

Solving PDEs by variational physics-informed neural networks: an a posteriori error analysis

*Original*

Solving PDEs by variational physics-informed neural networks: an a posteriori error analysis / Berrone, S., Canuto, C., Pintore, M.. - In: ANNALI DELL'UNIVERSITÀ DI FERRARA. SEZIONE 7: SCIENZE MATEMATICHE. - ISSN 0430-3202. - ELETTRONICO. - 68:(2022), pp. 575-595. [10.1007/s11565-022-00441-6]

*Availability:*

This version is available at: 11583/2972327 since: 2022-10-14T14:04:23Z

*Publisher:*

Springer

*Published*

DOI:10.1007/s11565-022-00441-6

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)



# Solving PDEs by variational physics-informed neural networks: an a posteriori error analysis

Stefano Berrone<sup>1</sup> · Claudio Canuto<sup>1</sup> · Moreno Pintore<sup>1</sup>

Received: 29 April 2022 / Accepted: 25 August 2022  
© The Author(s) 2022

## Abstract

We consider the discretization of elliptic boundary-value problems by variational physics-informed neural networks (VPINNs), in which test functions are continuous, piecewise linear functions on a triangulation of the domain. We define an a posteriori error estimator, made of a residual-type term, a loss-function term, and data oscillation terms. We prove that the estimator is both reliable and efficient in controlling the energy norm of the error between the exact and VPINN solutions. Numerical results are in excellent agreement with the theoretical predictions.

**Keywords** Deep neural networks · A posteriori error estimators · Petrov–Galerkin discretizations · Elliptic boundary-value problems

**Mathematics Subject Classification** 35A01 · 65L10 · 65L12 · 65L20 · 65L70

## 1 Introduction

The possibility of using deep-learning tools for solving complex physical models has attracted the attention of many scientists over the last few years. We have in mind in this paper models that are mathematically described by partial differential equations, supplemented by suitable boundary and initial conditions. In the most general setting, if no information on the model is available except the knowledge of some of its solutions, the model may be completely surrogated by one or more neural network,

---

✉ Claudio Canuto  
claudio.canuto@polito.it

Stefano Berrone  
stefano.berrone@polito.it

Moreno Pintore  
moreno.pintore@polito.it

<sup>1</sup> Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

trained by data (i.e., by the known solutions). However, in most situations of interest, the mathematical model is known (e.g., the Navier-Stokes equations describing an incompressible flow), and such information may be suitably exploited in training the network(s): one gets the so-called Physics Informed Neural Networks (PINNs). This approach was first proposed in [1], and it inspired further works such as e.g. [2] or [3], until the recent paper [4] which presents a very general framework for the solution of operator equations by deep neural networks. PINNs are trained by using the strong form of the differential equations, which are enforced at a set of points in the domain by suitably defining the loss function. In this sense, PINNs can be viewed as particular instances of least-square/collocations methods.

Based on the weak formulation of the differential model, the so-called Variational Physics-Informed Neural Networks (VPINNs), proposed in [5], enforce the equations by means of suitably chosen test functions, not necessarily represented by neural networks [6]; they are instances of least-square/Petrov–Galerkin methods. While the construction of the loss function is generally less expensive for PINNs than for VPINNs, the latter allow for the treatment of models with less regular solutions, as well as an easier enforcement of boundary conditions. In addition, the error analysis for VPINNs takes advantage of the available results for the discretization of variational problems, in fulfilling the assumptions of Lax-Richtmyer's theorem 'stability plus consistency imply convergence'. Actually, consistency results follow rather easily from the recently established approximation properties of neural networks in Sobolev spaces (see, e.g., [7–12]), whereas the derivation of stability estimates for the neural network solution appears to be a less trivial task: indeed, a neural network is identified by its weights, which are usually much more than the conditions enforced in its training. In other words, the training of a neural network is functionally an ill-posed problem.

To this respect, we considered in [13] a Petrov–Galerkin framework in which trial functions are defined by means of neural networks, whereas test functions are made of continuous, piecewise linear functions on a triangulation of the domain. Relying on an inf-sup condition between spaces of piecewise polynomial functions, we derived an a priori error estimate in the energy norm between the exact solution of an elliptic boundary-value problem and a high-order interpolant of a deep neural network, which minimizes the loss function. Numerical results indicate that the error follows a similar behavior when the interpolation operator is turned off.

The purpose of the present paper is to perform an a posteriori error analysis for VPINNs, i.e., to get estimates on the error which only depend on the computed VPINN solution, rather than the unknown exact solution. This is important to get a practical and quantitative information on the quality of the approximation. After setting the model elliptic boundary-value problem in Sect. 2, and the corresponding VPINN discretization in Sect. 2.1, we define in Sect. 3 a computable residual-type error estimator, and prove that it is both reliable and efficient in controlling the energy error between the exact solution and the VPINN solution (the so-called *generalization error* in the terminology of Learning Theory, see [14]). Reliability means that the global error is upper bounded by a constant times the estimator, efficiency means that the estimator cannot over-estimate the energy error, since the latter is lower bounded by a constant times the former up to data oscillation terms. The proposed estimator is

obtained by summing up several terms: one is the classical residual-type estimator in finite elements, measuring the bulk error inside each element of the triangulation as well as the inter-element gradient jumps; another term accounts for the magnitude of the loss function after minimization is performed (the so-called *optimization error*); the remaining terms measure data oscillations, i.e., the errors committed by locally projecting the equation's coefficients and right-hand side upon suitable polynomial spaces. The estimator can be written as a sum of elemental contributions, thereby allowing its use within an adaptive discretization strategy which refines the elements carrying the largest contributions to the estimator.

## 2 The model boundary-value problem

Let  $\Omega \subset \mathbb{R}^n$  be a bounded polygonal/polyhedral domain with Lipschitz boundary  $\Gamma = \partial\Omega$ .

Let us consider the model elliptic boundary-value problem

$$\begin{cases} Lu := -\nabla \cdot (\mu \nabla u) + \boldsymbol{\beta} \cdot \nabla u + \sigma u = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma, \end{cases} \tag{1}$$

where  $\mu, \sigma \in L^\infty(\Omega)$ ,  $\boldsymbol{\beta} \in (W^{1,\infty}(\Omega))^n$  satisfy  $\mu \geq \mu_0$ ,  $\sigma - \frac{1}{2} \nabla \cdot \boldsymbol{\beta} \geq 0$  in  $\Omega$  for some constant  $\mu_0 > 0$ , whereas  $f \in L^2(\Omega)$ .

Setting  $V = H_0^1(\Omega)$ , define the bilinear and linear forms

$$a : V \times V \rightarrow \mathbb{R}, \quad a(w, v) = \int_{\Omega} \mu \nabla w \cdot \nabla v + \boldsymbol{\beta} \cdot \nabla w v + \sigma w v, \tag{2}$$

$$F : V \rightarrow \mathbb{R}, \quad F(v) = \int_{\Omega} f v; \tag{3}$$

denote by  $\alpha \geq \mu_0$  the coercivity constant of the form  $a$ , and by  $\|a\|, \|F\|$  the continuity constants of the forms  $a$  and  $F$ . Problem (1) is formulated variationally as follows: Find  $u \in V$  such that

$$a(u, v) = F(v) \quad \forall v \in V. \tag{4}$$

**Remark 1** (Other boundary conditions) We just consider homogeneous Dirichlet conditions to keep technicalities at a minimum. However, the forthcoming formulation of the discretized problem and the a posteriori error analysis can be extended to cover the case of mixed Dirichlet-Neumann boundary conditions, namely  $u = g$  on  $\Gamma_D$ ,  $\mu \partial_n u = \psi$  on  $\Gamma_N$ , with  $\Gamma_D \cup \Gamma_N = \Gamma$ . We refer to [13, 15] for the general case.

### 2.1 The VPINN discretization

We aim at approximating the solution of Problem (1) by a generalized Petrov–Galerkin strategy.

To define the subset of  $V$  of trial functions, let us choose a fully-connected feed-forward neural network structure  $\mathcal{NN}$ , with  $n$  input variables and 1 output variable, identified by the number of layers  $L$ , the layer widths  $N_\ell$ ,  $\ell = 1, \dots, L$ , and the activation function  $\rho$ . Thus, each choice of the weights  $\mathbf{w} \in \mathbb{R}^N$  defines a mapping  $w^{\mathcal{NN}} : \mathbf{x} \mapsto w(\mathbf{x}, \mathbf{w})$ , which we think as restricted to the closed domain  $\bar{\Omega}$ ; let us denote by  $W^{\mathcal{NN}}$  the manifold containing all functions that can be generated by this neural network structure. Therefore, each function  $w \in W^{\mathcal{NN}}$  can be explicitly computed as:

$$\begin{aligned} x_0 &= \mathbf{x}, \\ x_\ell &= \rho(A_\ell x_{\ell-1} + b_\ell), \quad \ell = 1, \dots, L - 1, \\ w(\mathbf{x}) &= A_L x_{L-1} + b_L, \end{aligned} \tag{5}$$

where the matrices and vectors  $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$  and  $b_\ell \in \mathbb{R}^{N_\ell}$ ,  $\ell = 1, \dots, L$  are a suitable rearrangement of the weights in  $\mathbf{w}$ , with  $N_0 = n$  and  $N_L = 1$ . We enforce the homogeneous Dirichlet boundary conditions by multiplying each  $w$  by a fixed smooth function  $\Phi \in V$  (we refer to [15] for a general strategy to construct this function); we assume that  $v^{\mathcal{NN}} = \Phi w^{\mathcal{NN}}$  belongs to  $V$  for any  $w^{\mathcal{NN}} \in W^{\mathcal{NN}}$ . In conclusion, our manifold of trial functions will be

$$V^{\mathcal{NN}} = \{v^{\mathcal{NN}} \in V : v^{\mathcal{NN}} = \Phi w^{\mathcal{NN}} \text{ for some } w^{\mathcal{NN}} \in W^{\mathcal{NN}}\}.$$

To define the subspace of  $V$  of test functions, let us introduce a conforming, shape-regular triangulation  $\mathcal{T}_h = \{E_n : 1 \leq n \leq N_h\}$  of  $\bar{\Omega}$  with meshsize  $h > 0$ ; the generic element of the triangulation will be denoted by  $E$ . Let  $V_h \subset V$  be the linear subspace formed by the functions which are piecewise linear polynomials over the triangulation  $\mathcal{T}_h$ . Furthermore, let us introduce computable approximations of the forms  $a$  and  $F$  by numerical quadratures. Precisely, for any  $E \in \mathcal{T}_h$ , let  $\{(\xi_i^E, \omega_i^E) : i \in I^E\}$  be the nodes and weights of a quadrature formula of precision  $q \geq 2$  on  $E$ . Then, assuming that all data  $\mu, \boldsymbol{\beta}, \sigma, f$  are continuous in each element of the triangulation, we define the approximate forms

$$a_h(w, v) = \sum_{E \in \mathcal{T}_h} \sum_{i \in I^E} [\mu \nabla w \cdot \nabla v + \boldsymbol{\beta} \cdot \nabla w v + \sigma w v](\xi_i^E) \omega_i^E, \tag{6}$$

$$F_h(v) = \sum_{E \in \mathcal{T}_h} \sum_{i \in I^E} [f v](\xi_i^E) \omega_i^E. \tag{7}$$

With these ingredients at hand, we would like to approximate the solution of Problem (4) by some  $u^{\mathcal{NN}} \in V^{\mathcal{NN}}$  satisfying

$$a_h(u^{\mathcal{NN}}, v_h) = F_h(v_h) \quad \forall v_h \in V_h. \tag{8}$$

In order to handle this problem by the neural network, let us introduce a basis in  $V_h$ , say  $V_h = \text{span}\{\varphi_i : i \in I_h\}$ , and for any  $w \in V$  let us define the residuals

$$r_{h,i}(w) = F_h(\varphi_i) - a_h(w, \varphi_i), \quad i \in I_h, \tag{9}$$

as well as the loss function

$$R_h^2(w) = \sum_{i \in I_h} r_{h,i}^2(w). \tag{10}$$

Then, we search for a global minimum of the loss function in  $V^{\mathcal{NN}}$ , i.e., we consider the following minimization problem: *Find  $u^{\mathcal{NN}} \in V^{\mathcal{NN}}$  such that*

$$u^{\mathcal{NN}} \in \arg \min_{w \in V^{\mathcal{NN}}} R_h^2(w). \tag{11}$$

Existence of a minimum follows immediately from the fact that  $R_h^2$  is a continuous, quadratic function of its argument. Uniqueness may not occur. Indeed, any solution  $u^{\mathcal{NN}}$  of (8) annihilates the loss function, hence it is a solution of (11); such a solution may not be unique, since the set of equations (8) may be underdetermined (in particular, for  $f = 0$  one may obtain a non-zero  $u^{\mathcal{NN}}$ , see [13, Sect. 6.3]). On the other hand, system (8) may be overdetermined, and admit no solution; in this case, the loss function will have strictly positive minima.

**Remark 2** (Discretization with interpolation) In order to reduce and control the random effects related to the use of a network depending upon a large number of weights, in [13] we proposed to locally project the neural network upon a space of polynomials, before computing the loss function.

To be precise, we have considered a conforming, shape-regular partition  $\mathcal{T}_H = \{G_m : 1 \leq m \leq M_h\}$  of  $\bar{\Omega}$ , which is equal to or coarser than  $\mathcal{T}_h$  (i.e., each element  $E \in \mathcal{T}_h$  is contained in an element  $G \in \mathcal{T}_H$ ) but compatible with  $\mathcal{T}_h$  (i.e., its meshsize  $H > 0$  satisfies  $H \lesssim h$ ). Let  $V_H \subset V$  be the linear subspace formed by the functions which are piecewise polynomials of degree  $k_{\text{int}} = q + 1$  over the triangulation  $\mathcal{T}_H$ , and let  $\mathcal{I}_H : C^0(\bar{\Omega}) \rightarrow V_H$  be the associated element-wise Lagrange interpolation operator.

Given a neural network  $w \in V^{\mathcal{NN}}$ , let us denote by  $w_H = \mathcal{I}_H w^{\mathcal{NN}} \in V_H$  its piecewise polynomial interpolant. Then, the definition (9) of local residuals is modified as

$$\tilde{r}_{h,i}(w) = F_h(\varphi_i) - a_h(w_H, \varphi_i), \quad i \in I_h; \tag{12}$$

consequently, the loss function takes the form

$$\tilde{R}_h^2(w) = \sum_{i \in I_h} \tilde{r}_{h,i}^2(w), \tag{13}$$

and we define a new approximation of the solution of Problem (4) by setting

$$\tilde{u}_H^{\mathcal{NN}} = \mathcal{I}_H \tilde{u}^{\mathcal{NN}} \in V_H, \quad \text{where } \tilde{u}^{\mathcal{NN}} \in \arg \min_{w \in U^{\mathcal{NN}}} \tilde{R}_h^2(w). \quad (14)$$

In [13] we derived an a priori error estimate for the error  $\|u - \tilde{u}_H^{\mathcal{NN}}\|_V$ , and we documented the error decay as  $h \rightarrow 0$ , which turns out to have a more regular behavior than the error  $\|u - u^{\mathcal{NN}}\|_V$ , although the latter is usually smaller.

The subsequent a posteriori error analysis could be extended to give a control on the error produced by  $\tilde{u}_H^{\mathcal{NN}}$  as well. For the sake of simplicity, we do not pursue such a task here.

### 3 The a posteriori error estimator

In order to build an error estimator, let us first choose, for any  $E \in \mathcal{T}_h$  and any  $k \geq 0$ , a projection operator  $\Pi_{E,k} : L^2(E) \rightarrow \mathbb{P}_k(E)$  satisfying

$$\int_E \Pi_{E,k} \varphi = \int_E \varphi \quad \forall \varphi \in L^2(E). \quad (15)$$

This allows us to introduce approximate bilinear and linear forms

$$a_\pi(w, v) = \sum_{E \in \mathcal{T}_h} \int_E \Pi_{E,q}(\mu \nabla w) \cdot \nabla v + \Pi_{E,q-1}(\beta \cdot \nabla w + \sigma w) v, \quad (16)$$

$$F_\pi(v) = \sum_{E \in \mathcal{T}_h} \int_E (\Pi_{E,q-1} f) v, \quad (17)$$

which are useful in the forthcoming derivation. Indeed, the coercivity of the form  $a$  allows us to bound the  $V$ -norm of the error as follows:

$$\|u - u^{\mathcal{NN}}\|_{1,\Omega} \leq \frac{1}{\alpha} \sup_{v \in V} \frac{a(u - u^{\mathcal{NN}}, v)}{\|v\|_{1,\Omega}}. \quad (18)$$

We split the numerator as

$$\begin{aligned} a(u - u^{\mathcal{NN}}, v) &= F(v) - a(u^{\mathcal{NN}}, v) = \underbrace{F(v) - F_\pi(v)}_{(I)} + \underbrace{F_\pi(v) - a_\pi(u^{\mathcal{NN}}, v)}_{(III)} \\ &\quad + \underbrace{a_\pi(u^{\mathcal{NN}}, v) - a(u^{\mathcal{NN}}, v)}_{(II)} \end{aligned} \quad (19)$$

and we proceed to bound each term on the right-hand side.

The terms (I) and (II) account for the element-wise projection error upon polynomial spaces; they are estimated in the next two Lemmas.

**Lemma 1** *The quantity (I) defined in (19) satisfies*

$$|(\text{I})| \lesssim \left( \sum_{E \in \mathcal{T}_h} \eta_{\text{rhs},1}^2(E) \right)^{1/2} |v|_{1,\Omega}, \tag{20}$$

with

$$\eta_{\text{rhs},1}(E) = h_E \|f - \Pi_{E,q-1} f\|_{0,E}. \tag{21}$$

**Proof** Setting  $m_E(v) = \frac{1}{|E|} \int_E v$  and using (15), we get

$$(\text{I}) = \sum_{E \in \mathcal{T}_h} \int_E (f - \Pi_{E,q-1} f)(v - m_E(v)),$$

and we conclude using the bound  $\|v - m_E(v)\|_{0,E} \lesssim h_E |v|_{1,E}$ . □

**Lemma 2** *The quantity (II) defined in (19) satisfies*

$$|(\text{II})| \lesssim \left( \sum_{E \in \mathcal{T}_h} (\eta_{\text{coef},1}^2(E) + \eta_{\text{coef},2}^2(E) + \eta_{\text{coef},3}^2(E)) \right)^{1/2} |v|_{1,\Omega}, \tag{22}$$

with

$$\begin{aligned} \eta_{\text{coef},1}(E) &= \|\mu \nabla u^{\mathcal{N}N} - \Pi_{E,q}(\mu \nabla u^{\mathcal{N}N})\|_{0,E}, \\ \eta_{\text{coef},2}(E) &= h_E \|\beta \cdot \nabla u^{\mathcal{N}N} - \Pi_{E,q-1}(\beta \cdot \nabla u^{\mathcal{N}N})\|_{0,E}, \\ \eta_{\text{coef},3}(E) &= h_E \|\sigma u^{\mathcal{N}N} - \Pi_{E,q-1}(\sigma u^{\mathcal{N}N})\|_{0,E}. \end{aligned} \tag{23}$$

**Proof** It holds

$$\begin{aligned} (\text{II}) &= \sum_{E \in \mathcal{T}_h} \int_E (\mu \nabla u^{\mathcal{N}N} - \Pi_{E,q}(\mu \nabla u^{\mathcal{N}N})) \cdot \nabla v \\ &\quad + \sum_{E \in \mathcal{T}_h} \int_E (\beta \cdot \nabla u^{\mathcal{N}N} - \Pi_{E,q-1}(\beta \cdot \nabla u^{\mathcal{N}N}))(v - m_E(v)) \\ &\quad + \sum_{E \in \mathcal{T}_h} \int_E (\sigma u^{\mathcal{N}N} - \Pi_{E,q-1}(\sigma u^{\mathcal{N}N}))(v - m_E(v)), \end{aligned}$$

where we have used again (15). We conclude as in the proof of Lemma 1. □

Let us now focus on the quantity (III), which can be written as

$$(\text{III}) = \underbrace{F_\pi(v - v_h) - a_\pi(u^{\mathcal{N}N}, v - v_h)}_{(\text{IV})} + \underbrace{F_\pi(v_h) - a_\pi(u^{\mathcal{N}N}, v_h)}_{(\text{V})}, \quad \forall v_h \in V_h; \tag{24}$$

in turn, the quantity (V) can be written as

$$(V) = \underbrace{F_\pi(v_h) - F_h(v_h)}_{(VII)} + \underbrace{F_h(v_h) - a_h(u^{\mathcal{NN}}, v_h)}_{(VI)} + \underbrace{a_h(u^{\mathcal{NN}}, v_h) - a_\pi(u^{\mathcal{NN}}, v_h)}_{(VIII)}. \tag{25}$$

The bound of (IV) is standard in finite-element a posteriori error analysis: it involves the local bulk residuals

$$\text{bulk}_E(u^{\mathcal{NN}}) = \Pi_{E,q-1} f + \nabla \cdot \Pi_{E,q}(\mu \nabla u^{\mathcal{NN}}) - \Pi_{E,q-1}(\beta \cdot \nabla u^{\mathcal{NN}} + \sigma u^{\mathcal{NN}}) \tag{26}$$

and the interelement jumps at each edge  $e$  shared by two elements, say  $E_1$  and  $E_2$  with opposite normal unit vectors  $\mathbf{n}_1$  and  $\mathbf{n}_2$ , namely

$$\text{jump}_e(u^{\mathcal{NN}}) = \Pi_{E_1,q}(\mu \nabla u^{\mathcal{NN}}) \cdot \mathbf{n}_1 + \Pi_{E_2,q}(\mu \nabla u^{\mathcal{NN}}) \cdot \mathbf{n}_2; \tag{27}$$

in addition, one defines  $\text{jump}(u^{\mathcal{NN}}, e) = 0$  if  $e \subset \partial\Omega$ .

To derive the bound, the test function  $v_h$  in (24) is chosen as  $v_h = I_h^C v$ , the Clément interpolant of  $v$  on  $\mathcal{T}_h$  [16], which satisfies

$$\|v - I_h^C v\|_{k,E} \lesssim h_E^{1-k} |v|_{1,D_E}, \quad k = 0, 1, \tag{28}$$

where  $D_E = \cup\{E' \in \mathcal{T}_h : E \cap E' \neq \emptyset\}$ .

**Lemma 3** *The quantity (IV) defined in (24) satisfies*

$$|(IV)| \lesssim \left( \sum_{E \in \mathcal{T}_h} \eta_{\text{res}}^2(E) \right)^{1/2} |v|_{1,\Omega}, \tag{29}$$

where

$$\eta_{\text{res}}(E) = h_E \| \text{bulk}_E(u^{\mathcal{NN}}) \|_{0,E} + h_E^{1/2} \sum_{e \subset \partial E} \| \text{jump}_e(u^{\mathcal{NN}}) \|_{0,e}, \tag{30}$$

with  $\text{bulk}_E(u^{\mathcal{NN}})$  defined in (26) and  $\text{jump}_e(u^{\mathcal{NN}})$  defined in (27).

**Proof** We refer e.g. to [17] for more details. □

Before considering the quantity (VI), let us state a useful result of equivalence of norms.

**Lemma 4** *For any  $v_h = \sum_{i \in I_h} v_i \varphi_i \in V_h$ , let  $\mathbf{v} = (v_i)_{i \in I_h}$  be the vector of its coefficients. There exist constants  $0 < c_h \leq C_h$ , possibly depending on  $h$  such that*

$$c_h |v_h|_{1,\Omega} \leq \|\mathbf{v}\|_2 \leq C_h |v_h|_{1,\Omega} \quad \forall v_h \in V_h, \tag{31}$$

where  $\|\mathbf{v}\|_2 = \left( \sum_{i \in I_h} v_i^2 \right)^{1/2}$ .

**Proof** The result expresses the equivalence of norms in finite dimensional spaces. If the triangulation  $\mathcal{T}_h$  is quasi uniform, then one can prove by a standard reference-element argument that  $c_h \simeq h^{1-n/2}$  whereas  $C_h \simeq h^{-n/2}$ .  $\square$

We are now able to bound the quantity (VI) in terms of the loss function introduced in (10), as follows.

**Lemma 5** *The quantity (VI) defined in (25) satisfies*

$$|(VI)| \lesssim \eta_{\text{loss}} |v|_{1,\Omega}, \tag{32}$$

where

$$\eta_{\text{loss}} = C_h R_h(u^{\mathcal{N}\mathcal{N}}) \tag{33}$$

and the constant  $C_h$  is defined in (31).

**Proof** Writing  $v_h = \sum_{i \in I_h} v_i \varphi_i$ , it holds

$$(VI) = \sum_{i \in I_h} r_{h,i}(u^{\mathcal{N}\mathcal{N}}) v_i,$$

whence

$$|(VI)| \lesssim R_h(u^{\mathcal{N}\mathcal{N}}) \|v\|_2,$$

We conclude by using (31) and observing that

$$|v_h|_{1,\Omega} \lesssim |v|_{1,\Omega}, \tag{34}$$

since we have chosen  $v_h = I_h^C v$  and (28) holds.  $\square$

We are left with the problem of bounding the terms (VII) and (VIII) in (25). They are similar to the terms (I) and (II), respectively, but reflect the presence of the quadrature formula introduced in (6) and (7). In the forthcoming analysis, it will be useful to introduce the following notation for the quadrature-based discrete (semi-)norm on  $C^0(E)$ :

$$\|\varphi\|_{0,E,\omega} = \left( \sum_{t \in I^E} \varphi^2(\xi_t^E) \omega_t^E \right)^{1/2}. \tag{35}$$

**Lemma 6** *The quantity (VII) defined in (25) satisfies*

$$|(VII)| \lesssim \left( \sum_{E \in \mathcal{T}_h} \eta_{\text{rhs},2}^2(E) \right)^{1/2} |v|_{1,\Omega}, \tag{36}$$

with

$$\eta_{\text{rhs},2}(E) = h_E \|f - \Pi_{E,q-1} f\|_{0,E,\omega} + \|f - \Pi_{E,q} f\|_{0,E,\omega}. \quad (37)$$

**Proof** Recalling that the adopted quadrature rule has precision  $q$  and test functions  $v_h$  are piecewise linear polynomials, it holds

$$\begin{aligned} \text{(VII)} &= \sum_{E \in \mathcal{T}_h} \left( \int_E (\Pi_{E,q-1} f) v_h - \sum_{i \in I^E} f(\xi_i^E) v_h(\xi_i^E) \omega_i^E \right) \\ &= \sum_{E \in \mathcal{T}_h} \left( \sum_{i \in I^E} (\Pi_{E,q-1} f - f)(\xi_i^E) v_h(\xi_i^E) \omega_i^E \right) \\ &= \underbrace{\sum_{E \in \mathcal{T}_h} \left( \sum_{i \in I^E} (\Pi_{E,q-1} f - f)(\xi_i^E) (v_h - m_E(v_h))(\xi_i^E) \omega_i^E \right)}_{\text{(VIIa)}} \\ &\quad + \underbrace{\sum_{E \in \mathcal{T}_h} \left( \sum_{i \in I^E} (\Pi_{E,q-1} f - f)(\xi_i^E) \omega_i^E m_E(v_h) \right)}_{\text{(VIIb)}}. \end{aligned} \quad (38)$$

On the one hand, recalling the assumption  $q \geq 2$  and inequality (34) one has

$$\begin{aligned} |(\text{VIIa})| &\leq \sum_{E \in \mathcal{T}_h} \|f - \Pi_{E,q-1} f\|_{0,E,\omega} \|v_h - m_E(v_h)\|_{0,E,\omega} \\ &= \sum_{E \in \mathcal{T}_h} \|f - \Pi_{E,q-1} f\|_{0,E,\omega} \|v_h - m_E(v_h)\|_{0,E} \\ &\lesssim \sum_{E \in \mathcal{T}_h} h_E \|f - \Pi_{E,q-1} f\|_{0,E,\omega} |v_h|_{1,E} \\ &\lesssim \left( \sum_{E \in \mathcal{T}_h} h_E^2 \|f - \Pi_{E,q-1} f\|_{0,E,\omega}^2 \right)^{1/2} |v|_{1,\Omega}. \end{aligned} \quad (39)$$

On the other hand, we first observe that, by the exactness of the quadrature rule and (15), we get

$$\sum_{i \in I^E} (\Pi_{E,q-1} f)(\xi_i^E) \omega_i^E = \int_E \Pi_{E,q-1} f = \int_E f = \int_E \Pi_{E,q} f = \sum_{i \in I^E} (\Pi_{E,q} f)(\xi_i^E) \omega_i^E.$$

Hence,

$$\begin{aligned}
 |(\text{VIIb})| &\leq \sum_{E \in \mathcal{T}_h} \|f - \Pi_{E,q} f\|_{0,E,\omega} \|m_E(v_h)\|_{0,E} \\
 &\leq \sum_{E \in \mathcal{T}_h} \|f - \Pi_{E,q} f\|_{0,E,\omega} \|v_h\|_{0,E} \\
 &\lesssim \left( \sum_{E \in \mathcal{T}_h} \|f - \Pi_{E,q} f\|_{0,E,\omega}^2 \right)^{1/2} |v|_{1,\Omega}.
 \end{aligned}
 \tag{40}$$

This concludes the proof of Lemma 6. □

**Lemma 7** *The quantity (VIII) defined in (25) satisfies*

$$|(\text{VIII})| \lesssim \left( \sum_{E \in \mathcal{T}_h} (\eta_{\text{coef},4}^2(E) + \eta_{\text{coef},5}^2(E) + \eta_{\text{coef},6}^2(E)) \right)^{1/2} |v|_{1,\Omega}, \tag{41}$$

with

$$\begin{aligned}
 \eta_{\text{coef},4}(E) &= \|\mu \nabla u^{\mathcal{NN}} - \Pi_{E,q}(\mu \nabla u^{\mathcal{NN}})\|_{0,E,\omega}, \\
 \eta_{\text{coef},5}(E) &= h_E \|\boldsymbol{\beta} \cdot \nabla u^{\mathcal{NN}} - \Pi_{E,q-1}(\boldsymbol{\beta} \cdot \nabla u^{\mathcal{NN}})\|_{0,E,\omega}, \\
 &\quad + \|\boldsymbol{\beta} \cdot \nabla u^{\mathcal{NN}} - \Pi_{E,q}(\boldsymbol{\beta} \cdot \nabla u^{\mathcal{NN}})\|_{0,E,\omega} \\
 \eta_{\text{coef},6}(E) &= h_E \|\sigma u^{\mathcal{NN}} - \Pi_{E,q-1}(\sigma u^{\mathcal{NN}})\|_{0,E,\omega} \\
 &\quad + \|\sigma u^{\mathcal{NN}} - \Pi_{E,q}(\sigma u^{\mathcal{NN}})\|_{0,E,\omega}.
 \end{aligned}
 \tag{42}$$

**Proof** The term (VIII) can be written as

$$\begin{aligned}
 (\text{VIII}) &= \underbrace{\sum_{E \in \mathcal{T}_h} \left( \sum_{\iota \in I^E} (\mu \nabla u^{\mathcal{NN}})(\xi_\iota^E) \cdot \nabla v_h \omega_\iota^E - \int_E \Pi_{E,q}(\mu \nabla u^{\mathcal{NN}}) \cdot \nabla v_h \right)}_{(\text{VIIIa})} \\
 &\quad + \underbrace{\sum_{E \in \mathcal{T}_h} \left( \sum_{\iota \in I^E} (\boldsymbol{\beta} \cdot \nabla u^{\mathcal{NN}})(\xi_\iota^E) v_h(\xi_\iota^E) \omega_\iota^E - \int_E \Pi_{E,q-1}(\boldsymbol{\beta} \cdot \nabla u^{\mathcal{NN}}) v_h \right)}_{(\text{VIIIb})} \\
 &\quad + \underbrace{\sum_{E \in \mathcal{T}_h} \left( \sum_{\iota \in I^E} (\sigma u^{\mathcal{NN}})(\xi_\iota^E) v_h(\xi_\iota^E) \omega_\iota^E - \int_E \Pi_{E,q-1}(\sigma u^{\mathcal{NN}}) v_h \right)}_{(\text{VIIIc})}. \tag{43}
 \end{aligned}$$

Concerning (VIIIa), by the exactness of the quadrature rule and the fact that  $\nabla v_h$  is piecewise constant, one has

$$(\text{VIIIa}) = \sum_{E \in \mathcal{T}_h} \sum_{\iota \in I^E} (\mu \nabla u^{\mathcal{NN}} - \Pi_{E,q}(\mu \nabla u^{\mathcal{NN}}))(\xi_\iota^E) \cdot \nabla v_h \omega_\iota^E,$$

which easily gives

$$|(VIIIa)| \lesssim \left( \sum_{E \in \mathcal{T}_h} \|\mu \nabla u^{\mathcal{N}N} - \Pi_{E,q}(\mu \nabla u^{\mathcal{N}N})\|_{0,E,\omega}^2 \right)^{1/2} |v|_{1,\Omega}.$$

The terms (VIIIb) and (VIIIc) are similar to the term (VII) above, in which  $f$  is replaced by  $\beta \cdot \nabla u^{\mathcal{N}N}$  and  $\sigma u^{\mathcal{N}N}$ , respectively. Hence, they can be bounded as done for (VII). This concludes the proof of Lemma 7.  $\square$

At this point, we are ready to derive the announced a posteriori error estimates. In order to get an upper bound of the error, we concatenate (18), (19), (24), (25), and use the bounds given in Lemmas 1 to 7, arriving at the following result.

**Theorem 1** (a posteriori upper bound of the error) *Let  $u^{\mathcal{N}N} \in V^{\mathcal{N}N}$  satisfy (11). Then, the error  $u - u^{\mathcal{N}N}$  can be estimated from above as follows:*

$$|u - u^{\mathcal{N}N}|_{1,\Omega} \lesssim \left( \eta_{\text{res}} + \eta_{\text{loss}} + \eta_{\text{coef}} + \eta_{\text{rhs}} \right), \tag{44}$$

where

$$\eta_{\text{res}}^2 = \sum_{E \in \mathcal{T}_h} \eta_{\text{res}}^2(E), \quad \eta_{\text{coef}}^2 = \sum_{E \in \mathcal{T}_h} \sum_{k=1}^6 \eta_{\text{coef},k}^2(E), \quad \eta_{\text{rhs}}^2 = \sum_{E \in \mathcal{T}_h} \sum_{k=1}^2 \eta_{\text{rhs},k}^2(E). \tag{45}$$

We realize that the global estimator  $\eta = \eta_{\text{res}} + \eta_{\text{loss}} + \eta_{\text{coeff}} + \eta_{\text{rhs}}$  is the sum of four contributions:  $\eta_{\text{res}}$  is the classical residual-based estimator,  $\eta_{\text{loss}}$  measures how small the minimized loss function is, i.e., how well the discrete variational equations (8) are fulfilled, whereas  $\eta_{\text{coef}}$  and  $\eta_{\text{rhs}}$  reflect the error in approximating elementwise the coefficients of the operator and the right-hand side by polynomials of degrees related to the precision of the quadrature formula.

It is possible to derive from (44) an element-based a posteriori error estimator, which can be used to design an adaptive strategy of mesh refinement (see, e.g. [18]). To this end, from now on we assume that the basis  $\{\varphi_i : i \in I_h\}$  of  $V_h$ , introduced to define (9), is the canonical Lagrange basis associated with the nodes of the triangulation  $\mathcal{T}_h$ . Given any  $E \in \mathcal{T}_h$ , we introduce the elemental index set  $I_h^E = \{i \in I_h : E \subset \text{supp } \varphi_i\}$ , where  $\text{supp } \varphi_i$  is the support of  $\varphi_i$ , and we define a local contribution to the term  $\eta_{\text{loss}}$  as follows:

$$\eta_{\text{loss}}^2(E) = C_h^2 \sum_{i \in I_h^E} r_{h,i}^2(u^{\mathcal{N}N}), \tag{46}$$

which satisfies

$$\eta_{\text{loss}}^2 \leq \sum_{E \in \mathcal{T}_h} \eta_{\text{loss}}^2(E).$$

With this definition at hand, we can introduce the following elemental error estimator.

**Definition 1** (elemental error estimator) For any  $E \in \mathcal{T}_h$ , let us set

$$\eta^2(E) = \eta_{\text{res}}^2(E) + \eta_{\text{loss}}^2(E) + \sum_{k=1}^6 \eta_{\text{coef},k}^2(E) + \sum_{k=1}^2 \eta_{\text{rhs},k}^2(E), \tag{47}$$

where the addends in this sum are defined, respectively, in (30), (46), (23) and (42), (21) and (37).

Then, Theorem 1 can be re-formulated in terms of these quantities.

**Corollary 2** (localized a posteriori error estimator) The error  $u - u^{\mathcal{NN}}$  can be estimated as follows:

$$\|u - u^{\mathcal{NN}}\|_{1,\Omega} \lesssim \left( \sum_{E \in \mathcal{T}_h} \eta^2(E) \right)^{1/2}. \tag{48}$$

Inequality (48) guarantees the *reliability* of the proposed error estimator, namely the estimator does provide a computable upper bound of the discretization error. Next result assures that the estimator is also *efficient*, namely it does not overestimate the error.

**Theorem 3** (a posteriori lower bound of the error) Let  $u^{\mathcal{NN}} \in V^{\mathcal{NN}}$  satisfy (11). Then, the error  $u - u^{\mathcal{NN}}$  can be locally estimated from below as follows: for any  $E \in \mathcal{T}_h$  it holds

$$\eta_{\text{res}}(E) \lesssim \|u - u^{\mathcal{NN}}\|_{1,D_E} + \sum_{E' \subset D_E} \left( \sum_{k=1}^3 \eta_{\text{coef},k}^2(E') + \eta_{\text{rhs},1}^2(E') \right)^{1/2}, \tag{49}$$

$$\frac{c_h}{C_h} \eta_{\text{loss}}(E) \lesssim \|u - u^{\mathcal{NN}}\|_{1,D_E} + \sum_{E' \subset D_E} \left( \sum_{k=1}^6 \eta_{\text{coef},k}^2(E') + \sum_{k=1}^2 \eta_{\text{rhs},k}^2(E') \right)^{1/2}. \tag{50}$$

**Proof** To derive (49), let us first consider the bulk contribution to the estimator. We apply a classical argument in a posteriori analysis, namely we introduce a non-negative bubble function  $b_E \in V$  with support in  $E$  and such that  $\|\phi\|_{0,E} \simeq \|b_E^{1/2} \phi\|_{0,E}$  and  $\|\phi\|_{0,E} \simeq (\|b_E \phi\|_{0,E} + h_E |b_E \phi|_{1,E})$  for all  $\phi \in \mathbb{P}_q(E)$ .

Let us set  $w_E = \text{bulk}_E(u^{\mathcal{NN}}) b_E \in V$ . Then,

$$\|\text{bulk}_E(u^{\mathcal{NN}})\|_{0,E}^2 \lesssim \int_E \text{bulk}_E(u^{\mathcal{NN}})^2 b_E = \int_E \text{bulk}_E(u^{\mathcal{NN}}) w_E.$$

Writing

$$\begin{aligned} \text{bulk}_E(u^{\mathcal{N}N}) &= (f - Lu^{\mathcal{N}N}) + \nabla \cdot (\Pi_{E,q}(\mu \nabla u^{\mathcal{N}N}) - \mu \nabla u^{\mathcal{N}N}) \\ &\quad + \Pi_{E,q-1}(\boldsymbol{\beta} \cdot \nabla u^{\mathcal{N}N}) - \boldsymbol{\beta} \cdot \nabla u^{\mathcal{N}N} + \Pi_{E,q-1}(\sigma u^{\mathcal{N}N}) - \sigma u^{\mathcal{N}N} \\ &\quad + \Pi_{q-1,E}f - f, \end{aligned}$$

we obtain

$$\begin{aligned} \int_E \text{bulk}_E(u^{\mathcal{N}N}) w_E &= a(u - u^{\mathcal{N}N}, w_E) - \int_E (\Pi_{E,q}(\mu \nabla u^{\mathcal{N}N}) - \mu \nabla u^{\mathcal{N}N}) \cdot \nabla w_E \\ &\quad + \int_E (\Pi_{E,q-1}(\boldsymbol{\beta} \cdot \nabla u^{\mathcal{N}N}) - \boldsymbol{\beta} \cdot \nabla u^{\mathcal{N}N})(w_E - m(w_E)) \\ &\quad + \int_E (\Pi_{E,q-1}(\sigma u^{\mathcal{N}N}) - \sigma u^{\mathcal{N}N})(w_E - m(w_E)) \\ &\quad + \int_E (\Pi_{q-1,E}f - f)(w_E - m(w_E)), \end{aligned}$$

whence

$$\|\text{bulk}_E(u^{\mathcal{N}N})\|_{0,E}^2 \lesssim \left( |u - u^{\mathcal{N}N}|_{1,E} + \sum_{k=1}^3 \eta_{\text{coef},k}(E) + \eta_{\text{rhs},1}(E) \right) |w_E|_{1,E}.$$

Using  $|w_E|_{1,E} \lesssim h_E^{-1} \|\text{bulk}_E(u^{\mathcal{N}N})\|_{0,E}$ , we arrive at

$$h_E \|\text{bulk}_E(u^{\mathcal{N}N})\|_{0,E} \lesssim |u - u^{\mathcal{N}N}|_{1,E} + \sum_{k=1}^3 \eta_{\text{coef},k}(E) + \eta_{\text{rhs},1}(E). \tag{51}$$

Let us now turn to the jump contribution to the estimator. Given an edge  $e \subset \partial E$  shared with the element  $E'$ , we introduce a non-negative bubble function  $b_e \in V$ , with support in  $E \cup E'$  and such that  $\|\phi\|_{0,e} \simeq \|b_e^{1/2}\phi\|_{0,e}$  and  $(h_E^{-1/2}\|b_e\phi\|_{0,E} + h_E^{1/2}|b_e\phi|_{1,E}) \lesssim \|\phi\|_{0,e}$  for all  $\phi \in \mathbb{P}_q(E)$ .

Let us extend the function  $\text{jump}_e(u^{\mathcal{N}N})$  onto  $E \cup E'$  to be constant in the normal direction to  $e$ , obtaining a polynomial of degree  $q$  in each element. Let us set  $w_e = \text{jump}_e(u^{\mathcal{N}N})b_e \in V$ . Then, writing  $E_1 = E$  and  $E_2 = E'$ , one has

$$\begin{aligned} \|\text{jump}_e(u^{\mathcal{N}N})\|_{0,e}^2 &\lesssim \int_e \text{jump}_e(u^{\mathcal{N}N})^2 b_e = \int_e \text{jump}_e(u^{\mathcal{N}N}) w_e \\ &= \int_e \text{jump}_e(u^{\mathcal{N}N} - u) w_e \\ &= \sum_{i=1}^2 \int_{E_i} \nabla \cdot [(\Pi_{E_i,q}(\mu \nabla u^{\mathcal{N}N}) - \mu \nabla u) w_e] \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^2 \int_{E_i} [\nabla \cdot \Pi_{E_i,q}(\mu \nabla u^{\mathcal{N}\mathcal{N}}) - \nabla \cdot (\mu \nabla u)] w_e \\ &+ \sum_{i=1}^2 \int_{E_i} [\Pi_{E_i,q}(\mu \nabla u^{\mathcal{N}\mathcal{N}}) - \mu \nabla u] \cdot \nabla w_e . \end{aligned}$$

We now recall that

$$\nabla \cdot \Pi_{E_i,q}(\mu \nabla u^{\mathcal{N}\mathcal{N}}) = \text{bulk}_{E_i}(u^{\mathcal{N}\mathcal{N}}) - \Pi_{E_i,q-1}f + \Pi_{E_i,q-1}(\boldsymbol{\beta} \cdot \nabla u^{\mathcal{N}\mathcal{N}} + \sigma u^{\mathcal{N}\mathcal{N}}),$$

as well as  $\nabla \cdot (\mu \nabla u) = -f + \boldsymbol{\beta} \cdot \nabla u + \sigma u$ . We write  $u = u^{\mathcal{N}\mathcal{N}} + (u - u^{\mathcal{N}\mathcal{N}})$  and we proceed as in the proof of (51), using now the bounds  $\|w_e\|_{0,E_i} \lesssim h_{E_i}^{1/2} \|\text{jump}_e(u^{\mathcal{N}\mathcal{N}})\|_{0,e}$  and  $|w_e|_{1,E_i} \lesssim h_{E_i}^{-1/2} \|\text{jump}_e(u^{\mathcal{N}\mathcal{N}})\|_{0,e}$ , arriving at the bound

$$\begin{aligned} h_E^{1/2} \sum_{e \subset \partial E} \|\text{jump}_e(u^{\mathcal{N}\mathcal{N}})\|_{0,e} &\lesssim |u - u^{\mathcal{N}\mathcal{N}}|_{1,D_E} + \sum_{E' \subset D_E} h_{E'} \|\text{bulk}_{E'}(u^{\mathcal{N}\mathcal{N}})\|_{0,E'} \\ &+ \sum_{E' \subset D_E} \left( \sum_{k=1}^3 \eta_{\text{coef},k}(E') + \eta_{\text{rhs},1}(E') \right) . \end{aligned} \tag{52}$$

Together with (51), this gives the bound (49). In order to derive (50), we write (46) as

$$C_h^{-1} \eta_{\text{loss}}(E) = \left( \sum_{i \in I_h^E} r_{h,i}^2(u^{\mathcal{N}\mathcal{N}}) \right)^{1/2} = \sup_{\mathbf{v}} \frac{1}{\|\mathbf{v}\|_2} \sum_{i \in I_h^E} r_{h,i}(u^{\mathcal{N}\mathcal{N}}) v_i$$

where  $\mathbf{v} = (v_i) \in \mathbb{R}^{\text{card}I_h^E}$ . Defining the function  $v_h^E = \sum_{i \in I_h^E} v_i \varphi_i \in V_h$ , which is supported in  $D_E$ , and recalling (9), we have

$$\sum_{i \in I_h^E} r_{h,i}(u^{\mathcal{N}\mathcal{N}}) v_i = F_h(v_h^E) - a_h(u^{\mathcal{N}\mathcal{N}}, v_h^E).$$

By the left-hand inequality in (31), we obtain

$$\frac{c_h}{C_h} \eta_{\text{loss}}(E) \leq \sup_{v_h^E} \frac{F_h(v_h^E) - a_h(u^{\mathcal{N}\mathcal{N}}, v_h^E)}{|v_h^E|_{1,D_E}}.$$

Now we write

$$\begin{aligned} F_h(v_h^E) - a_h(u^{\mathcal{N}\mathcal{N}}, v_h^E) &= F_h(v_h^E) - F(v_h^E) \\ &+ f(v_h^E) - a(u^{\mathcal{N}\mathcal{N}}, v_h^E) \\ &+ a(u^{\mathcal{N}\mathcal{N}}, v_h^E) - a_h(u^{\mathcal{N}\mathcal{N}}, v_h^E) . \end{aligned}$$

The term  $F_h(v_h^E) - F(v_h^E) = [F_h(v_h^E) - F_\pi(v_h^E)] + [F_\pi(v_h^E) - F(v_h^E)]$  can be bounded as done for the terms (I) and (VII) above, yielding

$$|F_h(v_h^E) - F(v_h^E)| \lesssim \sum_{E' \subset D_E} (\eta_{\text{rhs},1}(E') + \eta_{\text{rhs},2}(E')) |v_h^E|_{1,E'}.$$

Similarly, the term  $a(u^{\mathcal{NN}}, v_h^E) - a_h(u^{\mathcal{NN}}, v_h^E)$  can be handled as done for the terms (III) and (VIII) above, obtaining

$$|a(u^{\mathcal{NN}}, v_h^E) - a_h(u^{\mathcal{NN}}, v_h^E)| \lesssim \sum_{E' \subset D_E} \left( \sum_{k=1}^6 \eta_{\text{coeff},k}(E') \right) |v_h^E|_{1,E'}.$$

Finally, one has  $|f(v_h^E) - a(u^{\mathcal{NN}}, v_h^E)| \lesssim |u - u^{\mathcal{NN}}|_{1,D_E} |v_h^E|_{1,D_E}$ , thereby concluding the proof of (50).  $\square$

**Remark 3** (Comparison with different error analyses) In the last few years a PINN error analysis involving approximation, optimization and generalization errors has emerged [14]. The *approximation error* is associated with the expressivity and the best approximation error of a neural network with a fixed architecture, the *optimization error* with the local minima found during the optimization phase, and the *generalization error* with the difference between the function represented by the trained neural network and the exact solution. In this perspective, the present paper provides a connection between the optimization and generalization errors. Indeed, Theorems 1 and 3 contain upper and lower bounds for the generalization error that are based on the value of the loss function and other suitable terms. We highlight that different results relating these two errors for PINNs are available in the literature (see [19, 20]). The analysis here proposed does not require any assumption on the training set, provides a computable information during training even when the current iterate is far from any good local minimum, and yields both global and local control on the  $H^1$  error (see Corollary 2).

## 4 Numerical results

Let us consider the two-dimensional domain  $\Omega = (0, 1)^2$  and the Poisson problem:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g & \text{on } \Gamma, \end{cases} \quad (53)$$

with the functions  $f$  and  $g$  such that the exact solution, represented in Fig. 1, is

$$u(x, y) = \tanh \left[ 2 \left( x^3 - y^4 \right) \right]. \quad (54)$$

Problem (53) is numerically solved by the VPINN discretization described in Sect. 2.1, extended to handle non-homogeneous Dirichlet condition as mentioned in Remark

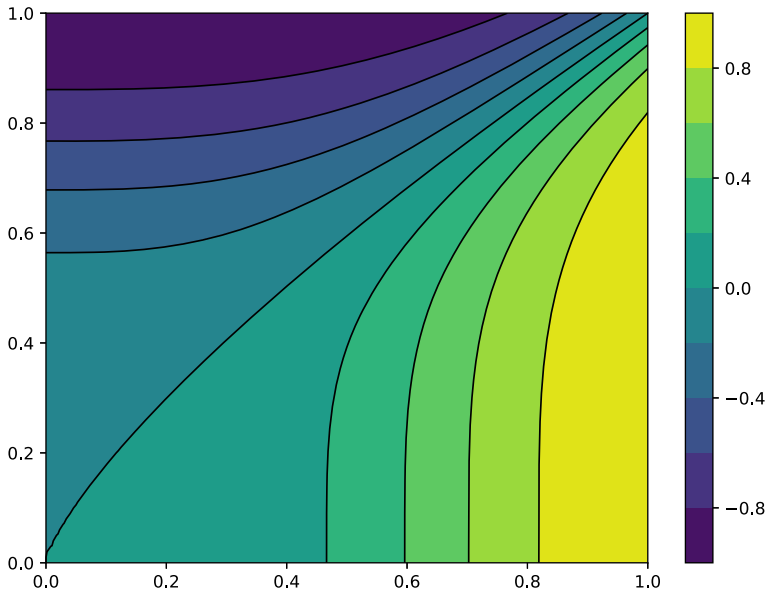


Fig. 1 Graphical representation of the exact solution  $u(x, y)$  in (54)

1. The used VPINN is a feed-forward fully connected neural network comprised by an input layer with input dimension  $n = 2$ , three hidden layers with 50 neurons each and an output layer with a single output variable; it thus contains 7851 trainable weights; furthermore, in all the layers except the output one the activation function is the hyperbolic tangent. The VPINN output is modified as described in [13] to exactly impose the Dirichlet boundary conditions. Gaussian quadrature rules of order  $q = 3$  are used in the definition of the loss function.

For ease of implementation, the orthogonal projection operators  $\Pi_{E,k}$ , defined in Sect. 3, are mimicked by interpolation operators as follows. Let us initially consider the elemental Lagrange interpolation operator  $\mathcal{I}_{E,k} : C^0(E) \rightarrow \mathbb{P}_k(E)$ ; then, to guarantee orthogonality to constants, the projection operator  $\tilde{\Pi}_{E,k} : C^0(E) \rightarrow \mathbb{P}_k(E)$  is defined by setting

$$\tilde{\Pi}_{E,k}\varphi := \mathcal{I}_{E,k}\varphi + \frac{\int_E (\varphi - \mathcal{I}_{E,k}\varphi)}{|E|}, \quad \forall \varphi \in C^0(E),$$

where, in practice, the integral  $\int_E (\varphi - \mathcal{I}_{E,k}\varphi)$  can be computed with quadrature rules that are more accurate than the ones used in the other operations. In this work we use quadrature rules of order 7 in each element.

The involved neural networks are initially trained using the first-order optimizer ADAM [21] for 3000 epochs with an exponentially decaying learning rate ranging between  $10^{-2}$  and  $10^{-3}$ . The corresponding solutions are then improved using the second-order BFGS optimizer [22] for 2000 epochs or until two subsequent iterates coincide (up to machine precision). Since the VPINN output is suitably modified to

**Table 1** Meshsizes  $h$  of the meshes used in Fig. 2, corresponding dimension  $N_{\text{qp}}$  of the training set and number  $\dim(V_h)$  of test functions

$h$	$N_{\text{qp}}$	$\dim(V_h)$	$h$	$N_{\text{qp}}$	$\dim(V_h)$	$h$	$N_{\text{qp}}$	$\dim(V_h)$
1.0	16	1	0.1939	616	62	0.0931	2808	320
0.5	64	5	0.1740	848	91	0.0885	3256	376
0.3757	112	7	0.1504	1048	116	0.08334	3704	432
0.3268	240	23	0.1384	1280	140	0.0785	4108	481
0.2716	316	29	0.125	1772	195	0.0737	4608	540
0.25	444	41	0.0965	2536	286	0.0696	5128	601

exactly satisfy the Dirichlet boundary conditions, the training set contains only the quadrature points required to compute the quantity (6).

The VPINN is trained on different meshes and the corresponding error estimators  $(\sum_{E \in \mathcal{T}_h} \eta^2(E))^{1/2}$  are computed. In Table 1 we indicate the number  $N_{\text{qp}}$  of quadrature points and the number  $\dim(V_h)$  of test functions required to construct the loss function using a mesh with meshsize  $h$ . We highlight that  $N_{\text{qp}}$  coincides with the dimension of the training set.

Once more, when exact integrals are involved, they are approximated with higher order quadrature rules. The obtained results are shown in Fig. 2, where the values of the  $H^1$ -error and the a posteriori estimator are displayed for several meshes of stepsize  $h$ . Remarkably, the error estimators (red dots) behave very similarly to the corresponding energy errors (blue dots). Moreover, coherently with the results discussed in [13], after an initial preasymptotic phase all dots are aligned on straight lines with slopes very close to 4 (the slope of the red line is 3.81, the slope of the blue line is 3.92). We remark that, since the used triangulations are quasi-uniform, both  $N_{\text{qp}}$  and  $\dim(V_h)$  approximately scale as  $O(h^{-2})$ ; therefore the qualitative behaviour shown in Fig. 2 does not change if any of these two quantities is represented on the  $x$ -axis.

It is also interesting to note that the terms appearing in the a posteriori estimator (recall (44)) exhibit different behaviors during the training of a single VPINN. This phenomenon is highlighted in Fig. 3, where one can observe the evolution of the quantities  $\eta_{\text{rhs}}$ ,  $\eta_{\text{coef}}$ ,  $\eta_{\text{res}}$ ,  $\eta_{\text{loss}}$ ,  $\eta$  and  $|u - u^{\mathcal{N}\mathcal{N}}|_{1,\Omega}$ , where each  $\eta_*$  stands for  $\eta_* = (\sum_{E \in \mathcal{T}_h} \eta_*^2(E))^{1/2}$  with  $*$   $\in$  {rhs, coef, res, loss}, during the 3000 epochs performed with the ADAM optimizer. It can be observed that, during this training, while the value of the loss function decreases, the accuracy remains almost constant because other sources of error, independent of the neural network, prevail.

## 5 Conclusions

We considered the discretization of a model elliptic boundary-value problem by variational physics-informed neural networks (VPINNs), in which test functions are continuous, piecewise linear functions on a triangulation of the domain. The scheme can be viewed as an instance of a least-square/Petrov–Galerkin method.

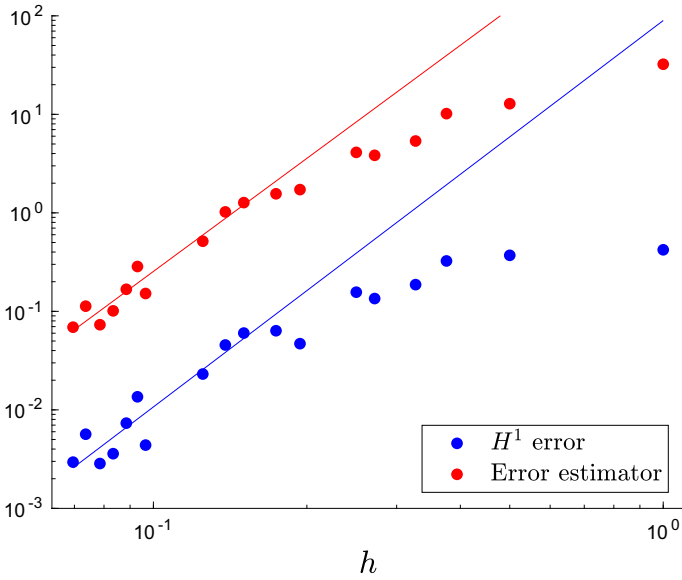


Fig. 2  $H^1$  errors (blue dots) obtained by training the same VPINN on different meshes, and corresponding error estimators (red dots) (colour figure online)

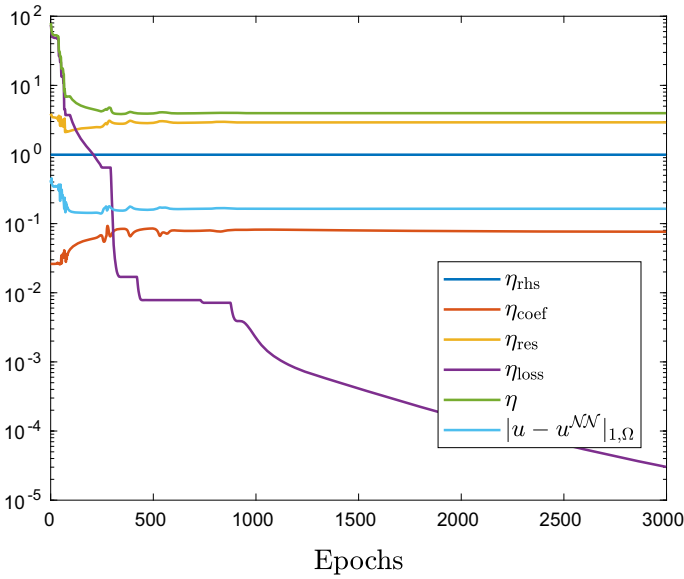


Fig. 3 Evolution of the addends of the error estimator  $\eta$  during training (colour figure online)

We introduced an a posteriori error estimator, which sums-up four contributions: the equation residual (measuring the elemental bulk residuals and the edge jump terms, for approximated coefficients and right-hand side), the coefficients' oscillation, the right-hand side's oscillation, and a scaled value of the loss-function. The latter term corresponds to an inexact solve of the algebraic system arising from the discretization of the variational equations.

The main result of the paper is the proof that the estimator provides a global upper bound and a local lower bound for the energy norm of the error between the exact and VPINN solutions. In other words, the a posteriori estimator is both reliable and efficient. Numerical results show an excellent agreement with the theoretical predictions.

In a forthcoming paper, we will investigate the use of the proposed estimator to design an adaptive strategy of discretization.

**Acknowledgements** The authors performed this research in the framework of the Italian MIUR Award "Dipartimenti di Eccellenza 2018-2022" granted to the Department of Mathematical Sciences, Politecnico di Torino (CUP: E11G18000350001). The research leading to this paper has also been partially supported by the SmartData@PoliTO center for Big Data and Machine Learning technologies. SB was supported by the Italian MIUR PRIN Project 201744KLJL-004, CC was supported by the Italian MIUR PRIN Project 201752HKH8-003. The authors are members of the Italian INdAM-GNCS research group.

**Funding** Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

## Declarations

**Conflict of interests** The authors have no relevant financial or non-financial interests to disclose.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. 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. *J. Comput. Phys.* **378**, 686–707 (2019). <https://doi.org/10.1016/j.jcp.2018.10.045>
2. Tartakovsky, A.M., Marrero, C.O., Perdikaris, P., Tartakovsky, G.D., Barajas-Solano, D.: Learning parameters and constitutive relationships with physics informed deep neural networks. arXiv preprint [arXiv:1808.03398](https://arxiv.org/abs/1808.03398) (2018)
3. Yang, Y., Perdikaris, P.: Adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys.* **394**, 136–152 (2019)
4. Lanthael, S., Mishra, S., Karniadakis, G.E.: Error estimates for deeponets: a deep learning framework in infinite dimensions. arXiv preprint [arXiv:2102.09618v2](https://arxiv.org/abs/2102.09618v2) (2021)
5. Kharazmi, E., Zhang, Z., Karniadakis, G.E.: VPINNs: variational physics-informed neural networks for solving partial differential equations. arXiv preprint [arXiv:1912.00873](https://arxiv.org/abs/1912.00873) (2019)

6. Khodayi-Mehr, R., Zavlanos, M.: VarNet: Variational neural networks for the solution of partial differential equations. In: *Learning for Dynamics and Control*, pp. 298–307 (2020). PMLR
7. Elbrächter, D., Perekrestenko, D., Grohs, P., Böleskei, H.: Deep neural network approximation theory. *IEEE Trans. Inf. Theory* **67**(5), 2581–2623 (2021)
8. Gühring, I., Kutyniok, G., Petersen, P.: Error bounds for approximations with deep ReLU neural networks in  $W^{s,p}$  norms. *Anal. Appl.* **18**(05), 803–859 (2020)
9. Opschoor, J.A., Petersen, P.C., Schwab, C.: Deep ReLU networks and high-order finite element methods. *Anal. Appl.* **18**(05), 715–770 (2020)
10. Kutyniok, G., Petersen, P., Raslan, M., Schneider, R.: A theoretical analysis of deep neural networks and parametric PDEs. *Constr. Approx.*, 1–53 (2021)
11. Opschoor, J.A., Schwab, C., Zech, J.: Exponential ReLU DNN expression of holomorphic maps in high dimension. *Constr. Approx.*, 1–46 (2021)
12. Gonon, L., Schwab, C.: Deep ReLU neural networks overcome the curse of dimensionality for partial integrodifferential equations. arXiv preprint [arXiv:2102.11707](https://arxiv.org/abs/2102.11707) (2021)
13. Berrone, S., Canuto, C., Pintore, M.: Variational physics informed neural networks: the role of quadratures and test functions. *J. Sci. Comput.* **92**(3), 1–27 (2022)
14. Kutyniok, G.: The mathematics of artificial intelligence. arXiv preprint [arXiv:2203.08890](https://arxiv.org/abs/2203.08890) (2022)
15. Sukumar, N., Srivastava, A.: Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Comput. Methods Appl. Mech. Engrg.* **389**, 114333–50 (2022). <https://doi.org/10.1016/j.cma.2021.114333>
16. Clément, P.: Approximation by finite element functions using local regularization. *Rev. Francaise Autom. Inf. Rech. Opér. Sér. Anal. Numér.* **9**(R–2), 77–84 (1975)
17. Verfürth, R.: *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Advanced Numerical Mathematics. Wiley-Teubner, Chichester (1996)
18. Nochetto, R.H., Veeser, A.: Primer of adaptive finite element methods. In: *Multiscale and Adaptivity: Modeling, Numerics and Applications*. Springer Lecture Notes in Math., CIME Series, vol. 2040, pp. 125–225 (2012)
19. De Ryck, T., Mishra, S.: Error analysis for physics informed neural networks (pinns) approximating kolmogorov pdes. arXiv preprint [arXiv:2106.14473](https://arxiv.org/abs/2106.14473) (2021)
20. De Ryck, T., Jagtap, A.D., Mishra, S.: Error estimates for physics informed neural networks approximating the navier-stokes equations. arXiv preprint [arXiv:2203.09346](https://arxiv.org/abs/2203.09346) (2022)
21. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014). <https://doi.org/10.48550/arXiv.1412.6980>
22. Wright, S., Nocedal, J., et al.: Numerical optimization. *Springer Sci.* **35**(67–68), 7 (1999)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.