mathcal{H}_0]$, then $\inf_{u \in \mathcal{H}_0} \|D[u] - f\|^2_{\mathrm{L}^2(X \to \mathbb{R}, \mu)} = 0$, and u_0 is therefore a classical solution.

Definition 3.9 (Generalized Green's function)**.** Let D, $\mathcal{H}_0$, and f be as in Definition 3.7. A *generalized Green's function* of D on $\mathcal{H}_0$ is any kernel $g : X \times X \to \mathbb{R}$ such that the operator

$$R_{\mathcal{H}_0} : \left\{ \begin{array}{lll} \mathrm{L}^2(X \to \mathbb{R}, \mu) & \longrightarrow & \mathcal{H}_D \\ f & \longmapsto & \left(x \mapsto \int_X g(x,s)\, f(s)\, \mu(\mathrm{d}s)\right) \end{array} \right.$$

satisfies

$$D \circ R_{\mathcal{H}_0} = \Pi_{D[\mathcal{H}_0]}, \tag{3.6}$$

where $\Pi_{D[\mathcal{H}_0]}$ denotes the orthogonal projection onto $D[\mathcal{H}_0]$.

***Remark* 3.10.** Since $\Pi_{D[\mathcal{H}_0]_{|D[\mathcal{H}_0]}} = \mathcal{I}_{\mathcal{H}_0}$, Definition 3.9 indeed generalizes Definition 3.4.

We now turn to the definition of the *reproducing property*, which constitutes the cornerstone of RKHS theory. We restrict our attention to the real case, as the complex case is not required for our purposes and only complicates the notation without adding any conceptual insight.

3.2.2 . Reproducing property

We start our journey from an abstract Hilbert space $\big(\mathcal{H}, \langle \cdot\,,\, \cdot \rangle_{\mathcal{H}}\big)$ and define the notion of reproducing property on a set X. Let us state the definition.

Definition 3.11 (Reproducing property)**.** A Hilbert space $\mathcal{H}$ has the *reproducing property* on the set X if

(i) $\mathcal{H}$ is a subspace of the space $\mathcal{F}(X \to \mathbb{R})$ of functions from X to $\mathbb{R}$.

(ii) For every $x \in X$, the point evaluation functional

$$\delta_x : \left\{ \begin{array}{ccc} \mathcal{H} & \to & \mathbb{R} \\ f & \mapsto & f(x) \end{array} \right. \tag{3.7}$$

is a continuous linear form on $\mathcal{H}$, *i.e.* $\delta_x \in \mathcal{H}'$.

***Remark* 3.12.** If $\mathcal{H}$ has the reproducing property, then for all $x \in X$ and $f \in \mathcal{H}$,

$$|f(x)| \leqslant C_x \, \|f\|_{\mathcal{H}} \; ; \qquad C_x := \|\delta_x\|_{\mathcal{H}'} = \sup_{\substack{f \in \mathcal{H} \\ f \neq 0}} \frac{|\delta_x(f)|}{\|f\|_{\mathcal{H}}}. \tag{3.8}$$

Definition-Proposition 3.13 (Reproducing kernel)**.** *If $\mathcal{H}$ has the reproducing property, then there exists a function $k : X \times X \to \mathbb{R}$ such that*

(i) for all $x \in X$, $k(x, \cdot) \in \mathcal{H}$;

(ii) for all $f \in \mathcal{H}$, $\langle k(x, \cdot)\,,\, f \rangle_{\mathcal{H}} = f(x)$.

The function k is called the **reproducing kernel** *of $\mathcal{H}$, and for each $x \in X$, the element $k_x := k(x, \cdot) \in \mathcal{H}$ is called the* **reproducer** *of x in $\mathcal{H}$.*

Proof. By the Riesz representation theorem: for each $x \in X$, the functional δ_x has a representing element $k_x \in \mathcal{H}$ such that

$$\langle k_x \,,\, f \rangle_{\mathcal{H}} = \delta_x(f) = f(x), \qquad \forall f \in \mathcal{H}. \tag{3.9}$$

We then set, for all $x, y \in X$,

$$k(x, y) := k_x(y). \tag{3.10}$$

□

When $\mathcal{H}$ has the reproducing property, we shall say that $\mathcal{H}$ has the reproducing property *with kernel* k. Later, once we have established that k completely determines $\mathcal{H}$ both algebraically and topologically, we will refer to such spaces as ***Reproducing Kernel Hilbert Spaces*** (RKHS).

Lemma 3.14. *For all* $x, y \in X$,

$$k(x,y) = \langle k_y \,,\, k_x \rangle_{\mathcal{H}} \,. \tag{3.11}$$

Proof. Applying the reproducing property to k_x, for all $x, y \in X$,

$$k(x,y) \overset{(3.10)}{=} k_x(y) = \delta_y(k_x) \overset{(3.9)}{=} \langle k_y \,,\, k_x \rangle_{\mathcal{H}} \,. \tag{3.12}$$

□

Corollary 3.15. *(i) For all* $x, y \in X$,

$$k(x,y) = k(y,x). \tag{3.13}$$

(ii) For all $x \in X$,

$$k(x,x) = \|k(x,\cdot)\|_{\mathcal{H}}^2 \geqslant 0. \tag{3.14}$$

(iii) For all $x \in X$,

$$C_x \overset{(3.8)}{=} \|\delta_x\|_{\mathcal{H}'} = \|k(x,\cdot)\|_{\mathcal{H}} \,. \tag{3.15}$$

(iv) For all $x \in X$,

$$k(x,x) = 0 \iff \forall f \in \mathcal{H},\, f(x) = 0. \tag{3.16}$$

Proof. (i) Symmetry follows from the symmetry of $\langle \cdot \,,\, \cdot \rangle_{\mathcal{H}}$.

(ii) For all $x \in X$,

$$k(x,x) \overset{(3.11)}{=} \langle k_x \,,\, k_x \rangle_{\mathcal{H}} = \|k_x\|_{\mathcal{H}}^2 = \|k(x,\cdot)\|_{\mathcal{H}}^2 \geqslant 0. \tag{3.17}$$

(iii) By the isometric identification $\mathcal{H}' \simeq \mathcal{H}$ from the Riesz theorem, we have $\|\delta_x\|_{\mathcal{H}'} = \|k(x,\cdot)\|_{\mathcal{H}}$. Equivalently, one may argue directly by the Cauchy–Schwarz–Buniakovski inequality:

$$\sup_{\substack{f \in \mathcal{H} \\ f \neq 0}} \frac{\langle k(x,\cdot) \,,\, f \rangle_{\mathcal{H}}}{\|f\|_{\mathcal{H}}} \leqslant \sup_{\substack{f \in \mathcal{H} \\ f \neq 0}} \frac{\|k(x,\cdot)\|_{\mathcal{H}} \, \|f\|_{\mathcal{H}}}{\|f\|_{\mathcal{H}}} = \|k(x,\cdot)\|_{\mathcal{H}} \,, \tag{3.18}$$

with equality when $f = k(x,\cdot)$.

(iv) If $k(x,x) = 0$, then $\|k(x,\cdot)\|_{\mathcal{H}} = 0$, hence $f(x) = \langle k(x,\cdot) \,,\, f \rangle_{\mathcal{H}} = 0$ for all $f \in \mathcal{H}$. Conversely, if all $f \in \mathcal{H}$ vanish at x, then in particular $k(x,\cdot)$ does, so $k(x,x) = 0$.

□

We add another corollary that is often omitted, but which connects to the classical notion of *Christoffel function*, widely used in least-squares regression (*cf.* Cohen and Migliorati (2017)).

Corollary 3.16 (Christoffel function)**.** *For all $x \in X$ such that $k(x,x) > 0$,*

$$\frac{1}{k(x,x)} = \min_{\substack{f \in \mathcal{H} \\ f(x)=1}} \|f\|_{\mathcal{H}}^2 , \tag{3.19}$$

$$\max_{\substack{f \in \mathcal{H} \\ f(x)=1}} \frac{1}{\|f\|_{\mathcal{H}}^2} = k(x,x). \tag{3.20}$$

Proof. By the fourth point of Corollary 3.15,

$$k(x,x) \neq 0 \quad \Longrightarrow \quad \exists f \in \mathcal{H} \text{ such that } f(x) \neq 0. \tag{3.21}$$

Since $\mathcal{H}$ is a vector space, we may define

$$g := \frac{f}{f(x)} \in \mathcal{H}, \tag{3.22}$$

which satisfies $g(x) = 1$. Therefore,

$$\{f \in \mathcal{H} : \ f(x) = 1\} \neq \varnothing. \tag{3.23}$$

Let us now fix $f \in \mathcal{H}$ such that $f(x) = 1$. By the Cauchy–Schwarz–Buniakovski inequality, we have

$$1 = f(x) = \langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}} \leqslant \|k(x,\cdot)\|_{\mathcal{H}}\, \|f\|_{\mathcal{H}}\,. \tag{3.24}$$

Using $\|k(x,\cdot)\|_{\mathcal{H}}^2 = k(x,x)$, this yields

$$\frac{1}{k(x,x)} \leqslant \|f\|_{\mathcal{H}}^2 , \tag{3.25}$$

$$\frac{1}{\|f\|_{\mathcal{H}}^2} \leqslant k(x,x). \tag{3.26}$$

Consequently,

$$\frac{1}{k(x,x)} \leqslant \inf_{\substack{f \in \mathcal{H} \\ f(x)=1}} \|f\|_{\mathcal{H}}^2 , \tag{3.27}$$

$$\sup_{\substack{f \in \mathcal{H} \\ f(x)=1}} \frac{1}{\|f\|_{\mathcal{H}}^2} \leqslant k(x,x). \tag{3.28}$$

The conclusion follows by taking the particular choice

$$f := \frac{k(x,\cdot)}{k(x,x)}. \tag{3.29}$$

□

Before reviewing some fundamental examples, we give some topological consequences of the reproducing property on $\mathcal{H}$

Proposition 3.17. *Suppose that $\mathcal{H}$ has the reproducing property on X and let $f \in \mathcal{H}$ and $f_n \in \mathcal{H}^{\mathbb{N}}$ such that $\lim_{n\to\infty} \|f_n - f\|_{\mathcal{H}} = 0$. Then for all $x \in X$*

$$\lim_{n\to\infty} |f_n(x) - f(x)| = 0, \tag{3.30}$$

i.e. the convergence also happens pointwise.

Proof. This is a simple application of the Cauchy–Schwarz–Buniakovski inequality, namely for all $x \in X$

$$|f_n(x) - f(x)| = |\langle f_n - f\,,\, k_x\rangle_{\mathcal{H}}| \leqslant \|f_n - f\|_{\mathcal{H}}\, \|k_x\|_{\mathcal{H}}\,. \tag{3.31}$$

□

Proposition 3.18. *The identity operator $\mathcal{I}_{\mathcal{H}}$ on $\mathcal{H}$ is an integral operator whose kernel is precisely the reproducing kernel $k : X \times X \to \mathbb{R}$.*

Proof. This follows directly from the definition. Indeed, for all $x \in X$ and $f \in \mathcal{H}$, we have

$$\mathcal{I}_{\mathcal{H}}(f)(x) = f(x) \overset{(3.9)}{=} \langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}}\,, \tag{3.32}$$

which is exactly the form required by Definition 3.2. □

Proposition 3.19. *Suppose that $\mathcal{H}$ is a finite-dimensional subspace of $\mathcal{F}(X \to \mathbb{R})$. Then $\mathcal{H}$ possesses the reproducing property.*

Proof. Since $\mathcal{H}$ is finite-dimensional, it admits a finite basis $(v_i) \in \mathcal{H}^d$, for some $d \in \mathbb{N}$. Without loss of generality, we may assume that (v_i) is orthonormal with respect to $\langle\cdot\,,\,\cdot\rangle_{\mathcal{H}}$ after applying a Gram–Schmidt procedure. Define, for all $x, y \in X$,

$$k(x,y) = \sum_{i=1}^{d} v_i(x) v_i(y). \tag{3.33}$$

Then, for all $x \in X$,

$$k(x,\cdot) = \sum_{i=1}^{d} v_i(x) v_i \in \mathcal{H}. \tag{3.34}$$

Moreover, by Parseval's identity, for every $f \in \mathcal{H}$ we have

$$f = \sum_{i=1}^{d} v_i \langle v_i\,,\, f\rangle_{\mathcal{H}}\,. \tag{3.35}$$

Hence, for all $x \in X$,

$$f(x) = \sum_{i=1}^{d} v_i(x)\langle v_i\,,\, f\rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{d} v_i(x) v_i\,,\, f \right\rangle_{\mathcal{H}} = \langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}}\,. \tag{3.36}$$

This shows that k is a reproducing kernel for $\mathcal{H}$. Finally, Equation (3.15) in Corollary 3.15 ensures that the evaluation functional δ_x is continuous for every $x \in X$, and thus that $\mathcal{H}$ indeed has the reproducing property. □

***Remark* 3.20.** We will show later that whenever a space is a RKHS, its kernel can always be expressed as a sum over a Hilbertian basis. However, in the finite-dimensional case considered here, we have implicitly relied on the fact that pointwise convergence is automatically guaranteed for a finite sum—something that no longer holds for an infinite one (consider, for instance, the Dirichlet kernel).

3.2.3 . Fondamental examples

***Example* 3.21** (The finite-dimensional case $\mathbb{R}^n$)**.** Let us first consider $\mathbb{R}^n$ endowed with its canonical inner product

$$\langle u\,,\, v\rangle_{\mathbb{R}^n} = \sum_{i=1}^{n} u_i v_i. \tag{3.37}$$

Define the set $X = \{1, \dots, n\}$. Each vector $f = (f(1), \dots, f(n)) \in \mathbb{R}^n$ can be interpreted as a function $f : X \to \mathbb{R}$ with $f(i)$ its i-th coordinate. For $i \in X$, the evaluation functional is therefore

$$\delta_i(f) = f(i). \tag{3.38}$$

Clearly δ_i is continuous, since

$$|f(i)| \leqslant \|f\|_{\mathbb{R}^n}\,. \tag{3.39}$$

By the Riesz representation theorem, there exists $k_i \in \mathbb{R}^n$ such that for all $f \in \mathbb{R}^n$

$$\langle k_i\,,\, f\rangle_{\mathbb{R}^n} = \delta_i(f) = f(i). \tag{3.40}$$

This means that $k_i = e_i$, where $(e_i)_{1\leqslant i\leqslant n}$ denotes the canonical basis of $\mathbb{R}^n$. Hence the reproducing kernel is

$$k(i,j) = \delta_{ij} = \mathcal{I}_{\mathbb{R}^n} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \tag{3.41}$$

i.e. the classical identity matrix.

More generally, let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive-definite matrix and endow $\mathbb{R}^n$ with the associated scalar product

$$\langle u\,,\, v\rangle_A = u^t A v = \sum_{1\leqslant i,j\leqslant n} u_i A_{i,j} v_j. \tag{3.42}$$

By the same reasoning, the associated reproducing kernel is then the matrix A^{-1}.

We will later see that, in a certain sense, having the reproducing property means that the linear algebra of the space behaves as in finite dimensions.

***Example* 3.22** (Sobolev space $\mathrm{H}^1_0(0,1)$)**.** Consider the Sobolev space $\mathrm{H}^1_0(0,1)$, introduced in Berezansky et al. (1996b, Chapter 12) as the completion of $\mathcal{C}^\infty_c(0,1)$ with respect to the norm

$$\|u\|^2_{\mathrm{H}^1_0(0,1)} = \int_0^1 (u')^2. \tag{3.43}$$

Every function $u \in \mathrm{H}^1_0(0,1)$ admits a continuous representative on $[0,1]$, with $u(0) = u(1) = 0$. For $x \in (0,1)$ and $u \in \mathrm{H}^1_0(0,1)$, we can write

$$u(x) - u(0) = \int_0^x u' = \int_0^1 \mathbb{1}_{[0,x]}\, u' = \big\langle \mathbb{1}_{[0,x]}\,,\, u' \big\rangle_{\mathrm{L}^2(0,1)}\,. \tag{3.44}$$

By Cauchy–Schwarz–Buniakovski inequality in $\mathrm{L}^2(0,1)$, this yields

$$|u(x)| \leqslant \big\|\mathbb{1}_{[0,x]}\big\|_{\mathrm{L}^2(0,1)}\, \big\|u'\big\|_{\mathrm{L}^2(0,1)} = \sqrt{x}\, \|u\|_{\mathrm{H}^1_0(0,1)}\,, \tag{3.45}$$

so point evaluation is continuous, and hence $\mathrm{H}^1_0(0,1)$ has the reproducing property.

The natural question that now arises is: what is the explicit form of the reproducing kernel associated with $\mathrm{H}^1_0(0,1)$? In general, identifying the reproducing kernel of a given Hilbert space is far from trivial, and the answer is often deeper than one might first expect. We shall follow the approach of Paulsen and Raghupathi (2016, Section 1.3.1), while deliberately rephrasing it in order to make the connection with the more general perspective introduced by Schwartz (1964) more transparent. We will return to this viewpoint in the conclusion. Let us consider the (continuous) embeddings

$$\mathrm{O}_{+\infty} : \mathcal{D}(0,1) \longrightarrow \mathrm{H}^1_0(0,1), \qquad \mathrm{O}_{-\infty} : \mathrm{H}^1_0(0,1) \longrightarrow \mathcal{D}'(0,1), \tag{3.46}$$

where $\mathcal{D}(0,1) = \mathcal{C}^\infty_c(0,1)$ denotes the space of test functions and $\mathcal{D}'(0,1)$ its dual, the space of distributions.

Since $\big(\mathcal{D}(0,1), \mathcal{D}'(0,1), \langle\cdot\,,\,\cdot\rangle_{\mathrm{L}^2(0,1)}\big)$ forms a duality,[1] these embeddings satisfy the fundamental property: for all $f \in \mathrm{H}^1_0(0,1)$ and $u \in \mathcal{D}(0,1)$,

$$\big\langle f\,,\, \mathrm{O}_{+\infty} u \big\rangle_{\mathrm{H}^1_0(0,1)} = \big\langle \mathrm{O}_{-\infty} f\,,\, u \big\rangle_{\mathrm{L}^2(0,1)}\,. \tag{3.48}$$

[1] We deliberately remain somewhat vague here, so as not to enter into the technical details required to define adjoints in the setting of locally convex topologies. For intuition, one may simply view this as the limiting case (in the sense of the extension of the rigging chain to its endpoints) of the identity valid for $u \in \mathrm{H}_+$ and $f \in \mathrm{H}_0$:

$$\big\langle f\,,\, \mathrm{O}_+ u \big\rangle_{\mathrm{H}_0} = \big\langle \mathrm{O}_- f\,,\, u \big\rangle_{\mathrm{H}_0} = \big\langle \mathbb{I}_0(\mathrm{O}_- f)\,,\, u \big\rangle_{\mathrm{H}_+} = \big\langle \mathrm{I}_+ f\,,\, u \big\rangle_{\mathrm{H}_+}\,. \tag{3.47}$$

In particular, for all $u, v \in \mathcal{D}(0,1)$,

$$\langle \mathrm{O}_{+\infty} v \,,\, \mathrm{O}_{+\infty} u \rangle_{\mathrm{H}^1_0(0,1)} = \big\langle \mathrm{O}_{-\infty}\big(\mathrm{O}_{+\infty} v\big) \,,\, u \big\rangle_{\mathrm{L}^2(0,1)} . \tag{3.49}$$

The operator

$$\mathcal{K} := \mathrm{O}_{-\infty}\mathrm{O}_{+\infty} : \mathcal{D}(0,1) \to \mathcal{D}'(0,1) \tag{3.50}$$

is called the ***Schwartz kernel*** of $\mathrm{H}^1_0(0,1)$. Explicitly,

$$\mathcal{K} : \left\{ \begin{array}{ccc} \mathcal{D}(0,1) & \to & \mathcal{D}'(0,1) \\ u & \mapsto & \Big(v \in \mathcal{D}(0,1) \mapsto \int_0^1 u'\, v' \Big) \end{array} \right. \tag{3.51}$$

By integration by parts, one finds that $\mathcal{K}(u)$ is the distribution $T_{-u''}$ associated with $-u''$, where we crucially use the fact that $\mathcal{D}(0,1)$ is canonically embedded into $\mathcal{D}'(0,1)$.

Let now $k_x \in \mathrm{H}^1_0(0,1)$ be the representer of x in $\mathrm{H}^1_0(0,1)$. Its reproducing property rewrites, for all $u \in \mathcal{D}(0,1)$, as

$$\langle k_x \,,\, \mathrm{O}_{+\infty}(u) \rangle_{\mathrm{H}^1_0(0,1)} = \int_0^1 \delta_x \, u = \int_0^1 \mathrm{O}_{-\infty}(k_x)\, u. \tag{3.52}$$

Since $\mathcal{D}(0,1)$ (more precisely $\operatorname{Im} \mathrm{O}_{+\infty}$) is dense in $\mathrm{H}^1_0(0,1)$, we can find $k_x^n \in \mathcal{D}(0,1)$ such that

$$\lim_{n\to\infty} \| \mathrm{O}_{+\infty}(k_x^n) - k_x \|_{\mathrm{H}^1_0(0,1)} = 0. \tag{3.53}$$

As $\mathrm{O}_{-\infty}$ is continuous, this implies, in the sense of distributions,

$$\mathrm{O}_{-\infty}(k_x) = \lim_{n\to\infty} \mathrm{O}_{-\infty}\big(\mathrm{O}_{+\infty} k_x^n\big) = \lim_{n\to\infty} \mathcal{K}(k_x^n) = \lim_{n\to\infty} T_{-(k_x^n)''} = \delta_x. \tag{3.54}$$

Since one may "exchange limits and derivatives" in the distributional sense, integrating twice gives

$$\lim_{n\to\infty} T_{-(k_x^n)'} = T_{\mathbb{1}_{[x,1]} + c_1}, \qquad \lim_{n\to\infty} T_{-(k_x^n)} = T_{\mathbb{1}_{[x,1]}(\cdot - x) + c_1 \cdot + c_2}, \tag{3.55}$$

with $c_1, c_2 \in \mathbb{R}$ some constants. Because convergence in $\mathrm{H}^1_0(0,1)$ implies pointwise convergence by Proposition 3.17, we deduce

$$T_{-k_x} = T_{\mathbb{1}_{[x,1]}(\cdot - x) + c_1 \cdot + c_2}, \tag{3.56}$$

and thus, by injectivity of $T : \mathrm{L}^1_{\mathrm{loc}}(0,1) \to \mathcal{D}'(0,1)$ (see for instance Berezansky et al. 1996a, Chapter 11, Section 1.4, Example 1.2),

$$-k_x = \mathbb{1}_{[x,1]}(\cdot - x) + c_1 \cdot + c_2, \tag{3.57}$$

which rewrites, for all $y \in [0,1]$ (up to renaming constants),

$$k_x(y) = \mathbb{1}_{[x,1]}(y)(x - y) + c_1\, y + c_2. \tag{3.58}$$

Imposing the boundary conditions gives

$$0 = k_x(0) = c_2, \qquad 0 = k_x(1) = (x-1) + c_1, \tag{3.59}$$

so that

$$\begin{aligned} k_x(y) &= \mathbb{1}_{[x,1]}(y)(x-y) + (1-x)\,y \\ &= \big(\mathbb{1}_{[0,1]}(y) - \mathbb{1}_{[x,1]}(y)\big)y + x\mathbb{1}_{[x,1]}(y) - xy \\ &= \mathbb{1}_{[0,x]}(y)\,y + x\mathbb{1}_{[x,1]}(y) - xy \\ &= \min(x,y) - xy. \end{aligned} \tag{3.60}$$

Thus, the Schwartz kernel $\mathcal{K}$, which is entirely determined by the inner product of $\mathrm{H}^1_0(0,1)$, also characterizes the reproducing kernel k_x. Conversely, as we will soon see, the reproducing kernel k_x determines both the space $\mathrm{H}^1_0(0,1)$ and its inner product, and therefore $\mathcal{K}$ itself. This reciprocity lies at the heart of Schwartz's generalization Schwartz (1964), which we briefly present in Section 3.6.

***Example* 3.23** ($\mathrm{L}^2(0,1)$ is not an RKHS)**.** Finally, consider by contrast $\mathrm{L}^2(0,1)$. Since a function $f \in \mathrm{L}^2(0,1)$ is defined only up to almost everywhere equivalence, point evaluation is not even well defined. More precisely, this means that the point evaluation functional

$$\delta_x : \begin{cases} \mathcal{F}\big((0,1) \to \mathbb{R}\big) & \to & \mathbb{R} \\ f & \mapsto & f(x) \end{cases}, \tag{3.61}$$

where $\mathcal{F}\big((0,1) \to \mathbb{R}\big)$ denotes the set of functions from $(0,1)$ to $\mathbb{R}$ endowed with the topology of pointwise convergence, does not extend continuously to $\mathrm{L}^2(0,1)$.

To see this explicitly, fix $x \in (0,1)$ and define a sequence of functions $(f_n) \in \mathrm{L}^2(0,1)^{\mathbb{N}}$ by

$$f_n(t) = \begin{cases} \big(\frac{t}{x}\big)^n, & 0 \leqslant t \leqslant x, \\ \big(\frac{1-t}{1-x}\big)^n, & x < t \leqslant 1. \end{cases} \tag{3.62}$$

Then $f_n(x) = 1$ for all $n \geqslant 1$, but a direct computation shows that

$$\lim_{n\to\infty} \|f_n\|_{\mathrm{L}^2(0,1)} = 0. \tag{3.63}$$

Hence,

$$|\delta_x(f_n)| = |f_n(x)| = 1, \qquad \|f_n\|_{\mathrm{L}^2(0,1)} \longrightarrow 0, \tag{3.64}$$

which violates continuity of δ_x.

As mentioned in Example 3.22, the kernel k in fact characterizes the Hilbert space $\mathcal{H}$ entirely. We shall develop this point in the next section.

3.3 . Characterization of $\mathcal{H}$ by its reproducing kernel

We begin by showing that, in any Hilbert space of functions with the reproducing property, the reproducers $k(x,\cdot)$ already generate a dense subspace. This is the starting point for the characterization of $\mathcal{H}$ by its kernel.

3.3.1 . Uniqueness and Representations of the Reproducing Kernel

Proposition 3.24 (Span of reproducers is dense)**.** *If $\mathcal{H}$ has the reproducing property on X with kernel k, then* $\operatorname{Span}\big(k(x,\cdot) : x \in X\big)$ *is dense in $\mathcal{H}$.*

Proof. Let $f \in \operatorname{Span}\big(k(x,\cdot) : x \in X\big)^{\perp}$. Then for every $x \in X$,

$$\langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}} = f(x) = 0. \tag{3.65}$$

Hence $f = 0$, which proves that $\operatorname{Span}\big(k(x,\cdot) : x \in X\big)^{\perp} = \{0\}$, and therefore the Span is dense in $\mathcal{H}$. □

We now take one step further and establish that the kernel k in fact completely characterizes the space: both *algebraically*, in the sense of determining the set of functions, and *topologically*, in the sense of fixing the norm.

Proposition 3.25 (k characterizes $\mathcal{H}$)**.** *Let $\mathcal{H}_1, \mathcal{H}_2$ be two Hilbert spaces having the reproducing property on X, with respective kernels k_1, k_2. If for all $x, y \in X$*

$$k_1(x,y) = k_2(x,y), \tag{3.66}$$

then:

(i) $\mathcal{H}_1 = \mathcal{H}_2$ ***(algebraic equality)***.

(ii) $\|\cdot\|_{\mathcal{H}_1} = \|\cdot\|_{\mathcal{H}_2}$ ***(topological equality)***.

Proof. Let $k = k_1 = k_2$ and define

$$\mathcal{W}_l := \Big(\operatorname{Span}\big(\,k(x,\cdot) : x \in X\,\big), \|\cdot\|_{\mathcal{H}_l}\Big) \subseteq \Big(\mathcal{H}_l, \|\cdot\|_{\mathcal{H}_l}\Big), \qquad l \in \{1,2\}. \tag{3.67}$$

For each $l \in \{1,2\}$ and $f \in \mathcal{W}_l$, there exist $m \in \mathbb{N}$, points $x_1, \dots, x_m \in X$ and coefficients $\alpha_1, \dots, \alpha_m \in \mathbb{R}$ such that, for all $x \in X$,

$$f(x) = \sum_{i=1}^{m} \alpha_i\, k_l(x_i, x) = \sum_{i=1}^{m} \alpha_i\, k(x_i, x). \tag{3.68}$$

Hence the values of f depend only on k, and not on the ambient Hilbert space, so that $\mathcal{W}_1 = \mathcal{W}_2$ *algebraically.*

Moreover, the norms also agree: for such f,

$$\|f\|^2_{\mathcal{H}_1} = \sum_{i,j=1}^{m} \alpha_i\alpha_j \,\langle k(x_i,\cdot)\,,\, k(x_j,\cdot)\rangle_{\mathcal{H}_1} = \sum_{i,j=1}^{m} \alpha_i\alpha_j\, k(x_i,x_j) = \|f\|^2_{\mathcal{H}_2}\,. \tag{3.69}$$

Thus

$$\big(\mathcal{W}_1, \|\cdot\|_{\mathcal{H}_1}\big) = \big(\mathcal{W}_2, \|\cdot\|_{\mathcal{H}_2}\big), \tag{3.70}$$

i.e. equality holds both algebraically and topologically.

Now let $f \in \mathcal{H}_1$. Since $\mathcal{W}_l$ is dense in $\mathcal{H}_l$ by Proposition 3.24, there exists a sequence $(f_n) \in \mathcal{W}_1^{\mathbb{N}}$ with $\|f - f_n\|_{\mathcal{H}_1} \to 0$. Because norms coincide on $\mathcal{W}_1 = \mathcal{W}_2$, the sequence (f_n) is also Cauchy in $\mathcal{H}_2$ complete, and hence converges to some $g \in \mathcal{H}_2$.

By Proposition 3.17, convergence in $\mathcal{H}_l$ implies pointwise convergence. Thus, for all $x \in X$,

$$g(x) = \lim_{n\to\infty} f_n(x) = f(x). \tag{3.71}$$

Therefore $f = g \in \mathcal{H}_2$. By symmetry, the same argument shows every that $g \in \mathcal{H}_2$ belongs to $\mathcal{H}_1$.

We conclude that

$$\big(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\big) = \big(\mathcal{H}_2, \|\cdot\|_{\mathcal{H}_2}\big), \tag{3.72}$$

both as sets and as Hilbert spaces. □

***Remark* 3.26.** Proposition 3.25 shows in particular that the reproducing kernel k is unique.

Warning: Note that Remark 3.26 does ***not*** mean that there cannot exist another function $k' : X \times X \to \mathbb{R}$ such that for all $f \in \mathcal{H}$ and $x \in X$

$$f(x) = \big\langle k'(x,\cdot)\,,\, f\big\rangle_{\mathcal{H}}\,. \tag{3.73}$$

For instance, if $\mathcal{H}_0$ is a subspace of $\mathcal{H}$ having the reproducing property on X with kernel k, then each $k(x,\cdot)$ is also a reproducer of x. The point of uniqueness here is that we do not have $k(x,\cdot) \in \mathcal{H}_0$. The next proposition makes this distinction precise.

Proposition 3.27 (Kernel of a closed subspace)**.** *Let $\mathcal{H}$ have the reproducing property on X with kernel k, and let $\mathcal{H}_0 \subset \mathcal{H}$ be a closed subspace. Then $\mathcal{H}_0$ has the reproducing property on X with kernel*

$$k_0(x,y) := \langle \Pi_{\mathcal{H}_0} k_y\,,\, k_x\rangle_{\mathcal{H}} = \big(\Pi_{\mathcal{H}_0} k_y\big)(x) = \big(\Pi_{\mathcal{H}_0} k_x\big)(y); \qquad x, y \in X, \tag{3.74}$$

where $\Pi_{\mathcal{H}_0} : \mathcal{H} \to \mathcal{H}_0$ is the orthogonal projection.

Proof. Let $f \in \mathcal{H}_0$ and $x \in X$. Since $\mathcal{H}_0 \subset \mathcal{H}$, we have

$$f(x) = \langle f\,,\, k_x\rangle_{\mathcal{H}} = \langle \Pi_{\mathcal{H}_0} f\,,\, k_x\rangle_{\mathcal{H}} = \langle f\,,\, \Pi_{\mathcal{H}_0}(k_x)\rangle_{\mathcal{H}}\,. \tag{3.75}$$

But $\Pi_{\mathcal{H}_0}(k_x) \in \mathcal{H}_0$, hence

$$f(x) = \langle f\,,\, \Pi_{\mathcal{H}_0}(k_x)\rangle_{\mathcal{H}_0}\,. \tag{3.76}$$

This shows that $\Pi_{\mathcal{H}_0}(k_x)$ is the reproducer of x in $\mathcal{H}_0$, which proves the claim. □

We can establish a converse to the previous result, namely:

Proposition 3.28. *Let $\mathcal{H}_0 \subset \mathcal{H}$ be a subspace that possesses the reproducing property on X for the metric induced by $\mathcal{H}$. Then the orthogonal projection $\Pi_{\mathcal{H}_0} : \mathcal{H} \to \mathcal{H}_0$ onto $\mathcal{H}_0$ is an integral operator whose kernel is the reproducing kernel k of $\mathcal{H}_0$. In other words, for all $f \in \mathcal{H}$ and all $x \in X$,*

$$\Pi_{\mathcal{H}_0}\big(f\big)(x) = \langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}}\,. \tag{3.77}$$

Proof. For any $f \in \mathcal{H}$ and $x \in X$, we have

$$\begin{aligned}\langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}} &\overset{k(x,\cdot)\in\mathcal{H}_0}{=} \langle \Pi_{\mathcal{H}_0} k(x,\cdot)\,,\, f\rangle_{\mathcal{H}} \overset{\Pi^*_{\mathcal{H}_0}=\Pi_{\mathcal{H}_0}}{=} \langle k(x,\cdot)\,,\, \Pi_{\mathcal{H}_0} f\rangle_{\mathcal{H}} \\ &\overset{\Pi_{\mathcal{H}_0} f\in\mathcal{H}_0}{=} \langle k(x,\cdot)\,,\, \Pi_{\mathcal{H}_0} f\rangle_{\mathcal{H}_0} = \big(\Pi_{\mathcal{H}_0} f\big)(x).\end{aligned} \tag{3.78}$$

According to Definition 3.2, this means precisely that k is the integral kernel associated with $\Pi_{\mathcal{H}_0}$. □

We now present an explicit recipe for constructing the reproducing kernel of a Hilbert space, once an orthonormal basis is known.

Proposition 3.29 (Kernel from an orthonormal basis)**.** *Let $\mathcal{H}$ be a Hilbert space having the reproducing property, with reproducing kernel k. Then for any orthonormal basis $(e_s)_{s\in\mathcal{S}}$ of $\mathcal{H}$, we have for all $x, y \in X$*

$$k(x,y) = \sum_{s\in\mathcal{S}} e_s(x) e_s(y), \tag{3.79}$$

where the sum converges pointwise, with all but countably many nonzero terms.

Proof. Fix $x \in X$. For each $s \in \mathcal{S}$, by the reproducing property,

$$e_s(x) = \langle e_s\,,\, k(x,\cdot)\rangle_{\mathcal{H}}\,. \tag{3.80}$$

Decomposing $k_x := k(x,\cdot) \in \mathcal{H}$ in the orthonormal basis $(e_s)_{s\in\mathcal{S}}$ gives

$$k_x = \sum_{s\in\mathcal{S}} \langle e_s\,,\, k_x\rangle_{\mathcal{H}}\; e_s = \sum_{s\in\mathcal{S}} e_s(x)\, e_s, \tag{3.81}$$

where the equality holds in the sense of $\mathcal{H}$, i.e.

$$\left\| k_x - \sum_{s\in\mathcal{S}} e_s(x)\, e_s \right\|_{\mathcal{H}} = 0. \tag{3.82}$$

By Proposition 3.17, convergence in $\mathcal{H}$ implies pointwise convergence, hence for all $y \in X$

$$k(x,y) = k_x(y) = \sum_{s\in\mathcal{S}} e_s(x)\, e_s(y). \tag{3.83}$$

□

In fact, this statement can be generalized. For this, we need the notion of *Parseval frames* (or simply *frames*), a concept widely used in applied mathematics (see Adcock and Huybrechs (2019, 2020a)).

Definition 3.30 (Parseval frame)**.** A family $(f_s)_{s\in S}$ in $\mathcal{H}$ is called a ***Parseval frame*** if it satisfies the Parseval property: for all $h \in \mathcal{H}$,

$$\|h\|_{\mathcal{H}}^2 = \sum_{s\in S} \left|\langle h\,,\, f_s\rangle_{\mathcal{H}}\right|^2. \tag{3.84}$$

Equivalently,

$$\forall\, h \in \mathcal{H}\ :\qquad h = \sum_{s\in S} \langle h\,,\, f_s\rangle_{\mathcal{H}}\, f_s. \tag{3.85}$$

At first sight this may appear somewhat mysterious. A useful characterization is that Parseval frames arise precisely as *projections* of an orthonormal basis of a larger Hilbert space. This point of view is particularly convenient in practice: for instance, consider $\mathrm{L}^2(\Omega)$ when Ω is a complicated domain. Constructing an orthonormal basis directly is difficult. However, if Ω is embedded in a hyper-rectangle in $\mathbb{R}^d$, the Fourier basis on the ambient rectangle projects to a Parseval frame for $\mathrm{L}^2(\Omega)$. We refer again to Adcock and Huybrechs (2019, 2020a) for further details.

Proposition 3.31 (Larson, Naimark)**.** *Let $\mathcal{H}$ be a Hilbert space and $(f_s)_{s\in S}$ a family in $\mathcal{H}$. Then $(f_s)_{s\in S}$ is a Parseval frame for $\mathcal{H}$ if and only if there exist:*

(i) a Hilbert space $\big(\mathcal{K}, \|\cdot\|_{\mathcal{K}}\big) \supseteq \big(\mathcal{H}, \|\cdot\|_{\mathcal{H}}\big)$ i.e. $\mathcal{H} \subseteq \mathcal{K}$ and $\|\cdot\|_{\mathcal{H}}$ is the restriction of $\|\cdot\|_{\mathcal{K}}$ to $\mathcal{H}$;

(ii) an orthonormal basis $(e_s)_{s\in S}$ of $\mathcal{K}$,

such that for all $s \in S$,

$$f_s = \Pi_{\mathcal{H}}(e_s), \tag{3.86}$$

where $\Pi_{\mathcal{H}} : \mathcal{K} \to \mathcal{H}$ is the orthogonal projection onto $\mathcal{H}$.

Proof. We refer to Paulsen and Raghupathi (2016, Propositon 2.9) for the the case where $\mathcal{S}$ is countable, following Larson. The general case is covered by Naimark's dilation theorem, which requires the machinery of projection–valued measures introduced in Section C.3 of Chapter C. □

We can now generalize Proposition 3.29; in fact, this leads to a characterization of Parseval frames in Hilbert spaces endowed with the reproducing property.

Theorem 3.32 (Papadakis)**.** *Let $\mathcal{H}$ have the reproducing property on X with kernel k. A family $(f_s)_{s\in S}$ in $\mathcal{H}$ is a Parseval frame if and only if*

$$k(x, y) = \sum_{s\in S} f_s(x)\, f_s(y) \tag{3.87}$$

with pointwise convergence.

Proof. Fix $x \in X$. Since $(f_s)_{s\in S}$ is a Parseval frame, we can expand the reproducer $k_x = k(x,\cdot) \in \mathcal{H}$ as

$$k_x = \sum_{s\in S} \langle k_x \,,\, f_s\rangle_{\mathcal{H}}\; f_s. \tag{3.88}$$

By the reproducing property,

$$\langle k_x \,,\, f_s\rangle_{\mathcal{H}} = f_s(x). \tag{3.89}$$

Hence

$$k_x = \sum_{s\in S} f_s(x)\, f_s. \tag{3.90}$$

Evaluating at $y \in X$ by means of the reproducing property, we obtain

$$k(x,y) = k_x(y) = \sum_{s\in S} f_s(x) f_s(y). \tag{3.91}$$

Conversely, assume that $(f_s)_{s\in S}$ is a family of functions on X such that

$$k(x,y) = \sum_{s\in S} f_s(x) f_s(y), \qquad x,y \in X. \tag{3.92}$$

Let $h = \sum_{j=1}^m \alpha_j k_{y_j}$ be any finite linear combination of reproducers. Then

$$\begin{aligned} \|h\|_{\mathcal{H}}^2 &= \sum_{i,j=1}^m \alpha_i\alpha_j \left\langle k_{y_j} \,,\, k_{y_i}\right\rangle_{\mathcal{H}} = \sum_{i,j=1}^m \alpha_i\alpha_j k(y_i,y_j) \\ &= \sum_{i,j=1}^m \alpha_i\alpha_j \sum_{s\in S} f_s(y_i) f_s(y_j) \\ &= \sum_{s\in S} \Big(\sum_{j=1}^m \alpha_j f_s(y_j)\Big)\Big(\sum_{i=1}^m \alpha_i f_s(y_i)\Big) \\ &= \sum_{s\in S} \Big|\sum_{j=1}^m \alpha_j f_s(y_j)\Big|^2. \end{aligned} \tag{3.93}$$

But

$$\sum_{j=1}^m \alpha_j f_s(y_j) = \sum_{j=1}^m \alpha_j \left\langle k_{y_j} \,,\, f_s\right\rangle_{\mathcal{H}} = \langle h \,,\, f_s\rangle_{\mathcal{H}}\,. \tag{3.94}$$

Thus

$$\|h\|_{\mathcal{H}}^2 = \sum_{s\in S} |\,\langle h \,,\, f_s\rangle_{\mathcal{H}}\,|^2. \tag{3.95}$$

By Proposition 3.24, $\mathrm{Span}\big(k_y \,:\, y \in X\big)$ is dense in $\mathcal{H}$. Hence the Parseval identity holds on a dense subspace and therefore extends to all of $\mathcal{H}$. Thus $(f_s)_{s\in S}$ is indeed a Parseval frame. □

We now show that, given a kernel $k : X \times X \to \mathbb{R}$ satisfying certain specific properties, one can always construct a Hilbert space on X enjoying the reproducing property. This is the classical *Aronszajn–Moore construction* (Aronszajn, 1950). Since in this setting $\mathcal{H}$ is completely determined by its kernel, we shall from this point onward speak not merely of a Hilbert space with the reproducing property, but of a *Reproducing Kernel Hilbert Space (RKHS)*.

3.3.2 . Aronszajn–Moore construction

We now turn to the converse question: starting from a *kernel function* k, can we construct an RKHS for which k is the reproducing kernel? In full generality the answer is negative. To make this possible, we must first identify and recover some fundamental properties that any reproducing kernel necessarily satisfies. We begin with some basic notions.

Definition 3.33 (Properties of kernels)**.** Let X be a nonempty set and let $k : X \times X \to \mathbb{R}$ be a function.

(i) k is ***symmetric*** if for all $x, y \in X$,

$$k(x, y) = k(y, x). \tag{3.96}$$

(ii) k is ***non-negative*** (or ***positive semi-definite***) if for all $n \in \mathbb{N}$, all $(x_i) \in X^n$, and all $(\alpha_i) \in \mathbb{R}^n$,

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j \, k(x_i, x_j) \;\geqslant\; 0. \tag{3.97}$$

(iii) k is ***definite*** if for all finite families of pairwise distinct points $(x_i) \in X^n$ and all $(\alpha_i) \in \mathbb{R}^n$,

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j \, k(x_i, x_j) = 0 \quad \Longrightarrow \quad \forall\, 1 \leqslant i \leqslant n \,:\, \alpha_i = 0. \tag{3.98}$$

(iv) k is ***positive definite*** (or ***strictly positive***) if it is both non-negative and definite.

(v) k is ***symmetric positive definite*** if it is both symmetric and positive definite.

***Remark* 3.34.** In view of Example 3.21, these notions are simply the natural generalization from matrices to kernels, extending the familiar properties of symmetric, positive semidefinite, and positive definite matrices to the infinite-dimensional setting.

Equivalently, one may restate Definition 3.33 by saying that k has a given property if and only if all its associated ***Gram matrices***

$$\big(k(x_i, x_j)\big)_{1 \leqslant i,j \leqslant n}, \qquad n \in \mathbb{N},\ (x_i) \in X^n, \tag{3.99}$$

have the corresponding property.

Notation 3.35. For kernels $k : X \times X \to \mathbb{R}$, we adopt the following shorthand:

(i) If k is *symmetric and non-negative* (positive semi-definite), we write $\mathbf{k \geqslant 0}$.

(ii) If k is *symmetric positive definite*, we write $\mathbf{k > 0}$.

Proposition 3.36 (Fundamental properties of reproducing kernels)**.** *If k is the reproducing kernel of a Hilbert space $\mathcal{H}$ of functions on X, then k is symmetric and non-negative.*

Proof. The symmetry has already been proved in Equation (3.13) of Corollary 3.15.

Let $n \in \mathbb{N}$, $x_1, \ldots, x_n \in X$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. Consider the function

$$f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \in \mathcal{H}. \tag{3.100}$$

Since norms are always non-negative, we have

$$0 \leqslant \|f\|_{\mathcal{H}}^2 = \langle f \, , \, f \rangle_{\mathcal{H}} = \sum_{i,j=1}^{n} \alpha_i \alpha_j \, \langle k(x_i, \cdot) \, , \, k(x_j, \cdot) \rangle_{\mathcal{H}} = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j). \tag{3.101}$$

This proves that k is non-negative. □

We now arrive at the central result of this section, and arguably of the entire theory: the Aronszajn–Moore construction of Reproducing Kernel Hilbert Spaces.

Theorem 3.37 (Moore)**.** *If $k \geqslant 0$ on $X \times X$, there exists a unique Hilbert space $\mathcal{H}(k)$ with the reproducing property on X and reproducing kernel k.*

This space is called the **Reproducing Kernel Hilbert Space (RKHS) *of* $\mathbf{k}$**.

Proof. For each $y \in X$, define the reproducing function of y, $k_y := k(\cdot, y) : X \to \mathbb{R}$. Let $\mathcal{W}$ denote the linear span of $\big(k_y : y \in X\big)$ inside the vector space of functions $\mathcal{F}(X \to \mathbb{R})$.

As observed in Proposition 3.25, whenever $\mathcal{W}$ sits inside a Hilbert space $\widetilde{\mathcal{H}}$ with reproducing kernel k, it is dense in $\widetilde{\mathcal{H}}$. The guiding idea is therefore to reconstruct $\widetilde{\mathcal{H}}$ starting from $\mathcal{W}$ alone. To this end, we must first equip $\mathcal{W}$ with a Hilbert topology, that is, introduce an inner product.

Define a bilinear form $B : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ by

$$B\left(\sum_{i=1}^{m} \alpha_i k_{x_i}, \sum_{j=1}^{n} \beta_j k_{y_j} \right) := \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j \, k(x_i, y_j). \tag{3.102}$$

Since k is symmetric, the form B is symmetric, and the non-negativity of k guarantees that $B(f,f) \geqslant 0$ for every $f \in \mathcal{W}$.

Well-definedness. Since a function in $\mathcal{W}$ may have several representations as a finite sum of reproducers, we must check that B is well defined, *i.e.* that if $f = \sum_i \alpha_i k_{x_i}$ is identically zero as a function on X, then $B(f,\cdot) = 0$ as a linear functional on $\mathcal{W}$.

Let f be such a function. For any $y \in X$ we compute

$$B(f, k_y) = \sum_i \alpha_i k(x_i, y) = f(y) = 0. \tag{3.103}$$

Since $\big(k_y : y \in X\big)$ spans $\mathcal{W}$, it follows that $B(f,g) = 0$ for all $g \in \mathcal{W}$. Thus the definition of B is independent of the chosen representation of f.

Definiteness. It follows that B is a positive semidefinite bilinear form on $\mathcal{W}$, and hence satisfies the Cauchy–Schwarz–Buniakovski inequality. Now suppose $f \in \mathcal{W}$ with $B(f,f) = 0$. Then for every $y \in X$,

$$|B(f, k_y)|^2 \leqslant \underbrace{B(f,f)}_{=0}\, B(k_y, k_y) = 0, \tag{3.104}$$

Hence $B(f,k_y) = 0$ for all $y \in X$, which means that $f(y) = 0$ for every y. Therefore f is identically zero as a function, and so $f = 0$ in $\mathcal{W}$. This proves that B is definite, and hence an inner product on $\mathcal{W}$, thereby making $\big(\mathcal{W}, B\big)$ into a pre–Hilbert space.

Completion and algebraic embedding into $\mathcal{F}(X \to \mathbb{R})$. Let $\big(\widetilde{\mathcal{H}}(k), \langle\cdot\,,\,\cdot\rangle_{\widetilde{\mathcal{H}}(k)}\big)$ be the Hilbert space obtained by completing $\mathcal{W}$ with respect to the inner product B, and let $\mathrm{O} : \big(\mathcal{W}, B\big) \to \big(\widetilde{\mathcal{H}}(k), \langle\cdot\,,\,\cdot\rangle_{\widetilde{\mathcal{H}}(k)}\big)$ denote the canonical isometric embedding. We now show that elements of $\widetilde{\mathcal{H}}(k)$ can be canonically identified with a subspace of $\mathcal{F}(X \to \mathbb{R})$. Define

$$\iota : \begin{cases} \widetilde{\mathcal{H}}(k) & \longrightarrow \quad \mathcal{F}(X \to \mathbb{R}) \\ h & \longmapsto \quad \Big(x \mapsto \langle h\,,\, \mathrm{O}(k_x)\rangle_{\widetilde{\mathcal{H}}(k)}\Big) \end{cases}. \tag{3.105}$$

Clearly ι is well defined and linear. We claim that ι is injective, *i.e.* $\operatorname{Ker}\iota = \{0\}$. Indeed, let $h \in \widetilde{\mathcal{H}}(k)$ with $\iota(h) = 0$. Then, by definition of ι, for all $x \in X$

$$\langle h\,,\, \mathrm{O}(k_x)\rangle_{\widetilde{\mathcal{H}}(k)} = 0, \tag{3.106}$$

which shows that $h \in \mathrm{O}(\mathcal{W})^{\perp}$. Since $\mathrm{O}(\mathcal{W})$ is dense in $\widetilde{\mathcal{H}}(k)$ by construction, we conclude that $h = 0$.

Isometric embedding into $\mathcal{F}(X \to \mathbb{R})$ and reproducing property. We endow $\mathcal{H}(k) := \operatorname{Im}\iota$ with the inner product

$$\langle\cdot\,,\,\cdot\rangle_{\mathcal{H}(k)} : \begin{cases} \mathcal{H}(k) \times \mathcal{H}(k) & \to \quad \mathbb{R}, \\ \big(\iota(g), \iota(h)\big) & \mapsto \quad \langle g\,,\, h\rangle_{\widetilde{\mathcal{H}}(k)} \end{cases}, \tag{3.107}$$

so that

$$\iota : \Big(\widetilde{\mathcal{H}}(k), \langle \cdot\,,\, \cdot \rangle_{\widetilde{\mathcal{H}}(k)}\Big) \longrightarrow \Big(\mathcal{H}(k), \langle \cdot\,,\, \cdot \rangle_{\mathcal{H}(k)}\Big), \tag{3.108}$$

is a surjective isometry.

Note also that $\mathcal{W} \subset \mathcal{H}(k)$, with the inclusion holding both algebraically and topologically, *i.e.* $\iota \circ \mathrm{O} = \mathcal{I}_{\mathcal{W}}$ holds both algebraically and topologically.

Finally, $\mathcal{H}(k)$ has the reproducing property with kernel k. Indeed, for all $y \in X$ and $\iota(h) \in \mathcal{H}(k)$,

$$\begin{aligned} \delta_y(\iota(h)) &\overset{(3.105)}{=} \delta_y\Big(x \mapsto \langle h\,,\, \mathrm{O}(k_x)\rangle_{\widetilde{\mathcal{H}}(k)}\Big) = \langle h\,,\, \mathrm{O}(k_y)\rangle_{\widetilde{\mathcal{H}}(k)} \\ &\overset{(3.107)}{=} \Big\langle \iota(h), \underbrace{\iota\big(\mathrm{O}(k_y)\big)}_{=\mathcal{I}_{\mathcal{W}}(k_y)} \Big\rangle_{\mathcal{H}(k)} = \langle \iota(h)\,,\, k_y\rangle_{\mathcal{H}(k)}\,. \end{aligned} \tag{3.109}$$

Uniqueness. Finally, if $\mathcal{H}_1$ and $\mathcal{H}_2$ are two RKHSs with reproducing kernel k, then by Proposition 3.25 we have

$$\big(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\big) = \big(\mathcal{H}_2, \|\cdot\|_{\mathcal{H}_2}\big), \tag{3.110}$$

that is, the two spaces coincide both algebraically and topologically.

This completes the Aronszajn–Moore construction. □

***Remark* 3.38.** It is worth stressing that Theorem 3.37 asserts more than equality up to isomorphism. An isomorphism would only guarantee a topological identification between Hilbert spaces. Here, by contrast, the reproducing kernel uniquely fixes the Hilbert space structure, so that equality holds both algebraically and topologically. This distinction is important: for example, every separable Hilbert space of countable dimension is (topologically) isomorphic to ℓ^2, but they are not algebraically the same space.

Note that it is generally an hard task to get a concrete description of the space $\mathcal{H}(k)$ starting from the kernel k. This is called the ***Reconstruction problem***. The classical example is the Hardy space $\mathrm{H}(\mathbb{D}) : \mathbb{D} \to \mathbb{C}$ of analytic functions on the disk $\mathbb{D}$ of ray 1, endowed with the hermitian product making monomials an orthonormal family[2]. We can show that the reproducing kernel is for all $z, w \in \mathbb{D}$

$$k(z, w) := \frac{1}{1 - \overline{w}z}. \tag{3.111}$$

Then in particular not polynomial in $\mathrm{Span}\big(k(z, \cdot) \,:\, z \in \mathbb{D}\big)$, while polynomials are in fact dense in $\mathrm{H}(\mathbb{D})$! Neverthelss, we shall not pay too much attention to this, as it is far from our considerations.

[2]Note that this sous-entend to extend our construction to Hermitian Spaces, something we did not cover for simplicity

Note that obtaining a concrete description of the space $\mathcal{H}(k)$ from the kernel k is in general a difficult task. This is known as the ***Reconstruction problem***.

A classical example is given by the Hardy space $\mathrm{H}(\mathbb{D})$ of analytic functions on the unit disk $\mathbb{D} \subset \mathbb{C}$, endowed with the Hermitian inner product for which the monomials form an orthonormal family.[3] Its reproducing kernel is, for all $z, w \in \mathbb{D}$,

$$k(z,w) = \frac{1}{1-\overline{w}z}. \tag{3.112}$$

In particular, observe that $\mathrm{Span}\big(k(z,\cdot) : z \in \mathbb{D}\big)$ does not contain any polynomial, whereas the polynomials themselves are dense in $\mathrm{H}(\mathbb{D})$.

Nevertheless, we shall not dwell on these considerations, as they lie outside the main scope of our discussion. We shall only mention a somewhat "funny" example.

***Example* 3.39.** Let $f : X \to \mathbb{R}$ be a nonzero function. Then

$$k : \left\{ \begin{array}{ccc} X \times X & \to & \mathbb{R} \\ (x,y) & \mapsto & f(x)f(y) \end{array} \right. \tag{3.113}$$

is positive semidefinite, and hence defines an RKHS $\mathcal{H}(k)$. In fact, $\mathcal{H}(k)$ is simply the one-dimensional subspace $\mathrm{Span}(f) \subset \mathcal{F}(X \to \mathbb{R})$, endowed with the norm $\|\alpha f\| = |\alpha|$.

Proof. To check positivity, let $(x_i) \in X^n$ and $(\alpha_i) \in \mathbb{R}^n$. Then

$$\sum_{i,j=1}^n \alpha_i \alpha_j k(x_i,x_j) = \sum_{i,j=1}^n \alpha_i \alpha_j f(x_i) f(x_j) = \left(\sum_{i=1}^n \alpha_i f(x_i) \right)^2 \geqslant 0, \tag{3.114}$$

so $k \geqslant 0$. Next, every reproducer is a scalar multiple of f, since

$$k_y(\cdot) = k(\cdot, y) = f(y) f(\cdot). \tag{3.115}$$

Hence the span $\mathcal{W} = \mathrm{Span}\big(k_y : y \in X\big)$ reduces to the one-dimensional space $\mathrm{Span}(f)$. Being finite-dimensional, it is automatically complete, so $\mathcal{H}(k) = \mathrm{Span}(f)$.

Finally, to identify the norm, pick $y \in X$ with $f(y) \neq 0$. Since $f = \frac{1}{f(y)}\, k_y$, we obtain

$$\|f\|^2_{\mathcal{H}(k)} = \frac{\|k_y\|^2_{\mathcal{H}(k)}}{|f(y)|^2} = \frac{k(y,y)}{|f(y)|^2} = \frac{f(y)^2}{f(y)^2} = 1. \tag{3.116}$$

Hence, for any $\alpha \in \mathbb{R}$, $\|\alpha f\|_{\mathcal{H}(k)} = |\alpha|$, which completes the description of $\mathcal{H}(k)$. □

[3] This implicitly requires extending our construction to complex Hilbert spaces, which we did not cover here for simplicity.

3.3.3 . Regularity of $\mathcal{H}(k)$ inherited from its kernel

To conclude this section, we illustrate the elegance of the theory by showing that the regularity of the kernel k determines the regularity of the functions in $\mathcal{H}(k)$. We begin with continuity, which is straightforward to establish.

Proposition 3.40. *Let $(X, \|\cdot\|_X)$ be a normed space and let $k : X \times X \to \mathbb{R}$ be a continuous positive semidefinite kernel with respect to the product topology on $X \times X$. Then every $f \in \mathcal{H}(k)$ is continuous on X.*

Proof. Fix $x \in X$ and let $(x_n) \in X^{\mathbb{N}}$ with $x_n \to x$. For any $f \in \mathcal{H}(k)$, by the reproducing property and Cauchy–Schwarz–Buniakovski inequality we have

$$|f(x_n) - f(x)| = \left|\langle k_{x_n} - k_x \,,\, f\rangle_{\mathcal{H}(k)}\right| \leqslant \|k_{x_n} - k_x\|_{\mathcal{H}(k)} \, \|f\|_{\mathcal{H}(k)} \,. \tag{3.117}$$

It therefore suffices to prove that $\|k_{x_n} - k_x\|^2_{\mathcal{H}(k)} \to 0$. Expanding it gives by the reproducing property

$$\langle k_{x_n} - k_x \,,\, k_{x_n} - k_x\rangle_{\mathcal{H}(k)} = k(x_n, x_n) - 2k(x, x_n) + k(x, x). \tag{3.118}$$

By continuity of k on $X \times X$, this indeed converges to 0 as $n \to \infty$. □

***Remark* 3.41.** This result can be generalized to non-sequential topologies. However, the proof in that setting is more technical, so we omit it here and refer the reader to Paulsen and Raghupathi (2016, Theorem 2.17) for details.

Note that we crucially use here the fact that k is continuous as a function on $X \times X$, and not merely separately continuous in each variable, since the latter would not guarantee the convergence of $k(x_n, x_n)$ to $K(x, x)$.

This result can be generalized in the following way.

Proposition 3.42. *Let $(X, \|\cdot\|_X)$ be a finite–dimensional normed space and let $k : X \times X \to \mathbb{R}$ be a positive semidefinite kernel of class $\mathcal{C}^{2m}$ with respect to the product topology on $X \times X$.*

Then every $f \in \mathcal{H}(k)$ is of class $\mathcal{C}^m$ on X. Furthermore, for every multi-index $|\alpha| \leqslant m$ and $x \in X$, there exist $k^\alpha_x \in \mathcal{H}(k)$ such that for all $f \in \mathcal{H}(k)$

$$\partial^\alpha f(x) = \langle k^\alpha_x \,,\, f\rangle_{\mathcal{H}(k)} \,. \tag{3.119}$$

Proof. We begin by fixing some notation. For multi-indices α, β, we denote by $\partial^{(\alpha,\beta)}k$ the mixed partial derivatives of k with respect to its two kernel arguments. Concretely, for a simple index i, we set

$$\partial^{(i,0)}k(x, y) := \lim_{\varepsilon \to 0} \frac{k(x + \varepsilon\, e_i, y) - k(x, y)}{\varepsilon}, \tag{3.120}$$

and for a simple index j,

$$\partial^{(i,j)}k(x, y) := \lim_{\varepsilon \to 0} \frac{\partial^{(i,0)}k(x, y + \varepsilon\, e_j) - \partial^{(i,0)}k(x, y)}{\varepsilon}. \tag{3.121}$$

Note that $\partial^{(\alpha,\beta)}k$ is a mixed partial derivative of total order $|\alpha| + |\beta|$. This hints at the requirement that k be of class $\mathcal{C}^{2m}$ in order to control derivatives of functions in $\mathcal{H}(k)$ up to order m, as we shall see.

By the Schwarz–Clairaut–Young theorem on the symmetry of mixed derivatives, these partial derivatives may be computed in any order, even when alternating between the two kernel variables.

Suppose now that such an element k_x^α, as described in Equation (3.119), exists for $x \in X$ and a multi–index α, and moreover lies in $\mathcal{H}(k)$. Then for each $y \in X$, the reproducing property gives

$$\langle k_x^\alpha \,,\, k_y \rangle = k_x^\alpha(y) \overset{(3.119)}{=} \partial^\alpha k_y(x) = \partial^{(\alpha,0)}k(x,y). \tag{3.122}$$

This entirely characterizes k_x^α as a function in $\mathcal{F}(X \to \mathbb{R})$. Hence, three things remain to be checked:

(i) that for all $x \in X$, $\partial^{(\alpha,0)}k(x,\cdot) \in \mathcal{H}(k)$,

(ii) that Equation (3.119) indeed holds for all $f \in \mathcal{H}(k)$,

(iii) that these partial derivatives are continuous.

We will establish these properties by induction on the order $m = |\alpha|$ of the partial derivative.

$\mathbf{m = 1}$, k of class $\mathcal{C}^2$. Fix $1 \leqslant i \leqslant \dim(X)$ and $y \in X$.

(i) $\boldsymbol{\partial^{(i,0)}k(x,\cdot) \in \mathcal{H}(k)}$. Since $X \times X$ is equipped with the product topology, for any ball B centered at y and contained in X, the set $B \times X$ is a neighborhood of (y,x) for every $x \in X$. Then for any $n > N$, with N large enough so that $y + 2^{-n}e_i \in B$, we have $k_{y+2^{-n}e_i} \in \mathcal{H}$, and we may define

$$c_n^i := \frac{k_{y+2^{-n}e_i} - k_y}{2^{-n}} \;\in\; \mathcal{H}. \tag{3.123}$$

Furthermore, for any $x \in X$,

$$\frac{k_{y+2^{-n}e_i}(x) - k_y(x)}{2^{-n}} = \frac{k\big((y,x) + (2^{-n}e_i, 0)\big) - k\big((y,x)\big)}{2^{-n}}, \tag{3.124}$$

and this converges to $\partial^{(i,0)}k(x,y)$ as $n \to \infty$. In other words, the sequence $(c_n^i)_{n \geqslant N}$ converges pointwise to the desired function.

It remains to show that the convergence also holds in $\|\cdot\|_{\mathcal{H}(k)}$. Since convergence in $\|\cdot\|_{\mathcal{H}(k)}$ implies pointwise convergence, uniqueness of the limit will then ensure that $\partial^{(i,0)}k(x,\cdot) \in \mathcal{H}(k)$.

Since $\mathcal{H}(k)$ is complete, it is enough to show that (c_n^i) is Cauchy. To lighten the notation, let us write $h_n := 2^{-n}$ and, with a slight abuse, use $k(x + h_n, \cdot)$

instead of $k(x + h_n e_i, \cdot)$. For $p, q \geqslant N$, we compute:

$$\begin{aligned}\left\langle c_p^i \,,\, c_q^i \right\rangle_{\mathcal{H}(k)} &= \left\langle \tfrac{k_{x+h_p} - k_x}{h_p} \,,\, \tfrac{k_{x+h_q} - k_x}{h_q} \right\rangle_{\mathcal{H}(k)} \qquad (3.125)\\ &= \frac{1}{h_p h_q}\Big[k(x + h_p, x + h_q) + k(x, x) - k(x + h_p, x) - k(x + h_q, x)\Big]\\ &= \frac{1}{h_p}\left[\frac{k(x + h_p, x + h_q) - k(x + h_p, x)}{h_q} - \frac{k(x + h_q, x) - k(x, x)}{h_q}\right].\end{aligned}$$

Taking the limit as $q \to \infty$, this gives

$$\lim_{q\to\infty} \left\langle c_p^i \,,\, c_q^i \right\rangle_{\mathcal{H}(k)} = \frac{1}{h_p}\Big[\partial^{(i,0)}k(x + h_p, x) - \partial^{(i,0)}k(x, x)\Big], \qquad (3.126)$$

and therefore

$$\lim_{p\to\infty} \lim_{q\to\infty} \left\langle c_p^i \,,\, c_q^i \right\rangle_{\mathcal{H}(k)} = \partial^{((i,i),0)}k(x, x). \qquad (3.127)$$

Note that if $p = q$, then since k is of class $\mathcal{C}^2$ on $X \times X$ (and in particular at (x, x)), we also have

$$\lim_{p\to\infty} \left\langle c_p^i \,,\, c_p^i \right\rangle_{\mathcal{H}(k)} = \partial^{((i,i),0)}k(x, x). \qquad (3.128)$$

Finally, observe that

$$\left\langle c_n^i - c_m^i \,,\, c_n^i - c_m^i \right\rangle_{\mathcal{H}(k)} = \left\langle c_n^i \,,\, c_n^i \right\rangle_{\mathcal{H}(k)} + \left\langle c_m^i \,,\, c_m^i \right\rangle_{\mathcal{H}(k)} - 2\left\langle c_n^i \,,\, c_m^i \right\rangle_{\mathcal{H}(k)}. \qquad (3.129)$$

Hence,

$$\lim_{m\to\infty} \lim_{n\to\infty} \left\langle c_n^i - c_m^i \,,\, c_n^i - c_m^i \right\rangle_{\mathcal{H}(k)} = 0, \qquad (3.130)$$

which shows that (c_n^i) is indeed Cauchy in $\mathcal{H}(k)$.

***Remark* 3.43.** Using the symmetry of k in Equation (3.126), we also obtain

$$\lim_{p\to\infty} \lim_{q\to\infty} \left\langle c_p^i \,,\, c_q^i \right\rangle_{\mathcal{H}(k)} = \partial^{(i,i)}k(x, x), \qquad (3.131)$$

which shows that for all $x \in X$,

$$\partial^{(i,i)}k(x, x) = \partial^{((i,i),0)}k(x, x). \qquad (3.132)$$

In particular, this identity requires only that k be of class $\mathcal{C}^1$ in each of its variables separately. However, this is not sufficient to ensure convergence of the diagonal terms

$$\lim_{p\to\infty} \left\langle c_p^i \,,\, c_p^i \right\rangle_{\mathcal{H}(k)} = \partial^{((i,i),0)}k(x, x). \qquad (3.133)$$

This clarifies why the stronger assumption $k \in \mathcal{C}^{2m}$ is imposed in order to recover $\mathcal{C}^m$ regularity for functions in $\mathcal{H}(k)$.

(*ii*) **Equation (3.119) holds**. Let $f \in \mathcal{H}(k)$ and $x \in X$. Then for any $n \geqslant N$,

$$\frac{f(x) - f(x + 2^{-n}e_i)}{2^{-n}} = \left\langle c_n^i \,,\, f \right\rangle_{\mathcal{H}(k)}. \tag{3.134}$$

Since c_n^i converges to $\partial^{(i,0)}k_x$ in $\mathcal{H}(k)$, the right-hand side converges to $\left\langle \partial^{(i,0)}k_x \,,\, f \right\rangle_{\mathcal{H}(k)}$ in $\mathbb{R}$. This proves both that $\partial^i f(x)$ exists and that it is reproduced by $\partial^{(i,0)}k_x$.

(*iii*) $\boldsymbol{\partial^i f(\cdot)}$ **is continuous**. This is essentially an adaptation of the proof of the Proposition 3.40, now applied to $\partial^{(i,0)}k_{(\cdot)}$. Namely, given $x \in X$ and a sequence $(x_n) \in X^{\mathbb{N}}$ converging to x,

$$|\partial^i f(x_n) - \partial^i f(x)| \leqslant \left\| \partial^{(i,0)}k_{x_n} - \partial^{(i,0)}k_x \right\|_{\mathcal{H}(k)} \|f\|_{\mathcal{H}(k)}. \tag{3.135}$$

Moreover, applying the reproducing property of Equation (3.119) to $\partial^{(\alpha,0)}k_x$ itself,

$$\left\| \partial^{(i,0)}k_{x_n} - \partial^{(i,0)}k_x \right\|^2_{\mathcal{H}(k)} = \partial^{(i,i)}k(x,x) + \partial^{(i,i)}k(x_n,x_n) - 2\,\partial^{(i,i)}k(x,x_n), \tag{3.136}$$

which converges to 0 as $n \to \infty$, since k is of class $\mathcal{C}^2$.

Induction step: k of class $\mathcal{C}^{\mathbf{2(m+1)}}$. Suppose that for every multi-index α with $|\alpha| \leqslant m$, we have $\partial^{(\alpha,0)}k_x \in \mathcal{H}(k)$ and that it satisfies Equation (3.119), thereby yielding continuous partial derivatives.

Fix now a multi-index α such that $|\alpha| = m$, let $1 \leqslant i \leqslant \dim(X)$, and $y \in X$. We shall denote by $[\alpha, i]$ the concatenation of the multi-index α and the index i.

As in the case $m = 1$, introduce the sequence

$$c_n^{[\alpha,i]} := \frac{\partial^{(\alpha,0)}k_{y+2^{-n}e_i} - \partial^{(\alpha,0)}k_y}{2^{-n}}. \tag{3.137}$$

Every term of this sequence lies in $\mathcal{H}(k)$, since by the induction hypothesis both $\partial^{(\alpha,0)}k_{y+2^{-n}e_i}$ and $\partial^{(\alpha,0)}k_y$ belong to $\mathcal{H}(k)$.

By the same arguments as in the base case, we see that this sequence converges pointwise to $\partial^{([\alpha,i],0)}k_y$ and is Cauchy in $\mathcal{H}(k)$. Hence it converges in $\|\cdot\|_{\mathcal{H}(k)}$ to $\partial^{([\alpha,i],0)}k_y$, showing that

$$\partial^{([\alpha,i],0)}k_y \in \mathcal{H}(k). \tag{3.138}$$

This proves point (*i*).

Points (*ii*) and (*iii*) follow by entirely analogous arguments, and we shall not repeat the details here. □

For the sake of completeness, let us state the case $\mathcal{C}^\infty$ separately, given its singular nature and particular importance.

Corollary 3.44. *Let $(X, \|\cdot\|_X)$ be a finite–dimensional normed space and let $k : X \times X \to \mathbb{R}$ be a positive semidefinite kernel of class $\mathcal{C}^\infty$ with respect to the product topology on $X \times X$.*

Then every $f \in \mathcal{H}(k)$ is of class $\mathcal{C}^\infty$ on X. Furthermore, for every multi-index α and $x \in X$, there exist $k_x^\alpha \in \mathcal{H}(k)$ such that for all $f \in \mathcal{H}(k)$

$$\partial^\alpha f(x) = \langle k_x^\alpha \,,\, f \rangle_{\mathcal{H}(k)} \,. \tag{3.119}$$

We now turn to a topic that is rarely discussed in standard RKHS textbooks, yet lies at the very core of this dissertation. Indeed, it constitutes the implicit cornerstone upon which all our subsequent contributions are built.

3.4 . Effects of Norm Modifications on the Reproducing Kernel

In this section, we address a deceptively simple but fundamental question: given an RKHS $\big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big)$, if we endow it with another norm $\|\cdot\|_{\text{bis}}$, does $\big(\mathcal{H}(k), \|\cdot\|_{\text{bis}}\big)$ remain a reproducing kernel Hilbert space?

Before answering this question, let us carefully restate what it entails. In this setting, the space $\mathcal{H}(k)$ remains unchanged *algebraically*, that is, it contains the same elements, but its topology is altered through the new norm $\|\cdot\|_{\text{bis}}$. However, we established in Section 3.3.2 that the kernel k uniquely characterizes $\mathcal{H}(k)$ both *algebraically* and *topologically*. Consequently, k can no longer serve as the reproducing kernel of $\big(\mathcal{H}(k), \|\cdot\|_{\text{bis}}\big)$. This observation shows that the question of whether the resulting space remains an RKHS is indeed meaningful.

We begin by establishing several positive results under specific, yet practically important, assumptions.

3.4.1 . First Positive Results

We first show that when the new norm $\|\cdot\|_{\text{bis}}$ is equivalent to the original norm $\|\cdot\|_{\mathcal{H}(k)}$, the space $\big(\mathcal{H}(k), \|\cdot\|_{\text{bis}}\big)$ remains an RKHS.

Proposition 3.45. *Let $\big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big)$ be an RKHS and let $\|\cdot\|_{bis}$ be a norm equivalent to $\|\cdot\|_{\mathcal{H}(k)}$. Then $\big(\mathcal{H}(k), \|\cdot\|_{bis}\big)$ is also an RKHS.*

Proof. We need to show that $\big(\mathcal{H}(k), \|\cdot\|_{\text{bis}}\big)$ has the reproducing property on X and is complete.

Reproducing property. Since $\|\cdot\|_{\text{bis}}$ is equivalent to $\|\cdot\|_{\mathcal{H}(k)}$, there exist constants $C_-, C_+ > 0$ such that for all $u \in \mathcal{H}(k)$,

$$C_- \, \|u\|_{\text{bis}} \;\leqslant\; \|u\|_{\mathcal{H}(k)} \;\leqslant\; C_+ \, \|u\|_{\text{bis}} \,. \tag{3.139}$$

Therefore, for all $x \in X$ and $f \in \mathcal{H}(k)$,

$$|f(x)| \leqslant C_x \, \|f\|_{\mathcal{H}(k)} \leqslant C_x \, C_+ \, \|f\|_{\text{bis}} \,, \tag{3.140}$$

which shows that $\big(\mathcal{H}(k), \|\cdot\|_{\text{bis}}\big)$ also has the reproducing property on X.

Completeness. Let $(c_n) \in \mathcal{H}^{\mathbb{N}}$ be a Cauchy sequence for $\|\cdot\|_{\text{bis}}$. Then, for all $n, m \in \mathbb{N}$,

$$\|c_n - c_m\|_{\text{bis}} \geqslant \frac{1}{C_+} \|c_n - c_m\|_{\mathcal{H}(k)} . \tag{3.141}$$

Thus (c_n) is also Cauchy for $\|\cdot\|_{\mathcal{H}(k)}$ and converges to some $c_\infty \in \mathcal{H}(k)$. Moreover,

$$\|c_n - c_\infty\|_{\text{bis}} \leqslant \frac{1}{C_-} \|c_n - c_\infty\|_{\mathcal{H}(k)} , \tag{3.142}$$

so (c_n) also converges to c_∞ in $\|\cdot\|_{\text{bis}}$.

Both conditions are satisfied, which concludes the proof. □

In the above proof of Proposition 3.45, the reproducing property relied only on the fact that $\|\cdot\|_{\text{bis}}$ is stronger than $\|\cdot\|_{\mathcal{H}(k)}$. Therefore, if we already assume that $\big(\mathcal{H}(k), \|\cdot\|_{\text{bis}}\big)$ is complete, the result still holds. We highlight it in the following proposition.

Proposition 3.46. *Let $\big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big)$ be an RKHS on X, and let $\big(H_+, \|\cdot\|_+\big)$ be a Hilbert space such that the embedding*

$$O_+ : \big(H_+, \|\cdot\|_+\big) \to \big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big) \tag{3.143}$$

is continuous and injective. Then $\big(H_+, \|\cdot\|_+\big)$ is an RKHS on X.

Proof. Completeness of H_+ is assumed. Thus, we only need to show the reproducing property, which follows by the same argument as in the proof of Proposition 3.45. □

***Remark* 3.47.** More generally, this result holds if and only if the image of the unit ball in H_+ is closed in $\operatorname{Im} \mathrm{O}_+$ for the topology induced by $\|\cdot\|_{\mathcal{H}(k)}$. However, proving this requires more advanced topological tools, which we will not develop here. We refer instead to Schwartz (1964, Proposition 1, Chapter 0).

Proposition 3.45 and Proposition 3.46 are important, but admittedly quite abstract. In particular, it is natural to ask whether we can recover explicit formulas for the reproducing kernel of the modified space from the original kernel k of $\big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big)$. This is precisely the purpose of the following proposition.

Proposition 3.48. *Let $\big(\mathcal{H}_+, \|\cdot\|_+\big)$ and $\big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big)$ be as in Proposition 3.46, and denote by*

$$O_+ : \big(\mathcal{H}_+, \|\cdot\|_+\big) \to \big(\mathcal{H}(k), \|\cdot\|_{\mathcal{H}(k)}\big) \tag{3.144}$$

the continuous injective embedding of $\mathcal{H}_+$ into $\mathcal{H}(k)$. Then the reproducing kernel k_+ of $\big(\mathcal{H}_+, \|\cdot\|_+\big)$ on X is given for all $x, y \in X$ by

$$k_+(x, y) := O_+^*[k(x, \cdot)]\,(y). \tag{3.145}$$

Proof. The proof is very similar to that of Proposition 3.27. Since $\mathrm{H}_+ \subset \mathcal{H}(k)$, we have for all $f \in \mathrm{H}_+$ and $x \in X$

$$f(x) = \langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}(k)} = \langle k(x,\cdot)\,,\, \mathrm{O}_+ f\rangle_{\mathcal{H}(k)} = \left\langle \mathrm{O}_+^* k(x,\cdot)\,,\, f\right\rangle_+ . \tag{3.146}$$

Thus $\mathrm{O}_+^* k(x,\cdot) \in \mathcal{H}_+$ reproduces evaluation at x on $\mathcal{H}_+$, so it must be precisely $k_+(x,\cdot)$. Evaluating at $y \in X$ gives

$$k_+(x,y) = \mathrm{O}_+^*[k(x,\cdot)]\,(y). \tag{3.147}$$

This establishes the claim. □

***Remark* 3.49.** Since by Proposition C.5 in Section C.2.1 of Chapter C, $\mathrm{O}_+^* \Pi_{\overline{\mathrm{Im}\,\mathcal{H}_+}^{\mathcal{H}(k)}} = \mathrm{O}_+^*$, we can also write for all $x, y \in X$:

$$k_+(x,y) = \left(\mathrm{O}_+^* \Pi_{\overline{\mathrm{Im}\,\mathcal{H}_+}^{\mathcal{H}(k)}}\right)[k(x,\cdot)]\,(y). \tag{3.148}$$

In other words, one may either pull back the kernel of $\mathcal{H}(k)$ via O_+^*, or equivalently pull back the kernel of the closed subspace $\overline{\mathrm{Im}\,\mathcal{H}_+}^{\mathcal{H}(k)}$, since by Proposition 3.27, this subspace kernel can be written explicitly as

$$k_{\overline{\mathrm{Im}\,\mathcal{H}_+}^{\mathcal{H}(k)}}(x,y) = \Pi_{\overline{\mathrm{Im}\,\mathcal{H}_+}^{\mathcal{H}(k)}}\big[k(x,\cdot)\big](y). \tag{3.149}$$

***Remark* 3.50.** By the definition of the reproducing property given in Definition 3.11, we know that H_+ is an RKHS if and only if $\delta_x \in \mathrm{H}_+'$. Note in particular that this condition does not require H_0 itself to be an RKHS.

We now turn to an important special case, where the new norm is defined through a bijective and bicontinuous operator D. This situation will play a central role in the sequel, as it will enable us to compute the inverse of D in an implicit yet elegant way.

3.4.2 . Norm induced by a bicontinuous operator D

Suppose we have an RKHS $\left(\mathcal{H}(k), \|\cdot\|_0\right)$ on X, a Hilbert space $\left(\mathcal{H}_1, \|\cdot\|_1\right)$ and a bicontinuous linear operator

$$D : \left(\mathcal{H}(k), \|\cdot\|_0\right) \to \left(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\right), \tag{3.150}$$

that is, D is bijective, continuous, and has a continuous inverse. We may then define on $\mathcal{H}(k)$ the new norm

$$\|u\|_1 := \|D[u]\|_{\mathcal{H}_1}\,, \qquad u \in \mathcal{H}(k). \tag{3.151}$$

Since $D : \left(\mathcal{H}(k), \|\cdot\|_0\right) \to \left(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\right)$ is bicontinuous, it follows that $\|\cdot\|_1$ is equivalent to $\|\cdot\|_0$. This ensures the existence of the continuous embedding operators:

$$\mathrm{O}_1 : \left(\mathcal{H}(k), \|\cdot\|_1\right) \to \left(\mathcal{H}(k), \|\cdot\|_0\right), \quad \mathrm{O}_0 : \left(\mathcal{H}(k), \|\cdot\|_0\right) \to \left(\mathcal{H}(k), \|\cdot\|_1\right). \tag{3.152}$$

Observe that this construction makes the operator

$$D_1 := D\mathrm{O}_1 : u \in \big(\mathcal{H}(k), \|\cdot\|_1\big) \mapsto D[u] \in \big(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\big) \tag{3.153}$$

an isometry.

Moreover, by Proposition 3.45, $\big(\mathcal{H}(k), \|\cdot\|_1\big)$ is an RKHS. The natural question is then: what is its reproducing kernel ? The answer is provided by the following proposition.

Proposition 3.51. *The reproducing kernel k_1 of $\big(\mathcal{H}(k), \|\cdot\|_1\big)$ is given, for all $x, y \in X$, by*

$$k_1(x,y) = \big(O_0\, A^{-1}\big)[k(x,\cdot)]\,(y) \;\simeq\; A^{-1}[k(x,\cdot)]\,(y), \tag{3.154}$$

where $A = D^*D$, which is invertible since D is.

In order to prove Proposition 3.51, we need the following lemma:

Lemma 3.52. *Let $\big(\mathcal{H}, \|\cdot\|_0\big)$ be an Hilbert space and $\|\cdot\|_1$ an equivalent norm on $\mathcal{H}$. Consider the associated embedding operators:*

$$O_1 : \big(\mathcal{H}, \|\cdot\|_1\big) \to \big(\mathcal{H}, \|\cdot\|_0\big), \qquad O_0 : \big(\mathcal{H}, \|\cdot\|_0\big) \to \big(\mathcal{H}, \|\cdot\|_1\big). \tag{3.155}$$

Then, we have:

$$O_0^{-1} = O_1\,; \qquad\qquad O_1^{-1} = O_0, \tag{3.156}$$

Proof. Since O_0 and O_1 are both *algebraically* the identity, we have

$$\mathrm{O}_1\mathrm{O}_0 = \mathcal{I}_0 : \big(\mathcal{H}(k), \|\cdot\|_0\big) \to \big(\mathcal{H}(k), \|\cdot\|_0\big), \tag{3.157}$$
$$\mathrm{O}_0\mathrm{O}_1 = \mathcal{I}_1 : \big(\mathcal{H}(k), \|\cdot\|_1\big) \to \big(\mathcal{H}(k), \|\cdot\|_1\big), \tag{3.158}$$

which precisely means

$$\mathrm{O}_0^{-1} = \mathrm{O}_1\,; \qquad\qquad \mathrm{O}_1^{-1} = \mathrm{O}_0, \tag{3.156}$$

with equality holding both algebraically and topologically. □

Proof of Proposition 3.51. From Proposition 3.48, we know that for all $x, y \in X$,

$$k_1(x,y) = \mathrm{O}_1^*[k(x,\cdot)]\,(y). \tag{3.159}$$

Thus, it suffices to prove that $\mathrm{O}_1^* = \mathrm{O}_0\mathrm{A}^{-1}$. We now prove that $D^*D = \mathrm{O}_0^*\mathrm{O}_0$. For any $u, v \in \mathcal{H}(k)$, we compute:

$$\langle D^*Du\,,\, v\rangle_0 = \langle Du\,,\, Dv\rangle_{\mathcal{H}_1} \overset{(3.151)}{=} \langle u\,,\, v\rangle_1 \overset{(3.152)}{=} \langle \mathrm{O}_0 u\,,\, \mathrm{O}_0 v\rangle_1 = \langle \mathrm{O}_0^*\mathrm{O}_0 u\,,\, v\rangle_0\,. \tag{3.160}$$

Hence, for all $u \in \mathcal{H}(k)$, $(D^*D - \mathrm{O}_0^*\mathrm{O}_0)[u]$ lies in $\overline{\mathcal{H}(k)}^{\perp_0} = \{0\}$. Thus we conclude that

$$\mathrm{A} = D^*D = \mathrm{O}_0^*\mathrm{O}_0. \tag{3.161}$$

Combining this identity with Lemma 3.52, we obtain

$$\mathrm{O}_0\mathrm{A}^{-1} = \mathrm{O}_0\,(\mathrm{O}_0^*\mathrm{O}_0)^{-1} = \mathrm{O}_0\mathrm{O}_0^{-1}\,(\mathrm{O}_0^*)^{-1} = \big(\mathrm{O}_0^{-1}\big)^* \overset{(3.156)}{=} \mathrm{O}_1^*. \tag{3.162}$$

This establishes the result. □

A variation of Proposition 3.51 arises when, instead of knowing a bicontinuous linear operator $D : \big(\mathcal{H}(k), \|\cdot\|_0\big) \to \big(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\big)$, we are directly given $\mathrm{A} = D^*D$. We state this in the following proposition.

Proposition 3.53. *Let $\big(\mathcal{H}(k), \|\cdot\|_0\big)$ be an RKHS on X with reproducing kernel k, and let $A : \big(\mathcal{H}(k), \|\cdot\|_0\big) \to \big(\mathcal{H}(k), \|\cdot\|_0\big)$ be a bicontinuous self-adjoint operator. Consider the associated norm, defined for all $u \in \mathcal{H}(k)$ by*

$$\|u\|_A^2 := \langle Au\,,\, u\rangle_0\,. \tag{3.163}$$

Let us also consider the embedding operator

$$O_0 : \big(\mathcal{H}(k), \|\cdot\|_0\big) \to \big(\mathcal{H}(k), \|\cdot\|_A\big). \tag{3.164}$$

Then $\big(\mathcal{H}(k), \|\cdot\|_A\big)$ is an RKHS on X with reproducing kernel k_A given, for all $x, y \in X$, by

$$k_A(x,y) = \big(O_0A^{-1}\big)[k(x,\cdot)](y) \;\simeq\; A^{-1}[k(x,\cdot)](y). \tag{3.165}$$

Proof. The proof follows the same line as that of Proposition 3.51. Consider the embedding operator $\mathrm{O}_\mathrm{A} : \big(\mathcal{H}(k), \|\cdot\|_\mathrm{A}\big) \to \big(\mathcal{H}(k), \|\cdot\|_0\big)$. By Lemma 3.52, we have

$$\mathrm{O}_0^{-1} = \mathrm{O}_\mathrm{A}, \qquad\qquad \mathrm{O}_\mathrm{A}^{-1} = \mathrm{O}_0. \tag{3.166}$$

Next, observe that $\mathrm{A} = \mathrm{O}_0^*\mathrm{O}_0$. Indeed, for all $u, v \in \mathcal{H}(k)$,

$$\langle \mathrm{O}_0 u\,,\, \mathrm{O}_0 v\rangle_0 = \langle \mathrm{O}_0^*\mathrm{O}_0 u\,,\, v\rangle_0 = \langle \mathrm{A}u\,,\, v\rangle_0\,, \tag{3.167}$$

where the last equality follows from the definition of the A-norm in Equation (3.163). We now proceed exactly as in the proof of Proposition 3.51:

$$\mathrm{O}_0\mathrm{A}^{-1} = \mathrm{O}_0\big(\mathrm{O}_0^*\mathrm{O}_0\big)^{-1} = \mathrm{O}_0\mathrm{O}_0^{-1}\big(\mathrm{O}_0^*\big)^{-1} = \big(\mathrm{O}_0^{-1}\big)^* \overset{(3.166)}{=} \mathrm{O}_\mathrm{A}^*. \tag{3.168}$$

Finally, Proposition 3.48 ensures that the reproducing kernel of $\big(\mathcal{H}(k), \|\cdot\|_\mathrm{A}\big)$ is precisely given by the formula above. □

In line with Proposition 3.51, we are now ready to state what is arguably the most important theorem of this dissertation, or at least the one that serves as a cornerstone.

Theorem 3.54. *The inverse of D is given, for every $f \in \mathcal{H}_1$ and $x \in X$, by the formula*

$$D^{-1}[f](x) = \langle D[k_1(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} = \big\langle D\big[A^{-1}[k(x,\cdot)]\big]\,,\, f\big\rangle_{\mathcal{H}_1}\,, \tag{3.169}$$

with k_1 the reproducing kernel of $\big(\mathcal{H}(k), \|\cdot\|_1\big)$.

Proof. Fix $f \in \mathcal{H}_1$ and $x \in X$. By definition of the inner product in $\mathcal{H}_1$ and the embedding operator, we first compute

$$\begin{aligned}\langle D[k_1(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} &= \big\langle D\big(O_1[k_1(x,\cdot)]\big)\,,\, f\big\rangle_{\mathcal{H}_1} \overset{(3.153)}{=} \langle D_1[k_1(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} \\ &= \langle k_1(x,\cdot)\,,\, D_1^* f\rangle_1\,.\end{aligned} \tag{3.170}$$

Since D_1 is by construction a bijective isometry between $\big(\mathcal{H}(k), \|\cdot\|_1\big)$ and $\big(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1}\big)$, it follows from Remark C.9 in Section C.2.2 of Chapter C, that

$$D_1^* = D_1^{-1}. \tag{3.171}$$

Moreover, recalling the definition of D_1 and the embeddings $\mathrm{O}_0, \mathrm{O}_1$ (see Equation (3.152)), we obtain

$$D_1^{-1} = \mathrm{O}_1^{-1} D^{-1} = \mathrm{O}_0 D^{-1}, \tag{3.172}$$

where the last equality uses the inverse relations in Equation (3.156).

Plugging this expression back into our computation gives

$$\langle D[k_1(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} = \big\langle k_1(x,\cdot)\,,\, \mathrm{O}_0 D^{-1} f\big\rangle_1 = \big(\mathrm{O}_0 D^{-1}\big)\,[f](x) = D^{-1}[f](x), \tag{3.173}$$

where the penultimate equality uses the reproducing property of k_1 in $\big(\mathcal{H}(k), \|\cdot\|_1\big)$, and the last equality comes from the fact that O_0 is *algebraically* the identity.

This proves the first identity in Equation (3.169). The second identity follows directly from Proposition 3.51, which expresses k_1 in terms of $\mathrm{A}^{-1}[k_0]$. □

***Remark* 3.55.** Alternatively, one could start with the second identity by observing that

$$\mathrm{A}^{-1} = (D^* D)^{-1} = D^{-1}\left(D^{-1}\right)^*. \tag{3.174}$$

Hence,

$$\begin{aligned}\left\langle D\big[\mathrm{A}^{-1}[k(x,\cdot)]\big]\,,\,f\right\rangle_{\mathcal{H}_1} &= \left\langle D\Big[D^{-1}\left(D^{-1}\right)^*[k(x,\cdot)]\Big]\,,\,f\right\rangle_{\mathcal{H}_1}\\ &= \left\langle \left(D^{-1}\right)^*[k(x,\cdot)]\,,\,f\right\rangle_{\mathcal{H}_1}\\ &= \left\langle k(x,\cdot)\,,\,D^{-1}f\right\rangle_0 = \left(D^{-1}f\right)(x),\end{aligned} \tag{3.175}$$

where we used the reproducing property of $k(x,\cdot)$ in the last equality. However, this derivation provides less insight into the underlying RKHS structure.

Note that we can immediately reformulate Theorem 3.54 as follows:

Corollary 3.56. *The Green's function associated with the operator D is given by $D[k_1(x,\cdot)]$.*

Proof. This follows directly from Definition 3.4. □

In the next subsection, we generalize those results under weaker assumptions on D.

3.4.3 . Norm induced by a non-bijective operator D

Let us remain within the framework of Section 3.4.2, but assume first that D is no longer surjective, while D^*D remains bicontinuous. Then the norm $\|\cdot\|_1$ defined in Equation (3.151), namely

$$\|u\|_1 := \|D[u]\|_{\mathcal{H}_1}\,, \qquad u \in \mathcal{H}(k), \tag{3.151}$$

is still a well-defined norm on $\mathcal{H}(k)$. Therefore, the whole construction from the previous section remains valid, except for the statement of Theorem 3.54: since D^{-1} no longer exists globally, we can only define $\left(D_{|\operatorname{Im} D}\right)^{-1}$.

One could of course restrict the admissible functions f from $\mathcal{H}_1$ to $\operatorname{Im} D$, but in fact a stronger result holds: the argument extends naturally to the generalized inverse of D.

Theorem 3.57. *Assume that D is not surjective, but that $A := D^*D$ remains bicontinuous. Then D admits a generalized inverse $D^\dagger$, which can be expressed, for every $f \in \mathcal{H}_1$ and $x \in X$, as*

$$D^\dagger[f](x) = \langle D[k_1(x,\cdot)]\,,\,f\rangle_{\mathcal{H}_1} = \left\langle D\big[A^{-1}[k(x,\cdot)]\big]\,,\,f\right\rangle_{\mathcal{H}_1}, \tag{3.176}$$

where k_1 denotes the reproducing kernel of $\left(\mathcal{H}(k), \|\cdot\|_1\right)$.

Proof. As in the proof of Theorem 3.54, the second equality in Equation (3.176) follows directly from Proposition 3.51. It therefore remains to show that the operator defined, for all $f \in \mathcal{H}_1$ and $x \in X$, by

$$D^\dagger[f](x) := \langle D[k_1(x,\cdot)]\,,\,f\rangle_{\mathcal{H}_1}\,, \tag{3.177}$$

is indeed a generalized inverse.

Fix $f \in \mathcal{H}_1$ and $x \in X$. Proceeding as in the proof of Theorem 3.54, we compute

$$\begin{aligned}\langle D[k_1(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} &= \big\langle D\big(O_1[k_1(x,\cdot)]\big)\,,\, f\big\rangle_{\mathcal{H}_1} \overset{(3.153)}{=} \langle D_1[k_1(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} \\ &= \langle k_1(x,\cdot)\,,\, D_1^* f\rangle_1 = D_1^*[f](x),\end{aligned} \tag{3.178}$$

where the last equality follows from the reproducing property of k_1.

Since D_1 is an isometry, Proposition C.8 in Section C.2.2 of Chapter C ensures that $\operatorname{Im} D_1 = \operatorname{Im} D$ is closed in $\mathcal{H}_1$, and that

$$D_1 D_1^* = \Pi_{\operatorname{Im} D}, \qquad\qquad D_1^* D_1 = \mathcal{I}_1, \tag{3.179}$$

where $\mathcal{I}_1$ denotes the identity operator introduced in Equation (3.158). As $D_1^* f \in \big(\mathcal{H}(k), \|\cdot\|_1\big)$, we have

$$D\left[D_1^* f\right] = D[\mathrm{O}_1 D_1^* f] = D_1[D_1^* f] = \Pi_{\operatorname{Im} D} f. \tag{3.180}$$

This shows that $D^\dagger$ is a right-inverse of D.

To prove that $D^\dagger$ is also a left-inverse, note that since D is injective, $\operatorname{Ker} D = \{0\}$ and thus $\operatorname{Ker} D^\perp = \mathcal{H}(k)$. Replacing f by $D_1[u]$, for any $u \in \mathcal{H}(k)$, in Equation (3.178), yields

$$D^\dagger[D[u]](x) = D_1^*\big[D\mathrm{O}_1\mathrm{O}_1^{-1}[u]\big](x) \overset{(3.153)}{=} D_1^*\big[D_1[\mathrm{O}_1^{-1}u]\big](x) \overset{(3.179)}{=} \big(\mathrm{O}_1^{-1}u\big)(x). \tag{3.181}$$

By Lemma 3.52, $\mathrm{O}_1^{-1} = \mathrm{O}_0$. Since O_0 acts as the identity operator at the algebraic level, this simplifies to

$$D^\dagger[D[u]](x) = \mathrm{O}_0[u](x) = u(x). \tag{3.182}$$

Hence $D^\dagger$ is also a left-inverse, which completes the proof. □

Note that Theorem 3.57 can be immediately reformulated as follows:

Corollary 3.58. *The generalized Green's function associated with the operator D is given by $D[k_1(x,\cdot)]$.*

Proof. This follows directly from Definition 3.9. □

Let us now turn to the case where D is no longer injective. In this situation, Equation (3.151) defines only a *pseudo-norm*, since for every $f \in \operatorname{Ker} D$ one has $\|f\|_1 = 0$. However, when restricted to the subspace $\operatorname{Ker} D^\perp \subset \mathcal{H}(k)$, Equation (3.151) becomes a proper norm. More precisely, under the assumption

that $D^{|\operatorname{Im} D}_{\operatorname{Ker} D^\perp}$ is bicontinuous, we can recover the same framework as in Section 3.4.2 by restricting our analysis to $\operatorname{Ker} D^\perp$ and replacing D with $D_{|\operatorname{Ker} D^\perp}$. If D happens to be surjective, this corresponds exactly to the setting of Section 3.4.2, whereas if it is not, we fall back to the framework of the previous subsection.

We conclude this discussion with a result that will prove particularly useful within this context.

Proposition 3.59. *The reproducing kernel k_1 of $\left(\operatorname{Ker} D^\perp, \|\cdot\|_1\right)$ is given, for all $x, y \in X$, by*

$$k_1(x,y) = \left(O_0\, A^\dagger\right)[k(x,\cdot)]\,(y) \;\simeq\; A^\dagger[k(x,\cdot)]\,(y), \tag{3.183}$$

where $A = D^*D$ and $A^\dagger$ denotes its generalized inverse.

Proof. We start from the identity for generalized inverses:

$$\mathrm{A}^\dagger = \left(\mathrm{A}^{|\operatorname{Ker} D^\perp}_{|\operatorname{Ker} D^\perp}\right)^{-1} \Pi_{\operatorname{Ker} D^\perp}, \tag{3.184}$$

which states that $\mathrm{A}^\dagger$ coincides with the inverse of A restricted to $\operatorname{Ker} D^\perp$, composed with the orthogonal projection onto $\operatorname{Ker} D^\perp$.

Recall from Proposition 3.27 that the reproducing kernel of the subspace $\operatorname{Ker} D^\perp$ is given, for all $x, y \in X$, by

$$k_{\operatorname{Ker} D^\perp}(x,y) = \Pi_{\operatorname{Ker} D^\perp}[k(x,\cdot)]\,(y). \tag{3.185}$$

Within this subspace, the operator $\mathrm{A}_{|\operatorname{Ker} D^\perp}$ is invertible. Therefore, applying Proposition 3.51 to the space $\operatorname{Ker} D^\perp$ equipped with this restricted operator yields:

$$k_1(x,y) = \left(\mathrm{O}_0\, \mathrm{A}^{-1}_{|\operatorname{Ker} D^\perp}\right)[k_{\operatorname{Ker} D^\perp}(x,\cdot)]\,(y) = \left(\mathrm{O}_0\, \mathrm{A}^\dagger\right)[k(x,\cdot)]\,(y), \tag{3.186}$$

which proves the claim. □

To prepare the transition to the next subsection, we state the following proposition, which may appear rather innocuous at first glance, but will reveal its full significance in the next section.

Proposition 3.60. *Suppose that $D_{|\operatorname{Ker} D^\perp}$ is an isometry. Then $D^\dagger$ is given, for every $f \in \mathcal{H}_1$ and $x \in X$, by*

$$D^\dagger[f](x) = \langle D[k(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1}\,. \tag{3.187}$$

In particular, for all $x, y \in X$, one has

$$\Pi_{\operatorname{Ker} D^\perp}[k(x,\cdot)]\,(y) = \langle D[k(x,\cdot)]\,,\, D[k(y,\cdot)]\rangle_{\mathcal{H}_1}\,. \tag{3.188}$$

Proof. Since $D_{|\operatorname{Ker} D^\perp}$ is an isometry, we have that $D^\dagger = D^*$. Therefore, for all $x \in X$ and $f \in \mathcal{H}_1$,

$$\langle D[k(x,\cdot)]\,,\, f\rangle_{\mathcal{H}_1} = \langle k(x,\cdot)\,,\, D^* f\rangle_0 = D^\dagger[f](x), \tag{3.189}$$

which proves the first part of the statement.

In particular, by setting $f = D[k(y,\cdot)]$ for any $y \in X$, we obtain

$$\langle D[k(x,\cdot)]\,,\, D[k(y,\cdot)]\rangle_{\mathcal{H}_1} = D^\dagger(D[k(y,\cdot)])\,(x) = \Pi_{\operatorname{Ker} D^\perp}[k(y,\cdot)]\,(x). \tag{3.190}$$

□

The underlying idea of Proposition 3.60 is that, in some cases, it may be more convenient to modify the metric on $\mathcal{H}_1$ rather than on $\mathcal{H}(k)$. This is particularly relevant when $\mathcal{H}_1$ itself is a reproducing kernel Hilbert space. This will be the focus of the next subsection.

3.4.4 . Combined norm modifications between RKHSs

Let us now consider the same framework as in Section 3.4.2, but assume in addition that $\mathcal{H}_1$ is itself a reproducing kernel Hilbert space on a set Y. To avoid confusion, we denote by $\mathcal{H}_0$ the space previously referred to as $\mathcal{H}(k)$, and we rename its associated kernel k as k_0. We further denote by k_1 the reproducing kernel of $\mathcal{H}_1$. **We emphasize that the two underlying sets X and Y are, *a priori*, completely unrelated.**

The key observation is the following identity, valid for all $x \in X$ and $y \in Y$:

$$\langle D[k_0(x,\cdot)]\,,\, k_1(y,\cdot)\rangle_{\mathcal{H}_1} = \langle k_0(x,\cdot)\,,\, D^*[k_1(y,\cdot)]\rangle_{\mathcal{H}_0}\,. \tag{3.191}$$

That is, there exists a natural *duality correspondence* between the kernels k_0 and k_1 mediated by the operator D and its adjoint D^*.

Consequently, we may apply the results of the previous sections in two equivalent ways:

- by acting on $\mathcal{H}_0$ and modifying the kernel k_0 so that D becomes an isometry; or
- by acting on $\mathcal{H}_1$ and modifying the kernel k_1 so that D^* becomes an isometry.

In both cases, making D (or equivalently D^*) an isometry automatically enforces the same property on its adjoint. Hence, the two constructions are perfectly symmetric: modifying the metric on the *input space* $\mathcal{H}_0$ or on the *output space* $\mathcal{H}_1$ leads to an equivalent outcome. We may therefore always choose the more convenient one.

To make this duality more concrete, we now state an alternative version of Theorem 3.54, which we shall reinterpret through the lens of reproducing kernels.

Corollary 3.61. *Consider the setting of Section 3.4.2. Then, the inverse of D is given, for every $f \in \mathcal{H}_1$ and $x \in X$, by the formula*

$$D^{-1}[f](x) = \big\langle A_*^{-1}[D[k(x,\cdot)]] \,,\, f\big\rangle_{\mathcal{H}_1}\,, \tag{3.192}$$

where $A_* = DD^*$.

Proof. Fix $f \in \mathcal{H}_1$ and $x \in X$. Then

$$\begin{aligned}\big\langle \mathrm{A}_*^{-1}[D[k(x,\cdot)]] \,,\, f\big\rangle_{\mathcal{H}_1} &= \big\langle (DD^*)^{-1}[D[k(x,\cdot)]] \,,\, f\big\rangle_{\mathcal{H}_1} \\ &= \big\langle (D^*)^{-1}D^{-1}[D[k(x,\cdot)]] \,,\, f\big\rangle_{\mathcal{H}_1} \\ &= \big\langle (D^{-1})^*[k(x,\cdot)] \,,\, f\big\rangle_{\mathcal{H}_1} \\ &= \big\langle k(x,\cdot) \,,\, D^{-1}[f]\big\rangle_{\mathcal{H}_0} \\ &= D^{-1}[f](x),\end{aligned} \tag{3.193}$$

where the last equality follows from the reproducing property of $k(x,\cdot)$. □

Now, since A_*^{-1} is self-adjoint, observe that Equation (3.192) can be equivalently rewritten as

$$D^{-1}[f](x) = \big\langle D[k(x,\cdot)] \,,\, \mathrm{A}_*^{-1} f\big\rangle_{\mathcal{H}_1}\,. \tag{3.194}$$

Furthermore, recall that $\mathrm{A}_* = DD^* = (D^*)^*D^*$, *i.e.* A_* plays for D^* the same role as A does for D. In particular, by Proposition 3.51, for all $y_1, y_2 \in Y$,

$$\widetilde{k}_1(y_1, y_2) = \mathrm{A}_*^{-1}[k_1(y_1,\cdot)](y_2) \tag{3.195}$$

is the reproducing kernel of $\mathcal{H}_1$ for the norm induced by D^*. However, note that k_1 does not explicitly appear in Equation (3.194). To make it emerge, observe that

$$\mathrm{A}^{-1}[f] = \mathrm{A}^{-1}\big[y \in Y \mapsto \langle k_1(y,\cdot) \,,\, f\rangle_{\mathcal{H}_1}\big] = \big[z \in Y \mapsto \Big\langle \widetilde{k}_1(z,\cdot) \,,\, f\Big\rangle_{\mathcal{H}_1}\big]. \tag{3.196}$$

Hence, Equation (3.194) becomes

$$\begin{aligned} D^{-1}[f](x) &= \left\langle D[k_0(x,\cdot)] \,,\, z \in Y \mapsto \left\langle \widetilde{k}_1(z,\cdot) \,,\, f\right\rangle_{\mathcal{H}_1}\right\rangle_{\mathcal{H}_1} \\ &= \left\langle k_0(x,\cdot) \,,\, D^*\left[z \in Y \mapsto \left\langle \widetilde{k}_1(z,\cdot) \,,\, f\right\rangle_{\mathcal{H}_1}\right]\right\rangle_{\mathcal{H}_1}. \end{aligned} \tag{3.197}$$

To make the connection clearer, set $f = k_1(y,\cdot)$ for some $y \in Y$. By the reproducing property of k_1, we have

$$D^*\left[z \in Y \mapsto \left\langle \widetilde{k}_1(z,\cdot) \,,\, k_1(y,\cdot)\right\rangle_{\mathcal{H}_1}\right] = D^*\Big[z \in Y \mapsto \widetilde{k}_1(z,y)\Big] = D^*\Big[\widetilde{k}_1(y,\cdot)\Big]. \tag{3.198}$$

Then, by another application of Proposition 3.51, we obtain

$$\Big\langle k_0(x,\cdot)\,,\, D^*\Big[\widetilde{k_1}(y,\cdot)\Big]\Big\rangle_{\mathcal{H}_1} = (D^*)^{-1}\,[k_0(x,\cdot)](y), \tag{3.199}$$

and thus the following identity holds:

$$D^{-1}[k_1(y,\cdot)](x) = \Big\langle k_0(x,\cdot)\,,\, D^*\Big[\widetilde{k_1}(y,\cdot)\Big]\Big\rangle_{\mathcal{H}_1} = (D^*)^{-1}\,[k_0(x,\cdot)](y). \tag{3.200}$$

A particularly useful aspect of the above reasoning arises when $f \in \mathcal{H}_1$ can be expressed as

$$f = \sum_{i=1}^{n} \lambda_i\, k_1(y_i,\cdot), \tag{3.201}$$

or, more generally, when f is the limit of such finite sums, an assumption always valid due to the density of $\mathrm{Span}\big(k_1(y,\cdot)\,:\,y\in Y\big)$ in $\mathcal{H}_1$ (*cf.* Proposition 3.24). By linearity, Equation (3.194) becomes

$$\begin{aligned} D^{-1}[f](x) &= \big\langle D[k(x,\cdot)]\,,\, \mathrm{A}_*^{-1} f\big\rangle_{\mathcal{H}_1} = \left\langle D[k(x,\cdot)]\,,\, \mathrm{A}_*^{-1}\sum_{i=1}^{n}\lambda_i k_1(y_i,\cdot)\right\rangle_{\mathcal{H}_1} \\ &= \sum_{i=1}^{n}\lambda_i\,\big\langle D[k(x,\cdot)]\,,\, \mathrm{A}_*^{-1}k_1(y_i,\cdot)\big\rangle_{\mathcal{H}_1}\,. \end{aligned} \tag{3.202}$$

In particular, this shows that we only need to determine the action of A_*^{-1} on the finite-dimensional subspace $\mathrm{Span}\big(k_1(y_i,\cdot)\,:\,1\leqslant i\leqslant n\big)$, a fact of clear practical relevance.

Finally, note that all these considerations extend naturally to the non-bijective case discussed in Section 3.4.3. In particular, if one can compute the kernel $\widetilde{k_1}$ that makes $D^*_{\mathrm{Im}\,D}$ an isometry, then by Proposition 3.60, the projection of the kernel k_0 onto $\mathrm{Ker}\,D^\perp$ can be expressed, for all $x_1, x_2 \in X$, as

$$\Pi_{\mathrm{Ker}\,D^\perp}[k(x_1,\cdot)](x_2) = \left\langle D[k(x_1,\cdot)]\,,\, z\in Y \mapsto \Big\langle \widetilde{k_1}(z,\cdot)\,,\, D[k(x_2,\cdot)]\Big\rangle_{\mathcal{H}_1}\right\rangle_{\mathcal{H}_1}. \tag{3.203}$$

We shall next illustrate these results through one of the most classical and instructive applications of RKHS theory: *interpolation*.

3.5 . Interpolation and Approximation in RKHSs

In this section, we illustrate the results established in Section 3.4.4 through the fundamental example of *interpolation* and *approximation* in a Reproducing Kernel Hilbert Space (RKHS), arguably one of the most classical and motivating applications of the theory.

We deliberately depart from the traditional presentation of (Paulsen and Raghupathi, 2016, Chapter 3), to which we refer the interested reader for further context and complementary insights.

We begin by formalizing the interpolation and approximation problems in RKHSs.

3.5.1 . Formal problem statement

We first introduce general definitions, even though they may not yet be well-posed.

Definition 3.62 (Interpolation)**.** Consider a set X and fix distinct points $(x_i) \in X^n$ together with prescribed values $(\lambda_i) \in \mathbb{R}^n$. We say that a function $g \in \mathcal{F}(X \to \mathbb{R})$ ***interpolates*** (λ_i) on (x_i) if and only if for all $1 \leqslant i \leqslant n$

$$g(x_i) = \lambda_i. \tag{3.204}$$

Definition 3.63 (Optimal approximation)**.** Consider a set X and fix distinct points $(x_i) \in X^n$ together with prescribed values $(\lambda_i) \in \mathbb{R}^n$. Let $\langle \cdot\,, \cdot \rangle_n$ be an inner product on $\mathbb{R}^n$. Given a subset $\mathcal{A} \subset \mathcal{F}(X \to \mathbb{R})$, we say that a function $g \in \mathcal{A}$ ***optimally approximates*** (λ_i) on (x_i) with respect to $\langle \cdot\,, \cdot \rangle_n$ if and only if it minimizes the least-squares error, namely

$$g \in \operatorname*{arg\,min}_{f \in \mathcal{A}} \sum_{i=1}^{n} \|f(x_i) - \lambda_i\|_n^2 \,. \tag{3.205}$$

The corresponding minimization problem is called the ***approximation problem***.

***Remark* 3.64.** Without additional assumptions, the minimum may fail to exist or to be unique. However, we will now see that in the case of Reproducing Kernel Hilbert Spaces (RKHSs), the problem is always well defined and admits a unique solution. More generally, RKHSs provide a particularly natural and robust framework for both interpolation and approximation problems.

Let us fix a set X, distinct points $(x_i) \in X^n$, and an RKHS $\mathcal{H}(k)$. Consider the associated finite subset $F := \{x_i : 1 \leqslant i \leqslant n\}$ and define

$$\iota_F : \begin{cases} \mathcal{H}(k) & \longrightarrow \quad \mathcal{F}(F \to \mathbb{R}) \simeq \mathbb{R}^n \\ f & \longmapsto \quad \big(x_i \mapsto \langle k_{x_i}\,, f \rangle_{\mathcal{H}(k)}\big) \;\simeq\; \big(\langle k_{x_i}\,, f \rangle_{\mathcal{H}(k)}\big)_{1 \leqslant i \leqslant n} \end{cases} . \tag{3.206}$$

Note that ι_F is simply the *algebraic restriction* of the evaluation map ι (introduced in Equation (3.105)) to the finite subset F. More precisely, for every $f \in \mathcal{H}(k)$, we have

$$\iota_F(f) = \iota(f)_{|F}. \tag{3.207}$$

In particular, ι_F is a continuous linear operator.

There is, however, one notable difference: since F is finite, $\mathcal{F}(F \to \mathbb{R})$ is a finite-dimensional vector space of dimension n and can therefore be identified with $\mathbb{R}^n$ via any enumeration of F, such as the one already fixed by $(x_i)_{1 \leqslant i \leqslant n}$. More precisely, to each $x_i \in F$ we associate the function $\delta_{x_i} : F \to \mathbb{R}$ defined by $\delta_{x_i}(x) = \mathbb{1}_{\{x_i = x\}}$, which corresponds to the canonical basis vector e_i of $\mathbb{R}^n$. This identification is precisely what the symbol "$\simeq$" indicates in Equation (3.206).

Under this identification, we can endow $\mathcal{F}(F \to \mathbb{R})$ with an Euclidean inner-product, for instance the canonical inner product: for all $f, g \in \mathcal{F}(F \to \mathbb{R})$,

$$\langle f \, , \, g \rangle_{\mathcal{F}(F \to \mathbb{R})} = \sum_{x \in F} f(x) g(x) = \sum_{i=1}^{n} f(x_i) g(x_i). \tag{3.208}$$

In particular, as already noted in Example 3.21, the space $\mathcal{F}(F \to \mathbb{R})$ is itself an RKHS whose reproducing kernels are the canonical basis functions. More precisely, for every $y \in F$,

$$k_y : x \in F \longmapsto \delta_y(x), \tag{3.209}$$

i.e. the reproducing kernel of $\mathcal{F}(F \to \mathbb{R})$ is simply the identity I_n of $\mathbb{R}^n$. From now on, for notational simplicity, we will use the identification $\mathcal{F}(F \to \mathbb{R}) \simeq \mathbb{R}^n$ and write f_i instead of $f(x_i)$ for all $f \in \mathcal{F}(F \to \mathbb{R})$.

To make the connection with interpolation and approximation explicit, let us highlight the following key observations:

- A function $f \in \mathcal{H}(k)$ *interpolates* the values $\lambda = (\lambda_i) \in \mathbb{R}^n$ at the points $(x_i)_{1 \leqslant i \leqslant n}$ *if and only if*

  $$\iota_F(f) = (\lambda_i)_{1 \leqslant i \leqslant n}, \tag{3.210}$$

  that is, if and only if $\lambda \in \operatorname{Im} \iota_F$. In particular, if ι_F is surjective, then the interpolation problem always admits a solution.

- A function $f \in \mathcal{H}(k)$ *optimally approximates* the values $\lambda = (\lambda_i) \in \mathbb{R}^n$ at the points $(x_i)_{1 \leqslant i \leqslant n}$ *if and only if*

  $$\iota_F(f) \in \operatorname*{arg\,min}_{\alpha \in \operatorname{Im} \iota_F} \|\alpha - \lambda\|_{\mathbb{R}^n}^2 . \tag{3.211}$$

We now establish a series of basic properties of ι_F, which will serve as preliminary steps toward the main results.

Lemma 3.65. *Define*

$$\mathcal{H}_F := \operatorname{Span}\big(k_{x_i} \, : \, 1 \leqslant i \leqslant n\big). \tag{3.212}$$

Then

$$\operatorname{Ker} \iota_F = \mathcal{H}_F^{\perp}. \tag{3.213}$$

Proof. Let $f \in \operatorname{Ker} \iota_F$. Then for any $1 \leqslant i \leqslant n$

$$0 = \iota_F(f)_i = \langle k_{x_i} \, , \, f \rangle_{\mathcal{H}(k)} \, , \tag{3.214}$$

which means that $f \perp k_{x_i}$ for all $1 \leqslant i \leqslant n$, and hence $f \in \mathcal{H}_F^{\perp}$. Conversely, if $f \in \mathcal{H}_F^{\perp}$, then for all $1 \leqslant i \leqslant n$

$$0 = \langle k_{x_i} \, , \, f \rangle_{\mathcal{H}(k)} = \iota_F(f)_i, \tag{3.215}$$

so $f \in \operatorname{Ker} \iota_F$. This proves the identity. □

***Remark* 3.66.** Note first that $\operatorname{Ker} \iota_F$ is closed, and that $\mathcal{H}_F$ has dimension $d := \dim \mathcal{H}_F \leqslant n$, which is also the dimension of $\operatorname{Im} \iota_F$. If $d < n$, then there exist $(\lambda_i) \in \mathbb{R}^n$ that cannot be interpolated by any $f \in \mathcal{H}(k)$, namely those $(\lambda_i) \notin \operatorname{Im} \iota_F$.

To understand this more concretely, suppose that there is a nontrivial linear dependence relation

$$\sum_{j=1}^{n} \alpha_j k_{x_j} = 0, \qquad \text{with some } \alpha_j \neq 0. \tag{3.216}$$

Then for every $f \in \mathcal{H}(k)$ we have

$$0 = \left\langle f \, , \, \sum_{j=1}^{n} \alpha_j k_{x_j} \right\rangle_{\mathcal{H}(k)} = \sum_{j=1}^{n} \alpha_j f(x_j). \tag{3.217}$$

This shows that $\dim \mathcal{H}_F < n$ precisely when the values of every $f \in \mathcal{H}(k)$ at the points (x_j) satisfy some fixed linear relation or equivalently, when the elements of $\operatorname{Im} \iota_F$ satisfy this fixed linear relation.

An immediate consequence of Lemma 3.65 is the following.

Proposition 3.67. *If there exists a function $f \in \mathcal{H}(k)$ that interpolates $(\lambda_i) \in \mathbb{R}^n$ on $(x_i)_{1 \leqslant i \leqslant n}$, then $\Pi_{\mathcal{H}_F}(f)$ is the unique function in $\mathcal{H}(k)$ of minimal norm that interpolates $(\lambda_i)_{1 \leqslant i \leqslant n}$.*

Proof. By Lemma 3.65, we have the orthogonal decomposition

$$\mathcal{H}(k) = \mathcal{H}_F \oplus^{\perp} \operatorname{Ker} \iota_F, \tag{3.218}$$

and the restriction $\iota_{F|\mathcal{H}_F}$ is injective. Let $f \in \mathcal{H}(k)$ interpolate $(\lambda_i)_{1 \leqslant i \leqslant n}$. Then

$$\begin{aligned} (\lambda_i)_{1 \leqslant i \leqslant n} = \iota_F(f) &= \iota_F\big(\Pi_{\mathcal{H}_F}(f) + \Pi_{\operatorname{Ker} \iota_F}(f)\big) \\ &= \iota_F\big(\Pi_{\mathcal{H}_F}(f)\big) + \underbrace{\iota_F\big(\Pi_{\operatorname{Ker} \iota_F}(f)\big)}_{=0}. \end{aligned} \tag{3.219}$$

Since ι_F is injective on $\mathcal{H}_F$, this shows that $\Pi_{\mathcal{H}_F}(f)$ is the unique element of $\mathcal{H}_F$ interpolating $(\lambda_i)_{1 \leqslant i \leqslant n}$.

To prove that $\Pi_{\mathcal{H}_F}(f)$ is the interpolant of minimal norm, let $g \in \mathcal{H}(k)$ be any other interpolant of $(\lambda_i)_{1 \leqslant i \leqslant n}$. By the Pythagora's theorem applied to the decomposition $\mathcal{H}(k) = \mathcal{H}_F \oplus^{\perp} \operatorname{Ker} \iota_F$, we have

$$\|g\|^2_{\mathcal{H}(k)} = \|\Pi_{\mathcal{H}_F} g\|^2_{\mathcal{H}(k)} + \|\Pi_{\operatorname{Ker} \iota_F} g\|^2_{\mathcal{H}(k)} \geqslant \|\Pi_{\mathcal{H}_F} g\|^2_{\mathcal{H}(k)} \,. \tag{3.220}$$

But by injectivity of $\iota_{F|\mathcal{H}_F}$, $\Pi_{\mathcal{H}_F} g$ must coincide with $\Pi_{\mathcal{H}_F} f$. Hence

$$\|g\|_{\mathcal{H}(k)} \geqslant \|\Pi_{\mathcal{H}_F} f\|_{\mathcal{H}(k)} \,, \tag{3.221}$$

with equality if and only if $g = \Pi_{\mathcal{H}_F} f$. This proves the claim. □

To conclude this section, we state a simple but useful correspondence between the surjectivity of ι_F and the injectivity of its adjoint ι_F^*.

Lemma 3.68. *The map ι_F is surjective if and only if its adjoint ι_F^* is injective.*

Proof. Assume first that ι_F is surjective. Then $\operatorname{Im} \iota_F = \mathcal{F}(F \to \mathbb{R}) \simeq \mathbb{R}^n$. Suppose now that $\iota_F^*[h] = 0$ for some $h \in \mathbb{R}^n$. For every $f \in \mathcal{H}(k)$ we have

$$\langle h \,,\, \iota_F(f) \rangle_{\mathbb{R}^n} = \langle \iota_F^* h \,,\, f \rangle_{\mathcal{H}(k)} = 0. \tag{3.222}$$

Since ι_F is surjective, every element of $\mathbb{R}^n$ can be written as $\iota_F(f)$ for some $f \in \mathcal{H}(k)$. Hence $\langle h \,,\, \lambda \rangle_{\mathbb{R}^n} = 0$ for all $\lambda \in \mathbb{R}^n$, which implies $h = 0$. Therefore, ι_F^* is injective.

Conversely, assume that ι_F^* is injective. We know from Proposition C.4 in Section C.2.1 of Chapter C that for any bounded operator T between Hilbert spaces,

$$\overline{\operatorname{Im} T} = (\operatorname{Ker} T^*)^{\perp}. \tag{3.223}$$

Applying this identity to $T = \iota_F$ gives

$$\overline{\operatorname{Im} \iota_F} = (\operatorname{Ker} \iota_F^*)^{\perp}. \tag{3.224}$$

Since ι_F^* is injective, $\operatorname{Ker} \iota_F^* = \{0\}$, and therefore $\overline{\operatorname{Im} \iota_F} = \mathbb{R}^n$. But $\operatorname{Im} \iota_F$ is finite-dimensional, hence already closed, which proves that ι_F is surjective. □

We begin by examining the case where ι_F is surjective, in which interpolation is always possible.

3.5.2 . Interpolation: the case of a surjective ι_F

Assuming that ι_F is surjective, we wish to determine, given $(\lambda_i)_{1 \leqslant i \leqslant n}$, the unique element $f \in \mathcal{H}_F$ that interpolates (λ_i), that is, we seek to invert $\iota_{F|\mathcal{H}_F}$.

Given the theoretical framework developed so far, it is natural to make use of the results established in Theorem 3.54. More precisely, since we have observed that $\mathcal{F}(F \to \mathbb{R})$ is itself an RKHS, we may apply the results of Section 3.4.4 in this finite-dimensional setting.

To do so, we introduce the following operator:

$$\mathcal{S}_F : \begin{cases} \mathbb{R}^n & \longrightarrow & \mathcal{H}(k) \\ (\lambda_i)_{1 \leqslant i \leqslant n} & \longmapsto & \displaystyle\sum_{i=1}^{n} \lambda_i \, k_{x_i} \end{cases} . \tag{3.225}$$

The following key result establishes the link between ι_F and $\mathcal{S}_F$.

Proposition 3.69. *The adjoint of ι_F is precisely the synthesis operator, that is,*

$$\iota_F^* = \mathcal{S}_F. \tag{3.226}$$

Proof. Let $f \in \mathcal{H}(k)$ and $h \in \mathbb{R}^n$. Then:

$$\begin{aligned} \langle h \, , \, \iota_F(f) \rangle_{\mathbb{R}^n} &= \sum_{i=1}^{n} h_i \, \langle k_{x_i} \, , \, f \rangle_{\mathcal{H}(k)} = \left\langle \sum_{i=1}^{n} h_i \, k_{x_i} \, , \, f \right\rangle_{\mathcal{H}(k)} \\ &= \langle \mathcal{S}_F(h) \, , \, f \rangle_{\mathcal{H}(k)} \, . \end{aligned} \tag{3.227}$$

Hence, by definition of the adjoint operator, $\iota_F^* = \mathcal{S}_F$. □

Since ι_F is assumed to be surjective, it follows from Lemma 3.68 that $\mathcal{S}_F = \iota_F^*$ is injective. We can therefore endow $\mathbb{R}^n$ with the inner product that makes $\mathcal{S}_F$ an isometry. According to Proposition 3.51, the reproducing kernel associated with this new inner product is given, for all $1 \leqslant i, j \leqslant n$, by

$$\widetilde{k}_1(i,j) = \mathrm{A}^{-1}[\delta_i](j), \tag{3.228}$$

where $\mathrm{A} = \mathcal{S}_F^* \mathcal{S}_F = \iota_F \mathcal{S}_F$.

To make this expression explicit, note that for any $h \in \mathbb{R}^n$,

$$\begin{aligned} \iota_F \mathcal{S}_F(h) = \iota_F \left(\sum_{i=1}^{n} h_i \, k_{x_i} \right) &= \left(\left\langle k_{x_j} \, , \, \sum_{i=1}^{n} h_i \, k_{x_i} \right\rangle_{\mathcal{H}(k)} \right)_{1 \leqslant j \leqslant n} \\ &= \left(\sum_{i=1}^{n} k(x_j, x_i) \, h_i \right)_{1 \leqslant j \leqslant n} = K_F \, h, \end{aligned} \tag{3.229}$$

where K_F is the *Gram matrix* defined by

$$K_F := \big(k(x_i, x_j) \big)_{1 \leqslant i,j \leqslant n}. \tag{3.230}$$

Substituting this into Equation (3.228) yields, for all $1 \leqslant i, j \leqslant n$,

$$\widetilde{k}_1(i,j) = \big(K_F^{-1} e_i \big)_j = (K_F^{-1})_{i,j}. \tag{3.231}$$

Hence, the reproducing kernel that makes $\mathcal{S}_F$ an isometry is simply the inverse of the Gram matrix K_F. Applying Theorem 3.54, we obtain for every $\lambda = (\lambda_i) \in \mathbb{R}^n$ and $x \in X$:

$$\iota_{F|\mathcal{H}_F}^{-1}(\lambda)(x) = \big\langle \iota_F(k_x) \, , \, K_F^{-1} \lambda \big\rangle_{\mathbb{R}^n} = \sum_{i=1}^{n} k(x_i, x) \, \big(K_F^{-1} \lambda \big)_i. \tag{3.232}$$

This completely answers our interpolation problem.

Proposition 3.70. *The unique element $f_{F,\lambda} \in \mathcal{H}_F$ that interpolates the prescribed values $(\lambda_i)_{1 \leqslant i \leqslant n}$ is given by*

$$f_{F,\lambda} = \sum_{i=1}^{n} k_{x_i} \left(K_F^{-1} \lambda\right)_i. \tag{3.233}$$

Note as a side result that, by Proposition 3.60, we have the following proposition.

Proposition 3.71. *The orthogonal projection onto $\mathcal{H}_F$ is given, for every $f \in \mathcal{H}(k)$, by*

$$\Pi_{\mathcal{H}_F}[f](x) = \left\langle \iota_F[k_x] \, , \, K_F^{-1} \, \iota_F[f] \right\rangle_{\mathbb{R}^n}. \tag{3.234}$$

In particular, the reproducing kernel of $\mathcal{H}_F$ is given, for all $x, y \in X$, by

$$k_{\mathcal{H}_F}(x, y) = \sum_{1 \leqslant i,j \leqslant n} k(x, x_i) \left(K_F^{-1}\right)_{i,j} k(x_j, y). \tag{3.235}$$

Proof. This is a direct application of Proposition 3.60, noting that $\mathcal{H}_F = \operatorname{Ker} \iota_F^{\perp}$ and that, by construction, the inner product induced by the kernel represented by K_F^{-1} makes ι_F an isometry.

The second statement follows immediately from Proposition 3.27, which gives the explicit expression of the reproducing kernel associated with a closed subspace. □

Finally, note that if we endow $\mathbb{R}^n$ with a different inner product, the interpolation result remains unchanged. To verify this, consider a new inner product $\langle \cdot \, , \, \cdot \rangle_G$ on $\mathbb{R}^n$, defined by a symmetric positive definite matrix G. The only modification in our construction is that the reproducing kernel of $\mathbb{R}^n$ is no longer the identity I_n, but rather G^{-1}, as discussed in Example 3.21, and that the adjoint ι_F^* now differs from $\mathcal{S}_F$. To make this explicit, for any $f \in \mathcal{H}(k)$ and $h \in \mathbb{R}^n$, we have

$$\begin{aligned} \langle \iota_F(f) \, , \, h \rangle_G &= \sum_{i=1}^{n} \left(\langle k_{x_i} \, , \, f \rangle_{\mathcal{H}(k)} \right) (Gh)_i \\ &= \left\langle \sum_{i=1}^{n} (Gh)_i \, k_{x_i} \, , \, f \right\rangle_{\mathcal{H}(k)}. \end{aligned} \tag{3.236}$$

Hence the adjoint operator ι_F^* takes the form:

$$\iota_F^* : \begin{cases} \mathbb{R}^n & \longrightarrow \quad \mathcal{H}(k) \\ (\lambda_i)_{1 \leqslant i \leqslant n} & \longmapsto \quad \displaystyle\sum_{i=1}^{n} (G\lambda)_i \, k_{x_i} \end{cases}. \tag{3.237}$$

For any $h \in \mathbb{R}^n$, we thus obtain

$$\iota_F\big(\iota_F^*(h)\big) = K_F \, G. \tag{3.238}$$

Note that $K_F G$ is self-adjoint with respect to the inner product $\langle \cdot\,,\,\cdot \rangle_G$. Applying Theorem 3.54, we therefore obtain, for every $\lambda = (\lambda_i) \in \mathbb{R}^n$ and $x \in X$:

$$\begin{aligned} \iota_{F|\mathcal{H}_F}^{-1}(\lambda)(x) = \left\langle \iota_F(k_x)\,,\, G^{-1}K_F^{-1}\lambda \right\rangle_G &= \sum_{1 \leqslant i,j \leqslant n} k(x_i, x)\, G_{i,j} \left(G^{-1}K_F^{-1}\lambda\right)_j \\ &= \sum_{i=1}^{n} k(x_i, x) \left(GG^{-1}K_F^{-1}\lambda\right)_i = \sum_{i=1}^{n} k(x_i, x) \left(K_F^{-1}\lambda\right)_i, \end{aligned} \tag{3.239}$$

which indeed yields the same expression as in Proposition 3.70.

***Remark* 3.72.** The underlying intuition is that, as long as we are in a purely interpolating regime, *i.e.* when ι_F is surjective, the choice of metric on $\mathbb{R}^n$ does not affect the interpolant. In contrast, when ι_F is not surjective, the situation changes: different metrics induce different approximation schemes, and hence different "best" interpolants.

To conclude this subsection, we provide a condition on the kernel k ensuring that ι_F is always surjective.

Proposition 3.73. *If k is symmetric positive definite in the sense of Definition 3.33, then ι_F is surjective for any choice of F.*

Proof. By Lemma 3.68, ι_F is surjective if and only if $\iota_F \iota_F^*$ is bijective. Moreover, from Equations (3.229) and (3.238), this bijectivity is equivalent to the invertibility of the Gram matrix K_F. Finally, by Remark 3.34 stated just after Definition 3.33, if k is symmetric positive definite, then every Gram matrix K_F is symmetric positive definite, and therefore invertible. This proves the claim. □

We now turn to the more subtle case where ι_F is not surjective, and interpolation must be replaced by approximation.

3.5.3 . Approximation: the case of a non-surjective ι_F

When ι_F is not surjective, directly inverting it, even when restricted to $\operatorname{Ker} \iota_F^{\perp}$, is impossible. Given $\lambda \in \mathbb{R}^n$, the best we can do is to solve the *minimization problem*

$$f_\star = \operatorname*{arg\,min}_{f \in \mathcal{H}_F} \|\iota_F(f) - \lambda\|_{\mathbb{R}^n}\,. \tag{3.240}$$

However, the key point is that the solution now depends on the choice of the metric on $\mathbb{R}^n$. This is precisely what is known as the *approximation problem*.

Equivalently, by the properties of orthogonal projections, we know that the best approximation of λ in $\operatorname{Im} \iota_F$ is given by $\Pi_{\operatorname{Im} \iota_F}(\lambda)$. Hence, the optimal procedure consists in first projecting λ onto $\operatorname{Im} \iota_F$ and then applying the inverse on this subspace, *i.e.* computing the *generalized inverse*. The crucial

observation is that this projection, and therefore the resulting approximation, depends on the chosen metric.

Indeed, by Proposition C.4 in Section C.2.1 of Chapter C, and since $\operatorname{Im} \iota_F$ is closed (as $\mathbb{R}^n$ is finite-dimensional), we have the orthogonal decomposition

$$\mathbb{R}^n = \operatorname{Im} \iota_F \oplus^{\perp} \operatorname{Ker} \iota_F^*. \tag{3.241}$$

Moreover, the adjoint ι_F^* depends on the metrics of both $\mathcal{H}(k)$ and $\mathbb{R}^n$. Consequently, the subspace $\operatorname{Ker} \iota_F^*$, and therefore the notion of "negligible" components, changes with the metric. Intuitively, the metric determines which parts of $\lambda \in \mathbb{R}^n$ are deemed informative and which should be disregarded.

Let us first consider the simplest case, where $\mathbb{R}^n$ is endowed with its canonical Euclidean inner product. Then, if ι_F is not surjective, Lemma 3.68 implies that ι_F^* is not injective. Consequently, we can only define a kernel on the subspace $(\operatorname{Ker} \iota_F^*)^{\perp} \subset \mathbb{R}^n$, such that $\iota^*_{F|(\operatorname{Ker} \iota_F^*)^{\perp}}$ becomes an isometry.

By applying Proposition 3.59, we obtain, for all $1 \leqslant i, j \leqslant n$,

$$\widetilde{k}_{1|(\operatorname{Ker} \iota_F^*)^{\perp}}(i,j) = \left(K_F^{\dagger} e_i\right)_j = (K_F^{\dagger})_{i,j}, \tag{3.242}$$

which is the generalized counterpart of Equation (3.231). Moreover, by Proposition C.4 (again in Section C.2.1 of Chapter C),

$$(\operatorname{Ker} \iota_F^*)^{\perp} = \operatorname{Im} \iota_F, \tag{3.243}$$

so that

$$\iota_F^{|(\operatorname{Ker} \iota_F^*)^{\perp}\dagger} = \iota_F^{|\operatorname{Im} \iota_F\dagger} = \iota_F^{\dagger}. \tag{3.244}$$

Then, applying Proposition 3.60, we find that for all $h \in \mathbb{R}^n$ and $x \in X$,

$$\iota_F^{\dagger}(h)(x) = \left\langle \iota_F[k(x,\cdot)] \,,\, \mathrm{A}^{\dagger} h \right\rangle_{\mathbb{R}^n} = \sum_{i=1}^{n} \left(\mathrm{A}^{\dagger} h\right)_i k_{x_i}(x). \tag{3.245}$$

We thus obtain the solution to the approximation problem.

Proposition 3.74. *The unique element $f_{F,\lambda} \in \mathcal{H}_F$ that best approximates the prescribed values $(\lambda_i)_{1 \leqslant i \leqslant n}$ in the least-squares sense, i.e. the minimizer*

$$f_\star = \operatorname*{arg\,min}_{f \in \mathcal{H}_F} \left\| \iota_F(f) - \lambda \right\|_{\mathbb{R}^n}, \tag{3.246}$$

is given explicitly by

$$f_{F,\lambda} = \sum_{i=1}^{n} k_{x_i} \left(K_F^{\dagger} \lambda\right)_i. \tag{3.247}$$

In particular, we can also generalize Proposition 3.71 to this setting.

Proposition 3.75. *The orthogonal projection onto $\mathcal{H}_F$ is given, for every $f \in \mathcal{H}(k)$, by*

$$\Pi_{\mathcal{H}_F}[f](x) = \left\langle \iota_F[k_x]\,,\, K_F^\dagger\, \iota_F[f] \right\rangle_{\mathbb{R}^n}. \tag{3.248}$$

In particular, the reproducing kernel of $\mathcal{H}_F$ is given, for all $x, y \in X$, by

$$k_{\mathcal{H}_F}(x,y) = \sum_{1 \leqslant i,j \leqslant n} k(x,x_i)\left(K_F^\dagger\right)_{i,j} k(x_j,y). \tag{3.249}$$

Sketch of proof. The proof follows exactly the same reasoning as in Proposition 3.71, but with the generalized inverse $K_F^\dagger$ replacing K_F^{-1}. In particular, we use that $\iota_F^\dagger = \mathcal{S}_F K_F^\dagger$ and that the projection onto $\mathcal{H}_F = \operatorname{Ker} \iota_F^\perp$ is given by

$$\Pi_{\mathcal{H}_F} = \iota_F^\dagger \iota_F = \mathcal{S}_F K_F^\dagger \iota_F. \tag{3.250}$$

Expanding this expression and applying it to $f \in \mathcal{H}(k)$ immediately yields the desired formula for $\Pi_{\mathcal{H}_F}[f](x)$. The corresponding kernel expression then follows by applying Proposition 3.27. □

Now, if we modify the metric on $\mathbb{R}^n$ to the one induced by a positive definite matrix G, we recall from Equation (3.238) that

$$\iota_F\big(\iota_F^*(h)\big) = K_F\, G. \tag{3.238}$$

To compute the generalized inverse, observe that for all $1 \leqslant i \leqslant n$, the vectors $v_i = G^{-\frac{1}{2}} e_i$ form an orthonormal basis of $\mathbb{R}^n$ for the inner product $\langle \cdot\,,\,\cdot \rangle_G$. Using the standard change-of-basis formula, the matrix of $\iota_F \iota_F^*$ in this orthonormal basis is

$$G^{\frac{1}{2}}\,(K_F G)\,G^{-\frac{1}{2}} = G^{\frac{1}{2}} K_F G^{\frac{1}{2}}. \tag{3.251}$$

Hence, in this basis, the generalized inverse of $\iota_F \iota_F^*$ is $\left(G^{\frac{1}{2}} K_F G^{\frac{1}{2}}\right)^\dagger$. Applying the inverse change of basis, the corresponding matrix in the canonical basis (e_i) is

$$G^{-\frac{1}{2}} \left(G^{\frac{1}{2}} K_F G^{\frac{1}{2}}\right)^\dagger G^{\frac{1}{2}}. \tag{3.252}$$

Using again Proposition 3.60, we obtain, for every $h \in \mathbb{R}^n$ and $x \in X$,

$$\begin{aligned}
\iota_F^\dagger(h)(x) &= \left\langle \iota_F[k(x,\cdot)]\,,\, G^{-\frac{1}{2}} \left(G^{\frac{1}{2}} K_F G^{\frac{1}{2}}\right)^\dagger G^{\frac{1}{2}} h \right\rangle_G \\
&= \sum_{1 \leqslant i,j \leqslant n} k(x_i,x)\, G_{i,j} \left(G^{-\frac{1}{2}} \left(G^{\frac{1}{2}} K_F G^{\frac{1}{2}}\right)^\dagger G^{\frac{1}{2}} h\right)_j \\
&= \sum_{i=1}^n \left(G^{\frac{1}{2}} \left(G^{\frac{1}{2}} K_F G^{\frac{1}{2}}\right)^\dagger G^{\frac{1}{2}} h\right)_i k_{x_i}(x).
\end{aligned} \tag{3.253}$$

***Remark* 3.76.** It is worth stressing that, in general,

$$\left(G^{\frac{1}{2}} K_F G^{\frac{1}{2}}\right)^{\dagger} \neq G^{-\frac{1}{2}} K_F^{\dagger} G^{-\frac{1}{2}}, \tag{3.254}$$

unless G and K_F are simultaneously diagonalizable (for instance, when they commute).

Before concluding this chapter with the generalization proposed by Schwartz, which paves the way for the most abstract formulation of RKHS theory, let us note that all the results established so far extend naturally to vector-valued functions, through the simple identification:

$$\mathcal{V} : \begin{cases} \mathcal{F}(X \to \mathbb{R}^n) & \longrightarrow & \mathcal{F}(X \times \{1, \dots, n\} \to \mathbb{R}) \\ f & \longmapsto & \Big((x, i) \in X \times \{1, \dots, n\} \mapsto f(x)_i\Big) \end{cases} . \tag{3.255}$$

3.6 . Beyond RKHS: Schwartz's Generalization of Reproducing Kernels

To conclude this chapter, we briefly mention the generalization of the theory of reproducing kernels due to Laurent Schwartz (Schwartz, 1964). This extension provides a unifying framework that encompasses the classical Hilbert-space setting as a particular case, while allowing for kernels that act between *dual pairs* of topological vector spaces.

3.6.1 . Schwartz's Framework

Let E be a locally convex topological vector space over $\mathbb{R}$, and let E' denote its topological dual. We denote by $\langle \cdot\,, \cdot \rangle_{E',E}$ the canonical duality pairing between E' and E. Schwartz then introduces two central notions.

Definition 3.77 (Hilbert subspace of E')**.** A Hilbert space $\mathcal{H}$ is said to be a ***Hilbert subspace of*** E' if there exists an (injective and continuous) embedding

$$\iota : \mathcal{H} \longrightarrow E'. \tag{3.256}$$

***Remark* 3.78.** Note in particular that the topology of $\mathcal{H}$ is stronger than the topology induced by E' through ι.

Definition 3.79 (Schwartz kernel)**.** A ***kernel relative to*** E is a continuous linear operator

$$\mathcal{K} : E \longrightarrow E'. \tag{3.257}$$

It is said to be:

- ***symmetric*** if, for all $\phi, \psi \in E$,

$$\langle \mathcal{K}(\phi)\,,\, \psi \rangle_{E',E} = \langle \mathcal{K}(\psi)\,,\, \phi \rangle_{E',E}\,; \tag{3.258}$$

- ***non-negative*** if, for all $\phi \in E$,

$$\langle \mathcal{K}(\phi)\,,\,\phi\rangle_{E',E} \geqslant 0. \tag{3.259}$$

A kernel relative to E that is both symmetric and non-negative is called a ***Schwartz kernel*** relative to E.

One of Schwartz's central results establishes an isomorphism between the set of Schwartz kernels relative to E and the Hilbert subspaces of E'. This correspondence is made explicit through the following pair of dual constructions:

- **From a Schwartz kernel to a Hilbert subspace.** Given a Schwartz kernel $\mathcal{K}$, define on $\operatorname{Im}\mathcal{K}$ the pre-Hilbertian inner product

$$\langle \mathcal{K}(\phi)\,,\,\mathcal{K}(\psi)\rangle_{\mathcal{K}} := \langle \mathcal{K}(\phi)\,,\,\psi\rangle_{E',E}\,, \qquad \text{for all } \phi,\psi \in E, \tag{3.260}$$

  and complete $\operatorname{Im}\mathcal{K}$ with respect to this inner product. Since $\mathcal{K}$ is continuous, the topology induced by $\langle\cdot\,,\,\cdot\rangle_{\mathcal{K}}$ is stronger than that of E', and hence the completion of $\operatorname{Im}\mathcal{K}$ can be mapped *algebraically* back into E'. This makes it a genuine Hilbert subspace of E', in the same spirit as the explicit construction of ι in the Aronszajn–Moore framework (see Section 3.3.2), where the completion was mapped back into $\mathcal{F}(X \to \mathbb{R})$.

- **From a Hilbert subspace to a Schwartz kernel.** Conversely, starting from a Hilbert subspace $\mathcal{H}$ of E' with embedding

$$\iota : \mathcal{H} \longrightarrow E', \tag{3.261}$$

  one defines the associated Schwartz kernel by

$$\mathcal{K} := \iota\,\iota^*. \tag{3.262}$$

  where the adjoint ι^* is taken with respect to the pair of dualities (E', E) and $(\mathcal{H}, \mathcal{H})$. Since ι is injective, it follows from Proposition C.4 in Section C.2.1 of Chapter C that $\operatorname{Im}\iota^*$ is dense in $\mathcal{H}$. This pre-Hilbertian space thus plays the same role as $\operatorname{Im}\mathcal{K}$ in the previous construction, and as the span $\operatorname{Span}\big(k_x : x \in X\big)$ in the classical RKHS framework.

One of the advantages of this generalization is that it simultaneously encompasses both RKHSs and Hilbert riggings.

3.6.2 . Recovering Moore–Aronszajn construction from Schwartz's Framework

To see how the classical RKHS theory fits into this broader framework, let us consider $E' = \mathcal{F}(X \to \mathbb{R})$, the space of real-valued functions on X

endowed with the topology of pointwise convergence.[4] It can be shown (see for instance Treves (2016)) that its topological dual E is the vector space generated by pointwise evaluation functionals (Dirac's distribution), *i.e.* finite linear combinations of such functionals, with the duality pairing characterized for all $x \in X$ and $f \in \mathcal{F}(X \to \mathbb{R})$ by

$$\langle f \,,\, \delta_x \rangle_{E',E} := f(x). \tag{3.263}$$

A Schwartz kernel $\mathcal{K}$ in this setting is a continuous operator $\mathcal{K} : E \to E'$, completely determined by its values on the Dirac evaluations δ_x. For each $x \in X$, we may thus define

$$k_x := \mathcal{K}(\delta_x) \in E' = \mathcal{F}(X \to \mathbb{R}), \tag{3.264}$$

and for all $x, y \in X$,

$$\langle \mathcal{K}(\delta_x) \,,\, \delta_y \rangle_{E',E} = \mathcal{K}(\delta_x)(y) = k_x(y). \tag{3.265}$$

Moreover, the symmetry and non-negativity of $\mathcal{K}$ ensure that, for all $x, y \in X$,

$$k_x(y) = k_y(x)\,; \qquad\qquad k_x(x) \geqslant 0. \tag{3.266}$$

Hence, we recover precisely the Moore–Aronszajn construction of reproducing kernel Hilbert spaces within Schwartz's generalized framework.

3.6.3 . Recovering Hilbert Rigging from Schwartz's Framework

Let us now show how the framework of *Hilbert riggings* (Berezansky et al., 1996b, Chapter 14) naturally connects with Schwartz's construction.

Recall that a Hilbert rigging (or *Gelfand triple*) consists of a chain of continuous, injective, and dense embeddings:

$$\mathrm{H}_+ \xrightarrow{\mathrm{O}_+} \mathrm{H}_0 \xrightarrow{\mathrm{O}_-} \mathrm{H}_-, \tag{3.267}$$

where H_+, H_0, and H_- are Hilbert spaces. These embeddings satisfy the duality relation

$$\langle \alpha \,,\, u \rangle_{\mathrm{H}_-,\mathrm{H}_+} = \langle \mathbb{I}_0(\alpha) \,,\, u \rangle_{\mathrm{H}_+}\,, \quad \text{for all } (\alpha, u) \in \mathrm{H}_- \times \mathrm{H}_+. \tag{3.268}$$

Interpreting this setting within Schwartz's formalism, we set

$$\iota := \mathrm{O}_-. \tag{3.269}$$

[4] Since Schwartz's theory is formulated in the language of dual pairs, the spaces E and E' play interchangeable roles. Hence, one may equivalently start by defining E' and then take E as its dual. See Mary (2003) for further details.

Then, for every $f \in \mathrm{H}_0$ and $u \in \mathrm{H}_+$, we have

$$\langle \mathrm{O}_-(f)\,,\, u\rangle_{\mathrm{H}_-,\mathrm{H}_+} = \langle \mathbb{I}_0 \mathrm{O}_-(f)\,,\, u\rangle_{\mathrm{H}_+} = \langle \mathrm{I}_+(f)\,,\, u\rangle_{\mathrm{H}_+} = \langle f\,,\, \mathrm{O}_+(u)\rangle_{\mathrm{H}_0}\,. \tag{3.270}$$

Hence $\iota^* = \mathrm{O}_+$, and the associated Schwartz kernel is

$$\mathcal{K} := \mathrm{O}_-\,\mathrm{O}_+. \tag{3.271}$$

The corresponding pre-Hilbert space is then

$$\mathrm{Im}\,\mathrm{O}_+ \subset \mathrm{H}_0, \tag{3.272}$$

and its completion is precisely H_0, viewed through the embedding $\iota = \mathrm{O}_-$ as a Hilbert subspace of H_-.

Even more interestingly, if we extend the rigging beyond the triple Equation (3.267), defining the limit spaces

$$\mathrm{H}_{+\infty} := \bigcap_{m\geqslant 0} \mathrm{H}_m, \qquad \mathrm{H}_{-\infty} := \bigcup_{m\geqslant 0} \mathrm{H}_{-m}, \tag{3.273}$$

we can consider the set of Schwartz kernels from $\mathrm{H}_{+\infty}$ to $\mathrm{H}_{-\infty}$ and their corresponding sub-Hilbertian spaces. Among these, though by no means exclusively, lie all the intermediate Hilbert spaces H_m generated by the initial rigging

$$\mathrm{H}_+ \xrightarrow{\mathrm{O}_+} \mathrm{H}_0 \xrightarrow{\mathrm{O}_-} \mathrm{H}_-. \tag{3.274}$$

As an illustrative example, let us consider the Sobolev rigging chain

$$\mathrm{H}^1_0(\Omega) \xrightarrow{\mathrm{O}_+} \mathrm{L}^2(\Omega) \xrightarrow{\mathrm{O}_-} \mathrm{H}^{-1}(\Omega), \tag{3.275}$$

whose limiting spaces are $\mathrm{H}_{+\infty} = \mathcal{D}(\Omega) = \mathcal{C}^\infty_c(\Omega)$ and $\mathrm{H}_{-\infty} = \mathcal{D}'(\Omega)$, the space of distributions. Among all the Hilbert subspaces generated by Schwartz kernels between these two extremal spaces, we recover all the generalized Sobolev spaces. In particular, the canonical embedding

$$T : \mathcal{D}(\Omega) \longrightarrow \mathcal{D}'(\Omega) \tag{3.276}$$

is itself a Schwartz kernel that generates the $\mathrm{L}^2(\Omega)$ space as its associated sub-Hilbertian structure.

Consequently, one possible way to study partial differential equations within the space of distributions is to regard them as operators acting between Hilbert subspaces of $\mathcal{D}'(\Omega)$ generated by Schwartz kernels. In particular, since these Hilbert subspaces can be seen as "generalized RKHSs", this correspondence enables us to apply the same line of reasoning as in Section 3.4.4, thereby unifying both frameworks through a common geometric principle: the modification of

metrics on embedded Hilbert structures. This viewpoint not only extends the reproducing kernel framework to the setting of distributions but also provides a natural bridge to the theory of Green's functions (see Duffy (2015)) and to the broader class of *partial inner product spaces* studied in Antoine and Trapani (2009).

This perspective closes the loop between the geometric view of RKHSs, the operator-theoretic structure of rigged Hilbert spaces, and the distributional framework introduced by Schwartz. It reveals that the notion of a reproducing kernel extends far beyond the realm of functions and can be understood as a universal representation of inner products between linear forms.

In the next part, where we begin to present our own contributions, we shall return to more applied considerations, since our goal is to bring this theory into the realm of machine learning, and therefore, to design algorithms that can actually run on computers ! Nevertheless, we will not entirely depart from this theoretical framework. In fact, much like in Example 3.22, we will focus on spaces that possess *both* structures simultaneously: an RKHS structure in the sense of Moore–Aronszajn, and a Schwartz-type structure in which the spaces under consideration are Hilbert subspaces of spaces of distributions.

Part II

Contributions

4 - ANaGRAM: A Natural Gradient Relative to Adapted Model for efficient PINNs learning

This chapter presents the results developed in our paper Schwencke and Furtlehner (2025).

In this paper we propose a natural gradient approach to PINNs which contributes to speed-up and improve the accuracy of the training. Based on an in depth analysis of the differential geometric structures of the problem, we come up with the following distinct contributions:

(i) We highlight a principled mathematical framework that restates natural gradient in an equivalent, yet simpler way, leading us to propose ANaGRAM, a general-purpose natural gradient algorithm of reduced complexity $\mathcal{O}(\min(P^2S, S^2P))$ compared to $\mathcal{O}(P^3)$, where $P = \#parameters$ and $S = \#batch\ samples$.

(ii) We reinterpret the PINNs framework from a functional analysis perspective in order to extend ANaGRAM to the PINN's context in a straightforward manner.

(iii) We establish a direct correspondence between ANaGRAM for PINNs and the Green's function of the operator on the tangent space.

The rest of this article is organized as follows: in Section 4.1, after introducing neural networks and parametric models in Section 4.1.1 from a functional analysis perspective, we review two concepts crucial to our work: PINNs framework in Section 4.1.2, and natural gradient in Section 4.1.3. In Section 4.2, we introduce the notions of empirical tangent space and an expression for the corresponding notion of empirical natural gradient leading to ANaGRAM, described in algorithm 2. In Section 4.3, after reinterpreting PINNs as a regression problem from the right functional perspective in Section 4.3.1, yielding ANaGRAM algorithm 3 for PINNs, we state in Section 4.4 that natural gradient matches the Green's function of the operator on the tangent space and analyse the consequence of this on the interpretation of PINNs training process under ANaGRAM. Finally, in Section 4.5, we show empirical evidences of the performance of ANaGRAM on a selected benchmark of PDEs.

4.1 . Position of the problem

4.1.1 . Parametric models and Neural Networks

We start by recalling the objects we introduced in Sections 2.1.1 and 2.1.2 of Chapter 2.

Parametric model: A differentiable mapping $u : \mathbb{R}^P \to \mathcal{H}$, as defined in Definition 2.8.

Differential of u: The differential of u at θ, denoted $\mathrm{d}u_\theta$, as defined in Definition 2.10.

In particular, following the manifold approximation perspective introduced in Section 2.1.2, we define

The image set of u : this is the set of functions reached by u, *i.e.* :

$$\mathcal{M} := \operatorname{Im} u := \left\{ u_\theta \,:\, \theta \in \mathbb{R}^P \right\} \subset \mathcal{H}. \tag{4.1}$$

The tangent space of u at θ : this is the image set of the differential of u at θ, *i.e.* the linear subspace of $\mathcal{H}$ consisting of the functions reached by $\mathrm{d}u_\theta$, *i.e.* :

$$T_\theta \mathcal{M} := \operatorname{Im} \mathrm{d}u_\theta = \operatorname{Span}\left(\partial_p u_\theta \,:\, 1 \leqslant p \leqslant P \right) \subset \mathcal{H}. \tag{4.2}$$

4.1.2 . Physics Informed Neural Networks (PINNs)

As in Section 2.1.6 of Chapter 2, we consider a domain $\Omega \subset \mathbb{R}^n$ endowed with a probability measure μ, with boundary $\partial\Omega$ equipped with a probability surface measure σ. We then introduce two differentiable differential operators:

$$D : \left\{ \begin{array}{ccc} \mathcal{H} & \to & \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \\ u & \mapsto & D[u] \end{array} \right. ; \quad B : \left\{ \begin{array}{ccc} \mathcal{H} & \to & \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma) \\ u & \mapsto & B[u] \end{array} \right. , \tag{4.3}$$

where $\mathcal{H}$ is a Hilbert space of functions from $\overline{\Omega}$ to $\mathbb{K}^m$, on which D and B are defined (see Berezansky et al. 1996b, Chapter 12). We can then define the associated PDE:

$$\begin{cases} D(u) = f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) & \text{in } \Omega \\ B(u) = g \in \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma) & \text{on } \partial\Omega \end{cases} . \tag{4.4}$$

4.1.3 . Natural Gradient

Let us consider a functional loss $\mathcal{L} : \mathcal{H} \to \mathbb{R}_+$ and a parametric model $u : \mathbb{R}^P \to \mathcal{H}$. By Proposition 2.37 in Section 2.1.5 of Chapter 2, the parametric natural gradient flow associated with $\mathcal{L}$ (omitting the initial condition) is given by:

$$\frac{\mathrm{d}}{\mathrm{dt}}\theta(\mathrm{t}) = \mathrm{du}^\dagger_{\theta_\mathrm{t}} \left(-\Pi_{\mathrm{T}_{\theta_\mathrm{t}}\mathcal{M}} \left(\nabla \mathcal{L}_{\mathrm{u}_{\theta_\mathrm{t}}} \right) \right). \tag{4.5}$$

Applying the approximation scheme introduced in Definition 2.48 of Section 2.2.4 in Chapter 2, with a learning rate $\eta > 0$, we obtain the natural gradient descent:

$$\theta_{t+1} \leftarrow \theta_t - \eta\, \mathrm{d}u^\dagger_{\theta_t} \left(\Pi_{T_{\theta_t}\mathcal{M}} \left(\nabla \mathcal{L}_{u_{\theta_t}} \right) \right). \tag{4.6}$$

4.2 . Empirical Natural Gradient and ANaGRAM algorithm

In this section, we introduce the first contribution of this work.

4.2.1 . Empirical Natural Gradient

Let us consider an $\mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$ regression setting, as presented in Example 2.17 of Section 2.1.3 in Chapter 2, with the associated loss:

$$\mathcal{L}_2 : \left\{ \begin{array}{ccc} \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) & \longrightarrow & \mathbb{R}_+ \\ v & \longmapsto & \dfrac{1}{2} \left\| v - f \right\|^2_{\mathrm{L}^2(\Omega \to \mathbb{R}, \mu)} \end{array} \right. . \tag{4.7}$$

As established in Example 2.21, the functional gradient is then given by $\nabla \mathcal{L}_v = v - f$.

Now, fixing a batch $(x_i)_{i=1}^S$ and considering the associated empirical MSE introduced in Definition 2.45, namely:

$$\mathcal{L}_{2,X}(v) := \frac{\mu_n(\Omega)}{2\,S} \sum_{i=1}^{S} \left\| v(x_i) - f(x_i) \right\|^2_{\mathbb{R}^m} , \tag{2.73}$$

we have shown in Section 2.2.3 that the functional natural gradient associated with $\mathcal{L}_{2,X}$ and the parametric model u is given by Equation (2.96), that is:

$$\nabla_{\rho(\theta)} \mathcal{L}_{2,X} = \frac{\mu_n(\Omega)}{S} \sum_{i=1}^{S} \left\langle f(x_i) - v(x_i) \,,\, \mathrm{NNTK}_\theta(x_i, \cdot) \right\rangle_{\mathbb{R}^m} . \tag{2.96}$$

In particular, we established (see Equation (2.98)) that the natural gradient descent update evolves within a subspace of the tangent space, for which we now provide a precise definition.

Definition 4.1 (empirical tangent space)**.** Given a parametric model $u : \mathbb{R}^P \to \mathcal{H}$, and a batch of points $(x_i)_{1 \leqslant i \leqslant S}$, the **empirical tangent space relative to the points** $(x_i)_{1 \leqslant i \leqslant S}$, is the space:

$$\widehat{T}^{NNTK}_{\theta,(x_i)} \mathcal{M} := \mathrm{Span}\left(NNTK_\theta(\cdot, x_i) \,:\, (x_i)_{1 \leqslant i \leqslant S} \right) \subset T_\theta \mathcal{M}. \tag{4.8}$$

When the context is clear, its name will be abbreviated **empirical tangent space** and it will be denoted:

$$\widehat{T}_\theta \mathcal{M} := \mathrm{Span}\left(NNTK_\theta(\cdot, x_i) \,:\, (x_i)_{1 \leqslant i \leqslant S} \right) \tag{4.9}$$

Building on this observation, and by analogy with Equation (4.6), we now define a new natural gradient update.

Definition 4.2 (empirical Natural Gradient (eNG))**.** The **empirical Natural Gradient (eNG)** is the update given by the projection of the functional gradient $\nabla \mathcal{L}_{u_{\theta_t}}$ on the empirical tangent space $\widehat{T}_\theta \mathcal{M}$, *i.e.* :

$$\theta_{t+1} \leftarrow \theta_t - \eta \, \mathrm{d}u^\dagger_{\theta_t} \left(\Pi_{\widehat{T}_{\theta_t} \mathcal{M}} \nabla \mathcal{L}_{u_{\theta_t}} \right). \tag{4.10}$$

Note that this update can also be interpreted, from a functional perspective, as an instance of the standard Nyström method (Sun et al., 2015), thereby bridging our approach with a large family of methods developed in this field.

4.2.2 . ANaGRAM approximation theorem

The remaining challenge is to design a tractable procedure to compute the update in Equation (4.10), since it involves the $NNTK_\theta$ kernel, which cannot be computed explicitly in our setting. Indeed, its evaluation would *a priori* require inverting the Gram matrix, introducing an additional layer of computational difficulty (see Remark 2.47 in Section 2.2.4 of Chapter 2). With this motivation, we now present a first result, stated in the following theorem, which constitutes one of the main contributions of this work:

Theorem 4.3 (ANaGRAM)**.** *Let us define for all* $1 \leqslant i \leqslant S$ *and for all* $1 \leqslant p \leqslant P$*:*

$$\widehat{\phi}_{\theta i,p} := \partial_p u_\theta(x_i)\,; \qquad \widehat{\nabla\mathcal{L}}_{u_\theta i} := \nabla\mathcal{L}_{u_\theta}(x_i) = u_\theta(x_i) - f(x_i). \tag{4.11}$$

Then:

$$du^\dagger_{\theta_t}\left(\Pi_{\widehat{T}_{\theta_t}\mathcal{M}}\nabla\mathcal{L}_{u_{\theta_t}}\right) = \left(\widehat{\phi}^\dagger_\theta + E^{metric}_\theta\right)\left(\widehat{\nabla\mathcal{L}}_{u_\theta} + E^\perp_\theta\right), \tag{4.12}$$

where E^{metric}_θ *and* $E^\perp_\theta$ *are correction terms, respectively accounting for the influence of the metric on the definition of the empirical tangent space, and for the subtraction of the evaluation of the orthogonal component*[1] *of the functional gradient, given by:*

$$E^{metric}_\theta = \widehat{V}_\theta\left(\boldsymbol{I}_P - \Pi_r\right)\widehat{V}^t_\theta G^\dagger_\theta \widehat{V}_\theta\,\Pi_r\,\left(\Pi_r\widehat{V}^t_\theta G^\dagger_\theta\widehat{V}_\theta\,\Pi_r\right)^\dagger\widehat{\Delta}^\dagger_\theta\widehat{U}^t_\theta, \tag{4.13}$$

with:

- $\Pi_r = \sum_{p=1}^r \boldsymbol{e}_p{\boldsymbol{e}_p}^t$*, the projection onto the* r *first coordinates of* $\mathbb{R}^P$*.*
- $\boldsymbol{I}_P$*, the identity of* $\mathbb{R}^P$
- $\widehat{U}_\theta\widehat{\Delta}_\theta\widehat{V}^t_\theta = SVD(\widehat{\phi}_\theta)$
- *for all* $1 \leqslant p, q \leqslant P$, $G_{\theta p,q} = \langle\partial_p u_\theta\,,\,\partial_q u_\theta\rangle_{\mathcal{H}}$

$$\begin{aligned} E^\perp_\theta &= \left(\langle NNTK_\theta(x_i,\cdot)\,,\,\nabla\mathcal{L}\rangle_{\mathcal{H}} - \nabla\mathcal{L}(x_i)\right)_{1\leqslant i\leqslant S} \\ &= \left(-\left(\Pi_{T^\perp_\theta\mathcal{M}}\nabla\mathcal{L}\right)(x_i)\right)_{1\leqslant i\leqslant S} \end{aligned} \tag{4.14}$$

Before proving Theorem 4.3, we need the following results.

[1] That is, the component orthogonal to the entire tangent space $T_\theta\mathcal{M}$.

Lemma 4.4. *Let $\mathcal{H}$ be a Hilbert space, and let $\mathcal{H}_0 := \mathrm{Span}(u_p : 1 \leqslant p \leqslant P) \subset \mathcal{H}$. Denote by $G_{pq} := \langle u_p \,,\, u_q \rangle_{\mathcal{H}}$ the Gram matrix of $(u_p)_{1 \leqslant p \leqslant P}$, and let $G = U \Delta^2 U^\top$ be its eigen-decomposition. Then, defining*

$$L_p := \sum_{1 \leqslant q \leqslant P} u_q U_{q,p} \Delta_p^\dagger, \tag{4.15}$$

we obtain an orthonormal basis $(L_p)_{1 \leqslant p \leqslant P}$ of $\mathcal{H}_0$. In particular, $\mathcal{H}_0$ is a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel

$$k(x, y) = \sum_{1 \leqslant p,q \leqslant P} u_p(x)\, G^\dagger_{p,q}\, u_q(y), \tag{4.16}$$

and the orthogonal projection $\Pi_{\mathcal{H}_0}$ onto $\mathcal{H}_0$ is an integral operator associated with this kernel. Furthermore, the functions (L_p) are the left singular vectors of the so-called ***synthesis operator***[2]*:*

$$\mathcal{T} : \left\{ \begin{array}{ccc} \mathbb{R}^P & \to & \mathcal{H}_0 \\ \alpha & \mapsto & \displaystyle\sum_{1 \leqslant p \leqslant P} \alpha_p u_p \end{array} \right. . \tag{4.17}$$

Proof. Since $\mathcal{H}_0$ is generated by the finite basis $(u_p)_{1 \leqslant p \leqslant P}$, it is an RKHS by Proposition 3.19 in Section 3.2.2 of Chapter 3. Furthermore there exist (for instance by the Gram-Schmidt procedure), an orthonormal basis $(V_p)_{1 \leqslant p \leqslant P}$ of $\mathcal{H}_0$. Then by Proposition 3.29 of Section 3.3.1 in Chapter 3, the reproducing kernel of $\mathcal{H}_0$ is given by: for all $x, y \in \Omega$

$$k(x, y) := \sum_{p=1}^{P} V_p(x) V_p(y). \tag{4.18}$$

Now applying Proposition 3.28 of Section 3.3.1 in Chapter 3, the operator Π defined by: for all $f \in \mathcal{H}$

$$\Pi(f) := \sum_{1 \leqslant p \leqslant P} V_p \langle f \,,\, V_p \rangle \tag{4.19}$$

is thus orthogonal projection onto $\mathcal{H}_0$. But the basis $(L_p)_{\leqslant p \leqslant P}$ of Equation (4.15) is precisely orthonormal. Indeed:

$$\begin{aligned} \langle L_p \,,\, L_q \rangle &= \left\langle \sum_{1 \leqslant k \leqslant P} u_k U_{k,p} \Delta_p^\dagger \,,\, \sum_{1 \leqslant l \leqslant P} u_l U_{l,q} \Delta_q^\dagger \right\rangle \\ &= \sum_{1 \leqslant k \leqslant P} \sum_{1 \leqslant l \leqslant P} \Delta_p^\dagger U_{p,k} \langle u_k \,,\, u_l \rangle U_{l,q} \Delta_q^\dagger \\ &= \boldsymbol{e}_p{}^t \Delta^\dagger U^t G U \Delta^\dagger \boldsymbol{e}_q \\ &= \boldsymbol{e}_p{}^t \Delta^\dagger U^t U \Delta^2 U^t U \Delta^\dagger \boldsymbol{e}_q = \delta_{pq}, \end{aligned} \tag{4.20}$$

[2]The terminology and notation follow Adcock and Huybrechs (2019).

where δ_{pq} is the Kronecker symbol such that $\delta_{pq} = 1$ if and only if $p = q$. Now building Π upon this basis yields: for all $f \in \mathcal{H}$

$$\begin{aligned}\Pi(f) &= \sum_{1 \leqslant p \leqslant P} L_p \langle L_p \,,\, f \rangle \\ &= \sum_{1 \leqslant p \leqslant P} \sum_{1 \leqslant k \leqslant P} \sum_{1 \leqslant l \leqslant P} u_k U_{k,p} \Delta_p^\dagger \left\langle u_l U_{l,p} \Delta_p^\dagger \,,\, f \right\rangle \\ &= \sum_{1 \leqslant k \leqslant P} \sum_{1 \leqslant l \leqslant P} u_k \left(\sum_{1 \leqslant p \leqslant P} U_{k,p} \Delta_p^{2\dagger} U_{p,l} \right) \langle u_l \,,\, f \rangle \\ &= \sum_{1 \leqslant k,l \leqslant P} u_k G_{k,l}^\dagger \langle u_l \,,\, f \rangle\end{aligned} \tag{4.21}$$

Thus, the kernel of the projection $\Pi_{\mathcal{H}_0}$ onto $\mathcal{H}_0$ is precisely:

$$k(x,y) = \sum_{1 \leqslant k,l \leqslant P} u_k(x) G_{k,l}^\dagger u_l(y). \tag{4.22}$$

Finally, let us write the SVD of the *synthesis* operator: $\forall \alpha \in \mathbb{R}^P$

$$\mathcal{T}(\alpha) = \sum_{1 \leqslant p \leqslant P} v_p \Lambda_p W_p^t \alpha \in \mathcal{H}_0. \tag{4.23}$$

Then in particular, we have: for all $1 \leqslant p \leqslant P$

$$v_p = \mathcal{T}(W_p \Lambda_p^\dagger), \tag{4.24}$$

and: for all $1 \leqslant p \leqslant P$

$$u_p = \mathcal{T}(\boldsymbol{e}_p). \tag{4.25}$$

This implies that: for all $1 \leqslant p, q \leqslant P$

$$\begin{aligned}G_{p,q} &= \langle u_p \,,\, u_q \rangle = \langle \mathcal{T}(\boldsymbol{e}_p) \,,\, \mathcal{T}(\boldsymbol{e}_q) \rangle \\ &\overset{(4.23)}{=} \sum_{1 \leqslant k,l \leqslant P} \boldsymbol{e}_p{}^t W_k \Lambda_k \underbrace{\langle v_k \,,\, v_l \rangle}_{=\delta_{kl}} \Lambda_l W_l^t \boldsymbol{e}_q = \boldsymbol{e}_p{}^t W \Lambda^2 W^t \boldsymbol{e}_q.\end{aligned} \tag{4.26}$$

This means that (W_p) and (Λ_p^2) are respectively the eigenvectors and eigenvalues of G. The result follows by unicity of eigen-decomposition and respective identification of (W_p) to (U_p) and (Λ_p) to (Δ_p) in Equation (4.15). □

This observation enable us to establish a simple but important result, linking RKHS theory, (N)NTK and natural gradient.

Corollary 4.5. *The $NNTK_\theta$ defined in Section 2.2.3 of Chapter 2 is the reproducing kernel of $T_\theta\mathcal{M}$ and the kernel of the projection $\Pi_{T_\theta\mathcal{M}} : L^2(\Omega \to \mathbb{R}, \mu) \to L^2(\Omega \to \mathbb{R}, \mu)$ onto $T_\theta\mathcal{M}$.*

Proof. This is a direct consequence of Lemma 4.4, since $T_\theta\mathcal{M} = \mathrm{Span}(\partial_p u_\theta : 1 \leqslant p \leqslant P)$. □

We can now prove Theorem 4.3.

Proof of Theorem 4.3. First of all, following Lemma 4.4, we see that the projection kernel into $\widehat{T}_\theta\mathcal{M}$ is given by: for all $x, y \in \Omega$

$$\hat{k}(x,y) = \sum_{1\leqslant i,j\leqslant S} NNTK_\theta(x_i,x)\widehat{G}^\dagger_{\theta i,j} NNTK_\theta(x_j,y), \tag{4.27}$$

with: for all $1 \leqslant i,j \leqslant S$, for all $\theta \in \mathbb{R}^P$

$$\widehat{G}_{\theta i,j} = \langle NNTK_\theta(x_i,\cdot)\,,\, NNTK_\theta(x_j,\cdot)\rangle_{\mathcal{H}} = NNTK_\theta(x_i,x_j), \tag{4.28}$$

where last equality comes from the fact that $NNTK_\theta$ is the reproducing kernel of $T_\theta\mathcal{M}$. Noting $\mathcal{H} = \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$, we can then simplify Equation (4.10):

$$\begin{aligned}
&\mathrm{d}u^\dagger_{\theta_t}\left(\Pi_{\widehat{T}_{\theta_t}\mathcal{M}}\nabla\mathcal{L}_{u_{\theta_t}}\right)\\
&= \mathrm{d}u^\dagger_{\theta_t}\left(x\in\Omega \mapsto \left\langle \hat{k}(x,\cdot)\,,\,\nabla\mathcal{L}_{u_{\theta_t}}\right\rangle_{\mathcal{H}}\right)\\
&= \sum_{1\leqslant i,j\leqslant S} \mathrm{d}u^\dagger_{\theta_t}\left(NNTK_{\theta_t}(x_i,\cdot)\right)\widehat{G}_{\theta_t}{}^\dagger_{i,j}\left\langle NNTK_{\theta_t}(x_j,\cdot)\,,\,\nabla\mathcal{L}_{u_{\theta_t}}\right\rangle_{\mathcal{H}}\\
&= \sum_{\substack{1\leqslant p,q\leqslant P\\ 1\leqslant i,j\leqslant S}} \mathrm{d}u^\dagger_{\theta_t}\left(\partial_p u_{\theta_t} G^\dagger_{\theta_t p,q}\right)\partial_q u_{\theta_t}(x_i)\widehat{G}_{\theta_t}{}^\dagger_{i,j}\left\langle NNTK_{\theta_t}(x_j,\cdot)\,,\,\nabla\mathcal{L}_{u_{\theta_t}}\right\rangle_{\mathcal{H}}\\
&= \sum_{\substack{1\leqslant p,q\leqslant P\\ 1\leqslant i,j\leqslant S}} G_{\theta_t}{}^\dagger_{p,q}\partial_q u_{\theta_t}(x_i)\,\widehat{G}_{\theta_t}{}^\dagger_{i,j}\left\langle NNTK_{\theta_t}(x_j,\cdot)\,,\,\nabla\mathcal{L}_{u_{\theta_t}}\right\rangle_{\mathcal{H}},
\end{aligned} \tag{4.29}$$

Using the empirical jacobian matrix introduced in Equation (4.11) of Theorem 4.3 statement, namely: for all $\theta \in \mathbb{R}^P$ and for all $1 \leqslant i \leqslant S$, for all $1 \leqslant p \leqslant P$

$$\widehat{\phi}_{\theta i,p} := \partial_p u_\theta(x_i). \tag{4.30}$$

Equation (4.28) rewrites: for all $\theta \in \mathbb{R}^P$

$$\widehat{G}_\theta = \widehat{\phi}_\theta G^\dagger_\theta \widehat{\phi}^t_\theta. \tag{4.31}$$

Introducing also: for all $\theta \in \mathbb{R}^P$ and for all $1 \leqslant i \leqslant S$

$$\widehat{\nabla\mathcal{L}}^{\|}_{\theta i} := \langle NNTK_\theta(x_i,\cdot)\,,\,\nabla\mathcal{L}_{u_\theta}\rangle_{\mathcal{H}}\,. \tag{4.32}$$

Equation (4.29) then rewrites:

$$\mathrm{d}u^\dagger_{\theta_t}\left(\Pi_{\widehat{T}_{\theta_t}\mathcal{M}}\nabla\mathcal{L}_{u_{\theta_t}}\right) = G^\dagger_{\theta_t}\widehat{\phi}^t_{\theta_t}\left(\widehat{\phi}_{\theta_t}G^\dagger_{\theta_t}\widehat{\phi}^t_{\theta_t}\right)^\dagger \widehat{\nabla\mathcal{L}}^{\|}_{\theta_t}. \tag{4.33}$$

Using now the SVD of $\widehat{\phi}_\theta$, also introduced in the statement of Theorem 4.3, namely: for all $\theta \in \mathbb{R}^P$

$$\widehat{\phi}_\theta = \widehat{U}_\theta\widehat{\Delta}_\theta\widehat{V}^t_\theta, \tag{4.34}$$

we may express the pseudo-inverse of $\widehat{\phi}_\theta$ as:

$$\widehat{\phi}_\theta^\dagger = \widehat{U}_\theta \widehat{\Delta}_\theta^\dagger \widehat{V}_\theta^t, \tag{4.35}$$

and rewrite Equation (4.31) as:

$$\widehat{G}_\theta = \widehat{U}_\theta \widehat{\Delta}_\theta \widehat{V}_\theta^t G_\theta^\dagger \widehat{V}_\theta \widehat{\Delta}_\theta \widehat{U}_\theta^t. \tag{4.36}$$

Let use denote $r \leqslant S$ the rank of $\widehat{\phi}_\theta$, and introduce $\Pi_r := \sum_{p=1}^r \boldsymbol{e}_p {\boldsymbol{e}_p}^t$ the projection onto the first r coordinates of the canonical basis $(\boldsymbol{e}_p)_{p=1}^P$ of $\mathbb{R}^P$. Observe that, in particular, $\widehat{\Delta}_\theta \widehat{\Delta}_\theta^\dagger = \Pi_r$. Then, noting that $\widehat{U}_\theta$ is orthogonal, and $\widehat{\Delta}_\theta$ diagonal:

$$\widehat{G}_\theta^\dagger = \widehat{U}_\theta \widehat{\Delta}_\theta^\dagger \left(\Pi_r \widehat{V}_\theta^t G_\theta^\dagger \widehat{V}_\theta \Pi_r \right)^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t = \widehat{U}_\theta \widehat{\Delta}_\theta^\dagger \Sigma_\theta^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t, \tag{4.37}$$

where

$$\Sigma_\theta := \Pi_r \widehat{V}_\theta^t G_\theta^\dagger \widehat{V}_\theta \Pi_r. \tag{4.38}$$

Inserting Equation (4.37) and Equation (4.34) in Equation (4.33), we get:

$$\begin{aligned}
\mathrm{d}u_{\theta_t}^\dagger \left(\Pi_{\widehat{T}_{\theta_t}\mathcal{M}} \nabla \mathcal{L}_{u_{\theta_t}} \right) =& G_\theta^\dagger \widehat{V}_\theta \underbrace{\widehat{\Delta}_\theta \widehat{U}_\theta^t \widehat{U}_\theta \widehat{\Delta}_\theta^\dagger}_{=\Pi_r} \Sigma_\theta^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t \widehat{\nabla \mathcal{L}}_\theta^{\|} \\
=& \widehat{V}_\theta \widehat{V}_\theta^t G_\theta^\dagger \widehat{V}_\theta \, \Pi_r \, \Sigma_\theta^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t \widehat{\nabla \mathcal{L}}_\theta^{\|} \\
=& \widehat{V}_\theta \big((\boldsymbol{I}_P - \Pi_r) + \Pi_r \big) \widehat{V}_\theta^t G_\theta^\dagger \widehat{V}_\theta \, \Pi_r \, \Sigma_\theta^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t \widehat{\nabla \mathcal{L}}_\theta^{\|} \\
=& \Big(\widehat{V}_\theta \Sigma_\theta \Sigma_\theta^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t \\
& + \underbrace{\widehat{V}_\theta \left(\boldsymbol{I}_P - \Pi_r \right) \widehat{V}_\theta^t G_\theta^\dagger \widehat{V}_\theta \, \Pi_r \, \Sigma_\theta^\dagger \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t}_{E_\theta^{\text{metric}}} \Big) \widehat{\nabla \mathcal{L}}_\theta^{\|} \\
=& \left(\widehat{V}_\theta \widehat{\Delta}_\theta^\dagger \widehat{U}_\theta^t + E_\theta^{\text{metric}} \right) \widehat{\nabla \mathcal{L}}_\theta^{\|} = \left(\widehat{\phi}_\theta^\dagger + E_\theta^{\text{metric}} \right) \widehat{\nabla \mathcal{L}}_\theta^{\|}.
\end{aligned} \tag{4.39}$$

Finally, by decomposing $\nabla \mathcal{L}_{u_\theta}$ into its collinear and orthogonal components to $T_\theta \mathcal{M}$, *i.e.* : for all $\theta \in \mathbb{R}^P$

$$\nabla \mathcal{L}_{u_\theta} = \Pi_{T_\theta \mathcal{M}} \left(\nabla \mathcal{L}_{u_\theta} \right) + \Pi_{T_\theta^\perp \mathcal{M}} \left(\nabla \mathcal{L}_{u_\theta} \right), \tag{4.40}$$

and using the notation $\widehat{\nabla \mathcal{L}}_{u_\theta}$ introduced in Equation (4.11) of Theorem 4.3 statement, we have: for all $\theta \in \mathbb{R}^P$, for all $1 \leqslant i \leqslant S$

$$\widehat{\nabla \mathcal{L}}_{u_\theta\, i} = \nabla \mathcal{L}_{u_\theta}(x_i) = \Pi_{T_\theta \mathcal{M}} \left(\nabla \mathcal{L}_{u_\theta} \right)(x_i) + \Pi_{T_\theta^\perp \mathcal{M}} \left(\nabla \mathcal{L}_{u_\theta} \right)(x_i) \tag{4.41}$$

$$= \langle NNTK_\theta(x_i, \cdot) \, , \, \nabla \mathcal{L}_{u_\theta} \rangle_{\mathcal{H}} + \Pi_{T_\theta^\perp \mathcal{M}} \left(\nabla \mathcal{L}_{u_\theta} \right)(x_i) \tag{4.42}$$

$$= \widehat{\nabla \mathcal{L}}_{\theta i}^{\|} - {E_\theta^\perp}_i, \tag{4.43}$$

where Equation (4.42) comes from the fact that $NNTK_\theta$ is the kernel defining $\Pi_{T_\theta\mathcal{M}}$, and Equation (4.43) uses the defintion given by Equation (4.32) and the notation $E_\theta^\perp$ introduced in Theorem 4.3 statement, specified in Equation (4.14). Thus $\widehat{\nabla\mathcal{L}}_\theta^{\|} = \widehat{\nabla\mathcal{L}}_{u_\theta} + E_\theta^\perp$, which concludes. □

In some important cases the correction terms E_θ^{metric} and $E_\theta^\perp$ vanishes.

Proposition 4.6. *If u can be factorized as $u_\theta = L_{\theta_1}[C_{\theta_2}]$, with $\theta = (\theta_1, \theta_2) \in \mathbb{R}^{P_1+P_2}$, $C : \mathbb{R}^{P_2} \to \mathcal{H}_1$, $L : \mathbb{R}^{P_1} \to \mathcal{F}(\mathcal{H}_1 \to L^2(\Omega, \mu))$ linear in θ_1, and $f = 0$, then $E_\theta^\perp = 0$.*

Proof. From the discussion at beginning of Section 4.2, more precisely the identification of the Fréchet derivative of the functional loss in Equation (4.7), we have that the functional gradient for quadratic regression is:

$$\nabla\mathcal{L}_{u_\theta} = u_\theta - f. \tag{4.44}$$

Assuming that $f = 0$, this reduces to:

$$\nabla\mathcal{L}_{u_\theta} = u_\theta. \tag{4.45}$$

Now using the assumption and notations of Proposition 4.6, we see that:

$$\partial_{\theta_1} u_\theta(\theta_1) = \mathrm{d}L_{\theta_1}(\theta_1)[C_{\theta_2}] = L_{\theta_1}[C_{\theta_2}] = u_{|(\theta_1,\theta_2)} = u_\theta, \tag{4.46}$$

where the second equality comes from the linearity of L with respect to θ_1. In particular this implies that $u_\theta \in T_\theta\mathcal{M}$, which conludes. □

E_θ^{metric} cancels out in the following case.

Proposition 4.7. *There exist P points $(\hat{x}_i)$ such that $\widehat{T}^{NNTK}_{\theta,(x_i)}\mathcal{M} = T_\theta\mathcal{M}$. Then notably $E_\theta^{metric} = 0$.*

Proof. Let $d := \dim(T_\theta\mathcal{M}) \leqslant P$. Since $NNTK_\theta$ is the reproducing kernel of $T_\theta\mathcal{M}$ by Corollary 4.5, we have by Proposition 3.24 in Section 3.3.1 of Chapter 3 that:

$$\overline{\mathrm{Span}(NNTK_\theta(\cdot, x) \,:\, x \in \Omega)} = T_\theta\mathcal{M}. \tag{4.47}$$

In particular, $\overline{\mathrm{Span}(NNTK_\theta(\cdot, x) \,:\, x \in \Omega)}$ is also of finite dimension $d \leqslant P$. Since $(NNTK_\theta(\cdot, x))_{x\in\Omega}$ is a generating family, one may extract a free subfamily out of it, which will be of cardinal $d \leqslant P$, *i.e.* there exist $d \leqslant P$ points $(\hat{x}_i)_{1\leqslant i\leqslant d}$ such that

$$\widehat{T}_\theta\mathcal{M} = \mathrm{Span}\left(NNTK_\theta(\cdot, \hat{x}_i) \,:\, 1 \leqslant i \leqslant d\right) = T_\theta\mathcal{M}, \tag{4.48}$$

and thus:

$$\Pi_{\widehat{T}_\theta\mathcal{M}}\nabla\mathcal{L} = \Pi_{T_\theta\mathcal{M}}\nabla\mathcal{L}.$$

If $d < P$, the sequence $(\hat{x}_i)_{1\leqslant i\leqslant d}$ can be extended with additional $P - d$ arbitrary points. □

4.2.3 . ANaGRAM algorithm

As a first approximation, we can neglect those two terms, yielding the following vanilla algorithm:

Algorithme 2 : vanilla ANaGRAM

Input :

- $u : \mathbb{R}^P \to \mathrm{L}^2(\Omega, \mu)$ `// neural network architecture`
- $\theta_0 \in \mathbb{R}^P$ `// initialization of the neural network`
- $f \in \mathrm{L}^2(\Omega, \mu)$ `// target function of the quadratic regression`
- $(x_i) \in \Omega^S$ `// a batch in` Ω
- $\epsilon > 0$ `// cutoff level to compute the pseudo inverse`

repeat

$\quad \widehat{\phi}_{\theta_t} \leftarrow (\partial_p u_{\theta_t}(x_i))_{1 \leqslant i \leqslant S,\, 1 \leqslant p \leqslant P}$ `// Computed via auto-differentiation`

$\quad \widehat{U}_{\theta_t}, \widehat{\Delta}_{\theta_t}, \widehat{V}^t_{\theta_t} \leftarrow SVD\left(\widehat{\phi}_{\theta_t}\right)$

$\quad \widehat{\Delta}_{\theta_t} \leftarrow \left(\widehat{\Delta}_{\theta_t p} \text{ if } \widehat{\Delta}_{\theta_t p} > \epsilon \text{ else } 0\right)_{1 \leqslant p \leqslant P}$

$\quad \widehat{\nabla \mathcal{L}} \leftarrow (u_{\theta_t}(x_i) - f(x_i))_{1 \leqslant i \leqslant S}$

$\quad d_{\theta_t} \leftarrow \widehat{V}_{\theta_t} \widehat{\Delta}^{\dagger}_{\theta_t} \widehat{U}^t_{\theta_t} \widehat{\nabla \mathcal{L}}$

$\quad \eta_t \leftarrow \underset{\eta \in \mathbb{R}^+}{\arg\min} \sum_{1 \leqslant i \leqslant S} \left(f(x_i) - u_{\theta_t - \eta d_{\theta_t}}(x_i)\right)^2$ `// Using e.g. line search`

$\quad \theta_{t+1} \leftarrow \theta_t - \eta_t\, d_{\theta_t}$

until *stop criterion met*

Note that Algorithm 2 is equivalent to Gauss-Newton algorithm (Cai et al., 2019) applied to the empirical loss in Equation (2.73) also considered recently in Jnini et al. (2024) with a different setting. Nevertheless, our work aims at a more general approach, giving rise to different algorithms depending on the approximations of $E^{\text{metric}}_{\theta}$ and $E^{\perp}_{\theta}$. One of the pleasant byproducts of the ANaGRAM framework is also that it leads to a straightforward criterion to choose points in the batch, namely:

$$(x_i^*) := \underset{(x_i) \in \Omega^S}{\arg\min} \left\| \Pi_{\mathrm{Span}(NNTK_\theta(x_i, \cdot) : 1 \leqslant i \leqslant S)} \left(\nabla \mathcal{L}_{u_\theta}\right) - \nabla \mathcal{L}_{u_\theta} \right\|^2_{\mathrm{L}^2(\Omega \to \mathbb{R}, \mu)}, \quad (4.49)$$

which is amenable to various approximations, subject to further investigations. Taking the best advantage of this criterion should eventually allow us to use natural gradient in a mini-batch setting while staying close to the convergence rate of the full batch natural gradient as characterized in Xu et al. (2024). We will now show how ANaGRAM can be applied to the PINNs framework.

4.3 . Adaptation of empirical Natural Gradient and ANaGRAM to PINNs

Generalizing ANaGRAM to PINNs only requires to change the problem perspective.

4.3.1 . PINNs as a least-squares regression problem

The only difference between the losses:

- Equation (2.72) of Definition 2.43 in Section 2.2.2 of Chapter 2, namely:

$$\begin{aligned}\mathcal{L}_{X_D,X_B}(v) :=& \frac{\mu_n(\Omega)}{2\,S_D}\sum_{i=1}^{S_D}\big\|\big(D[v]-f\big)(x_{D,i})\big\|^2_{\mathbb{R}^{m_D}} \\ &+\frac{\sigma_n(\partial\Omega)}{2\,S_B}\sum_{i=1}^{S_B}\big\|\big(B[v]-g\big)(x_{B,i})\big\|^2_{\mathbb{R}^{m_B}}\,,\end{aligned} \tag{2.72}$$

- Equation (2.73) of Definition 2.45 in Section 2.2.2 of Chapter 2, namely:

$$\mathcal{L}_{2,X}(v) := \frac{\mu_n(\Omega)}{2\,S}\sum_{i=1}^{S}\|v(x_i)-f(x_i)\|^2_{\mathbb{R}^m}\,, \tag{2.73}$$

is the use of the differential operator D and the boundary operator B in Equation (2.72). More precisely, PINNs and standard quadratic regression problems are essentially similar, except that in the case of PINNs we use the compound model $(D,B)\circ u$ instead of u directly, where, using the definitions of Equation (4.3):

$$(D,B)\circ u : \left\{\begin{array}{ccccc}\mathbb{R}^P & \to & \mathcal{H} & \to & \mathrm{L}^2\left(\Omega,\partial\Omega\right)\\ \theta & \mapsto & u_\theta & \mapsto & (D[u_\theta],B[u_\theta])\end{array}\right. , \tag{4.50}$$

with

$$\mathrm{L}^2\left(\Omega,\partial\Omega\right) := \mathrm{L}^2(\Omega\to\mathbb{R},\mu)\times\mathrm{L}^2(\partial\Omega\to\mathbb{R},\sigma). \tag{4.51}$$

To make the adaptation explicit, we compare in Table 4.1 below the objects introduced in Section 4.1 to define natural gradient and empirical natural gradient in classical quadradic regression context, with their equivalents for the PINNs case.

Natural neural tangent kernel and empirical tangent space pose certain technical difficulties and are therefore dealt with separately in Section 4.3.2 below.

Table 4.1: Comparison of objects introduced in Section 4.1 for classical and PINNs regression.

	Classical regression	PINNs regression
Image set of the model	$\begin{aligned}\mathcal{M} &:= \operatorname{Im} u \\ &= \left\{u_\theta : \theta \in \mathbb{R}^P\right\} \\ &\subset \mathrm{L}^2(\Omega)\end{aligned}$	$\begin{aligned}\Gamma &:= \operatorname{Im}\big((D,B)\circ u\big) \\ &= \left\{(D[u_\theta], B[u_\theta]) : \theta \in \mathbb{R}^P\right\} \\ &\subset \mathrm{L}^2(\Omega,\partial\Omega) = \mathrm{L}^2(\Omega)\times\mathrm{L}^2(\partial\Omega)\end{aligned}$
Model differential	$\begin{aligned}&\mathrm{d}u_\theta \in \mathcal{L}in\left(\mathbb{R}^P \to \mathrm{L}^2(\Omega)\right) \\ &\quad\text{such that:} \\ u_{\theta+h} = \;& u_\theta \\ &+ \mathrm{d}u_\theta(h) \\ &+ o(\|h\|_2)\end{aligned}$	$\begin{aligned}&\mathrm{d}\left((D,B)\circ u\right)_\theta \in \mathcal{L}in\left(\mathbb{R}^P \to \mathrm{L}^2(\Omega,\partial\Omega)\right) \\ &\quad\text{such that:} \\ \left((D,B)\circ u\right)_{\theta+h} = \;& \left((D,B)\circ u\right)_\theta \\ &+ \mathrm{d}\left((D,B)\circ u\right)_\theta(h) \\ &+ o(\|h\|_2)\end{aligned}$
Partial derivatives	$\begin{aligned}\partial_p u_\theta &= \mathrm{d}u_\theta(\boldsymbol{e}_p) \\ &= \lim_{\varepsilon\to 0}\frac{u_{\theta+\varepsilon\boldsymbol{e}_p} - u_\theta}{\varepsilon}\end{aligned}$	$\begin{aligned}\partial_p\left((D,B)\circ u\right)_\theta =& \mathrm{d}\left((D,B)\circ u\right)_\theta(\boldsymbol{e}_p) \\ =& \left(\partial_p D[u_\theta], \partial_p B[u_\theta]\right) \\ =& \Big(\mathrm{d}D_{u_\theta}\left(\partial_p u_\theta\right), \\ & \mathrm{d}B_{u_\theta}\left(\partial_p u_\theta\right)\Big)\end{aligned}$
Tangent space	$\begin{aligned}T_\theta\mathcal{M} &:= \operatorname{Im}\mathrm{d}u_\theta \\ &= \operatorname{Span}\left(\partial_p u_\theta\right) \\ &= \left\{\sum_{p=1}^P h_p\partial_p u_\theta : \right. \\ &\qquad \left.(h_p)\in\mathbb{R}^P\right\}\end{aligned}$	$\begin{aligned}T_\theta\Gamma &:= \operatorname{Im}\mathrm{d}\big((D,B)\circ u\big)_\theta \\ &= \operatorname{Span}\left(\partial_p D[u_\theta], \partial_p B[u_\theta]\right) \\ &= \left\{\sum_{p=1}^P h_p\left(\partial_p D[u_\theta], \partial_p B[u_\theta]\right) : \right. \\ &\qquad \left.(h_p)\in\mathbb{R}^P\right\}\end{aligned}$
Functional loss	$\mathcal{L} : \begin{cases}\mathrm{L}^2(\Omega) \to \mathbb{R}^+ \\ v \qquad\quad \mapsto \frac{1}{2}\|v-f\|^2_{\mathrm{L}^2(\Omega)}\end{cases}$	$\mathcal{L} : \begin{cases}\mathrm{L}^2(\Omega,\partial\Omega) \to \mathbb{R}^+ \\ v \qquad\quad\;\; \mapsto \frac{1}{2}\|v-(f,g)\|^2_{\mathrm{L}^2(\Omega,\partial\Omega)}\end{cases}$
Functional gradient	$\begin{aligned}\nabla\mathcal{L}_\theta =& \nabla\mathcal{L}_{u_\theta} \\ =& \underbrace{u_\theta - f}_{\in\mathrm{L}^2(\Omega)}\end{aligned}$	$\begin{aligned}\nabla\mathcal{L}_\theta =& \nabla\mathcal{L}_{((D,B)\circ u)_\theta} \\ =& \underbrace{\big((D,B)\circ u\big)_\theta - (f,g)}_{\in\mathrm{L}^2(\Omega,\partial\Omega)}\end{aligned}$
Natural gradient	$\mathrm{d}u_\theta^\dagger\left(\Pi_{T_\theta\mathcal{M}}\left(\nabla\mathcal{L}_\theta\right)\right)$	$\mathrm{d}\big((D,B)\circ u\big)_\theta^\dagger\left(\Pi_{T_\theta\Gamma}\left(\nabla\mathcal{L}_\theta\right)\right)$

4.3.2 . Natural Neural Tangent Kernel and empirical Tangent Space of PINNs

Seeing PINNs as a quadratic regression problem with respect to the coumpound model of Equation (4.50), as established in Section 4.3.1, we see that the "natural" defintion of NNTK that arises from Lemma 4.4 is: for all $\theta \in \mathbb{R}^P$, for all $x, y \in (\Omega \times \partial\Omega)$

$$NNTK_\theta(x,y) = \sum_{1 \leqslant p,q \leqslant P} \partial_p \left((D,B)\circ u\right)_\theta (x) G^\dagger_{\theta\, p,q} \partial_q \left((D,B)\circ u\right)_\theta (y)^t.$$

with:

$$G_\theta := \left\langle \partial_p \left((D,B)\circ u\right)_\theta \, , \, \partial_q \left((D,B)\circ u\right)_\theta \right\rangle_{\mathrm{L}^2(\Omega,\partial\Omega)} \tag{4.52}$$

The problem lies in the fact, that in order to define the empirical Tangent Space, we would like to be able to separate Ω and $\partial\Omega$ contributions. To do this, we have to remark that the coumpound model defined in Equation (4.50) outputs functions that have a two-dimensional output, *i.e.* the function is vector-valuated and not scalar-valuated anymore. More precisely, we have that for all

$$f \in \mathrm{Im}((D,B)\circ u) = \Gamma \subset \mathrm{L}^2(\Omega, \partial\Omega) = \mathrm{L}^2(\Omega \to \mathbb{R}) \times \mathrm{L}^2(\partial\Omega \to \mathbb{R}),$$

there exist $f_\Omega \in \mathrm{L}^2(\Omega \to \mathbb{R})$ and $f_{\partial\Omega} \in \mathrm{L}^2(\partial\Omega \to \mathbb{R})$ such that $f = (f_\Omega, f_{\partial\Omega})$. Thus for all $x = (x_\Omega, x_{\partial\Omega}) \in \Omega \times \partial\Omega$

$$f(x) = (f_\Omega(x_\Omega), f_{\partial\Omega}(x_{\partial\Omega})) \in \mathbb{R}^2. \tag{4.53}$$

Hence, the associated reproducing kernel should be a bit revisited. In particular the reproducing property rewrites (Alvarez et al., 2012, Section 3.2): for all $f = (f_\Omega, f_{\partial\Omega}) \in T_\theta\Gamma \subset \mathrm{L}^2(\Omega, \partial\Omega)$, for all $x = (x_\Omega, x_{\partial\Omega}) \in \Omega \times \partial\Omega$ and for all $c \in \mathbb{R}^2$,

$$\left\langle f \, , \, NNTK_\theta(\cdot, x)c \right\rangle = f(x)^T c = f_\Omega(x_\Omega)c_1 + f_{\partial\Omega}(x_{\partial\Omega})c_2. \tag{4.54}$$

In particular, we have:

$$f_\Omega(x_\Omega) = \left\langle f \, , \, NNTK_\theta(\cdot, x)\boldsymbol{e}^{(1)} \right\rangle; \tag{4.55}$$

$$f_{\partial\Omega}(x_{\partial\Omega}) = \left\langle f \, , \, NNTK_\theta(\cdot, x)\boldsymbol{e}^{(2)} \right\rangle. \tag{4.56}$$

This means that the contributions coming from Ω and $\partial\Omega$ are linearly independent and can therefore be separated. More precisely, this can be done by defining the partial $NNTKs$:

- for all $y \in \Omega \times \partial\Omega$, for all $x_\Omega \in \Omega$, for all $x_{\partial\Omega} \in \partial\Omega$:

$$\begin{aligned} NNTK^\Omega_\theta(y, x_\Omega) &:= NNTK_\theta(y, (x_\Omega, x_{\partial\Omega}))\boldsymbol{e}^{(1)} \\ &= \sum_{1 \leqslant p,q \leqslant P} \partial_p((D,B)\circ u)_\theta(y) G^\dagger_{\theta\, p,q} \partial_q (D\circ u)_\theta(x_\Omega), \end{aligned} \tag{4.57}$$

- for all $y \in \Omega \times \partial\Omega$, for all $x_{\partial\Omega} \in \partial\Omega$, for all $x_\Omega \in \Omega$:

$$\begin{aligned} NNTK_\theta^{\partial\Omega}(y, x_{\partial\Omega}) &:= NNTK_\theta(y, (x_\Omega, x_{\partial\Omega}))\boldsymbol{e}^{(2)} \\ &= \sum_{1 \leqslant p,q \leqslant P} \partial_p((D, B) \circ u)_\theta(y) G_{\theta\, p,q}^\dagger \partial_q (B \circ u)_\theta(x_{\partial\Omega}), \end{aligned} \tag{4.58}$$

Then in particular: for all $x = (x_\Omega, x_{\partial\Omega}) \in \Omega \times \partial\Omega$, for all $f \in T_\theta\Gamma$

- $f_\Omega(x_\Omega) = \big\langle f\,,\, NNTK_\theta^\Omega(\cdot, x_\Omega)\big\rangle = \big\langle f\,,\, NNTK_\theta(\cdot, x)\boldsymbol{e}^{(1)}\big\rangle$.
- $f_{\partial\Omega}(x_{\partial\Omega}) = \big\langle f\,,\, NNTK_\theta^{\partial\Omega}(\cdot, x_{\partial\Omega})\big\rangle = \big\langle f\,,\, NNTK_\theta(\cdot, x)\boldsymbol{e}^{(2)}\big\rangle$.

This allows us to define an empirical tangent space for PINNs in the same way as in Equation (4.8), namely: Given two batches $(x_i^\Omega) \in \Omega^{S_\Omega}$ and $(x_i^{\partial\Omega}) \in \partial\Omega^{S_{\partial\Omega}}$, we define the associated empirical tangent space:

$$\begin{aligned} \widehat{T}^{NNTK}_{\theta,(x_i^\Omega),(x_j^{\partial\Omega})} := \mathrm{Span}\Big(&NNTK_\theta^\Omega(\cdot, x_i^\Omega), NNTK_\theta^{\partial\Omega}(\cdot, x_j^{\partial\Omega}) \,: \\ &1 \leqslant i \leqslant S_\Omega, 1 \leqslant j \leqslant S_{\partial\Omega}\Big) \end{aligned} \tag{4.59}$$

Empirical Natural Gradient and associated ANaGRAM derivation poses then no particular difficulty and are done in a similar way to Section 4.2. The resulting algorithm is detailed in Algorithm 3.

4.3.3 . NNTK *vs* NTK in PINNs context

To make this more concrete, we plot below some NTK and NNTK for the heat equation and the 2D Laplace equation. What we observe is that the NNTK yields a much more specialized kernel that NTK, in the sense that it is much more localized and also perfectly centered on the reference points. This localization property is expected to be more pronounced as the complexity of the model is increased. The drastic discrepancy observed between NTK and NNTK explain why PINNs fail to train under classical descent while empirical natural gradient solves this issue. The excellent locality of the NNTK kernel leads indeed to a much better optimization schema because the residues are indeed arbitrarily shrunk in small regions around each sample batch point by independent modifications of the function which is obviously impossible with the NTK. When increasing the complexity of the model the spatial range of the NNTK is expected to decrease accordingly and then more points will be needed to "percolate" the optimization over the domain. In such case if the batch size increases too much various principled strategies can be considered to control the complexity of the empirical natural gradient, precisely because the NNTK defines a natural distance between points which can be leverage to define an approximate block Gram matrix. In our point of view this is one of the great advantages of the empirical tangent space over the "parameters" tangent space, because no such good metric is given for free in parameters space.

Algorithme 3 : vanilla ANaGRAM for PINNs

Input :

- $u : \mathbb{R}^P \to \mathcal{H}$ `// neural network architecture`
- $\theta_0 \in \mathbb{R}^P$ `// initialization of the neural network`
- $D : \mathcal{H} \to \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$ `// differential operator`
- $B : \mathcal{H} \to \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma)$ `// boundary operator`
- $f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$ `// source term`
- $g \in \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma)$ `// boundary value`
- $(x_i^D) \in \Omega^{S_D}$ `// a batch in` Ω
- $(x_i^B) \in \Omega^{S_B}$ `// a batch in` $\partial\Omega$
- $\epsilon > 0$ `// cutoff level to compute the pseudo inverse`

repeat

$\widehat{\phi}_{\theta_t} \leftarrow \left(\left(\partial_p D[u_{\theta_t}](x_i^D)\right)_{i=1}^{S_D}, \quad \left(\partial_p B[u_{\theta_t}](x_i^B)\right)_{i=1}^{S_B} \right)_{p=1}^{P}$
`// via autodiff`

$\widehat{V}_{\theta_t}, \widehat{\Delta}_{\theta_t}, \widehat{U}^t_{\theta_t} \leftarrow SVD(\widehat{\phi}_{\theta_t})$

$\widehat{\Delta}_{\theta_t} \leftarrow \left(\widehat{\Delta}_{\theta_t r} \text{ if } \widehat{\Delta}_{\theta_t r} > \epsilon \text{ else } 0 \right)_{1 \leqslant r \leqslant P}$

$\widehat{\nabla \mathcal{L}} \leftarrow \begin{pmatrix} \left(D[u_{\theta_t}](x_i^D) - f(x_i^D)\right)_{1 \leqslant i \leqslant S_D} \\ \left(B[u_{\theta_t}](x_i^B) - g(x_i^B)\right)_{1 \leqslant i \leqslant S_B} \end{pmatrix}$

$d_{\theta_t} \leftarrow \widehat{V}_{\theta_t} \widehat{\Delta}^{\dagger}_{\theta_t} \widehat{U}^t_{\theta_t} \widehat{\nabla \mathcal{L}}$

$\eta_t \leftarrow \underset{\eta \in \mathbb{R}^+}{\arg\min} \frac{1}{2S_D} \sum_{1 \leqslant i \leqslant S_D} \left(f(x_i^D) - D[u_{\theta_t - \eta d_{\theta_t}}](x_i^D) \right)^2 + \frac{1}{2S_B} \sum_{1 \leqslant i \leqslant S_B} \left(g(x_i^B) - B[u_{\theta_t - \eta d_{\theta_t}}](x_i^B) \right)^2$ `// Using e.g. line search`

$\theta_{t+1} \leftarrow \theta_t - \eta_t \, d_{\theta_t}$

until *stop criterion met*

NTK and NNTK plots of Laplace 2 D equation

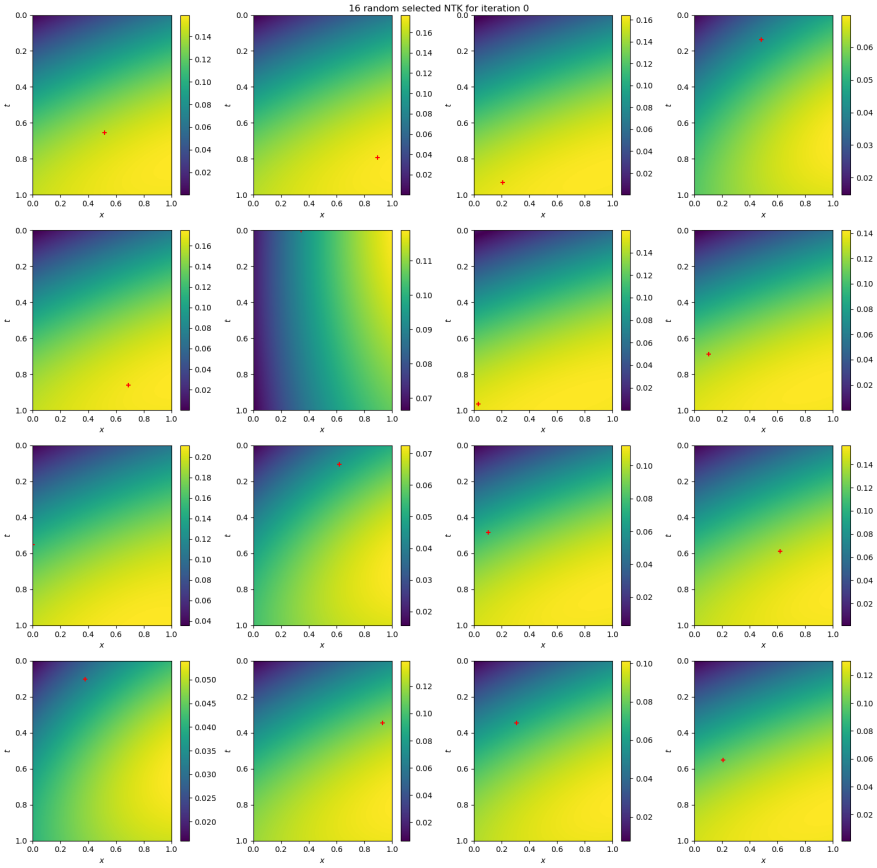


Figure 4.1: NTK at initialization for Laplace equation in 2 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NTK_{\theta_0}(\cdot, x_i)$

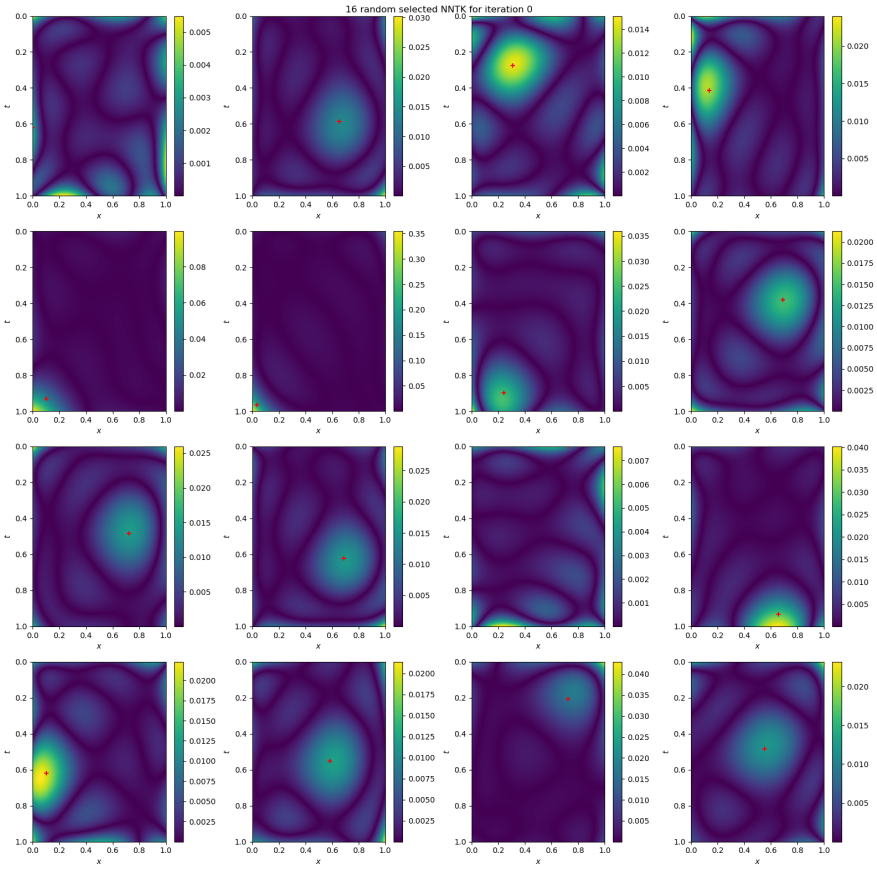


Figure 4.2: NNTK at initialization for Laplace equation in 2 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NNTK_{\theta_0}(\cdot, x_i)$

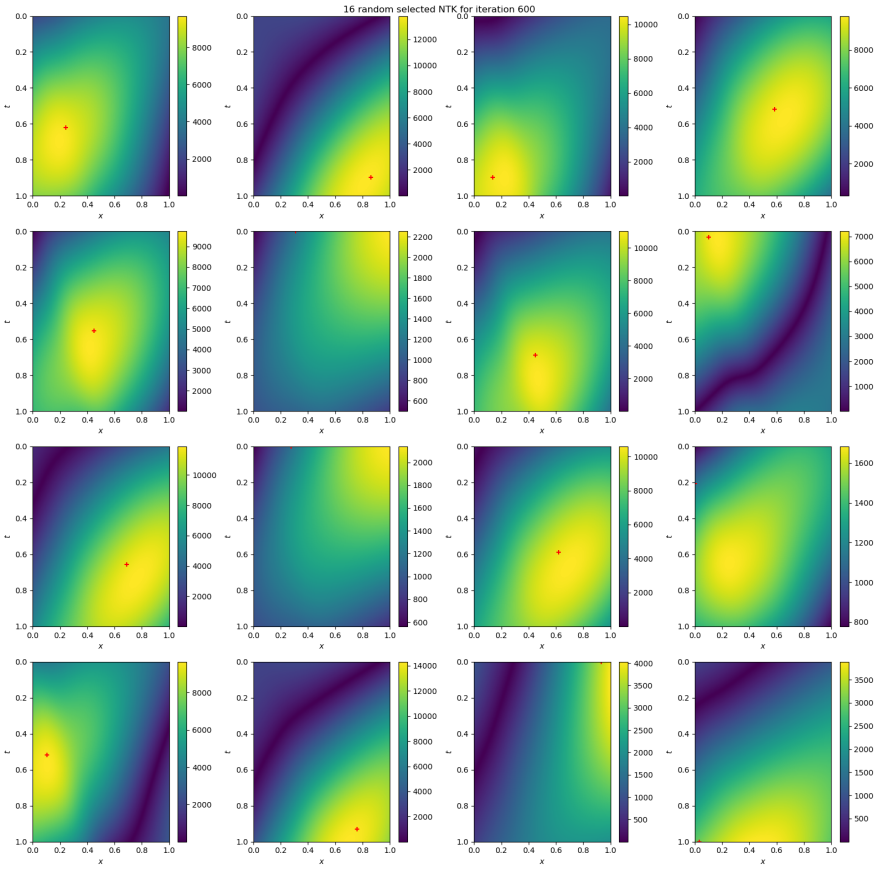


Figure 4.3: NTK at the end of optimization for Laplace equation in 2 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NTK_{\theta_{\text{end}}}(\cdot, x_i)$

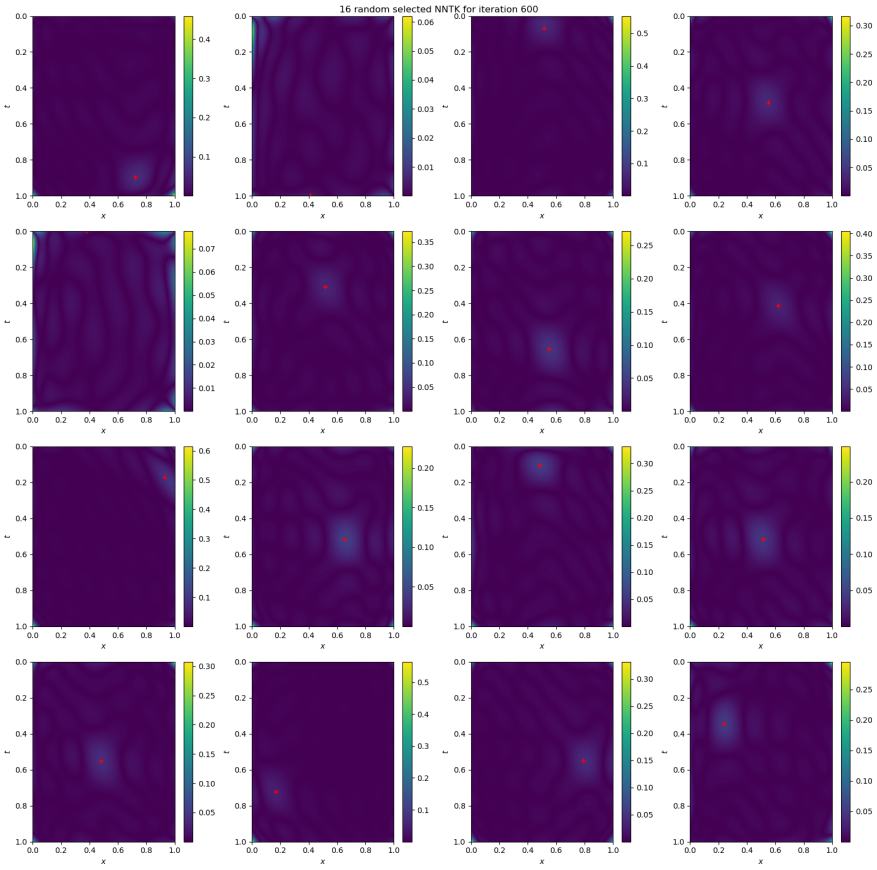


Figure 4.4: NNTK at the end of optimization for Laplace equation in 2 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NNTK_{\theta_{\text{end}}}(\cdot, x_i)$

NTK and NNTK plots of 1+1 D Heat equation

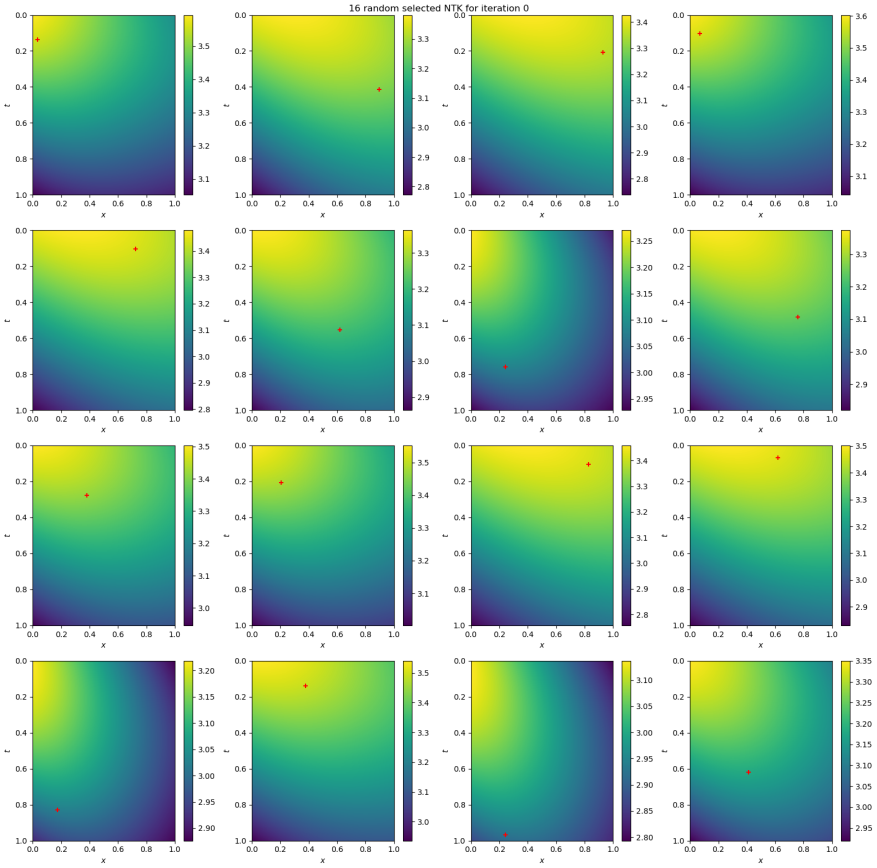


Figure 4.5: NTK at initialization for Heat equation in 1+1 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NTK_{\theta_0}(\cdot, x_i)$

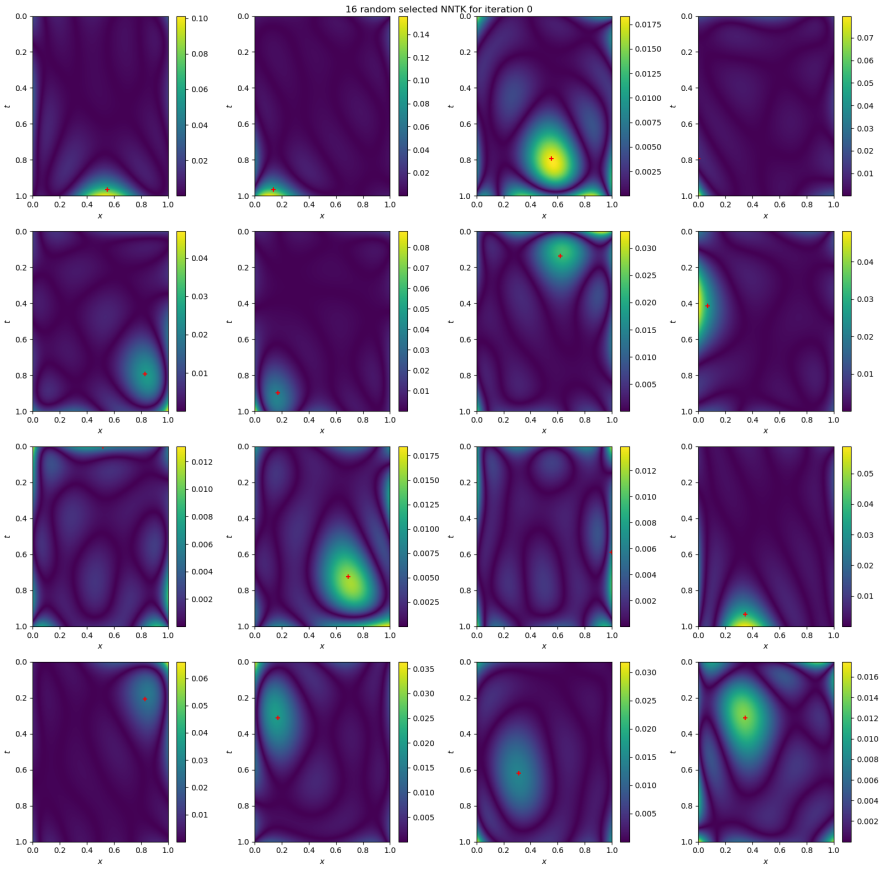


Figure 4.6: NNTK at initialization for Heat equation in 1+1 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NNTK_{\theta_0}(\cdot, x_i)$

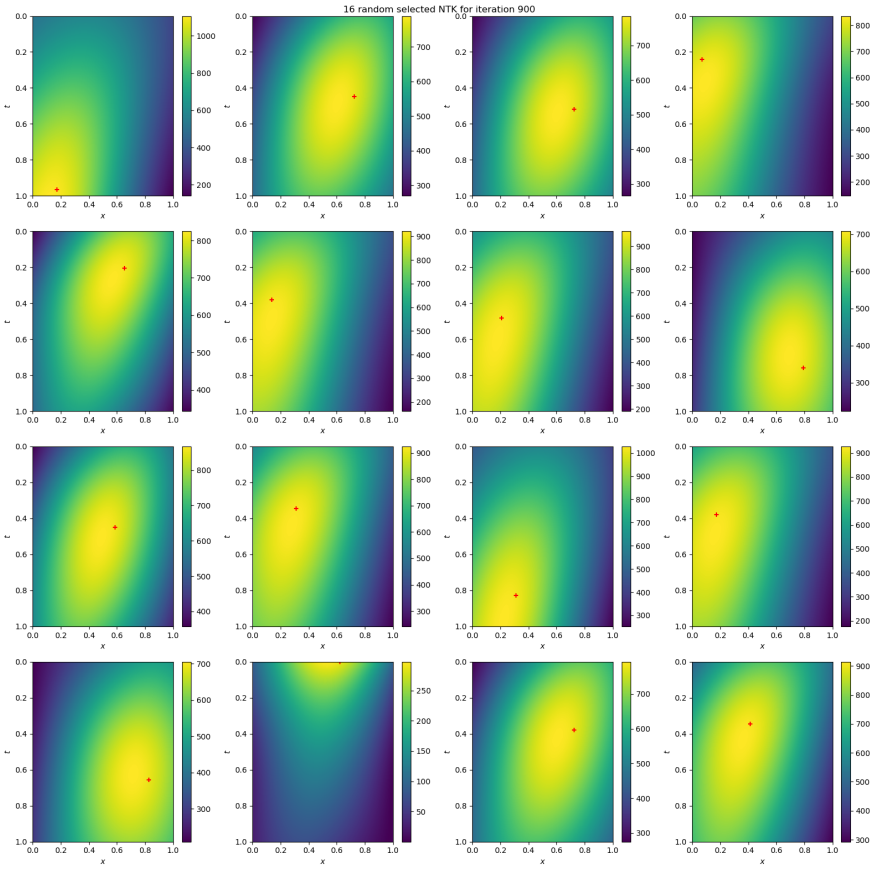


Figure 4.7: NTK at the end of optimization for Heat equation in 1+1 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NTK_{\theta_{\text{end}}}(\cdot, x_i)$

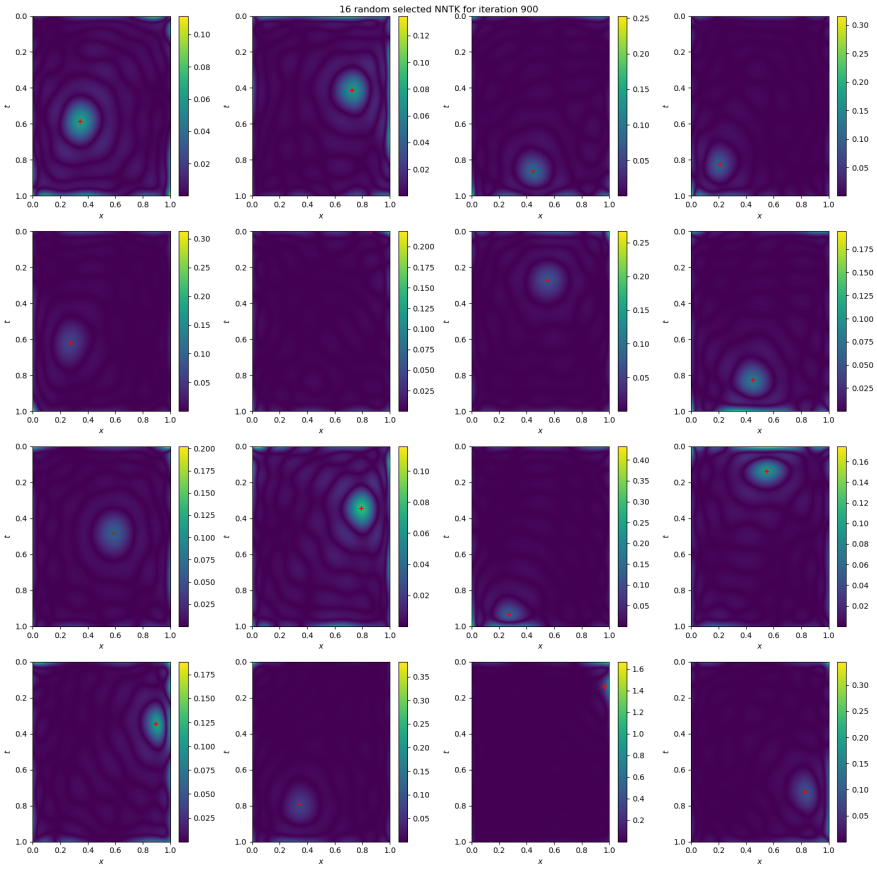


Figure 4.8: NNTK at the end of optimization for Heat equation in 1+1 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $NNTK_{\theta_{\text{end}}}(\cdot, x_i)$

We now present the link between PINN's natural gradient and Green's function.

4.4 . PINNs Natural Gradient is a Green's Function

Knowing the Green's function of a linear operator is one of the most optimal ways of solving the associated PDE, since it then suffices to estimate an integral to approximate a solution (Duffy, 2015). However, this requires prior knowledge of the Green's function, which is not always possible. Here, we show that using the natural gradient for PINNs implicitly uses the operator's Green's function.

4.4.1 . Statement of the connection with Green's functions

and interpretation

Theorem 4.8. *Let $D : \mathcal{H} \to L^2(\Omega \to \mathbb{R}, \mu)$ be a linear differential operator and $u : \mathbb{R}^P \to \mathcal{H}$ a parametric model. Then for all $\theta \in \mathbb{R}^P$, the generalized Green's function (see Definition 3.9 in Section 3.2.1 of Chapter 3) of D on $T_\theta\mathcal{M} = \operatorname{Im} du_\theta$ is given by: for all $x, y \in \Omega$*

$$g_{T_\theta\mathcal{M}}(x,y) := \sum_{1 \leqslant p,q \leqslant P} \partial_p u_\theta(x)\, G^\dagger_{p,q} \partial_q D[u_\theta](y), \tag{4.60}$$

with: for all $1 \leqslant p, q \leqslant P$

$$G_{pq} := \langle \partial_p D[u_\theta]\,,\, \partial_q D[u_\theta] \rangle_{L^2(\Omega \to \mathbb{R}, \mu)}\,. \tag{4.61}$$

In particular, the natural gradient of PINNs can be rewritten:

$$\theta_{t+1} \leftarrow \theta_t - \eta\, du^\dagger_{\theta_t} \left(x \in \Omega \mapsto \int_\Omega g_{T_{\theta_t}\mathcal{M}}(x,y) \nabla \mathcal{L}_{\theta_t}(y) \mu(dy) \right), \tag{4.62}$$

A few comments should be made about Equation (4.62). First, if $\eta = 1$, then the natural gradient can be understood as the least-square's solution of $D[u] = f$ at order 1, *i.e.* in the affine space $u_{\theta_t} + T_{\theta_t}\mathcal{M}$. However, it does not hold *a priori* that:

- $D[u_{\theta_t} + T_{\theta_t}\mathcal{M}]$ correctly approximates $f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$.
- $u_{\theta_t} + T_{\theta_t}\mathcal{M}$ correctly approximates the image space $\mathcal{M} = \operatorname{Im} u$.

Multiplying by a learning rate $\eta \ll 1$ is then essential. In this way, natural gradient can be understood as moving in the direction of the solution of $D[u] = f$ in the affine space $u_{\theta_t} + T_{\theta_t}\mathcal{M}$, and thus getting closer to the solution, while expecting that the change induced by this update will improve the approximation space $u_{\theta_{t+1}} + T_{\theta_{t+1}}\mathcal{M}$[3]. On the other hand, when we approach the end of the optimization, *i.e.* when the space $D[u_{\theta_t} + T_{\theta_t}\mathcal{M}]$ approximates f "well enough", while $\mathrm{d}u_{\theta_t}$ approximates "well enough" $\mathcal{M}$, then it is in our best interest to solve the equation completely, *i.e.* to take learning rates η close to 1. This is why the use of line search in ANaGRAM (*cf.* line 6 in Algorithm 3) is essential. We should then conclude that the quality of the solution found by the parametric model u **depends only** on:

- How well $\Gamma = \{D[u_\theta] : \theta \in \mathbb{R}^P\}$ can approximate the source $f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$.
- The curvature of Γ. More precisely, if its non-linear structure prevents convergence to a $D[u_\theta]$ such that $f - D[u_\theta]$ is non-negligible, while being orthogonal to the tangent space $D[T_{\theta_t}\mathcal{M}]$.

[3] To our best knowledge, rigorous proof of this phenomenon has not yet been provided. We can therefore only rely on empirical evidence, which is illustrated in Section 4.5.

If we assume now that D is also nonlinear, then all the above analysis also holds for the linear operators $\mathrm{d}D_{u_{\theta_t}}$, the difference being that the operator changes at each step. This means that in the case of non-linear operators, we have to deal with both the non-linearity of D and u, but that does not change the overall dynamic.

Finally, assuming that both D and u are linear (this is for instance the case when we assume u to be a linear combination of basis functions, like in Finite Elements, or Fourier Series). Then "learning" u_θ with natural gradient (and learning rate 1) corresponds to solve the equation in the least-squares sense with a generalized Green's function.

4.4.2 . Proof of Theorem 4.8

In order to prove Theorem 4.8, we need the following result.

Proposition 4.9. *Let $D : \mathcal{H} \to L^2(\Omega \to \mathbb{R}, \mu)$ be a linear differential operator, and $\mathcal{H}_0 := \mathrm{Span}(u_p : 1 \leqslant p \leqslant P) \subset \mathcal{H}$ a subspace isometrically embedded in $\mathcal{H}$. Then the generalized Green's function (see Definition 3.9 in Section 3.2.1 of Chapter 3) of D on $\mathcal{H}_0$ is given by: for all $x, y \in \Omega$*

$$g_{\mathcal{H}_0}(x,y) := \sum_{1 \leqslant p,q \leqslant P} u_p(x)\, G^\dagger_{p,q} D[u_q](y), \tag{4.63}$$

with: for all $1 \leqslant p, q \leqslant P$,

$$G_{p,q} := \langle D[u_p]\,,\, D[u_q] \rangle_{L^2(\Omega \to \mathbb{R}, \mu)}\,. \tag{4.64}$$

Proof. By definition:

$$D[\mathcal{H}_0] = \{D[u] \,:\, u \in \mathcal{H}_0\} = \mathrm{Span}\left(D[u_p] \,:\, 1 \leqslant p \leqslant P\right),$$

Thus Lemma 4.4 applies and yields that: for all $x, y \in \Omega$

$$k(x,y) := \sum_{p,q \in \mathbb{N}} D[u_p](x)\, G^\dagger_{p,q} D[u_q](y), \tag{4.65}$$

with: for all $1 \leqslant p, q \leqslant P$,

$$G_{ij} := \langle D[u_p]\,,\, D[u_q] \rangle_{\mathrm{L}^2(\Omega \to \mathbb{R}, \mu)}\,, \tag{4.66}$$

is the kernel of the projection $\Pi_{D[\mathcal{H}_0]} : \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \to \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$ onto the RKHS $D[\mathcal{H}_0]$. This means that for all $f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$,

$$\begin{aligned}
\Pi_{D[\mathcal{H}_0]}(f)(x) &= \int_\Omega k(x,y) f(y)\, \mu(\mathrm{d}y) && (4.67)\\
&= \int_\Omega \sum_{1 \leqslant p,q \leqslant P} D[u_p](x)\, G^\dagger_{p,q} D[u_q](y) f(y)\, \mu(\mathrm{d}y) && (4.68)\\
&= \sum_{1 \leqslant p,q \leqslant P} D[u_p](x)\, G^\dagger_{p,q} \int_\Omega D[u_q](y) f(y)\, \mu(\mathrm{d}y) && (4.69)\\
&= D\left[\sum_{1 \leqslant p,q \leqslant P} u_p(\cdot)\, G^\dagger_{p,q} \int_\Omega D[u_q](y) f(y)\, \mu(\mathrm{d}y) \right](x), && (4.70)
\end{aligned}$$

where Equation (4.70) comes from the linearity of D. Refering to Definition 3.9 in Section 3.2.1 of Chapter 3, this exactly means that the kernel defined for all $x, y \in \Omega$ by,

$$g_{\mathcal{H}_0}(x,y) := \sum_{p,q} u_p(x)\, G^\dagger_{p,q} D[u_q](y)$$

is the generalized Green's function of the operator D on $\mathcal{H}_0$. □

Proof of Theorem 4.8. This a simple application of Proposition 4.9 to the space $\mathcal{H}_0 = \mathrm{Span}\left(\partial_p u_\theta : 1 \leqslant p \leqslant P\right) = \mathrm{Im}\,\mathrm{d}u_\theta = T_\theta\mathcal{M}$. To conclude with the proof of Equation (4.62), let us note that Equation (3.6) of Definition 3.9 in Section 3.2.1 of Chapter 3, namely,

$$D \circ R_{\mathcal{H}_0} = \Pi_{D[\mathcal{H}_0]} \tag{3.6}$$

can be rewritten as:

$$R_{\mathcal{H}_0} = D^\dagger_{|\mathcal{H}_0} \circ \Pi_{D[\mathcal{H}_0]}. \tag{4.71}$$

Specifically, we have:

$$\mathrm{d}\big(D \circ u\big)^\dagger_\theta \circ \Pi_{D[\mathcal{H}_0]} = \mathrm{d}u^\dagger_\theta \circ D^\dagger \circ \Pi_{D[\mathcal{H}_0]} = \mathrm{d}u^\dagger_\theta \circ R_{\mathcal{H}_0} \tag{4.72}$$

Since, by definition, $R_{\mathcal{H}_0}$ is the operator associated with the Green's function $g_{\mathcal{H}_0}$, this directly implies that the natural gradient of PINNs:

$$\theta_{t+1} \leftarrow \theta_t - \eta\, \mathrm{d}\big((D,B)\circ u\big)^\dagger_{\theta_t} \left(\Pi_{T_{\theta_t}\Gamma}\left(\nabla\mathcal{L}_{\theta_t}\right)\right),$$

can be expressed as:

$$\theta_{t+1} \leftarrow \theta_t - \eta\, \mathrm{d}u^\dagger_{\theta_t} \left(x \in \Omega \mapsto \int_\Omega g_{T_{\theta_t}\mathcal{M}}(x,y) \nabla\mathcal{L}_{\theta_t}(y) \mu(\mathrm{d}y)\right),$$

which concludes. □

***Remark* 4.10.** We could have provided a more direct proof by relying on the results presented in Section 3.4.2 of Chapter 3. However, since we will revisit this point in greater detail in Chapter 5 (and because we originally derived this result prior to the framework introduced in Section 3.4.2 of Chapter 3), we choose to retain the present proof without developing it further.

4.4.3 . Illustration of estimated Green's function for PINNs

Green's function plots of Laplace equation in 2 D

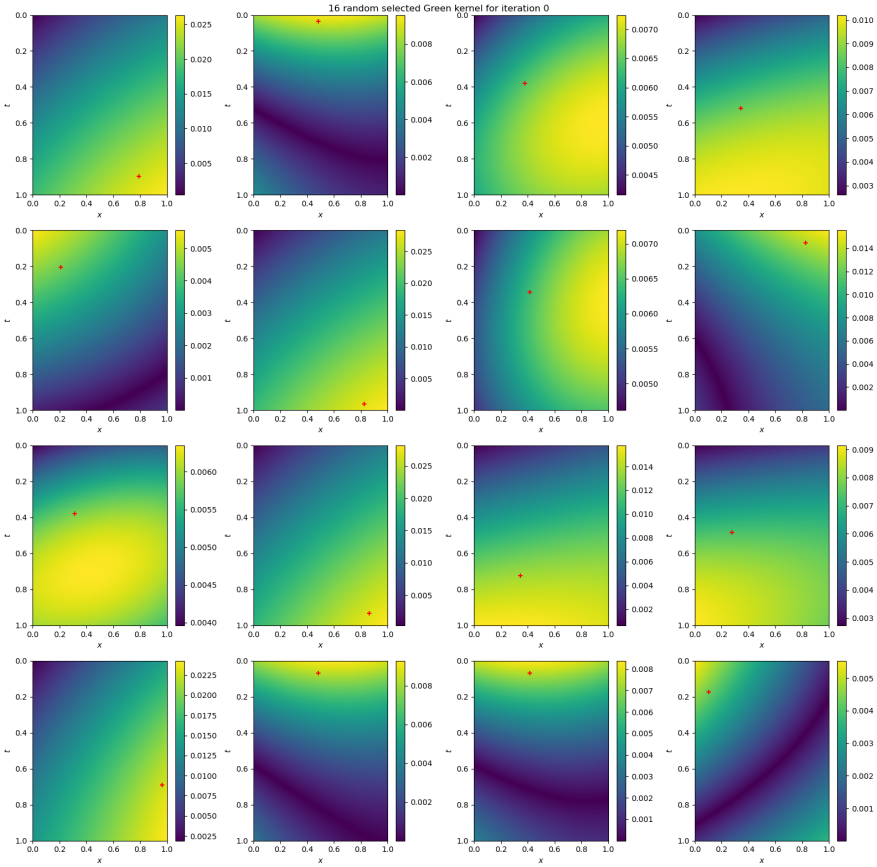


Figure 4.9: Green's function of the operator on the tangent space at initialization for Laplace equation in 2 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $g_{T_{\theta_0}\mathcal{M}}(\cdot, x_i)$

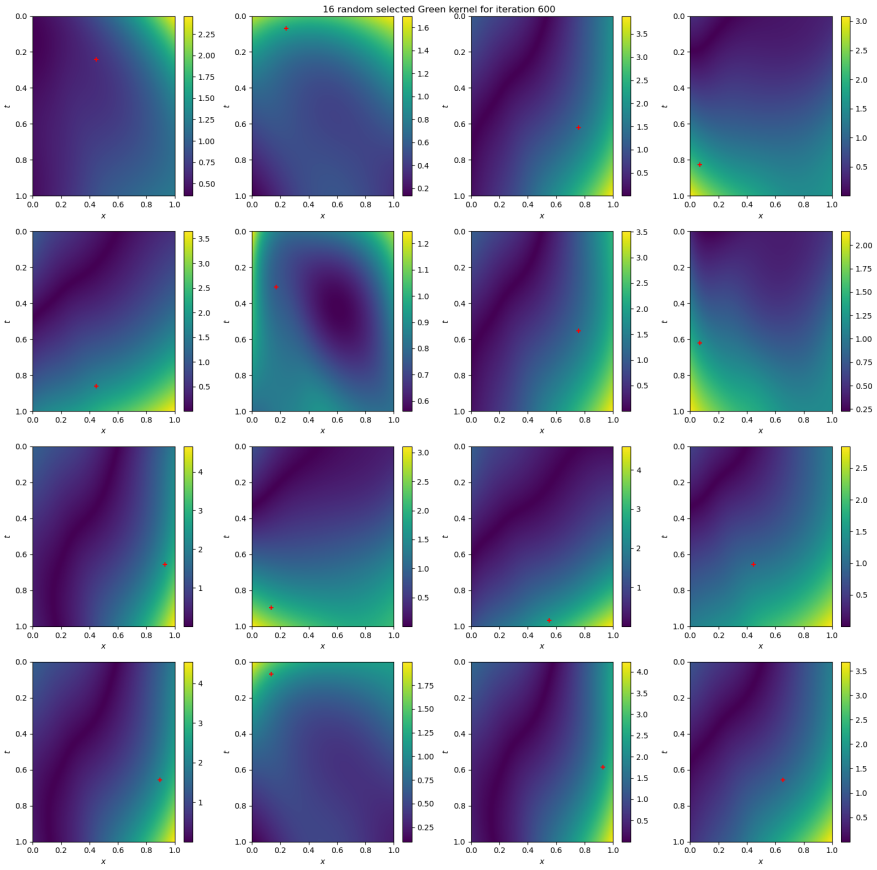


Figure 4.10: Green's function of the operator on the tangent space at the end of optimization for Laplace equation in 2 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $g_{T_{\theta_{\text{end}}}\mathcal{M}}(\cdot, x_i)$

Green's function plots of Heat equation in 1+1 D

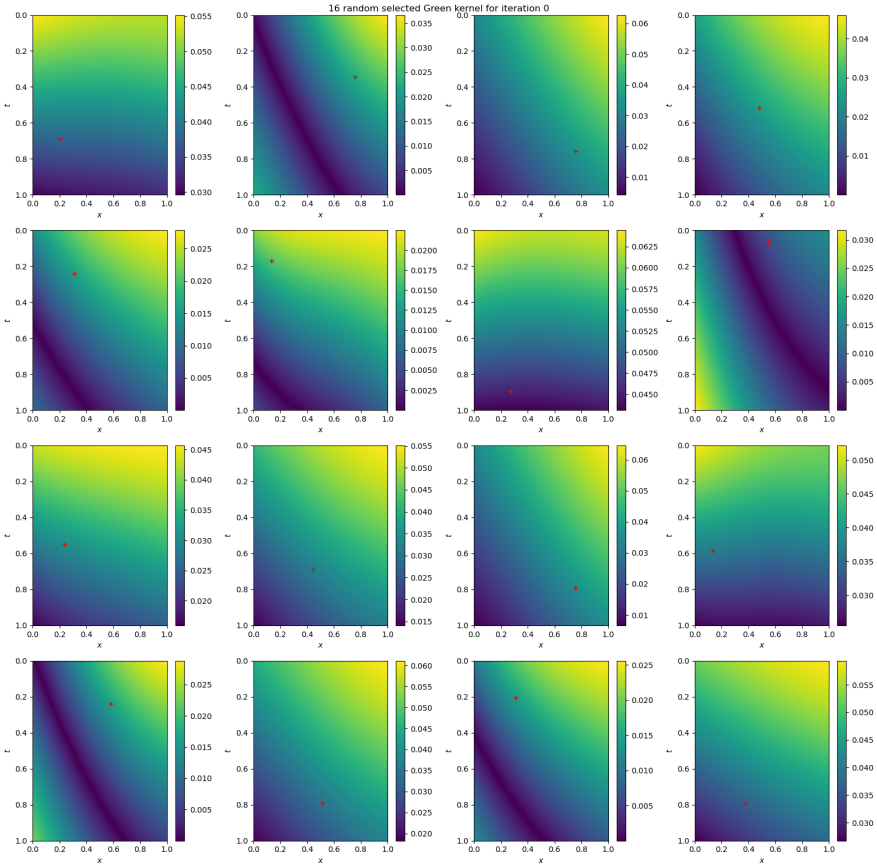


Figure 4.11: Green's function of the operator on the tangent space at initialization for Heat equation in 1+1 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $g_{T_{\theta_0}\mathcal{M}}(\cdot, x_i)$

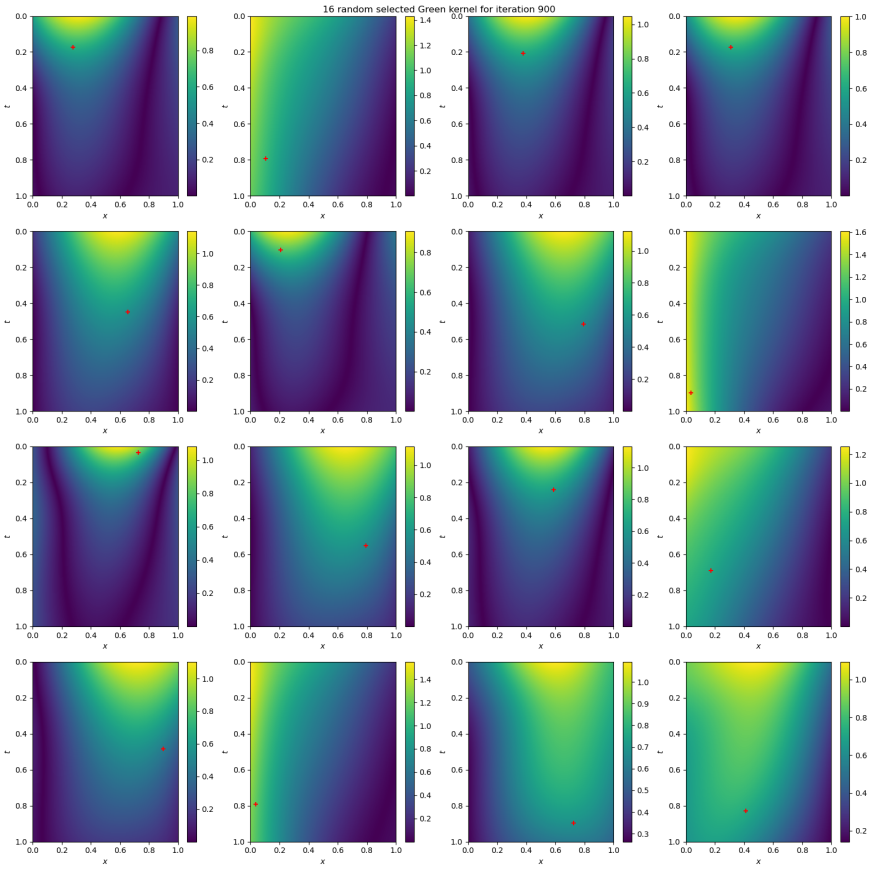


Figure 4.12: Green's function of the operator on the tangent space at the end of optimization for Heat equation in 1+1 D. Reading: the red cross on each subfigure representing a point x_i, the plot represents the function $g_{T_{\theta_{\text{end}}}\mathcal{M}}(\cdot, x_i)$

4.5 . Experiments

We test ANaGRAM on four problems: 2 D Laplace equation ; 1+1 D heat equation ; 5 D Laplace equation ; and 1+1 D Allen-Cahn equation. The first three problems comes from Müller and Zeinhofer (2023), while the last one is proposed in Lu et al. (2021b).

For training, we use multilayer perceptrons with varying layer sizes and tanh activations, along with fixed batches of points: a batch of size S_D to discretize Ω and a batch of size S_B to discretize $\partial\Omega$. The layer size specifications, cutoff factor ϵ, values of S_D and S_B, and discretization procedures are specified separately for each problem. Currently, the cutoff factor is chosen manually and warrants further investigation (see Chapter 5).

For these various problems, we display as a function of gradient descent steps, the medians over 10 different initializations, of:

- **$\mathbf{L^2}$ error $\boldsymbol{E_{L^2}}$.** Given test points $(x_i)_{i=1}^{S}$,

$$E_{\mathrm{L}^2}(\theta) := \sqrt{\frac{1}{S_{\mathrm{L}^2}} \sum_{i=1}^{S_{\mathrm{L}^2}} |u_\theta(x_i) - u^*(x_i)|^2}, \tag{4.73}$$

  where u^* is a known solution to the PDE and S is taken 10 times bigger than the Ω batch size S_D.

- **Test loss $\boldsymbol{E_{\text{test}}}$.** E_{test} is the empirical PINNs loss ℓ of Equation (2.72) in Definition 2.43 of Section 2.2.2 in Chapter 2, applied to the MLP, namely:

$$\begin{aligned} \ell(\theta) := &\frac{1}{2S_D} \sum_{i=1}^{S_D} \left(D[u_\theta](x_i^D) - f(x_i^D)\right)^2 \\ &+ \frac{1}{2S_B} \sum_{i=1}^{S_B} \left(B[u_\theta](x_i^B) - g(x_i^B)\right)^2, \end{aligned} \tag{4.74}$$

  computed with a distinct set of points, of size 5 times bigger than the Ω batch size S_D.

Shaded areas indicate the range between the first and third quartiles.

We compare ANaGRAM to Energy Natural Gradient descent (E-NGD) (Müller and Zeinhofer, 2023), vanilla gradient descent (GD) with line-search, Adam with exponentially decaying learning-rate after 10^{15} steps as in Müller and Zeinhofer (2023) as well as L-BFGS. The corresponding CPU times are also provided in tables for reference. The code is made avaible at `https://github.com/IloneM/ANaGRAM/` and further implementation and computation details are provided in Section A.1.1 of Chapter A.

4.5.1 . 2 D Laplace equation

We consider the two dimensional Laplace equation:

$$\begin{cases} \Delta u = -2\pi^2 \sin(\pi x_1) \sin(\pi x_2) & \text{in } \Omega = [0,1]^2 \\ u = 0 & \text{on } \partial\Omega \end{cases}, \tag{4.75}$$

and its solution:

$$u^*(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2). \tag{4.76}$$

We choose $S_D = 900$ equi-distantly spaced points in the interior of Ω and $S_B = 120$ equally spaced points on the boundary $\partial\Omega$ (30 on each side). ANaGRAM, E-NGD and L-BFGS are applied for 2000 iterations each, while GD and Adam are trained for 20×10^3 iterations. The network consists of a single hidden layer with a width of 32, resulting in a total of $P = 129$ parameters. The cutoff factor is set to $\epsilon = 1 \times 10^{-6}$.

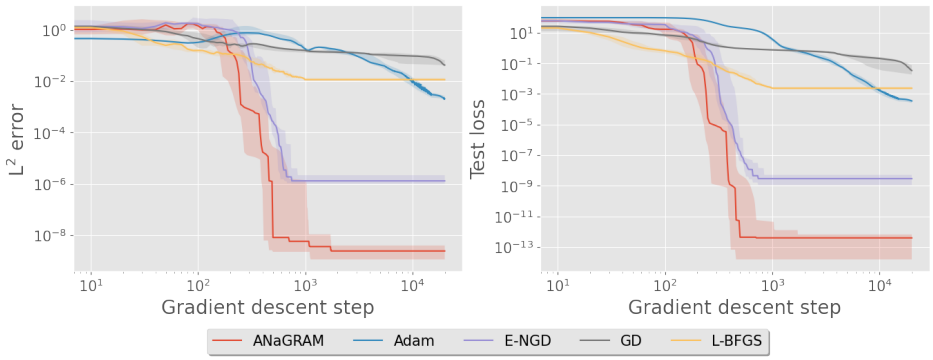


Figure 4.13: Median absolute L^2 errors and Test losses for the 2 D Laplace equation.

Table 4.2: Execution time for the 2 D Laplace equation

CPU time (s)	Per step	Full
ANaGRAM	7.16e-02	**1.25e+02**
Adam	**1.23e-02**	2.44e+02
E-NGD	1.94e-01	1.88e+02
GD	2.07e-02	4.13e+02
L-BFGS	1.95e-01	1.95e+02

4.5.2 . 1+1 D Heat equation

We consider the $(1+1)$ dimensional Heat equation:

$$\begin{cases} \partial_t u - \frac{1}{4}\partial_{xx} u = 0 & \text{in } \Omega = [0,1]^2 \\ u = 0 & \text{on } \partial\Omega_{\text{border}} = [0,1] \times \{0,1\} \\ u(0,x) = \sin(\pi x) & \text{on } \partial\Omega_0 = \{0\} \times [0,1] \end{cases}, \tag{4.77}$$

and its solution:

$$u^*(t,x) = \exp\left(-\frac{\pi^2 t}{4}\right) \sin(\pi x). \tag{4.78}$$

We choose $S_D = 900$ equi-distantly spaced points in the interior of Ω and $S_B = 90$ equally spaced points on the boundary $\partial\Omega$ (30 on $\partial\Omega_0$ and 30 on each side of $\partial\Omega_{\text{border}}$). ANaGRAM, E-NGD and L-BFGS are applied for 2000 iterations each, while GD and Adam are trained for 20×10^3 iterations. The network consists of a single hidden layer with a width of 64, resulting in a total of $P = 257$ parameters. The cutoff factor is set to $\epsilon = 1 \times 10^{-5}$.

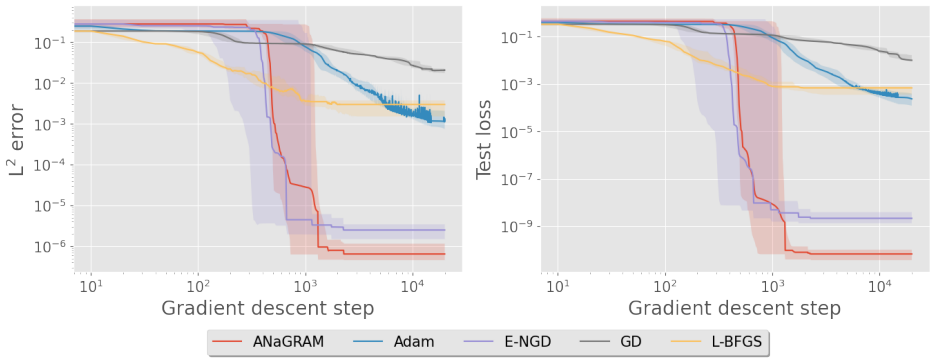


Figure 4.14: Median absolute L^2 errors and Test losses for the Heat equation.

Table 4.3: Execution time for the Heat equation

CPU time (s)	Per step	Full
ANaGRAM	1.29e-01	**3.78e+02**
Adam	**2.12e-02**	4.15e+02
E-NGD	1.78e-01	4.04e+02
GD	3.87e-02	7.68e+02
L-BFGS	1.30e-01	3.91e+02

4.5.3 . 5 D Laplace equation

We consider the five dimensional Laplace equation:

$$\begin{cases} \Delta u = \pi^2 \sum_{k=1}^{5} \sin(\pi x_k) & \text{in } \Omega = [0,1]^5 \\ u = \sum_{k=1}^{5} \sin(\pi x_k) & \text{on } \partial\Omega \end{cases}, \tag{4.79}$$

and its solution:

$$u^*(x) = \sum_{k=1}^{5} \sin(\pi x_k). \tag{4.80}$$

We choose $S_D = 4000$ uniformly drawn points in the interior of Ω and $S_B = 500$ uniformly drawn points on the boundary $\partial\Omega$. ANaGRAM, E-NGD and L-BFGS are applied for 1000 iterations each, while GD and Adam are trained for 20×10^3 iterations. The network consists of a single hidden layer with a width of 64, resulting in a total of $P = 449$ parameters. The cutoff factor is set to $\epsilon = 5.10^{-7} \times \Delta_{\theta \max}$, where $\Delta_{\theta \max}$ is the maximal eigenvalue of $\widehat{\phi}_\theta$ (*cf.* line 1 of algorithm 3).

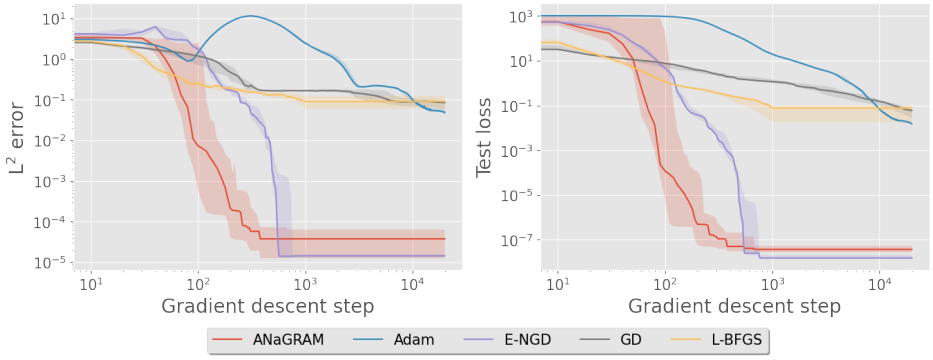


Figure 4.15: Median absolute L^2 errors and Test losses for the 5 D Laplace equation.

Table 4.4: Execution time for the 5 D Laplace equation

CPU time (s)	Per step	Full
ANaGRAM	7.18e-01	4.88e+02
Adam	**6.65e-02**	1.29e+03
E-NGD	6.52e+00	4.96e+03
GD	2.69e-01	5.38e+03
L-BFGS	2.96e-01	**2.96e+02**

4.5.4 . 1+1 D Allen-Cahn equation

We consider the $(1+1)$ dimensional Allen-Cahn equation:

$$\begin{cases} \partial_t u - 10^{-3}\,\partial_{xx} u - 5(u-u^3) = 0 & \text{in } \Omega = [0,1]\times[-1,1] \\ u = -1 & \text{on } \partial\Omega_{\text{border}} = [0,1]\times\{-1,1\}\,. \\ u(0,x) = x^2\cos(\pi x) & \text{on } \partial\Omega_0 = \{0\}\times[-1,1] \end{cases} \tag{4.81}$$

A closed-form solution is not known; we therefore rely on an approximate one obtained through a classical numerical solver.

We choose $S_D = 900$ equi-distantly spaced points in the interior of Ω and $S_B = 90$ equally spaced points on the boundary $\partial\Omega$ (30 on $\partial\Omega_0$ and 30 on each side of $\partial\Omega_{\text{border}}$). ANaGRAM and L-BFGS are applied for 4000 iterations each, E-NGD for 1000 iterations, while classical gradient descent (GD) and Adam are trained for 50×10^3 iterations. The network consists of three hidden layers with a width of 20, resulting in a total of $P = 921$ parameters. The cutoff factor is set to $\epsilon = 5.10^{-7} \times \Delta_{\theta\text{max}}$, where $\Delta_{\theta\text{max}}$ is the maximal eigenvalue of $\widehat{\phi}_\theta$ (*cf.* line 1 of algorithm 3).

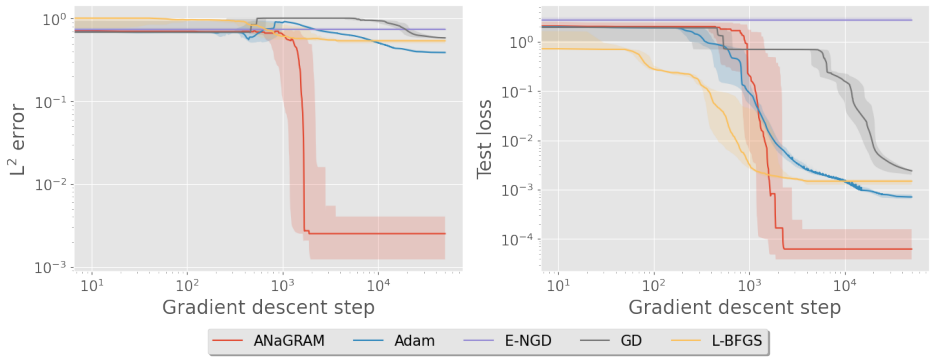


Figure 4.16: Median absolute L^2 errors and Test losses for the Allen-Cahn equation.

Table 4.5: Execution time for the 1+1 D Allen-Cahn

CPU time (s)	Per step	Full
ANaGRAM	6.01e-01	2.16e+03
Adam	**2.82e-02**	**1.18e+03**
E-NGD	1.30e+00	6.52e+03
GD	8.59e-02	4.28e+03
L-BFGS	4.07e-01	1.60e+03

Results summary : We demonstrated that our approach can achieve comparable accuracy to Müller and Zeinhofer (2023) on linear problems, consistent with the equivalence established in Section A.2 of Chapter A, while maintaining a per-step computational cost at most reasonably higher than that of Adam. Excluding Adam and GD, which consistently get stuck at high error levels, the bottom line is that ANaGRAM consistently outperforms both E-NGD and L-BFGS, often by a significant margin, on at least one or even both criteria: precision and computation time. The cases where the computation times of E-NGD and ANaGRAM are similar occur when small-sized architectures are sufficient for the problem.

4.6 . Conclusion and Perspectives

We introduced empirical Natural Gradient, a new kind of natural gradient that scales linearly with respect to the number of parameters and extend it to PINNs framework through a mathematically principled reformulation. We show that this update implicitly corresponds to the use of the Green's function of the operator. We give empirical evidences that this optimization in

its simplest form (vanilla ANaGRAM) already achieves highly accurate solutions, comparable to Müller and Zeinhofer (2023) for linear PDEs at a fraction of the computational cost, and with significant improvements for non-linear equations, for which equivalence of the two algorithms does not hold anymore.

Still, the present formulation of the algorithm has two limitations: one concerns the chosing procedure of the batch points, which is so far limited to simple heuristics; the second is the hyperparameter tunning, more specifically the cutoff factor (see Chapter 5), which is so far chosen by hand, while it may probably be automatically chosen based on the spectrum of $\widehat{\phi}_\theta$.

Important perspectives include exploring approximations schemes for terms E_θ^{metric} (*e.g.* using Nyström's methods, *cf.* Sun et al. (2015)) and $E_\theta^\perp$ (*e.g.* using Cohen and Migliorati (2017)), introduced in Theorem 4.3, the design of an optimal collocation points procedure, coupled with SVD cut-off factor adaptation strategy for ANaGRAM, as well as incorporation of common optimization techniques, such as momentum. From a theoretical point of view, it seems particularly important to us to include data assimilation in this theoretical setting, and understand its regularizing effect, while establishing connections to classical solvers such as FEMs.

5 - AMStraMGRAM: Adaptive Multi-cutoff Strategy Modification for ANaGRAM

This chapter presents the results developed in our paper Schwencke et al. (2025).

In this paper, we study the performance and training dynamics of ANaGRAM, with a particular emphasis on the role of the chosen cutoff. Typically, the training loss of ANaGRAM exhibits slow convergence during the early iterations, followed by a sudden drop near the end of training, a behavior also observed in the E-NGD method (Müller and Zeinhofer, 2023). We show that this behavior is closely related to what we refer to as the *flattening phenomenon*, which we define and characterize through the *reconstruction error*, a novel metric that quantifies the fraction of the loss signal discarded under different cutoff choices. Building on this observation, our adaptive multi-cutoff strategy, AMStraMGRAM, effectively exploits this phenomenon, yielding improvements of several orders of magnitude across various PDE benchmarks. We precede this empirical analysis with a functional-analytic perspective linking cutoff (and ridge) regularization to the theory of generalized Green operators. This connection clarifies that cutoff regularization is not merely a numerical trick, but rather an essential theoretical component for stable and accurate training.

5.1 . Position of the Problem

5.1.1 . Parametric Models and Neural Networks

As in Chapter 4, we start by recalling the objects we introduced in Sections 2.1.1 and 2.1.2 of Chapter 2.

Parametric model: A differentiable mapping $u : \mathbb{R}^P \to \mathcal{H}$, as defined in Definition 2.8.

Differential of u: The differential of u at θ, denoted $\mathrm{d}u_\theta$, as defined in Definition 2.10.

In particular, following the manifold approximation perspective introduced in Section 2.1.2, we define

The image set of u : this is the set of functions reached by u, *i.e.* :

$$\mathcal{M} := \operatorname{Im} u := \left\{ u_\theta \,:\, \theta \in \mathbb{R}^P \right\} \subset \mathcal{H}. \tag{5.1}$$

The tangent space of u at θ : this is the image set of the differential of u at θ, *i.e.* the linear subspace of $\mathcal{H}$ consisting of the functions reached by $\mathrm{d}u_\theta$, *i.e.* :

$$T_\theta\mathcal{M} := \operatorname{Im} \mathrm{d}u_\theta = \operatorname{Span}\left(\partial_p u_\theta \,:\, 1 \leqslant p \leqslant P\right) \subset \mathcal{H}. \tag{5.2}$$

5.1.2 . Physics-Informed Neural Networks (PINNs)

As in Section 2.1.6 of Chapter 2, we consider a domain $\Omega \subset \mathbb{R}^d$ equipped with a probability measure μ, and its boundary $\partial\Omega$ equipped with a probability surface measure σ. We introduce two linear differential operators:

$$D : \left\{ \begin{array}{ccc} \mathcal{H} & \to & \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \\ u & \mapsto & D[u] \end{array} \right. ; \quad B : \left\{ \begin{array}{ccc} \mathcal{H} & \to & \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma) \\ u & \mapsto & B[u] \end{array} \right. , \tag{5.3}$$

where D denotes a differential operator, and B represents a boundary operator.

The corresponding partial differential equation (PDE) reads:

$$\begin{cases} D(u) = f \in \mathrm{L}^2(\Omega, \mu), & \text{in } \Omega, \\ B(u) = g \in \mathrm{L}^2(\partial\Omega, \sigma), & \text{on } \partial\Omega. \end{cases} \tag{5.4}$$

Given a parametric model $u : \mathbb{R}^P \to \mathcal{H}$, a PINN approximates the solution of Equation (5.4) by minimizing the empirical quadratic loss

$$\ell(\theta) := \frac{1}{2S_D} \sum_{i=1}^{S_D} \big(D[u_\theta](x_i^D) - f(x_i^D)\big)^2 + \frac{1}{2S_B} \sum_{i=1}^{S_B} \big(B[u_\theta](x_i^B) - g(x_i^B)\big)^2 . \tag{5.5}$$

5.1.3 . Natural Gradient

Following Section 4.3.1 in Chapter 4, the PINN optimization problem can, without loss of generality, be formulated as a functional least-squares problem in $\mathrm{L}^2(\Omega, \mu)$. Given $f \in \mathrm{L}^2(\Omega, \mu)$, we define the corresponding functional loss as

$$\mathcal{L}(u) := \frac{1}{2} \left\| u - f \right\|^2_{\mathrm{L}^2(\Omega,\mu)} . \tag{5.6}$$

Its discretized counterpart, the empirical loss, is then given by

$$\ell(\theta) := \frac{1}{2S} \sum_{i=1}^{S} \left(u_\theta(x_i) - f(x_i)\right)^2 . \tag{5.7}$$

As detailed in Section 4.1.3 of Chapter 4, the associated natural gradient descent then takes the form

$$\theta_{t+1} \leftarrow \theta_t - \eta \, \mathrm{d}u^\dagger_{\theta_t}\big(\Pi_{T_{\theta_t}\mathcal{M}}(\nabla\mathcal{L}_{|u_{\theta_t}})\big), \tag{5.8}$$

where $\eta > 0$ is the learning rate and $\mathrm{d}u^{\dagger}_{\theta_t}$ denotes the pseudo-inverse of the differential $\mathrm{d}u_{\theta_t}$. According to Proposition 2.37 in Section 2.1.5 of Chapter 2, this update is equivalently written in matrix form as

$$\theta_{t+1} \leftarrow \theta_t - \eta\, G^{\dagger}_{\theta_t} \nabla_{\theta_t} \mathcal{L} \circ u, \tag{5.9}$$

where $G_{\theta_t} \in \mathbb{R}^{P \times P}$ is the Gram matrix defined by

$$(G_{\theta_t})_{p,q} = \langle \partial_p u_{\theta_t} \,,\, \partial_q u_{\theta_t} \rangle_{\mathrm{L}^2(\Omega,\mu)} \,. \tag{5.10}$$

5.1.4 . Empirical Natural Gradient

Direct computation of Equation (5.9) is often impractical due to the $O(P^3)$ computational cost associated with inverting G_{θ_t}. The *Empirical Natural Gradient* (ENG), introduced in Section 4.2.1 of Chapter 4, addresses this limitation by replacing the matrix inversion with the following update rule:

$$\theta_{t+1} \leftarrow \theta_t - \eta\, \widehat{\phi}^{\dagger} \widehat{\nabla \mathcal{L}}_{\theta_t}; \quad \widehat{\phi}_{i,p} := \partial_p u_\theta(x_i); \quad \left(\widehat{\nabla \mathcal{L}}_{\theta}\right)_i := u_\theta(x_i) - f(x_i), \tag{5.11}$$

where $\widehat{\phi}^{\dagger} = \widehat{U}\widehat{\Delta}^{\dagger}\widehat{V}^T$ is obtained from the singular value decomposition (SVD) of $\widehat{\phi} = \widehat{V}\widehat{\Delta}\widehat{U}^T$. This formulation reduces the computational cost to $O(\min(P^2 S, P S^2))$, while preserving the geometric structure of the natural gradient. A comparable level of efficiency was later achieved using a Cholesky factorization approach by Guzmán-Cordero et al. (2025).

In what follows, we adopt a slight abuse of notation by omitting the explicit dependence on θ whenever it is clear from context. When iteration indices matter, we explicitly write t to emphasize the connection to θ_t.

5.2 . Regularization

We distinguish between two main regularization schemes:

(i) *ridge regression*, which consists in adding a factor $\alpha^2 I_d$ (or, according to conventions, $\alpha^{-2} I_d$) to the Gram matrix G_θ in Equation (5.9) (or its approximation $\widehat{\mathcal{G}}_\theta$), thereby making it invertible

(ii) *cutoff regularization*, a scheme that applies a binary threshold (used in ANaGRAM):

$$\widehat{\Delta}^{\dagger}_{t,i} = \begin{cases} \widehat{\Delta}^{-1}_{t,i}, & \text{if } \widehat{\Delta}_{t,i} \geqslant \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{5.12}$$

Here α denotes the cutoff threshold. This regularization is the focus of our analysis in Section 5.4

In this section, we provide a geometric interpretation of each scheme and further show that cutoff regularization extends previously established connections between natural gradient methods and Green's function theory (Schwencke and Furtlehner, 2025).

5.2.1 . Why Regularization is Necessary

We recall that our goal is to solve the operator equation $D[u] = f$ by minimizing the squared residual

$$\|D[u] - f\|^2_{\mathrm{L}^2(\Omega,\mu)} \,. \tag{5.13}$$

For simplicity, assume D is linear. Then the mapping

$$u \in \mathcal{C}^\infty(\Omega) \longmapsto \|D[u]\|_{\mathrm{L}^2(\Omega,\mu)} \tag{5.14}$$

defines a semi-norm on $\mathcal{C}^\infty(\Omega)$. In fact (see Berezansky et al. 1996b, Chapter 12), we can "upgrade" this semi-norm into a true norm by introducing the following generalized Sobolev norm:

$$\|\cdot\|_{\widetilde{\mathcal{H}}_D} : \left\{ \begin{array}{lcl} \mathcal{C}^\infty(\Omega \to \mathbb{R}) & \to & \mathbb{R}^+ \\ u & \mapsto & \sqrt{\|u\|^2_{\mathrm{L}^2(\Omega\to\mathbb{R},\mu)} + \|D[u]\|^2_{\mathrm{L}^2(\Omega\to\mathbb{R},\mu)}} \end{array} \right. \tag{5.15}$$

Clearly, for any u,

$$\|u\|_{\mathrm{L}^2(\Omega,\mu)} \leqslant \|u\|_{\widetilde{\mathcal{H}}_D} \,, \tag{5.16}$$

which guarantees that $\|\cdot\|_{\widetilde{\mathcal{H}}_D}$ is *definite*, i.e. $\|u\|_{\widetilde{\mathcal{H}}_D} = 0 \iff u = 0$.

Completing $\mathcal{C}^\infty(\Omega)$ with respect to $\|\cdot\|_{\widetilde{\mathcal{H}}_D}$ yields a generalized Sobolev space $\left(\mathcal{H}_D, \|\cdot\|_{\mathcal{H}_D}\right)$. This Hilbert space is the largest subspace of $\mathrm{L}^2(\Omega,\mu)$ on which D is continuous. Indeed, for every $u \in \mathcal{H}_D$,

$$\|D[u]\|_{\mathrm{L}^2(\Omega,\mu)} \leqslant \|u\|_{\mathcal{H}_D} \,. \tag{5.17}$$

Since our goal is to solve $D[u] = f$, we need D to be continuously invertible. That is, we need the reverse inequality of Equation (5.17) to hold (up to a constant $\alpha > 0$). Formally, if D were algebraically invertible (bijective as a mapping), this condition would read:

$$\begin{array}{rl} & \left(\exists \alpha > 0,\, \forall u \in \mathcal{H}_D,\, \|u\|_{\mathcal{H}_D} \leqslant \alpha \, \|D[u]\|_{\mathrm{L}^2(\Omega\to\mathbb{R},\mu)}\right) \\ \iff & \left(\exists \alpha > 0,\, \forall u \in \mathcal{H}_D,\, \left\|D^{-1}[D[u]]\right\|_{\mathcal{H}_D} \leqslant \alpha \, \|D[u]\|_{\mathrm{L}^2(\Omega\to\mathbb{R},\mu)}\right) \\ \iff & \left(\exists \alpha > 0,\, \forall f \in \mathrm{L}^2(\Omega \to \mathbb{R},\mu),\, \left\|D^{-1}[f]\right\|_{\mathcal{H}_D} \leqslant \alpha \, \|f\|_{\mathrm{L}^2(\Omega\to\mathbb{R},\mu)}\right) \end{array} \,. \tag{5.18}$$

Operator ill-conditioning. Even if D is bijective, Equation (5.18) may fail to hold, i.e. D can be ill-conditioned. Suppose there exists a subspace $\mathcal{H}_K \subset \mathcal{H}_D$ such that D acts compactly on $\mathcal{H}_K$ with infinite rank. Then D admits a singular value decomposition (Kress, 2014, Theorem 15.16): for $u \in \mathcal{H}_K$,

$$D[u] = \sum_{n\in\mathbb{N}} e_n \lambda_n \langle v_n \,,\, u \rangle_{\mathcal{H}_D} \,, \tag{5.19}$$

with (v_n) orthonormal in $\mathcal{H}_D$, (e_n) orthonormal in $\mathrm{L}^2(\Omega, \mu)$, and $\lambda_n \to 0$ as $n \to \infty$.

For Equation (5.18) to hold, we would need $\inf_n \lambda_n > 0$, contradicting $\lambda_n \to 0$. This is exactly the classical inverse problem setting: D is bijective but ill-conditioned, and regularization is unavoidable. Among the many schemes developed, Tikhonov regularization is the canonical example (Kirsch, 2021).

Non-bijectivity. If D is not bijective, two additional issues may occur.

Non-surjectivity. If $\operatorname{Im} D$ is a closed subspace, we can still obtain a solution by replacing the target f with its projection $\Pi_{\operatorname{Im} D} f$. Note that minimizing $\|D[u] - f\|^2_{\mathrm{L}^2(\Omega,\mu)}$ yields precisely this least-squares solution.

Non-injectivity. The lack of injectivity is a much more subtle issue. Since D is linear and continuous, its null space $\operatorname{Ker} D$ is a closed subspace of $\mathcal{H}_D$. In principle, one could restrict the domain of D to $\operatorname{Ker} D^\perp$ to make it injective. The problem, however, is that identifying $\operatorname{Ker} D$ is typically just as hard as solving the original problem itself, since it amounts to characterizing all $u \in \mathcal{H}_D$ such that $D[u] = 0$. Therefore, unless one can rely on theoretical results that explicitly describe $\operatorname{Ker} D$, or construct a subspace $\mathcal{H}_0 \subset \mathcal{H}_D$ for which $\operatorname{Ker} D \cap \mathcal{H}_0$ is explicitly known (so that D can be restricted to $\mathcal{H}_0$), it is generally impossible to "get rid of" $\operatorname{Ker} D$ in practice.

On the other hand, if we do not filter out $\operatorname{Ker} D$, this has the unwanted consequence of introducing "spurious" low-energy signals. To be concrete, suppose we approximate our solution in a space $\mathcal{H}_K$ with orthonormal basis $(u_n)_{n\in\mathbb{N}}$. Assume there exists a subsequence $(u_n^S) \notin \operatorname{Ker} D$ converging towards $\operatorname{Ker} D$. Since $\operatorname{Ker} D$ is closed (by continuity of D), this means

$$\lim_{n\to\infty} \left\|\Pi_{\operatorname{Ker} D} u_n^S - u_n^S\right\|^2_{\mathcal{H}_D} = 0. \tag{5.20}$$

In particular, since (u_n^S) is orthonormal, it follows, after extraction, that for all $n \in \mathbb{N}$,

$$\left\|\Pi_{\operatorname{Ker} D} u_n^S\right\|^2_{\mathcal{H}_D} \geqslant 1 - 2^{-n}. \tag{5.21}$$

Thus we have,

$$
\begin{aligned}
0 < \|D\big[u_n^S\big]\|_{\mathcal{H}_D}^2 &= \big\|D\big[\Pi_{\operatorname{Ker} D^\perp} u_n^S + \Pi_{\operatorname{Ker} D} u_n^S\big]\big\|_{\mathcal{H}_D}^2 \\
&= \Big\|D\big[\Pi_{\operatorname{Ker} D^\perp} u_n^S\big] + \underbrace{D\big[\Pi_{\operatorname{Ker} D} u_n^S\big]}_{=0}\Big\|_{\mathcal{H}_D}^2 \\
&= \big\|D\big[\Pi_{\operatorname{Ker} D^\perp} u_n^S\big]\big\|_{\mathcal{H}_D}^2 \\
&\overset{(5.17)}{\leqslant} \big\|\Pi_{\operatorname{Ker} D^\perp} u_n^S\big\|_{\mathcal{H}_D}^2 \\
&= 1 - \|\Pi_{\operatorname{Ker} D} u_n^S\|_{\mathcal{H}_D}^2 \\
&\overset{(5.21)}{\leqslant} 2^{-n}.
\end{aligned}
\tag{5.22}
$$

In particular, if (for simplicity) the vectors (u_n) are right singular vectors of D, then the vectors (u_n^S) will correspond to singular values vanishing at least as fast as (2^{-n}). Crucially, however, these vanishing singular values do not reflect an intrinsic ill-conditioning of D, but rather an *artificial* ill-conditioning induced by the choice of approximation space $\mathcal{H}_K$. In other words, the spurious instability arises from how we approximate the operator, not from the operator itself. For more details on this approximation-induced phenomenon, see Adcock and Huybrechs (2019, 2020b).

These remarks highlight the *inevitable need for regularization* in practice. In the next section, we will provide a geometric interpretation of the two regularization schemes introduced at the beginning of this section, emphasizing how fundamentally different they are in nature.

***Remark* 5.1.** The above discussion becomes even more critical when we restrict ourselves to a finite-dimensional approximation space $\mathcal{H}_{\text{app}} \subset \mathcal{H}_D$. In this case, the restriction D_{app} is automatically compact, since it is of finite rank. As a consequence, both types of ill-conditioning described above may occur simultaneously. This highlights once again why regularization is not merely convenient but *unavoidable* in numerical practice.

5.2.2 . Ridge-regression

Returning to the definition given at the beginning of this section, recall that Ridge regression amounts to adding $\alpha^2 I_d$ (for some $\alpha > 0$) to the Gram matrix G_θ introduced in Equation (5.9), namely,

. (5.9)

We can reformulate this observation in the following way: given a parametric model,

$$u : \mathbb{R}^P \to \mathrm{L}^2(\Omega \to \mathbb{R}, \mu), \tag{5.23}$$

consider the *regularized model*

$$u^\alpha : \left\{ \begin{array}{ccc} \mathbb{R}^P & \to & \mathrm{L}^2(\Omega,\mu) \times \mathbb{R}^P \\ \theta & \mapsto & (u_\theta, \alpha\,\theta). \end{array} \right. \tag{5.24}$$

The Gram matrix of this regularized model is exactly $G_\theta + \alpha^2 I_d$. Suppose further that regression is performed with respect to some function $f \in \mathrm{L}^2(\Omega,\mu)$. Then we must adapt the objective to the regularized model, replacing f with the pair

$$(f, \alpha\,\theta) \in \mathrm{L}^2(\Omega,\mu) \times \mathbb{R}^P. \tag{5.25}$$

A straightforward computation shows that, for all $1 \leqslant p \leqslant \min(P,S)$,

$$\begin{aligned} \left\langle \partial_p u^\alpha_\theta \,,\, (f, \alpha\,\theta) - u^\alpha_\theta \right\rangle_{\mathrm{L}^2(\Omega,\mu)\times\mathbb{R}^P} &= \left\langle \partial_p u_\theta \,,\, f - u_\theta \right\rangle_{\mathrm{L}^2(\Omega,\mu)} + \underbrace{\alpha \left\langle \boldsymbol{e}_p \,,\, \theta - \theta \right\rangle_{\mathbb{R}^P}}_{=0} \\ &= \left\langle \partial_p u_\theta \,,\, f - u_\theta \right\rangle_{\mathrm{L}^2(\Omega,\mu)}, \end{aligned} \tag{5.26}$$

with $(\boldsymbol{e}_p)_{1\leqslant p\leqslant P}$ denoting the canonical basis of $\mathbb{R}^P$. Thus, regression of $(f, \alpha\,\theta)$ with the regularized model is exactly equivalent to Ridge regression. Equivalently, Ridge regression corresponds to replacing the original model u by the regularized model u^α, and replacing the objective f by $(f, \alpha\,\theta)$. From this point of view, the choice of $\alpha\,\theta$ as the secondary target may be interpreted as a *default assumption* in the absence of prior information on the parameters: one simply uses the current parameters as a reference target.

We can now extract several fundamental facts:

1. As $\alpha \to 0$, the regularized model u^α tends in operator norm to the unregularized model $(u, 0)$ (i.e. u by abuse of notation). Indeed,

$$\sup_{\|\theta\|_{\mathbb{R}^P}=1} \|\mathrm{d}u^\alpha_\theta - (\mathrm{d}u_\theta, 0)\|_{\mathrm{L}^2(\Omega,\mu)\times\mathbb{R}^P} = \alpha \sup_{\|\theta\|_{\mathbb{R}^P}=1} \|\theta\|_{\mathbb{R}^P} = \alpha. \tag{5.27}$$

2. The model u^α is injective and continuous. Since $\mathrm{d}u_\theta$ is continuous (as $\mathbb{R}^P$ is finite-dimensional), the only possible source of non-injectivity is $\operatorname{Ker} \mathrm{d}u^\alpha_\theta$. But

$$\operatorname{Ker} \mathrm{d}u^\alpha_\theta = \operatorname{Ker} \mathrm{d}u_\theta \cap \operatorname{Ker}(\alpha I_{\mathbb{R}^P}) \subset \operatorname{Ker}(\alpha I_{\mathbb{R}^P}) = \{0\}, \tag{5.28}$$

   hence injectivity. Restricting u^α to its image makes it algebraically bijective, and the inverse is continuous since

$$\alpha\,\|\theta\|_{\mathbb{R}^P} \leqslant \|\mathrm{d}u^\alpha_\theta\|_{\mathrm{L}^2(\Omega,\mu)\times\mathbb{R}^P}. \tag{5.29}$$

   By the equivalence stated in Equation (5.18), this implies that $(\mathrm{d}u^\alpha_\theta)^{-1}$ is continuous. Consequently, $\operatorname{Im} \mathrm{d}u^\alpha_\theta$ is closed in $\mathrm{L}^2(\Omega,\mu)\times\mathbb{R}^P$, since it is the inverse image of a closed set under $(\mathrm{d}u^\alpha_\theta)^{-1}$. Therefore least-squares solution is well-defined.

3. The least-squares solution of $u^\alpha = (f, 0)$ is influenced by α as follows: $(f, 0)$ is projected onto

$$\operatorname{Im} \mathrm{d}u_\theta^\alpha = \operatorname{Span}\big((\partial_p u_\theta, \alpha e_p) : 1 \leqslant p \leqslant P\big). \tag{5.30}$$

In particular, even if $f \in \operatorname{Im} \mathrm{d}u_\theta$ and $f \neq 0$, we still have $(f, 0) \notin \operatorname{Im} \mathrm{d}u_\theta^\alpha$ (since $\mathrm{d}u_\theta(0) = 0$). Consequently,

$$\Big(\Pi_{\operatorname{Im} \mathrm{d}u_\theta^\alpha}(f, 0)\Big)_1 \neq f, \tag{5.31}$$

where the subscript 1 denotes projection onto the first component in $\mathrm{L}^2(\Omega, \mu) \times \mathbb{R}^P$.

We illustrate these phenomena in Figure 5.1a.

Building on the preceding analysis, we now show how ridge regression can be extended to the functional setting. To this end, let us consider again the operator $D : \mathcal{H}_D \to \mathrm{L}^2(\Omega, \mu)$ introduced in Section 5.2.1. Analogously to what we did for the parametric model u, we define the *regularized operator* at level $\alpha > 0$ as

$$D^\alpha : \left\{ \begin{array}{ccc} \mathcal{H}_D & \to & \mathrm{L}^2(\Omega, \mu) \times \mathcal{H}_D \\ u & \mapsto & (D[u], \alpha u) \end{array} \right. . \tag{5.32}$$

The corresponding target becomes the *regularized objective* $(f, \alpha u)$.

At this level of generality, the equivalence with Gram-matrix regularization no longer holds, since we are dealing with infinite-dimensional operators for which no direct Gram-matrix representation exists. Nevertheless, the fundamental properties remain valid, namely:

1. When $\alpha \to 0$, the regularized operator D^α converges to $(D, 0)$ in the operator-norm sense, i.e. to D by a mild abuse of notation. Indeed, we have

$$\sup_{\|u\|_{\mathcal{H}_D}=1} \|D^\alpha[u] - (D, 0)\|_{\mathrm{L}^2(\Omega,\mu)\times\mathcal{H}_D} = \alpha \sup_{\|u\|_{\mathcal{H}_D}=1} \|u\|_{\mathcal{H}_D} = \alpha. \tag{5.33}$$

2. The operator D^α is injective and continuous. Indeed, D is continuous by the very construction of $\mathcal{H}_D$ (see Section 5.2.1), and injectivity follows since

$$\operatorname{Ker} D^\alpha = \operatorname{Ker} D \cap \operatorname{Ker}(\alpha I_{\mathcal{H}_D}) \subseteq \operatorname{Ker}(\alpha I_{\mathcal{H}_D}) = \{0\}. \tag{5.34}$$

Restricting D^α to its image makes it algebraically bijective, and the inverse is continuous: we have $\alpha \|u\|_{\mathcal{H}_D} \leqslant \|D^\alpha[u]\|_{\mathrm{L}^2(\Omega,\mu)\times\mathcal{H}_D}$, which by the equivalence in Equation (5.18) implies that $\big(D^\alpha\big)^{-1}$ is continuous. Consequently, $\operatorname{Im} D^\alpha$ is closed in $\mathrm{L}^2(\Omega, \mu) \times \mathcal{H}_D$, since it is the inverse image of a closed set under $(D^\alpha)^{-1}$. Therefore least-squares solution is well-defined.

3. Least-squares solutions of the regularized problem $D^\alpha[u] = (f, 0)$ are impacted by α in the following way: we are projecting $(f, 0)$ onto

$$\operatorname{Im} D^\alpha = \operatorname{Span}\Big((D[h], \alpha h) \,:\, h \in \mathcal{H}_D \Big). \tag{5.35}$$

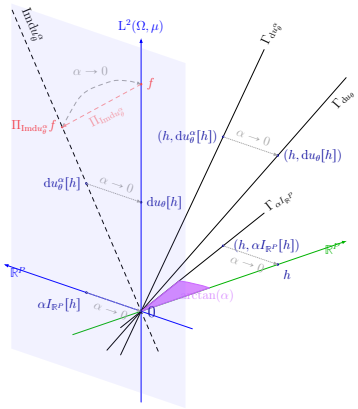
In particular, even if $f \in \operatorname{Im} D$ with $f \neq 0$, we have $(f, 0) \notin \operatorname{Im} D^\alpha$ (since $D[0] = 0$), and hence

$$\big(\Pi_{\operatorname{Im} D^\alpha}(f, 0)\big)_1 \neq f, \tag{5.36}$$

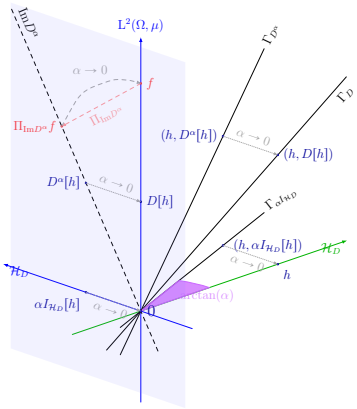
where the subscript 1 denotes the first coordinate in $\mathrm{L}^2(\Omega, \mu) \times \mathcal{H}_D$.

We illustrate these phenomena in Figure 5.1b.

In summary, Ridge regression can be interpreted as a modification of the operator D, rendering it injective and continuously invertible on its image. However, this comes at a price: the regularized solutions are *never* exact solutions of the original equation $D[u] = f$, even when α is arbitrarily small, since we are in fact solving a different operator equation. This marks a fundamental distinction from cutoff regularization, which instead acts directly on the approximation space, as we shall see in the next section.



(a) **Illustration of parametric Ridge regression.**
The green region represents the solution space, while the blue regions denote the target spaces. As $\alpha \to 0$, the regularized graph $\Gamma_{\mathrm{d}u_\theta^\alpha}$ of $\mathrm{d}u_\theta^\alpha$ approaches the graph $\Gamma_{\mathrm{d}u_\theta}$ of $\mathrm{d}u_\theta$, with the angle between them vanishing at rate $\arctan(\alpha)$. The key consequence is that the projection of the objective f onto $\operatorname{Im} \mathrm{d}u_\theta^\alpha$ follows a non-linear path as $\alpha \to 0$, coinciding with $\Pi_{\operatorname{Im} \mathrm{d}u_\theta} f$ only asymptotically.



(b) **Illustration of functional Ridge regression.**
The green region represents the solution space, while the blue regions denote the target spaces. As $\alpha \to 0$, the regularized graph Γ_{D^α} of D^α approaches the graph $\Gamma_{\mathcal{H}_D}$ of D, with the angle between them vanishing at rate $\arctan(\alpha)$. The key consequence is that the projection of the objective f onto $\operatorname{Im} D^\alpha$ follows a non-linear path as $\alpha \to 0$, coinciding with $\Pi_{\operatorname{Im} D} f$ only asymptotically.

Figure 5.1: Illustrations of Ridge regression.

5.2.3 . Cutoff regression

In Equation (5.12) at the beginning of the section, we introduced cutoff regularization from the SVD perspective: given the differential $\mathrm{d}u_\theta$ of the model u, at the point θ, and its singular value decomposition $\mathrm{d}u_\theta = V_\theta \Delta_\theta U_\theta^T$, the cutoff-regularized pseudo-inverse $\mathrm{d}u_\theta^{\dagger_\alpha}$ at level $\alpha > 0$ is defined as

$$\mathrm{d}u_\theta^{\dagger_\alpha} := U_\theta \Delta_\theta^{\dagger_\alpha} V_\theta^T\,; \qquad \Delta_{\theta,p}^{\dagger_\alpha} := \begin{cases} \Delta_{\theta,p}^{-1} & \text{if } \Delta_{\theta,p} \geqslant \alpha \\ 0 & \text{otherwise} \end{cases}, \; 1 \leqslant p \leqslant P. \tag{5.37}$$

Let us reinterpret this construction. Denote by $N_\alpha \in \mathbb{N}$ the number of singular values larger than α. Equivalently, assuming $(\Delta_{\theta,p})_{1\leqslant p\leqslant P}$ is non-increasing,

$$N_\alpha := \arg\max_{p\in\mathbb{N}} \{\, \Delta_{\theta,p} \geqslant \alpha \,\}. \tag{5.38}$$

Define

$$\Theta_\alpha := \mathrm{Span}\{U_{\theta,p} : 1 \leqslant p \leqslant N_\alpha\}, \qquad T^0_{N_\alpha}\mathcal{M} := \mathrm{Span}\{V_{t,p} : 1 \leqslant p \leqslant N_\alpha\}, \tag{5.39}$$

so that $T^0_{N_\alpha}\mathcal{M} = \mathrm{d}u_\theta\,(\Theta_\alpha)$. We then have

$$\left(\mathrm{d}u_{\theta_{|\Theta_\alpha}}^{|T^0_{N_\alpha}\mathcal{M}}\right)^{-1} = \mathrm{d}u_\theta^{\dagger_\alpha}, \tag{5.40}$$

meaning that the restriction $\mathrm{d}u_\theta^\alpha := \mathrm{d}u_{\theta_{|\Theta_\alpha}}$ of $\mathrm{d}u_\theta$ to the domain Θ_α becomes invertible once its codomain is restricted to its image $T^0_{N_\alpha}\mathcal{M}$, with inverse given precisely by the cutoff pseudo-inverse $\mathrm{d}u_\theta^{\dagger_\alpha}$. Moreover, for any $h \in \Theta_\alpha$,

$$\begin{aligned} \left\|\mathrm{d}u_\theta(h)\right\|_{\mathrm{L}^2(\Omega,\mu)} = \left\|V_\theta \Delta_\theta U_\theta^T h\right\|_{\mathrm{L}^2(\Omega,\mu)} &\overset{V_\theta \text{ unitary}}{=} \left\|\Delta_\theta U_\theta^T h\right\|_{\mathbb{R}^P} \\ &\overset{h\in\Theta_\alpha}{\geqslant} \alpha \left\|U_\theta^T h\right\|_{\mathbb{R}^P} \overset{U_\theta \text{ unitary}}{=} \alpha \left\|h\right\|_{\mathbb{R}^P}. \end{aligned} \tag{5.41}$$

In other words, Equation (5.18) is satisfied by $\mathrm{d}u_\theta^\alpha$.

Thus, while ridge regularization modifies the model itself, cutoff regularization instead restricts the domain of the model so that, on this restricted domain, Equation (5.18) holds and the model becomes invertible. We summarize the fundamental properties:

1. We have

$$\bigcap_{\alpha>0} \left(\mathbb{R}^P \backslash \Theta_\alpha\right) = \mathrm{Ker}\,\mathrm{d}u_\theta, \tag{5.42}$$

that is, $\lim_{\alpha\to 0} \mathbb{R}^P\backslash\Theta_\alpha = \mathrm{Ker}\,\mathrm{d}u_\theta$, since for all $\alpha > \beta$ we have $\Theta_\alpha \subset \Theta_\beta$ and thus $\mathbb{R}^P\backslash\Theta_\beta \subset \mathbb{R}^P\backslash\Theta_\alpha$. Similarly, $\lim_{\alpha\to 0} T^0_{N_\alpha}\mathcal{M} = \mathrm{Im}\,\mathrm{d}u_\theta$. Moreover, for each $\alpha > 0$, the restriction $\mathrm{d}u_\theta^\alpha$ coincides with $\mathrm{d}u_\theta$ on Θ_α.

2. By Equation (5.41), $\mathrm{d}u_\theta^\alpha$ is injective and continuous. Restricting it to its image $T^0_{N_\alpha}\mathcal{M}$ makes it bijective and bicontinuous, with inverse precisely equal to the cutoff pseudo-inverse $\mathrm{d}u_\theta^{\dagger_\alpha}$. In particular $\mathrm{d}u\,(\Theta_\alpha)$ is closed in $\mathrm{L}^2(\Omega,\mu)$, since it is the inverse image of a closed set under $\mathrm{d}u_\theta^{\dagger_\alpha}$. Therefore least-squares solution is well-defined.

3. Solving the least-squares problem $\mathrm{d}u_\theta^\alpha = f$ is now altered in the following way: the target f is first projected onto $T^0_{N_\alpha}\mathcal{M} = \operatorname{Im}\mathrm{d}u_\theta^\alpha$. In particular, if for some $\alpha > 0$ we already have $f \in \operatorname{Im}\mathrm{d}u_\theta^\alpha$, then the regularized least-squares formulation recovers an *exact solution* to the problem. This stands in sharp contrast with Ridge regression, where such exact recovery can only occur *asymptotically* in the limit $\alpha \to 0$.

As in Section 5.2.2, we now need to reinterpret the cutoff regularization in order to extend it to the functional setting. Let us return once more to the operator $D : \mathcal{H}_D \to \mathrm{L}^2(\Omega,\mu)$ introduced at the beginning of the section. In general, one cannot define an SVD for such an operator (except when it is compact). We must therefore appeal to the spectral theorem for bounded self-adjoint operators, which relies on the notion of a *projection-valued measure* (PVM; also called a resolution of the identity). The main statements and definitions are recalled in Section C.3 of Chapter C. For a comprehensive introduction, we refer the reader to Berezansky et al. (1996b, Chapter 13).

Since D is continuous, we can define its adjoint $D^* : \mathrm{L}^2(\Omega,\mu) \to \mathcal{H}_D$, and hence the self-adjoint operator $D^*D : \mathcal{H}_D \to \mathcal{H}_D$. Applying the spectral theorem (*cf.* Theorem C.12), we obtain a projection-valued measure π_D on $\mathbb{R}$ endowed with its Borel σ-algebra, such that

$$D^*D = \int_{\mathbb{R}_+} \lambda\, \pi_D(\mathrm{d}\lambda), \qquad\qquad I_{\mathcal{H}_D} = \int_{\mathbb{R}_+} \pi_D(\mathrm{d}\lambda), \tag{5.43}$$

where the integration is restricted to $\mathbb{R}_+$ since D^*D is a positive operator. We can then define

$$\Pi_D^\alpha := \int_{\alpha^2}^{+\infty} \pi_D(\mathrm{d}\lambda), \tag{5.44}$$

which is an orthogonal projection in $\mathcal{H}_D$ since π_D is a projection-valued measure. We then define the regularized space $\mathcal{H}_D^\alpha$ at level $\alpha > 0$ by

$$\mathcal{H}_D^\alpha := \operatorname{Im}\Pi_D^\alpha \subset \mathcal{H}_D. \tag{5.45}$$

For any $u \in \mathcal{H}_D^\alpha$, we compute

$$\begin{aligned}
\|D[u]\|^2_{\mathrm{L}^2(\Omega,\mu)} &= \langle D[u]\,,\, D[u]\rangle_{\mathrm{L}^2(\Omega,\mu)} = \langle u\,,\, D^*D[u]\rangle_{\mathcal{H}_D} \\
&= \left\langle u\,,\, \int_{\mathbb{R}_+} \lambda \pi_D(\mathrm{d}\lambda) u \right\rangle_{\mathcal{H}_D} \\
&\overset{u\in\mathcal{H}_D^\alpha}{=} \left\langle u\,,\, \int_{\mathbb{R}_+} \lambda_1 \pi_D(\mathrm{d}\lambda_1) \int_{\alpha^2}^{+\infty} \pi_D(\mathrm{d}\lambda_2) u \right\rangle_{\mathcal{H}_D} \\
&\overset{\pi_D \text{ PVM}}{=} \left\langle u\,,\, \int_{\alpha^2}^{+\infty} \lambda \pi_D(\mathrm{d}\lambda) u \right\rangle_{\mathcal{H}_D} \\
&= \int_{\alpha^2}^{+\infty} \lambda \langle u\,,\, \pi_D(\mathrm{d}\lambda) u\rangle_{\mathcal{H}_D} \\
&\geqslant \alpha^2 \int_{\alpha^2}^{+\infty} \langle u\,,\, \pi_D(\mathrm{d}\lambda) u\rangle_{\mathcal{H}_D} \\
&\overset{u\in\mathcal{H}_D^\alpha}{=} \alpha^2 \langle u\,,\, u\rangle_{\mathcal{H}_D} = \alpha^2 \|u\|^2_{\mathcal{H}_D}\,.
\end{aligned} \tag{5.46}$$

That is,

$$\|D[u]\|_{\mathrm{L}^2(\Omega,\mu)} \;\geqslant\; \alpha \|u\|_{\mathcal{H}_D}\,, \tag{5.47}$$

so that Equation (5.18) is verified on $\mathcal{H}_D^\alpha$. We denote

$$D^\alpha := D_{|\mathcal{H}_D^\alpha}, \tag{5.48}$$

the restriction of D to the domain $\mathcal{H}_D^\alpha$. We can now list the fundamental properties:

1. We have
$$\bigcap_{\alpha>0} (\mathcal{H}_D \backslash \mathcal{H}_D^\alpha) = \operatorname{Ker} D \tag{5.49}$$
that is, $\lim_{\alpha\to 0} \mathcal{H}_D \backslash \mathcal{H}_D^\alpha = \operatorname{Ker} D$, since for all $\alpha > \beta$, $\mathcal{H}_D^\alpha \subset \mathcal{H}_D^\beta$ (*cf.* property 4 of Definition C.10). Moreover, by continuity of D, we also have $\lim_{\alpha\to 0} D\,[\mathcal{H}_D^\alpha] = \operatorname{Im} D$. Finally, for each $\alpha > 0$, D^α coincides with D on $\mathcal{H}_D^\alpha$ by construction.

2. As established by Equation (5.18), D^α is injective and continuous. When restricted to its image, it is therefore bijective and bicontinuous, hence invertible. In particular $D\,[\mathcal{H}_D^\alpha]$ is closed in $\mathrm{L}^2(\Omega,\mu)$, since it is the inverse image of a closed set under $(D^\alpha)^{-1}$. Therefore least-squares solution is well-defined.

3. The least-squares solution of $D^\alpha = f$ is now modified as follows: one projects f onto $\operatorname{Im} D^\alpha$. In particular, if for some $\alpha > 0$ we already have $f \in \operatorname{Im} D^\alpha$, then the regularized least-squares formulation recovers an *exact solution* to the problem $D[u] = f$. This stands in sharp contrast with Ridge regression, where such exact recovery can only occur *asymptotically* in the limit $\alpha \to 0$.

5.2.4 . Connection to Green's Function

To further emphasize the distinction between the two regularization schemes, we now reinterpret them through the lens of the Green's functions associated with the operator D. Recall that in Section 4.4.1 of Chapter 4, we established a connection between the natural gradient for PINNs and Green's functions in the finite-dimensional setting. Our aim in this section is twofold:

(i) to generalize Proposition 4.9 to arbitrary Reproducing Kernel Hilbert Spaces;

(ii) to establish a direct connection with the regularization framework introduced above, thereby providing a new interpretation of Green's functions in the context of regularized operators.

Consider the operator $D : \mathcal{H}_D \to \mathrm{L}^2(\Omega, \mu)$ from Section 5.2.1, and assume that there exists an RKHS $\mathcal{H}_0$ isometrically embedded in $\mathcal{H}_D$ [1]. For the generalized Green's function[2] of D to be well defined, the range $D[\mathcal{H}_0]$ must form a closed subspace of $\mathrm{L}^2(\Omega, \mu)$. As argued earlier, this is guaranteed if D is continuously invertible: indeed, in this case

$$D[\mathcal{H}_0] = (D^{-1})^{-1}[\mathcal{H}_0], \tag{5.50}$$

and the inverse image of a closed subspace under a continuous operator is closed.

Key observation. To generalize Proposition 4.9 in Chapter 4, we require D to be continuously invertible. Conveniently, this is precisely the property enforced by the regularization schemes we introduced earlier.

In what follows, we first focus on the cutoff regularization, which offers the clearest interpretation in terms of Green's functions. We then briefly mention the case of Ridge regression.

We now focus on the restriction of D to the domain $\mathcal{H}_0$. Since the operator $D^*D : \mathcal{H}_D \to \mathcal{H}_D$, when restricted to $\mathcal{H}_0$, does not *a priori* map $\mathcal{H}_0$ into itself, we must first adapt the framework introduced in Section 5.2.3 in order to apply the spectral theorem stated in Theorem C.12.

Because $\mathcal{H}_0$ is isometrically embedded in $\mathcal{H}_D$, we have for all $u, v \in \mathcal{H}_0$,

$$\begin{aligned} \langle D[u]\,,\, D[v]\rangle_{\mathrm{L}^2(\Omega,\mu)} &= \big\langle D\big[\Pi_{\mathcal{H}_0} u\big]\,,\, D\big[\Pi_{\mathcal{H}_0} v\big]\big\rangle_{\mathrm{L}^2(\Omega,\mu)} \\ &= \big\langle \Pi_{\mathcal{H}_0} u\,,\, D^* D\big[\Pi_{\mathcal{H}_0} v\big]\big\rangle_{\mathcal{H}_D} \\ &= \big\langle u\,,\, \big(\Pi_{\mathcal{H}_0} D^* D \Pi_{\mathcal{H}_0}\big)[v]\big\rangle_{\mathcal{H}_D}\,, \end{aligned} \tag{5.51}$$

[1] For instance, if $\mathcal{H}_0$ is a finite-dimensional RKHS, see Proposition 3.19 in Section 3.2.2 of Chapter 3.

[2] See Definition 3.9 in Section 3.2.1 of Chapter 3.

where we used in the last step that $\Pi_{\mathcal{H}_0}$ is self-adjoint.

We can therefore apply the spectral theorem to the bounded self-adjoint operator $\Pi_{\mathcal{H}_0} D^* D \Pi_{\mathcal{H}_0} : \mathcal{H}_0 \to \mathcal{H}_0$, thus obtaining an analogue of the decomposition given in Equation (5.43):

$$\Pi_{\mathcal{H}_0} D^* D \Pi_{\mathcal{H}_0} = \int_{\mathbb{R}_+} \lambda \, \pi_D^{\mathcal{H}_0}(\mathrm{d}\lambda), \qquad I_{\mathcal{H}_0} = \int_{\mathbb{R}_+} \pi_D^{\mathcal{H}_0}(\mathrm{d}\lambda). \tag{5.52}$$

Regularized spaces. Fixing $\alpha > 0$, and analogously to Equations (5.44) and (5.45), we define the regularized projection and subspace:

$$\Pi^{\alpha}_{D,\mathcal{H}_0} := \int_{\alpha^2}^{+\infty} \pi_D^{\mathcal{H}_0}(\mathrm{d}\lambda), \qquad \mathcal{H}^{\alpha}_{D,\mathcal{H}_0} := \operatorname{Im} \Pi^{\alpha}_{D,\mathcal{H}_0} \subset \mathcal{H}_0 \subset \mathcal{H}_D. \tag{5.53}$$

Let $k : \Omega \times \Omega \to \mathbb{R}$ be the reproducing kernel of $\mathcal{H}_0$. Then, by Proposition 3.27 in Section 3.3.1 of Chapter 3, $\mathcal{H}^{\alpha}_{D,\mathcal{H}_0}$ is an RKHS with reproducing kernel

$$k_\alpha(x, y) := \Pi^{\alpha}_{D,\mathcal{H}_0}[k(x, \cdot)](y), \qquad \forall x, y \in \Omega. \tag{5.54}$$

Norm equivalence. Since $\mathcal{H}^{\alpha}_{D,\mathcal{H}_0} \subset \mathcal{H}_0 \subset \mathcal{H}_D$, inequality in Equation (5.17) remains valid, i.e. for all $u \in \mathcal{H}^{\alpha}_{D,\mathcal{H}_0}$,

$$\|D[u]\|_{\mathrm{L}^2(\Omega,\mu)} \leqslant \|u\|_{\mathcal{H}_D} . \tag{5.55}$$

Furthermore, by an argument entirely analogous to Equation (5.46), we also have

$$\|D[u]\|_{\mathrm{L}^2(\Omega,\mu)} \geqslant \alpha \, \|u\|_{\mathcal{H}_D} , \qquad \forall u \in \mathcal{H}^{\alpha}_{D,\mathcal{H}_0}. \tag{5.56}$$

In particular, the functional

$$\|\cdot\|_D : \left\{ \begin{array}{lcl} \mathcal{H}^{\alpha}_{D,\mathcal{H}_0} & \to & \mathbb{R} \\ u & \mapsto & \|D[u]\|_{\mathrm{L}^2(\Omega,\mu)} \end{array} \right. \tag{5.57}$$

defines a norm equivalent to $\|\cdot\|_{\mathcal{H}_D}$ on $\mathcal{H}^{\alpha}_{D,\mathcal{H}_0}$. We can therefore apply the results of Section 3.4.2 in Chapter 3, and obtain:

Theorem 5.2. *The generalized Green's function of the operator D in the regularized space $\mathcal{H}^{\alpha}_{D,\mathcal{H}_0}$ is given, for all $x, y \in \Omega$, by*

$$g_D(x, y) := D[k_D(x, \cdot)](y), \tag{5.58}$$

where k_D denotes the reproducing kernel of $\left(\mathcal{H}^{\alpha}_{D,\mathcal{H}_0}, \|\cdot\|_D\right)$, explicitly expressed for all $x, y \in \Omega$ as

$$k_D(x, y) = \int_{\alpha^2}^{+\infty} \lambda^{-1} \, \pi_D^{\mathcal{H}_0}(d\lambda)[k(x, \cdot)](y). \tag{5.59}$$

Proof. Direct application of Theorem 3.54 in Section 3.4.2 of Chapter 3. □

The key insight of Theorem 5.2 is that, in the PINNs setting, we implicitly construct the reproducing kernel k_D associated with the norm $\|\cdot\|_D$ on the regularized tangent space $T_\theta^\alpha \mathcal{M}$ of the neural network manifold $\mathcal{M}$, at cutoff level α. This kernel is precisely the PINNs NNTK introduced in Section 4.3.2 of Chapter 4.

A crucial consequence is that the regularization of the Gram matrix is not merely a "numerical trick" to guarantee stability: it is the very mechanism that ensures the Green's function is well defined.

Conceptual interpretation. This perspective offers a deeper understanding of the overall procedure: rather than inverting the operator D explicitly, we construct a kernel k_D whose associated metric turns D into an isometry, thereby ensuring that D^* naturally acts as the generalized left-inverse of D. The key insight lies in the following observations:

(i) The adjoint operator D^* never needs to be computed explicitly, as it is implicitly encoded in the identity

$$\langle D[k_D(x,\cdot)]\,,\, f\rangle_{\mathrm{L}^2(\Omega,\mu)} = \langle k_D(x,\cdot)\,,\, D^* f\rangle_D\,. \tag{5.60}$$

(ii) By virtue of the reproducing property, the same identity provides a direct means of evaluating the generalized solution $D^* f$.

Comparison with Ridge regression. An analogous analysis holds for Ridge regression. However, instead of inverting D "via isometry", we invert the augmented operator $\big(D, \alpha \mathrm{I}_{\mathcal{H}_D}\big)$.

Building on these insights, we will propose a principled strategy to gradually decrease the regularization threshold α, thereby enlarging the approximation space Θ_α and its associated tangent space $T_\theta^\alpha \mathcal{M}$.

This strategy is motivated by a detailed empirical study of ANaGRAM's training dynamics, based on a novel metric called the *Reconstruction Error of the Functional Gradient*, which we introduce in the next section.

5.3 . Reconstruction Error of the Functional Gradient

We begin by defining the *Reconstruction Error of the Functional Gradient* (hereafter abbreviated as the *reconstruction error* or simply *RCE*). We then establish a set of relevant and insightful properties that will allow us to analyze the training dynamics of ANaGRAM.

5.3.1 . Definition

Let $\theta \in \mathbb{R}^P$, and consider the empirical feature matrix $\widehat{\phi} \in \mathbb{R}^{S\times P}$ and the empirical functional gradient $\widehat{\nabla\mathcal{L}} \in \mathbb{R}^S$, as defined in Equation (5.11). Namely, for all $1 \leqslant i \leqslant S$ and $1 \leqslant p \leqslant P$, we have:

$$\widehat{\phi}_{i,p} := \partial_p u_\theta(x_i), \qquad \widehat{\nabla\mathcal{L}}_i := u_\theta(x_i) - f(x_i). \tag{5.61}$$

Let us now consider the empirical tangent spaces obtained by taking different ranges of left singular vectors of

$$\widehat{\phi} = \widehat{V}\widehat{\Delta}\widehat{U}^T. \tag{5.62}$$

For $1 \leqslant M, N \leqslant \mathrm{r_{svd}}$, we define

$$\widehat{T_N^M\mathcal{M}} = \mathrm{Span}(\widehat{V}_{t,i} \,:\, M \leqslant i \leqslant N). \tag{5.63}$$

We then introduce the *reconstruction error* (RCE), defined as

$$\mathrm{RCE}_N^S := \frac{1}{\sqrt{S}}\left\|\widehat{V}\Pi_N^0\widehat{V}^T\widehat{\nabla\mathcal{L}} - \widehat{\nabla\mathcal{L}}\right\|_{\mathbb{R}^S} = \frac{1}{\sqrt{S}}\left\|\Pi^{\perp}_{\widehat{T_N^0\mathcal{M}}}\widehat{\nabla\mathcal{L}} - \widehat{\nabla\mathcal{L}}\right\|_{\mathbb{R}^S}, \tag{5.64}$$

where $\Pi_N^M \in \mathbb{R}^{\mathrm{r_{svd}}\times\mathrm{r_{svd}}}$ denotes the projection matrix onto $\widehat{T_N^M\mathcal{M}}$, given by

$$\Pi_N^M = \begin{pmatrix} 0_M & 0 & 0 \\ 0 & I_{N-M} & 0 \\ 0 & 0 & 0_{\mathrm{r_{svd}}-N} \end{pmatrix} = \sum_{p=M+1}^{N} \boldsymbol{e}_p\boldsymbol{e}_p^T, \tag{5.65}$$

with $(\boldsymbol{e}_p)_{1\leqslant p\leqslant \mathrm{r_{svd}}}$ denoting the canonical basis of $\mathbb{R}^{\mathrm{r_{svd}}}$.

Intuitively, the reconstruction error quantifies how much information from the functional gradient signal is lost when restricting to the first N singular directions in the SVD. The next section formalizes this interpretation and introduces several related properties.

5.3.2 . Properties of the Reconstruction Error

We begin with a result characterizing the difference between two levels of the reconstruction error.

Proposition 5.3. *For $1 \leqslant M \leqslant N \leqslant r_{svd}$, the following identity holds:*

$$\left(RCE_M^S\right)^2 - \left(RCE_N^S\right)^2 = \frac{1}{S}\left\|\Pi_N^M\widehat{V}^T\widehat{\nabla\mathcal{L}}\right\|^2_{\mathbb{R}^S}. \tag{5.66}$$

To prove this result, we first establish the following technical lemma.

Lemma 5.4. *Let $1 \leqslant N \leqslant r_{svd}$. Then*

$$S\left(RCE_N^S\right)^2 = \widehat{\nabla\mathcal{L}}_\theta^T\widehat{\nabla\mathcal{L}}_\theta - \sum_{p=1}^{N}\left(\widehat{V}_{\theta_p}^T\widehat{\nabla\mathcal{L}}_\theta\right)^2. \tag{5.67}$$

Proof. This follows from a straightforward calculation:

$$
\begin{aligned}
S\left(\mathrm{RCE}_N^S\right)^2 =& \left\langle \widehat{V}\Pi_N^0\widehat{V}^t\widehat{\nabla\mathcal{L}_\theta} - \widehat{\nabla\mathcal{L}_\theta}\,,\, \widehat{V}\Pi_N^0\widehat{V}^t\widehat{\nabla\mathcal{L}_\theta} - \widehat{\nabla\mathcal{L}_\theta}\right\rangle_{\mathbb{R}^S} \\
=& \widehat{\nabla\mathcal{L}_\theta}^t\widehat{\nabla\mathcal{L}_\theta} + \widehat{\nabla\mathcal{L}_\theta}^t\widehat{V}\overbrace{\Pi_N^0\underbrace{\widehat{V}^t\widehat{V}}_{=I_d}\Pi_N^0}^{=\Pi_N^0}\widehat{V}^t\widehat{\nabla\mathcal{L}_\theta} \\
& - 2\widehat{\nabla\mathcal{L}_\theta}^t\widehat{V}\Pi_N^0\widehat{V}^t\widehat{\nabla\mathcal{L}_\theta} \\
=& \widehat{\nabla\mathcal{L}_\theta}^t\widehat{\nabla\mathcal{L}_\theta} - \widehat{\nabla\mathcal{L}_\theta}^t\widehat{V}\Pi_N^0\widehat{V}^t\widehat{\nabla\mathcal{L}_\theta} \\
=& \widehat{\nabla\mathcal{L}_\theta}^t\widehat{\nabla\mathcal{L}_\theta} - \widehat{\nabla\mathcal{L}_\theta}^t\widehat{V}\left(\sum_{p=1}^N \boldsymbol{e}_p\boldsymbol{e}^{(p)^t}\right)\widehat{V}^t\widehat{\nabla\mathcal{L}_\theta} \\
=& \widehat{\nabla\mathcal{L}_\theta}^t\widehat{\nabla\mathcal{L}_\theta} - \sum_{p=1}^N\left(\widehat{V}_{\theta_p}^t\widehat{\nabla\mathcal{L}_\theta}\right)^2,
\end{aligned}
\tag{5.68}
$$

where in the second equality, we used the orthogonality of $\widehat{V}$ and the fact that Π_N^0 is a projection operator. □

Proof of Proposition 5.3. Fix $1 \leqslant N \leqslant M \leqslant \mathrm{r_{svd}}$. Then applying Lemma 5.4 to RCE_M^S and RCE_N^S, we get

$$
S\left(\mathrm{RCE}_M^S\right)^2 = \widehat{\nabla\mathcal{L}_\theta}^t\widehat{\nabla\mathcal{L}_\theta} - \sum_{p=1}^M\left(\widehat{V}_{\theta_p}^t\widehat{\nabla\mathcal{L}_\theta}\right)^2; \tag{5.69}
$$

$$
S\left(\mathrm{RCE}_N^S\right)^2 = \widehat{\nabla\mathcal{L}_\theta}^t\widehat{\nabla\mathcal{L}_\theta} - \sum_{p=1}^N\left(\widehat{V}_{\theta_p}^t\widehat{\nabla\mathcal{L}_\theta}\right)^2. \tag{5.70}
$$

Therefore

$$
\begin{aligned}
S\left(\left(\mathrm{RCE}_N^S\right)^2-\left(\mathrm{RCE}_M^S\right)^2\right) &= \sum_{p=1}^{N}\left(\widehat{V}_{\theta_p}^t\widehat{\nabla\mathcal{L}}_\theta\right)^2-\sum_{p=1}^{M}\left(\widehat{V}_{\theta_p}^t\widehat{\nabla\mathcal{L}}_\theta\right)^2 \\
&\overset{M\leqslant N}{=} \sum_{p=M+1}^{N}\left(\widehat{V}_{\theta_p}^t\widehat{\nabla\mathcal{L}}_\theta\right)^2 \\
&= \sum_{p=M+1}^{N}\left(\boldsymbol{e}^{(p)^t}\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\right)^2 \\
&= \sum_{p=M+1}^{N}\left(\widehat{\nabla\mathcal{L}}_\theta^t\widehat{V}\boldsymbol{e}_p\right)\left(\boldsymbol{e}^{(p)^t}\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\right) \\
&= \widehat{\nabla\mathcal{L}}_\theta^t\widehat{V}\underbrace{\left(\sum_{p=M+1}^{N}\boldsymbol{e}_p\boldsymbol{e}^{(p)^t}\right)}_{=\Pi_N^M \text{ by Equation (5.65)}}\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta \\
&= \left\langle\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\,,\,\Pi_N^M\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\right\rangle_{\mathbb{R}^S} \\
&\overset{\Pi_N^{M^2}=\Pi_N^M}{\overset{\Pi_N^{M^t}=\Pi_N^M}{=}} \left\langle\Pi_N^M\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\,,\,\Pi_N^M\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\right\rangle_{\mathbb{R}^S} \\
&= \left\|\Pi_N^M\widehat{V}^t\widehat{\nabla\mathcal{L}}_\theta\right\|_{\mathbb{R}^S}^2,
\end{aligned}
\tag{5.71}
$$

where we use in the penultimate equality, the fact that Π_N^M is an orthogonal projection. □

We next establish an asymptotic interpretation of the reconstruction error.

Proposition 5.5. *RCE_N^S is a non-increasing function of N, i.e. for all $1 \leqslant M, N \leqslant r_{svd}$:*

$$
M \leqslant N \implies RCE_M^S \geqslant RCE_N^S. \tag{5.72}
$$

Furthermore, assuming that $(x_i)_{i=1}^S$ are i.i.d samples from μ, we have μ-almost surely

$$
\lim_{S\to\infty} RCE_N^S = \left\|\nabla\mathcal{L}_{u_\theta}-\Pi_{T_N^0\mathcal{M}}\nabla\mathcal{L}_{u_\theta}\right\|_{L^2(\Omega,\mu)} = \left\|\Pi_{\left[T_N^0\mathcal{M}\right]^\perp}\nabla\mathcal{L}_{u_\theta}\right\|_{L^2(\Omega,\mu)}, \tag{5.73}
$$

where $T_N^M\mathcal{M} = \mathrm{Span}(V_{t,i} : M \leqslant i \leqslant N)$, with $(V_{t,i})_{1\leqslant i\leqslant r_{svd}}$, the left singular-vectors of the differential du_θ ordered in a decreasing order according to their associated singular values.

***Remark* 5.6.** Note that for $1 \leqslant i \leqslant N$, the left singular vectors $\widehat{V}_{t,i} \in \mathbb{R}^S$ of $\widehat{\phi}$, can be seen as discretized versions of $V_{t,i}$ in Proposition 5.5. Indeed, a "weak convergence" holds, *i.e.* for all $h \in \mathcal{H}$,

$$\frac{1}{S}\sum_{j=1}^{S} \widehat{V}_{t,i,j} h(x_j) = \frac{1}{S}\sum_{j=1}^{S} V_{t,i}(x_j)h(x_j) \overset{S\to\infty}{\to} \langle V_{t,i}, h\rangle_{L^2} \tag{5.74}$$

***Remark* 5.7.** Viewed through the lens of Proposition 5.5, the RCE is closely related to the notion of an *expressivity bottleneck* introduced in Verbockhaven et al. (2024). Indeed, it quantifies the portion of the learning signal that is not captured when truncating the natural gradient computation to the first N components. Hence, this metric provides an explicit means to assess and compare different cutoff strategies. Importantly, it incurs no additional computational cost, as ANaGRAM already computes the required SVD.

Proof of Proposition 5.5. The first statement is a direct consequence of Proposition 5.3 proven above.

Let us now show that the second statement takes place. Since $\nabla\mathcal{L}_{u_\theta} \in \mathrm{L}^2(\Omega,\mu)$ and $\operatorname{Im} \mathrm{d}u_\theta \subset \mathrm{L}^2(\Omega,\mu)$, the law of large numbers yields that for all $1 \leqslant p, q \leqslant P$

$$\begin{aligned}\lim_{S\to\infty} \frac{1}{S}\sum_{i=1}^{S} [\nabla\mathcal{L}_{u_\theta}(x_i)]^2 &= \lim_{S\to\infty} \frac{1}{S}\widehat{\nabla\mathcal{L}}_\theta^t \widehat{\nabla\mathcal{L}}_\theta \\ &= \int_\Omega [\nabla\mathcal{L}_{u_\theta}(x)]^2 \,\mu(\mathrm{d}x) \quad a.s,\end{aligned} \tag{5.75}$$

$$\begin{aligned}\lim_{S\to\infty} \frac{1}{S}\sum_{i=1}^{S} \partial_p u_\theta(x_i)\nabla\mathcal{L}_{u_\theta}(x_i) &= \lim_{S\to\infty} \frac{1}{S}\widehat{\phi}_{\theta_p}^t \widehat{\nabla\mathcal{L}}_\theta \\ &= \int_\Omega \partial_p u_\theta(x)\nabla\mathcal{L}_{u_\theta}(x)\mu(\mathrm{d}x) \quad a.s,\end{aligned} \tag{5.76}$$

$$\begin{aligned}\lim_{S\to\infty} \frac{1}{S}\sum_{i=1}^{S} \partial_p u_\theta(x_i)\partial_q u_\theta(x_i) &= \lim_{S\to\infty} \frac{1}{S}\widehat{\phi}_{\theta_p}^t \widehat{\phi}_{\theta_q} \\ &= \int_\Omega \partial_p u_\theta(x)\partial_q u_\theta(x)\mu(\mathrm{d}x) \quad a.s.\end{aligned} \tag{5.77}$$

In particular, this implies

$$\lim_{S\to\infty} \frac{1}{S}\widehat{\phi}^t\widehat{\phi} = G_\theta = U_\theta \Delta_\theta^2 U_\theta^t \quad a.s. \tag{5.78}$$

Since the eigenvectors (and eigenvalues) are continuous functions of the matrix coefficients (by polynomial dependence through the characteristic polynomial) and taking into account that $\frac{1}{S}\widehat{\phi}^t\widehat{\phi} = \frac{1}{S}\widehat{U}\Delta_\theta^2\widehat{U}^t$, this yields

$$\lim_{S\to\infty} \widehat{U} = U_\theta \quad a.s; \qquad \lim_{S\to\infty} \frac{1}{S}\widehat{\Delta}^2 = \Delta_\theta^2 \quad a.s. \tag{5.79}$$

By continuity of the square root and the inverse on $\mathbb{R}_+^*$, we get that for all $1 \leqslant p \leqslant P$ such that $\Delta_{\theta_p} > 0$

$$\lim_{S\to\infty} \sqrt{S}\widehat{\Delta}_{\theta_p}^{-1} = \Delta_{\theta_p}^{-1} \quad a.s, \tag{5.80}$$

and thus for all $1 \leqslant p \leqslant P$ such that $\Delta_{\theta_p} > 0$, we have *μ-almost surely*

$$\begin{aligned}
\lim_{S\to\infty} \frac{1}{\sqrt{S}} \widehat{V}_{\theta_p}^T \widehat{\nabla \mathcal{L}_\theta} &= \lim_{S\to\infty} \sqrt{S}\widehat{\Delta}_{\theta_p}^{-1} \widehat{U}_{\theta_p}^t \left(\sum_{q=1}^{P} \boldsymbol{e}^{(q)} \boldsymbol{e}^{(q)^t} \right) \frac{1}{S} \widehat{\phi}^t \widehat{\nabla \mathcal{L}_\theta} \\
&= \sum_{q=1}^{P} \left(\lim_{S\to\infty} \sqrt{S}\widehat{\Delta}_{\theta_p}^{-1} \right) \left(\lim_{S\to\infty} \widehat{U}_{\theta_p}^t \boldsymbol{e}^{(q)} \right) \left(\lim_{S\to\infty} \frac{1}{S} \widehat{\phi}_{\theta_q}^t \widehat{\nabla \mathcal{L}_\theta} \right) \\
&= \sum_{q=1}^{P} \Delta_{\theta_p}^{-1} U_{\theta_p}^t \boldsymbol{e}^{(q)} \int_\Omega \partial_q u_\theta(x) \nabla \mathcal{L}_{u_\theta}(x) \mu(\mathrm{d}x) \\
&= \int_\Omega \mathrm{d}u_\theta \left(U_{\theta_p} \Delta_{\theta_p}^{-1} \right)(x) \nabla \mathcal{L}_{u_\theta}(x) \mu(\mathrm{d}x) \\
&= \int_\Omega V_{\theta_p}(x) \nabla \mathcal{L}_{u_\theta}(x) \mu(\mathrm{d}x),
\end{aligned} \tag{5.81}$$

where we used in the last equality, the identification of the singular vectors of $\mathrm{d}u_\theta$ etablished in Equation (4.15) of Lemma 4.4 in Section 4.2.2 of Chapter 4. Combining Equations (5.75) and (5.81) with Lemma 5.4, this yields *μ-almost surely*,

$$\lim_{S\to\infty} \left(\mathrm{RCE}_N^S \right)^2 = \int_\Omega \nabla \mathcal{L}_{u_\theta}(x)^2 \mu(\mathrm{d}x) - \sum_{p=1}^{N} \left(\int_\Omega V_{\theta_p}(x) \nabla \mathcal{L}_{u_\theta}(x) \mu(\mathrm{d}x) \right)^2 \tag{5.82}$$

Furthermore, by Fubini's theorem, we have *μ-almost surely*

$$\begin{aligned}
&\sum_{p=1}^{N} \left(\int_\Omega V_{\theta_p}(x) \nabla \mathcal{L}_{u_\theta}(x) \mu(\mathrm{d}x) \right)^2 \\
&= \int_{\Omega^2} \nabla \mathcal{L}_{u_\theta}(x) \left(\sum_{p=1}^{N} V_{\theta_p}(x) V_{\theta_p}(y) \right) \nabla \mathcal{L}_{u_\theta}(y) \mu(\mathrm{d}x) \mu(\mathrm{d}y) \\
&= \int_\Omega \nabla \mathcal{L}_{u_\theta}(x) \Pi_{\mathrm{Span}(V_{\theta_i} : 1 \leqslant i \leqslant N)} \nabla \mathcal{L}_{u_\theta}(x) \mu(\mathrm{d}x) \\
&= \left\| \Pi_{\mathrm{Span}(V_{\theta_i} : 1 \leqslant i \leqslant N)} \nabla \mathcal{L}_{u_\theta} \right\|_{\mathrm{L}^2(\Omega,\mu)}^2,
\end{aligned} \tag{5.83}$$

where, in the second equality, we used Proposition 3.28 from Section 3.3.1 in Chapter 3, and in the third equality, we used the fact that projections are idempotent, namely,

$$\left(\Pi_{\mathrm{Span}(V_{\theta_i} : 1 \leqslant i \leqslant N)} \right)^2 = \Pi_{\mathrm{Span}(V_{\theta_i} : 1 \leqslant i \leqslant N)}. \tag{5.84}$$

Substituting Equation (5.83) into Equation (5.82) yields

$$\lim_{S\to\infty}\left(\mathrm{RCE}_N^S\right)^2 = \|\nabla\mathcal{L}_{u_\theta}\|^2_{\mathrm{L}^2(\Omega,\mu)} - \left\|\Pi_{\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)}\nabla\mathcal{L}_{u_\theta}\right\|^2_{\mathrm{L}^2(\Omega,\mu)} \quad a.s. \tag{5.85}$$

Writing the canonical decomposition in Hilbert spaces,

$$\nabla\mathcal{L}_{u_\theta} = \Pi_{\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)}\nabla\mathcal{L}_{u_\theta} + \Pi_{\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)^\perp}\nabla\mathcal{L}_{u_\theta}, \tag{5.86}$$

and noting that $\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)$ is a closed subspace, the Pythagora's theorem gives

$$\begin{aligned}\|\nabla\mathcal{L}_{u_\theta}\|^2_{\mathrm{L}^2(\Omega,\mu)} = &\left\|\Pi_{\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)}\nabla\mathcal{L}_{u_\theta}\right\|^2_{\mathrm{L}^2(\Omega,\mu)} \\ &+ \left\|\Pi_{\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)^\perp}\nabla\mathcal{L}_{u_\theta}\right\|^2_{\mathrm{L}^2(\Omega,\mu)}.\end{aligned} \tag{5.87}$$

Finally, substituting this expression into Equation (5.85) yields,

$$\lim_{S\to\infty}\left(\mathrm{RCE}_N^S\right)^2 = \left\|\Pi_{\mathrm{Span}(V_{\theta_i}\,:\,1\leqslant i\leqslant N)^\perp}\nabla\mathcal{L}_{u_\theta}\right\|^2_{\mathrm{L}^2(\Omega,\mu)}. \tag{5.88}$$

This completes the proof. □

To conclude this section, we combine Proposition 5.5 and Proposition 5.3 to derive an asymptotic interpretation of the difference between two levels of reconstruction error.

Corollary 5.8. *For* $1 \leqslant M \leqslant N \leqslant r_{svd}$*:*

$$\lim_{S\to\infty}\left(RCE_M^S\right)^2 - \left(RCE_N^S\right)^2 = \left\|\Pi_{T_N^M\mathcal{M}}\nabla\mathcal{L}_{u_\theta}\right\|^2_{L^2(\Omega)} \tag{5.89}$$

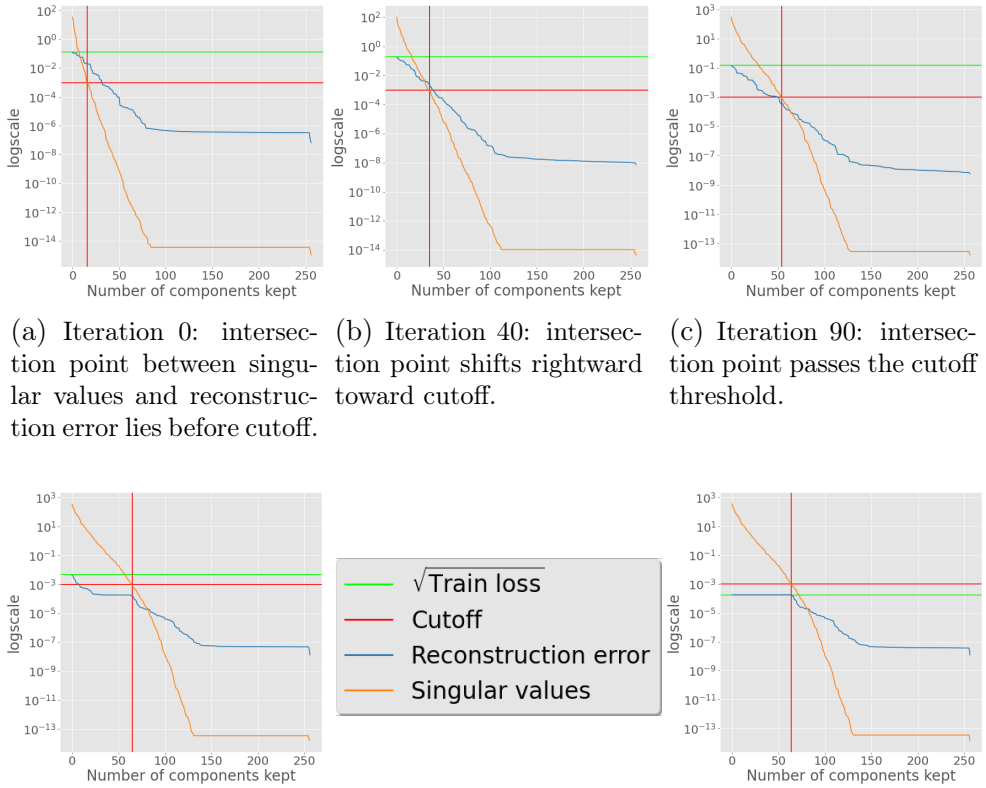
Proof. The result follows by applying the same reasoning as in the proof of Proposition 5.5 to Equation (5.66) in Proposition 5.3. □

5.4 . Insights on ANaGRAM's Training Dynamics

In this section, we use the reconstruction error introduced in Section 5.3 to highlight several empirical phenomena observed throughout the training dynamics of ANaGRAM.

5.4.1 . Empirical Observations: Flattening

This section aims to illustrate the evolution of the singular values, training loss, and reconstruction error throughout the ANaGRAM's optimization process. To gain further insight, we first depict in Figure 5.2 these key quantities during an actual training run of ANaGRAM on the Heat equation.



(a) Iteration 0: intersection point between singular values and reconstruction error lies before cutoff.

(b) Iteration 40: intersection point shifts rightward toward cutoff.

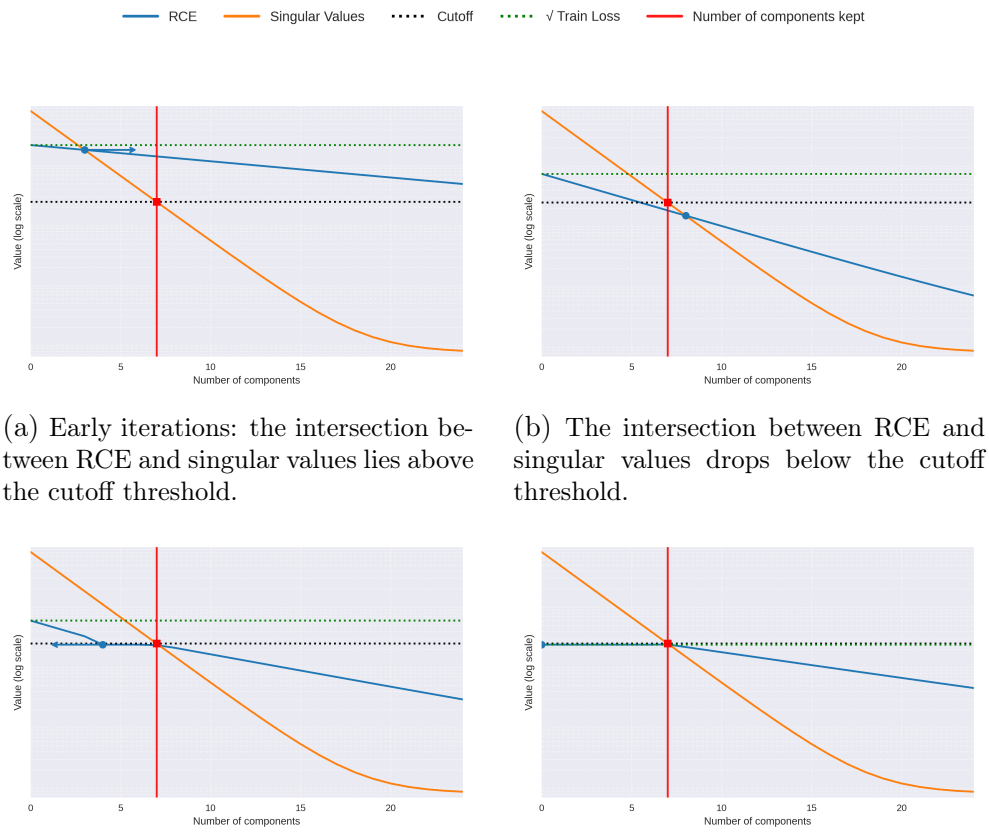
(c) Iteration 90: intersection point passes the cutoff threshold.

(d) Iteration 120. Beginning of *flattening*: reconstruction errors stabilizes at constant level before cutoff.

(e) Iteration 150: Complete flattening. Training loss reaches the flattened reconstruction error level.

Figure 5.2: **Evolution of the key quantities during ANaGRAM training on the Heat equation.** The learning rate is determined via line search, and the cutoff threshold is fixed at $\alpha = 10^{-3}$. The training loss corresponds to $\left\|\widehat{\nabla \mathcal{L}}\right\|^2$. The flattening phenomenon emerges around iteration 120 and is completed by iteration 150. Overall, the dynamics exhibit two distinct phases, culminating in the flattening of the reconstruction error.

For clarity, we schematize these phases in Figure 5.3, highlighting the phenomena previously illustrated in Figure 5.2.



(a) Early iterations: the intersection between RCE and singular values lies above the cutoff threshold.

(b) The intersection between RCE and singular values drops below the cutoff threshold.

(c) Onset of the flattening regime: a plateau in RCE emerges near $\mathrm{r}_{\mathrm{cutoff}}$ and progressively propagates toward zero.

(d) Final stage: complete flattening and convergence.

Figure 5.3: **ANaGRAM training dynamics.** We illustrate four characteristic phases of training: (a) early evolution, (b) intersection between the RCE and singular values crossing the cutoff level, (c) onset of the flattening regime, and (d) complete flattening corresponding to the final loss level. Although the vertical scale varies across plots, the cutoff level remains fixed. The number of components retained by ANaGRAM, denoted $\mathrm{r}_{\mathrm{cutoff}}$, corresponds to the intersection point between the cutoff level line and the singular value curve.

Let α denote a cutoff level, and let $\mathrm{r}_{\mathrm{cutoff}}$ be the number of components retained by the cutoff, that is,

$$\mathrm{r}_{\mathrm{cutoff}}(t) = \max\{\, j : \widehat{\Delta}_{t,j} \geqslant \alpha \,\}. \tag{5.90}$$

In Figure 5.3, we observe several distinct stages of the training process. Initially, the y-coordinate of the intersection between the reconstruction error and the singular values lies above the cutoff level (Figures 5.2a and 5.3a). As training progresses, this coordinate gradually decreases until it reaches the cutoff level (Figure 5.2b), and eventually falls below it (Figures 5.2c and 5.3b).

Throughout this phase, the training loss—corresponding to the reconstruction error with zero retained components (green curves in the figures; see also Remark 5.9 below)—remains nearly constant.

At this point, a phenomenon we refer to as *flattening* occurs: once the y-coordinate of the intersection between the reconstruction error and the singular values becomes significantly lower than the cutoff level, the reconstruction error begins to *flatten* over the interval $[N_{\text{flat}}, \mathrm{r}_{\text{cutoff}}]$, where N_{flat} is the smallest integer such that

$$\mathrm{RCE}^S_{N_{\text{flat}}} - \mathrm{RCE}^S_{\mathrm{r}_{\text{cutoff}}} \approx 0. \tag{5.91}$$

Eventually, this phenomenon propagates toward smaller numbers of retained components (Figures 5.2d and 5.3c), until $N_{\text{flat}} = 0$ (Figures 5.2e and 5.3d). At this stage, the reconstruction error remains constant across all retained components, and the training concludes with the loss reaching the cutoff level.

***Remark* 5.9.** The flattening phenomenon sheds light on the sharp drop in training loss observed near the end of optimization, as reported in Section 4.5 of Chapter 4. Indeed, by combining Equations (5.7), (5.11) and (5.64) and using that $\Pi^0_0 = 0$, we obtain

$$\begin{aligned} \mathrm{RCE}^{S^2}_0 &\overset{(5.64)}{=} \frac{1}{S}\left\|\widehat{V}\Pi^0_0\widehat{V}^t\widehat{\nabla\mathcal{L}} - \widehat{\nabla\mathcal{L}}\right\|^2_{\mathbb{R}^S} = \frac{1}{S}\left\|\widehat{\nabla\mathcal{L}}\right\|^2_{\mathbb{R}^S} \\ &\overset{(5.11)}{=} \frac{1}{S}\sum_{i=1}^{S}\left(u_\theta(x_i) - f(x_i)\right)^2 \overset{(5.7)}{=} \ell(\theta). \end{aligned} \tag{5.92}$$

Thus, the last iterations of flattening are **directly responsible for the sudden drop of train loss** at the end of the training.

***Remark* 5.10.** Since

$$\left(\mathrm{RCE}^S_M\right)^2 - \left(\mathrm{RCE}^S_N\right)^2 = \left(\mathrm{RCE}^S_M - \mathrm{RCE}^S_N\right)\left(\mathrm{RCE}^S_M + \mathrm{RCE}^S_N\right), \tag{5.93}$$

the flattening phenomenon described in Equation (5.91) can be reinterpreted, in light of Proposition 5.3, as

$$\frac{1}{S}\left\|\Pi^{N_{\text{flat}}}_{\mathrm{r}_{\text{cutoff}}}\widehat{V}^T\widehat{\nabla\mathcal{L}}\right\|^2_{\mathbb{R}^S} \approx 0. \tag{5.94}$$

Furthermore, by Corollary 5.8, this admits the following asymptotic interpretation:

$$\left\|\Pi_{T^{N_{\text{flat}}}_{\mathrm{r}_{\text{cutoff}}}\mathcal{M}}\nabla\mathcal{L}_{u_\theta}\right\|^2_{\mathrm{L}^2(\Omega)} \approx 0. \tag{5.95}$$

In other words, the projection of the functional gradient (proportional to the residual error) onto the subspace spanned by the left singular vectors indexed by $[N_{\text{flat}}, \mathrm{r}_{\text{cutoff}}]$ vanishes. As $N_{\text{flat}} \to 0$, this indicates that the signal projected onto the first $\mathrm{r}_{\text{cutoff}}$ components retained by the cutoff becomes negligible.

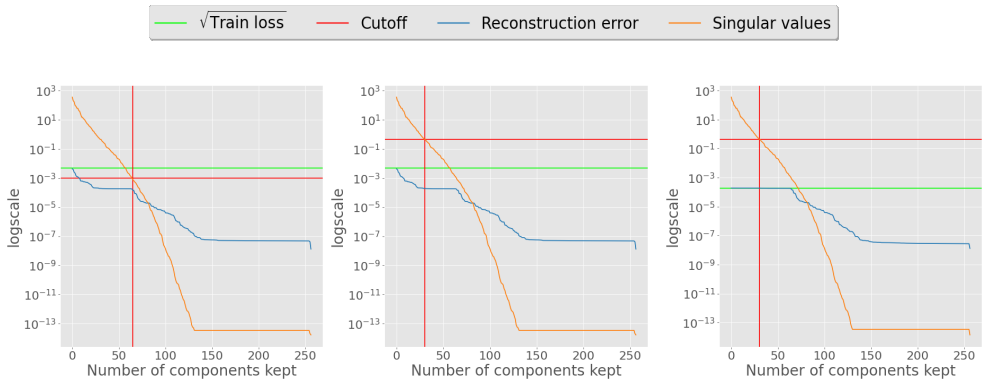
Hence, the flattening phenomenon indicates that the model has fully exploited all usable information contained in these components—no further learning can occur, as the remaining signal has already been captured.

***Remark* 5.11.** We observe that for components N beyond $\mathrm{r}_{\mathrm{cutoff}}(t)$, *i.e.* for $N > \mathrm{r}_{\mathrm{cutoff}}$, the reconstruction error continues to decrease. This suggests that additional information remains to be captured within the functional eigenspace spanned by the components associated with smaller eigenvalues.

5.4.2 . Incomplete Flattening and Adaptive Strategies

In practice, some experiments reveal that the flattening phenomenon may remain incomplete, with $\lim_{t\to\infty} N_{\mathrm{flat}} = N_{\mathrm{flat}}^{\infty} > 0$. In such cases, the system stays in a configuration similar to that shown in Figure 5.3c and never (at least within a reasonable number of iterations) reaches the fully flattened state illustrated in Figure 5.3d. This naturally raises the question: *what happens if we adjust the cutoff to retain exactly N_{flat}^{∞} components ?*

We illustrate this idea in Figure 5.4 below, using the same Heat equation experiment shown in Figure 5.2.



(a) Iteration 120 of ANaGRAM with a fixed cutoff at 10^{-3}.

(b) Same iteration, but displaying an adjusted cutoff.

(c) Next iteration after applying the natural gradient with the adjusted cutoff.

Figure 5.4: **Illustration of "instant flattening" through adaptive cutoff adjustment.** In (b), the cutoff is adjusted to retain $N_{\mathrm{flat}} = 30$ components, corresponding approximately to the left "elbow" of the reconstruction error curve. In (c), after applying **a single natural gradient step with this adjusted cutoff**, the reconstruction error curve becomes completely flat for all retained components, matching the previous flattening level observed over the range $[N_{\mathrm{flat}}, \mathrm{r}_{\mathrm{cutoff}}]$. As a result, the square root of the training loss decreases by two orders of magnitude in a single step.

Empirically, performing a single natural gradient step with this adjusted cutoff often leads to an immediate and complete flattening ($N_{\mathrm{flat}} = 0$), accompanied by a sharp reduction in training loss. This sudden flattening, occurring

when the cutoff is restricted to a small number of features, is typically associated with a learning rate close to one, as determined by the line search. A plausible explanation is that this step corresponds to an iteration within the *lazy training* regime, where the NNTK remains nearly constant, and the optimization proceeds linearly (and therefore rapidly) in the space of learned features. This hypothesis, however, calls for further investigation.

These empirical observations motivate the introduction of an adaptive algorithm: by dynamically adjusting the cutoff, we aim to accelerate convergence and achieve higher accuracy. This adaptive strategy is detailed in the next section.

5.5 . Algorithmic Design: Exploiting Flattening

Building upon the empirical analysis presented in Section 5.4, we propose a principled algorithm that monitors and exploits the flattening phenomenon observed in ANaGRAM's training dynamics. The method relies on tracking the intersection between the reconstruction error and the singular values to automatically determine an adaptive cutoff, ensuring that the training reaches a prescribed target precision ϵ. This cutoff is dynamically updated at each iteration to reflect the current learning state and to prevent premature flattening, since if flattening occurs too early, the training may stagnate at a higher loss level. Conversely, when the intersection between the reconstruction error and the singular values falls below the target precision ϵ, the algorithm enforces flattening so that the final training loss reaches the desired accuracy. Importantly, this introduces a new feature: the user can now prescribe the desired final precision level, and if the algorithm converges, it will do so exactly at that target value.

The detailed description of the algorithm is provided in the following subsections.

5.5.1 . Adaptive Cutoff Strategy

We now present the adaptive cutoff strategy used to select the cutoff rank $\mathrm{r}_{\mathrm{cutoff}}$, which determines how many components of $\widehat{\Delta}$ are retained at each iteration of ANaGRAM. The algorithm dynamically adjusts $\mathrm{r}_{\mathrm{cutoff}}$ based on the intersection between the reconstruction error and the singular values.

We first define two key quantities:

Intersection rank:

$$\mathrm{r}_{\mathrm{int}}(t) := \max\left\{ j : \mathrm{RCE}_j^S(t) \leqslant \widehat{\Delta}_{t,j} \right\}, \tag{5.96}$$

Precision rank:

$$\mathrm{r}_{\epsilon}(t) := \max\left\{ j : \mathrm{RCE}_j^S(t) \geqslant \epsilon \right\}. \tag{5.97}$$

Given a target precision level $\epsilon > 0$, the cutoff rank is then selected according to the following rule:

$$\mathrm{r}_{\mathrm{cutoff}}(t) = \begin{cases} \mathrm{r}_{\mathrm{int}}(t), & \text{if } \mathrm{RCE}^S_{\mathrm{r}_{\mathrm{int}}(t)}(t) > \epsilon, \\ \mathrm{r}_{\epsilon}(t), & \text{if } \mathrm{RCE}^S_{\mathrm{r}_{\mathrm{int}}(t)}(t) \leqslant \epsilon. \end{cases} \tag{5.98}$$

The algorithm stops when $\mathrm{r}_{\epsilon}(t) = 0$, meaning that the reconstruction error RCE^S_0 (equivalent to the training error) has fallen below the prescribed precision threshold. It exploits the geometric relationship between the empirical tangent space and the functional gradient: by tracking their intersection, it maximizes the projection of the functional gradient onto the tangent space while avoiding premature flattening. Once the intersection reaches the target precision, the algorithm leverages the flattening phenomenon to attain the desired accuracy.

In the following subsection, we outline the main components of the proposed algorithm, termed *AMStraMGRAM* (Adaptive Multi-cutoff Strategy Modification for ANaGRAM).

5.5.2 . Outline of the AMStraMGRAM Algorithm

For clarity of exposition, we present in Algorithm 4 only the core elements of AMStraMGRAM, accompanied by a visual illustration in Figure 5.5, highlighting the different phases of the algorithm.

The full version, detailed in Section B.2 and summarized in Algorithm 7, accounts for several empirical irregularities observed in the evolution of the RCE and the singular values, which are discussed in greater detail in Section B.2.4.

Algorithme 4 : Outline of the Adaptative MultiCutoff Strategy for ANaGRAM (AMStraMGRAM)

Input : $u : \mathbb{R}^P \to \mathrm{L}^2(\Omega, \mu)$, $\theta_0 \in \mathbb{R}^P$, $f \in \mathrm{L}^2(\Omega, \mu)$, $(x_i) \in \Omega^S$, $\epsilon > 0$, $T_{\max} \in \mathbb{N}$

`// Initialization`

$t \leftarrow 0$

$\widehat{\phi}_0 \leftarrow (\partial_p u_{\theta_0}(x_i))_{i,p}$ for $1 \leqslant i \leqslant S$ and $1 \leqslant p \leqslant P$

$\widehat{U}_0, \widehat{\Delta}_0, \widehat{V}_0^T \leftarrow \texttt{SVD}\left(\widehat{\phi}_0\right)$

$\widehat{\nabla \mathcal{L}}_0 \leftarrow (u_{\theta_0}(x_i) - f(x_i))_i$ for $i \in 1, \ldots, S$

Compute (RCE_j^S) for all $1 \leqslant j \leqslant \mathrm{r_{svd}}$ following Equation (5.64)

repeat

  `// Compute adaptive ranks`

  Compute $\mathrm{r_{int}}$ and r_ϵ using Equations (5.96) and (5.97)

  `// Determine a final cutoff rank`

  if $RCE_{r_{int}}^S > \epsilon$ **then**

    $\mathrm{r_{cutoff}} \leftarrow \mathrm{r_{int}}$ `// Track intersection`

  else

    $\mathrm{r_{cutoff}} \leftarrow \mathrm{r}_\epsilon$ `// Lock on precision`

  `// Natural gradient step`

  Set $\widehat{\Delta}_t \leftarrow \begin{cases} \widehat{\Delta}_{t,i} & \text{if } i \leqslant \mathrm{r_{cutoff}}, \\ 0 & \text{else;} \end{cases}$

  Compute θ_{t+1} using th ANaGRAM step of Equation (5.11)

  `// Updates for next iteration`

  $\widehat{\phi}_{t+1} \leftarrow \left(\partial_p u_{\theta_{t+1}}(x_i)\right)_{i,p}$

  $\widehat{U}_{t+1}, \widehat{\Delta}_{t+1}, \widehat{V}_{t+1}^T \leftarrow \texttt{SVD}\left(\widehat{\phi}_{t+1}\right)$

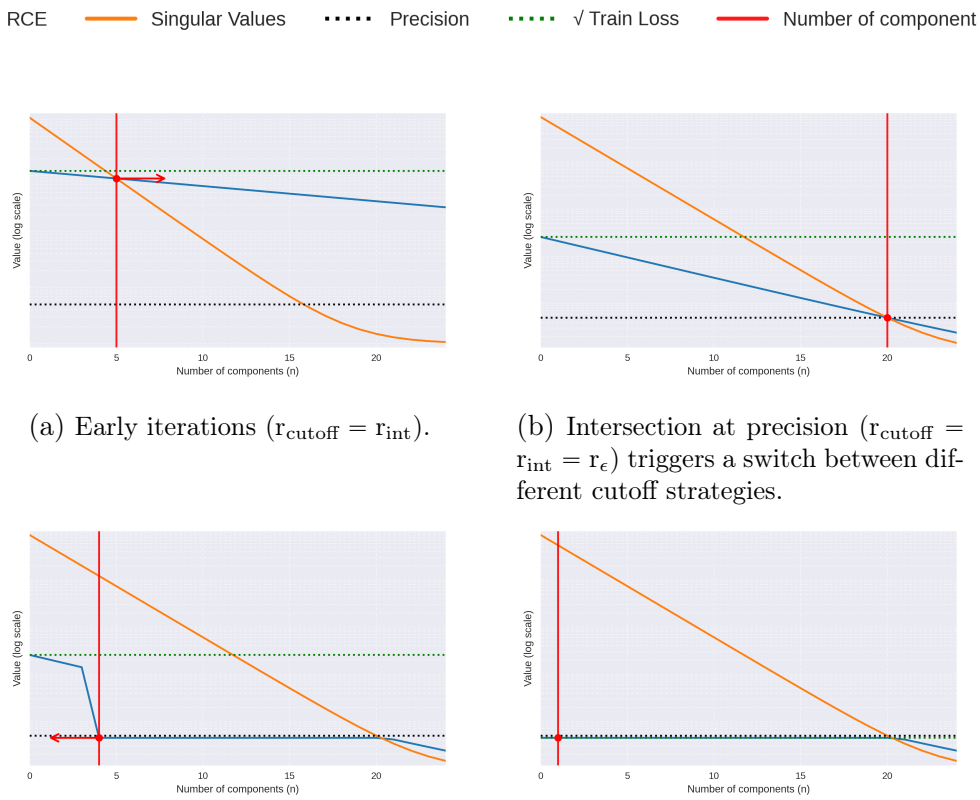
  $\widehat{\nabla \mathcal{L}}_{t+1} \leftarrow \left(u_{\theta_{t+1}}(x_i) - f(x_i)\right)_i$

  Compute RCE_j^S for all $1 \leqslant j \leqslant \mathrm{r_{svd}}$ following Equation (5.64)

  $t \leftarrow t + 1$

until $r_\epsilon = 0$ *or* $t \geqslant T_{\max}$

Output : θ_t



(a) Early iterations ($r_{cutoff} = r_{int}$).

(b) Intersection at precision ($r_{cutoff} = r_{int} = r_{\epsilon}$) triggers a switch between different cutoff strategies.

(c) Flattening: error plateaus across retained components $r_{cutoff} = r_{\epsilon}$.

(d) Final iteration: full flattening and convergence.

Figure 5.5: **Dynamics of the adaptive multi-cutoff strategy in AMStraMGRAM.** Evolution through four key phases: (a) initial exploration, (b) intersection reaching the target precision, (c) onset of the flattening regime, and (d) final convergence. Red arrows (when present) indicate the dynamics of the retained rank, pointing right for an increase and left for a decrease.

We present a summary of the experimental results in the following section.

5.6 . Experimental Results

We first compare, in Table 5.1, our method implemented in JAX (Bradbury et al., 2018), with the ANaGRAM algorithm on the benchmark problems presented in Section 4.5 of Chapter 4, using modified datasets. The implementation is available at `https://anonymous.4open.science/r/AMStraMGRAM-8D1B/`. As shown, our method consistently outperforms ANaGRAM across all equations.

Table 5.1: Performance comparison between AMStraMGRAM and ANaGRAM. The adaptive strategy demonstrates significant improvements across all benchmark problems, with L^2 error improvements of up to 8 orders of magnitude.

Experiment	Mean Squared Error (MSE)		L_2 Error	
	Ours	ANaGRAM	Ours	ANaGRAM
Heat Equation	**6.29e-29 ± 6.78e-30**	8.56e-11 ± 7.05e-11	**2.32e-14 ± 1.14e-14**	1.28e-06 ± 1.75e-06
Laplace 2D	**1.46e-28 ± 1.87e-29**	4.27e-13 ± 4.66e-13	**2.24e-15 ± 2.52e-16**	3.49e-09 ± 3.58e-09
Laplace 5D	**2.04e-08 ± 1.16e-08**	6.37e-08 ± 7.01e-08	**2.12e-05 ± 8.15e-06**	4.00e-05 ± 2.93e-05
Allen–Cahn	**3.19e-11 ± 2.37e-11**	2.19e-04 ± 4.16e-04	**5.87e-05 ± 6.25e-06**	4.32e-03 ± 5.93e-03

We then compare our method with the baseline approaches from Urbán et al. (2025) on the benchmark problems presented in their work. In our case, boundary constraints do not need to be explicitly enforced. The sampling methodology also differs slightly: we draw data points from a fixed grid, following the same procedure described in Section 4.5 of Chapter 4, whereas Urbán et al. (2025) employ mini-batching with randomly sampled points.

Table 5.2: Performance comparison between AMStraMGRAM and baseline Urbán et al. (2025) methods. Our method demonstrates improvements across benchmark problems, without requiring enforcement of boundary constraints.

Experiment	Mean Squared Error (MSE)		L_2 Error	
	Ours	SSBroyden*	Ours	SSBroyden*
One-dimensional Burgers (1DB)	**2.99e-12 ± 9.26e-13**	2.92e-10 ± 1.45e-10	**1.5e-06 ± 9.43e-7**	1.59e-06 ± 1.02e-6
Non-Linear Poisson (k=1)	**8.51e-24 ± 2.24e-24**	3.03e-16 ± 3.82e-16	6.81e-10 ± 1.41e-09	**9.29e-12 ± 5.85e-12**
Allen–Cahn (AC)	3.19e-11 ± 2.37e-11	**6.42e-12 ± 5.52e-12**	5.87e-05 ± 6.25e-06	**3.94e-06 ± 1.72e-06**

* refer to method from Urbán et al. (2025) with adaptive sampling and hard constraint enforcement on boundary conditions.

5.7 . Limitations

Despite its strong performance, AMStraMGRAM may still suffer from overfitting, especially in problems involving sharp transitions, such as the Allen–Cahn equation. While the algorithm drives the training error down to machine precision on the sampled points, the learned function can develop high-frequency oscillations between them—especially in regions of high curvature where the approximation is most challenging. These artifacts, visible as "overfitting lines" in Figure 5.6, reflect the underlying sampling grid (notably around $x = \pm 0.5$). They arise because the SVD cutoff effectively projects each update onto a low-rank subspace of the tangent space. This subspace often aligns with the grid axes, inducing an anisotropic smoothing that perfectly fits data along sampled lines but interpolates poorly in the under-sampled regions between them. Once the flattening phase begins, training enters a quasi-linear

regime that can "lock in" these geometric artifacts.

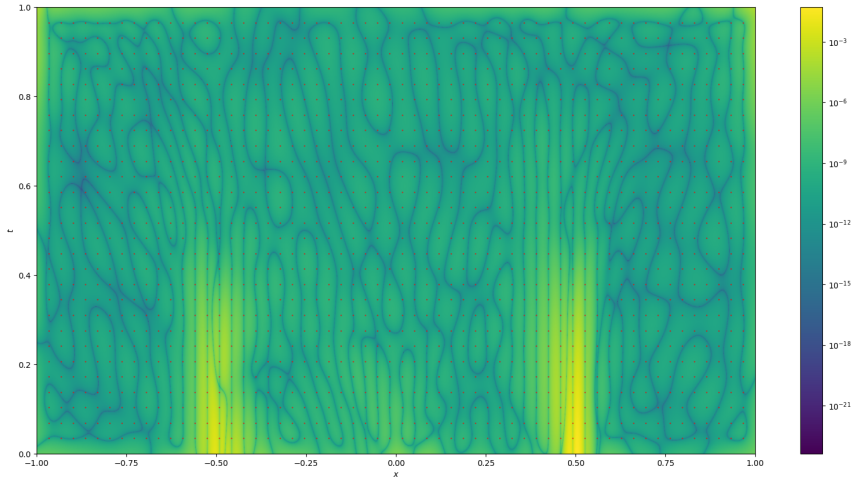


Figure 5.6: **Allen–Cahn overfitting.** Residual lines align with the sampling grid. Low-rank (post-cutoff) tangent projections fit perfectly along the sampled lines, whereas under-sampled regions with sharp transitions develop weakly constrained oscillations.

Conversely, increasing the sampling density in those poorly approximated regions effectively mitigates overfitting, yielding a more accurate global approximation, as shown in Figure 5.7.

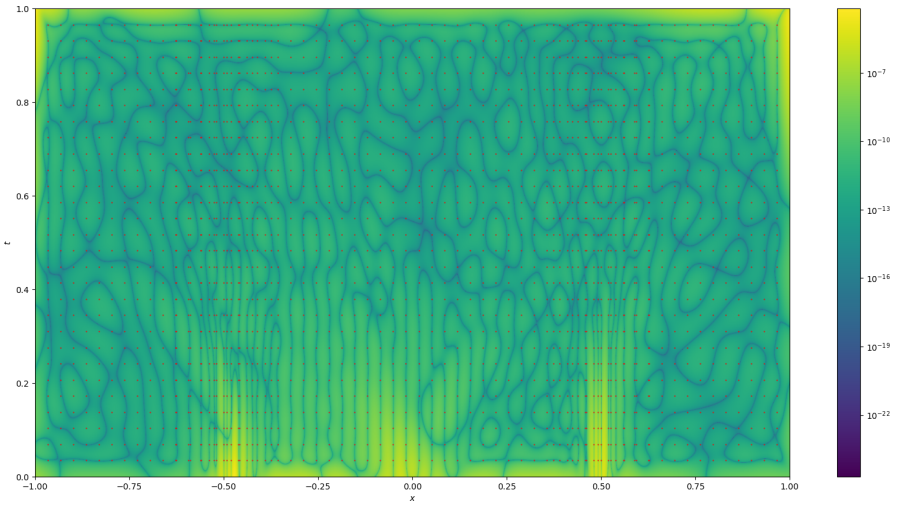


Figure 5.7: **Allen–Cahn oversampling.** Increasing sampling density in overfitted regions reduces overfitting artifacts and improves overall accuracy.

This phenomenon highlights that, although our method substantially improves upon ANaGRAM, the quality of the final solution remains fundamentally constrained by the sampling strategy. Addressing such overfitting requires

co-designing the sampler and the optimizer. Promising directions include adaptive sampling, where new collocation points are iteratively added in regions of high reconstruction error, or curriculum-based approaches that progressively refine the sampling grid during training.

5.8 . Conclusion

In this work, we introduced AMStraMGRAM, an adaptive multi-cutoff strategy that enhances the ANaGRAM natural gradient method for training Physics-Informed Neural Networks (PINNs). Our study provided a comprehensive theoretical analysis of regularization, establishing its connection with Green's function theory and demonstrating its central role in building stable and efficient numerical solvers. We also clarified how additional, spurious sources of ill-conditioning can arise from numerical implementations, emphasizing that regularization is not a mere computational fix but a fundamental component of the underlying theory.

Furthermore, we developed an analytical framework to explain ANaGRAM's convergence behavior, uncovering the *flattening phenomenon* that characterizes its training dynamics. Building on this insight, AMStraMGRAM automatically adapts its cutoff regularization during training, improving convergence and precision. Nevertheless, we observed that the method may still exhibit overfitting in problems with sharp transitions, underscoring the importance of sampling strategies in achieving optimal accuracy.

Future work will focus on integrating residual-based sampling methods, deriving formal convergence guarantees for the adaptive cutoff scheme, and extending the approach to higher-dimensional PDEs and more complex geometries. In addition, exploring the interplay between network architecture, optimization dynamics, and data sampling will be crucial for balancing representational capacity and stability. Ultimately, this work suggests that, with careful algorithmic design, PINNs can reach the level of precision required for practical scientific computing, paving the way toward robust mesh-free numerical methods in computational science.

6 - Conclusion and Perspectives

Summary of the Work

The central goal of this dissertation has been to advance the theoretical and algorithmic understanding of *Physics-Informed Neural Networks* (PINNs) through the lens of differential geometry and functional analysis. By revisiting the foundations of natural gradient methods and reproducing kernel, we introduced a principled and computationally efficient framework for their application to Physics Informed Neural Networks (PINNs).

The first contribution of this work is the development of the **ANaGRAM algorithm**, a natural gradient method whose design stems from a rigorous reformulation of optimization in function space. ANaGRAM provides a theoretically grounded simplification of existing natural-gradient-based approaches, reducing their computational complexity from $O(P^3)$ to $O(\min(PN^2, P^2N))$, thereby placing it only a factor $\min(P, N)$ above standard stochastic gradient descent. Beyond its efficiency, ANaGRAM preserves the geometric optimality of the natural gradient while revealing deep connections with the Green's function of the underlying differential operator. This result offers a new theoretical interpretation of PINN training as an *optimal linear update* process in a properly regularized function space.

The second contribution is the introduction of **AMStramGRAM**, an adaptive multi-cutoff extension of ANaGRAM. This method provides a principled way to adjust the spectral regularization throughout training. By adapting cutoff thresholds based on the spectral evolution of the learning process, AMStramGRAM automatically selects the most relevant subspace of directions at each iteration, reaching machine-level precision without requiring manual hyperparameter tuning. This adaptation strategy not only stabilizes training but also prevents over-regularization, ensuring optimal generalization and convergence behavior.

From an empirical standpoint, both ANaGRAM and AMStramGRAM demonstrate competitive or superior performance compared to state-of-the-art PINN optimizers across a range of benchmark PDE problems. In particular, on linear operators, the method achieves L^2 errors as low as 10^{-14}, a level of accuracy comparable to high-order finite element methods. To the best of our knowledge, this is the first time a PINN-based approach has reached numerical precision competitive with classical deterministic solvers. This achievement suggests that, under suitable regularization, neural networks can serve as mesh-free numerical solvers that combine the flexibility of learning methods with the rigor of traditional numerical analysis.

Broader Impact and Theoretical Implications

Beyond the algorithmic advances, this work makes a conceptual contribution toward bridging the gap between *machine learning* and *applied mathematics*. By formulating PINN training within a rigorous Hilbert space and Reproducing Kernel Hilbert Spaces (RKHS) framework, we establish a direct correspondence between functional optimization, kernel methods, and operator theory. In particular, we demonstrate that natural gradients in PINNs correspond to optimal descent flows following the Green's function associated with the target operator. This result provides a precise mathematical interpretation of neural operator learning, explicitly linking their training dynamics to classical inverse problem theory.

Furthermore the analysis of the regularization mechanisms underlying ANaGRAM and AMStramGRAM elucidates the role of spectral filtering in stabilizing the learning process. It offers a unifying theoretical foundation connecting natural gradient descent, Tikhonov regularization, and spectral cutoff techniques, thereby reconciling contemporary deep learning optimization with long-established principles of functional regularization in applied mathematics.

Perspectives and Future Work

While the present work establishes a solid foundation, several research directions naturally arise:

- **Adaptive sampling and collocation strategies.** A key challenge that remains open is the optimal selection of collocation points in PINNs. Our ANaGRAM framework provides an explicit criterion that can be exploited in conjunction with AMStramGRAM's adaptive regularization scheme to guide sampling toward the most informative regions of the domain.

- **Data assimilation and regularization effects.** Integrating observational data into this theoretical setting would enable the study of implicit regularization phenomena arising from data-driven constraints, further bridging the gap between data assimilation and operator learning.

- **Algorithmic extensions.** Incorporating common optimization techniques such as momentum, could further accelerate convergence and stability. Similarly, hybridization with second-order methods offers a promising direction for efficiently handling stiff or nonlinear PDEs.

- **Extension to operator learning.** The geometric insights of ANaGRAM naturally extend to neural operators and other infinite-dimensional

learning settings, providing a promising avenue for scalable and interpretable operator learning.

- **Applications to control theory.** Finally, the framework could be applied to Hamilton–Jacobi–Bellman (HJB) equations, potentially offering a new generation of high-precision, learning-based control solvers.

Outlook

In summary, this dissertation demonstrates that the principles of Reproducing Kernel Hilbert Spaces (RKHSs), differential geometry and spectral regularization can be leveraged to construct theoretically sound and practically efficient learning algorithms for scientific computing. The results obtained here bridge the conceptual gap between natural gradient optimization, functional analysis, and numerical PDE solvers. The next pivotal challenge lies in extending these ideas to nonlinear operators and high-dimensional systems, ultimately paving the way for a new generation of adaptive, learning-based solvers that merge the rigor of mathematics with the flexibility of modern machine learning.

Bibliography

Abramowitz, M. and I. A. Stegun (1968): *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, vol. 55, US Government printing office.

Adcock, B. and D. Huybrechs (2019): "Frames and Numerical Approximation," *SIAM Review*, 61, 443–473.

——— (2020a): "Frames and Numerical Approximation II: Generalized Sampling," .

——— (2020b): "Frames and Numerical Approximation II: Generalized Sampling," .

Albawi, S., T. A. Mohammed, and S. Al-Zawi (2017): "Understanding of a Convolutional Neural Network," in *2017 International Conference on Engineering and Technology (ICET)*, Ieee, 1–6.

Alvarez, M. A., L. Rosasco, and N. D. Lawrence (2012): "Kernels for Vector-Valued Functions: A Review," .

Amari, S. (1967): "A Theory of Adaptive Pattern Classifiers," *IEEE Transactions on Electronic Computers*, 299–307.

Amari, S.-I. (1998): "Natural Gradient Works Efficiently in Learning," *Neural computation*, 10, 251–276.

Antoine, J.-P. and C. Trapani (2009): *Partial Inner Product Spaces: Theory and Applications*, vol. 1986 of *Lecture Notes in Mathematics*, Berlin, Heidelberg: Springer.

Aronszajn, N. (1950): "Theory of Reproducing Kernels," *Transactions of the American mathematical society*, 68, 337–404.

Arora, S., N. Cohen, N. Golowich, and W. Hu (2018): "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks," *arXiv preprint arXiv:1810.02281*.

Asch, M., M. Bocquet, and M. Nodet (2016): *Data Assimilation: Methods, Algorithms, and Applications*, SIAM.

Bach, F. (2024): *Learning Theory from First Principles*, MIT press.

Bach, F. and L. Chizat (2021): "Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization," *arXiv preprint arXiv:2110.08084*.

Bai, Q., S. Rosenberg, and W. Xu (2022): "A Geometric Understanding of Natural Gradient," .

Barucq, H., M. Duprez, F. Faucher, E. Franck, F. Lecourtier, V. Lleras, V. Michel-Dansac, and N. Victorion (2025): "Enriching Continuous Lagrange Finite Element Approximation Spaces Using Neural Networks," .

Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind (2018): "Automatic Differentiation in Machine Learning: A Survey," *Journal of Marchine Learning Research*, 18, 1–43.

Bélieres–Frendo, A., E. Franck, V. Michel-Dansac, and Y. Privat (2024): "Geometric Shape Optimization for Dirichlet Energy with Physics Informed and Symplectic Neural Networks," *Learning*.

Benner, P., A. Cohen, M. Ohlberger, and K. Willcox (2017): *Model Reduction and Approximation: Theory and Algorithms*, SIAM.

Berezansky, Y. M., Z. G. Sheftel, and G. F. Us (1996a): *Functional Analysis: Vol. I*, vol. 85 of *Operator Theory Advances and Applications*, Birkhäuser.

——— (1996b): *Functional Analysis. Vol. II*, vol. 86 of *Operator Theory Advances and Applications*, Birkhäuser.

Berg, J. and K. Nyström (2018): "A Unified Deep Artificial Neural Network Approach to Partial Differential Equations in Complex Geometries," *Neurocomputing*, 317, 28–41.

Berner, J., P. Grohs, G. Kutyniok, and P. Petersen (2021): "The Modern Mathematics of Deep Learning," *arXiv preprint arXiv:2105.04026*, 86–114.

Beyer, H.-G. (2001): *The Theory of Evolution Strategies*, Springer Science & Business Media.

Bishop, C. M. and N. M. Nasrabadi (2006): *Pattern Recognition and Machine Learning*, vol. 4, Springer.

Bon, D., B. Caris, and O. Mula (2025): "Stable Nonlinear Dynamical Approximation with Dynamical Sampling," .

Boullé, N. and A. Townsend (2024): "Chapter 3 - A Mathematical Guide to Operator Learning," in *Handbook of Numerical Analysis*, ed. by S. Mishra and A. Townsend, Elsevier, vol. 25 of *Numerical Analysis Meets Machine Learning*, 83–125.

Boursier, E., L. Pillaud-Vivien, and N. Flammarion (2022): "Gradient Flow Dynamics of Shallow ReLU Networks for Square Loss and Orthogonal Inputs," .

Bradbury, J., R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang (2018): "JAX: Composable Transformations of Python+NumPy Programs," .

Brezis, H. (2011): *Functional Analysis, Sobolev Spaces and Partial Differential Equations*, New York, NY: Springer New York.

Brunton, S. L., J. L. Proctor, and J. N. Kutz (2016): "Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems," *Proceedings of the National Academy of Sciences*, 113, 3932–3937.

Butcher, J. C. (2016): *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons.

Cai, T., R. Gao, J. Hou, S. Chen, D. Wang, D. He, Z. Zhang, and L. Wang (2019): "Gram-Gauss-Newton Method: Learning Overparameterized Neural Networks for Regression Problems," .

Cao, Y., Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu (2020): "Towards Understanding the Spectral Bias of Deep Learning," .

Chan, T. F. and T. P. Mathew (1994): "Domain Decomposition Algorithms," *Acta numerica*, 3, 61–143.

Chen, R. T., Y. Rubanova, J. Bettencourt, and D. K. Duvenaud (2018): "Neural Ordinary Differential Equations," *Advances in neural information processing systems*, 31.

Chen, S., P. Givi, C. Zheng, and X. Jia (2024): "Physics-Enhanced Neural Operator for Simulating Turbulent Transport," .

Chizat, L. and F. Bach (2018): "On the Global Convergence of Gradient Descent for Over-Parameterized Models Using Optimal Transport," *Advances in neural information processing systems*, 31.

Chizat, L., E. Oyallon, and F. Bach (2019): "On Lazy Training in Differentiable Programming," *Advances in Neural Information Processing Systems*, 32.

Cho, K., B. van Merrienboer, D. Bahdanau, and Y. Bengio (2014): "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," .

Cohen, A. and R. DeVore (2015): "Approximation of High-Dimensional Parametric PDEs," *Acta Numerica*, 24, 1–159.

Cohen, A. and G. Migliorati (2017): "Optimal Weighted Least-Squares Methods," *The SMAI Journal of computational mathematics*, 3, 181–203.

Corliss, G. F. (1988): "Applications of Differentiation Arithmetic," in *Reliability in Computing*, Elsevier, 127–148.

Cuomo, S., V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli (2022): "Scientific Machine Learning through Physics-Informed Neural Networks: Where We Are and What's Next," .

Dangel, F., J. Müller, and M. Zeinhofer (2025): "Kronecker-Factored Approximate Curvature for Physics-Informed Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 37, 34582–34636.

Daubechies, I., R. DeVore, S. Foucart, B. Hanin, and G. Petrova (2022): "Nonlinear Approximation and (Deep) $$$\backslash$mathrm ${$ReLU$}$ $$ ReLU Networks," *Constructive Approximation*, 55, 127–172.

Davis, P. J. and P. Rabinowitz (2007): *Methods of Numerical Integration*, Courier Corporation.

De Ryck, T. (2024): "Mathematical Guarantees for Physics-Informed Machine Learning," Ph.D. thesis, ETH Zurich.

De Ryck, T., F. Bonnet, S. Mishra, and E. de Bézenac (2024): "An Operator Preconditioning Perspective on Training in Physics-Informed Machine Learning," .

De Ryck, T., S. Lanthaler, and S. Mishra (2021): "On the Approximation of Functions by Tanh Neural Networks," *Neural Networks*, 143, 732–750.

de Wolff, T., H. Carrillo, L. Martí, and N. Sanchez-Pi (2021): "Assessing Physics Informed Neural Networks in Ocean Modelling and Climate Change Applications," in *AI: Modeling Oceans and Climate Change Workshop at ICLR 2021*.

DeepMind, I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, A. Dedieu, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou, S. Kapturowski, T. Keck, I. Kemaev, M. King, M. Kunesch, L. Martens, H. Merzic, V. Mikulik, T. Norman, G. Papamakarios, J. Quan, R. Ring, F. Ruiz, A. Sanchez, L. Sartran, R. Schneider, E. Sezener, S. Spencer, S. Srinivasan, M. Stanojević, W. Stokowiec, L. Wang, G. Zhou, and F. Viola (2020): "The DeepMind JAX Ecosystem," .

Deshpande, M., S. Agarwal, and A. K. Bhattacharya (2022): "Investigations on Convergence Behaviour of Physics Informed Neural Networks across Spectral Ranges and Derivative Orders," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 1172–1179.

Dissanayake, M. W. M. G. and N. Phan-Thien (1994): "Neural-network-based Approximations for Solving Partial Differential Equations," *Communications in Numerical Methods in Engineering*, 10, 195–201.

Dolbeault, M. and A. Cohen (2022a): "Optimal Pointwise Sampling for L2 Approximation," *Journal of Complexity*, 68, 101602.

——— (2022b): "Optimal Sampling and Christoffel Functions on General Domains," *Constructive Approximation*, 56, 121–163.

Doumèche, N. (2025): "Physics-Informed Machine Learning : A Mathematical Framework with Applications to Time Series Forecasting," Ph.D. thesis, Sorbonne Université.

Doumèche, N., F. Bach, G. Biau, and C. Boyer (2024a): "Physics-Informed Kernel Learning," .

——— (2024b): "Physics-Informed Machine Learning as a Kernel Method," in *The Thirty Seventh Annual Conference on Learning Theory*, PMLR, 1399–1450.

——— (2025): "Fast Kernel Methods: Sobolev, Physics-Informed, and Additive Models," .

Dubey, S. R., S. K. Singh, and B. B. Chaudhuri (2022): "Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark," *Neurocomputing*, 503, 92–108.

Duffy, D. G. (2015): *Green's Functions with Applications*, Chapman and Hall/CRC.

E, W. and B. Yu (2017): "The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems," *arXiv:1710.00211 [cs, stat]*.

Edmunds, D. E. and W. D. Evans (2018): *Spectral Theory and Differential Operators*, Oxford University Press.

Elman, J. L. (1990): "Finding Structure in Time," *Cognitive Science*, 14, 179–211.

Emmerich, M., O. M. Shir, and H. Wang (2018): "Evolution Strategies," in *Handbook of Heuristics*, ed. by R. Martí, P. Panos, and M. G. C. Resende, Cham: Springer International Publishing, 1–31.

Evans, L. (2018): *Measure Theory and Fine Properties of Functions*, Routledge.

Fiorini, C., C. Flint, L. Fostier, E. Franck, R. Hashemi, V. Michel-Dansac, and W. Tenachi (2025): "Generalizing the SINDy Approach with Nested Neural Networks," .

Fletcher, R. (2000): *Practical Methods of Optimization*, John Wiley & Sons.

Franck, E., V. Michel-Dansac, L. Navoret, and V. Vigon (2025): "Neural Semi-Lagrangian Method for High-Dimensional Advection-Diffusion Problems," .

Gallot, S., D. Hulin, and J. Lafontaine (1990): *Riemannian Geometry*, Universitext, Berlin, Heidelberg: Springer Berlin Heidelberg.

Gelfand, I. M., S. V. Fomin, and R. A. Silverman (2000): *Calculus of Variations*, Courier Corporation.

George, T., C. Laurent, X. Bouthillier, N. Ballas, and P. Vincent (2018): "Fast Approximate Natural Gradient Descent in a Kronecker Factored Eigenbasis," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 31.

Glorot, X. and Y. Bengio (2010): "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

Goldberg, Y. (2022): *Neural Network Methods for Natural Language Processing*, Springer Nature.

Goodfellow, I., Y. Bengio, and A. Courville (2016): *Deep Learning*, MIT press.

Griewank, A. and A. Walther (2008): *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM.

Grosse, R. and J. Martens (2016): "A Kronecker-factored Approximate Fisher Matrix for Convolution Layers," in *Proceedings of The 33rd International Conference on Machine Learning*, PMLR, 573–582.

Grossmann, T. G., U. J. Komorowska, J. Latz, and C.-B. Schönlieb (2024): "Can Physics-Informed Neural Networks Beat the Finite Element Method?" *IMA Journal of Applied Mathematics*, 89, 143–174.

Gruhlke, R., A. Nouy, and P. Trunschke (2024): "Optimal Sampling for Stochastic and Natural Gradient Descent," .

Guzmán-Cordero, A., F. Dangel, G. Goldshlager, and M. Zeinhofer (2025): "Improving Energy Natural Gradient Descent through Woodbury, Momentum, and Randomization," .

Haberstich, C., A. Nouy, and G. Perrin (2022): "Boosted Optimal Weighted Least-Squares," *Mathematics of Computation*, 91, 1281–1315.

Hansen, N. (2016): "The CMA Evolution Strategy: A Tutorial," *arXiv preprint arXiv:1604.00772*.

Heskes, T. (2000): "On "Natural" Learning and Pruning in Multilayered Perceptrons," *Neural Computation*, 12, 881–901.

Hochreiter, S. and J. Schmidhuber (1997): "Long Short-Term Memory," *Neural computation*, 9, 1735–1780.

Hofmann, T., B. Schölkopf, and A. J. Smola (2008): "Kernel Methods in Machine Learning," *The Annals of Statistics*, 36, 1171–1220.

Iserles, A. (2009): *A First Course in the Numerical Analysis of Differential Equations*, 44, Cambridge university press.

Jacot, A., F. Gabriel, and C. Hongler (2018): "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," *Advances in neural information processing systems*, 31.

Jacot, A., F. Ged, B. Şimşek, C. Hongler, and F. Gabriel (2022): "Saddle-to-Saddle Dynamics in Deep Linear Networks: Small Initialization Training, Symmetry, and Sparsity," .

Jagtap, A. D. and G. E. Karniadakis (2021): "Extended Physics-informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations." in *AAAI Spring Symposium: MLPS*, vol. 10.

Jagtap, A. D., E. Kharazmi, and G. E. Karniadakis (2020): "Conservative Physics-Informed Neural Networks on Discrete Domains for Conservation Laws: Applications to Forward and Inverse Problems," *Computer Methods in Applied Mechanics and Engineering*, 365, 113028.

Jerrell, M. E. (1997): "Automatic Differentiation and Interval Arithmetic for Estimation of Disequilibrium Models," *Computational Economics*, 10, 295–316.

Jin, X., S. Cai, H. Li, and G. E. Karniadakis (2021): "NSFnets (Navier-Stokes Flow Nets): Physics-informed Neural Networks for the Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, 426, 109951.

Jnini, A., F. Vella, and M. Zeinhofer (2024): "Gauss-Newton Natural Gradient Descent for Physics-Informed Computational Fluid Dynamics," .

Karnakov, P., S. Litvinov, and P. Koumoutsakos (2022): "Solving Inverse Problems in Physics by Optimizing a Discrete Loss Is Much More Faster and Accurate without Neural Networks," .

Kharazmi, E., Z. Zhang, and G. E. Karniadakis (2019): "Variational Physics-Informed Neural Networks For Solving Partial Differential Equations," .

——— (2021): "Hp-VPINNs: Variational Physics-Informed Neural Networks with Domain Decomposition," *Computer Methods in Applied Mechanics and Engineering*, 374, 113547.

Khodayi-Mehr, R. and M. Zavlanos (2020): "VarNet: Variational Neural Networks for the Solution of Partial Differential Equations," in *Learning for Dynamics and Control*, PMLR, 298–307.

Kim, C., S. Kim, J. Kim, D. Lee, and S. Kim (2021): "Automated Learning Rate Scheduler for Large-batch Training," .

Kingma, D. P. and J. Ba (2014): "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*.

Kirsch, A. (2021): *An Introduction to the Mathematical Theory of Inverse Problems*, vol. 120 of *Applied Mathematical Sciences*, Cham: Springer International Publishing.

Kissas, G., Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris (2020): "Machine Learning in Cardiovascular Flows Modeling: Predicting Arterial Blood Pressure from Non-Invasive 4D Flow

MRI Data Using Physics-Informed Neural Networks," *Computer Methods in Applied Mechanics and Engineering*, 358, 112623.

Kovachki, N., Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar (2021): "Neural Operator: Learning Maps between Function Spaces," *arXiv preprint arXiv:2108.08481*.

Kovachki, N. B., S. Lanthaler, and A. M. Stuart (2024): "Chapter 9 - Operator Learning: Algorithms and Analysis," in *Handbook of Numerical Analysis*, ed. by S. Mishra and A. Townsend, Elsevier, vol. 25 of *Numerical Analysis Meets Machine Learning*, 419–467.

Kress, R. (2014): *Linear Integral Equations*, vol. 82 of *Applied Mathematical Sciences*, New York, NY: Springer.

Krishnapriyan, A., A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney (2021): "Characterizing Possible Failure Modes in Physics-Informed Neural Networks," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 34, 26548–26560.

Kumar, S. K. (2017): "On Weight Initialization in Deep Neural Networks," *arXiv preprint arXiv:1704.08863*.

Lagaris, I. E., A. Likas, and D. I. Fotiadis (1998): "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," *IEEE transactions on neural networks*, 9, 987–1000.

Lau, G. K. R., A. Hemachandra, S.-K. Ng, and B. K. H. Low (2024): "PINNACLE: PINN Adaptive ColLocation and Experimental Points Selection," in *The Twelfth International Conference on Learning Representations*.

LeCun, Y., Y. Bengio, and G. Hinton (2015): "Deep Learning," *nature*, 521, 436–444.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998): "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, 86, 2278–2324.

Lemieux, C. (2009): *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer Science & Business Media.

Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken (1993): "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural networks*, 6, 861–867.

Lewkowycz, A. (2021): "How to Decay Your Learning Rate," .

Li, Z. and S. Arora (2019): "An Exponential Learning Rate Schedule for Deep Learning," .

Li, Z., N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar (2020): "Fourier Neural Operator for Parametric Partial Differential Equations," *arXiv preprint arXiv:2010.08895*.

Li, Z., H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar (2021): "Physics-Informed Neural Operator for Learning Partial Differential Equations," *arXiv:2111.03794 [cs, math]*.

Linnainmaa, S. (1976): "Taylor Expansion of the Accumulated Rounding Error," *BIT Numerical Mathematics*, 16, 146–160.

Liu, D. and Y. Wang (2021): "A Dual-Dimer Method for Training Physics-Constrained Neural Networks with Minimax Architecture," *Neural Networks*, 136, 112–125.

Liu, D. C. and J. Nocedal (1989): "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, 45, 503–528.

Liu, S., C. Su, J. Yao, Z. Hao, H. Su, Y. Wu, and J. Zhu (2024): "Preconditioning for Physics-Informed Neural Networks," .

Liu, Z., Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark (2025): "KAN: Kolmogorov-Arnold Networks," .

Lu, L., P. Jin, and G. E. Karniadakis (2021a): "DeepONet: Learning Nonlinear Operators for Identifying Differential Equations Based on the Universal Approximation Theorem of Operators," *Nature Machine Intelligence*, 3, 218–229.

Lu, L., X. Meng, Z. Mao, and G. E. Karniadakis (2021b): "DeepXDE: A Deep Learning Library for Solving Differential Equations," *SIAM Review*, 63, 208–228.

Luo, K., J. Zhao, Y. Wang, J. Li, J. Wen, J. Liang, H. Soekmadji, and S. Liao (2025): "Physics-Informed Neural Networks for PDE Problems: A Comprehensive Review," *Artificial Intelligence Review*, 58, 323.

Martens, J. and R. Grosse (2015): "Optimizing Neural Networks with Kronecker-factored Approximate Curvature," in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, 2408–2417.

Mary, X. (2003): "Hilbertian Subspaces, Subdualities and Applications," Ph.D. thesis, INSA de Rouen.

McClenny, L. and U. Braga-Neto (2022): "Self-Adaptive Physics-Informed Neural Networks Using a Soft Attention Mechanism," .

Meng, C., S. Griesemer, D. Cao, S. Seo, and Y. Liu (2025): "When Physics Meets Machine Learning: A Survey of Physics-Informed Machine Learning," *Machine Learning for Computational Science and Engineering*, 1, 20.

Menier, E. (2024): "Deep Learning for Reduced Order Modeling," Ph.D. thesis, Université Paris-Saclay.

Moukalled, F., L. Mangani, and M. Darwish (2016): *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*, vol. 113 of *Fluid Mechanics and Its Applications*, Cham: Springer International Publishing.

Mukherjee, A. and J. Liu (2023): "Bridging Physics-Informed Neural Networks with Reinforcement Learning: Hamilton-Jacobi-Bellman Proximal Policy Optimization (HJBPPO)," .

Müller, J. and M. Zeinhofer (2023): "Achieving High Accuracy with PINNs via Energy Natural Gradient Descent," in *International Conference on Machine Learning*, PMLR, 25471–25485.

——— (2024): "Position: Optimization in SciML Should Employ the Function Space Geometry," in *Forty-First International Conference on Machine Learning*.

Narkhede, M. V., P. P. Bartakke, and M. S. Sutaone (2022): "A Review on Weight Initialization Strategies for Neural Networks," *Artificial intelligence review*, 55, 291–322.

Nastorg, M. (2024): "Scalable GNN Solutions for CFD Simulations," Ph.D. thesis, Université Paris-Saclay.

Neal, R. M. (1996): "Priors for Infinite Networks," in *Bayesian Learning for Neural Networks*, Springer, 29–53.

Nesterov, Y. E. (1983): "A Method of Solving a Convex Programming Problem with Convergence Rate O\backslashbigl(K^2\backslashbigr)," in *Doklady Akademii Nauk*, Russian Academy of Sciences, vol. 269, 543–547.

Neuberger, J. (2009): *Sobolev Gradients and Differential Equations*, Springer Science & Business Media.

Nguyen, T. N. K., T. Dairay, R. Meunier, C. Millet, and M. Mougeot (2023): "Fixed-Budget Online Adaptive Learning for Physics-Informed Neural Networks. Towards Parameterized Problem Inference," in *Computational Science – ICCS 2023*, ed. by J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Cham: Springer Nature Switzerland, 453–468.

Nguyen, T. N. K., T. Dairay, R. Meunier, and M. Mougeot (2022a): "Physics-Informed Neural Networks for Non-Newtonian Fluid Thermo-Mechanical Problems: An Application to Rubber Calendering Process," *Engineering Applications of Artificial Intelligence*, 114, 105176.

Nguyen, T.-N.-K., C. Millet, T. Dairay, R. Meunier, and M. Mougeot (2022b): "Adaptive Physics-Informed Deep Learning for the Parameterization of Orographic Gravity Waves," in *AGU Fall Meeting Abstracts*, vol. 2022, U35C–0534.

Nouy, A. and B. Michel (2024): "Weighted Least-Squares Approximation with Determinantal Point Processes and Generalized Volume Sampling," .

Ollivier, Y. (2015a): "Riemannian Metrics for Neural Networks I: Feedforward Networks," *Information and Inference: A Journal of the IMA*, 4, 108–153.

——— (2015b): "Riemannian Metrics for Neural Networks II: Recurrent Networks and Learning Symbolic Data Sequences," *Information and Inference: A Journal of the IMA*, 4, 154–193.

——— (2017): "True Asymptotic Natural Gradient Optimization," .

Paulsen, R. R. and T. B. Moeslund (2020): *Introduction to Medical Image Analysis*, Springer Nature.

Paulsen, V. I. and M. Raghupathi (2016): *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, vol. 152, Cambridge university press.

Pflaum, M. J. (2001): *Analytic and Geometric Study of Stratified Spaces*, vol. 1768 of *Lecture Notes in Mathematics*, Berlin, Heidelberg: Springer.

Qian, N. (1999): "On the Momentum Term in Gradient Descent Learning Algorithms," *Neural networks*, 12, 145–151.

Rahaman, N., A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville (2019): "On the Spectral Bias of Neural Networks," in *International Conference on Machine Learning*, PMLR, 5301–5310.

Raissi, M., H. Babaee, and P. Givi (2019a): "Deep Learning of Turbulent Scalar Mixing," *Physical Review Fluids*, 4, 124501.

Raissi, M., P. Perdikaris, and G. Karniadakis (2019b): "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.

Raissi, M., Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis (2019c): "Deep Learning of Vortex-Induced Vibrations," *Journal of Fluid Mechanics*, 861, 119–137.

Raissi, M., A. Yazdani, and G. E. Karniadakis (2020): "Hidden Fluid Mechanics: Learning Velocity and Pressure Fields from Flow Visualizations," *Science*, 367, 1026–1030.

Rathore, P., W. Lei, Z. Frangella, L. Lu, and M. Udell (2024): "Challenges in Training PINNs: A Loss Landscape Perspective," .

Rosenblatt, F. (1958): "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological review*, 65, 386.

Rudner, T. G., F. Wenzel, Y. W. Teh, and Y. Gal (2019): "The Natural Neural Tangent Kernel: Neural Network Training Dynamics under Natural Gradient Descent," in *4th Workshop on Bayesian Deep Learning (NeurIPS 2019)*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1985): "Learning Internal Representations by Error Propagation," Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science.

Sahli Costabal, F., Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl (2020): "Physics-Informed Neural Networks for Cardiac Activation Mapping," *Frontiers in Physics*, 8.

Santambrogio, F. (2017): "{Euclidean, Metric, and Wasserstein} Gradient Flows: An Overview," *Bulletin of Mathematical Sciences*, 7, 87–154.

Schilders, W. H., H. A. van der Vorst, and J. Rommes (2008): *Model Order Reduction: Theory, Research Aspects and Applications*, Springer Science & Business Media.

Schwartz, L. (1964): "Sous-espaces hilbertiens d'espaces vectoriels topologiques et noyaux associés (Noyaux reproduisants)," *Journal d'Analyse Mathématique*, 13, 115–256.

——— (1966): "Théorie Des Distributions, Paris, Hermann, 1966," *MR209834*.

Schwencke, N. and C. Furtlehner (2025): "ANaGRAM: A Natural Gradient Relative to Adapted Model for Efficient PINNs Learning," in *The Thirteenth International Conference on Learning Representations*.

Schwencke, N., C. Rousselot, A. Shilova, and C. Furtlehner (2025): "AMStramGRAM: Adaptive Multi-Cutoff Strategy Modification for ANaGRAM," .

Shen, J., T. Tang, and L.-L. Wang (2011): *Spectral Methods: Algorithms, Analysis and Applications*, Springer Science & Business Media.

Shi, Z., Z. Hu, M. Lin, and K. Kawaguchi (2024): "Stochastic Taylor Derivative Estimator: Efficient Amortization for Arbitrary Differential Operators," *Advances in Neural Information Processing Systems*, 37, 122316–122353.

Shilova, A., T. Delliaux, P. Preux, and B. Raffin (2024): "Learning HJB Viscosity Solutions with PINNs for Continuous-Time Reinforcement Learning," .

Siegel, J. W., Q. Hong, X. Jin, W. Hao, and J. Xu (2023): "Greedy Training Algorithms for Neural Networks and Applications to PDEs," *Journal of Computational Physics*, 484, 112084.

Singer, M. F. (2009): "Introduction to the Galois Theory of Linear Differential Equations," *Algebraic theory of differential equations*, 357, 1–82.

Sipser, M. (2012): *Introduction to the Theory of Computation*, Cengage Learning.

Sirignano, J. and K. Spiliopoulos (2018): "DGM: A Deep Learning Algorithm for Solving Partial Differential Equations," *Journal of Computational Physics*, 375, 1339–1364.

Somacal, A. (2024): "Model Reduction for Forward Simulation and Inverse Problems : Towards Non-Linear Approaches," Ph.D. thesis, Sorbonne Université.

Stocker, T. (2011): *Introduction to Climate Modelling*, Springer Science & Business Media.

Strikwerda, J. C. (2004): *Finite Difference Schemes and Partial Differential Equations, Second Edition*, Society for Industrial and Applied Mathematics.

Sun, L., H. Gao, S. Pan, and J.-X. Wang (2020): "Surrogate Modeling for Fluid Flows Based on Physics-Constrained Deep Learning without Simulation Data," *Computer Methods in Applied Mechanics and Engineering*, 361, 112732.

Sun, S., J. Zhao, and J. Zhu (2015): "A Review of Nyström Methods for Large-Scale Machine Learning," *Information Fusion*, 26, 36–48.

Telgarsky, M. (2016): "Benefits of Depth in Neural Networks," in *Conference on Learning Theory*, PMLR, 1517–1539.

Treves, F. (2016): *Topological Vector Spaces, Distributions and Kernels: Pure and Applied Mathematics, Vol. 25*, vol. 25, Elsevier.

Urbán, J. F., P. Stefanou, and J. A. Pons (2025): "Unveiling the Optimization Process of Physics Informed Neural Networks: How Accurate and Competitive Can PINNs Be?" *Journal of Computational Physics*, 523, 113656.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017): "Attention Is All You Need," *arXiv:1706.03762 [cs]*.

Verbockhaven, M. (2025): "Spotting Expressivity Bottlenecks in Neural Networks and Fixing Them by Optimal Architecture Growth," Ph.D. thesis, Université Paris-Saclay.

Verbockhaven, M., S. Chevallier, and G. Charpiat (2024): "Growing Tiny Networks: Spotting Expressivity Bottlenecks and Fixing Them Optimally," *arXiv preprint arXiv:2405.19816*.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors (2020): "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, 17, 261–272.

Voulodimos, A., N. Doulamis, A. Doulamis, and E. Protopapadakis (2018): "Deep Learning for Computer Vision: A Brief Review," *Computational intelligence and neuroscience*, 2018.

Wang, Q., Y. Ma, K. Zhao, and Y. Tian (2022a): "A Comprehensive Survey of Loss Functions in Machine Learning," *Annals of Data Science*, 9, 187–212.

Wang, S., A. K. Bhartari, B. Li, and P. Perdikaris (2025a): "Gradient Alignment in Physics-informed Neural Networks: A Second-Order Optimization Perspective," .

Wang, S., B. Li, Y. Chen, and P. Perdikaris (2024): "PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks," *Journal of Machine Learning Research*, 25, 1–51.

Wang, S. and P. Perdikaris (2021): "Deep Learning of Free Boundary and Stefan Problems," *Journal of Computational Physics*, 428, 109914.

Wang, S., S. Sankaran, and P. Perdikaris (2022b): "Respecting Causality Is All You Need for Training Physics-Informed Neural Networks," .

Wang, S., S. Sankaran, H. Wang, and P. Perdikaris (2023): "An Expert's Guide to Training Physics-informed Neural Networks," .

Wang, S., Y. Teng, and P. Perdikaris (2021a): "Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks," *SIAM Journal on Scientific Computing*, 43, A3055–A3081.

Wang, S., H. Wang, and P. Perdikaris (2021b): "On the Eigenvector Bias of Fourier Feature Networks: From Regression to Solving Multi-Scale PDEs with Physics-Informed Neural Networks," *Computer Methods in Applied Mechanics and Engineering*, 384, 113938.

Wang, Y. and C.-Y. Lai (2024): "Multi-Stage Neural Networks: Function Approximator of Machine Precision," *Journal of Computational Physics*, 504, 112865.

Wang, Y., J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, and Y. Liu (2025b): "Kolmogorov–Arnold-Informed Neural Network: A Physics-Informed Deep Learning Framework for Solving Forward and Inverse Problems Based on Kolmogorov–Arnold Networks," *Computer Methods in Applied Mechanics and Engineering*, 433, 117518.

Wendt, J. F., ed. (2009): *Computational Fluid Dynamics*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Werbos, P. J. (1982): "Applications of Advances in Nonlinear Sensitivity Analysis," in *System Modeling and Optimization*, ed. by R. F. Drenick and F. Kozin, Berlin/Heidelberg: Springer-Verlag, vol. 38, 762–770.

Williams, C. (1996): "Computing with Infinite Networks," *Advances in neural information processing systems*, 9.

Wu, Z., S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu (2020): "A Comprehensive Survey on Graph Neural Networks," *IEEE transactions on neural networks and learning systems*, 32, 4–24.

Xu, X., T. Du, W. Kong, Y. Li, and Z. Huang (2024): "Convergence Analysis of Natural Gradient Descent for Over-parameterized Physics-Informed Neural Networks," .

Yang, G. (2019): "Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation," *arXiv preprint arXiv:1902.04760*.

Yang, L., X. Meng, and G. E. Karniadakis (2021): "B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data," *Journal of Computational Physics*, 425, 109913.

Zhao, Z., X. Ding, and B. A. Prakash (2024): "PINNsFormer: A Transformer-Based Framework For Physics-Informed Neural Networks," .

Zhu, Y., N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris (2019): "Physics-Constrained Deep Learning for High-Dimensional Surrogate Modeling and Uncertainty Quantification without Labeled Data," *Journal of Computational Physics*, 394, 56–81.

Zienkiewicz, O. C. and R. L. Taylor (2013): *The Finite Element Method: Its Basis and Fundamentals*, Butterworth-Heinemann.

A - Complements to ANaGRAM (Chapter 4)

A.1 . Complementary material for the Experiments

A.1.1 . Description of the experimental setting

Description of the Method We base our code on Müller and Zeinhofer (2023). For our 4 numerical experiments, we apply ANaGRAM gradient steps as defined in the Algorithm 3. As in Müller and Zeinhofer (2023), we choose the interval $[0, 1]$ for the line search determining the learning rate, 1 corresponding to solving the (linearized PDE) with the Green's function (*cf.* Section 4.4). The neural network weights are initialized using the Glorot normal initialization (Glorot and Bengio, 2010).

Computation Details As in Müller and Zeinhofer (2023), our implementation relies on JAX (Bradbury et al., 2018), where all derivatives are computed using JAX automatic differentiation, and the singular value decomposition computation is carried out by the scipy (Virtanen et al., 2020) implementation of JAX. Stochastic gradient descent, Adam, as well as L-BFGS relies on the implementation of DeepMind et al. (2020). All experiments were run on a 11th Gen Intel® Core™ i7-1185G7 @ 3.00GHz Laptop CPU in double precision (float64).

A.1.2 . Figures relative to computation time

2 D Laplace

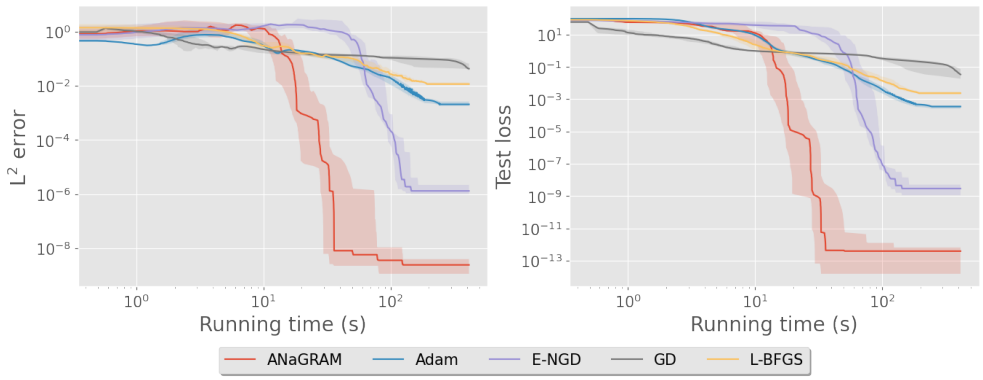


Figure A.1: Median absolute L^2 errors and Test losses for the 2 D Laplace equation across 10 different initializations for the five optimizers, relative to computation time. The shaded area indicates the range between the first and third quartiles.

Heat

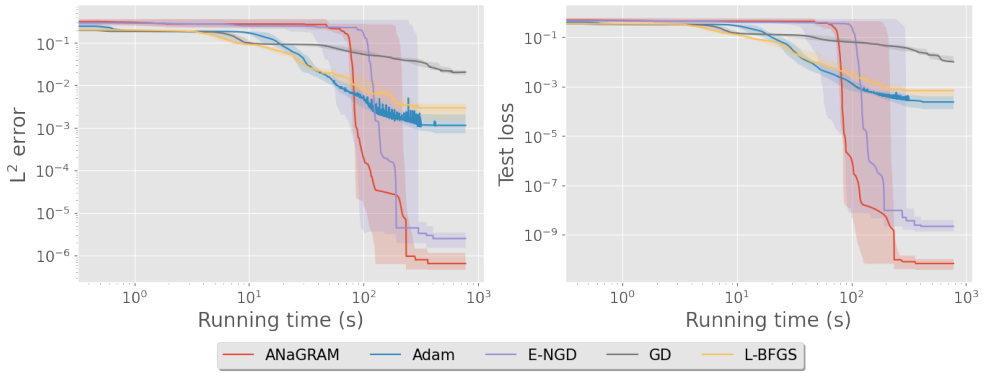


Figure A.2: Median absolute L^2 errors and Test losses for the Heat equation across 10 different initializations for the five optimizers, relative to computation time. The shaded area indicates the range between the first and third quartiles.

5 D Laplace

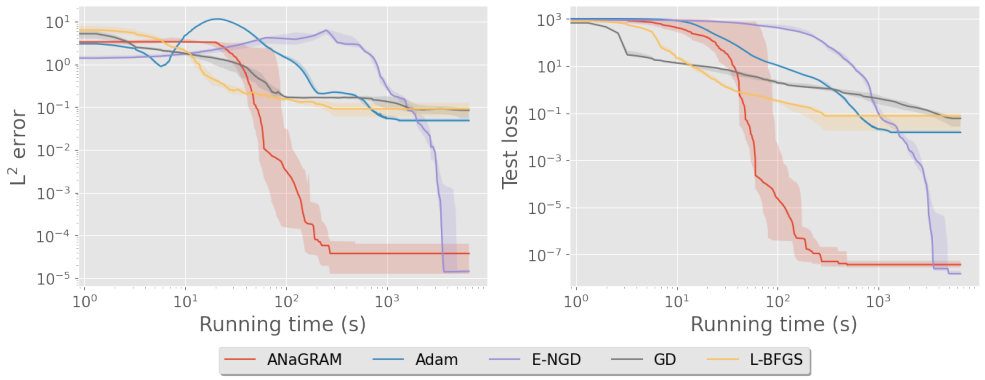


Figure A.3: Median absolute L^2 errors and Test losses for the 5 D Laplace equation across 10 different initializations for the five optimizers, relative to computation time (except for ENGD for which we only took 3 initializations). The shaded area indicates the range between the first and third quartiles.

Allen-Cahn

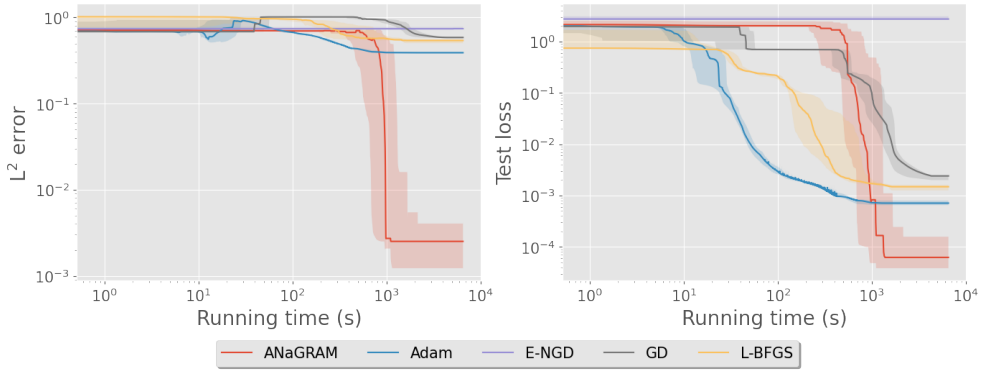


Figure A.4: Median absolute L^2 errors and Test losses for the Allen-Cahn equation across 10 different initializations for the five optimizers, relative to computation time (except for ENGD for which we only took 3 initializations). The shaded area indicates the range between the first and third quartiles.

A.1.3 . Statistical tables of results

2 D Laplace

Table A.1: Median, Maximum and Minimum L^2-errors of the optimizers for the 2 D Laplace equation.

	Median	Minimum	Maximum
ANaGRAM	**2.42e-09**	**1.70e-10**	**1.19e-08**
Adam	2.05e-03	1.67e-03	2.86e-03
E-NGD	1.31e-06	8.43e-07	3.87e-05
GD	4.25e-02	1.01e-02	1.25e-01
L-BFGS	1.15e-02	3.08e-03	1.55e-02

Table A.2: Median, Maximum and Minimum of the test loss of the optimizers for the 2 D Laplace equation.

	Median	Minimum	Maximum
ANaGRAM	**3.85e-13**	**8.49e-15**	**1.43e-12**
Adam	3.51e-04	2.29e-04	4.31e-04
E-NGD	2.91e-09	1.01e-10	3.57e-08
GD	3.42e-02	4.32e-03	1.76e-01
L-BFGS	2.37e-03	8.91e-04	9.09e-03

Table A.3: Mean and Standard deviation of L^2-errors of the optimizers for the 2 D Laplace equation.

	mean	std
ANaGRAM	**3.49e-09**	**3.58e-09**
Adam	2.19e-03	4.18e-04
E-NGD	5.37e-06	1.18e-05
GD	5.41e-02	1.57e-02
L-BFGS	1.13e-02	2.94e-03

Table A.4: Mean and Standard deviation of of the test loss of the optimizers for the 2 D Laplace equation.

	mean	std
ANaGRAM	**4.27e-13**	**4.66e-13**
Adam	3.37e-04	7.66e-05
E-NGD	7.00e-09	1.11e-08
GD	5.39e-02	5.39e-02
L-BFGS	3.04e-03	2.23e-03

Heat

Table A.5: Median, Maximum and Minimum L^2-errors of the optimizers for the Heat equation.

	Median	Minimum	Maximum
ANaGRAM	**6.48e-07**	**3.67e-07**	**6.15e-06**
Adam	1.07e-03	5.96e-04	3.94e-03
E-NGD	2.50e-06	1.02e-06	6.38e-06
GD	2.02e-02	1.04e-02	2.39e-02
L-BFGS	2.97e-03	5.14e-04	6.73e-03

Table A.6: Median, Maximum and Minimum test loss of the optimizers for the Heat equation.

	Median	Minimum	Maximum
ANaGRAM	**6.82e-11**	**1.90e-11**	**2.50e-10**
Adam	2.37e-04	1.11e-04	1.41e-03
E-NGD	2.18e-09	5.62e-10	1.35e-08
GD	1.01e-02	4.70e-03	1.93e-02
L-BFGS	6.84e-04	5.85e-05	3.34e-03

Table A.7: Mean and Standard deviation of L^2-errors of the optimizers for the Heat equation.

	mean	std
ANaGRAM	**1.28e-06**	**1.75e-06**
Adam	1.55e-03	5.19e-04
E-NGD	2.89e-06	1.77e-06
GD	1.92e-02	9.60e-04
L-BFGS	3.09e-03	1.74e-03

Table A.8: Mean and Standard deviation of test loss of the optimizers for the Heat equation.

	mean	std
ANaGRAM	**8.56e-11**	**7.05e-11**
Adam	3.63e-04	3.93e-04
E-NGD	3.53e-09	3.83e-09
GD	1.20e-02	1.05e-03
L-BFGS	8.54e-04	9.16e-04

5 D Laplace

Table A.9: Median, Maximum and Minimum L^2-errors of the optimizers for the 5 D Laplace equation.

	Median	Minimum	Maximum
ANaGRAM	3.76e-05	**6.99e-06**	8.23e-05
Adam	4.86e-02	3.41e-02	6.08e-02
E-NGD	**1.40e-05**	1.18e-05	**1.64e-05**
GD	8.44e-02	1.50e-02	1.28e-01
L-BFGS	9.08e-02	1.55e-02	1.71e-01

Table A.10: Median, Maximum and Minimum test loss of the optimizers for the 5 D Laplace equation.

	Median	Minimum	Maximum
ANaGRAM	3.68e-08	**1.03e-08**	2.20e-07
Adam	1.53e-02	1.02e-02	2.54e-02
E-NGD	**1.51e-08**	1.50e-08	**2.51e-08**
GD	6.00e-02	6.37e-03	1.11e-01
L-BFGS	7.65e-02	5.34e-03	2.25e-01

Table A.11: Mean and Standard deviation of L^2-errors of the optimizers for the 5 D Laplace equation.

	mean	std
ANaGRAM	4.00e-05	2.93e-05
Adam	4.83e-02	8.06e-03
E-NGD	**1.41e-05**	**2.29e-06**
GD	7.64e-02	1.75e-02
L-BFGS	1.00e-01	2.19e-02

Table A.12: Mean and Standard deviation of test loss of the optimizers for the 5 D Laplace equation.

	mean	std
ANaGRAM	6.37e-08	7.01e-08
Adam	1.63e-02	4.19e-03
E-NGD	**1.84e-08**	**5.55e-09**
GD	5.64e-02	3.60e-02
L-BFGS	8.20e-02	6.80e-02

Allen-Cahn

Table A.13: Median, Maximum and Minimum L^2-errors of the optimizers for the Allen-Cahn equation.

	Median	Minimum	Maximum
ANaGRAM	**2.51e-03**	**6.14e-04**	**2.04e-02**
Adam	3.90e-01	3.78e-01	4.80e-01
E-NGD	7.39e-01	7.32e-01	8.10e-01
GD	5.86e-01	5.43e-01	8.37e-01
L-BFGS	5.40e-01	4.33e-01	7.45e-01

Table A.14: Median, Maximum and Minimum test loss of the optimizers for the Allen-Cahn equation.

	Median	Minimum	Maximum
ANaGRAM	**6.22e-05**	**1.45e-05**	1.38e-03
Adam	7.10e-04	6.29e-04	**1.18e-03**
E-NGD	2.74e+00	2.58e+00	3.43e+00
GD	2.41e-03	1.70e-03	1.93e-02
L-BFGS	1.48e-03	9.08e-04	5.09e-03

Table A.15: Mean and Standard deviation of L^2-errors of the optimizers for the Allen-Cahn equation.

	mean	std
ANaGRAM	**4.32e-03**	5.93e-03
Adam	4.02e-01	**5.19e-04**
E-NGD	7.60e-01	4.24e-02
GD	6.06e-01	5.99e-02
L-BFGS	5.49e-01	6.85e-02

Table A.16: Mean and Standard deviation of test loss of the optimizers for the Allen-Cahn equation.

	mean	std
ANaGRAM	**2.19e-04**	4.16e-04
Adam	7.81e-04	**1.01e-04**
E-NGD	2.92e+00	4.51e-01
GD	3.95e-03	5.41e-03
L-BFGS	1.77e-03	1.19e-03

A.1.4 . Additional experiment : Burgers equation

We consider the $(1+1)$ dimensional Burger equation:

$$\begin{cases} \partial_t u + u\,\partial_x u - \nu \partial_{xx} u = 0 & \text{in } \Omega = [0,1] \times [-1,1] \\ u = 0 & \text{on } \partial\Omega_{\text{border}} = [0,1] \times \{-1,1\} \\ u(0,x) = -\sin(\pi x) & \text{on } \partial\Omega_0 = \{0\} \times [-1,1] \end{cases} \tag{A.1}$$

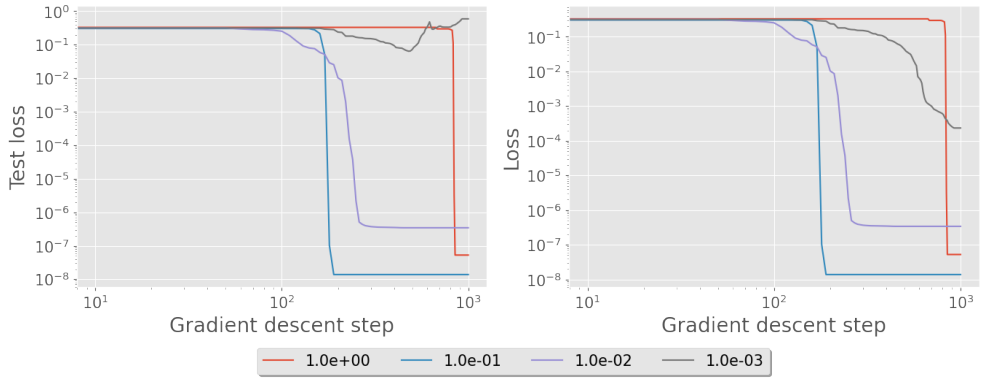


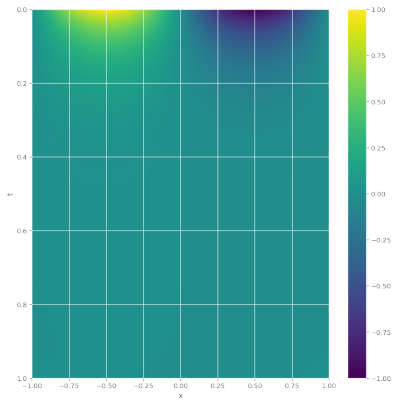
Figure A.5: Test and train losses for the Burgers equation for various viscosities ν.

We considered the burger with 4 viscosities $\nu \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$. Since a shock develops in finite time as $\nu \to 0$, using a regular grid did not yield sufficiently accurate results. Consequently, we manually adjusted the grid. Specifically, we defined two discretizations D_t, D_x, corresponding to $[0,1]$ and $[-1,1]$, respectively, as follows:

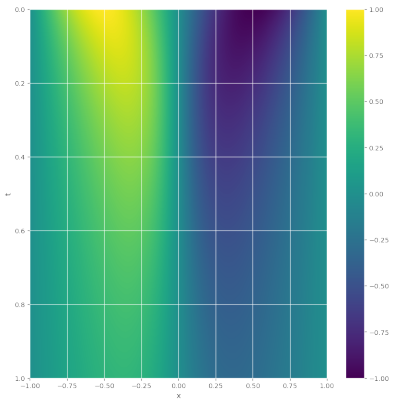
- D_t: A regular grid discretization with 240 points.
- D_x: A non-regular grid discretization constructed as:
  - A regular-grid of $60 = \frac{240}{4}$ points on $[-1, -0.25)$,
  - A grid defined as $-2^{-D_{\log}}$ on $[-0.25, 0)$, where $D_{\log}$ is a regular grid of $60 = \frac{240}{4}$ point on $[2, 32]$,
  - The point 0,
  - A grid defined as $2^{-D_{\log}}$ on $(0, 0.25]$, where $D_{\log}$ is a regular grid of $60 = \frac{240}{4}$ point on $[2, 32]$,
  - A regular grid of $60 = \frac{240}{4}$ points on $(0.25, 1]$.

With these definitions, the following grids were constructed:

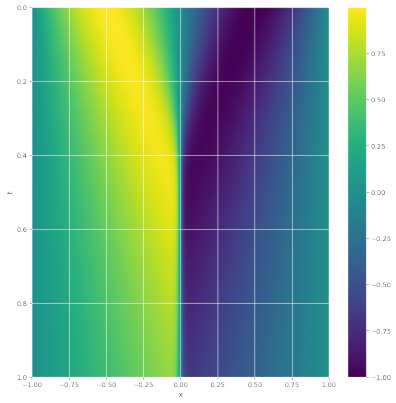
- $G_\Omega = D_t \times D_x$: The grid for the domain $\Omega = [0,1] \times [-1,1]$,
- $G_{\partial\Omega_{\text{border}}} = D_t \times \{-1, 1\}$: The grid for the boundary $\partial\Omega_{\text{border}} = [0,1] \times \{-1,1\}$,



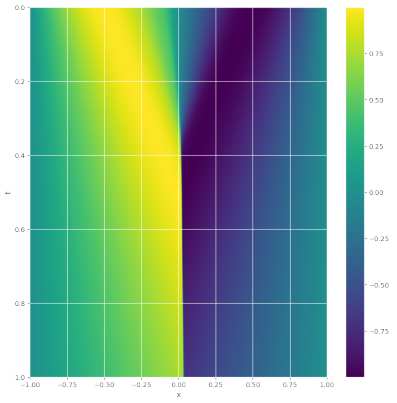
(a) $\nu = 1$



(b) $\nu = 10^{-1}$



(c) $\nu = 10^{-2}$



(d) $\nu = 10^{-3}$

Figure A.6: PINN solution profiles for the Burgers equation for various viscosities ν

- $G_{\partial\Omega_0} = \{0\} \times D_x$: The grid for the boundary $\partial\Omega_0 = \{0\} \times [-1, 1]$.

We applied ANaGRAM for 1000 iterations, using a neural network with three hidden layers, each containing 32 neurons, resulting in a total of $P = 2241$ parameters. The cutoff factor is set to $\epsilon = 5.10^{-7} \times \Delta_{\theta\max}$, where $\Delta_{\theta\max}$ represents the largest eigenvalue of $\widehat{\phi}_\theta$ (see line 1 of Algorithm 3). In Figure A.5, we present the training and test losses for the different viscosities. The test loss was calculated using a grid constructed similarly to the training grid but with five times as many points.

For a viscosity of $\nu = 10^{-3}$, the test loss does not appear to converge, whereas the train loss does. This behavior can be attributed to the grid computation method, which disproportionately emphasizes points near the shock. This explanation is supported by the solution profile shown in Figure A.6, where, for $\nu = 10^{-3}$, the shock in the learned solution is slightly shifted from $x = 0$ toward $x > 0$.

A.1.5 . A practical example : Derivation of generalized Green's function for Laplacian operator, based on PINN's natural gradient formulation on a Fourier's basis

We will illustrate Theorem 4.8 on a parametric model given by partial Fourier's series (*cf.* Example 2.12) for the Laplace operator on $[0,1]^d$:

$$\Delta : \left\{ \begin{array}{ccc} \mathrm{H}^2([0,1]^d \to \mathbb{C}) & \to & \mathrm{L}^2([0,1]^d \to \mathbb{C}) \\ v & \mapsto & \sum_{l=1}^d \partial_{ll} v \end{array} \right. \tag{A.2}$$

For this purpose, let us then fix $N \in \mathbb{N}$ and consider the associated partial Fourier's Serie S_N as defined in Equation (2.8):

$$S_N : \left\{ \begin{array}{ccc} \mathbb{R}^{[\![-N,N]\!]^d} & \to & \mathrm{L}^2([0,1]^d \to \mathbb{C}) \\ (\alpha_{k_1,\ldots,k_d}) & \mapsto & \left(\begin{array}{c} x \\ \cap \\ [0,1]^d \end{array} \mapsto \sum_{k_1=-N}^{N} \cdots \sum_{k_d=-N}^{N} \alpha_{k_1,\ldots,k_d} e^{2i\pi \left(\sum_{l=1}^d k_l x_l\right)} \right) \end{array} \right. \tag{2.8}$$

We will then derive according to Theorem 4.8 the generalized Green's function of Δ on the tangent space of S_N defined in Equation (2.10), namely:

$$\mathcal{M} = T_\theta \mathcal{M} = \mathrm{Span}\left(x \in [0,1]^d \mapsto e^{2i\pi \left(\sum_{l=1}^d k_l x_l\right)} \ : \ k \in [\![-N,N]\!]^d \right) \tag{2.10}$$

To this end, let define: for all $k \in [\![-N,N]\!]^d$

$$e_k : \left\{ \begin{array}{ccc} [0,1]^d & \to & \mathbb{C} \\ x & \mapsto & e^{2i\pi \left(\sum_{l=1}^d k_l x_l\right)} \end{array} \right. \tag{A.3}$$

and compute: for all $k \in [\![-N,N]\!]^d$, for all $1 \leqslant m \leqslant d$

$$\frac{\partial^2}{\partial x_m^2} \ e_k = -(2\pi)^2 k_m^2 e_k, \tag{A.4}$$

then: for all $k \in [\![-N,N]\!]^d$

$$\Delta \ e_k = -(2\pi)^2 \left(\sum_{m=1}^d k_m^2 \right) e_k, \tag{A.5}$$

and thus: for all $k_1, k_2 \in [\![-N,N]\!]^d$

$$G_{k_1,k_2} := \langle \Delta \ e_{k_1} \ , \ \Delta \ e_{k_2} \rangle = (2\pi)^4 \left(\sum_{m=1}^d {k_1}_m^2 \right)^2 \delta_{k_1,k_2}, \tag{A.6}$$

where δ_{k_1,k_2} is the Kronecker symbol such that $\delta_{k_1,k_2} = 1$ if and only if $k_1 = k_2$. This implies that:

$$G^\dagger_{k_1,k_2} = (2\pi)^{-4} \left(\sum_{m=1}^d {k_1}_m^2 \right)^{-2} (1 - \delta_{k_1,0}) \, \delta_{k_1,k_2}, \tag{A.7}$$

yielding:

$$
\begin{aligned}
g_{T_\theta\mathcal{M}}(x,y) &= \sum_{k_1,k_2\in[\![-N,N]\!]^d} e_{k_1}(x) G^{\dagger}_{k_1,k_2} \overline{\Delta\, e_{k_2}}(y) \\
&= \sum_{k_1,k_2\in[\![-N,N]\!]^d} e_{k_1}(x)(2\pi)^{-4} \left(\sum_{m=1}^{d} k_{1_m}^2\right)^{-2} (1-\delta_{k_1,0})\,\delta_{k_1,k_2} \\
&\qquad\qquad \times \left(-(2\pi)^2 \left(\sum_{m=1}^{d} k_{2_m}^2\right) \overline{e_{k_2}}(y)\right) \\
&= \frac{-1}{(2\pi)^2} \sum_{k\in[\![-N,N]\!]^d\setminus\{0\}} \frac{e_k(x)\,\overline{e_k}(y)}{\sum_{m=1}^{d} k_m^2} \\
&= \frac{-1}{(2\pi)^2} \sum_{k\in[\![-N,N]\!]^d\setminus\{0\}} \frac{\exp\left(2i\pi\left(\sum_{l=1}^{d} k_l(x_l-y_l)\right)\right)}{\sum_{l=1}^{d} k_l^2}
\end{aligned}
\tag{A.8}
$$

Finally, in order to add some consistency to our illustration, let us show for $d=1$ that in the limit $N\to\infty$, we indeed find the classical Green's function of the Laplacian operator. In this case let us first remark, that we have:

$$
\begin{aligned}
g_{T_\theta\mathcal{M}}(x,y) &= \frac{-1}{(2\pi)^2} \sum_{k\in[\![-N,N]\!]\setminus\{0\}} \frac{\exp\left(2i\pi k(x-y)\right)}{k^2} \\
&= \frac{-1}{2\pi^2} \sum_{k=1}^{N} \frac{\cos(2\pi k(x-y))}{k^2}.
\end{aligned}
\tag{A.9}
$$

Since $|\cos(2\pi k(x-y))| \leqslant 1$, the serie is absolutly convergent and thus in the limit $N\to\infty$, we have:

$$
g_\infty(x,y) := \lim_{N\to\infty} g_{T_\theta\mathcal{M}}(x,y) = \frac{-1}{2\pi^2} C_2(2\pi(x-y)), \tag{A.10}
$$

with

$$
C_2(z) = \sum_{k=1}^{+\infty} \frac{\cos(kz)}{k^2}. \tag{A.11}
$$

From Abramowitz and Stegun (1968, page 1005), we know that for all $z\in(0,2\pi)$:

$$
C_2(z) = \frac{\pi^2}{6} - \frac{\pi z}{2} + \frac{z^2}{4}, \tag{A.12}
$$

Thus, thanks to the parity of C_2 we have for all $(x-y)\in(0,1)$:

$$
\begin{aligned}
g_\infty(x,y) &= \frac{-1}{2\pi^2}\left(\frac{\pi^2}{6} - \frac{\pi 2\pi|x-y|}{2} + \frac{(2\pi(x-y))^2}{4}\right) \\
&= -\frac{1}{12} + \frac{|x-y|}{2} - \frac{(x-y)^2}{2}.
\end{aligned}
\tag{A.13}
$$

which is indeed a Green function for the Laplacian in 1d.

A.2 . Mathematical equivalence of ANaGRAM and Energy Natural Gradient implementation (Müller and Zeinhofer, 2023) for linear operators

Let us begin by briefly recalling the definition of Energy Natural Gradient, as introduced in Müller and Zeinhofer (2023). Given an Energy (Equation 2 in Müller and Zeinhofer (2023)):

$$E(u) = \int_\Omega (\mathcal{D}[u] - f)^2 \mathrm{d}x + \tau \int_{\partial\Omega} (B[u] - g)^2 \mathrm{d}s, \tag{A.14}$$

associated to operators $D : \mathcal{H} \to \mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$, $B : \mathcal{H} \to \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma)$, and a parametric model $u : \mathbb{R}^P \to \mathcal{H}$, we have the following definition (Definition 1 in Müller and Zeinhofer (2023)):

Definition A.1 (Energy Natural Gradient)**.** Consider the problem

$$\min_{\theta \in \mathbb{R}^P} L(\theta), \tag{A.15}$$

where

$$L(\theta) := E(u_{|\theta}). \tag{A.16}$$

Denote the Euclidean gradient by $\nabla L(\theta)$. Then we call

$$\nabla^E L(\theta) := G_E^\dagger(\theta) \nabla L(\theta), \tag{A.17}$$

the *energy natural gradient (E-NG)*, where G_E is a Gram matrix defined by: for all $1 \leqslant p, q \leqslant P$

$$G_E(\theta)_{pq} := D^2 E(u_\theta)(\partial_{\theta_p} u_\theta, \partial_{\theta_q} u_\theta), \tag{A.18}$$

with $D^2 E$ being the second derivative of E in the Fréchet sense.

As noted in Equation (9) of Müller and Zeinhofer (2023), in the case where D and B are linear, Equation (A.18) reduces to:for all $1 \leqslant p, q \leqslant P$

$$G_E(\theta)_{pq} = \int_\Omega D[\partial_{\theta_p} u_\theta](x) D[\partial_{\theta_q} u_\theta](x) \mathrm{d}x + \tau \int_{\partial\Omega} B[\partial_{\theta_p} u_\theta](s) B[\partial_{\theta_q} u_\theta](s) \mathrm{d}s, \tag{A.19}$$

Note that in this case, by setting $\tau = 1$ and taking μ and σ uniform, this corresponds exactly to the Gram matrix:for all $1 \leqslant p, q \leqslant P$

$$G_{\theta pq} := \left\langle \partial_p \big((D, B) \circ u\big)_{|\theta}, \partial_q \big((D, B) \circ u\big)_{|\theta} \right\rangle_{\mathrm{L}^2(\Omega, \partial\Omega)}, \tag{A.20}$$

where the compound model $(D, B) \circ u$ and the space $\mathrm{L}^2(\Omega, \partial\Omega)$ are those introduced in Equation (4.50). Now the key element lies in the sentence quoted from page 6, section 4.1 of Müller and Zeinhofer (2023) (which is confirmed in practice by the code):

> "The integrals in (15) are computed using the same collocation points as in the definition of the PINN loss function L in (14)."

This means that the same set of collocation points is used to evaluate the integrals defining $L(\theta)$ in Equation (A.16) and the integrals defining $G_E(\theta)$ in Equation (A.19).

Let us fix some collocations points $(x_i^D)_{i=1}^{S_D}$ in Ω, and $(x_i^B)_{i=1}^{S_B}$ in $\partial\Omega$. Then Equation (A.16) discretization exactly corresponds up to factor $\frac{1}{2}$, to the scalar loss of PINNs defined in Equation (4.74).

.

The Euclidean gradient $\nabla L(\theta)$ is then approximated by the Euclidean gradient $\nabla \ell(\theta)$ given by: for all $1 \leqslant p \leqslant P$

$$\begin{aligned}(\nabla \ell(\theta))_p =& \frac{1}{S_D} \sum_{i=1}^{S_D} \partial_p D[u_{|\theta}](x_i^D) \left(D[u_{|\theta}](x_i^D) - f(x_i^D) \right) \\ &+ \frac{1}{S_B} \sum_{i=1}^{S_B} \partial_p B[u_{|\theta}](x_i^B) \left(B[u_{|\theta}](x_i^B) - g(x_i^B) \right)\end{aligned} \tag{A.21}$$

Let us define:

- for all $1 \leqslant p \leqslant P$, for all $1 \leqslant i \leqslant S_D$, and for all $1 \leqslant j \leqslant S_B$:

$$\widehat{\phi}_{\theta\ p,i}^D := \partial_p D[u_{|\theta}](x_i^D); \qquad \widehat{\phi}_{\theta\ p,j}^B := \partial_p B[u_{|\theta}](x_j^B), \tag{A.22}$$

- for all $1 \leqslant i \leqslant S_D$ and for all $1 \leqslant j \leqslant S_B$:

$$\widehat{\nabla \mathcal{L}}_{\theta\ i}^D := D[u_{|\theta}](x_i^D) - f(x_i^D); \quad \widehat{\nabla \mathcal{L}}_{\theta\ j}^B := B[u_{|\theta}](x_j^B) - g(x_j^B), \tag{A.23}$$

and finally:

$$\widehat{\phi}_\theta := \left(\widehat{\phi}_\theta^D, \quad \widehat{\phi}_\theta^B \right); \quad \widehat{\nabla \mathcal{L}}_\theta := \begin{pmatrix} \widehat{\nabla \mathcal{L}}_\theta^D \\ \widehat{\nabla \mathcal{L}}_\theta^B \end{pmatrix}; \quad \widehat{\Lambda} := \begin{pmatrix} \frac{1}{S_D} \boldsymbol{I}_{S_D} & 0 \\ 0 & \frac{1}{S_B} \boldsymbol{I}_{S_B} \end{pmatrix}. \tag{A.24}$$

Thus Equation (A.21) can be rewritten as:

$$\nabla \ell(\theta) = \widehat{\phi}_\theta \widehat{\Lambda} \widehat{\nabla \mathcal{L}}_\theta. \tag{A.25}$$

But since the same set of collocation points is used to evaluate the integrals defining $G_E(\theta)$, we also have that Equation (A.18) is discretized by:

$$G_E(\theta) \simeq \widetilde{G_E(\theta)} := \widehat{\phi}_\theta \widehat{\Lambda} \widehat{\phi}_\theta^t. \tag{A.26}$$

This implies that Equation (A.17) is approximated by:

$$\nabla^E L(\theta) \simeq \widetilde{\nabla^E L(\theta)} := \widetilde{G_E(\theta)}^\dagger \nabla \ell(\theta) = \widehat{\phi}_\theta^\dagger \widehat{\nabla \mathcal{L}}_\theta, \tag{A.27}$$

which corresponds indeed to the update direction in line 5 in ANaGRAM algorithm 3.

To conclude, it should be noted that when D or B are not linear, then the equivalence between Equation (A.19) and Equation (A.20) not longer holds, with the result that E-NGD and ANaGRAM are no longer equivalent either.

B - Complements to AMStraMGRAM (Chapter 5)

B.1 . Our vocabulary

- **Domain (Ω).**
- **Boundary ($\partial\Omega$).**
- **Differential operators (D, B).**
- **Cutoff (α_t).** A threshold below which the components of the matrix $\widehat{\Delta}$ are truncated, *i.e.* $\widehat{\Delta} \leftarrow \begin{cases} \widehat{\Delta} & \text{if } \widehat{\Delta} \geqslant \alpha_t, \\ 0 & \text{else.} \end{cases}$
- **Full rank ($\mathbf{r_{svd}}$).** A full rank of feature matrix $\widehat{\phi}$ that we assume, without loss of generality, to be equal to $\min(P, S)$.
- **Rank ($\mathbf{r_{cutoff}}$).** A number of $\widehat{\Delta}$ components that are retained when computing a pseudo-inverse of $\widehat{\Delta}$ in ANaGRAM. Depending on a current regime of the training and a desired effect, it can be set at $\mathrm{r}_{\mathrm{int}}$ or r_{ϵ}.
- **Flattening.** The phenomenon described in Section 5.4.1, when reconstruction error starts to stabilize for a range of possible ranks.
- **Flat cutoff ($N_{\mathbf{flat}}$).** A number of components that corresponds to the beginning of flattening in reconstruction error curve.
- **Feature matrix ($\widehat{\phi} \in \mathbb{R}^{P \times S}$).** It is defined by a jacobian $\partial_p u_\theta(x_i)$, which is used in an ANaGRAM's update to "project" a functional gradient onto parameter space of θ.
- **Precision (ϵ).** A hyperparameter of AMStraMGRAM that prescribes a target error level that the algorithm should achieve.
- **Intersection rank ($\mathbf{r_{int}}$).** Defined in Equation (5.96), roughly speaking it corresponds to a number of components at which reconstruction error and singular values curves are intersecting.
- **Precision rank ($\mathbf{r_\epsilon}$).** Defined in Equation (5.97), it corresponds to a number of components at which reconstruction error curve and precision level are intersecting.
- **Functional gradient ($\nabla\mathcal{L}$).** A Frechet derivative of squared L^2 loss $\mathcal{L}$, its negative gives the "ideal" update direction in non-parametric case.

- **Empirical functional gradient ($\widehat{\nabla\mathcal{L}} \in \mathbb{R}^S$).** A vector obtain by evaluating $\nabla\mathcal{L}$ on some finite number of samples $x_i \in \Omega$, for $1 \leqslant i \leqslant S$.

- **Parametric model (u_θ).** A function parametrized with θ that serves to approximate a solution to a problem (regression or PDE). Typically, it is a neural network, where θ are its full set of weights.

- **Differential of the model ($\mathrm{d}\, u_\theta$).** Defined as

$$\mathrm{d}\, u_\theta(h) = \sum_{p=1}^{P} h_p \frac{\partial u}{\partial \theta_p} = \lim_{\varepsilon \to 0} \frac{u_{|\theta+\varepsilon h} - u_\theta}{\varepsilon}. \tag{B.1}$$

  It measures how much u_θ changes in a given direction h.

- **Tangent space ($T_\theta\mathcal{M}$).** Image of a differential of the model, giving a space of possible updates for a model u_θ.

- **SVD components of $\widehat{\phi}$ ($\widehat{U}$, $\widehat{\Delta}$, $\widehat{V}$).** In particular, $\widehat{\phi} = \widehat{U}\widehat{\Delta}\widehat{V}^T$, where $\widehat{U} \in \mathbb{R}^{P\times S}$ is a left singular vector matrix, $\widehat{\Delta} \in \mathbb{R}^{\mathrm{r_{svd}}\times \mathrm{r_{svd}}}$ is a diagonal matrix with singular values on a diagonal ordered in a decreasing order and $\widehat{V}$ is a left singular vector matrix.

- **Functional singular vectors ($V_{t,i}$).** Left singular vectors of the differential $\mathrm{d}u_\theta$.

- **Empirical tangent space ($T_N^M\mathcal{M}$).** A subspace of tangent space $T_\theta\mathcal{M}$, restricted to a span of the left functional singular vectors $V_{t,i}$ corresponding to a range of components from M to N, *i.e.* $\mathrm{Span}(V_{t,i} : 1 \leqslant M \leqslant N \leqslant N)$.

- **Discretized empirical tangent space ($\widehat{T_N^M\mathcal{M}}$).** A version of $T_N^M\mathcal{M}$ discretized on a set of samples $(x_i)_{i=1}^S$ coming from Ω.

- **Reconstruction error (RCE_N^S).** A measure identifying the portion of the functional gradient signal that is lost when restricting $\widehat{\nabla\mathcal{L}}$ to $\widehat{T_N^0\mathcal{M}}$.

- **Feature development phase.** The early phase in the training, during which high volatility is observed in both quantities of interest with high sensitivity to the choice of $\mathrm{r_{cutoff}}$.

- **Flattening phase.** The later phase in the training, during which reconstruction error starts to flatten for some values of N, at the same time singular values dominate over reconstruction error for all retained components, resulting in a drop of training loss.

B.2 . Practical Implementation Considerations

While the principled algorithm discussed in the main paper and summarized in Algorithm 4 provides a sound framework, empirical observations reveal that additional mechanisms are necessary for robust performance across diverse PDE problems. This section describes additional modifications to make the algorithm more practical.

B.2.1 . The Dual Cutoff Strategy: Addressing Empirical Challenges

Our experiments reveal that the single cutoff approach, while theoretically elegant, suffers from numerical instabilities and incomplete convergence in practice. We observed three critical issues:

1. **Ignition failure:** The intersection between reconstruction error and singular values sometimes fails to evolve, preventing the algorithm from reaching lower error values.
2. **Retreating dynamics:** The intersection rank may decrease during training, disrupting convergence.
3. **Incomplete flattening:** Without additional stabilization, the flattening phenomenon may not complete, leading to suboptimal final accuracy.

To address these challenges, we introduce a dual cutoff strategy inspired by the staged design of rocket launches:

B.2.2 . Three-Phase Training Dynamics

Ignition Phase

We initialize two cutoffs:

- **Minimum cutoff ($\mathbf{r}_{\min}$):** Set at the intersection point $\mathrm{r}_{\mathrm{int}}(t)$
- **Maximum cutoff ($\mathbf{r}_{\max}$):** Set at the "elbow" of the singular value curve (see algorithm 4)

The algorithm performs two natural gradient steps per iteration, one with each cutoff. If the intersection position remains static after both updates, we increment $\mathrm{r}_{\max}$ by one to promote exploration of additional gradient components.

This phase ends when $\mathrm{r}_{\min}$ reaches $\mathrm{r}_{\max}$—an event we term **liftoff**.

Ascent Phase

During ascent, both cutoffs track the moving intersection, but with a stability mechanism:

$$\mathrm{r}_{\max}(t) = \max(\mathrm{r}_{\max}(t-1), \mathrm{r}_{\mathrm{int}}(t)). \tag{B.2}$$

This monotonicity constraint prevents the intersection rank from falling to zero, which would disrupt training dynamics.

Stage Separation and Precision Locking

When $\mathrm{RCE}^{S}_{\mathrm{r_{int}}(t)}(t) \leqslant \epsilon$, we trigger **stage separation**:

- $\mathrm{r_{min}}$ is fixed at the precision level: $\mathrm{r_{min}} = \mathrm{r}_{\epsilon}(t)$
- $\mathrm{r_{max}}$ continues tracking the intersection to maintain stability

The algorithm continues until $\mathrm{r_{min}} = 0$ (**booster return**), indicating complete convergence. The final algorithm that combines all three stages is outlined in Algorithm 7.

B.2.3 . Complete Practical Algorithm

Algorithme 5 : Find elbow

Function `FindElbow`

  Input $(x_i) \in \mathbb{R}^m$ `// an increasing sequence of` $m \in \mathbb{N}$ `points in` $\mathbb{R}$

  - $\widehat{f} \in \mathbb{R}^m$ `// a decreasing function evaluated at points` (x_i)

  `/* Clockwise normal vector to` $\left(x_m - x_1, \widehat{f}_m - \widehat{f}_1\right)$ `*/`

  $\overrightarrow{n} \leftarrow \left(\widehat{f}_m - \widehat{f}_1, x_1 - x_m\right) \in \mathbb{R}^2$

  $(s_j)_{1 \leqslant j \leqslant m} \leftarrow \left(\left\langle \overrightarrow{n}\,,\, \left(x_j - x_1, \widehat{f}_j - \widehat{f}_1\right) \right\rangle_{\mathbb{R}^2}\right)_{1 \leqslant j \leqslant m}$

  Output : $\underset{1 \leqslant j \leqslant m}{\arg\max}\, s_j$

end

Algorithme 6 : Reconstruction Errors

Function `ReconstructionErrors`

  Input $\widehat{V}^t \in \mathbb{R}^{\mathrm{r_{svd}},S}$ `// left singular vectors of the Jacobian` $\widehat{\phi}$

  - $\widehat{\nabla\mathcal{L}} \in \mathbb{R}^S$ `// Evaluated functional gradient`

  begin Initialization

    $\widehat{\Sigma} \leftarrow 0 \in \mathbb{R}^S$ `// cumulative approximation of` $\widehat{\nabla\mathcal{L}}$

    $\mathrm{RCE}^S \leftarrow 0 \in \mathbb{R}^{\mathrm{r_{svd}}}$ `// cumulated reconstruction erros`

    $\widehat{c} \leftarrow \widehat{V}^t\widehat{\nabla\mathcal{L}} \in \mathbb{R}^{\mathrm{r_{svd}}}$

  end

  foreach $j \in (1, \ldots, r_{svd})$ **do**

    $\widehat{\Sigma} \leftarrow \widehat{\Sigma} + \widehat{c}_j$

    $\mathrm{RCE}_j^S \leftarrow \left\|\widehat{\Sigma} - \widehat{c}\right\|_2$

  end foreach

  Output : RCE^S

end

Algorithme 7 : AMStraMGRAM

Input :

- $u : \mathbb{R}^P \to \mathrm{L}^2(\Omega, \mu)$ `// neural network architecture`
- $\theta_0 \in \mathbb{R}^P$ `// initialization of the neural network`
- $f \in \mathrm{L}^2(\Omega, \mu)$ `// target function of the quadratic regression`
- $(x_i) \in \Omega^S$ `// a batch in` Ω
- $\epsilon > 0$ `// precision level of the optimization`

$\lambda \leftarrow False$ `// Liftoff indicator`
$\widehat{\phi}_{\theta_0} \leftarrow (\partial_p u_{\theta_0}(x_i))_{1 \leqslant i \leqslant S,\, 1 \leqslant p \leqslant P}$ `// via autodiff`
$\widehat{U}_{\theta_0}, \widehat{\Delta}_{\theta_0}, \widehat{V}^t_{\theta_0} \leftarrow \mathtt{SVD}\left(\widehat{\phi}_{\theta_0}\right)$
$\widehat{\nabla \mathcal{L}}_{\theta_0} \leftarrow (u_{\theta_0}(x_i) - f(x_i))_{1 \leqslant i \leqslant S}$
$\mathrm{RCE}_0^S \leftarrow \mathtt{ReconstructionErrors}\left(\widehat{V}^t_{\theta_0}, \widehat{\nabla \mathcal{L}}_{\theta_0}\right)$
$\mathrm{r}_{\max 0} \leftarrow \mathtt{FindElbow}\left((1, \ldots, \mathrm{r}_{\mathrm{svd}}), \widehat{\Delta}_{\theta_0}\right)$
repeat
  $\mathrm{r}_{1t} \leftarrow \#\left\{ \mathrm{RCE}^S_{0_j} \leqslant \widehat{\Delta}_{\theta_{t_j}} \,:\, 1 \leqslant j \leqslant \mathrm{r}_{\mathrm{svd}} \right\}$ `// with # being`
  $\mathrm{r}_{2t} \leftarrow \#\left\{ \mathrm{RCE}^S_{0_j} \geqslant \epsilon \,:\, 1 \leqslant j \leqslant \mathrm{r}_{\mathrm{svd}} \right\}$ `// the cardinal`
  $\mathrm{r}_{\min t} \leftarrow \min(\mathrm{r}_{1t}, \mathrm{r}_{2t})$
  $\mathrm{r}_{\max t} \leftarrow \max(\mathrm{r}_{1t}, \mathrm{r}_{\max t-1})$
  if $not\, \lambda_t$ **then**
    if $r_{\min t} \geqslant r_{\max t}$ **then**
      $\lambda_t \leftarrow \mathrm{True}$
    else if $r_{\min t-1} = r_{\min t}$ **then**
      $\mathrm{r}_{\max t} \leftarrow \mathrm{r}_{\max t} + 1$
  foreach $r_{cutoff} \in \{r_{\max t}, r_{\min t}\}$ **do**
    $\widehat{\Delta}_{\theta_t} \leftarrow \left(\widehat{\Delta}_{\theta_t, p} \text{ if } p \geqslant \mathrm{r}_{\mathrm{cutoff}} \text{ else } 0\right)_{1 \leqslant p \leqslant P}$
    $\widehat{\nabla \mathcal{L}}_{\theta_t} \leftarrow (u_{\theta_t}(x_i) - f(x_i))_{1 \leqslant i \leqslant S}$
    $d_{\theta_t} \leftarrow \widehat{V}_{\theta_t} \widehat{\Delta}^{\dagger}_{\theta_t} \widehat{U}^t_{\theta_t} \widehat{\nabla \mathcal{L}}_{\theta_t}$
    $\eta_t \leftarrow \arg\min_{\eta \in \mathbb{R}^+} \sum_{1 \leqslant i \leqslant S} \left(f(x_i) - u_{\theta_t - \eta d_{\theta_t}}(x_i) \right)^2$
    $\theta_{t+1} \leftarrow \theta_t - \eta_t\, d_{\theta_t}$
    $\widehat{\phi}_{\theta_{t+1}} \leftarrow \left(\partial_p u_{\theta_{t+1}}(x_i)\right)_{1 \leqslant i \leqslant S,\, 1 \leqslant p \leqslant P}$ `// via autodiff`
    $\widehat{U}_{\theta_{t+1}}, \widehat{\Delta}_{\theta_{t+1}}, \widehat{V}^t_{\theta_{t+1}} \leftarrow \mathtt{SVD}\left(\widehat{\phi}_{\theta_{t+1}}\right)$
  end
until $r_{1t} = 0$ *or* $t \geqslant T_{\max}$

B.2.4 . Empirical Justification for Design Choices

The dual cutoff strategy addresses specific empirical challenges we observed:

Dual gradient steps: Without the second cutoff, training dynamics sometimes stagnate. The dual approach provides both stability (via r_{min}) and exploration (via r_{max}).

Elbow initialization: The elbow point marks where singular values cease contributing meaningful signal, providing a natural upper bound for exploration.

Monotonic r_{max}: Prevents catastrophic retreat of the intersection point, which we observed in complex equations like Allen-Cahn.

Stage separation timing: Triggered precisely when the intersection error drops below target precision, ensuring optimal utilization of the flattening phenomenon.

B.3 . Deep dive on selected experiments

In this section we look at curves of training and estimations obtained with AMStraMGRAM on benchmark of PDEs.

B.3.1 . One Dimensional Burgers Equation

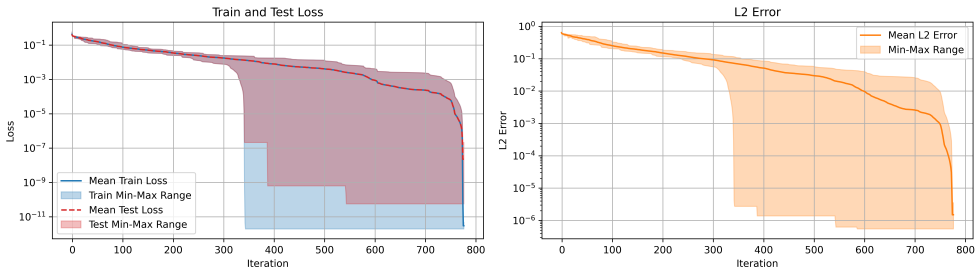
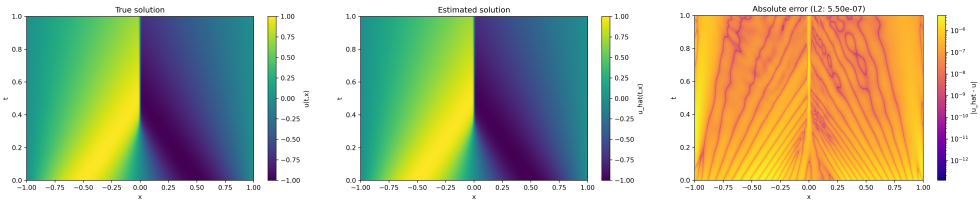


Figure B.1: Training metrics for the One-Dimensional Burgers equation, showing convergence behavior with our adaptive multi-cutoff strategy.



(a) True solution (b) Estimated solution (c) Error

Figure B.2: Results for One Dimensional Burgers Equation with cutoff 10^{-6}.

B.3.2 . Heat Equation

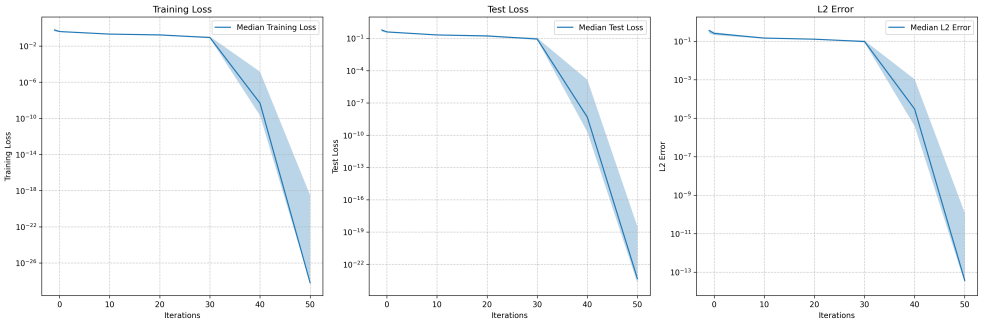
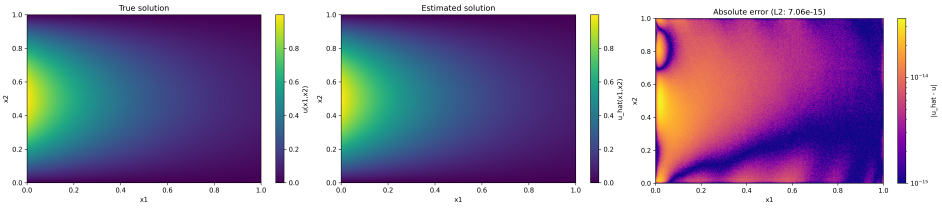


Figure B.3: Convergence results for the Heat equation showing the L_2 error over iterations. Our method (AMStraMGRAM) converges faster and reaches a lower final error than ANaGRAM and baselines. Variability across runs is due to differing feature development speed from the random initialization.



(a) True solution (b) Estimated solution (c) Error

Figure B.4: Results for the Heat equation (solution cutoff 10^{-14}). The error remains uniformly low over the domain, illustrating the effectiveness of the adaptive multi-cutoff strategy.

B.3.3 . Laplace Equations (L2D and L5D)

For the Laplace equation in 2D, our method also demonstrates remarkable performance improvements over the baselines. The convergence is both faster and reaches a significantly lower error plateau.

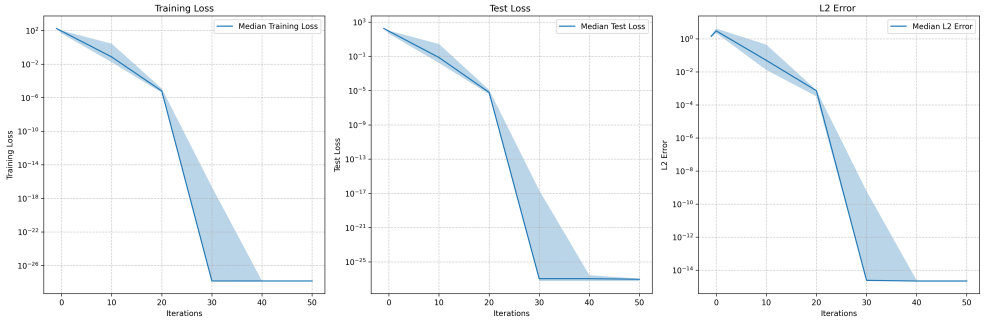
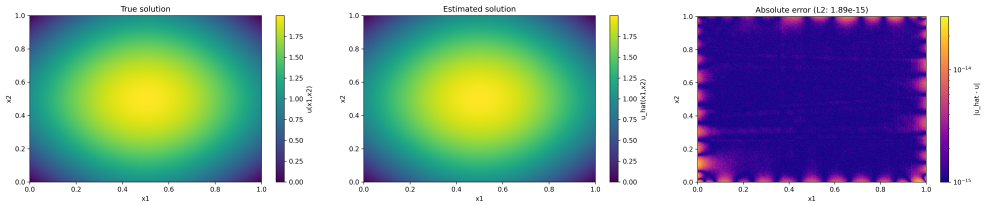


Figure B.5: Convergence results for the Laplace 2D problem, showing the L_2 error over iterations. Our method (AMStraMGRAM) achieves both faster convergence and lower final error compared to ANaGRAM and other baseline methods. The observed variance between runs can be explained by different speed of convergence depending on the initialization.



(a) True solution (b) Estimated solution (c) Error

Figure B.6: Results for Laplace 2D Equation with cutoff 10^{-6}.

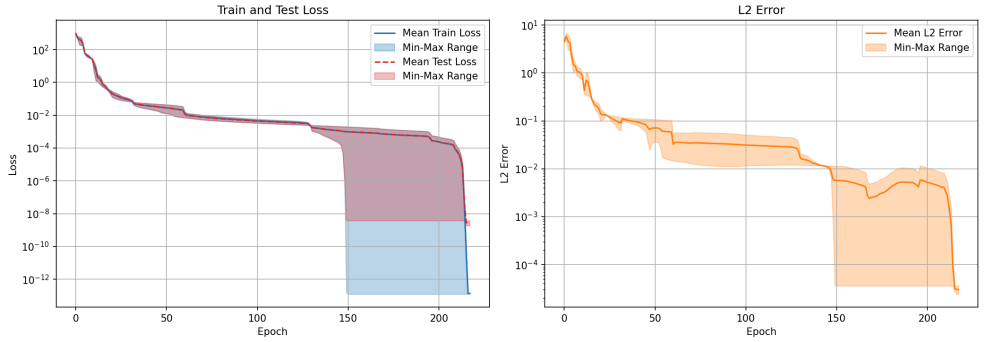


Figure B.7: Convergence results for the Laplace 5D problem, showing the L_2 error over iterations. Our method (AMStraMGRAM) achieves faster convergence but not lower final error compared to ANaGRAM and other baseline methods. We see that seeds change the speed of convergence of the algorithm

B.3.4 . Non Linear Poisson Equation

To compare ourselves with Urbán et al. (2025), we select ($K = 1$).

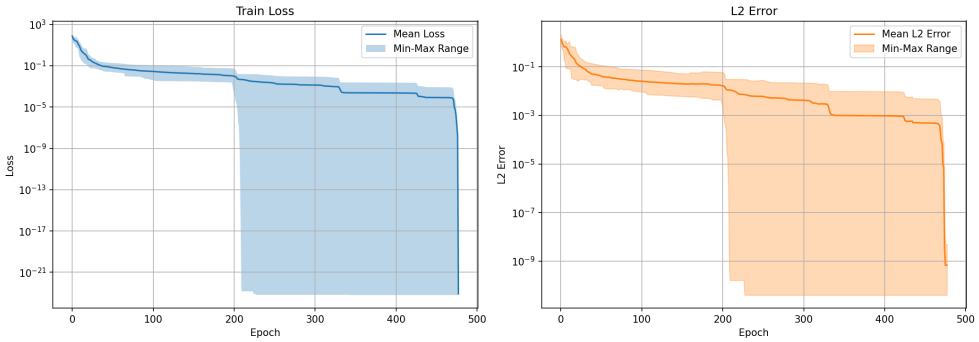
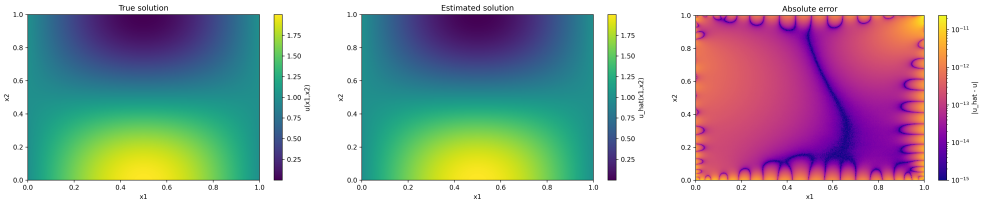


Figure B.8: Convergence results for the Non Linear Poisson equation, showing the L_2 error over iterations. Our method (AMStraMGRAM) achieves both faster convergence and lower final error compared to ANaGRAM and other baseline methods.



(a) True solution (b) Estimated solution (c) Error

Figure B.9: Results for the Nonlinear Poisson equation (cutoff 10^{-4}).

B.3.5 . Allen-Cahn Equation

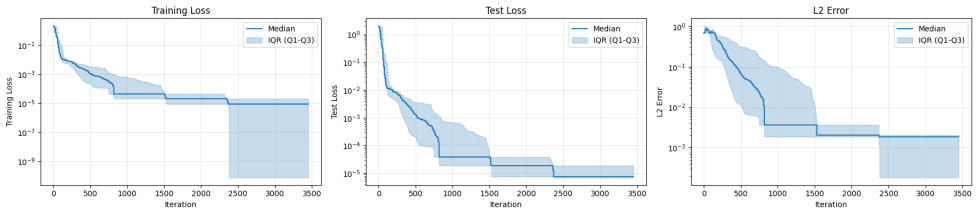
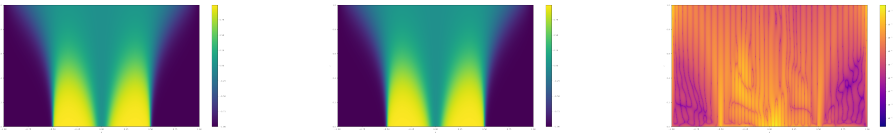


Figure B.10: Training curves for the Allen-Cahn equation, showing the evolution of loss and error over iterations.



(a) True solution (b) Estimated solution (c) Error

Figure B.11: Results on the Allen-Cahn equation, showing the error distribution (left), model prediction (middle), and true solution (right). The error is mostly present in regions with the "sharpest" transitions, which exemplifies the challenge of accurately capturing sharp interfaces still remains even for our advanced optimization approach.

C - Supplementary Mathematical Results

This chapter offers a brief overview of several complementary results that are used repeatedly throughout the thesis.

C.1 . Notations

We use the following notations.

- $\mathbb{N}_k$ denote the set of natural integers starting at integer k.

  In particular, $\mathbb{N}_0$ be the set of natural integers with 0 included and $\mathbb{N}_1$ the set of natural integers with 0 excluded.

- $\mathcal{F}(A \to B)$ denote the set of functions from set A to set B. When $A = B$ we simply denote $\mathcal{F}(A)$.

- $\mathcal{C}^0(A \to B)$ denote the set of continuous functions from A to B, when A and B are Banach spaces. When $A = B$ we simply denote $\mathcal{C}^0(A)$.

- $\mathcal{C}^k(A \to B)$ denote the set of k times continuously differentiable functions from A to B, when A and B are Banach spaces, with $k \in [1, +\infty) \cap \mathbb{N}_1$. When $A = B$ we simply denote $\mathcal{C}^k(A)$.

- $\mathcal{C}^\infty(A \to B)$ denote the set of indefinitely continuously differentiable functions from A to B, when A and B are Banach spaces. When $A = B$ we simply denote $\mathcal{C}^\infty(A)$.

- $\mathcal{C}_c^k(A \to B)$ denote the set of functions in $\mathcal{C}^k(A \to B)$ for $k \in ([0, +\infty) \cap \mathbb{N}_0) \cup \{\infty\}$, with compact support. When $A = B$ we simply denote $\mathcal{C}_c^k(A)$.

  When $k = 0$, we simply note $\mathcal{C}_c(A \to B)$ or $\mathcal{C}_c(A)$ if $A = B$.

- $\mathrm{d}f_x$ denote the differential of a function $f \in \mathcal{C}^1(A \to B)$ evaluated at point $x \in A$.

- $\|\cdot\|_{p,m}$ denote the p-norm $\|\cdot\|_{p,m} : x \in \mathbb{R}^m \mapsto \sqrt[p]{\sum_{i=1}^m |x_i|^p}$ on $\mathbb{R}^m$ with $1 \leqslant p < +\infty$. When the context is unambiguous, we simply denote it $\|\cdot\|_p$.

  Of paramount importance is the Euclidean norm, which is the 2-norm.

- $(\boldsymbol{e}_i)_{i=1}^d$ wille denote the canonical basis of $\mathbb{R}^d$, *i.e.* the basis such that the coordinates of e_i are zero except at position i for all $1 \leqslant i \leqslant d$.

- $\mathrm{L}^p(A \to \mathbb{R}^m, \lambda)$ denote the L^p quotient space of functions:

- $\mathcal{M}_{m,n}(\mathbb{K})$ denote the set of matrices with m lines and n columns with coefficients in a field $\mathbb{K}$.
- Given a set sequence $(A_i)_{1\leqslant i\leqslant d}$, $\prod_{i=1}^{d} A_i$ design the Cartesian cross-product of the k terms $(A_i)_{1\leqslant i\leqslant d}$. More formally, it is the operation defined by induction by:
  - if $d = 1$: $\prod_{i=1}^{d} A_i = A_1$
  - otherwise: $\prod_{i=1}^{d} A_i = \left(\prod_{i=1}^{d-1} A_i\right) \times A_d$
- Given a set sequence $(A_i)_{0\leqslant i\leqslant d}$ and a function sequence $(\alpha_i)_{1\leqslant i\leqslant d} \in \prod_{i=1}^{d} \mathcal{F}(A_{i-1} \to A_i)$, $\circ_{i=1}^{d}\alpha_i \in \mathcal{F}(A_0 \to A_d)$ design the $d-1$ consecutive functions composition. More formally, it is the operation defined by induction by:
  - if $d = 1$: $\circ_{i=1}^{d}\alpha_i = \alpha_1 \in \mathcal{F}(A_0 \to A_1)$
  - otherwise: $\circ_{i=1}^{d}\alpha_i = \underbrace{\alpha_d}_{\in\mathcal{F}(A_{d-1}\to A_d)} \circ \underbrace{\left(\circ_{i=1}^{d-1}\alpha_i\right)}_{\in\mathcal{F}(A_0\to A_{d-1})}$

C.2 . Adjoints

In the remainder of this chapter, we work in a Hilbert space setting. We shall therefore consider two Hilbert spaces $\mathcal{H}_\mathcal{A}$ and $\mathcal{H}_\mathcal{B}$, endowed with their respective inner products $\langle\cdot\,,\,\cdot\rangle_\mathcal{A}$ and $\langle\cdot\,,\,\cdot\rangle_\mathcal{B}$.

The *adjoint* T^* of an operator $T : \mathcal{H}_\mathcal{A} \to \mathcal{H}_\mathcal{B}$ is defined as an operator satisfying the fundamental property: for all $x \in \mathcal{H}_\mathcal{A}$ and $y \in \mathcal{H}_\mathcal{B}$,

$$\langle T(x)\,,\,y\rangle_{\mathcal{H}_\mathcal{B}} = \langle x\,,\,T^*(y)\rangle_{\mathcal{H}_\mathcal{A}}\,. \tag{C.1}$$

What is immediately apparent is that this notion depends on the choice of inner products, and therefore on the underlying metric structures. This dependence is a key feature, and one that is of central importance in our developments.

We begin by constructing the adjoint in the case of a bounded operator.

C.2.1 . Adjoint of a Bounded Operator

Let us fix a bounded operator $T : \mathcal{H}_\mathcal{A} \to \mathcal{H}_\mathcal{B}$. The existence of an adjoint operator T^* satisfying Equation (C.1) follows directly from the Riesz–Fréchet representation theorem. Indeed, for any $y \in \mathcal{H}_\mathcal{B}$, consider the map

$$l_y : \left\{ \begin{array}{ccc} \mathcal{H}_\mathcal{A} & \to & \mathbb{R} \\ x & \mapsto & \langle T(x)\,,\,y\rangle_{\mathcal{H}_\mathcal{B}} \end{array} \right. . \tag{C.2}$$

This map is continuous because T is bounded. Hence, by the Riesz–Fréchet theorem, there exists a unique $g_y \in \mathcal{H}_\mathcal{A}$ such that, for all $x \in \mathcal{H}_\mathcal{A}$,

$$\langle T(x)\,,\,y\rangle_{\mathcal{H}_\mathcal{B}} = \langle x\,,\,g_y\rangle_{\mathcal{H}_\mathcal{A}}\,. \tag{C.3}$$

We therefore define, for each $y \in \mathcal{H}_{\mathcal{B}}$,

$$T^*(y) := g_y. \tag{C.4}$$

Moreover, by bilinearity of the inner product $\langle \cdot\,,\,\cdot \rangle_{\mathcal{H}_{\mathcal{B}}}$, we have for all $\lambda \in \mathbb{R}$, $y_1, y_2 \in \mathcal{H}_{\mathcal{B}}$, and $x \in \mathcal{H}_{\mathcal{A}}$,

$$\begin{aligned} \langle x\,,\, T^*(\lambda y_1 + y_2)\rangle_{\mathcal{H}_{\mathcal{A}}} &= \langle T(x)\,,\, \lambda y_1 + y_2\rangle_{\mathcal{H}_{\mathcal{B}}} \\ &= \lambda \langle T(x)\,,\, y_1\rangle_{\mathcal{H}_{\mathcal{B}}} + \langle T(x)\,,\, y_2\rangle_{\mathcal{H}_{\mathcal{B}}} \\ &= \lambda \langle x\,,\, T^*(y_1)\rangle_{\mathcal{H}_{\mathcal{A}}} + \langle x\,,\, T^*(y_2)\rangle_{\mathcal{H}_{\mathcal{A}}}\,. \end{aligned} \tag{C.5}$$

Thus T^* is linear. In the following, we show that T^* is bounded.

Proposition C.1. *Let $T : \mathcal{H}_{\mathcal{A}} \to \mathcal{H}_{\mathcal{B}}$ be a bounded linear operator between Hilbert spaces, and let $T^* : \mathcal{H}_{\mathcal{B}} \to \mathcal{H}_{\mathcal{A}}$ be its adjoint. Then T^* is bounded and $\|T^*\| = \|T\|$.*

Proof. First, for any $y \in \mathcal{H}_{\mathcal{B}}$,

$$\begin{aligned} \|T^* y\|_{\mathcal{H}_{\mathcal{A}}} &= \sup_{\|x\|_{\mathcal{H}_{\mathcal{A}}} \leqslant 1} \left| \langle x\,,\, T^* y\rangle_{\mathcal{H}_{\mathcal{A}}} \right| = \sup_{\|x\|_{\mathcal{H}_{\mathcal{A}}} \leqslant 1} \left| \langle Tx\,,\, y\rangle_{\mathcal{H}_{\mathcal{B}}} \right| \\ &\leqslant \sup_{\|x\|_{\mathcal{H}_{\mathcal{A}}} \leqslant 1} \|Tx\|_{\mathcal{H}_{\mathcal{B}}}\ \|y\|_{\mathcal{H}_{\mathcal{B}}} = \|T\|\ \|y\|_{\mathcal{H}_{\mathcal{B}}}\,, \end{aligned} \tag{C.6}$$

so T^* is bounded and $\|T^*\| \leqslant \|T\|$. For the reverse inequality, for any $x \in \mathcal{H}_{\mathcal{A}}$ we have

$$\|Tx\|^2_{\mathcal{H}_{\mathcal{B}}} = \langle Tx\,,\, Tx\rangle_{\mathcal{H}_{\mathcal{B}}} = \langle T^*Tx\,,\, x\rangle_{\mathcal{H}_{\mathcal{A}}} \leqslant \|T^*T\|\ \|x\|^2_{\mathcal{H}_{\mathcal{A}}} \leqslant \|T^*\|\ \|T\|\ \|x\|^2_{\mathcal{H}_{\mathcal{A}}}\,, \tag{C.7}$$

Taking the supremum over all $\|x\|_{\mathcal{H}_{\mathcal{A}}} = 1$ yields

$$\|T\|^2 \leqslant \|T^*\|\ \|T\|\,. \tag{C.8}$$

If $T = 0$, the claim is trivial. Otherwise, divide Equation (C.8) by $\|T\| > 0$ to get $\|T\| \leqslant \|T^*\|$. Combining with Equation (C.6) gives $\|T^*\| = \|T\|$. □

Since T^* is bounded, it admits an adjoint $(T^*)^*$. The next proposition establishes that this procedure is consistent, in the sense that $(T^*)^* = T$.

Proposition C.2. *Let $T : \mathcal{H}_{\mathcal{A}} \to \mathcal{H}_{\mathcal{B}}$ be a bounded linear operator between Hilbert spaces and let $T^* : \mathcal{H}_{\mathcal{B}} \to \mathcal{H}_{\mathcal{A}}$ be its adjoint. Then $(T^*)^* = T$.*

Proof. By definition of the adjoint of T^*, for all $x \in \mathcal{H}_{\mathcal{A}}$ and $y \in \mathcal{H}_{\mathcal{B}}$,

$$\langle T^* y\,,\, x\rangle_{\mathcal{H}_{\mathcal{A}}} = \langle y\,,\, (T^*)^* x\rangle_{\mathcal{H}_{\mathcal{B}}}\,. \tag{C.9}$$

Using the adjoint relation for T gives $\langle T^* y\,,\, x\rangle_{\mathcal{H}_{\mathcal{A}}} = \langle y\,,\, Tx\rangle_{\mathcal{H}_{\mathcal{B}}}$. Hence for all $x, y \in \mathcal{H}_{\mathcal{A}}$

$$\langle y\,,\, (T^*)^* x\rangle_{\mathcal{H}_{\mathcal{B}}} = \langle y\,,\, Tx\rangle_{\mathcal{H}_{\mathcal{B}}}\,, \tag{C.10}$$

and by uniqueness in the Riesz representation, $(T^*)^* x = Tx$ for all x, i.e. $(T^*)^* = T$. □

Proposition C.3. *Let $T : \mathcal{H}_{\mathcal{A}} \to \mathcal{H}_{\mathcal{B}}$ be a bounded bijection with bounded inverse $T^{-1} : \mathcal{H}_{\mathcal{B}} \to \mathcal{H}_{\mathcal{A}}$. Then*

$$(T^*)^{-1} = (T^{-1})^*. \tag{C.11}$$

Proof. Let $w \in \mathcal{H}_{\mathcal{A}}$ and $x \in \mathcal{H}_{\mathcal{A}}$. Put $y := (T^*)^{-1}w \in \mathcal{H}_{\mathcal{B}}$. By the adjoint identity for T,

$$\langle x\,,\, w\rangle_{\mathcal{H}_{\mathcal{A}}} = \langle x\,,\, T^*y\rangle_{\mathcal{H}_{\mathcal{A}}} = \langle Tx\,,\, y\rangle_{\mathcal{H}_{\mathcal{B}}}\,. \tag{C.12}$$

Taking $x = T^{-1}z$ with arbitrary $z \in \mathcal{H}_{\mathcal{B}}$ yields for all $z \in \mathcal{H}_{\mathcal{B}}$

$$\left\langle T^{-1}z\,,\, w\right\rangle_{\mathcal{H}_{\mathcal{A}}} = \langle z\,,\, y\rangle_{\mathcal{H}_{\mathcal{B}}}\,. \tag{C.13}$$

By Riesz, this means $y = (T^{-1})^*w$. Since $y = (T^*)^{-1}w$ by definition, we obtain $(T^*)^{-1}w = (T^{-1})^*w$ for all $w \in \mathcal{H}_{\mathcal{A}}$, i.e. $(T^*)^{-1} = (T^{-1})^*$. □

In what follows, we highlight the fundamental interplay between the ranges and kernels of both T and its adjoint T^*.

Proposition C.4. *Let $T : \mathcal{H}_{\mathcal{A}} \to \mathcal{H}_{\mathcal{B}}$ be a bounded linear operator between Hilbert spaces. Then*

$$\overline{\operatorname{Im} T^*}^{\|\cdot\|_{\mathcal{H}_{\mathcal{A}}}} \;=\; (\operatorname{Ker} T)^{\perp}, \tag{C.14}$$

and

$$\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_{\mathcal{B}}}} \;=\; (\operatorname{Ker} T^*)^{\perp}. \tag{C.15}$$

Proof. We begin with a classical identity:

$$(\operatorname{Im} T)^{\perp} \;=\; \operatorname{Ker} T^*. \tag{C.16}$$

Indeed, if $y \in (\operatorname{Im} T)^{\perp}$, then for all $x \in \mathcal{H}_{\mathcal{A}}$ one has

$$0 = \langle Tx\,,\, y\rangle_{\mathcal{H}_{\mathcal{B}}} = \langle x\,,\, T^*y\rangle_{\mathcal{H}_{\mathcal{A}}}\,, \tag{C.17}$$

which implies $T^*y = 0$, *i.e.* $y \in \operatorname{Ker} T^*$. Conversely, if $y \in \operatorname{Ker} T^*$, then $T^*y = 0$, hence for all $x \in \mathcal{H}_{\mathcal{A}}$:

$$\langle Tx\,,\, y\rangle_{\mathcal{H}_{\mathcal{B}}} = \langle x\,,\, T^*y\rangle_{\mathcal{H}_{\mathcal{A}}} = 0, \tag{C.18}$$

so $y \in (\operatorname{Im} T)^{\perp}$. This proves Equation (C.16).

Now, applying the general fact that $M^{\perp\perp} = \overline{M}$ for any subspace M of a Hilbert space, we deduce directly that

$$\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_{\mathcal{B}}}} = (\operatorname{Ker} T^*)^{\perp}, \tag{C.19}$$

which is precisely Equation (C.15).

Finally, applying the same reasoning to T^* instead of T, and recalling that $(T^*)^* = T$ (see Proposition C.2), we obtain Equation (C.14). This completes the proof of both identities. □

We now establish an identity involving adjoints that often proves useful.

Proposition C.5. *Let* $\Pi_{\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}}} : \mathcal{H}_\mathcal{B} \to \mathcal{H}_\mathcal{B}$ *denote the orthogonal projection onto the closure* $\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}}$ *of* $\operatorname{Im} T \subset \mathcal{H}_\mathcal{B}$ *with respect to the norm* $\|\cdot\|_{\mathcal{H}_\mathcal{B}}$*. Then*

$$T^* \;=\; T^* \, \Pi_{\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}}} . \tag{C.20}$$

Proof. First observe that $\operatorname{Im} T^\perp$ is closed, hence $\Pi_{\operatorname{Im} T^\perp}$ is well defined. For any $y \in \mathcal{H}_\mathcal{B}$ and $x \in \mathcal{H}_\mathcal{A}$ we have

$$\langle T^* \Pi_{\operatorname{Im} T^\perp} y \, , \, x \rangle_{\mathcal{H}_\mathcal{A}} = \langle \Pi_{\operatorname{Im} T^\perp} y \, , \, Tx \rangle_{\mathcal{H}_\mathcal{B}} = \langle y \, , \, \Pi_{\operatorname{Im} T^\perp} Tx \rangle_{\mathcal{H}_\mathcal{B}} = 0, \tag{C.21}$$

since $Tx \in \operatorname{Im} T$. Next, recall the orthogonal decomposition

$$\mathcal{H}_\mathcal{B} = \overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}} \oplus^\perp \operatorname{Im} T^\perp . \tag{C.22}$$

Therefore, for any $y \in \mathcal{H}_\mathcal{B}$,

$$\begin{aligned} T^* y &= T^* \Big(\Pi_{\operatorname{Im} T^\perp} y + \Pi_{\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}}} y \Big) \\ &= \underbrace{T^* (\Pi_{\operatorname{Im} T^\perp} y)}_{=0} + T^* \Big(\Pi_{\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}}} y \Big) \\ &= T^* \Big(\Pi_{\overline{\operatorname{Im} T}^{\|\cdot\|_{\mathcal{H}_\mathcal{B}}}} y \Big) . \end{aligned} \tag{C.23}$$

This is precisely the claimed identity in Equation (C.20). □

When T is only densely defined rather than everywhere defined and bounded, the situation becomes more delicate. Nevertheless, as we shall see, one can still introduce two meaningful notions of adjoint, both of which will play an important role.

C.2.2 . Adjoints for densely defined closed operators

Let $T : \mathcal{D}(T) \subset \mathcal{H}_\mathcal{A} \to \mathcal{H}_\mathcal{B}$ be densely defined and closed (see Berezansky et al. 1996b, Chapter 12, Section 2). Two inner products may be used on the domain: the ambient $\langle \cdot \, , \, \cdot \rangle_{\mathcal{H}_\mathcal{A}}$ and the graph inner product $\langle \cdot \, , \, \cdot \rangle_{\Gamma_T}$ (see Berezansky et al. 1996b, Chapter 12, Section 1.2).

Adjoint with respect to $\langle \cdot \, , \, \cdot \rangle_{\mathcal{H}_\mathcal{A}}$. It does not fundamentally differ from the construction given in Section C.2.1, except that one must now be careful about two aspects. First, the map

$$l_y : \left\{ \begin{array}{ccc} \mathcal{D}(T) & \to & \mathbb{R} \\ x & \mapsto & \langle T(x) \, , \, y \rangle_{\mathcal{H}_B} \end{array} \right. \tag{C.24}$$

is no longer necessarily bounded. Second, it is *a priori* defined only for $x \in \mathcal{D}(T)$. We therefore set

$$\mathcal{D}(T^*) := \Big\{ y \in \mathcal{H}_\mathcal{B} \;\Big|\; x \in \mathcal{D}(T) \mapsto \langle T(x) \, , \, y \rangle_{\mathcal{H}_\mathcal{B}} \text{ is bounded} \Big\} . \tag{C.25}$$

Clearly $\mathcal{D}(T^*) \neq \varnothing$, since $0 \in \mathcal{D}(T^*)$. As expected, $\mathcal{D}(T^*)$ serves as the (not necessarily dense) domain of T^*.

Now let $y \in \mathcal{D}(T^*)$. By assumption, $l_y : \mathcal{D}(T) \to \mathbb{R}$ is continuous, i.e. there exists $C_y > 0$ such that for all $x \in \mathcal{D}(T)$,

$$|l_y(x)| \leqslant C_y \|x\|_{\mathcal{H}_\mathcal{A}} . \tag{C.26}$$

By density of $\mathcal{D}(T)$ in $\mathcal{H}_\mathcal{A}$, we can thus extend l_y to $\mathcal{H}_\mathcal{A}$ by setting, for $x \in \mathcal{H}_\mathcal{A}$,

$$l_y(x) := \lim_{n\to\infty} l_y(x_n), \tag{C.27}$$

where $(x_n) \in \mathcal{D}(T)^{\mathbb{N}}$ is any sequence with $x_n \to x$.

As in Section C.2.1, we may then invoke the Riesz–Fréchet representation theorem to obtain $g_y \in \mathcal{H}_\mathcal{A}$ representing l_y, and define $T^*(y) := g_y$. By the same reasoning, $T^* : (\mathcal{D}(T^*), \langle \cdot\,,\, \cdot \rangle_{\mathcal{H}_\mathcal{B}}) \to (\mathcal{H}_\mathcal{A}, \langle \cdot\,,\, \cdot \rangle_{\mathcal{H}_\mathcal{A}})$ is linear. We now establish that the operator T^* must in fact be closed.

Proposition C.6. *Let $T : \mathcal{D}(T) \subset \mathcal{H}_\mathcal{A} \to \mathcal{H}_\mathcal{B}$ be a densely defined linear operator between Hilbert spaces. Then the adjoint $T^* : \mathcal{D}(T^*) \subset \mathcal{H}_\mathcal{B} \to \mathcal{H}_\mathcal{A}$ is closed.*

Proof. To show that T^* is closed, let (y_n) be a sequence in $\mathcal{D}(T^*)$ such that $y_n \to y$ in $\mathcal{H}_\mathcal{B}$ and $T^* y_n \to z$ in $\mathcal{H}_\mathcal{A}$. For every $x \in \mathcal{D}(T)$ we have

$$\langle Tx\,,\, y_n \rangle_{\mathcal{H}_\mathcal{B}} = \langle x\,,\, T^* y_n \rangle_{\mathcal{H}_\mathcal{A}} . \tag{C.28}$$

Passing to the limit $n \to \infty$ gives for all $x \in \mathcal{D}(T)$

$$\langle Tx\,,\, y \rangle_{\mathcal{H}_\mathcal{B}} = \langle x\,,\, z \rangle_{\mathcal{H}_\mathcal{A}} . \tag{C.29}$$

Since $\mathcal{D}(T)$ is dense in $\mathcal{H}_\mathcal{A}$, this identity shows by definition that $y \in \mathcal{D}(T^*)$ and $T^* y = z$. Hence the graph of T^* is closed, i.e. T^* is a closed operator. □

Finally, we introduce a simple but important definition:

Definition C.7 (Symmetric and self-adjoint operator)**.** $T : \mathcal{D}(T) \subset \mathcal{H}_\mathcal{A} \to \mathcal{H}_\mathcal{A}$ is:

Symmetric if for all $x \in \mathcal{D}(T) \cap \mathcal{D}(T^*)$, one has $T(x) = T^*(x)$.

Self-adjoint if $T = T^*$, i.e. $\mathcal{D}(T) = \mathcal{D}(T^*)$ and $T(x) = T^*(x)$ for all $x \in \mathcal{D}(T)$. In particular, every self-adjoint operator is symmetric.

Adjoint with respect to $\langle \cdot\,,\,\cdot \rangle_{\Gamma_T}$. Note that since $(\mathcal{D}(T), \|\cdot\|_{\Gamma_T})$ is a Hilbert space, Section C.2.1 already guarantees the existence of an adjoint. Nevertheless, we now provide a more geometric construction, which is both conceptually enlightening and practically useful, as it yields a concrete "recipe" for computing adjoints. This approach is adapted from (Neuberger, 2009, Theorem 5.3).

Since T is assumed to be closed, its graph Γ_T is closed in $\mathcal{H}_\mathcal{A} \times \mathcal{H}_\mathcal{B}$. In particular, one can project onto Γ_T. We now wish to recover Equation (C.1). Let $y \in \mathcal{H}_\mathcal{B}$. Then

$$\begin{aligned} \langle T(x)\,,\, y\rangle_{\mathcal{H}_\mathcal{B}} &= \langle x\,,\, 0\rangle_{\mathcal{H}_\mathcal{A}} + \langle T(x)\,,\, y\rangle_{\mathcal{H}_\mathcal{B}} \\ &= \langle (x, T(x))\,,\, (0, y)\rangle_{\mathcal{H}_\mathcal{A}\times\mathcal{H}_\mathcal{B}} \\ &= \langle \Pi_{\Gamma_T}(x, T(x))\,,\, (0, y)\rangle_{\mathcal{H}_\mathcal{A}\times\mathcal{H}_\mathcal{B}} \\ &= \langle (x, T(x))\,,\, \Pi_{\Gamma_T}(0, y)\rangle_{\mathcal{H}_\mathcal{A}\times\mathcal{H}_\mathcal{B}}\,. \end{aligned} \tag{C.30}$$

By definition of Γ_T, there exists $\alpha \in \mathcal{D}(T)$ such that $(\alpha, T(\alpha)) = \Pi_{\Gamma_T}(0, y)$. Substituting this into Equation (C.30), we obtain

$$\begin{aligned} \langle T(x)\,,\, y\rangle_{\mathcal{H}_\mathcal{B}} &= \langle (x, T(x))\,,\, (\alpha, T(\alpha))\rangle_{\mathcal{H}_\mathcal{A}\times\mathcal{H}_\mathcal{B}} \\ &= \langle x\,,\, \alpha\rangle_{\Gamma_T} = \langle x\,,\, \Pi_1\Pi_{\Gamma_T}(0, y)\rangle_{\Gamma_T}\,. \end{aligned} \tag{C.31}$$

Hence the adjoint can be expressed as

$$T^* :\ y \in \mathcal{H}_\mathcal{B} \longmapsto \Pi_1\Pi_{\Gamma_T}(0, y). \tag{C.32}$$

This result is both surprising and powerful: it shows that the adjoint, defined *a priori* as an abstract object, can in fact be computed concretely as a composition of projections.

Before introducing differential operators—which will serve as the principal motivating examples for all of the abstract constructions developed thus far—we state a result of fundamental importance.

Proposition C.8. *If $U : \mathcal{H}_\mathcal{A} \to \mathcal{H}_\mathcal{B}$ is an isometry, then*

$$U^*U = \mathcal{I}_{\mathcal{H}_\mathcal{A}}, \qquad\qquad UU^* = \Pi_{\operatorname{Im} U}. \tag{C.33}$$

In particular $\operatorname{Im} U$ *is closed in* $\mathcal{H}_\mathcal{B}$.

Proof. The first identity follows from the fact that for all $x, y \in \mathcal{H}_\mathcal{A}$,

$$\langle x\,,\, U^*U[y]\rangle_{\mathcal{H}_\mathcal{A}} = \langle U[x]\,,\, U[y]\rangle_{\mathcal{H}_\mathcal{B}} = \langle x\,,\, y\rangle_{\mathcal{H}_\mathcal{A}}\,. \tag{C.34}$$

Thus $(U^*U(y) - y) \in \mathcal{H}_\mathcal{A}^\perp$, *i.e.* $U^*U = \mathcal{I}_{\mathcal{H}_\mathcal{A}}$. For the second, the key point is to show that $\operatorname{Im} U$ is closed, i.e. $\operatorname{Im} U = \overline{\operatorname{Im} U}$.

Let $y \in \overline{\operatorname{Im} U}$, and $(y_n) \in \operatorname{Im} U^{\mathbb{N}}$ with $y_n \to y$. Since (y_n) is Cauchy, and $y_n = U(x_n)$ for some $(x_n) \in \mathcal{H}_{\mathcal{A}}^{\mathbb{N}}$, we have

$$\|U(x_n) - U(x_m)\|_{\mathcal{H}_{\mathcal{B}}} = \|x_n - x_m\|_{\mathcal{H}_{\mathcal{A}}}\,, \tag{C.35}$$

so (x_n) is also Cauchy and converges to $x \in \mathcal{H}_{\mathcal{A}}$, since $\mathcal{H}_{\mathcal{A}}$ is complete. Since U is an isometry, we have for all $x \in \mathcal{H}_{\mathcal{A}}$

$$\|U(x)\|_{\mathcal{H}_{\mathcal{B}}} = \|x\|_{\mathcal{H}_{\mathcal{A}}}\,. \tag{C.36}$$

In particular, U is bounded with operator norm $\|U\| = 1$, and hence continuous. Thus $U(x) = y$, hence $y \in \operatorname{Im} U$. We conclude that $\operatorname{Im} U$ is closed in $\mathcal{H}_{\mathcal{B}}$. Finally:

- For $y \in \operatorname{Im} U$, say $y = U(x)$, we have

$$UU^*(y) = U(U^*U)(x) = U(x) = y. \tag{C.37}$$

- For $y \in (\operatorname{Im} U)^{\perp}$, we check that $UU^*(y) = 0$. Indeed, for any $z \in \mathcal{H}_{\mathcal{B}}$,

$$\langle z\,,\, UU^*(y)\rangle_{\mathcal{H}_{\mathcal{B}}} = \langle UU^*(z)\,,\, y\rangle_{\mathcal{H}_{\mathcal{B}}} = 0, \tag{C.38}$$

  since $UU^*(z) \in \operatorname{Im} U$. Thus $UU^*(y) \in \mathcal{H}_{\mathcal{B}}^{\perp}$, *i.e.* $UU^*(y) = 0$.

□

***Remark* C.9.** In particular, every isometry U is injective. If, in addition, U is surjective, then $U^* = U^{-1}$. This identity also holds if U is co-restricted to its image.

C.3 . Elements of General Spectral Theory

Definition C.10 (Projection-valued measure)**.** Let $(X, \mathcal{A})$ be a measurable space, and let $\mathcal{H}$ be a Hilbert space. A *projection-valued measure* (PVM) on $(X, \mathcal{A})$ is a map

$$\pi : \mathcal{A} \to \mathcal{L}_b(\mathcal{H} \to \mathcal{H}), \tag{C.39}$$

where $\mathcal{L}_b(\mathcal{H} \to \mathcal{H})$ denotes the set of bounded linear operators on $\mathcal{H}$, such that:

1. for every $A \in \mathcal{A}$, $\pi(A)$ is an orthogonal projection on $\mathcal{H}$;
2. $\pi(\varnothing) = 0$ and $\pi(X) = I_{\mathcal{H}}$, where $I_{\mathcal{H}}$ is the identity operator on $\mathcal{H}$;
3. $\pi(A \cap B) = \pi(A)\pi(B)$ for all $A, B \in \mathcal{A}$;

4. for every countable family $(A_i)_{i=1}^{\infty}$ of disjoint measurable subsets of X,

$$\pi\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \pi(A_i), \tag{C.40}$$

where the series converges in the strong operator topology.

***Remark* C.11.** Projection-valued measures behave like ordinary measures, except that their values are operators instead of scalars. In particular, one can define integrals of the form

$$\int_X f(\lambda)\, \pi(\mathrm{d}\lambda) \tag{C.41}$$

for suitable measurable functions $f : X \to \mathbb{R}$ (or $\mathbb{C}$), extending the notion of the Lebesgue integral to operator-valued settings. For details, see (Berezansky et al., 1996b, Chapter 13).

Theorem C.12 (Spectral theorem for bounded self-adjoint operators)**.** *Let $\mathcal{H}$ be a Hilbert space, and let $A : \mathcal{H} \to \mathcal{H}$ be a bounded self-adjoint operator. Then there exists a projection-valued measure π on the Borel σ-algebra $\mathcal{B}(\mathbb{R})$ such that*

$$A = \int_{\mathbb{R}} \lambda\, \pi(d\lambda) = \int_{\sigma(A)} \lambda\, \pi(d\lambda), \tag{C.42}$$

where $\sigma(A)$ denotes the spectrum of A.

Definition C.13 (Coercive operator)**.** A self-adjoint operator $A : \mathcal{H} \to \mathcal{H}$ is said to be *coercive* if there exists a constant $c > 0$ such that

$$\langle Ax\,,\, x\rangle_{\mathcal{H}} \geqslant c\, \|x\|_{\mathcal{H}}^2\,, \qquad \text{for all } x \in \mathcal{H}. \tag{C.43}$$

Proposition C.14 (Spectral characterization of boundedness and coercivity)**.** *Let $A : \mathcal{H} \to \mathcal{H}$ be a bounded self-adjoint operator.*

1. *A is bounded if and only if its spectrum $\sigma(A)$ is bounded both above and below.*

2. *A is coercive if and only if $\sigma(A)$ is bounded below by a strictly positive constant.*

Proposition C.15 (Inverse and square root)**.** *Let $A : \mathcal{H} \to \mathcal{H}$ be a bounded, self-adjoint, and coercive operator with spectral decomposition*

$$A = \int_{\sigma(A)} \lambda\, \pi(d\lambda), \tag{C.44}$$

where π is the projection-valued measure associated with A.

1. *The inverse of A is given by*

$$A^{-1} = \int_{\sigma(A)} \frac{1}{\lambda}\, \pi(d\lambda), \tag{C.45}$$

and is bounded with $\left\|A^{-1}\right\| \leqslant \frac{1}{\inf \sigma(A)}$.

2. *The positive square root of A is defined by*

$$A^{1/2} = \int_{\sigma(A)} \sqrt{\lambda}\, \pi(d\lambda), \tag{C.46}$$

which is again a bounded, self-adjoint, and coercive operator satisfying

$$(A^{1/2})^2 = A. \tag{C.47}$$

***Remark* C.16.** The spectral calculus allows one to define $f(A)$ for any bounded Borel function $f : \sigma(A) \to \mathbb{R}$ by

$$f(A) := \int_{\sigma(A)} f(\lambda)\, \pi(\mathrm{d}\lambda), \tag{C.48}$$

extending the polynomial functional calculus to arbitrary continuous (or even measurable) functions.