Approximation Error of Sobolev Regular Functions with tanh Neural Networks: Theoretical Impact on PINNs

Benjamin Girault, Rémi Emonet, Amaury Habrard, Jordan Patracone, Marc Sebban

Approximation Error of Sobolev Regular Functions with tanh Neural Networks: Theoretical Impact on PINNs

Benjamin Girault, Rémi Emonet, Amaury Habrard, Jordan Patracone, and Marc Sebban

Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School, Inria, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT- ETIENNE, France

Abstract. Considering the key role played by derivatives in *Partial Differential Equations* (PDEs), using the `tanh` activation function in *Physics-Informed Neural Networks* (PINNs) yields useful smoothness properties to derive theoretical guarantees in Sobolev norm. In this paper, we conduct an extensive functional analysis, unveiling tighter approximation bounds compared to prior works, especially for higher order PDEs. These better guarantees translate into smaller PINN architectures and improved generalization error with arbitrarily small Sobolev norms of the PDE residuals.

Keywords: Approximation error · Sobolev space · Physics-informed neural network · `tanh` activation function.

1 Introduction

Physics-informed Machine Learning [8,11] is an emerging topic that has received a tremendous interest from the scientific community. It consists in leveraging and embedding physical laws that govern some dynamical system into data-driven models. Ignoring the fundamental principles of the underlying theory may indeed lead to ill-posed problems and thus, yet optimal, implausible solutions yielding poor generalization. This domain knowledge can be exploited in various ways in deep-learning for (i) designing suitable physics-regularized loss functions, (ii) initializing models with meaningful parameters, (iii) guiding the design of consistent neural architectures, or (iv) building (theory+data)-driven hybrid models. A recent deep learning framework reflecting this new paradigm is the family of *physics-informed neural networks* (PINNs) [12] that are trained to predict solutions of a dynamical system while respecting partial differential equations (PDE). Despite a remarkable effectiveness, PINNs have been shown to also face pathological behaviors. In particular, they can be subject to overfitting phenomena, as illustrated in [1,6]. In this context, it has become crucial to study the theoretical properties of this new deep learning framework to gain a comprehensive grasp of its capabilities and limitations. However, despite an increasing interest in the past few years, the theoretical foundation of PINNs is

still at its early stage. Like any other machine learning model, in addition to the *optimization error* (which is hard to bound due to the highly non-convex nature of the optimization problem), the total error of PINNs depends on two main sources: (a) *estimation* error and (b) *approximation* error.

The *estimation error* (a) is mainly due to the finite amount of data available for learning, raising the question of the derivation of convergence/consistency guarantees. In [6], the risk consistency of PINNs is proved by employing a ridge regularization. This result ensures in an asymptotic regime that minimizing the empirical PDE residuals amounts to minimizing the corresponding generalization risk. Moreover, the authors show that plugging an additional Sobolev regularization allows to guarantee a strong convergence (in Sobolev norm) of the minimizer of the empirical risk to the (supposedly unique) solution of the PDE. In [4], the same kind of convergence results is provided for the Navier-Stokes PDE but this time, in L^2-norm and resorting to a quadrature rule instead of concentration inequalities. In [13], by following the Schauder approach, the authors prove an upper bound of the expected Hölder regularized PINN loss for linear second order elliptic and parabolic PDEs. In [14], the Rademacher complexity is used in an analysis based on the uniform law of large numbers to derive error estimates for linear PDEs. Following the work of [13], [15] presents a convergence analysis in H^2-norm when minimizing a Lipschitz regularized PINN loss function. Finally, in [6], the authors present the first theoretical analysis for hybrid models involving (observation or simulation) labelled data and physical knowledge.

The *approximation error* (b) is induced by the possibly limited capacity of the chosen function class. Fortunately, deep neural networks have been shown to be universal approximators [10]. On the other hand, as pointed out in [5], the universality results do not give bounds on the width of the networks. Due to this limitation, in the last few years, an abundant literature aimed at establishing quantitative results on the expressivity of neural networks (*e.g.*, see [7,9,16]). Most of these papers addressed the case of ReLU networks with error estimates either in L^∞ or Sobolev norm. However, since this activation function is piece-wise linear with vanishing derivatives, it cannot be efficiently used to solve PDEs that often require high-order derivatives. Beyond its use in recurrent neural networks, the `tanh` activation function has been shown to be a very good smooth candidate for PINNs for which measuring approximation errors in *high-order Sobolev norm* is crucial. In [5], the authors present the first expressivity guarantees in this setting by giving explicit bounds on the size of the PINN required for the network to be provably accurate.

Saving the question of convergence for future investigations, we focus in this paper on the approximation error of shallow neural networks and illustrate its impact on PINNs. Following and extending the line of work introduced in [5], our contribution is three-fold:

1. Through a rigorous functional analysis, we derive **tighter approximation bounds** of Sobolev-regular functions using 2-layer neural networks.

2. We show that our results hold not only for `tanh` but also for a **larger family of activation functions** that share common properties for ensuring the approximation guarantees.
3. We study the impact of the aforementioned results on PINNs illustrated in the case of the `tanh` function. We prove that **making the PDE residuals small requires much less neurons** compared to prior works. We study this behavior on the Navier-Stokes equations.

The rest of the paper is organized as follows: sec. 2 is dedicated to the presentation of the notations and background knowledge on PINNs. sec. 3 presents then our approximation bound and showcases its theoretical impact on PINNs. Finally, in sec. 4 and sec. 5, we conduct a rigorous study of the derivatives of the `tanh` function and demonstrate how these insights can be applied to tighten the approximation bound.

2 Preliminary Background and Notations

In this section, we introduce the definitions and notations necessary for the understanding of the rest of this paper. We also recall that deriving approximation guarantees of *Physics-informed neural networks* (PINNs) [12] boils down to analyzing K-times continuously differentiable functions that correspond to the residuals of the considered partial differential equations (PDEs).

2.1 PINNs and Functional Analysis

PINNs are neural architectures that are trained by integrating the information from both data measurements and physical knowledge in the form of PDEs. To illustrate how PINNs work, let us take the example of the incompressible Navier-Stokes equations that are defined as follows:

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{1}{\rho}\nabla p + \nu \nabla^2 u \qquad \nabla \cdot u = 0, \tag{1}$$

where u is the velocity vector field, p is the pressure, ρ is the fluid density, and ν is the kinematic viscosity. The equation set contains a momentum conservation equation, describing the evolution of velocity u over time, alongside a divergence constraint ensuring incompressibility. Together, these equations, augmented by appropriate boundary and initial conditions, provide a rigorous modeling of fluid motion across various scales. We will notably leverage these equations to illustrate the impact of our improved approximation guarantees derived in this paper.

The solutions to these equations can be parameterized by NNs, denoted as $\hat{u}$ and $\hat{p}$, which take any spatio-temporal location as input and output the corresponding approximation of u and p at that location. An unregularized algorithm for training these NNs with PDE knowledge aims at learning the parameters of NNs $\hat{u}$ and $\hat{p}$ by minimizing a loss function that includes a first term of measurements from the suitable initial and boundary conditions and a second

one representing the interior residuals measured at so-called *collocations points*. Those residuals, that characterize the amount of violation of the physical laws, are defined from (1) as follows:

$$\widehat{R}_1 = \frac{\partial \hat{u}}{\partial t} + (\hat{u} \cdot \nabla)\hat{u} + \frac{1}{\rho}\nabla \hat{p} - \nu \nabla^2 \hat{u} \qquad \widehat{R}_2 = \nabla \cdot \hat{u}. \tag{2}$$

The `tanh` activation function is usually preferred over the piece-wise linear ReLU to train PINNs in order to benefit from its smoothness and non vanishing derivatives. Despite being often very effective, PINNs have been shown to be subject to overfitting phenomena, as illustrated in [6] where examples are depicted with exploding derivatives and an L^2-norm of the PINN parameters tending to infinity while satisfying the PDE. In this context, it has become crucial to study the theoretical properties of this new deep learning framework. In this paper, we address this task from the *approximation error* perspective, *i.e.* by upper bounding the residuals, as defined by $\widehat{R}_1$ and $\widehat{R}_2$ in the case of the Navier-Stokes PDEs, and that are supposed to be equal to 0 by virtue of (1) (*i.e.* $R_1 = R_2 = 0$). To do so and considering that K-order derivatives are involved in the residual functions, we need to perform a functional analysis in Sobolev norm so as to penalize poor estimates of high-order derivatives.

For the sake of generality, let us consider the general goal of approximating a function f on the compact space $\Omega \subset \mathbb{R}^d$ by `tanh` based window functions in the context of windowed polynomial approximation of the form:

$$\widehat{f}(x) = \sum_{\mathbf{i}} \Phi_{\mathbf{i}}(x) q_{\mathbf{i}}(x). \tag{3}$$

To approximate f by $\widehat{f}$, Ω is partitioned into hypercubes $I_{\mathbf{i}}$, and each window $\Phi_{\mathbf{i}}$ is an approximate window of this hypercube. Under some hypotheses (see sec. 4), it is then enough for $q_{\mathbf{i}}$ to be an approximation of f where $\Phi_{\mathbf{i}}$ is (approximately) non-zero to obtain an accurate approximation $\hat{f}$ of f on Ω.

In [5], the authors use this setting to build a 2-layer `tanh` network with arbitrarily small approximation error for differentiable functions. In this context, studying the windows is therefore not enough, and an analysis of the derivatives involved in the equation is tantamount to quantifying and controlling the approximation error of the solution obtained. More precisely, these windows serve the combined role of a window on hypercubes and of a smoothing operator for the approximations and their derivatives. In [4], the authors alter this setting by restricting the maximum derivative order being approximated to 2, while better approximating power functions within the network. They then apply it to obtain approximation guarantees for a solution of a PDE.

Our goal here is twofold: (i) generalizing the notion of window $\Phi_{\mathbf{i}}$, and (ii) performing a careful functional analysis of the `tanh`-based windows and their derivatives to derive tighter guarantees compared to [4,5]. Note that we do not study the impact of the individual approximations $\{q_{\mathbf{i}}\}_{\mathbf{i}}$, but only that of the windows (we refer the interested reader to [4,5] for details on the impact of these approximations).

2.2 Notations

We use bold lower case letter for vectors, bold upper case letters for matrices and fraktur lower case letters (*e.g.*, $\mathfrak{a}$) for multi-indices. A multi-index is a tuple of non-negative integers and is typically used to compactly express (higher) order derivatives and multivariate monomials. For a multivariate function f with d-dimensional input, a partial derivative of order $|\mathfrak{a}| := \sum_l \mathfrak{a}_l$ is defined as

$$D^{\mathfrak{a}} f := \frac{\partial^{|\mathfrak{a}|} f}{\partial x_1^{\mathfrak{a}_1} \cdots \partial x_d^{\mathfrak{a}_d}},$$

where $\mathfrak{a}_l$ is thus the order of differentiation with respect to the variable x_l. Similarly, $\mathbf{x}^{\mathfrak{a}} := \sum_l x_l^{\mathfrak{a}_l}$ defines a monomial.

To facilitate the reading of the following sections, we regroup here the main notations of the paper:

- K: largest derivative order to approximate ($K = 2$ for Navier-Stokes PDEs),
- m: maximum order of derivatives used in the Sobolev norm,
- $k \leqslant m$: order of a derivative,
- n: finite difference order used to approximate products using `tanh` networks,
- d: dimension of the input space (typically $d = 3$ for NS),
- $\Omega = \times_{l=1}^{d}$: separable input space,
- $\Omega_l = [a_0^l, a_0^l + \delta \cdot N_l] \subset \mathbb{R}$: input domain along dimension l,
- δ: length of an hypercube $I_{\mathfrak{i}}$,
- N_l: number of subdivisions along dimension l,
- $I_{\mathfrak{i}} = \times_{l=1}^{d} I_{\mathfrak{i}_l}^l$: hypercube indexed with multi-index $\mathfrak{i}$,
- $I_i^l = [a_{i-1}^l, a_i^l]$: interval along dimension l, with $a_i^l = a_0^l + i\delta$,
- $|\mathfrak{a}| = \sum_l \mathfrak{a}_l$: absolute value, or sum, of a multi-index,
- $P_{n,d} = \{\mathfrak{a} \in \mathbb{N}^d : |\mathfrak{a}| = n\}$: set of multi-indices, all d-tuples summing to n,
- $|P_{n,d}| = \binom{n+d-1}{n}$: number of d-tuples summing to n.
- $\{D^{\mathfrak{a}} f : \mathfrak{a} \in P_{n,d}\}$: set of partial derivatives of order n, in d dimension.

The Sobolev space of order k of functions with domain $\Omega \subseteq \mathbb{R}^d$ and with respect to the $L^p(\Omega)$-norm is defined as the set of functions having all its derivatives of order up to k having finite $L^p(\Omega)$-norm:

$$W^{k,p}(\Omega) = \{f \in L^p(\Omega) : \forall |\mathfrak{a}| \leqslant k, D^{\mathfrak{a}} f \in L^p(\Omega)\}.$$

Taking the ℓ^p-norm of the vector of derivatives of order exactly k equips the Sobolev space with a semi-norm denoted as $|f|_{W^{k,p}(\Omega)} := (\sum_{|\mathfrak{a}|=k} \|D^{\mathfrak{a}} f\|_{L^p(\Omega)}^p)^{1/p}$ if $p < \infty$, and $|f|_{W^{k,p}(\Omega)} := \max_{|\mathfrak{a}|=k} \|D^{\mathfrak{a}} f\|_{L^\infty(\Omega)}$ otherwise. Further aggregating all the semi-norms *up to order* m (using again the ℓ^p-norm) finally defines Sobolev norm $\|f\|_{W^{m,p}(\Omega)} := \|\{|f|_{W^{k,p}(\Omega)} : k \leqslant m\}\|_p = (\sum_{|\mathfrak{a}|\leqslant m} \|D^{\mathfrak{a}} f\|_{L^p(\Omega)}^p)^{1/p}$.

3 Approximation Bound for Sobolev Functions

In this section, we introduce the main contribution of this paper: a tighter approximation bound for Sobolev-regular functions using 2-layer `tanh` neural net-

works. Intuitively, this bound is constructed by quantifying two intertwined approximations: the approximation of f by a windowed polynomial approximation (see eq. (3)), and to what extent the latter can be modeled by a shallow `tanh` neural network. As such, the capacity to represent windows using a `tanh` network has significant implications. This will be manifested in our bound through a window parameter α (see sec. 4) whose value is discussed in sec. 3.2. Finally, this result serves as the foundation for devising approximation guarantees of PINNs, explored in sec. 3.3, in the context of Navier-Stokes PDEs.

3.1 General Bound for `tanh` Networks

We present in the following thm. 1 our main approximation bound valid for any Sobolev norm of order K.

Theorem 1. *Let $d, n \geqslant 2$, $K \geqslant 1$, $m \geqslant K+1$, $\zeta > 0$, $f \in W^{m,2}(\Omega)$. Then for every $0 < \delta \leqslant 1$ creating a partition of Ω, there exists a `tanh` neural network $\hat{f}^\delta$ with 2 hidden layers of widths at most $3\lceil\frac{m+n-2}{2}\rceil|P_{m-1,d+1}| + \sum_l \frac{\mu(\Omega_l)}{\delta}1$, and at most $3\lceil\frac{d+n}{2}\rceil|P_{d+1,d+1}|\frac{\mu(\Omega)}{\delta^d}$, where $\mu(.)$ returns the volume of a space, and such that $\forall 0 \leqslant k \leqslant K$, we have:*

$$\|f - \hat{f}^\delta\|_{W^{k,2}(\Omega)} \leqslant \mathcal{C}^{[\text{approx}]}_{k,m,d} \Delta^{[\tanh]}_{k,m,d}(\delta; \alpha_{\epsilon,\delta}) |f|_{W^{m,2}(\Omega)} \tag{4}$$

where $\epsilon = 3^d \delta^{m+d}/\mu(\Omega)$, $\mathcal{P}_k$ is `tanh` specific and given lem. 3, $\alpha_{\epsilon,\delta}$ is also `tanh` specific and key to study the impact of δ, and:

$$\Delta^{[\tanh]}_{k,m,d}(\delta; \alpha_{\epsilon,\delta}) = \mathcal{P}^d_k \alpha^k_{\epsilon,\delta} \delta^m \tag{5}$$

$$\mathcal{C}^{[\text{approx}]}_{k,m,d} = 2^{\min(1,2k)} 3^{3d/2}(1+\zeta)\mathcal{C}_{k,m,d} \tag{6}$$

$$\mathcal{C}_{k,m,d} = \max_{0 \leqslant l \leqslant k} \binom{d+l-1}{l}^{\frac{1}{2}} \left(\frac{(m-l)!}{\left(\lceil\frac{m-l}{d}\rceil!\right)^2}\right)^{\frac{1}{2}} \left(\frac{3\sqrt{d}}{\pi}\right)^{m-l}. \tag{7}$$

Before presenting the proof of this result, we propose to discuss the improvements achieved over [4,5]. First, compared to [4, thm. B.7] and as mentioned above, our theorem generalizes to any maximum Sobolev norm of order K without the limitation to $K \leqslant 2$ [4] or $n = 2$ [5]. This generalization is obtained at the price of an additional factor of 2^k that comes from the upper bound in lem. 3 for arbitrary values of K. We also have a $3^{d/2}$ term from aggregating all Sobolev norms on each cube $I_\mathfrak{j}$. Another important point is that we also make explicit in this theorem the dependence on `tanh` activation function and on the associated window parameter α (see sec. 4), such that the bound can be instantiated with our two proposed methods from sec. 4.2. Importantly, this constant α does not depend on f anymore, given that ϵ is now independent from f. In other words, the `tanh` windows involved are now defined independently of the function f being approximated.

Remark 1. It is unclear why the term $3^{d/2}$ does not appear in [4, thm. B.7], especially when working with respect to the $W^{k,2}$ Sobolev space instead of $W^{k,\infty}$ as in [5].

We now present the proof of thm. 1 below. This proof requires some lemmas that are presented later in the paper, and rely on the assumptions in def. 1.

Proof. This proof follows most of the principles of that of [4, thm. B.7], but using our key contributions and extending to any order K. First, the approximation error is rewritten using the triangular inequality with:

$$\|f - \hat{f}^{\delta}\|_{W^{k,2}(\Omega)} \leqslant \|f(1 - \sum_{\mathtt{i}} \Phi_{\mathtt{i}})\|_{W^{k,2}(\Omega)} + \|\sum_{\mathtt{i}} (f - q_{\mathtt{i}}^{\delta})\Phi_{\mathtt{i}}\|_{W^{k,2}(\Omega)} + \|\sum_{\mathtt{i}} (\times - \widehat{\times})(q_{\mathtt{i}}^{\delta}, \rho_{\mathtt{i}_1}^1, \ldots, \rho_{\mathtt{i}_d}^d)\|_{W^{k,2}(\Omega)} \quad (8)$$

where $\times$ is the function returning the product of its inputs and $\widehat{\times}$ the corresponding approximation with a shallow `tanh` network. Using lem. 1, we obtain that the first norm is zero, and we are left with upper bounding the two remaining norms. To that end, we consider a hypercube $I_{\mathtt{j}}$ for multiindex $\mathtt{j}$ not related to $\mathtt{i}$ in the sums above, and compute the norms over each of these sets.

Adapting [4, eq. (B.30)], using lem. 4 instead of [4, lem. B.3] for $k \geqslant 1$:

$$\begin{aligned}
&\|\sum_{\mathtt{i}} (f - q_{\mathtt{i}}^{\delta})\Phi_{\mathtt{i}}\|_{W^{k,2}(I_{\mathtt{j}})} \\
&\leqslant 2^k 3^d \left(\mathcal{C}^{[\mathtt{j}]}_{k,m,d}(f)\delta^{m-K} + \frac{\eta}{\delta^k}\right) \alpha^k 2^{k-1} \mathcal{P}_k^d + 2^k \frac{\mu(\Omega)}{\delta^d} (\mathcal{C}^{[\mathtt{j}]}_{k,m,d}(f) + \eta) \alpha^k 2^{k-1} \mathcal{P}_k^d \epsilon \\
&\leqslant \alpha^k \delta^m 2^{2k} \mathcal{P}_k^d 3^d \mathcal{C}^{[\mathtt{j}]}_{k,m,d}(f)(1 + \frac{\zeta}{2}), \qquad (9)
\end{aligned}$$

when $\delta \leqslant 1$ and when we choose:

$$\eta \leqslant \delta^m \mathcal{C}^{[\mathtt{j}]}_{k,m,d}(f)\zeta/2 \qquad \text{and} \qquad \epsilon \leqslant \delta^{m+d} 3^d / \mu(\Omega). \qquad (10)$$

For $k = 0$, using $|\sigma(x)| \leqslant 1$ leads to:

$$\begin{aligned}
\|\sum_{\mathtt{i}} (f - q_{\mathtt{i}}^{\delta})\Phi_{\mathtt{i}}\|_{W^{0,2}(I_{\mathtt{j}})} &\leqslant 3^d \left(\mathcal{C}^{[\mathtt{j}]}_{0,m,d}(f)\delta^m + \eta\right) + \frac{\mu(\Omega)}{\delta^d} (\mathcal{C}^{[\mathtt{j}]}_{0,m,d}(f) + \eta)\epsilon \\
&\leqslant 2\delta^m 3^d \mathcal{C}^{[\mathtt{j}]}_{0,m,d}(f)(1 + \frac{\zeta}{2}), \qquad (11)
\end{aligned}$$

when $\eta \leqslant \delta^m \mathcal{C}^{[\mathtt{j}]}_{0,m,d}(f)\zeta/2$.

We now adapt [4, eq. (B.32)], using lem. 4 instead of [4, lem. B.3], and using lem. 5 instead of [4, lem. A.6]:

$$\begin{aligned}
&\|\sum_{\mathtt{i}} (\times - \widehat{\times})(q_{\mathtt{i}}^{\delta}, \rho_{\mathtt{i}_1}^1, \ldots, \rho_{\mathtt{i}_d}^d)\|_{W^{k,2}(\Omega)} \\
&\qquad \leqslant \delta^{-d} (\mu(\Omega))^{3/2} \mathcal{C}^{[\mathrm{win}]}_{k,d} h \left(\alpha^k 2^{k-1} \mathcal{P}_k^d + \|q_{\mathtt{i}}\|_{W^{k,\infty}(\Omega)}\right)^k \\
&\qquad \leqslant \alpha^k \delta^m 2^{2k} \mathcal{P}_k^d 3^d \mathcal{C}_{k,m,d}(f) \frac{\zeta}{2}, \qquad (12)
\end{aligned}$$

when

$$h \leqslant \max_{1 \leqslant k \leqslant K} \frac{\alpha^k 2^{\min(1,2k)} \mathcal{P}_k^d \delta^{m+d} 3^d \mathcal{C}_{k,m,d} \frac{\zeta}{2}}{(\mu(\Omega))^{3/2} \mathcal{C}_{k,d}^{[\text{win}]} \left(\alpha^k 2^{\min(0,k-1)} \mathcal{P}_k^d + \|q_\text{i}\|_{W^{k,\infty}(\Omega)}\right)^k}. \tag{13}$$

This choice of constant h is still valid for $k = 0$.

More precisely, the constant $\mathcal{C}_{k,m,d}^{[\text{j}]}(f)$ is the approximation error from the Bramble-Hilbert lemma on the polynomial approximation of degree $m-1$ of f on the cube J_j composed of I_j and its neighbors [4, lem. A.4]. We have then:

$$\mathcal{C}_{k,m,d}^{[\text{j}]}(f) = \mathcal{C}_{k,m,d} |f|_{W^{m,2}(J_\text{j})}.$$

Further decomposing the semi-norm of f above on the smaller cubes I_j, and summing the upper bounds of the squared norms in eq. (9) and (12) over all cubes I_j, we obtain the bound of the statement by noticing that each of these semi-norms is involved in at most 3^d terms. □

3.2 Instantiation for our Proposed Methods for Choosing α

We now study the bound in thm. 1 with respect to the window's length δ by leveraging our proposed methods to derive α in sec. 4.2. Our objective is to illustrate the gain brought by them. More precisely, we are interested in the term $\Delta_{k,m,d}^{[\tanh]}(\delta;\alpha)$ involved in (4). We first discuss it as a function of δ on the example shown on fig. 1, and then introduce two corollaries giving the asymptotic behavior of $\Delta_{k,m,d}^{[\tanh]}(\delta;\alpha)$ as δ goes to 0. In order to facilitate the presentation, we propose to consider in this part the two methods for deriving α as black-boxes: the first method introduced in thm. 3 is labeled as "Simple" and the second one provided in thm. 4 is denoted as "Lambert".

First, fig. 1 shows $\Delta_{k,m,d}^{[\tanh]}(\delta;\alpha)$ as a function of δ, for a polynomial approximation of degree $m-1=K$, and a space of unit volume and dimension $d=3$. There are several key observations to be made. First of all, this term is a decreasing function only below some threshold. This shows the three regimes for $\Delta_{k,m,d}^{[\tanh]}(\delta;\alpha)$: (i) when δ is large, α can be small, and the `tanh` step function and its derivative are smooth, thus with a contained magnitude, (ii) when δ is small, the asymptotic behavior applies, and the non-smoothness of the derivatives of the `tanh` step function (α^k large) is balanced by the vanishing δ^m term, and (iii) in the intermediate regime, the term α^k grows too quickly compared to δ^m vanishing, and $\Delta_{k,m,d}^{[\tanh]}(\delta;\alpha)$ increases when δ decreases. More precisely, the worse error is attained for a cube length of 0.1, and decreasing this error requires either increasing this length but without the possibility to decrease the error as much as desired, or decreasing this length significantly at the cost of increasing the size of the network. Indeed, the size of the layers are in the order $1/\delta$ and $1/\delta^d$, thus significantly increasing as δ decreases.

Interestingly, fig. 1 shows that our proposed methods for choosing α allow for larger values of δ for a given error $\Delta_{k,m,d}^{[\tanh]}(\delta;\alpha)$. For example, for $K=2$, we can have δ be larger up to one order of magnitude, which in turn **decreases the size of the network.**

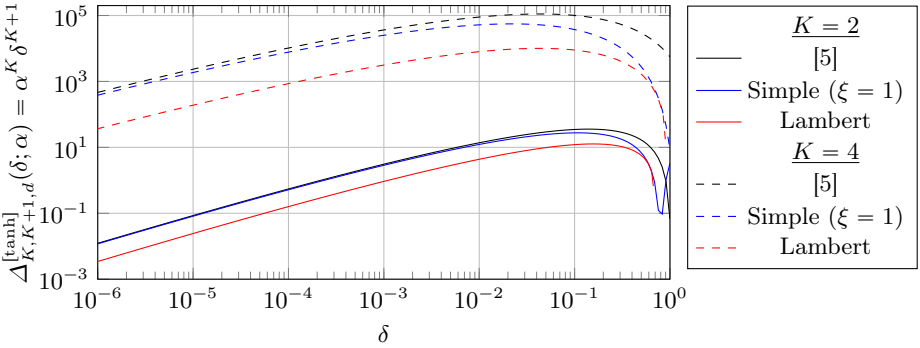


Fig. 1: Values of $\Delta^{[\tanh]}_{k,m,d}(\delta;\alpha)$ in (4) as a function of δ for $K = 2$ and $K = 4$, $d = 3$, $m = K + 1$, and the two methods proposed in sec. 4.2, compared to the method from [5]. In all cases, the value of ϵ is given by thm. 1.

The rest of this section is devoted to asymptotic equivalences for $\Delta^{[\tanh]}_{k,m,d}(\delta;\alpha)$ with both our proposed methods. In particular, the cor. 1 fixes the issue with [4, lem. B.4] (see rem. 2). Interestingly, the two corollaries below are very similar, with the simple method having the same asymptotic equivalence than the method based on the Lambert W Function, but when ξ goes to 2. In other words, using the Lambert W Function avoids this parameter ξ.

Corollary 1. *If δ is small enough, using thm. 3 yields the upper bound:*

$$\Delta^{[\tanh]}_{k,m,d}(\delta;\alpha^{[\text{simple}]}_{\epsilon,\delta}(\xi)) = \left(\frac{m+d+K}{\xi}\right)^k \ln^k\left(\frac{1}{\delta}\right)\delta^{m-k} + o\left(\ln^k\left(\frac{1}{\delta}\right)\delta^{m-k}\right).$$

Corollary 2. *Using thm. 4, the upper bound of thm. 1 becomes:*

$$\Delta^{[\tanh]}_{k,m,d}(\delta;\alpha^{[\text{Lambert}]}_{\epsilon,\delta}) = \left(\frac{m+d+K}{2}\right)^k \ln^k\left(\frac{1}{\delta}\right)\delta^{m-k} + o\left(\ln^k\left(\frac{1}{\delta}\right)\delta^{m-k}\right).$$

3.3 Theoretical Impact on PINNs: the Case of Navier-Stokes PDEs

The functional analysis in Sobolev norm conducted above is now applied to upper bound the Navier-Stokes (NS) PDEs residuals, $\widehat{R}_1$ and $\widehat{R}_2$ as stated in sec. 2.

To fully characterize the PDEs, we need to specify boundary and initial conditions. Hereafter, we assume periodic boundary conditions so that $\Omega = \mathbb{T}^{d-1} \times [0,T]$ where $\mathbb{T}^{d-1} = [0,1)^{d-1}$ denotes the $(d-1)$-dimensional torus and $T > 0$ is some time horizon. We also consider an initial condition $u_0 \in W^{r,2}(\mathbb{T}^{d-1})$ with Sobolev regularity $r \in \mathbb{N}$.

The following bound is derived by directly leveraging our novel approximation bound into the proof of [4, thm. 3.1.].

Theorem 2. *Let $n \geqslant 2$, $d \geqslant 2$, $r \geqslant 1$, $m \geqslant 3$, and let $u_0 \in W^{r,2}(\mathbb{T}^{d-1})$ with $r > \frac{d}{2} + 2m$ and $\nabla \cdot u_0 = 0$. It holds that:*

- *there exist $T > 0$ and a classical solution u to the NS equations such that $u \in W^{m,2}(\Omega)$, $\nabla p \in W^{m-1,2}(\Omega)$, $\Omega = \mathbb{T}^{d-1} \times [0, T]$, and $u(t = 0) = u_0$*
- *for every $\delta \in (0, 1)$ creating a partition of $\mathbb{T}^{d-1}$, there exist* `tanh` *neural networks $\hat{u}_l$, $1 \leqslant l \leqslant d-1$, and $\hat{p}$, each with two hidden layers, of widths $3\left\lceil \frac{m+n-2}{2} \right\rceil |P_{m-1,d+1}| + \lceil T\delta^{-1} \rceil + (d-1)\delta^{-1} - 1$ for the first layer and $3\left\lceil \frac{d+n}{2} \right\rceil |P_{d+1,d+1}| \lceil T\delta^{-1} \rceil \delta^{-(d-1)}$ for the next, such that for every $1 \leqslant l \leqslant d$-$1$,*

$$\left\| \widehat{R}_1 \right\|_{L^2(\Omega)} = \left\| \frac{\partial \hat{u}_l}{\partial t} + (\hat{u} \cdot \nabla)\hat{u}_l + \frac{1}{\rho}(\nabla \hat{p})_l - \nu \nabla^2 \hat{u}_l \right\|_{L^2(\Omega)} \leqslant C_1 \tag{14}$$

$$\left\| \widehat{R}_2 \right\|_{L^2(\Omega)} = \| \nabla \cdot \hat{u} \|_{L^2(\Omega)} \leqslant C_2 \tag{15}$$

$$\| (u_0)_l - \hat{u}_l(t = 0) \|_{L^2(\mathbb{T}^{d-1})} \leqslant C_3, \tag{16}$$

with the upper-bounds C_1, C_2, C_3 being defined as follows

$$\begin{aligned} C_1 = {} & \mathcal{C}^{[\text{approx}]}_{0,m,d} \Delta^{[\tanh]}_{0,m,d}(\delta; \alpha_{\epsilon,\delta}) |u|_{W^{m,2}(\Omega)} \|u_l\|_{W^{1,\infty}(\Omega)} \sqrt{d-1} \\ & + \mathcal{C}^{[\text{approx}]}_{1,m,d} \Delta^{[\tanh]}_{1,m,d}(\delta; \alpha_{\epsilon,\delta}) |u|_{W^{m,2}(\Omega)} (1 + \sqrt{d-1} \max_l \|\hat{u}_l\|_{W^{0,\infty}(\Omega)}) \\ & + \mathcal{C}^{[\text{approx}]}_{1,m-1,d} \Delta^{[\tanh]}_{1,m-1,d}(\delta; \alpha_{\epsilon,\delta}) |p|_{W^{m-1,2}(\Omega)} \rho^{-1} \\ & + \mathcal{C}^{[\text{approx}]}_{2,m,d} \Delta^{[\tanh]}_{2,m,d}(\delta; \alpha_{\epsilon,\delta}) |u|_{W^{m,2}(\Omega)} \nu \sqrt{d-1} \end{aligned} \tag{17}$$

$$C_2 = \mathcal{C}^{[\text{approx}]}_{1,m,d} \Delta^{[\tanh]}_{1,m,d}(\delta; \alpha_{\epsilon,\delta}) |u|_{W^{m,2}(\Omega)} \sqrt{d-1} \tag{18}$$

$$C_3 = \mathcal{C}^{[\text{approx}]}_{1,m,d} \Delta^{[\tanh]}_{1,m,d}(\delta; \alpha_{\epsilon,\delta}) |u|_{W^{m,2}(\Omega)} \sqrt{\frac{2 \max\{2\text{diam}(\Omega), d\}}{\text{rad}(\Omega)}}, \tag{19}$$

where $\text{diam}(\Omega)$ *is the diameter of Ω and* $\text{rad}(\Omega)$ *is the radius of the largest d-dimensional ball that can be inscribed into Ω.*

In light of cor. 1 and 2, the upper-bounds C_1, C_2, C_3 vary in $\ln^2(\delta^{-1})\delta^{m-2}$, $\ln(\delta^{-1})\delta^{m-1}$ and $\ln(\delta^{-1})\delta^{m-1}$, respectively. Our bound additionally improves upon [4] by having tighter multiplicative constants and decouples the multiplicative constants from the functions u and p. Specifically, our analysis illustrated in fig. 1 implies that, for the same approximation errors as in [4], we can afford δ one order of magnitude greater, thus **significantly reducing the size of the PINNs' layers.**

4 Differentiable Smoothing Windows

In this section, we focus on the approximation of the smoothing windows $\{\Phi_i\}_i$, appearing in eq.(3) with a shallow neural network of activation function σ. These windows should verify the following properties:

- $\forall \mathbf{x} \in I_{\mathbf{i}}, \sum_{\mathbf{j} \in \mathcal{N}_d(\mathbf{i})} \Phi_{\mathbf{j}}(\mathbf{x}) \approx 1$: the value of $\hat{f}$ on $I_{\mathbf{i}}$ is determined by the hypercube $I_{\mathbf{i}}$ and its neighbors,
- $\forall \mathbf{x} \in I_{\mathbf{i}}, \sum_{\mathbf{j} \notin \mathcal{N}_d(\mathbf{i})} \Phi_{\mathbf{j}}(\mathbf{x}) \approx 0$: the influence of all other hypercubes on the value of $\hat{f}$ on $I_{\mathbf{i}}$ is negligible,

where $\mathcal{N}_d(\mathbf{i})$ is the set of hypercube indices corresponding to $\mathbf{i}$ and its neighbors (sharing at least a vertex with $I_{\mathbf{i}}$). Intuitively, in order to construct smooth 1D windows, the activation function σ should be step-like: smooth windows can be obtained by averaging a left and right smooth step function $\gamma : y \mapsto \sigma(\alpha y)$ for some window parameter $\alpha > 0$. Intuitively, the goal is to build a smooth step-like building block from a shallow network as defined in def. 1, and linearly combine these building blocks to get smooth 1D windows def. 2, and finally multidimensional windows def. 3.

In def. 1, we precisely identify the 6 conditions that must be satisfied by the step function. Then, leveraging sec. 5, we conduct an in-depth study based on the `tanh` activation function, and devise the appropriate value for the shared parameter α. In appendix C, we propose an alternative definition of step function with most of the properties set forth by def. 1 proved (while the missing being left as a conjecture, but verified in practice).

4.1 General Step Function and Differentiable Windows $\Phi_{\mathbf{i}}$

The definition below formalizes the key properties for these step functions:

Definition 1 (1D Differentiable Step Function). *Let $K \geqslant 1$. For a window of size δ, a* 1D K-times differentiable step function *is a function γ verifying for some sharpness $\epsilon > 0$ and for all $1 \leqslant k \leqslant K$:*

$[\mathcal{A}_1]$ $1 - \gamma$ *and* $|\gamma^{(k)}|$ *are decreasing on* $[\delta, +\infty)$,
$[\mathcal{A}_2]$ $1 - \gamma(\delta) \leqslant \epsilon$,
$[\mathcal{A}_3]$ $|\gamma^{(k)}(\delta)| \leqslant \epsilon$,
$[\mathcal{A}_4]$ γ *is odd:* $\forall x, \gamma(-x) = -\gamma(x)$,
$[\mathcal{A}_5]$ $\lim_{+\infty} \gamma = 1$,
$[\mathcal{A}_6]$ $\forall x, |\gamma(x)| \leqslant 1$.

To define the 1D windows below, we extend the partition $\bigcup_{i=1}^{N_l} I_i^l$ of Ω_l to $\mathbb{R}$ with $\bar{I}_i^l = [\bar{a}_{i-1}^l, \bar{a}_i^l]$ where $\bar{a}_0^l = -\infty$, $\bar{a}_{N_l}^l = \infty$, and $\bar{a}_i^l = a_i^l$ otherwise. These windows are then defined on the whole space $\mathbb{R}^d$, and sum to 1 everywhere.

Definition 2 (1D Differentiable Smoothing Window). *Let γ as in def. 1 with sharpness ϵ, and $\bar{I}_i^l = [\bar{a}_{i-1}^l, \bar{a}_i^l]$ be an interval of length δ. Then ρ_i is a 1D differentiable smoothing window of sharpness ϵ defined as:*

$$\rho_i(x) = \frac{\gamma\big(x - \bar{a}_{i-1}^l\big) + \gamma\big(\bar{a}_i^l - x\big)}{2},$$

with the convention $\gamma(\infty) = \lim_{\infty} \gamma$.

Intuitively, assumption $[\mathcal{A}_1]$ states that the smoothing window ρ_i and its derivatives, while being defined on $\mathbb{R}$ and possibly non-monotonous, are decreasing at distance at least δ from either end of the interval $\bar{I}_i$. In addition assumptions $[\mathcal{A}_2]$ and $[\mathcal{A}_3]$, state that at distance exactly δ from $\bar{I}_i$, the magnitude of the window is no greater than its sharpness ϵ. Assumption $[\mathcal{A}_5]$ is enough to ensure that the limit exists, and coupled with assumption $[\mathcal{A}_4]$, it ensures that the average of both steps does not diverge at the limit. Finally, assumption $[\mathcal{A}_6]$ ensures an overall bound for γ. Therefore, these conditions ensure minimal influence of the window ρ_i outside on any interval I_j that is not a neighbor of i ($j \notin \mathcal{N}(i)$). The following definition generalizes the 1D window to a multidimensional separable window:

Definition 3 (Multidimensional Window Function). *Let $I_{\mathbf{i}}$ be a d-dimensional hypercube of length δ. Let $\{\rho^l_{i_l}\}_l$ be a set of 1D windows of sharpness ϵ as in def. 2 for 1D windows of length δ. Let $\Phi_{\mathbf{i}}$ be the separable multidimensional window defined using the 1D windows of def. 2:*

$$\Phi_{\mathbf{i}}(\mathbf{x}) = \prod_{l=1}^{d} \rho^l_{i_l}(x_l).$$

Lemma 1. *The windows of def. 3 verify:*

$$\forall \mathbf{x} \in \mathbb{R}^d, \quad \textstyle\sum_{\mathbf{i}} \Phi_{\mathbf{i}}(\mathbf{x}) = 1.$$

Proof. Using the separability of $\Phi_{\mathbf{i}}$ and summing over all the windows we obtain:

$$\sum_{\mathbf{i}} \Phi_{\mathbf{i}}(\mathbf{x}) = \sum_{\mathbf{i}} \prod_{l=1}^{d} \rho^l_{i_l}(x_l) = \prod_{l=1}^{d} \sum_{i_l=1}^{N_l} \rho^l_{i_l}(x_l) = \prod_{l=1}^{d} \frac{1}{2}\left(\gamma(x - \bar{a}^l_0) + \gamma(\bar{a}^l_{N_l} - x)\right) = 1,$$

where we used assumption $[\mathcal{A}_4]$ to cancel terms in successive windows, and assumption $[\mathcal{A}_5]$ or the last equality. □

In the next section, we study the peculiar case of `tanh`-based step function. Additionally, in appendix C, we introduce a novel piecewise polynomial step function verifying the constraints of def. 1, even for $\epsilon = 0$.

4.2 tanh-Based Step Function

We now devise a step function $\gamma(y) = \sigma(\alpha y)$ built from the hyperbolic tangent function $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, with $\alpha > 0$, so that the properties in def. 1 are satisfied.

This function verifies assumptions $[\mathcal{A}_4]$ to $[\mathcal{A}_6]$ while the remaining assumptions of def. 1 are met for an appropriate choice of α depending on δ and ϵ. Here, we improve upon [5] by proposing two novel methods to achieve smaller values of α. Proofs are deferred to appendix B.

Theorem 3. *Let $K > 0$. For any $\xi \in (0, 2)$, the step function $\gamma(y) = \sigma(\alpha y)$ verifies all hypotheses of def. 1 with α chosen as:*

$$\alpha = \frac{1}{\delta} \max\left(R_K, \kappa_K, \frac{1}{2}\ln\left[\frac{2-\epsilon}{\epsilon}\right], \frac{1}{\xi}\ln\left[\frac{2}{\epsilon}\left(\frac{2K}{(2-\xi)e\delta}\right)^K\right]\right).$$

The case $\xi = 1$ roughly corresponds to [5, eq. (A.21)]. However, since our upper bound for $\sigma^{(k)}$ is tighter than [5], the value of α given by thm. 3 is smaller. For example, for $K = 2$, $\delta = 0.1$, and $\epsilon = 0.1$ we obtain $\alpha = 83.8$, while [5] gives $\alpha = 97.6$, and for $K = 4$, we obtain $\alpha = 165.3$ vs $\alpha = 220.7$.

Notice also the range of x on which the bound is verified: They are identical for $k \leqslant 5$ since $\kappa_5 = -\infty$, but our bound is more restrictive for larger values of k. PDEs of order larger than 4 are however not often encountered in practice.

Remark 2. The value of α given in [4] is incorrect. For example, when $\delta = 0.1$ ($N = 10$ using [4, lem. B.4] notation) and $\epsilon = 0.1$, [4, lem. B.4] yields $\alpha \approx 62.95$, which in turn yields $\alpha^2|\sigma^{(2)}(\alpha\delta)| \approx 0.108 > \epsilon$, thus violating the constraints set to choose α.

Below, we propose a novel approach to define α, where we avoid to bound $x^k \exp(-x)$ to get a tighter bound on the derivatives of the `tanh` step function γ, and a smaller value of α.

Theorem 4. *The step function $\gamma(y) = \sigma(\alpha y)$ verifies all hypotheses of def. 1 with α chosen as:*

$$\alpha = \frac{1}{\delta} \max\left(R_K, \kappa_K, \frac{1}{2}\ln\left[\frac{2-\epsilon}{\epsilon}\right], -\frac{K}{2} W_{-1}\left(-\sqrt[K]{\frac{\epsilon}{2}}\frac{\delta}{K}\right)\right),$$

with $0 < \epsilon \leqslant 2\left(\frac{K}{e\delta}\right)^K$, and where W_{-1} is the -1 *branch of the* Lambert W Function.

The condition on ϵ above is actually mild. For example, for $K = 2$ (*e.g.* Navier-Stokes) and $\delta = 1$, the upper bound is $\epsilon \leqslant 1.08$, and if $\delta < 0.2$ (condition stated in [4]), we have $\epsilon \leqslant 759.5$. The smallest condition on ϵ is for $K = 1$ and $\delta = 1$ with $\epsilon \leqslant 0.735$.

5 Novel tanh Derivatives Analysis

In this section, we derive new properties for the hyperbolic tangent activation function that plays a key role in PINNs. At the core of this functional analysis lies the classical ODE solved by this function: $\sigma'(x) = 1 - \sigma^2(x)$. The theorem below leverages it to derive a novel expression for the derivatives of the `tanh` at any order. We then derive an overall upper bound for the magnitude of these derivatives (lem. 3), and an asymptotic upper bound for x large enough (thm. 6).

Theorem 5. *Let $k \geqslant 1$. The k^{th} derivative of the* tanh *function verifies:*

$$\forall x \in \mathbb{R}, \quad \sigma^{(k)}(x) = (-2)^{k-1}\sigma(x)p_k(\sigma(x)), \tag{20}$$

where $p_k[X]$ is a polynomial of degree $k-1$ verifying:

$$p_k[X] = \begin{cases} 1 & \text{if } k = 1, \\ Xp_{k-1}[X] - \frac{1}{2}(1-X^2)p'_{k-1}[X] & \text{otherwise.} \end{cases} \tag{21}$$

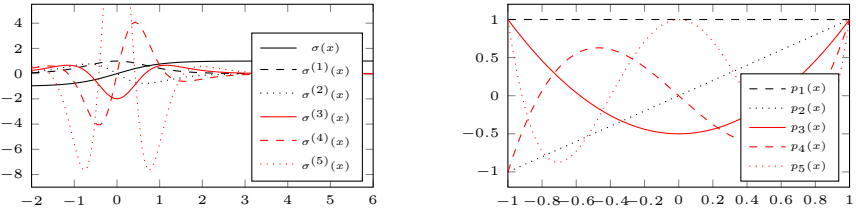


Fig. 2: Graphs of (left) the hyperbolic tangent and its derivatives, and (right) of the polynomials p_k defined in thm. 5.

Proof. The relation is trivially true for $k = 1$.

Let us suppose that the relation is true for $k - 1$.

$$\begin{aligned}\sigma^{(k)}(x) &= \frac{d\sigma^{(k-1)}}{dx}(x)\\ &= (-2)^{k-2}\left[\sigma''(x)p_{k-1}(\sigma(x)) + \sigma'(x)\sigma'(x)p'_{k-1}(\sigma(x))\right]\\ &= (-2)^{k-2}\left[-2\sigma'(x)\sigma(x)p_{k-1}(\sigma(x)) + \sigma'(x)^2 p'_{k-1}(\sigma(x))\right]\\ &= (-2)^{k-1}\sigma'(x)\big[\sigma(x)p_{k-1}(\sigma(x)) - \tfrac{1}{2}(1-\sigma^2(x))p'_{k-1}(\sigma(x))\big] \qquad (22)\end{aligned}$$

where we used the relation $\sigma' = 1 - \sigma^2$ for the third equality. Therefore, we have $\deg(p_k) = \deg(p_{k-1}) + 1 = k - 1$ and the desired recurrence relation. □

Lemma 2. *For any $k \geqslant 1$, we have $p_k(1) = 1$ and $p'_k(1) = 2^{k-1} - 1$.*

Proof. The property is trivially true for $k = 1$ since $p_1[X] = 1$. Suppose that $p_{k-1}(1) = 1$ and $p'_{k-1}(1) = 2^{k-2} - 1$. Then $p_k(1) = 1p_{k-1}(1) - 0 = 1$, and

$$\begin{aligned}p'_k(1) &= p_{k-1}(1) + p'_{k-1}(1) - \tfrac{1}{2}\left(-2p'_{k-1}(1) + 0p''_{k-1}(1)\right)\\ &= 1 + 2p'_{k-1}(1) = 2^{k-1} - 1.\end{aligned}$$

□

Lemma 3. *For any $k \geqslant 1$, let $\mathcal{P}_k = \max_{[-1,1]} |p_k|$. It holds:*

$$\forall x \in \mathbb{R}, \quad |\sigma^{(k)}(x)| \leqslant 2^{k+1} \min(e^{-2x}, e^{2x})\mathcal{P}_k.$$

Proof. Using assumption $[\mathcal{A}_4]$, we first observe that $|\sigma'(x)| \leqslant 2^2 \min(e^{-2x}, e^{2x})$, such that $|\sigma^{(k)}| \leqslant 2^{k+1} \min(e^{-2x}, e^{2x})|p_k(\sigma(x))|$. Finally, the statement follows from $\forall x, \sigma(x) \in (-1, 1)$. □

Lemma 4. *For any $k \geqslant 1$, it holds:*

$$\forall x \in \mathbb{R}, \quad |\sigma^{(k)}(x)| \leqslant 2^{k-1}\mathcal{P}_k.$$

Proof. Immediate after observing that $|\sigma'(x)| \leqslant 1$. □

Remark 3. We also have $|\sigma''(x)| \leqslant 1$ [4, lem. B.3], however, this does generalize since $|\sigma^{(3)}(0.4)| > 2^2$.

Table 1: Values of p_k and its roots (thm. 5), and the corresponding R_k depending on k (thm. 7). Values of κ_k (thm. 6) are obtained through computation of the extrema of p_k (none in $(-1,1)$ for $k \leqslant 5$, with $k=6$ the first for which $\kappa_k \neq -\infty$).

k	p_k	κ_k	$\mathcal{P}_k$	Roots p_k	R_k
1	1	$-\infty$	1	$\varnothing$	0
2	X	$-\infty$	1	$\{0\}$	$\operatorname{atanh}(1/\sqrt{3})$
3	$\frac{1}{2}(3X^2-1)$	$-\infty$	1	$\{\pm 1/\sqrt{3}\}$	$\operatorname{atanh}(\sqrt{2/3})$
4	$X(3X^2-2)$	$-\infty$	1	$\{0, \pm\sqrt{2/3}\}$	$\operatorname{atanh}(\sqrt{1/2+\sqrt{105}/30})$
5	$\frac{15}{2}X^2(X^2-1)+1$	$-\infty$	1	$\{\pm\sqrt{1/2 \pm \sqrt{105}/30}\}$	$\operatorname{atanh}(\sqrt{2/3+1/\sqrt{15}})$
6	$\frac{15}{2}X(X^2-1)(3X^2-1)+X$			$\{0, \pm\sqrt{2/3 \pm 1/\sqrt{15}}\}$	

Theorem 6. *For any $k \geqslant 1$, there exists $\kappa_k \in \mathbb{R} \cup \{-\infty\}$ such that:*

$$\forall x \in (\kappa_k, +\infty), \quad |\sigma^{(k)}(x)| \leqslant 2^{k+1} \min(e^{-2x}, e^{2x}).$$

Proof. Using lem. 3, we are left to bound the magnitude of p_k. Since $p_1(x) = 1$ is constant, choosing $\kappa_1 = -\infty$ verifies the statement. For $k > 1$, since p_k is continuously differentiable and $p_k'(1) > 0$ (lem. 2), then there exists $\rho < 1$ such that $p_k'(\rho) > 0$. Therefore, p_k is strictly increasing on $[\rho, 1]$, and for any $\theta \in [\rho, 1)$, we have $p_k(\theta) < p_k(1) = 1$ (lem. 2), and there exists $\theta \in [\rho, 1)$ such that we also have $p_k(\theta) > -1$. Choosing $\kappa_k = \sigma^{-1}(\max(-1, \theta))$ is enough. □

Remark 4. For $k \leqslant 5$, thm. 6 gives an upper bound to the k^{th} derivative of tanh for $x \in \mathbb{R}$, thus strictly improving the bound [5, lem. A.4] by a factor of k^{k+1} (8 for $k=2$, 81 for $k=3$, and 1024 for $k=4$). For larger values of k, the two results are not comparable since our upper bound is not valid for any value of x. However, PDEs of order 6 or more are not often encountered in practice.

Theorem 7. *Let $k \geqslant 1$. Let y_k be the largest zero of p_{k+1} on $[0,1)$, or 0 if it has none in this interval. Then $\sigma^{(k)}$ is monotonous on $[R_k, +\infty)$ with $R_k = \operatorname{atanh}(y_k)$.*

Proof. Finding R_k such that $\sigma^{(k)}$ is monotonous on $[R_k, \infty)$ is equivalent to finding the largest zero of its derivative $\sigma^{(k+1)}$. Using thm. 5, $\sigma^{(k+1)}(x) = (-2)^k \sigma'(x) p_{k+1}(\sigma(x))$, and its zeros coincides with the zeros of $p_{k+1}(\sigma(x))$ since σ' is positive. Therefore, either p_{k+1} has no zero on $[0,1)$, in which case $\sigma^{(k)}$ is monotonous on $[0, +\infty)$, or we can choose y_k to be its largest zero in $[0,1)$. Setting $R_k = \operatorname{atanh}(y_k)$ ensures that $\sigma^{(k+1)}$ has no zero in (R_k, ∞), and $\sigma^{(k)}$ is monotonous on this interval. □

6 Conclusion

In this paper, we presented a series of theoretical contributions for the analysis of the approximation error of `tanh` neural networks adapted to PINNs. We notably provided a tighter approximation bound that stands for Sobolev-regular

functions at any order, and instantiated it for the case of Navier-Stokes PDEs. This bound comes with a decrease of the required size of the network. Our analysis is general enough to consider possible extensions of these results to different activation functions and a large set of PDE-based models of physical phenomena.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bajaj, C., McLennan, L., Andeen, T., Roy, A.: Recipes for when physics fails: recovering robust learning of physics informed neural networks. Machine Learning: Science and Technology **4**(1), 015013 (feb 2023). https://doi.org/10.1088/2632-2153/acb416
2. Constantine, G.M., Savits, T.H.: A multivariate Faa di Bruno formula with applications. Transactions of the American Mathematical Society **348**(2), 503–520 (1996). https://doi.org/10.1090/S0002-9947-96-01501-2
3. Corless, R.M., Gonnet, G.H., Hare, D.E., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. Advances in Computational mathematics **5**, 329–359 (1996). https://doi.org/10.1007/BF02124750
4. De Ryck, T., Jagtap, A.D., Mishra, S.: Error estimates for physics-informed neural networks approximating the Navier–Stokes equations. IMA Journal of Numerical Analysis p. drac085 (01 2023). https://doi.org/10.1093/imanum/drac085
5. De Ryck, T., Lanthaler, S., Mishra, S.: On the approximation of functions by tanh neural networks. Neural Networks **143**, 732–750 (2021)
6. Doumèche, N., Biau, G., Boyer, C.: Convergence and error analysis of pinns (2023)
7. Gühring, I., Kutyniok, G., Petersen, P.: Error bounds for approximations with deep relu neural networks in $w^{s,p}$ norms (2019)
8. Hao, Z., Liu, S., Zhang, Y., Ying, C., Feng, Y., Su, H., Zhu, J.: Physics-informed machine learning: A survey on problems, methods and applications (2023)
9. Herrmann, L., Opschoor, J.A.A., Schwab, C.: Constructive deep relu neural network approximation. J. Sci. Comput. **90**(1), 75 (2022)
10. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5), 359–366 (1989)
11. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nature Reviews Physics **3**(6) (5 2021). https://doi.org/10.1038/s42254-021-00314-5
12. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics **378**, 686–707 (2019)
13. Shin, Y.: On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. Communications in Computational Physics **28**(5), 2042–2074 (Jun 2020)
14. Shin, Y., Zhang, Z., Karniadakis, G.E.: Error estimates of residual minimization using neural networks for linear pdes (2023)
15. Wu, S., Zhu, A., Tang, Y., Lu, B.: Convergence of physics-informed neural networks applied to linear second-order elliptic interface problems (2023)
16. Yarotsky, D.: Error bounds for approximations with deep relu networks. Neural Networks **94**, 103–114 (2017). https://doi.org/10.1016/j.neunet.2017.07.002

A Proofs for Asymptotic Guarantees of $\boldsymbol{\Delta}$

Proof (cor. 1). If using thm. 3, and assuming δ small enough for α being defined by the last term in the max, we obtain using the value of ϵ given by thm. 1:

$$\alpha^k \delta^m = \delta^{m-k} \frac{1}{\xi^k} \ln^k \left(\frac{2\mu(\Omega)(2K)^K}{3^d(2-\xi)e} \frac{1}{\delta^{m+d+K}} \right)$$

The statement follows after refactoring the bound. □

Proof (cor. 2). Let $\delta > 0$ be small enough such that α is given by the last term in the max in thm. 4. Using the asymptotic formula of W_{-1} from [3], we have the following asymptotic relation for α as δ goes to 0:

$$\alpha = \frac{-K}{2\delta} \left(L_1 - L_2 + O\left(\frac{L_2}{L_1} \right) \right) \tag{23}$$

where $L_1 = \ln(\sqrt[K]{\frac{\epsilon}{2}} \frac{\delta}{K})$ and $L_2 = \ln(-L_1)$. Further developing these values with $\epsilon = \frac{3^d}{\mu(\Omega)} \delta^{m+d}$ as in thm. 1 yields:

$$L_1 = \left(C := \ln \left(\frac{1}{K} \sqrt[K]{\frac{3^d}{2\mu(\Omega)}} \right) \right) + \left(1 + \frac{m+d}{K} \right) \ln(\delta) \tag{24}$$

$$L_2 = \ln \left(1 + \frac{m+d}{K} \right) + \ln(-\ln(\delta)) + \ln \left(1 + \frac{KC}{(m+d+K)} \frac{1}{\ln(\delta)} \right). \tag{25}$$

In particular, we obtain as $\delta \to 0$:

$$\frac{L_2}{L_1} \sim \frac{K}{m+d+K} \frac{\ln(-\ln(\delta))}{\ln(\delta)} \xrightarrow[\delta \to 0]{} 0. \tag{26}$$

This yields:

$$\alpha = \frac{\text{-}K}{2\delta} \left[\frac{m+d+K}{K} \ln(\delta) - \ln(\text{-}\ln(\delta)) + B^K_{d,m}(\Omega) + O\left(\frac{\ln(\text{-}\ln(\delta))}{\ln(\delta)} \right) \right] \tag{27}$$

$$B^K_{d,m}(\Omega) = \ln \left(\frac{1}{m+d+K} \sqrt[K]{\frac{3^d}{2\mu(\Omega)}} \right), \tag{28}$$

and the asymptotic behavior follows. □

B Proofs for Tanh-Based Step Functions

Proof (thm. 3). Choosing $\delta\alpha \geqslant R_K$, assumption $[\mathcal{A}_1]$ is verified using thm. 7 for derivatives of order $k \geqslant 1$, while $\sigma(x)$ is increasing on $\mathbb{R}$.

For assumption $[\mathcal{A}_2]$ to be verified, it is enough to have:

$$1 - \gamma(\delta) = \frac{2e^{-2\alpha\delta}}{1 + e^{-2\alpha\delta}} \leqslant \epsilon. \tag{29}$$

This condition is verified with the third term in the max of the theorem statement.

Let $1 \leqslant k \leqslant K$, $\xi \in (0,2)$, and $\beta = \alpha\delta \in (\kappa_k, +\infty)$, then:

$$\begin{aligned}
|\gamma^{(k)}(\delta)| &= (\beta/\delta)^k |\sigma^{(k)}(\beta)| \\
&\leqslant 2^{k+1}(\beta/\delta)^k \exp(-2\beta) \\
&= \frac{2^{k+1}}{\delta^k(2-\xi)^k} \left((2-\xi)\beta\right)^k \exp(-(2-\xi)\beta) \exp(-\xi\beta) \\
&\leqslant \frac{2^{k+1}}{\delta^k(2-\xi)^k} k^k e^{-k} \exp(-\xi\beta) \qquad (30)
\end{aligned}$$

where we used thm. 6 for the first inequality, and $\max_{x \geqslant 0} x^k \exp(-x) \leqslant k^k e^{-k}$ for the last inequality. Therefore, for assumption $[\mathcal{A}_3]$ to be verified, it is enough to choose β such that:

$$\frac{2^{k+1}}{\delta^k(2-\xi)^k} k^k e^{-k} \exp(-\xi\beta) \leqslant \epsilon \qquad (31)$$

which is the case for the third term in the max above. □

Proof (thm. 4). Proof of thm. 3 already shows that this value of α is such that assumptions $[\mathcal{A}_1]$, $[\mathcal{A}_2]$ and $[\mathcal{A}_4]$ to $[\mathcal{A}_6]$ are verified.

Let $k \geqslant 1$ and $\beta = \alpha\delta$. Using thm. 6, it is then enough to choose $\beta \geqslant R_k$ such that:

$$|\gamma^{(k)}(\delta)| = (\beta/\delta)^k |\sigma^{(k)}(\beta)| \leqslant 2^{k+1}(\beta/\delta)^k \exp(-2\beta) = \epsilon. \qquad (32)$$

We rewrite this equality in the form $x\exp(x) = y$:

$$\left(-\frac{2}{k}\beta\right) \exp\left(-\frac{2}{k}\beta\right) = \sqrt[k]{\frac{\epsilon}{2}} \frac{-\delta}{k} \qquad (33)$$

which is solved using Lambert W Function of parameter -1:

$$\beta = -\frac{k}{2} W_{-1}\left(-\sqrt[k]{\frac{\epsilon}{2}} \frac{\delta}{k}\right) \qquad (34)$$

where parameter -1 to the Lambert W function is chosen since W_p must be decreasing for β to increase as ϵ goes to 0. To complete the proof, note that W_{-1} is defined only on $[-1/e, \infty)$, giving the condition $\epsilon \leqslant 2\,(k/e\delta)^k$. □

C Piecewise Polynomial Step Alternative to Tanh

Our goal in this section is to create a piecewise polynomial window using def. 2 and a step function verifying all assumptions of def. 1. Let γ be the piecewise step function verifying:

$$\gamma(y) = \begin{cases} -1 & \text{if } x < -\eta/2 \\ g_K(2y/\eta) & \text{if } x \in [-\eta/2, \eta/2] \\ 1 & \text{if } x > \eta/2, \end{cases} \qquad (35)$$

Table 2: Coefficients $g_{p,K}$ ($p \leqslant K$) of g_K for $K \leqslant 5$.

K	0	1	2	3	4	5
0	1					
1	$3/2$	$-1/2$				
2	$15/8$	$-5/4$	$3/8$			
3	$35/16$	$-35/16$	$21/16$	$-5/16$		
4	$315/128$	$-105/32$	$189/64$	$-45/32$	$35/128$	
5	$693/256$	$-1155/256$	$693/128$	$-495/128$	$385/256$	$-63/256$

with the constraints that γ and its derivatives up to order K are continuous, especially for $y = \pm\eta/2$, and that g_K is an odd polynomial function. This last condition ensures that assumption $[\mathcal{A}_4]$ is verified with $\epsilon = 0$. If $2\eta < \delta$, then $\gamma(y)$ for $|y| > \delta$, γ is constant, and assumptions $[\mathcal{A}_1]$ to $[\mathcal{A}_3]$ and $[\mathcal{A}_5]$ are verified. Assumption $[\mathcal{A}_6]$ is left as a conjecture. We now show that how to build the polynomial g_K from these constraints.

Oddness and continuity of γ and its derivatives are then ensured with the following conditions:

- $g_K(-z) = -g_K(z)$,
- $g_K(-1) = -1$,
- $g_K(1) = 1$,
- $g_K^{(k)}(\pm 1) = 0$, for any $1 \leqslant k \leqslant K$.

Since g_K is a polynomial, the first condition is equivalent to g_K having no coefficient of even degree:

$$g_K(z) = z \sum_{p=0}^{P} g_{p,K} z^{2p}. \tag{36}$$

We notice that the remaining conditions, for a given $0 \leqslant k \leqslant K$, form couples of equivalent conditions ($g_K^{(k)}(-1) = 0 \Leftrightarrow g_K^{(k)}(1) = 0$). Therefore, we have a system of $K+1$ equations with $P+1$ unknowns whose solution is a spline. Choosing $P = K$ is then enough. Table 2 gives the exact polynomial coefficients up to $K = 5$.

D Sobolev Norm of Compositions, with Windows

The following lemma improves upon [5, lem. A.7] for the specific case of approximating the composition of two functions where the innermost one has all but one output that are univariate. This corresponds to the second `tanh` layer of neural network construction in [5]. In addition, the more complex expression for the constant yields in this case an improvement of several orders of magnitude. In the particular case of a 2D Navier-Stockes PDE solution ($d = 3$, $K = 2$), we get 3 orders of magnitude ($1.3 \cdot 10^5$ instead of $2.9 \cdot 10^8$) in addition to the power 1 instead of k on the max term.

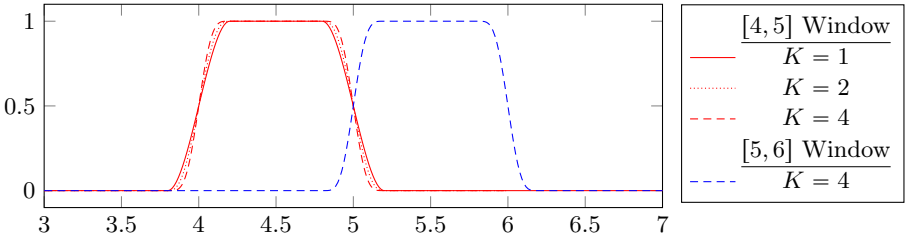


Fig. 3: Polynomial windows with $\eta = 0.4$, $\delta = 1$.

Lemma 5. *Let $\mathcal{X}$ be a d-dimensional space, and $\mathcal{Y}$ a $d+1$-dimensional space. Let $f \in \mathcal{C}^k(\mathcal{Y}, \mathbb{R})$ and $g = [g^{[1]}, \dots, g^{[d]}, g^{[d+1]}] \in \mathcal{C}^k(\mathcal{X}, \mathcal{Y})$ such that for the first d components $g^{[l]}, l \leqslant d$ of g verify $g^{[l]}(\mathbf{x}) = \rho_l(x_l)$, for some functions $\rho_l \in \mathcal{C}^k(\mathbb{R}, \mathbb{R})$. We have then*

$$\|f \circ g\|_{W^{k,\infty}(\mathcal{X})} \leqslant C_{k,d}^{[win]} \|f\|_{W^{k,\infty}(\mathcal{Y})} \max\left(\max_{1 \leqslant l \leqslant d} \|g^{[l]}\|_{W^{k,\infty}(\mathbb{R})}^k, \|g^{[d+1]}\|_{W^{k,\infty}(\mathcal{X})}^k \right)$$

where the constant $C_{k,d}^{[win]}$ (with "win" for window) is upper bounded with:

$$C_{k,d}^{[win]} \leqslant k! |P_{k,k}| |P_{k,d+2}| \sum_{i=0}^{k} \binom{k}{i} (d(k+1))^{k-i} |P_{k,(d-1)i+k+1}|. \tag{37}$$

Proof. This proof follows the same idea of [5, lem. A.7], but taking into account the specific properties of g in the composition $h = f \circ g$.

We start with the multivariate *Faà di Bruno's Formula* from [2]:

$$D^{\mathfrak{a}}(f \circ g) = (\mathfrak{a}!) \sum_{1 \leqslant |\mathfrak{b}| \leqslant k} D^{\mathfrak{b}} f \sum_{p(\mathfrak{a},\mathfrak{b})} \prod_{j=1}^{k} \frac{(D^{\mathfrak{l}^{[j]}} g)^{\mathfrak{r}^{[j]}}}{(\mathfrak{r}^{[j]}!)(\mathfrak{l}^{[j]}!)^{|\mathfrak{r}^{[j]}|}} \tag{38}$$

where the set $p(\mathfrak{a}, \mathfrak{b})$ is defined as:

$$p(\mathfrak{a}, \mathfrak{b}) = \Big\{ (\mathfrak{r}^{[1]}, \dots, \mathfrak{r}^{[k]}, \mathfrak{l}^{[1]}, \dots, \mathfrak{l}^{[k]}) : \exists 1 \leqslant s \leqslant k \tag{39}$$

$$\forall 1 \leqslant j \leqslant k - s, \mathfrak{r}^{[j]} = \mathbf{0} \text{ and } \mathfrak{l}^{[j]} = \mathbf{0} \tag{40}$$

$$\forall k - s + 1 \leqslant j \leqslant k, |\mathfrak{r}^{[j]}| > 0 \tag{41}$$

$$\mathbf{0} \prec \mathfrak{l}^{[k-s+1]} \prec \dots \prec \mathfrak{l}^{[k]} \tag{42}$$

$$\sum_{j=1}^{k} \mathfrak{r}^{[j]} = \mathfrak{b} \text{ and } \sum_{j=1}^{k} |\mathfrak{r}^{[j]}| \mathfrak{l}^{[j]} = \mathfrak{a} \Big\} \tag{43}$$

where $\mathfrak{e} \prec \mathfrak{f}$ when either $|\mathfrak{e}| < |\mathfrak{f}|$, or $\mathfrak{e}$ precedes $\mathfrak{f}$ according to the lexicographic order, and $\mathbf{x}^{\mathfrak{e}} = (x_1^{\mathfrak{e}_1}, \dots)$.

We observe in the product above that anytime one of the term is zero, the whole product is zero, such that our goal is to find the corresponding elements of

$p(\mathfrak{a},\mathfrak{b})$ that lead to term equal to zero. Let j be such that $|\mathfrak{r}^{[j]}|>0$. Let $p\leqslant d$ be an index such that $[\mathfrak{r}^{[j]}]_p>0$. Then, for the whole product to be non-zero, we need $D^{\mathfrak{l}^{[j]}}g^{[p]}$ to be non-zero. Given that $g^{[p]}$ depends only on x_p, this implies that $\mathfrak{l}^{[j]}_q=0$ for any $q\neq p$. Therefore:

$$\mathfrak{r}^{[j]}=(0,\dots,0,\mathfrak{r}^{[j]}_{d+1}) \qquad \text{or} \qquad \mathfrak{l}^{[j]}=(0,\dots,0,\mathfrak{l}^{[j]}_p,0,\dots,0) \tag{44}$$

Moreover, since $|\mathfrak{r}^{[j]}|>0$, then $|\mathfrak{l}^{[j]}|>0$ (ordering constraint), such that ${\mathfrak{l}^{[j]}}_p>0$. Therefore, if there were $q\neq p$ such that $[\mathfrak{r}^{[j]}]_p>0$, then $D^{\mathfrak{l}^{[j]}}g^{[q]}=\partial^{\mathfrak{l}^{[j]}_p}g^{[q]}/\partial x_p^{\mathfrak{l}^{[j]}_p}=0$, which would lead to a zero term in the product. Therefore:

$$\begin{cases}\mathfrak{r}^{[j]}=(0,\dots,0,\mathfrak{r}^{[j]}_{d+1})\\ \mathfrak{l}^{[j]}\text{ unconstrainted}\end{cases} \quad \text{or} \quad \begin{cases}\mathfrak{r}^{[j]}=(0,\dots,0,\mathfrak{r}^{[j]}_{p},0,\dots,0,\mathfrak{r}^{[j]}_{d+1})\\ \mathfrak{l}^{[j]}=(0,\dots,0,\mathfrak{l}^{[j]}_p,0,\dots,0)\end{cases} \tag{45}$$

Using the property $\sum_{j=1}^k|\mathfrak{r}^{[j]}|=|\mathfrak{b}|\leqslant k$, that there are $P_{|\mathfrak{b}|,k}\leqslant P_{k,k}$ choices for $\{|\mathfrak{r}^{[j]}|\}_j$. For each of these choices, there are at most $(1+d)^k$ choices for $\{p_j\}_j$ (with $p_j=0$ corresponding to left choice type above). Finally, for each of these choices where there is a value for p_j, there are $|\mathfrak{r}^{[j]}|\leqslant k$ choices for $r^{[j]}_{p_j}$. We obtain an upper bound $P_{k,k}(1+dk)^k$ on the number of values for $\mathfrak{r}=(\mathfrak{r}^{[1]},\dots,\mathfrak{r}^{[d+1]})$. Developing the power, we obtain a sum where $i=|\{j:p_j=0\}_j|$:

$$|\{\mathfrak{r}\}|\leqslant P_{k,k}\sum_{i=0}^{k}\binom{k}{i}\left(d(k+1)\right)^{k-i}. \tag{46}$$

For $\mathfrak{l}=(\mathfrak{l}^{[1]},\dots,\mathfrak{l}^{[d]})$, when there are i unconstrained choices for $\mathfrak{l}^{[j]}$, and $k-i$ choices $\mathfrak{l}^{[j]}$ where they are almost all zero except for entry p_j, we have $di+k-i$ non-zero coefficients to choose for $\mathfrak{l}$, with all coefficients location being constrained. In addition, the last summation constraint in the definition of $p(\mathfrak{a},\mathfrak{b})$ implies that $\sum_{j=1}^k|\mathfrak{l}^{[j]}|\leqslant k$, such that $|\mathfrak{l}|\leqslant k$. Therefore, the number of choices for $\mathfrak{l}$ is upper bounded by

$$\sum_{t=1}^{k}P_{t,di+k-i}=P_{k,di+k-i+1}-1\leqslant P_{k,(d-1)i+k+1}. \tag{47}$$

This yields the following upper bound:

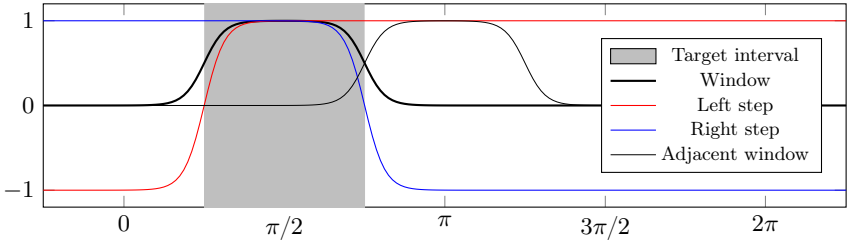
$$|p(\mathfrak{a},\mathfrak{b})|\leqslant P_{k,k}\sum_{i=0}^{k}\binom{k}{i}\left(d(k+1)\right)^{k-i}P_{k,(d-1)i+k+1} \tag{48}$$

The rest of the proof follows [5, lem. A.7], using $|\{\mathfrak{b}:1\leqslant|\mathfrak{b}|\leqslant k\}|\leqslant P_{k,d+2}$, $\mathfrak{a}!\leqslant k!$, $\max_{\mathcal{X}}|D^{\mathfrak{b}}f|\leqslant\|f\|_{W^{k,\infty}(\mathcal{Y})}$, and

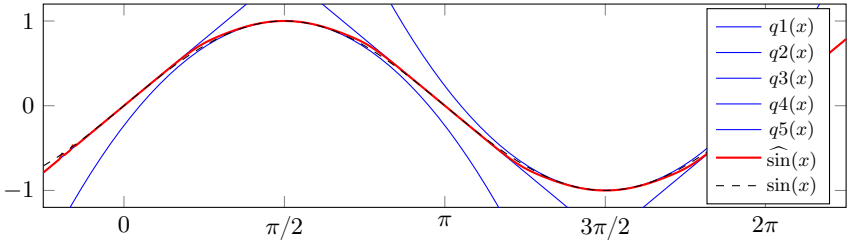
$$\left|\prod_{j=1}^{k}\frac{(D^{\mathfrak{l}^{[j]}}g)^{\mathfrak{r}^{[j]}}}{(\mathfrak{r}^{[j]}!)(\mathfrak{l}^{[j]}!)^{|\mathfrak{r}^{[j]}|}}\right|\leqslant\Big(\max_{1\leqslant l\leqslant d+1}\|g^{[l]}\|_{W^{k,\infty}(\mathcal{X})}\Big)^k \tag{49}$$

$$\leqslant\max\Big(\max_{1\leqslant l\leqslant d}\|g^{[l]}\|_{W^{k,\infty}(\mathbb{R})},\|g^{[d+1]}\|_{W^{k,\infty}(\mathcal{X})}\Big)^k \tag{50}$$

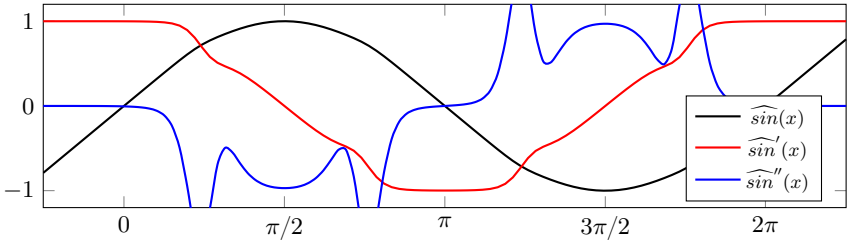
using the property $|\sum_{j=1}^k\mathfrak{r}^{[j]}|=|\mathfrak{b}|\leqslant k$. □



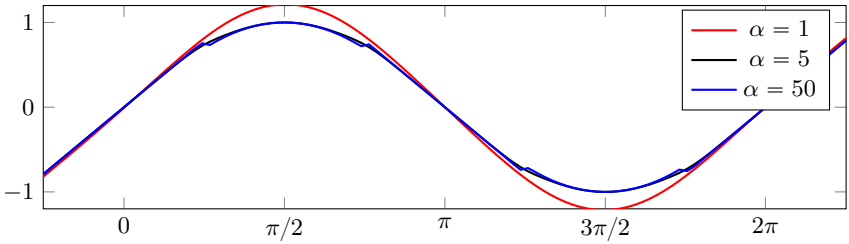
(a) Window construction ($\alpha = 5$).



(b) Sine approximation.



(c) Approximation $\widehat{\sin}$ and its derivatives.



(d) Approximation $\widehat{\sin}$ for various values of α.

Fig. 4: Construction of a tanh window $\Phi_2(x)$ from averaging two tanh steps, with the other windows displayed with thin lines. Approximation $\widehat{\sin}(x) = \sum_i q_i(x)\Phi_i(x)$ of one period of the sine function, using 5 windows of length $\pi/2$, and polynomial approximations q_i of degree at most 2 from the series of sine and cosine.