KØBENHAVNS UNIVERSITET
INSTITUT FOR MATEMATISKE FAG

Masters thesis in Mathematics

Approximating solutions to partial differential equations with physics-informed reproducing kernels

Date: June 30 2025

Supervisor: Oswin Krause

Department: Department of Computer Science

Author: Valdemar Skou Knudsen

Title: Approximating solutions to partial differential equations with physics-informed reproducing kernels.

Abstract: In this thesis, we present and explore the reproducing kernel Hilbert space regression framework in the context of estimating behavior governed by physics-based partial differential equations. We implement the framework with different optimization procedures and equation examples, and evaluate our results based on how we the underlying equation behavior is captured. We present a method for transforming the reproducing kernel to naturally uphold global properties imposed by such equations, in the form of conservation laws, and demonstrate that the resulting kernel expansion estimators are more physically valid and also yield improved numerical accuracy.

Emnebeskrivelse: I denne afhandling præsenterer og udforsker vi reproducing kernel Hilbert space-regressionsmetoden (RKHSs) i en kontekst hvor opførsel, der er styret af fysik-baserede partielle differentialligninger (PDEs) skal estimeres. Vi implementerer denne metode for forskellige optimeringsprocedurer og for forskellige ligninger, og evaluerer resultaterne ud fra hvorvidt den bagvedliggende ligningsadfærd gengives korrekt. Vi præsenterer en metode til at transformere en RKHS kerne, så den naturligt opretholder de globale egenskaber, i form at bevarelseslove, som sådanne ligninger pålægger, og vi demonstrerer at de optimerede kernel expansion estimatorer er bedre til at opholde den ønskede fysiske opførsel, og endvidere også er mere numerisk nøjagtige.

Supervisor: Oswin Krause

Date of submission: June 30, 2025

Contents

1 Introduction

Within optimization and statistical learning theory, physics-informed methods are becoming increasingly popular for solving a wide variety of problems. Data-driven methods have become almost ubiquitous in every scientific field, but the underlying statistical fitting procedures may fail to capture underlying physical principles or enforce conservation laws. As such, careful consideration or modification is required in order to apply these methods successfully. We will investigate a particular example of such a modification: when an aspect of the data/system may be considered as governed by a partial differential equation (PDE). In this situation, we want our solution procedure to, in some way, respect either the PDE itself, or some property derived from the PDE (for instance a conservation law). Examples in recent literature where this ”physics-informed” approach has been considered in an optimization framework are numerous, including in machine learning [10], reproducing kernel Hilbert spaces (RKHS) [9], and Gaussian processes, [7].

This thesis focuses on the reproducing kernel Hilbert space approach, a framework within functional analysis that is increasingly popular for applications in a wide variety of optimization problems and within statistical learning theory. Associated to a so-called reproducing kernel function K is a unique Hilbert $\mathcal{H}$ space of functions, with properties such as regularity or other differential constraints closely related to the kernel itself. Solving a minimization problem in $\mathcal{H}$ then means exactly that K is acting as an implicit prior to the solution space: any solution to a minimization problem in $\mathcal{H}$ may be written as a linear combination of basis functions that are directly derived from K.

A reproducing kernel K may be transformed by a bounded linear operator L to become a reproducing kernel for a new Hilbert space, in order to satisfy some additional constraint or property. Rather than relying on the optimizer to capture this property sufficiently from the data, we incorporate it directly into our implicit prior. This can be any linear constraint, such as enforcing the inclusion of a variable in the approximate solution, to a integro-differential operator (as long as it is bounded) acting on K to obey some physical property. This latter scenario is what we will refer to as a physics-informed RKHS framework.

We do a investigation into the physics-informed RKHS framework. In particular, we study the effects of transforming a kernel K to obey conservation laws derived from the underlying physics. Using these transformed kernels, we approximate solutions to a linear PDE boundary value problem, the heat equation, using a residual-loss based method. We also consider data-driven approximations to a nonlinear differential equation, the Richards equation. The overall purpose is therefore twofold: comparing to baseline (Gaussian reproducing kernel) solutions, we investigate if the transformed kernel is a better direct solver, and if it better captures underlying physical behavior not directly found in the data. The methodology can however be implemented to any PDE problem where a theoretical conservation law can be derived.

The heat equation is both widely used in applications, but is also a common toy-model problem for evaluating numerical efficiency; and is a well-behaved equation numerically and theoretically, with a Green's function representation. The nonlinear Richards equation is used for modelling the movement of water in unsaturated soils, and is commonly found in geoscience and agriculture applications. It is infamously very difficult to effectively solve numerically [6]. Using these two equations as our main examples, we can therefore see how the transformed kernel framework performs in both simple and complicated scenarios.

The first part of the thesis (chapter 2) gives a brief introduction to the theory of PDEs and RKHSs, sufficient for our applications. In particular, we explain what a boundary value problem for a partial differential equation is, and then define our two main examples. In the RKHS section, we define the basic framework and then state and prove the results necessary for us to consider the differential kernel transformations that we will use.

The second part of the thesis (chapter 3 and 4) is more application-focused: we discuss the methodology for defining our kernel transformations and then implement them in two different ways with our example boundary value problems. We compare and contrast the two methods with a baseline implementation, probing accuracy, physical validity, and computational cost. Finally we contextualize our results in the discussion section, and speculate how we could extend the methods used in the thesis to other interesting applications.

1.0.1 Prerequisites

The thesis assumes some basic familiarity with partial differential equations and functional analysis, enough to be comfortable with some terms from a first course in both. Nevertheless, some of this is refreshed in chapter 2. Basic programming knowledge in `Python` is used throughout the thesis and is therefore also assumed.

1.0.2 Acknowledgments

I express sincere gratitude to my advisor Oswin Krause for his great supervision and insight in the material discussed in the thesis. Most importantly for his patience with me during the thesis process.

I would like to also thank my friends and family for helpful comments and motivation.

1.1 State of the art

RKHSs have already found extensive use within the numerical solution of PDEs. Within linear equations (such as the heat equation), there is particularly inspiring development found in [9]. Here, the authors combine functional linear regression and operator RKHSs to find the optimal Green's function of the fundamental solution to various linear PDEs. In particular they use ensemble input-output data (in the form of an initial/boundary condition on the PDE domain, and then ”true” function values of the PDE solution) to compute a best-fit fundamental solution that can then used to solve any similar boundary value problem (with varying success). This is particularly interesting since it goes beyond solving just a single boundary value problem. The Green's function solves a particular boundary value problem related to the PDE, which means in particular that it inherits conservation laws and other properties related to the equation. The authors in [9] briefly considered this fact, and designed kernels that coordinate symmetry or time-invariance properties. However, they stopped short of attempting to impose more complicated physical conservation laws. An additional limitation to their approach is that it cannot be applied to almost every nonlinear problem, since the existence of a fundamental solution is usually false.

RKHS methods are sometimes used to solve nonlinear equations, for instance in [2]. Here, the authors are limited to solving one boundary value problem at a time. The main benefit to this method is that analytical derivatives can be computed in the kernel expansion, which is superior to automatic differentiation, finite differences, or other tools. The kernel expansion approximate solution will also be an analytical function. Similar methodology is often found physics-informed neural network (PINN) literature, for instance [11], where orthogonal polynomial expansions are used to solve boundary value problems. Modifying the basis functions in these methods to better adhere to the underlying physics is not considered in either of these articles, and is generally rarely found in the context of nonlinear equations.

In [7], the authors construct a general method for transforming Gaussian processes to obey a linear differential constraint. Their results show both improved convergence results and greater physical validity of solutions. Incorporating similar techniques into a RKHS framework and possibly extending them to nonlinear constraints is therefore a significant point of research interest.

Note that physics-informed neural networks are also increasingly growing in popularity, see [10]. In this book, the authors consider various technique that leverage a governing physical equation in loss function formulation for neural networks. These physics-informed loss functions are shown to produce better convergence and also act as a regulatory agent for the network by essentially projecting into a submanifold of possible solutions. However since this approach is purely formulated in terms of the loss-function, no such projection actually takes place, and as such it is difficult still to fully guarantee validity of these methods. As such, the approach in [2], and similar in a RKHS setting, are more interesting from a purely mathematical perspective, but more importantly they have a potential upside in terms of practical applications.

While there is great interest in extending these methods to nonlinear equations, so far it has not been applied to the Richards equation. In [6], a high-level overview of the current state of numerical methods for the Richards equation is given, focusing exclusively on traditional discretization methods that are mesh-based. The specific nonlinearity of the equation means that most of these methods will be extremely unstable due to high gradient values occurring at the wetting front, especially when running high fidelity simulations (often required to guarantee desired error estimates).

2 Background and theory

2.1 Partial Differential Equations (PDEs)

2.1.1 Terminology, classifications, examples

In this section we give a brief overview on some basic terminology, examples, and results within the theory of partial differential equations. The purpose of this subsection is to provide sufficient theoretical background for the existence and uniqueness comments in the introductory sections, as well as the later numerical approaches. The main point of reference, standard in the literature, is [5]. Let $\Omega \subseteq \mathbb{R}^n$ be open and $u : \Omega \to \mathbb{R}$. A partial differential equation (PDE) of order d with unknown $u = u(x)$ may be formulated as:

$$F(D^d u, D^{d-1} u, \cdots, Du, u, x) = 0, \quad x \in \Omega, \tag{1}$$

where $F : \mathbb{R}^{n^d} \times \mathbb{R}^{n^{d-1}} \times \cdots \times \mathbb{R}^n \times \mathbb{R} \times \Omega$ is given, and $D^k u$ is the k'th derivative tensor of u (the equation (1) may equivalently be formulated using differential operators). We solve the equation, possibly along with auxiliary boundary conditions on some set $\Gamma \subseteq \partial\Omega$, by finding explicit solution formulae, or, failing that, by obtaining abstract existence, uniqueness, and regularity (higher orders of differentiability) properties of solutions. Even when existence and uniqueness is guaranteed, explicit solution formulae are not possible for the vast majority of equations. This means that numerical approaches are necessary if one wishes to explicitly evaluate the solution. In particular, mathematical models which include PDEs will need to include complicated algorithms that accurately and efficiently can compute approximations of the desired solution. Even for these applied purposes, it is still important to have existence and uniqueness guaranteed, since these properties directly correspond to solution algorithms being well-behaved: A PDE with an infinite number of solutions (i.e. the Eikonal equation $|\nabla u| = 1$) cannot be solved with numerical algorithms unless additional constraints are imposed.

The traditional strategy when proving existence and uniqueness for a given PDE involves first classifying the type of equation. The most important and immediate classification is that of linearity: we say that the equation (1) is linear if has the form:

$$\sum_{|\alpha| \leq d} a_\alpha(x) D^\alpha u = f(x), \tag{2}$$

for given functions a_α ($|\alpha| \leq d$), f. If $f \equiv 0$ then we say the equation is homogeneous. The equation (1) is semi-linear if it has the form:

$$\sum_{|\alpha| = d} a_\alpha(x) D^\alpha u + a_0(D^{d-1} u, \ldots, Du, u, x) = 0. \tag{3}$$

The equation (1) is quasi-linear if it has the form:

$$\sum_{|\alpha|=d} a_\alpha(D^{d-1}u, \ldots, Du, u, x) D^\alpha u + a_0(D^{d-1}u, \ldots, Du, u, x) = 0. \tag{4}$$

If the equation (1) is nonlinear with respect to its highest order derivatives, then we say that it is fully nonlinear. An informal intuition is that the more nonlinearity affects higher derivatives (going from linear to fully nonlinear) in the equation, the more difficult it is to solve (i.e. prove existence and uniqueness). Numerical approaches are also similarly affected, in the sense that more nonlinearity results in a more complicated approximation scheme nonlinear discrete equation, increasing computation time.

Another very important classification of PDEs is that of ellipticity. Equations that are elliptic can often be solved by a well-understood family of methods. Restricting ourselves to second order equations for brevity, i.e. $F(D^2u, Du, u, x) = 0$, we say that the equation is (uniformly) elliptic if there exist positive constants $\lambda \leq \Lambda$ such that:

$$\lambda \left\| N \right\| \leq F(M+N, \ldots) - F(M, \ldots) \leq \Lambda \left\| N \right\|, \quad \forall M \in \mathcal{S}, \quad \forall N \geq 0, \tag{5}$$

where $\mathcal{S}$ is the space of symmetric matrices, $N \geq 0$ means that $N \in \mathcal{S}$ is non-negative symmetric, and $\|N\| = \sup_{|z|\leq 1} |Nz|$ is the usual matrix operator norm. An equation of the form

$$u_t - F(D^2u, Du, u, x) = 0, \tag{6}$$

where $t > 0$ is interpreted as a time-variable, is called parabolic if F is elliptic. In that sense, the temporal evolution of u is balanced by the elliptic operator. Elliptic equations may therefore be interpreted as steady-state (time independent) solutions for parabolic equations.

Example 1 (Heat equation)**.** The heat equation for known functions $f : \Omega \to \mathbb{R}$ and $k : \mathbb{R} \to \mathbb{R}$, and an unknown function $u : \Omega \times [0, \infty) \to \mathbb{R}$ is defined as:

$$\partial_t u(x,t) - \nabla \cdot (k(x) \nabla u(x,t)) = f(x,t), \quad x \in \Omega, \tag{7}$$

When normalized ($k \equiv 1$) this reduces to $\partial_t u - \Delta u = f$, where Δ is the Laplace operator.

The heat equation arises from combining the law of energy conservation with Fourier's law of heat conduction to describe the temperature evolution over time in a material body Ω. The unknown function u is called the *temperature field*, while the known function k is the *thermal conductivity* function, and f is the source/sink term (external heat input or loss).

The heat equation is a linear parabolic equation, which means that existence of solutions is guaranteed (see [5] on parabolic equations), and uniqueness of solutions is also true contingent on decay constraints at infinity, or on Ω being bounded.

Example 2 (Richards equation)**.** The Richards equation (in isotropic conditions) for known functions $f : \mathbb{R} \times \Omega \times [0, \infty) \to \mathbb{R}$ and $K : \mathbb{R} \to \mathbb{R}$, and unknown functions $\theta, h : \Omega \times [0, \infty) \to \mathbb{R}$ is defined as:

$$\partial_t \theta(x,t) - \nabla \cdot (K(\theta(x,t))(\nabla h(x,t) + e_z)) = f(\theta(x,t), x, t), \tag{8}$$

where e_z denotes the unit vector in the vertical dimension.

The Richards equation combines the law of mass-conservation with the Darcy-Buckingham law of flor in porous media to describe the movement and root-water intake of water over time in a variably saturated soil-column Ω. The unknown functions θ, h are respectively called the *moisture content* function and the *pressure-head* function, while the known function K is called the *unsaturated hydraulic conductivity* function, and f is the sink term (usually root-water uptake).

The equation contains two unknown variables (θ and h), which means that the equation is not immediately well-posed. To remedy this we assume a constitutive relationship between θ and h, i.e. $h = h(\theta)$, meaning that our primary unknown is the moisture content function. Assuming $\Omega \subseteq \mathbb{R}$ (one spatial dimension), we can then rewrite the equation as:

$$\partial_t \theta - \partial_x (K(\theta) h'(\theta) \partial_x \theta + K(\theta)) = 0. \tag{9}$$

Note that it is also possible to expand all the derivatives. Observe that the Richards equation is a nonlinear analogue of the heat equation, specifically from assuming that the conductivity is dependent on the unknown. As a result, the Richards equation is a quasilinear equation (the highest order derivative is $\partial_{xx}\theta$, which has $K(\theta)h'(\theta)$ as its coefficient) of elliptic-parabolic type (elliptic when $\partial_t \theta = 0$), which means that the existence (and uniqueness) of a solution θ requires careful theoretical consideration. The standard reference for existence is [1], under various regularity assumptions on K. In particular, if K has zeros then the reference only guarantees the existence of a weak-limit. This can freely be assumed, as $K = 0$ implies impermeability in the soil, violating the isotropic assumption. Areas of impermeability (say, a rock in the soil column) may be of interest, but can instead be defined by considering a different (non-simply connected) domain Ω with appropriate boundary conditions.

The nonlinearity of the Richards equation described above means that obtaining closed-form solutions is not possible, except for special toy-model cases. Therefore, numerical approximations are appropriate when using this equation for modelling purposes. While many theoretical properties relating to regularity of (8) and similar equations exist in the literature, such results are very difficult to apply when moving into the discrete world. Numerical research on the Richards equation is vast, and the current state of affairs was also briefly described in the introduction. A more thorough discussion is found in [6], but one overarching aspect is readily summarized and bears repeating: there is still significant theoretical and computational shortcomings in most numerical solvers. The equation has a switch-like behavior, i.e. when the soil becomes saturated and $\partial_t \theta = 0$. Numerically this results in local steady-state behavior, which means very large derivative values in the intermediate areas, causing erratic behavior in the solver and therefore also significant uncertainty about the convergence of any numerical solver. Often, a division by zero will cause the whole approximate solution to be invalid.

In higher dimensions ($\Omega \subseteq \mathbb{R}^n$, $n = 2, 3$) there is significant computational cost because of the curse of dimensionality, compounded by the fact that linearization is usually required, leading to iteration schemes and many matrix inversions. As mentioned in [6], all of these issues arise in almost any traditional numerical solver, and consequently new solution methods using novel techniques continue to be of great scientific interest.

2.1.2 Boundary value problems

Assume now that $\Omega \subseteq \mathbb{R}^d$ is bounded and, for time-dependent problems, that we have a designated stopping time $T > 0$. A boundary value problem (BVP) consists of a PDE (1) along with additional (differential) conditions imposed on the unknown function on the boundary $\partial\Omega \times (0, T]$. Additionally, an additional point-wise constraint is usually imposed at $t = 0$, i.e. $u(x, 0) = u_0(x)$ for all $x \in \Omega$. Restricting ourselves again to second order equations for brevity, a general BVP for a time-dependent problem can then be written as:

$$\begin{cases} \partial_t u - F(D^2 u, Du, u, x) = 0, & (x,t) \in \Omega \times (0,T) \\ G(Du, u, x) = 0, & (x,t) \in \partial\Omega \times (0,T) \ , \\ u(x,0) = u_0(x), & x \in \Omega \end{cases} \tag{10}$$

where G is some real-valued function defining our boundary condition. The most common boundary conditions are the Dirichlet (point-wise) condition $u(x,t) = g(x,t)$ and the Neumann condition $\partial_n u(x,t) = g(x,t)$ on $(x,t) \in \partial\Omega \times (0,T)$, where n is the exterior normal of Ω. Mixed conditions (Robin) are also commonly used.

Example 3 (Heat equation, cont.)**.** The normalized one-dimensional heat equation on the spatiotemporal domain $[-1, 1] \times [0, T]$ with Neumann boundary conditions is the BVP given by:

$$\begin{cases} \partial_t u - \partial_{xx} u = 0, & (x,t) \in (1,1) \times (0,T] \\ u_x(-1,t) = u_x(1,t) = 0, & t \in (0,T] \\ u(x,0) = u_0(x), & x \in [-1,1] \end{cases} \tag{11}$$

where $u_0 : [-1, 1] \to \mathbb{R}$ is known.

In this setting, the Neumann boundary conditions can be interpreted as an insulator for the domain $[-1, 1]$. Indeed, they ensure a total mass/energy conservation property of the unknown solution function:

$$\begin{aligned}\partial_t \int_{[-1,1]} u(x,t)\,dx &= \int_{[-1,1]} \partial_{xx} u(x,t)\,dx \\ &= u_x(1,t) - u_x(-1,t) \\ &= 0.\end{aligned} \tag{12}$$

In particular this means that:

$$\int_{[-1,1]} u(x,t)\,dx = \int_{[-1,1]} u(x,0)\,dx =: m_0. \tag{13}$$

In numerical contexts, we may therefore judge the validity of any approximate solution by how well it upholds this criteria. Note also that this constraint is linear u.

Example 4 (Richards equation, cont.)**.** The one-dimensional Richards equation on a soil column of length L with constant Dirichlet boundary conditions is the BVP on the spatiotemporal domain $[0, L] \times [0, T]$ given by:

$$\begin{cases} \partial_t \theta - \partial_x (K(\theta) h'(\theta) \partial_x \theta + K(\theta)) = 0, & (x,t) \in (0, L) \times (0, T] \\ \theta(0,t) = \theta_s,\, \theta(L,t) = \theta_r & t \in (0, T] \\ \theta(x,0) = \theta_0(x), & x \in [0, L] \end{cases}. \tag{14}$$

Recall that the Richards equation is a nonlinear analogue of the heat equation, so a mass balance property is also expected. However, the nonlinearity in the equation as well as the more complicated boundary condition prevents a neat derivation as in the previous example. Instead we follow the principle that the total mass added to the domain has the be equal to the net flux into the domain:

$$\int_{[0,L]} (\theta(x,T) - \theta(x,0))\,dx = \int_{[0,T]} (q(L,t) + q(0,t))\,dt, \tag{15}$$

where $q(x,t) = K(\theta(x,t)) h'(\theta(x,t)) \partial_x \theta(x,t) + K(\theta(x,t))$. As in the previous example, we may again judge the validity of any approximate solution by how well it upholds this criteria. Note however that this constraint is highly nonlinear with respect to θ, due to the right-hand term.

We also mention that for well positive K and monotone constitutive relation $\theta = \theta(h)$, a maximum principle argument shows that $\theta(x,t) \in [\theta_r, \theta_s]$ for all (x,t), so solutions to the BVP must be bounded.

2.2 Reproducing Kernel Hilbert Spaces (RKHSs)

2.2.1 Definitions and basic examples

In this section we introduce reproducing kernel Hilbert spaces (RKHSs) with enough depth for our applications: we need to be able to construct basic RKHS and then construct subspaces according to constraints that we impose on our functions. As such we need to know criteria for when a set of functions is a RKHS, and under what transformations it remains one. Our main references for this secion is [8] and [3]. We begin with a definition:

Definition 1. Let $\mathcal{F}(X)$ denote the set of complex-valued functions on some non-empty background set X. A reproducing kernel Hilbert space (RKHS) is then a tuple $(\mathcal{H}, K)$ where $\mathcal{H} \subseteq \mathcal{F}(E)$ is a Hilbert space of functions, and $K : X \times X \to \mathcal{H}$ satisfies the reproducing property: for all $p \in X$ and $f \in \mathcal{H}$ we have that $K_p := K(\cdot, p) \in \mathcal{H}$ and $f(p) = \langle f, K_p \rangle_{\mathcal{H}}$.

Observe that the main object in the RKHS definition is in fact the reproducing kernel K, since it allows us to point-wise evaluate every function $f \in \mathcal{H}$.

Definition 2. A function $K : X \times X \to \mathbb{C}$ is called positive definite on X if for any function $X : X \to \mathbb{C}$ and any finite index set $F \subseteq \mathbb{N}$,

$$\sum_{i,j \in F} \overline{X(p_i)} X(p_j) K(p_i, p_j) \geq 0. \tag{16}$$

An immediate connection between these definitions is found in the following theorem:

Theorem 2.1 (Moore-Aronszajn). *Let $K : X \times X \to \mathbb{C}$ be symmetric and positive definite. Then there exists a unique Hilbert space $\mathcal{H} = \mathcal{H}_K(X)$ of functions on X that has K as its reproducing kernel.*

Proof. Consider $H_0 = \text{span}(K_x \mid x \in X)$ and define an inner product on H_0 as follows:

$$\left\langle \sum_{i=1}^{n} a_i K_{x_i}, \sum_{j=1}^{m} b_j K_{y_j} \right\rangle_{H_0} := \sum_{\substack{i \leq n \\ j \leq m}} a_i b_j K(x_i, y_j). \tag{17}$$

Note that symmetry and positive-definiteness of the inner product follow directly from our assumptions on K. Additionally we trivially have $\langle K_x, K_y \rangle_{H_0} = K(x, y)$ for all $x, y \in X$. Let $\mathcal{H}$ be the completion of H_0 with respect to this inner product; its elements are infinite sums with respect to the basis functions $\{K_x \mid x \in X\}$ which converge in the Cauchy sense. For $f \in \mathcal{H}$ and $x \in X$ we verify the reproducing property:

$$\langle f, K_x \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} a_j \langle K_{x_i}, K_x \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} a_j \langle K_{x_i}, K_x \rangle_{\mathcal{H}_0} = \sum_{j=1}^{\infty} a_j K(x_j, x) = f(x), \tag{18}$$

by construction. To see uniqueness, suppose that $\mathcal{G}$ is another complete Hilbert space of functions with K as its reproducing kernel. In particular this means that $K_x \in \mathcal{G}$ so $H_0 \subseteq \mathcal{G}$ and then also $\mathcal{H} \subseteq \mathcal{G}$ by minimality. For the converse inclusion, let $f \in \mathcal{G}$ and write $f = f_{\mathcal{H}} + f_{\mathcal{H}^\perp}$. For all $x \in X$ we have (since K is reproducing kernel for both $\mathcal{G}$ and $\mathcal{H}$):

$$f(x) = \langle f, K_x \rangle_{\mathcal{G}} = \langle f_{\mathcal{H}}, K_x \rangle_{\mathcal{H}} + \underbrace{\langle f_{\mathcal{H}^\perp}, K_x \rangle_{\mathcal{G}}}_{=0} = f_{\mathcal{H}}(x), \tag{19}$$

so $f_{\mathcal{H}^\perp} \equiv 0$ and hence $f \in \mathcal{H}$, finishing the proof. □

The Moore-Aronszajn theorem tells us that that there is a one-to-one correspondence between the reproducing kernel and its Hilbert space. Consequently, whenever K is a reproducing kernel, we shall from here on write its associated space of functions as $\mathcal{H}_K(X)$. Moreover, we can now write:

$$\mathcal{H}_K(X) = \overline{\left\{ x \mapsto \sum_{i=1}^{n} \alpha_i K(x, x_i) \mid n \in \mathbb{N}, \forall i = 1, \ldots, n : \alpha_i \in \mathbb{C}, x_i \in X \right\}} \tag{20}$$

The Moore-Aronszajn theorem also tells us that we just need to check positive-definiteness to verify if a map K is a reproducing kernel to *some* Hilbert space. Since many kernels that we will consider are radial and norm-based, i.e. $K(x, y) = k(\|x - y\|)$, positive-definiteness is required of k instead, in which we can use the famous Bochner theorem, a simplified version stated as follows:

Theorem 2.2 (Bochner). *$k \in C(\mathbb{R}^s)$ is positive definite if and only if it is the Fourier transform of a finite non-negative (Borel) measure μ, i.e. for all $x \in \mathbb{R}^s$*

$$k(x) = \int_{\mathbb{R}^s} e^{i\langle x, y \rangle_{\mathbb{R}^s}} \, d\mu(y). \tag{21}$$

Proof. We shall prove only the forward (and easier) direction, which happens to also be direction we need to quickly establish the RKHS property for various radially-defined kernels. So, let $x_1, \ldots, x_N \in \mathbb{R}^d$ and $c_1, \ldots, c_N \in \mathbb{C}$ be arbitrary, and compute:

$$\begin{aligned} \sum_{i,j=1}^{N} c_i \overline{c_j} k(x_i - x_j) &= \sum_{i,j=1}^{N} c_j \overline{c_j} \int_{\mathbb{R}^s} e^{i\langle x_i - x_j, y \rangle_{\mathbb{R}^s}} \, d\mu(y) \\ &= \int_{\mathbb{R}^s} \left| \sum_{j=1}^{N} c_j e^{i\langle x_i, y \rangle_{\mathbb{R}^s}} \right|^2 d\mu(y) \geq 0, \end{aligned} \tag{22}$$

the final inequality a result of integrating a non-negative function against a non-negative measure. □

As previously stated we are now able to easily prove when a function is a RKHS kernel, and we can therefore consider some concrete examples:

Example 5. When $X = \mathbb{R}^s$, the *Gaussian kernel* for fixed $\sigma > 0$ and $x' \in X$ is defined as:

$$\phi(x, x') = e^{\frac{-\|x-x'\|_{\mathbb{R}^s}^2}{\sigma^2}}. \tag{23}$$

To see that the Gaussian kernel indeed is a reproducing kernel, observe simply that $\phi(x, x') = k(x - x')$ where k is a standard (unnormalized) Gaussian. A standard calculation shows that the Fourier transform of k is given by $\hat{k}(\xi) = (\pi\sigma^2)^{s/2} e^{-\pi^2\sigma^2\|\xi\|^2}$, which is clearly a non-negative function, and is thus a density of a finite non-negative measure. Taking the inverse Fourier transform, Bochner's theorem then applies to show that k is positive definite and thus a reproducing kernel.

2.2.2 Mercer's theorem

We now turn to Mercer's theorem, which gives a useful characterization of any reproducing kernel K. A starting point is the so-called Hilbert-Schmidt operator of K. Suppose that μ is a Radon measure, and that K satisfies the boundedness property:

$$\int_{X\times X} |K(x,y)|^2 \, d(\mu \times \mu)(x,y) < \infty. \tag{24}$$

Then the operator:

$$I_K(f)(x) = \int_X K(x,y) f(y) \, d\mu(y) \tag{25}$$

is compact, self-adjoint, and positive on $f \in L^2(\mu)$. The spectral theorem then implies that there exists a (countable) set J, and positive eigenvalues $(\lambda_j)_{j\in J} \subseteq [0, \infty)$ with $\lambda_j \to 0$ for $j \to \infty$, and associated eigenfunctions $(\varphi_j)_{j\in J} \subseteq L^2(\mu)$ such that:

$$T_K \varphi_j = \lambda_j \varphi_j \tag{26}$$

for all $j \in J$. Mercer's theorem provides a much stronger characterization of the kernel K than the spectral theorem. Indeed, it gives an explicit formula for K in terms of the information already given. In particular it is also a direct generalization, since using this explicit formula also reproduces all the properties from the spectral theorem. We choose to present and prove Mercer's theorem for arbitrary Radon measures, a more general formulation.

Theorem 2.3 (Mercer)**.** *Let μ be a Radon measure, and suppose that $K : X \times X \to \mathbb{R}$ is continuous and symmetric on supp(μ) and satisfies:*

$$\int_{X\times X} K(x,y) f(x) \overline{f(y)} d(\mu \times \mu)(x,y) \geq 0 \tag{27}$$

for all $f \in L^2(\mu)$, as well as the boundedness property above. Let $(\lambda_j)_{j\in J}$ be the set of positive eigen-

values of I_K, and $(\varphi_j)_{j\in J} \in L^2(\mu)$ the corresponding eigenfunctions. Then we have (with uniform convergence) an equality:

$$K = \sum_{j\in J} \lambda_j \varphi_j \otimes \varphi_j. \tag{28}$$

Proof. We may freely assume that $J = \mathbb{N}$ since $L^2(\mu \times \mu)$ is separable (in the case where J is finite we just need to show equality). For arbitrary $f \in C(X)$ we consider:

$$I(f) = \int_{X\times X} \left(K(x,y) - \sum_{j=1}^{N} \lambda_j \varphi_j(x)\varphi_j(y) \right) \overline{f(x)} f(y) d(\mu \times \mu)(x,y). \tag{29}$$

Applying Fubini to change the order of integration, we recognize the spectral decomposition of the Hilbert Schmidt operator I_K and compute:

$$\begin{aligned} I(f) &= \int_X \left(\int_X K(x,y) f(y)\, d\mu(y) - \sum_{j=1}^{N} \int_X \lambda_j \varphi_j(x)\varphi_j(y) f(x)\, d\mu(y) \right) \overline{f(x)} d\mu(x) \\ &= \int_X \left(\int_X \sum_{j=N+1}^{\infty} \lambda_j \varphi_j(x) f(y)\, d\mu(y) \right) \overline{f(x)}\, d\mu(x) \\ &= \sum_{j=1}^{N+1} \lambda_j |\langle f, \lambda_j \rangle|^2 \geq 0. \end{aligned} \tag{30}$$

The positivity then holds for all $N \in \mathbb{N}$. We can now proceed with the equality: suppose for the sake of contradiction that there exists $x \in X$ such that:

$$K(x,x) < \sum_{j=1}^{\infty} \lambda_j \varphi_j(x)^2, \tag{31}$$

and pick $N \in \mathbb{N}$ such that the inequality holds for the finite sum:

$$K(x,x) < \sum_{j=1}^{N} \lambda_j \varphi_j(x)^2. \tag{32}$$

K is continuous, hence its eigenfunctions φ_j are also continuous. The above inequality can therefore be extended to an open neighborhood $x \in U \subseteq X$ in both variables. There exists a nonzero function $f \in C(X)$ such that $\text{supp}(f) = U$, but this means that $I(f) < 0$ which contradicts the previously established inequality on $I(f)$ and we therefore conclude for all $x \in X$ that:

$$K(x,x) \geq \sum_{j=1}^{\infty} \lambda_j \varphi_j(x)^2. \tag{33}$$

Now, suppose that the inequality is sharp for some $x \in X$, and let $\varepsilon > 0$ be such that:

$$K(x,x) > \varepsilon + \sum_{j=1}^{N} \lambda_j \varphi_j(x)^2, \tag{34}$$

for some $N \in \mathbb{N}$. Continuity again ensures that we can pick an open neighborhood $x \in U \subseteq X$ such that:

$$K(y,z) > \varepsilon + \sum_{j=1}^{N} \lambda_j \varphi_j(y)\varphi_j(z) \tag{35}$$

for all $y, z \in U$. Again there exists a nonzero function $f \in C(X)$ such that supp$(f) = U$, and we have that:

$$I(f) \geq \int_{U\times U} \varepsilon f(y) f(z) \, d(\mu \times \mu)(y,z) = \varepsilon \, \|f\|^2_{L^2(\mu\times\mu)} > 0. \tag{36}$$

However, this also contradicts the spectral theorem for the operator I_k, which ensures that $I(f) \to 0$ for $N \to \infty$. We there conclude that:

$$K(x,x) = \sum_{j=1}^{\infty} \lambda_j \varphi_j(x)^2 \tag{37}$$

for all $x \in X$. So, by continuity of the sum-function on the diagonal we know that the sum converges uniformly on the diagonal. A standard Cauchy-Schwarz argument then ensures that it thus converges uniformly on all of X, hence it has a continuous limit function that agrees with K as an integral kernel. Hence we have equality $(\mu \times \mu)-$a.e, but continuity again ensures that we have equality on all of X, i.e. that:

$$K(x,y) = \sum_{j=1}^{\infty} \lambda_j \varphi_j(x)\varphi_j(y) \tag{38}$$

for all $x, y \in X$. □

On the contrary, and Mercer expansion is positive definite, and hence also a kernel. Indeed, by orthogonality we have for any $x_1, \ldots, x_N \in X$ and $c_1, \ldots, c_N \in \mathbb{C}$ that:

$$\sum_{i,j=1}^{N} c_i \overline{c_j} K(x_i, x_j) = \sum_{j=1}^{\infty} \lambda_j \left(\sum_{i=1}^{N} c_i \varphi_j(x_i) \right)^2 \geq 0. \tag{39}$$

This gives us another way to classify reproducing kernels. From Mercer's theorem we also obtain another representation of our RKHS:

Corollary 2.4. *With notation as above, we have that:*

$$\mathcal{H}_K(X) = \left\{ \sum_{j=1}^{\infty} c_j \varphi_j \middle| c_i \in \mathbb{C}, \sum_{j=1}^{\infty} \frac{|c_j|^2}{\lambda_j} < \infty \right\}, \tag{40}$$

with norm given by:

$$\|f\| = \sqrt{\sum_{j=1}^{\infty} \frac{|c_j|^2}{\lambda_j}}. \tag{41}$$

Proof. Let H denote the space on the right-hand side of the equation above. We immediately see that:

$$K_x = \sum_{j=1}^{\infty} \lambda_j \lambda(x) \in H, \tag{42}$$

and furthermore we also have:

$$\langle K_x, K_y \rangle_H = \sum_{j=1}^{\infty} \frac{\lambda_j^2 \varphi_j(x)\varphi_j(y)}{\lambda_j} = \sum_{j=1}^{\infty} \lambda_j \varphi_j(x)\varphi_j(y) = K(x,y), \tag{43}$$

which means we exactly identify H as our RKHS. □

2.2.3 Linear operators in sufficiently smooth RKHS

We wish to understand how the reproducing kernel property is preserved under transformation by linear operators, such as differentiation and integration. This is necessary since to impose constraints on $\mathcal{H}_K$ we need to transform the kernel K. We begin with a theoretical result that gives a crude regularity bound on the RKHS:

Theorem 2.5. *Suppose that K is twice continuously differentiable. Then $\mathcal{H}_K(\mathbb{R}) \subseteq C^1(\mathbb{R})$ and the embedding is continuous.*

Proof. We must first show that $\partial_x K(x, x') \in \mathcal{H}_K(X)$ for any $x' \in X$. So, for $h \in X$ we consider the difference quotient of K and use the reproducing property to compute:

$$\begin{aligned}
\left\| \frac{K_{x+h} - K_x}{h} \right\|_{\mathcal{H}_K(\mathbb{R})}^2 &= \frac{1}{h^2}(\langle K_{x+h}, K_{x+h} \rangle_{\mathcal{H}_K(X)} - 2\langle K_x, K_{x+h} \rangle_{\mathcal{H}_K(X)} + \langle K_x, K_x \rangle_{\mathcal{H}_K(X)}) \\
&= \frac{1}{h^2}(K(x+h, x+h) - 2K(x+h, x) + K(x,x)) \\
&= \frac{1}{h^2}(h^2 \partial_x \partial_{x'} K(x,x) + o(h^2)) \\
&\to \partial_x \partial_{x'} K(x,x) < \infty,
\end{aligned} \tag{44}$$

where the third equality follows from a multi-variable Taylor expansion of the terms $K(x+h, x+h)$ and $K(x+h, x)$. By completeness of the Hilbert space, we get that the limit function for $h \to 0$ is in the RKHS, $g_x \in \mathcal{H}_K(X)$, and we have a point-wise equality $g_x(y) = \partial_x K(x,y)$. For an arbitrary $f \in \mathcal{H}_K(X)$, we may therefore conclude that the functional $f \mapsto \langle f, \partial_x^\alpha k(x, \cdot) \rangle_{\mathcal{H}_K}$ is bounded and linear, and we have that:

$$\partial^\alpha f(x) = \sum_{i=1}^n \alpha_i \partial_x^\alpha K(x, x_i) = \left\langle \sum_{i=1}^n \alpha_i K(\cdot, x_i), \partial_x^\alpha K(x, \cdot) \right\rangle_{\mathcal{H}_K} = \langle f, \partial_x^\alpha K(x, \cdot) \rangle_{\mathcal{H}_K}, \tag{45}$$

and the first equality means in particular that $\partial_x f \in \mathcal{H}_K(X)$. To see continuity in the inclusion, we first use the Cauchy-Schwarz inequality to obtain:

$$|\partial^\alpha f(x)| = |\langle f, \partial_x^\alpha k(x, \cdot)\rangle_{\mathcal{H}_K}| \leq \|f\|_{\mathcal{H}_K} \cdot \|\partial_x^\alpha K(x, \cdot)\|_{\mathcal{H}_K} =: C_x \|f\|_{\mathcal{H}_K}, \tag{46}$$

for all $x \in \Omega$. Since $\partial_x^\alpha K(x, \cdot)$ is locally uniformly bounded we have for all compact subsets $K \subseteq \Omega$ that:

$$\sup_{x \in K} |\partial^\alpha f(x)| \leq C \|f\|_{\mathcal{H}_K}. \tag{47}$$

Using the definition of the $\mathcal{C}^m$ norm we conclude that:

$$\|f\|_{\mathcal{C}^m(K)} = \sum_{\alpha \leq m} \sup_{x \in K} |\partial^\alpha f(x)| \leq C \|f\|_{\mathcal{H}_K}, \tag{48}$$

which concludes the proof. □

By induction we obtain the following result: Let D be a differential operator of order $\leq m$. Then $D_x K(x, x')$ is a reproducing kernel for the space $D\mathcal{H}_K(X) := \{Df \mid f \mid \mathcal{H}_K(X)\}$. Similarly, applying an integral operator to our kernel also results in a reproducing kernel for the corresponding image space.

In particular, both differential and integral operators therefore give rise to a bounded operator $L : \mathcal{H}_K(X) \to \mathcal{H}$ into *some* Hilbert space. We now explain how this allows us to transform the basis expansion to satisfy related constraints:

Theorem 2.6. *Let $L : \mathcal{H}_K(X) \to \mathcal{H}$ be a bounded linear operator, and define for $\alpha > 0$ an inner product:*

$$\langle f_1, f_2 \rangle_{\mathcal{H}_{K_\lambda}(X)} = \alpha \langle f_1, f_2 \rangle_{\mathcal{H}_K(X)} + \langle Lf_1, Lf_2 \rangle_{\mathcal{H}}, \tag{49}$$

for $f_1, f_2 \in \mathcal{H}_K(X)$. Then $(\mathcal{H}_K(X), \langle \cdot, \cdot \rangle_{\mathcal{H}_{K_\lambda}(X)})$ is a RKHS whose reproducing kernel is given by

$$K_\alpha(p, q) = [(\alpha + L^* L)^{-1} K_q](p), \tag{50}$$

for $p, q \in X$.

Proof. Since L is a bounded operator, the operator $\alpha + L^* L$ is then also bounded, self-adjoint, and coercive on $\mathcal{H}_K(X)$, and hence so is its inverse. This means that $(K_\alpha)_q \in \mathcal{H}_K(X)$ for all $q \in X$. We compute:

$$\begin{aligned}\langle (K_\alpha)_q, (K_\alpha)_p \rangle_{\mathcal{H}_{K_\alpha}(X)} &= \alpha \langle (K_\alpha)_q, (K_\alpha)_p \rangle_{\mathcal{H}_K(X)} + \langle L^* L[(K_\alpha)_q], (K_\alpha)_p \rangle_{\mathcal{H}_K(X)} \\ &= \langle (\alpha + L^* L) K_\alpha(\cdot, q), K_\alpha(\cdot, p) \rangle_{\mathcal{H}_K(X)} \\ &= \langle K_q, [(\alpha + L^* L)^{-1} K_p] \rangle_{\mathcal{H}_K(X)} \\ &= \langle [(\alpha + L^* L)^{-1} K_q], K_p \rangle_{\mathcal{H}_K(X)} \\ &= [(\alpha + L^* L)^{-1} K_q](p) = K_\alpha(p, q),\end{aligned} \tag{51}$$

which concludes the proof. □

So, the set of functions in the Hilbert space $\mathcal{H}_K(X)$ is uniquely determined from both K and K_α, but with different norms. In particular this means that that any element $f \in \mathcal{H}_K(X)$ can be written as a basis expansion from both kernels in a point-wise sense. $L^* L$ may represent the Gram matrix of a linear system of equations, and as such we can construct a new reproducing kernel for our space of functions that is explicitly defined to satisfy this system.

3 Methodology

3.1 Model and estimator in PDE approximation setting

In this section we describe our methodology, formulating the two different types of minimization problems that we consider, and then how to implement the appropriate physics-informed kernel transformations that we will use as basis functions.

There are essentially two different ways to solve use RKHS kernel expansions to solve a PDE in a data-driven context. The first method is to simply optimize the parameter matrix α so that the kernel expansion is best-fit to trajectory data that we assume fulfills some general time-dependent BVP (10). Let $X = \Omega \times [0, T]$ be our background set, and let K be some reproducing kernel for $\mathcal{H}_K(X)$. Suppose that $((t_0, d_0), \dots, (t_N, d_N)) \in \mathbb{R}_{\geq 0}^{N+1} \times L^2(\Omega)^{N+1}$ are temporal slices of the solution to (10). Given $u \in \mathcal{H}_K(X)$, we can then define the discretized empirical risk:

$$R(u) := \frac{1}{N+1} \sum_{i=0}^{N} \|u(\cdot, t_i) - d_i\|_{L^2(\Omega)}^2 , \tag{52}$$

as well as the penalized empirical risk $R_\lambda(u) = R(u) + \lambda \|u\|_{\mathcal{H}_K(X)}^2$.

Our second method is to directly solve the BVP using the kernel expansion and optimization approach, i.e. we consider a residual based empirical risk for $u \in \mathcal{H}_K(X)$ given by:

$$R(u) := \left\|\partial_t u - F(D^2 u, Du, u, \cdot)\right\|_{L^2}^2 + \|G(Du, u, \cdot)\|_{L^2}^2 + \|u(\cdot, 0) - u_0\|_{L^2}^2 . \tag{53}$$

We may also modify this risk with a tuple $(\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}_{>0}^3$ of weights for each of the components in this loss. These values are usually decided experimentally. We also consider the penalized empirical risk:

$$R_\lambda(u) = R(u) + \lambda \|u\|_{\mathcal{H}_k}^2 , \tag{54}$$

which is a strictly convex objective, meaning that there exists a unique minimizer (see Appendix B). In practice, we will minimize over a discrete set of points of $\Omega \times [0, T]$. Let $\mathcal{X}_{bas} \subseteq \Omega$ and $\mathcal{T}_{bas} \subseteq [0, T]$ be such discrete spatial and temporal points, and let $X = \mathcal{X}_{bas} \times \mathcal{T}_{bas}$. In this case, the continuous norms above are replaced by discrete sums, and by (20) we have that any $u \in \mathcal{H}_K(X)$ is given by:

$$u(x, t) = u_\alpha(x, t) = \sum_{\substack{x_i \in \mathcal{X}_{bas} \\ t_j \in \mathcal{T}_{bas}}} \alpha_{ij} K((x, t), (x_i, t_j)), \tag{55}$$

where $\alpha_{ij} \in \mathbb{R}$ for all i and j. Then the RKHS minimization problem (which has a unique solution) is equivalent to finding:

$$\widehat{\alpha} = \underset{\alpha \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{T}|}}{\arg\min} \; R_\lambda(u_\alpha). \tag{56}$$

Our methodology therefore works on any set of discretized mesh points, and finding the optimal matrix of coefficients $\alpha = (\alpha_{ij})_{ij}$ can be accomplished with any optimization procedure.

3.2 Kernel transformations

We now explain how to define and explicitly compute kernel transformations that force the image kernels to obey conservation laws such as (13) and (15), which may be implemented to solve the minimization problems from the previous subsection. We consider both discrete and continuous variants.

3.2.1 Continuous kernel derivative centering

Let $\Omega \subseteq \mathbb{R}^n$ be compact and consider the mass-conservation constraint (13) on time-dependent functions $u(x,t) : \Omega \times [0,\infty) \to \mathbb{R}$:

$$\int_\Omega u(x,t)\,dx = \int_\Omega u(x,0)\,dx, \tag{57}$$

or equivalently by defining:

$$MB(u) := \partial_t \int_\Omega u(x,t)\,dx, \tag{58}$$

that we require $MB(u) = 0$. Suppose that $u \in \mathcal{H}(= \mathcal{H}_K(\Omega))$ for some reproducing kernel K. In this scenario, u is a linear kernel expansion with K as base kernel, and since $MB(u_1 + u_2) = MB(u_1) + MB(u_2)$, the constraint on u will therefore automatically hold if $MB(K(x,\cdot)) = 0$. Imposing this on K is equivalent to projecting into the orthogonal subspace of:

$$S = \text{span}(K_{(x',t')} \mid MB(K_{(x',t')})\} \subseteq \mathcal{H}_K(X). \tag{59}$$

which in turn is equivalent to centering the kernel derivative $\partial_t K$. For the mass-conservation constraint, we do this by first defining for all $(x',t') \in \Omega \times [0,\infty)$

$$K_1(x, x't, t') = \partial_t K(x, x', t, t') - \frac{1}{|\Omega|}\int_\Omega K(z, x', t, t')\,dz, \tag{60}$$

then integrating to define the new kernel:

$$\widetilde{K}(x, x', t, t') = \int_0^t K_1(x, x', s, t')\,ds. \tag{61}$$

This is a combination of integral and differential operators applied to our kernel, so with sufficient regularity assumptions it results in a new valid kernel. We readily verify that:

$$\begin{aligned}\int_\Omega \partial_t \widetilde{K}(x, x', t, t')\,dx &= \int_\Omega K_1(x, x', t, t')\,dx \\ &= \int_\Omega \partial_t K(x, x', t, t') - \frac{1}{\Omega}\int_\Omega \partial_t K(z, x', t, t')\,dz\,dx \\ &= 0.\end{aligned} \tag{62}$$

However, this particular approach has an immediate drawback. Applying Fubini's theorem to the double integral:

$$\begin{aligned}\int_\Omega \widetilde{K}(x,x',t,t')\,dx &= \int_\Omega \int_0^t K_1(x,x',s,t')\,ds\,dx \\ &= \int_0^t \int_\Omega K_1(x,x's,t')\,dx\,ds = 0,\end{aligned} \tag{63}$$

means that the orthogonal subspace of S is not only the mass-conserving functions of $\mathcal{H}$, but also the functions with zero total mass over Ω. To remedy this still obey (13), and still have the minimization problem take place be formulated in terms of the RKHS norm, we consider an affine transformation of the Hilbert space:

$$\mathcal{H}_{\widetilde{K}(X)} + m_0 := \{f + m_0 \mid f \in \mathcal{H}_{\widetilde{K}(X)}\}. \tag{64}$$

This method turns out to not yield good results, so we also consider a discrete version in the next subsection.

3.2.2 A discrete variation of mass-balance constraint

Let $\mathcal{T}_{con} = \{t_0, \ldots, t_N\} \subseteq [0,\infty)$ be a set of time-points. We now seek to impose the mass-balance constraint on $u(x,t) : \Omega \times [0,\infty) \to \mathbb{R}$ in a point-wise manner:

$$L_{t_a}[u] := \int_\Omega u(x,t_a)\,dx - \int_\Omega u(x,t_0)\,dx = 0, \quad \forall t_a \in \mathcal{T}_{con} \tag{65}$$

Similarly to before, let K be our (base) reproducing kernel and define for $z' = (x',t') \in \Omega \times [0,\infty)$ the function:

$$Q(z') := (L_{t_0}K[\cdot,z'], \ldots, L_{t_N}K[\cdot,z'])^T \in \mathbb{R}^{N+1}, \tag{66}$$

i.e. the combined vector of the constraint functionals. Note that by the Riesz representation theorem there exists for each $a \in \mathcal{T}_{con}$ a unique $q_a \in \mathcal{H}$ such that $L_{t_a}(f) = \langle f, q_a \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$. The operator $L : \mathcal{H} \to \mathbb{R}^{N+1}$ defined by $f \mapsto (L_{t_0}[f], \ldots, L_{t_N}[f])^T$. Its adjoint operator $L^* : \mathbb{R}^{N+1} \to \mathcal{H}$ is by definition represented as:

$$c \mapsto \sum_{i=0}^{N} c_i q_i. \tag{67}$$

We can then define the Gram matrix $A = LL^*$, which for our mass-balance functional expands to:

$$\begin{aligned}A_{ab} &= L_{t_a}[q_b] \\ &= L_a L_b K \\ &= \int\int_{\Omega\times\Omega} [k(x,t_a),(x',t_b)) - k(x,t_a),(x',0)) - k((x,0),(x',t_b)) + k((x,0),(x',0))]\,dx\,dx'.\end{aligned} \tag{68}$$

The kernel symmetry $K(z,z') = K(z',z)$ implies that we may write $A = Q(\mathcal{X}_{bas}, \mathcal{T}_{bas}) \cdot Q(\mathcal{X}_{bas}, \mathcal{T}_{bas})^T$. We may now define the orthogonal projection onto the kernel ker$L \subseteq \mathcal{H}$ in the RKHS, $\widetilde{K}(z,z') =$

$K(z, z') - Q(z)^T A^{-1} Q(z')$. By construction we have for any $z' \in \Omega \times [0, \infty)$ that:

$$\begin{aligned} L_t[\widetilde{K}(\cdot, z')] &= L[K(\cdot, z')] - L[Q^T(\cdot) A^{-1} Q(z')] \\ &= Q(z') - L[Q(\cdot)]^T A^{-1} Q(z') \\ &= Q(z') - A A^{-1} Q(z') \\ &= 0 \end{aligned} \tag{69}$$

where the first equality used linearity of L, and the second and third used the reproducing property $A_{ab} = L_{t_a}[q_b]$. So, by linearity again we conclude that any kernel expansion of $\widetilde{K}$:

$$f(x, t) = \sum_{i,j} \alpha_{ij} \widetilde{K}(x, t, x_i, t_j) \tag{70}$$

satisfies the mass conservation property at all time-steps $t_a \in \mathcal{T}_{con}$. A big benefit of this method is that all the components used in defining the transformed kernel are relatively easy to explicitly compute for many common base kernel choices.

Example 6 (Heat equation conservation law)**.** We consider the heat equation boundary value problem from before (11), where our domain is $\Omega = [-1, 1]$. If our base kernel is the Gaussian kernel $K(z, z') = K((x, t), (x', t')) = k_x(x, x') k_t(t, t')$, then to derive the transformed kernel $\widetilde{K}$, we first select a set of temporal basis points $\mathcal{T}_{bas}$. A small computation yields:

$$\begin{aligned} Q_a(x, t) &= \int_{[-1,1]} K(y, t_a, x, t)\, dy - \int_{[-1,1]} K(y, t_0, x, t)\, dy \\ &= \left(e^{-\frac{(t_a - t)^2}{2\sigma_t^2}} - e^{-\frac{(t_0 - t)^2}{2\sigma_t^2}} \right) \int_{[-1,1]} e^{-\frac{(y-x)^2}{2\sigma_x^2}}\, dy \\ &= \left(e^{-\frac{(t_a - t)^2}{2\sigma_t^2}} - e^{-\frac{(t_0 - t)^2}{2\sigma_t^2}} \right) \sqrt{\frac{\pi}{2}} \sigma_x \left(\text{erf}\left(\frac{1 - x}{\sqrt{2}\sigma_x} \right) - \text{erf}\left(\frac{-1 - x}{\sqrt{2}\sigma_x} \right) \right), \end{aligned} \tag{71}$$

where we recognized the integral symmetry around x, and used the error function. Since we also need the transformed kernel derivatives we also compute:

$$\partial_x Q_a(x, t) = \left(e^{-\frac{(t_a - t)^2}{2\sigma_t^2}} - e^{-\frac{(t_0 - t)^2}{2\sigma_t^2}} \right) \left(e^{-\frac{(1+x)^2}{2\sigma_x^2}} - e^{-\frac{(1-x)^2}{2\sigma_x^2}} \right), \tag{72}$$

using the fact that $\text{erf}'(y) = (2/\sqrt{\pi}) e^{-y^2}$. Additionally:

$$\partial_{xx} Q_a(x, t) = \partial_x Q_a(x, t) = \left(e^{-\frac{(t_a - t)^2}{2\sigma_t^2}} - e^{-\frac{(t_0 - t)^2}{2\sigma_t^2}} \right) \left(-\frac{1 + x}{\sigma_x^2} e^{-\frac{(1+x)^2}{2\sigma_x^2}} - \frac{1 - x}{\sigma_x^2} e^{-\frac{(1-x)^2}{2\sigma_x^2}} \right), \tag{73}$$

and

$$\partial_t Q_a(x,t) = \left(-\frac{t_a - t}{\sigma_t^2} e^{-\frac{(t_a-t)^2}{2\sigma_t^2}} - \frac{t_0 - t}{\sigma_t^2} e^{-\frac{(t_0-t)^2}{2\sigma_t^2}} \right) \sqrt{\frac{\pi}{2}} \sigma_x \left(\text{erf}\left(\frac{1-x}{\sqrt{2}\sigma_x}\right) - \text{erf}\left(\frac{-1-x}{\sqrt{2}\sigma_x}\right) \right). \quad (74)$$

With these formulae we can easily compute the transformed kernel derivatives, i.e. $\partial_x \widetilde{K}(z,z') = \partial_x K(z,z') - \partial_x Q(z)^T A^{-1} Q(z')$, and likewise for the other derivatives, since A and $Q(z')$ both do not depend on x or t.

3.2.3 Linearization of nonlinear conservation law

We now proceed with a strategy for producing a similar kernel for the nonlinear mass-balance equation (15) for the Richards equation. With the discrete set of time-points $t_a \in \mathcal{T}_{con}$, we may again formulate it as a list of operator constraints:

$$N_{t_a}[\theta] := \int_{[0,L]} \theta(x,t_a)\,dx - \int_{[0,L]} \theta(x,t_0)\,dx - \int_{[0,t_a]} (q(\theta)(L,t) + q(\theta)(0,t)\,dt = 0, \quad \forall t_a \in \mathcal{T}_{con}. \quad (75)$$

The issue here is that $N_{t_a}[\theta]$ is a nonlinear (integro-differential) operator in θ, so the Riesz representation theorem does not apply. Our solution is to linearize N_{t_a} (for all $t_a \in \mathcal{T}_{con}$). We introduce the Kirchhoff transform:

$$u(\theta) = \int_{\theta_r}^{\theta} K(s) \frac{dh}{ds}\,ds, \quad (76)$$

which allows us to reformulate the Richards equation as $\partial_t \theta = \partial_{xx} u - \partial_x K(\theta)$. Integrating the equation on the spatial domain $[0,L]$ gives:

$$\frac{d}{dt} M(t) = q^*(0,t) - q^*(L,t), \quad (77)$$

where $q^*(x,t) = -\partial_x u(\theta(x,t))$ and

$$M(t) = \int_0^L \theta(x,t)\,dx. \quad (78)$$

Integrating this equation again on the spatial domain $[t_0, t_a] = [0, t_a]$ gives:

$$M(t_a) - M(t_0) + \int_0^{t_a} \partial_x u(L,t) - \partial_x(0,t)\,dt = 0, \quad (79)$$

which is exactly the conservation law, now linear in θ in the first term(s), and linear in u in the second term. To obtain a linear conservation law, we therefore assume a linear relation $\theta = \alpha u + \beta$, where $\alpha = (\theta_s - \theta_r)/u_s$ and $\beta = \theta_r$. This essentially means that we have averaged the nonlinarity $K(\theta)$ out of the equation. Our final linear conservation law is then given by:

$$L_{t_a}[\theta] = \int_{[0,L]} \theta(x,t_a) - \theta(x,t_0)\,dx - \frac{1}{\alpha}\int_{[t_0,t_a]} \partial_x\theta(L,t) - \partial_x\theta(0,t)\,dt. \tag{80}$$

The construction is then identical to before; we have that the transformed kernel is given by:

$$\widetilde{K}(z,z') = K(z,z') - Q(z)^T A^{-1} Q(z'), \tag{81}$$

where $Q_a(z') = L_{t_a}[K(\cdot,z')]$ and $A = Q(\mathcal{X}_{bas}, \mathcal{T}_{bas}) \cdot Q(\mathcal{X}_{bas}, \mathcal{T}_{bas})^T$.

A first order Taylor expansion of q about $\theta(x,t) - \theta_s$ yields that:

$$q(\theta(x,t)) = q(\theta_s) + q'(\theta_s)(\theta(x,t) - \theta_s) + q''(\xi)(\theta(x,t) - \theta_s)^2/2, \tag{82}$$

for some $\xi = \xi(x,t) \in [\theta(x,t), \theta_s]$. Using this identity along with the definition of the linearized operator, we compute for any t_a:

$$\begin{aligned}
|N_{t_a}[\theta] - L_{t_a}[\theta]| &= \int_{[0,t_a]} \frac{1}{\alpha}(\partial_x\theta(L,t) - \partial_x\theta(0,t)) - (q(\theta)(L,t) - q(\theta)(0,t))\,dt \\
&\leq \int_{[0,t_a]} |\frac{1}{2}q''(\xi)(\theta(L,t) - \theta_s)^2 - \frac{1}{2}q''(\xi)(\theta(0,t) - \theta_s)^2|\,dt \\
&\leq \sup_{\xi\in[\theta_r,\theta_s]} |q''(\xi)| \max_{(x,t)} |\theta(x,t) - \theta_s|^2 t_a \\
&\leq \sup_{\xi\in[\theta_r,\theta_s]} |q''(\xi)||\theta_r - \theta_s|^2 t_a.
\end{aligned} \tag{83}$$

The above supremum is bounded and computable for most constitutive relations. We therefore conclude that the error between the nonlinear and linearized conservation laws must be linear in time.

Deriving the transformed kernel for the linearized conservation law (81) is similar to the heat equation example. We also again use the Gaussian kernel as our base kernel. The transformed kernel is then given by $\widetilde{K}(z,z') = K(z,z') - Q(z)^T A^{-1} Q(z')$, where A is the gram matrix and the coordinates $Q_a(x,t) = I_1 - \frac{1}{\alpha} I_2$, where:

$$\begin{aligned}
I_1 &= \int_{[0,L]} K(y,t_a,x,t) - K(y,t_0,x,t)\,dy \\
&= \left(e^{-\frac{(t_a-t)^2}{2\sigma_t^2}} - e^{-\frac{(t_0-t)^2}{2\sigma_t^2}}\right)\sqrt{\frac{\pi}{2}}\sigma_x\left(\operatorname{erf}\left(\frac{L-x}{\sqrt{2}\sigma_x}\right) - \operatorname{erf}\left(\frac{-x}{\sqrt{2}\sigma_x}\right)\right),
\end{aligned} \tag{84}$$

and

$$
\begin{aligned}
I_2 &= \int_{[t_0,t_a]} e^{-\frac{(s-t)^2}{2\sigma_t^2}}\, ds \left(\frac{x-L}{2\sigma_x^2} e^{-\frac{(L-x)^2}{2\sigma_x^2}} - \frac{x}{2\sigma_x^2} e^{-\frac{x^2}{2\sigma_x^2}} \right) \\
&= \sqrt{\frac{\pi}{2}}\sigma_t \left(\operatorname{erf}\left(\frac{t-t_a}{\sqrt{2}\sigma_t} \right) - \operatorname{erf}\left(\frac{t-t_0}{\sqrt{2}\sigma_t} \right) \right) \left(\frac{x-L}{2\sigma_x^2} e^{-\frac{(L-x)^2}{2\sigma_x^2}} - \frac{x}{2\sigma_x^2} e^{-\frac{x^2}{2\sigma_x^2}} \right).
\end{aligned} \tag{85}
$$

We omit the intermediate calculations, since they are almost identical to the heat equation example. The derivatives are also computed similarly, and are omitted.

4 Experiments and results

In this section we describe experiments to evaluate our transformed kernel in terms of its estimation abilities. This takes the form of two different comparisons with a baseline (pre-transformation) kernel, in two different estimation settings; one is a residual-based loss, and the other is based on pre-generated reference data. The code to run these experiments is written in 4 `Python` notebooks that each define the different kernel classes, confirm their functionality, run the optimization procedures, and finally generates various plots and tables summarizing the resulting information. The notebooks are found on `https://github.com/valdemarskou/`.

4.1 Heat equation residual learning

To evaluate the transformed kernel for the linear conservation law (13), we solve the full heat equation BVP (based on (56) and compare with a baseline (Gaussian) kernel $K(z, z') = K((x, t), (x', t')) = k_x(x, x')k_t(t, t')$, where:

$$k_x(x, x') = e^{-\frac{(x-x')^2}{2\sigma_x^2}}, \quad k_t(t, t') = e^{-\frac{(t-t')^2}{2\sigma_t^2}}, \tag{86}$$

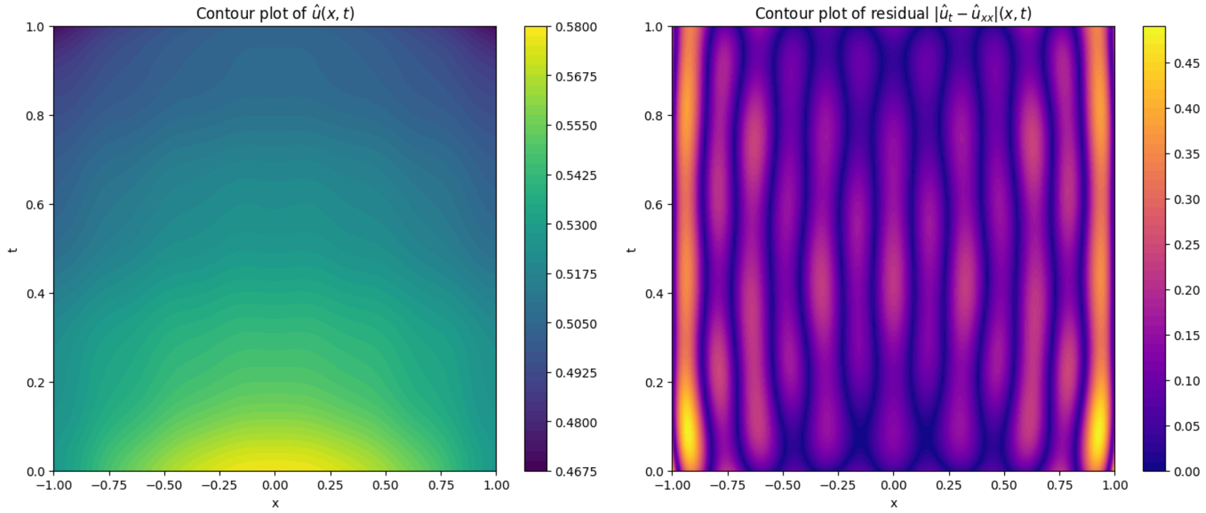
A detailed description and initialization process of the `Python` implementations is found in within the respective notebooks. Here we mention only that a set of spatiotemporal basis points $(\mathcal{X}_{bas}, \mathcal{T}_{bas})$ is required to define the kernel class and formulate the minimization problem in our solver.

We run the solver with uniform basis points $\mathcal{X}_{bas} = (x_i)_{i=0}^N$ with $x_i = x_0 + i\Delta x = -1 + i\Delta x$ and $\Delta x = 2/N$, and $\mathcal{T}_{bas} = (t_j)_{j=0}^M$ with $t_j = t_0 + j\Delta t = j\Delta t$ and $\Delta t = T/M$ for various values of N and M. We evaluate the solution qualitatively by its contour plot, seeing its ability to capture the expected behavior of the PDE, but also quantitatively by computing the PDE residual on a denser grid and plotting the conservation law defect over time.

The solution method is also evaluated in practical terms, namely by studying if convergence is obtained within a certain number of maximum iterations, and by recording the computation times in order to reach a stopping point.

4.1.1 Results

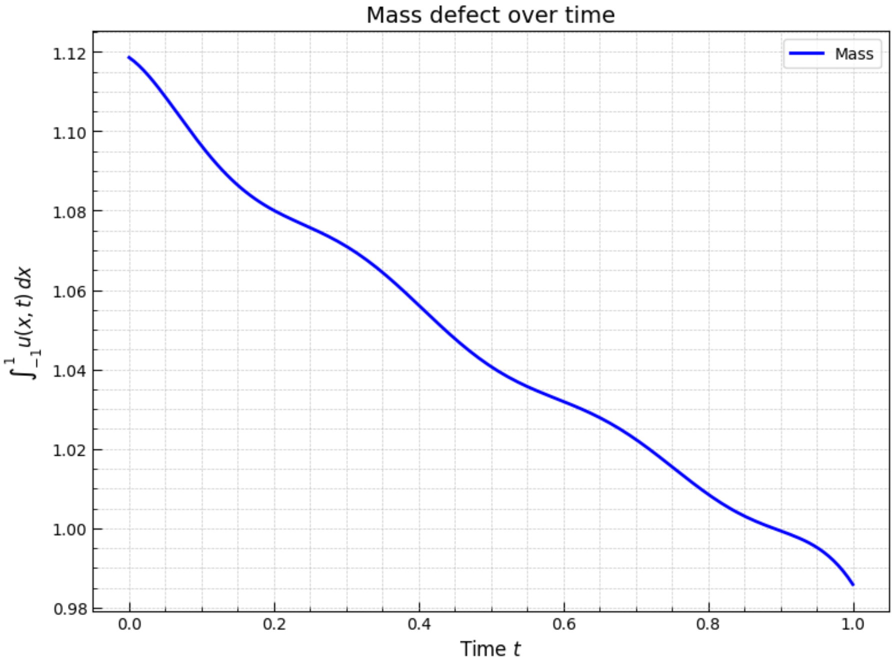
We consider an initial condition of the form $u_0(x) = \sin(\pi(x + 1)/2)$. A visualization of the baseline computed solution $\hat{u}(x, t)$ with $N = M = 20$ to the BVP is shown below, along with the PDE residual. The optimization step is done using the `L-BFGS-B` optimizer from the `Python-SciPy` library, with 10000 as the maximum number of iterations, and a gradient norm tolerance set to 10^{-10}.



The optimizer converges by this criteria after 5397 iterations. We immediately see a poor result from the contour plot: the optimizer has converged to a local minimum that does not correspond to the true solution to the BVP. Indeed, our theoretical initial condition u_0 attains values close to 1 near $x = 0$, yet $\hat{\theta}(0, x)$ is never above 0.6. This is verified by computing the theoretical total mass of u_0:

$$\int_{[-1,1]} u_0(x)\, dx \approx 1.27, \tag{87}$$

while our the total mass of $\hat{u}(x, 0)$ over the spatial domain is significantly lower (see the next plot). While the optimizer is unable to obtain the correct solution, we may still study the physical validity of its output. The residual plot above shows larger errors near the spatial boundary points. This indicates that the learned solution is not strongly enforcing the Neumann boundary condition. Since the Neumann condition is exactly related to the mass conservation property, we would expect there to be some inaccuracy on this end as well. Indeed, the plotted solution appears visually to not exactly conserve mass. A plot of the total mass over time confirms this suspicion:



A clear loss in the total mass of the solution implies that the solution is not physically valid either. To remedy this issue we may change the weights in the loss function to place emphasis on the spatial boundary term. We may likewise change the weights on the initial condition term, in an attempt to obtain convergence towards the correct solution. However, this this results in the learned function $\hat{\theta}(x,t)$ becoming more level in its initial heat distribution, and then slowly dissipating (in other words, it forces fast convergence to an even worse solution).

For completeness sake we give a partial overview of the baseline solver performance for varying amounts of basis points and maximum iterations allowed in the optimization step (with same convergence criteria as before). The final loss and convergence information is shown in the following table:

Final (L^2-based) loss and convergence success (y/n)

	$N = M = 10$	$N = M = 20$	$N = M = 30$
$max\_iter = 1000$	0.130 (n)	0.181 (n)	0.978 (n)
$max\_iter = 3000$	0.130 (y)	0.139 (n)	0.673 (n)
$max\_iter = 10000$	0.130 (y)	0.135 (y)	0.510 (n)

The values in the table are not very meaningful, since an incorrect local minimum is achieved which means that some nonzero lower bound will never be crossed. When the number of spatiotemporal basis points are higher, the optimizer naturally requires more iterations in order to achieve convergence. The final loss is worse for higher values of M and N due to stacking: A 0.510 loss across 30×30 mesh points is better than a 0.130 loss across 10×10 points. We might therefore also look at the loss in terms of the

L^∞-norm, however it is not very interesting for the baseline results.

We investigate if the number of basis points affect the final mass loss by recording the quantity:

$$\left| \int_{[-1,1]} u(x,T)\,dx - \int_{[-1,1]} u_0(x)\,dx \right|, \tag{88}$$

for same basis points and maximum number of iterations as the previous table (the above value can be computed analytically for any parameter matrix α):

Mass lost from $t = 0$ to $t = T$

	$N = M = 10$	$N = M = 20$	$N = M = 30$
$max\_iter = 1000$	0.159	0.697	0.978
$max\_iter = 3000$	0.154	0.303	0.673
$max\_iter = 10000$	0.154	0.285	0.510

Similar to before, the values reflect the existence of a lower bound due to the incorrect local minimum. It is however interesting that the conservation law defect becomes worse when considering higher number of basis points, since this error is no longer due to stacking. This most likely suggests that the optimizer is incorrectly moving further away from the true solution and instead towards the incorrect local minimum.

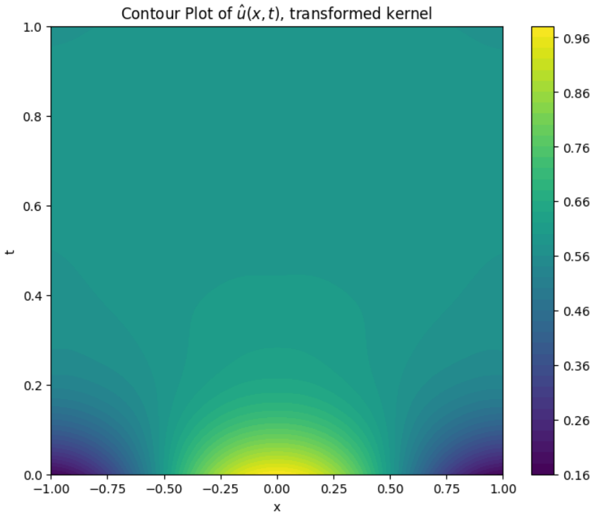
Finally, we consider the computation times for the simulations summarized above. This is important since it directly influences the efficacy of our implementation in a practical setting.

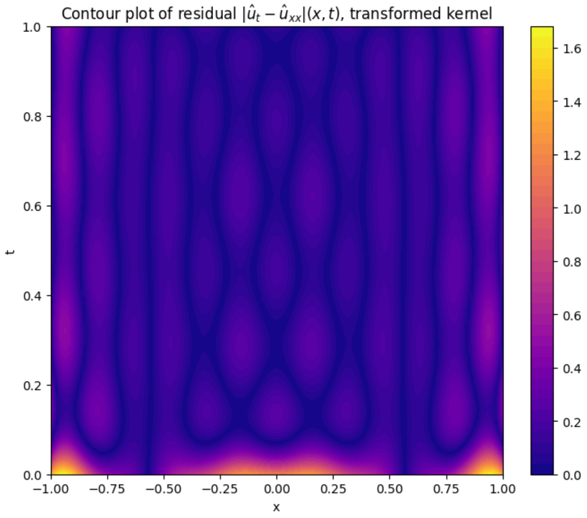
Computation time to obtain optimized weight matrix α (s)

	$N = M = 10$	$N = M = 20$	$N = M = 30$
$max\_iter = 1000$	97	113	125
$max\_iter = 3000$	115	339	380
$max\_iter = 10000$	115	610	1261

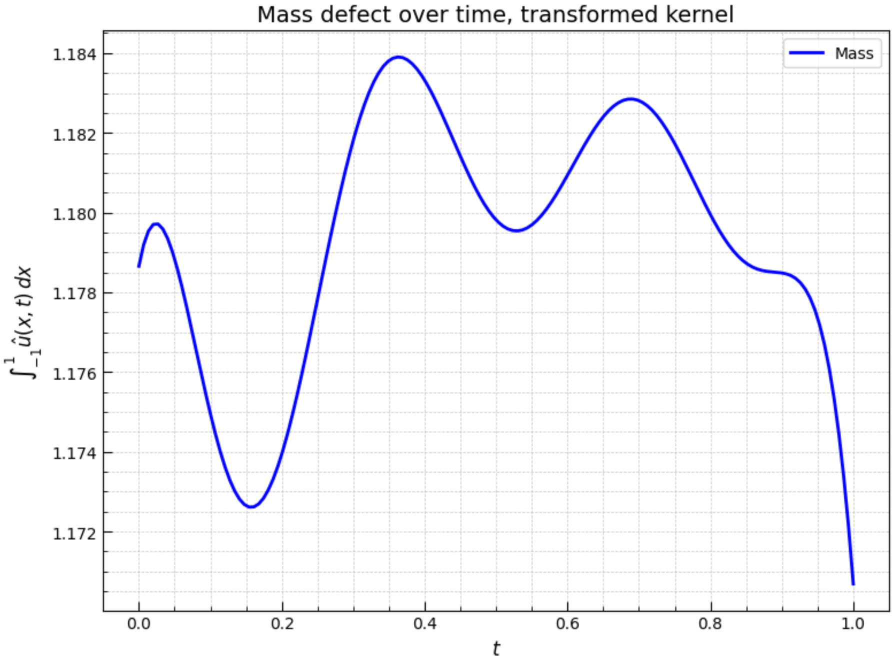
This behavior is somewhat expected, as both higher number of maximum iterations (as long as convergence is not reached early), and a higher number of parameters in the matrix α will increase computation times.

We now present the same results for the transformed kernel $\widetilde{K}$, starting with the contour plot of the contour plot of $\hat{u}(x,t)$ and its residual when $N = M = 20$:





The optimizer does not converge after 10000 iterations, so the above does not represent a true minimum. We do however see that the optimizer has converged towards the true solution to the BVP. The initial condition appears almost correct, and the behavior over time is what we theoretically except: convergence to steady state temperature. The residual error is overall lower than the baseline, but has very large values near the boundary with $t = 0$: the approximate function does not accurately capture the initial heat flow. However, the residual error along the spatial boundary is much smaller than the baseline, which indicates that conservation law transformation has had a positive effect on properly capturing the physical behavior. A plot of the total mass over time immediately confirms that our approximate solution is much improved from the baseline:



The above plot is to be expected since u is constructed to conserve mass exactly at the temporal basis points. However, it has indirectly enforced the Neumann condition, and thus created a more physically valid solution. The error bottleneck at the temporal boundary $t = 0$ can possibly be manually corrected by either forcing the parameters in α to exactly adhere to the initial condition, or by changing the weights in the loss function to place more emphasis on the initial term. (note, however, that the fact of no such weights being placed on the spatial boundary conditions suggests that we have an inherent improvement by virtue of our kernel choice). Putting such a weight on the initial condition term in the loss function does indeed yield some partial improvement. Running the optimization with $N = M = 20$ again, we record the quantities:

$$\int_{[-1,1]} \hat{u}(x,0) - u_0(x)\,dx \approx 0.0182, \quad \int_{[-1,1]} \hat{u}(x,T) - u_0(x)\,dx \approx 0.0231, \tag{89}$$

indicating that $\hat{u}(x,t)$ better captures the initial condition, and that the mass defect over time is still minimal. A notable drawback seems to be that the residual error increases from this change: we record from the above experiment, a value $\|\partial_t \hat{u} - \partial_{xx}\hat{u}\|_{L^\infty(X)} \approx 2.6$, whereas using the unchanged weights this value was closer to 1.6 (see the previous residual contour plot). As such, we shall stay with the unchanged weights, and again summarize some of the properties of the optimized approximate solution for different mesh sizes and optimization iterations:

Final (L^2-based) loss and convergence success (y/n)

	$N = M = 10$	$N = M = 20$	$N = M = 30$
$max\_iter = 1000$	0.0499 (n)	0.078 (n)	0.288 (n)
$max\_iter = 3000$	0.0295 (n)	0.0510 (n)	0.114 (n)
$max\_iter = 10000$	0.0275 (y)	0.0337 (n)	-

An additional experiment revealed that for $N = M = 20$, the optimizer converged within ≈ 13000 iterations. We do not obtain the result for $N = M = 30$ and $max\_iter = 10000$ due to the computational cost. The total mass defect and computation times for each experiment is summarized in the following tables:

Mass lost from $t = 0$ to $t = T$

	$N = M = 10$	$N = M = 20$	$N = M = 30$
$max\_iter = 1000$	0.0884	0.206	0.708
$max\_iter = 3000$	0.0516	0.122	0.401
$max\_iter = 10000$	0.0427	0.109	-

Computation time to obtain optimized weight matrix α (s)

	$N = M = 10$	$N = M = 20$	$N = M = 30$
$max\_iter = 1000$	53	120	1020
$max\_iter = 3000$	120	360	3420
$max\_iter = 10000$	240	1320	-

The mass defect is markedly improved from the baseline experiments, in terms of absolute value but more importantly also in terms of stability. As we saw in the previous plot, the transformed kernel has almost no mass defect: the errors in the table thus represent the errors near the initial condition. The larger mass defect present with $N = M = 30$ is due to an overall inaccuracy in the learned solution: A very high number of iterations seems required to reach convergence.

This leads us to the computation time results, which we see is a lot higher than the baseline experiments, especially at higher values of N and M. In general it seems to take twice as long to reach convergence for lower values of N and M. It is hard to meaningfully compare directly with the baseline example (since it only reached local minima), but we can conclude that there is a significant increase in computational complexity (dependent on N and M) for learning the optimal parameter matrix α.

Since the implementation is similar to [11] (where we in particular have closed formulas for all derivatives), we would expect the performance to be similar. The definition of the transformed kernel implies a higher number of matrix multiplications needed to formulate the loss function, but since basis points are fixed, and the training grid is also fixed, everything except for the final assembling with a new parameter matrix α could be pre-computed. Partial steps were taken in this direction during implementation, but these results suggests and error or omission in this avenue.

We discuss the implications of these considerations in the next chapter.

4.2 Richards Equation with reference trajectory data

We now study the transformed kernel for the nonlinear Richards equation, with a different methodology than the heat equation. The nonlinearity involved in the Richards equation means that the residual-based loss function does not produce good results. As such, to evaluate whether the transformed kernel produces results that are more physically valid, we shall compare to a baseline implementation its ability to learn a PDE trajectory from pre-computed reference data. We first recall the one-dimensional BVP:

$$\begin{cases} \partial_t\theta - \partial_x(K(\theta)h'(\theta)\partial_x\theta + K(\theta)) = 0, & (x,t) \in (0,L) \times (0,T] \\ u(0,t) = a,\, u(L,t) = b & t \in (0,T] \\ u(x,0) = u_0(x), & x \in [0,L] \end{cases} . \tag{90}$$

In order to generate the reference trajectory, the BVP must first be a well-defined problem; specifically, we need to specialize the constitutive relations, and consider the widely-used Van Genuchten curve (see [4]):

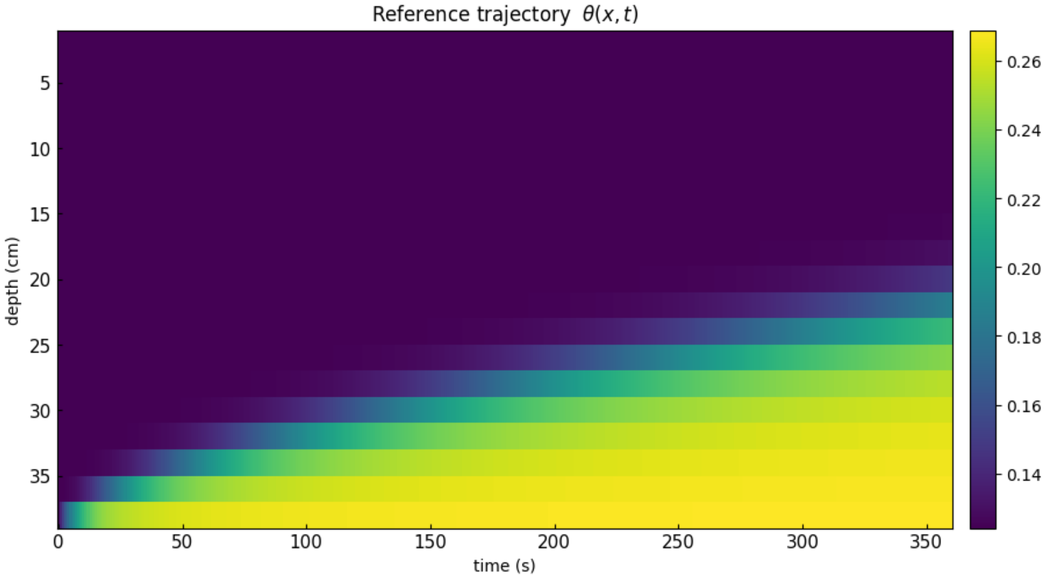
$$\theta(h) = \theta_r + (\theta_s - \theta_r)\frac{\alpha}{\alpha + |h|^\beta}. \tag{91}$$

We briefly summarize the significance and explicit values (with units) of each of the parameters, although this is not quite necessary for the actual implementation. The residual water content $\theta_r = 0.075$ and the saturated water content $\theta_s = 0.287$ bounds the moisture content in the domain. $\alpha = 1.611 \times 10^6$ is related to the air-entry pressure, and $\beta = 3.96$ is related to the pore-size distribution. In particular, this formula means that θ is bounded in the interval $[\theta_r, \theta_s]$. We close the equation by considering for the unsaturated hydraulic conductivity:

$$K(h) = K_s\frac{A}{A + |h|^\gamma}, \tag{92}$$

where $K_s = 0.00944 cm\, s^{-1}$ is the saturated hydraulic conductivity, and $A = 1.1.75 \times 10^6$ and $\gamma = 4.74$ are shape parameters.

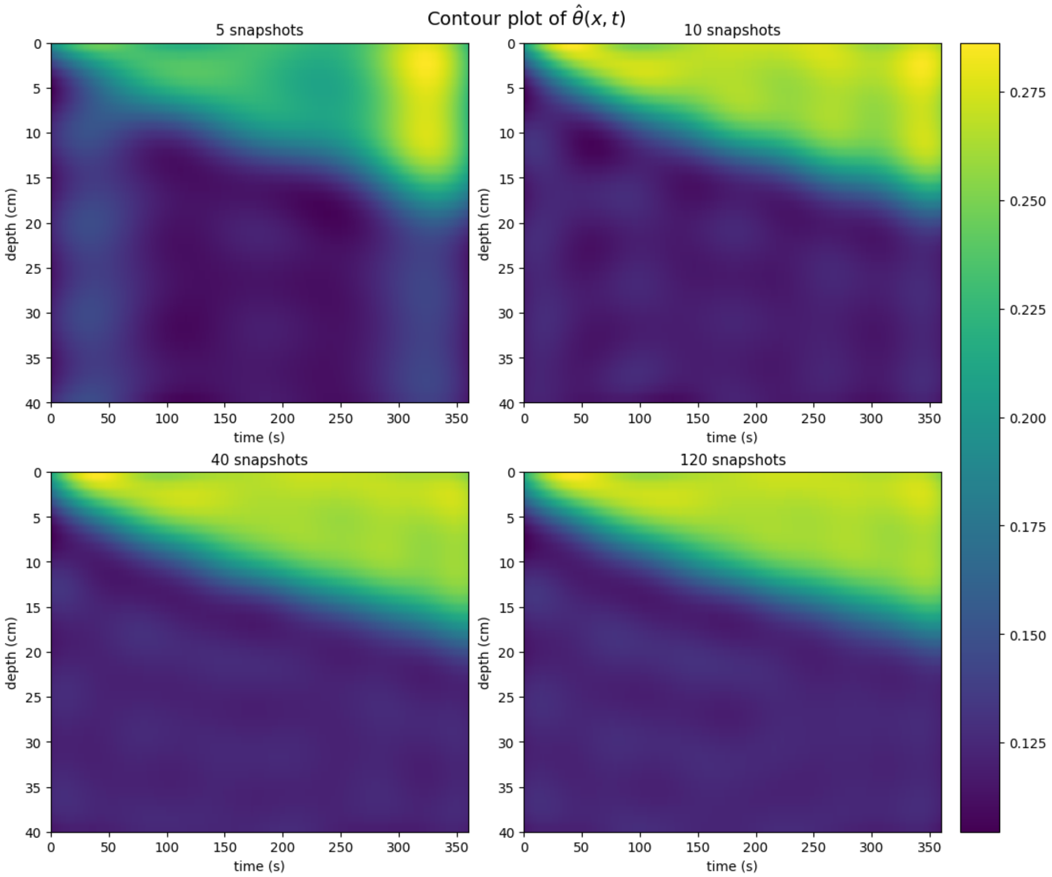
We consider the domain $X = [0, 40] \times [0, 360]$ and use a finite difference method-based solver to compute a reference trajectory:



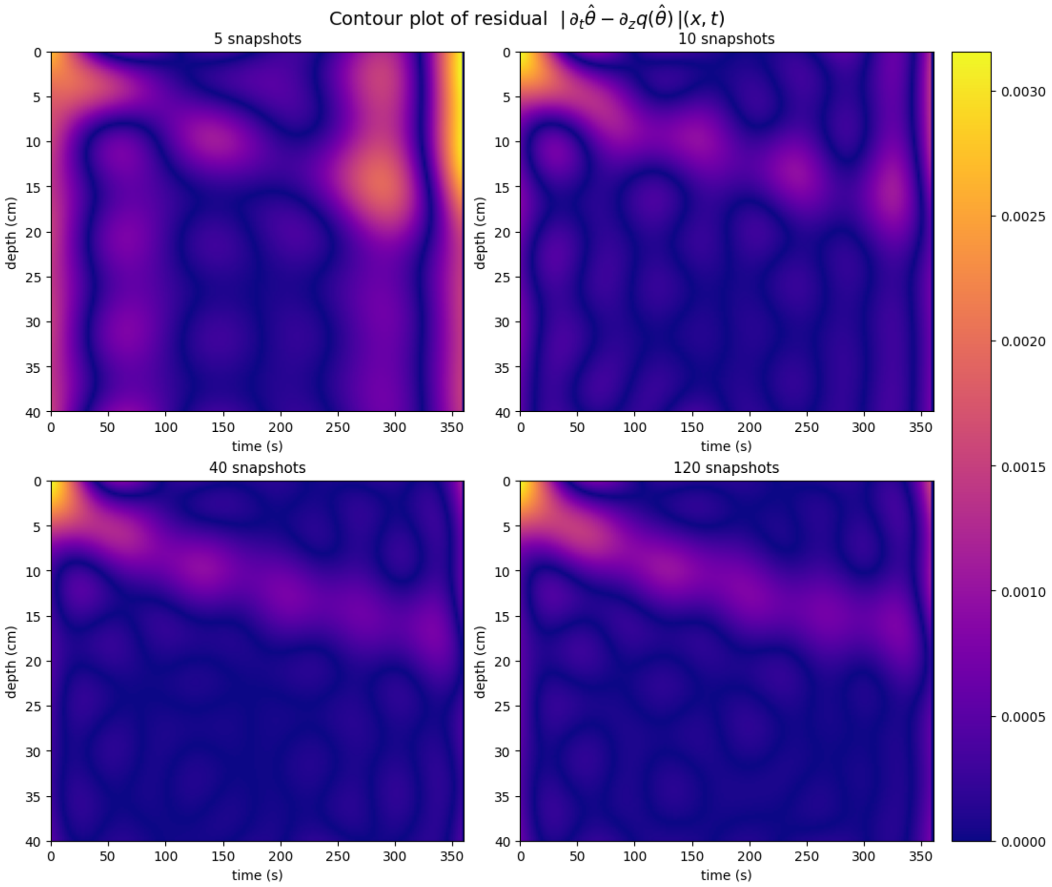
A 5% Gaussian noise is added to the dataset in order to mimic realistic sensor errors. We run the optimization by selecting a uniform set of N snapshots from the reference trajectory. Each snapshot at, say, time t_a is a discretization of the soil column with a reference moisture content at each point, i.e. $((z_0, \theta(z_0, t_a)), \ldots, (z_N, \theta(z_N, t_a)))$.

4.2.1 Results

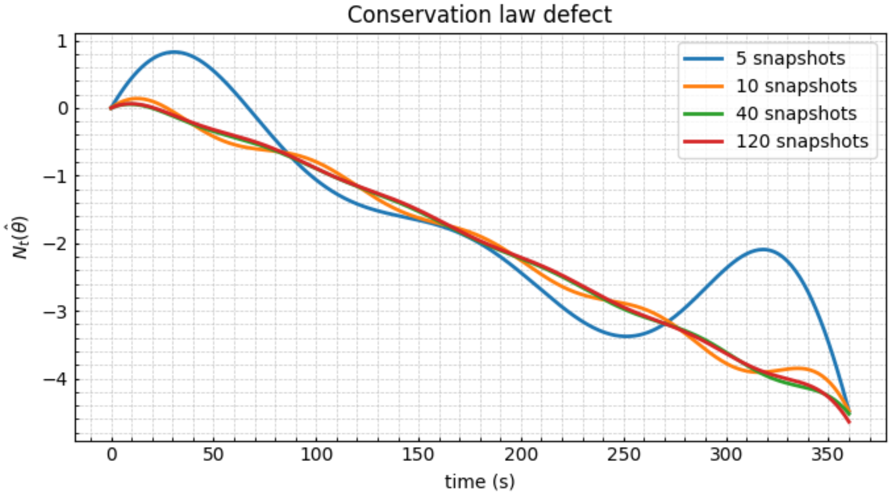
A visualization of the approximate function $\hat{\theta}(x,t)$ for $N \in \{5, 10, 40, 120\}$ with $z_i = i$ for $i = 0, \dots, 40$ is seen below:



Observe that 5 snapshots is too few for the optimizer to accurately capture the wetting front behavior, and as such the whole trajectory is not learned accurately. The improvement from 5 to 10 snapshots is significant, but afterwards the difference is marginal. The kernel expansion visually approximates the reference trajectory fairly well, but the underlying behavior from the Richards equation is not immediately captured. This is seen in the following visualization of the residual of $\hat{\theta}$ for the above approximate solutions:



Observe the two areas of high residual error common among all 4 approximate solutions: The bottom left area, representing the initial condition near the top spatial boundary, and a roughly quadratic line representing the wetting front. Since these are exactly the two areas where real change occurs in the trajectory, this represents an undesirable result. Since the conservation law (15) is directly related to the differential behavior over time, and also to the spatial boundary, the large residual error would indicate a large violation in the conservation law. The following plot of the mass balance defect over time confirms this:



Interestingly, all 4 approximate solutions end up having roughly the same final negative conservation law defect. Since the final mass at time $T = 360$ seems to be equal among the above contour plots, this error must then stem from the boundary fluxes over time. The conservation law defect shows that the error accumulates significantly over time, even if the point-wise residual error is relatively small. As this happens even with a very high number of snapshots, we see that a physics-informed alternative to this approach is desirable.

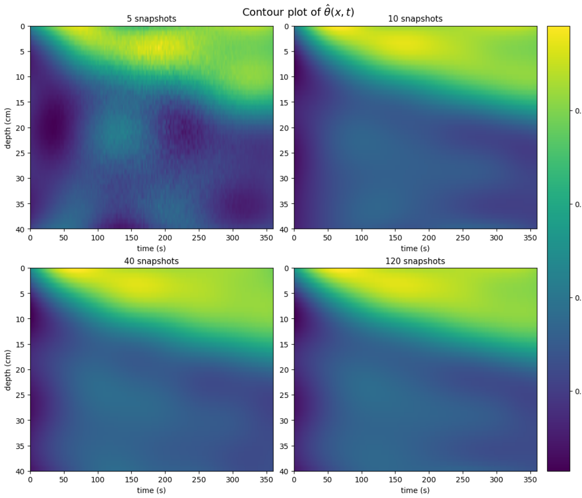
Finally we summarize the residual behavior on a dense grid of the approximate solution when changing the number N of spatial sample points at each snapshot.

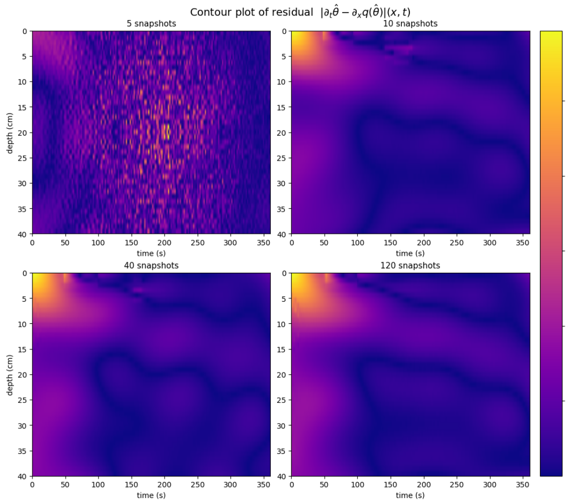
Spatial ablation behavior of $\hat{\theta}(x, t)$, 40 snapshots

	$\left\|\partial_t\hat{\theta} - \partial_x q(\hat{\theta})\right\|_{L^2}$	$\left\|\partial_t\hat{\theta} - \partial_x q(\hat{\theta})\right\|_{L^\infty}$	$\left\|\hat{\theta} - \theta_{true}\right\|_{L^2}$
$N = 5$	0.166	0.593	0.0126
$N = 10$	0.0464	0.187	5.65×10^{-5}
$N = 20$	0.0424	0.109	3.88×10^{-5}
$N = 40$	0.037	0.0869	2.13×10^{-5}

Here we also notice that 5 spatial samples for each snapshot is insufficient to accurately capture the behavior of the true wetting front behavior. Similarly to before, the difference is marginal beyond $N = 10$, both in the point-wise error and in the residual accuracy. We conclude that the baseline implementation has a moderate ability to capture the inherent physics of the trajectory, even when removing data-points from the optimization procedure.

We now present the same results for the transformed kernel $\widetilde{K}$. A visualization of the resulting approximate function $\hat{\theta}(x,t)$ and its PDE residual for $N \in \{5, 10, 40, 120\}$ snapshots with $z_i = i$ for $i = 0, \ldots, 40$ is seen below:

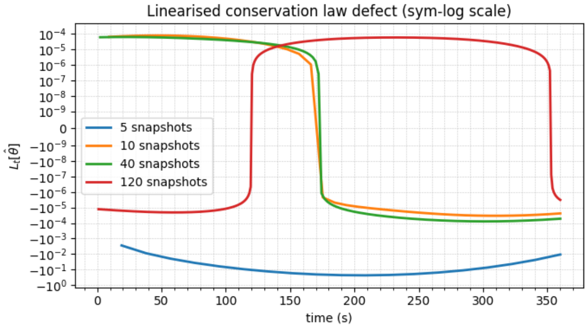


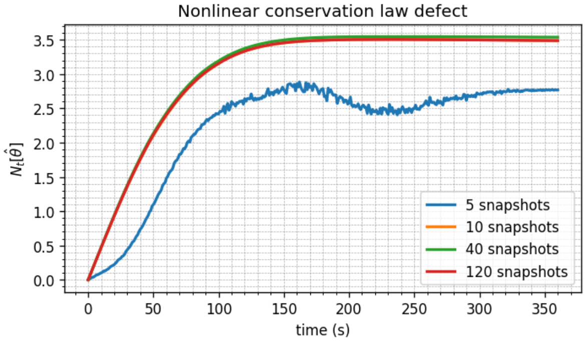


It is not clear if there is any improvement from the baseline kernel K. Indeed, the immediate difference that stands out is that the transformed kernel expansions are more sensitive to the Gaussian noise added to the dataset, especially for the lower values of snapshots. This is seen in both contour plots, where the optimized moisture content functions and their residuals are a lot more discontinuous than the baseline for 5 snapshots. For 10 snapshots and beyond, $\hat{\theta}(x,t)$ visually resembles the reference trajectory, but appears to have more mass in the dry-soil areas. This could be due to the linearized conservation law constraint imposed upon any kernel expansion of $\widetilde{K}$.

We also observe the same behavior in the residual than the baseline example. The areas of high residual error is smaller across the whole trajectory, but is even more focused near the spatiotemporal boundary. The errors are also slightly more similar for the different snapshot number values, which implies that the conservation law transformation imposes a rigid structure on the residual. This suggests that the conservation law constraint explicitly plays a role in capturing some underlying behavior of the trajectory, although not necessarily the correct behavior.

We conclude with a comparison of the linearized conservation law defect, which the kernel is designed to uphold, and the true nonlinear conservation law defect:





The linearized conservation law defect is very small for all 4 approximate solutions, indicating that the transformed kernel expansion indeed upholds the constraint as written. However, the nonlinear conservation law defect behaves similarly to the baseline examples, increasing to a large value (although slightly smaller than the baseline) before stagnating. This tells us that the linearization procedure does not accurately capture the true behavior of the conservation law, and therefore adding is as a constraint to the kernel does not help our estimations. The discontinuous behavior of the conservation law for 5 snapshots further informs us how the transformed kernel is more sensitive to the noise added in the trajectory.

We conclude that the transformed kernel does not yield improved results, although the evidence above strongly implies that the flaw lies with the linearization procedure, and not with the kernel implementation.

5 Discussion

In this section we give a detailed summary of the results from the previous section, and interpret them within the context of physics-based methods. We have evaluated the efficacy of our method in two very different estimation contexts: using a residual-based loss, and using a data-driven method with a reference trajectory. As such, we discuss the results in these two contexts separately.

5.1 On using the transformed kernel in residual learning

Compared to the baseline Gaussian kernel, the transformed kernel was fundamentally superior in the simple fact that it generated the correct solution to the heat equation BVP: The transformation adds necessary structure needed for the optimizer to avoid erroneous local minima. This is similar to [9], where the authors used a time-invariant exponential kernel in their heat equation experiments, and it goes unmentioned if their experiments failed without prior kernel transformation.

We therefore evaluate the transformed kernel on its own merits. Our results indicate $N = M = 20$ to be the ideal number of spatial and temporal basis points: the optimizer converges within a reasonable amount of time and iterations, and the results are significantly better than the baseline.

There are two aspect of the results for the transformed kernel that prevents us from claiming an objective improvement in results compared to similar existing literature ([11] and [9]). The first is large residual error present at the temporal boundary $t = 0$, which as already mentioned in the results section, indicates that the method is unable to properly estimate a crucial aspect of the BVP. We also saw that placing a higher weighted emphasis on the initial condition term in the loss function lead to a higher residual error, which indicates that the two objectives are somehow inversely related in the optimization process. It is possible that some combination of weights would remedy this issue: since the spatial boundary was indirectly enforced by the kernel transformation, it could be that weighing it less in the optimization procedure would not have adverse effects. This was not examined in our experiments. Additionally, further insight could be gained by implementing the same experiment for other (linear, time dependent) equations, where we could derive the corresponding kernel transformations as well. Indeed, if this structural issue was present in other equations as well, a potential remedy would both be more important and possibly easier to locate.

The second negative is the large computation times, which as mentioned in the results section is most likely due to an error in the implementation of the optimization. Still, the cost is a significant disadvantage when comparing to results in [11]. Note that for solving a single BVP, an optimization or PINN-based method is never as efficient as simply implementing a traditional finite difference or finite element method (FDM/FEM). As such, the primary objective of this work is to show an improvement in accuracy and physical validity, and in this area we have succeeded. The fact of our implementation containing inefficiencies, and in particular is not designed for high performance computing setups on GPUs, does not take away from this success.

5.2 On using the transformed kernel in data-driven solution

Here our results are partially limited due to the big constraint error added from the linearization procedure. The estimators defined from the transformed kernel perfectly upheld the linearized conservation constraint, but otherwise the results were not improved in any significant way. The conservation laws are derived essentially from integrating the differential equation over the its domain, and thus they represent a global average of the local trajectory behavior. The central question of this thesis is essentially if placing this global constraint on our reproducing kernel will produce estimators (in the form of kernel expansions) that are then also better at capturing the local behavior of the underlying equation. The fact that the linearized conservation law has such a large error to the true equation behavior prevents us from concluding in as strongly in the positive that this is also the case in this data-driven setting.

The one positive takeaway from this approach comes from the fact that the true trajectory can be very effectively captured with a sparse dataset (i.e. < 40 snapshots and fewer spatial data-points). This is the case using both the baseline and the physics-informed transformed kernels.

A more detailed analysis is required to fully assert that the transformed kernels are superior in this data-driven context. In particular, implementing the method for a linear equation (or where an exact linear conservation law may otherwise be derived) is of great interest, and could potentially yield more positive conclusive results.

5.3 Conclusion

In this thesis, we presented a general technique to incorporate physics-informed integro-differential conservation law constraints into RKHS kernel design. Through PDE and RKHS foundations, we argued that our technique yields well-defined transformed kernels. Through a linear heat equation and a nonlinear Richards equation example, we showed that the transformed kernel yields improved results compared to a baseline non-transformed kernel, in two different estimation settings. While the technique is strictly defined for linear constraints, we showed that nonlinear constraints may be linearized and thus also used in defining a kernel transformation.

An interesting direction to extend the work in this thesis involves combining multiple physics-informed constraints into kernel design; for instance a time-invariant kernel that also upholds the conservation law (13) could potentially be even more effective at solving the heat equation. However, it is unclear if constructing such a kernel is possible. The natural idea is to successively apply the relevant kernel transformations, but unless this process commutes we would need some additional argument to ensure that both constraints are upheld.

Another potential direction is to apply the transformed kernel framework to estimate the fundamental solution operator of a linear PDE. The fundamental solution is a particular solution to its associated PDE, so estimating them as transformed kernel expansions might improve the results in [9], similar to how we saw clear improvement. The Green's function could also be (partially) learned using the residual learning method (i.e. with the loss function (53)), thus giving the ability to solve many BVPs numerically with only a single optimization procedure.

References

[1] H.W. Alt and S. Luckhaus. “Quasilinear elliptic-parabolic differential equations”. In: *Math Z* 183 (1983), pp. 311–341. DOI: https://doi.org/10.1007/BF011764744.

[2] Nourhane Attia and Ali Akgül. “A reproducing kernel Hilbert space method for nonlinear partial differential equations: applications to physical equations”. In: *Physica Scripta* 97.10 (Sept. 2022), p. 104001. DOI: 10.1088/1402-4896/ac8958. URL: https://dx.doi.org/10.1088/1402-4896/ac8958.

[3] C. Carmeli et al. *Vector valued reproducing kernel Hilbert spaces and universality*. 2008. arXiv: 0807.1659 [math.FA]. URL: https://arxiv.org/abs/0807.1659.

[4] M.A. Celia, E.T. Bouloutas, and R.L. Zarba. “A general mass-conservative numerical solution for the unsaturated flow equation”. In: *Water resources research* 26 (1990), pp. 1483–1496. DOI: https://doi.org/10.1029/WR026i007p01483.

[5] Lawrence C. Evans. *Partial differential equations*. 2nd. American Mathematical Society, 2010.

[6] M.W. Farthing and F.L. Ogden. “Numerical Solution of Richards’ Equation: A Review of Advances and Challenges”. In: *Soil Science Society of America Journal* 81.6 (2017), pp. 1257–1269. DOI: https://doi.org/10.2136/sssaj2017.02.0058.

[7] Carl Jidling et al. *Linearly constrained Gaussian processes*. 2017. arXiv: 1703.00787 [stat.ML]. URL: https://arxiv.org/abs/1703.00787.

[8] Saburou Saitoh and Yoshihiro Sawano. *Theory of Reproducing Kernels and Applications*. Vol. 44. Developments in Mathematics. Springer, 2016. ISBN: 978-981-10-0529-9. DOI: 10.1007/978-981-10-0530-5.

[9] George Stepaniants. *Learning Partial Differential Equations in Reproducing Kernel Hilbert Spaces*. 2023. arXiv: 2108.11580 [math.ST]. URL: https://arxiv.org/abs/2108.11580.

[10] N. Thuerey et al. *Physics-based Deep Learning*. WWW, 2021. URL: https://physicsbaseddeeplearning org.

[11] Tianchi Yu et al. *Spectral Informed Neural Network: An Efficient and Low-Memory PINN*. 2024. arXiv: 2408.16414 [cs.LG]. URL: https://arxiv.org/abs/2408.16414.

A Appendix: Existence of minimizers in Hilbert spaces

In this Appendix we include a proof of existence and uniqueness of minimizers to certain Hilbert space functions, the main reason being that we use it in the background chapter and a standard reference was not identified.

Theorem A.1 (Minimizers in Hilbert spaces)**.** *Let $\mathcal{H}$ be a Hilbert space, and $J : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ a strictly convex, coercive, and lower semicontinuous functional which is not identically ∞. Then J has a unique minimizer in $\mathcal{H}$.*

Proof. Define $m = \inf_{f \in \mathcal{H}} J(f) \in \mathbb{R}$ and let $(f_n)_{n \in \mathbb{N}} \in \mathcal{H}$ be such that $J(f_n) \to m$. Coercivity implies that f_n is eventually bounded, which we may assume to be true for all $n \in \mathbb{N}$ without loss of generality. The Banach-Alaoglu theorem then implies that we can pass to a weakly convergent subsequence, thus assuming again without loss of generality that $f_n \rightharpoonup f^* \in \mathcal{H}$. By definition of lower semicontinuity we then have that $J(f^*) \leq \liminf_{n \to \infty} J(f_n) = m$, which means exactly that f^* minimizes the functional.

To see uniqueness, simply observe that if $J(f_1) = J(f_2) = m$ then for any $t \in [0, 1]$, strictly convexity implies that the convex combination $f_t = tf_1 + (1 - t)f_2$ satisfies:

$$J(f_1) < tJ(f_1) + (1 - t)J(f_2) = m, \tag{93}$$

a contradiction, so the minimizer must be unique. □