Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Multi-level neural networks for high-dimensional parametric obstacle problems

Martin Eigel[1], Cosmas Heiß[2], Janina Schütte[1]

submitted: April 15, 2025

[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: martin.eigel@wias-berlin.de
janina.schuette@wias-berlin.de

[2] École Polytechnique Fédérale de Lausanne
Rte Cantonale
1015 Lausanne
Switzerland
E-Mail: cosmas.heiss@epfl.che

No. 3193
Berlin 2024

2020 *Mathematics Subject Classification.* 65N22, 68T07, 65N30, 65N55.

Key words and phrases. Obstacle problems, multilevel, convolutional neural networks, expressivity theory.

ME and JS acknowledge funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in the the priority programme SPP 2298 "Theoretical Foundations of Deep Learning".

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Multi-level neural networks for high-dimensional parametric obstacle problems

Martin Eigel, Cosmas Heiß, Janina Schütte

Abstract

A new method to solve computationally challenging (random) parametric obstacle problems is developed and analyzed, where the parameters can influence the related partial differential equation (PDE) and determine the position and surface structure of the obstacle. As governing equation, a stationary elliptic diffusion problem is assumed. The high-dimensional solution of the obstacle problem is approximated by a specifically constructed convolutional neural network (CNN). This novel algorithm is inspired by a finite element constrained multigrid algorithm to represent the parameter to solution map. This has two benefits: First, it allows for efficient practical computations since multi-level data is used as an explicit output of the NN thanks to an appropriate data preprocessing. This improves the efficacy of the training process and subsequently leads to small errors in the natural energy norm. Second, the comparison of the CNN to a multigrid algorithm provides means to carry out a complete a priori convergence and complexity analysis of the proposed NN architecture. Numerical experiments illustrate a state-of-the-art performance for this challenging problem.

1 Introduction

Free boundary problems arise in different research and engineering areas. They constitute solutions of a PDE with a priori unknown boundaries. Well-known examples include classes of obstacle problems and variational inequalities. The solution of a classical obstacle problem describes the position of an elastic membrane as a function u, which is fixed on the boundary of the domain D, always lies above some known obstacle φ and is under the influence of some forcing f, see [31]. The interface where the membrane touches the obstacle is a priori unknown. On the part of the domain where the membrane hangs freely, the position function fulfills a stationary diffusion equation. Fixing u on the boundary, the problem considered here has the form

$$\begin{cases} u(x,\mathbf{y}) \geq \varphi(x,\mathbf{y}) & \text{for all } x \in D \\ -\nabla\cdot(\kappa(x,\mathbf{y})\nabla u(x,\mathbf{y})) = f(x,\mathbf{y}) & \text{for all } x \text{ such that } u(x,\mathbf{y}) > \varphi(x,\mathbf{y}) \\ -\nabla\cdot(\kappa(x,\mathbf{y})\nabla u(x,\mathbf{y})) \geq f(x,\mathbf{y}) & \text{for all } x \in D \end{cases}, \tag{1}$$

depending on some coefficient κ and some countably infinite dimensional parameter vector $\mathbf{y} \in \Gamma \subseteq \mathbb{R}^{\mathbb{N}}$. The dependence on κ has for instance also been considered in [16, 25]. Obstacle problems are found in a variety of applications, namely the Stefan problem describing the process of ice melting in water, can be rewritten in form of the obstacle problem [12]. Furthermore, the obstacle problem finds applications in financial mathematics [5, 6, 27] and the minimizer of interaction energies can be written

DOI 10.20347/WIAS.PREPRINT.3193 Berlin 2024

in terms of the solution of an obstacle problem [8], which are encountered e.g. in in physics in particle behavior [22, 23] or in biology in collective behavior of animals [3, 11, 15]. For more information on practical uses of the obstacle problem, we refer to [17, 18, 31] and references therein.

With the gaining popularity of scientific machine learning (SciML), neural networks have been applied to solve obstacle problems in different ways, where the equations can be incorporated into a loss function. The obstacle condition can either be encoded in the neural network architecture, as e.g. in the first approach in [38] or in [2], or it can be enforced by penelization, as e.g. in the second approach in [38] or in [9]. In [1], the variational formulation of the obstacle problem is rewritten in a min-max formulation that is then used for training.

In the work presented here the coefficient and the obstacle depend on some high-dimensional stochastic parameter vector. In this setting the obstacle problem has to be solved for a large number of realizations of the vector. Classical mathematical solvers, as for example presented in [20] apply to a large class of coefficients. We exploit and inherent distribution of the parameters to develop more efficient tuned surrogate models mapping realizations of κ and φ to solutions of the obstacle problem.

A surrogate model mapping φ to the solution has been derived in [33]. The proximal neural network architecture with activation functions enforcing the obstacle condition was analyzed in the more general setting of variational inequalities. The convergence achieved in [33, Theorem 4.2] based on a fixed κ and φ is comparable to the convergence achieved in the present work for variable κ and φ as the analysis is based on an iterative scheme to approximate the solution in both cases. The architecture is implemented for the obstacle problem with variable obstacles [33, Example 4.4, Section 6.3], where φ can be mapped to the solution of the problem. The architecture in our work is analyzed and implemented for a variable obstacle and an additional variable coefficient and forcing.

Here, a multi-level decomposition of the solutions is utilized to implement individual networks approximating a coarse solution and fine grid corrections. As in this decomposition corrections on fine grids are only of small values, only a low accuracy is needed on fine grids with many parameters. This can be made use of by implementing comparably small NN architectures on high levels in terms of either the number of trainable parameters or number of samples on fine grids. Stochastic properties of the parametric obstacle problem are computed and analyzed based on multi-level decompositions of finite element spaces in [25] for fixed obstacles and in [4] for stochastic obstacles, based on adaptive finite element methods in [26].

The considered architecture inspired by the CNN constructed in [21] for parametric partial differential equations is analyzed in terms of expressivity specifically for the obstacle problem, i.e. with respect to the needed number of trainable parameters to achieve a required accuracy. We prove that CNNs can approximate a projected Richardson iteration leading to bounds on the number of parameters only depending logarithmically on the required accuracy. Furthermore, we prove that a multigrid algorithm based on the projected Richardson iteration and a monotone restriction operator can be approximated by the applied architecture. This shows that the surrogate model is at least as expressive as the multigrid solver.

The CNN architecture is tested for different parameter dimensions and for constant and variable obstacles and elasticities.

Main contributions:

- A CNN architecture mapping the coefficient, obstacle, and force to the solution of the obstacle

problem is presented.

- The architecture is analyzed in terms of expressivity. To the best of the authors knowledge, the achieved theoretical convergence results have so far only been derived for obstacle-to-solution maps for other architectures. The main result is presented in Theorem 5.3. Assuming that the coefficient is uniformly bounded from below and above there exists a constant $C > 0$ such that for any $\varepsilon > 0$ there exists a CNN Ψ with the number of parameters bounded by $\#\Psi \leq C \log(\varepsilon^{-1})$ such that and for all $\mathbf{y} \in \Gamma$ parameterizing the coefficient, force and obstacle it holds

$$\|\Psi(\boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi}) - \mathbf{u}(\cdot, \mathbf{y})\|_{H^1} \leq C \left(\|f\|_* + \|\boldsymbol{\varphi}\|_{H^1}\right) \varepsilon,$$

  where $\mathbf{u}$ is the collection of finite element coefficients of the solution of a discretized parametric obstacle problem and $\boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi}$ are the discretized coefficient, force, and obstacle.

- The combination of the provably well suited architecture and a multi-level decomposition of solutions as an output of the CNN leads to state-of-the-art numerical results.

2 Preliminaries

Throughout this work let $D \subseteq \mathbb{R}^d$ be a domain with a smooth boundary and $\Gamma \subseteq \mathbb{R}^{\mathbb{N}}$ be a countably infinite dimensional parameter space. Furthermore, let $\varphi : D \times \Gamma \to \mathbb{R}$ be a smooth obstacle such that $\varphi(x, \mathbf{y}) \leq 0$ for all $x \in \partial D, \mathbf{y} \in \Gamma$. Let $\kappa : D \times \Gamma \to \mathbb{R}$ be a coefficient, which is uniformly bounded from above and below, i.e. there exist a constants $\mathfrak{c}, \mathfrak{C} > 0$ such that for all $\mathbf{y} \in \Gamma$ and $x \in D$ it holds $\mathfrak{c} \leq \kappa(x, \mathbf{y}) \leq \mathfrak{C}$. This implies uniform ellipticity of the differential operator. Let $f : D \times \Gamma \to \mathbb{R}$ be the forcing such that $f(\cdot, \mathbf{y}) \in L^2(D)$ for each $\mathbf{y} \in \Gamma$.

For $v \in H_0^1(D)$, we make use of the norms

$$\begin{aligned} \|v\|_{L^2(D)}^2 &\coloneqq \int_D v^2 \,\mathrm{d}x, \\ \|v\|_{H^1(D)}^2 &\coloneqq \|v\|_{L^2(D)}^2 + \|\nabla v\|_{L^2(D)}^2, \\ \|v\|_{A_{\mathbf{y}}}^2 &\coloneqq \int_D \kappa(\cdot, \mathbf{y}) \langle \nabla v, \nabla v \rangle \,\mathrm{d}x. \end{aligned}$$

For $v_2 \in L^2(D) \hookrightarrow H^{-1}$, we associate v_2 with its associated function in H^{-1} to define the dual norm by

$$\|v_2\|_{H^{-1}} \coloneqq \sup_{\substack{v \in H^1 \\ \|v\|_{H^1} = 1}} \int_D v_2(x) v(x) \,\mathrm{d}x. \tag{2}$$

Generating training samples for our approach relies on numerical methods for solving problem (1). As our method is heavily inspired by multigrid solvers, we now introduce the finite element based

methodology usually applied to this type of problem. Here solutions are approximated in finite dimensional subspaces of $H_0^1(D)$. It leads to an algebraic equation to identify the coefficients of a linear combination of basis functions determined by a mesh. In this work, classical P1 finite element spaces are considered. We skip well-known results about standard FEM and refer to [36, 14, 7] for a detailed overview.

In our setting, let $\mathcal{T}$ be a uniform triangulation of the domain D with nodes $\mathcal{N}$. For the number of nodes in the triangulation $N := |\mathcal{N}|$ and each node $i = 1, \dots, N$ let λ_i be the nodal hat function, which is linear on every triangle, equal to 1 at node i and 0 at every other node. The set of all such hat functions is then referred to a s the P1 FE basis. Let $h > 0$ be the minimal side length over all triangles in $\mathcal{T}$. The considered finite element space is defined by $V_h := \operatorname{span}\{\lambda_i : i = 1, \dots N\}$. Functions $v_h \in V_h$ can then be written as linear combination of basis functions in the form $v_h = \sum_{i=1}^N \mathbf{v}_i \lambda_i$ with coefficients $\mathbf{v} \in \mathbb{R}^N$.

Note that the uniform lower and upper bounds on κ imply the existence of constants $c_{H^1}, C_{H^1}, C_{V_h} > 0$ such that for all $v_h \in V_h$ it holds that

$$c_{H^1} \|v_h\|_{A_{\mathbf{y}}} \leq \|v_h\|_{H^1(D)} \leq C_{H^1} \|v_h\|_{A_{\mathbf{y}}}, \tag{3}$$

$$\|\langle \nabla v_h, \nabla v_h \rangle\|_{L^2(D)} \leq C_{V_h} \|v_h\|_{L^2(D)}. \tag{4}$$

The second equation is often called *reverse Poincaré inequality*.

3 Parametric obstacle problem

The considered *obstacle problem* is described in (1). In our parametric setting, the obstacle, the coefficient and the forcing depend on some possibly countably infinite dimensional parameter vector $\mathbf{y} \in \Gamma \subseteq \mathbb{R}^{\mathbb{N}}$. This assumption is opposed to $\mathbf{y}$ only parameterizing the obstacle as for instance implemented in [33].

Note that the contact set, i.e. the area where the solution is equal to the obstacle, is not known in advance. An example of a parameter field sample, the solution and the contact set is depicted in Figure 3.1.

For the finite element approach, the problem is expressed in terms of a variational formulation. Discretizing the obstacle $\varphi(\cdot, \mathbf{y})$, the ellipticity $\kappa(\cdot, \mathbf{y})$ in V_h, testing the forcing f in V_h, and discretizing the test functions v and the solution u in the set $K := \{v_h \in V_h : v_h \geq \varphi \text{ a.e. in } D\}$ the following variational formulation can be derived.

Problem 3.1 (Variational parametric obstacle problem)**.** *Find $u_h \in K$ such that for all $v_h \in K$ it holds*

$$\int_D \kappa(\cdot, \mathbf{y}) \langle \nabla u_h, \nabla(v_h - u_h) \rangle \, \mathrm{d}x \geq \int_D f(x)(v_h - u_h)(x) \, \mathrm{d}x.$$

With uniform ellipticity as assumed here, it can be shown that a unique solution of (3.1) exists for every $\mathbf{y} \in \Gamma$, e.g. see [19, Chapter 2.2] for a proof. Note that choosing V_h to be the finite element space of low order polynomials, e.g. here $P1$ basis functions, is sufficient. This is a result of the solution of the nonlinear problem in general not being "very smoothöver the whole domain despite smooth data,

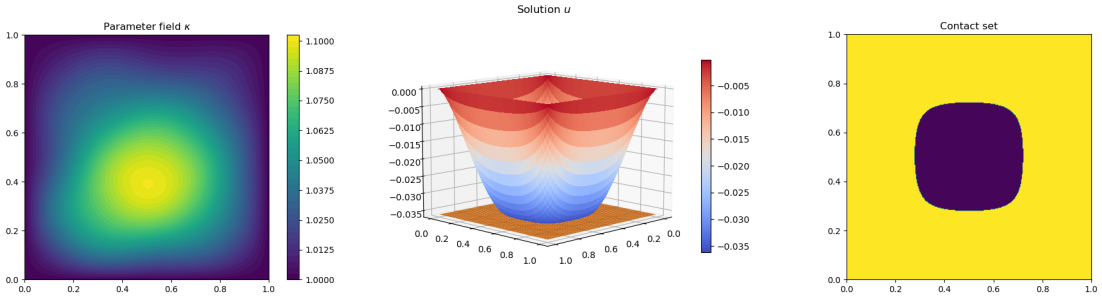

Figure 3.1: An example realization of a field κ, the respective solution to the obstacle problem u and the corresponding contact set indicating where the solution is equal to the obstacle are shown for a constant obstacle $\varphi \equiv -0.036$. The solution is equal to the obstacle in the purple part in the last image while it satisfies the PDE on the yellow part of the domain. Since the contact set is unknown in advance, it is part of the solution for the given parameter field.

e.g. see [10, Chapter 5]. In terms of the finite element coefficients the problem is equivalently written in the following form, e.g. see [19, Chapter 2.3].

Problem 3.2 (Variational parametric obstacle problem, discretized)**.** *Let* $A_{\mathbf{y}} \in \mathbb{R}^{N\times N}$ *be the discretized operator and* $\mathbf{f} \in \mathbb{R}^N$ *the tested forcing defined for* $i, j = 1, \dots, N$ *by*

$$(A_{\mathbf{y}})_{i,j} = \int_D \kappa_h(x, \mathbf{y}) \left\langle \nabla\lambda_i(x), \nabla\lambda_j(x) \right\rangle \, \mathrm{d}x \quad \textit{and} \quad \mathbf{f}_i = \int_D f(x)\lambda_i(x) \, \mathrm{d}x. \tag{5}$$

Find $\mathbf{u} \in \mathbb{R}^N$ *such that for the discretized obstacle* $\varphi_h = \sum_{i=1}^N \boldsymbol{\varphi}_i \lambda_i$ *it holds*

$$\begin{cases} \mathbf{u}_i \geq \boldsymbol{\varphi}_i & \textit{for } i = 1, \dots, N, \\ (A_{\mathbf{y}}\mathbf{u})_i \geq \mathbf{f}_i & \textit{for } i = 1, \dots, N, \\ (A_{\mathbf{y}}\mathbf{u})_i = \mathbf{f}_i & \textit{for } \mathbf{u}_i > \boldsymbol{\varphi}_i. \end{cases} \tag{6}$$

Since φ is not constrained to be zero on the boundary, one has to apply the discretization of the obstacle with care. The utilization fo $\mathrm{P}1$ elements here circumvents this consideration as the obstacle condition can equivalently be enforced on inner vertices. Additionally, note that it holds

$$\|v_h\|_{A_{\mathbf{y}}}^2 = \sum_{i,j=1}^N \mathbf{v}_i \mathbf{v}_j \int_D \kappa(\cdot, \mathbf{y}) \left\langle \nabla\lambda_i, \nabla\lambda_i \right\rangle \, \mathrm{d}x = \mathbf{v}^\intercal A_{\mathbf{y}} \mathbf{v} =: \|\mathbf{v}\|_{A_{\mathbf{y}}}^2 . \tag{7}$$

4 Multigrid solver

In this work the focus lies on solving the presented discretized obstacle problem (6) with a suitable NN architecture, which provides practical and theoretical benefits. For the theoretical underpinnings it is common to analyze NNs with respect to the number of trainable parameters as a measure of representation complexity. In the next chapters it is shown that our NN architecture is able to approximate a classical constrained multigrid solver. This means that the network is at least as expressive as

multigrid solvers with the possibility to find more accurate solutions. The central property we require for the later analysis is that the algorithm exhibits a structure amenable to an efficient NN approximation. This then allows the derivation of a quantitative convergence guarantee with complexity bounds. The solver provably converges and is based on a multigrid algorithm with a projection method as a smoother on every grid as detailed subsequently.

4.1 Projected Richardson iteration

Numerous algorithms have been developed in the past decades to solve Theorem 3.1 many of which are iterative approaches, see e.g. [20, 19, 37]. As a preparation for the later CNN construction, a projection method related to the Richardson iteration is introduced in the following. The particular version considered here can be found in [37, Section 3] and is called *projected Richardson iteration* throughout this work.

The main idea is to iteratively update an approximate solution with a weighted residual and apply a projection operation to enforce the given obstacle constraint. Suppressing the dependence on $\mathbf{y} \in \Gamma$ in the notation, let $\omega > 0$ be some damping parameter, $A \in \mathbb{R}^{N\times N}$ be defined as in (5), $\mathbf{f}$ the tested right-hand side and $\boldsymbol{\varphi}$ be the finite element coefficients of the obstacle of the parametric obstacle problem. Then, the algorithm consists of iterating dampened updates given by

$$\begin{aligned} \mathbf{u}^{(0)} &\coloneqq \boldsymbol{\varphi}, \\ \mathbf{u}^{(k+1)} &\coloneqq \max\left\{\mathbf{u}^{(k)} + \omega\left(\mathbf{f} - A\mathbf{u}^{(k)}\right), \boldsymbol{\varphi}\right\}, \end{aligned} \tag{8}$$

where the maximum is to be understood component-wise. To analyze the convergence of the algorithm, the residual $\mathbf{e}^{(k)} \coloneqq \mathbf{u}^{(k)} - \mathbf{u}$ is considered, where $\mathbf{u}$ is the solution of Theorem 3.2. With (6), the solution satisfies

$$\max\left\{\mathbf{u} + \omega(\mathbf{f} - A\mathbf{u}), \boldsymbol{\varphi}\right\} = \mathbf{u} + \max\left\{\omega(\mathbf{f} - A\mathbf{u}), \boldsymbol{\varphi} - \mathbf{u}\right\} = \mathbf{u}.$$

This can be seen by considering that for $i = 1, \dots, N$ on the one hand $\boldsymbol{\varphi}_i - \mathbf{u}_i < 0$ implies that $(\mathbf{f} - A\mathbf{u})_i = 0$ and therefore the maximum is zero. On the other hand, $\boldsymbol{\varphi}_i - \mathbf{u}_i = 0$ implies that $(\mathbf{f} - A\mathbf{u})_i \leq 0$ also leading to a maximum of zero. Using that the mapping $\mathbf{x} \mapsto \max\{\mathbf{x}, \boldsymbol{\varphi}\}$ is a contraction, the residual can be bounded as follows.

$$\begin{aligned} \left\|\mathbf{e}^{(k+1)}\right\|_A &= \left\|\mathbf{u}^{(k+1)} - \mathbf{u}\right\|_A \\ &= \left\|\max\left\{\mathbf{u}^{(k)} + \omega\left(\mathbf{f} - A\mathbf{u}^{(k)}\right), \boldsymbol{\varphi}\right\} - \max\left\{\mathbf{u} + \omega(\mathbf{f} - A\mathbf{u}), \boldsymbol{\varphi}\right\}\right\|_A \\ &\leq \left\|\mathbf{u}^{(k)} + \omega\left(\mathbf{f} - A\mathbf{u}^{(k)}\right) - (\mathbf{u} + \omega(\mathbf{f} - A\mathbf{u}))\right\|_A \\ &= \left\|\mathbf{e}^{(k)} + \omega A(\mathbf{u} - \mathbf{u}^{(k)})\right\|_A \\ &\leq \|I - \omega A\|_A \left\|\mathbf{e}^{(k)}\right\|_A. \end{aligned} \tag{9}$$

Therefore, the rate of convergence of the method is bounded by the energy norm of $I - \omega A$, where $\omega > 0$ needs to be chosen appropriately to ensure a contraction. For $A \coloneqq A_{\mathbf{y}}$ defined as in (5), we choose and bound ω independently of $\mathbf{y}$ to ensure convergence of the projected Richardson iteration for any $\mathbf{y} \in \Gamma$. First, $\omega_{\mathbf{y}}$ is chosen dependent on $\mathbf{y}$ such that the norm is bounded by a constant smaller than 1 also depending on $\mathbf{y}$.

Lemma 4.1 (generalization of [34, Lemma B.2] or [7, Lemma 4.3]). *Let $\mathbf{y} \in \Gamma$ and $\kappa(\cdot, \mathbf{y}) > 0$ everywhere. Then for any nonzero $\mathbf{w} \in \mathbb{R}^N$ it holds that*

$$\|(I - \omega_{\mathbf{y}} A_{\mathbf{y}})\mathbf{w}\|_{A_{\mathbf{y}}} \leq (1 - \omega_{\mathbf{y}}) \|\mathbf{w}\|_{A_{\mathbf{y}}},$$

where $0 < \omega_{\mathbf{y}} \leq \sigma_{\max}(A_{\mathbf{y}})^{-1}$.

The proof can be found in Appendix A. Second, ω is chosen independently of $\mathbf{y}$ such that the operator norm is bounded by a constant smaller than 1, which is also independent of $\mathbf{y}$.

Lemma 4.2. *Assume that κ is uniformly bounded, i.e. there exists a constant $\mathfrak{C} > 0$ such that $\kappa(x, \mathbf{y}) \leq \mathfrak{C}$ for all $x \in D, \mathbf{y} \in \Gamma$. Then, for all $0 < \omega \leq \frac{1}{\mathfrak{C} C_{V_h}}$ and $\mathbf{y} \in \Gamma$ it holds $\|I - \omega A_{\mathbf{y}}\|_{A_{\mathbf{y}}} \leq 1 - \omega$.*

Proof. First, we note that the maximal eigenvalue of $A_{\mathbf{y}}$ is bounded as can be derived as follows. Let $\mathbf{v} \in \mathbb{R}^{\dim V_h}$ be an eigenvector of $A_{\mathbf{y}}$ corresponding to the maximal eigenvalue and $v_h \in V_h$ be the corresponding finite element function. Then, for the equivalence constant C_{V_h} of the inverse Poincare inequality in the finite dimensional space V_h it holds that

$$\sigma_{\max}(A_{\mathbf{y}}) = \frac{\mathbf{v}^T A_{\mathbf{y}} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} = \frac{\int_D \kappa_h(\cdot, \mathbf{y}) \langle \nabla v_h, \nabla v_h \rangle \, \mathrm{d}x}{\int_D v_h^2 \, \mathrm{d}x} \leq \mathfrak{C} \frac{\int_D \langle \nabla v_h, \nabla v_h \rangle \, \mathrm{d}x}{\int_D v_h^2 \, \mathrm{d}x} \leq \mathfrak{C} C_{V_h}^2.$$

Choosing $0 < \omega \leq (\mathfrak{C} C_{V_h}^2)^{-1} \leq \sigma_{\max}(A_{\mathbf{y}})^{-1}$ for all $\mathbf{y} \in \Gamma$ yields the claim with Theorem 4.1. □

4.2 Geometric multigrid

Note that the convergence rate $1 - \omega$ in Theorem 4.1 is close to one for small ω, i.e. for large constants C_{V_h} defined in (4). In case of a uniform triangulation as considered here it can be shown that $C_{V_h} = Ch^{-1}$ is a possible choice for some constant $C > 0$ only depending on the angles of the triangles, see e.g. [14, Lemma 1.26] or [35]. In the two dimensional setting with a uniform triangulation as considered here, the nodes are arranged on a uniform grid. Therefore, the minimal side length of all triangles is given by $h = (\sqrt{N} - 1)^{-1}$ leading to $C_{V_h} = C(\sqrt{N} - 1)$. Theorem 4.1 therefore yields $\omega \lesssim N^{-1}$. This relationship implies slow convergence of the projected Richardson iteration for high fidelity discretizations, which is a reason why it cannot be considered state-of-the-art when solving discretized PDEs.

Instead, the projected Richardson iteration provides the basis for geometric multigrid methods, which are able to efficiently solve the problem at hand [20, 24]. An interplay between different meshes can lead to a speedup in convergence with a number of necessary iterations independent of the grid fidelity. Such results have been shown for instance for discrete Poisson problems in [14, Theorem 2.14] and [7, Theorem 4.2].

Multigrid methods are based on a set of $L \in \mathbb{N}$ triangulations $\mathcal{T}_1, \ldots, \mathcal{T}_L$, e.g. generated by a uniform or adaptive mesh refinement starting on the coarsest mesh $\mathcal{T}_1$, with nodes $\mathcal{N}_1, \ldots, \mathcal{N}_L$ and $N_\ell := |\mathcal{N}_\ell|$ for $\ell = 1, \ldots, L$ such that the corresponding subsequent finite element spaces are nested

$$V_1 \subseteq \ldots \subseteq V_L \subseteq H_0^1(D).$$

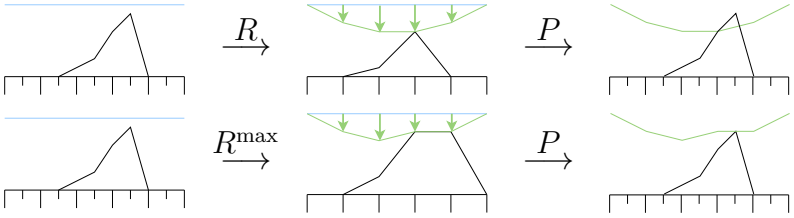

Figure 4.1: The first row images show the weighted restriction as defined in [21]. A visualization of the restriction operator defined in Theorem 4.4 is depicted in the second row images. In both rows the first image illustrates an obstacle in black and an initial guess for the solution in blue. The second images show the restricted obstacle together with a coarse grid solution in green. The last images depicts the prolongated coarse grid solution together with the true obstacle. It can be seen that taking a maximum, when restricting the obstacle, is critical for the coarse grid solution to still be above or equal to the true obstacle on the finer grid. The dependence on a level ℓ is suppressed in the notation.

The method then projects approximate solutions to finer spaces or restricts them to coarser spaces, applying smoothing iterations (the projected Richardson iteration) on coarse and fine grids successively. The considered prolongation operator used to interpolate functions on coarse grids into spaces on fine grids in the discretized setting is defined as follows.

Definition 4.3 (Prolongation matrices)**.** *Let $L \in \mathbb{N}$ be the number of grids and for some $\ell \in \{1, \dots, L\}$ let V_ℓ and $V_{\ell+1}$ as above. Then, the* prolongation matrix *$P_\ell \in \mathbb{R}^{N_{\ell+1} \times N_\ell}$ is the matrix representation of the canonical embedding of V_ℓ into $V_{\ell+1}$ under their respective finite element basis functions.*

The considered restriction operator maps coefficients on a fine grid to coefficients on a coarse grid such that the obstacle condition is still satisfied. The problem of applying the the restriction used in [21] to the obstacle problem is shown in Figure 4.1.

Definition 4.4 (Monotone restriction operator)**.** *For $\ell = 1, \dots, L-1$ let $V_\ell := \operatorname{span}\{\lambda_i^{(\ell)}\}_{i=1}^{N_\ell}$ and $V_{\ell+1} := \operatorname{span}\{\lambda_i^{(\ell+1)}\}_{i=1}^{N_{\ell+1}}$ be two nested P1 finite element spaces as above. Then, define the* monotone restriction operator *$R_\ell^{\max} : \mathbb{R}^{N_{\ell+1}} \to \mathbb{R}^{N_\ell}$ by*

$$(R_\ell^{\max}\mathbf{u})_i = \max\left\{\mathbf{u}_j : \operatorname{supp}\lambda_j^{(\ell+1)} \subseteq \operatorname{supp}\lambda_i^{(\ell)}\right\}.$$

These operators are used in the multigrid V-Cycle with monotone restriction (VCMR) in Algorithm 1. Starting on the finest grid $\ell = L$, the algorithm performs k projected Richardson iterations on the given grid. Subsequently, the problem is projected to a coarser grid to approximate a correcting term, where computations are cheaper. To approximate this term, the VCMR is called again with inputs restricted to the next coarser grid $\ell - 1$. The obstacle is restricted by the monotone restriction operator and the residual and the operator are restricted by the transposed prolongation (or weighted restriction) operator. On the coarsest level $\ell = 1$, the solution to the input problem is computed directly, e.g. by applying projected Richardson iterations until the algorithm has converged. The correction terms are returned to the higher levels and added to the current solution approximations. After the correction is added, another m smoothing steps are performed. The notation $\mathrm{VCMR}_{k,k_0,\ell}^m$ is used to describe the application of the $\mathrm{VCMR}_{k,\ell}$ $m \in \mathbb{N}$ times with k_0 smoothing steps on the coarsest grid.

Empirically, the described multigrid method provides a significant speedup. While for some problems the speedup can be quantified theoretically [14, 7], to the best of our knowledge this has not been shown for the obstacle problem. In contrast to multigrid methods for PDEs the nonlinearity of the obstacle introduces errors in the coarse grid corrections through the monotone restriction, which can slow down asymptotic convergence rates as for instance described in [20, Section 5.2]. The reason for using the projected Richardson iteration on every grid, despite it not being a state-of-the-art method (see [20, 37]), is its simplicity and the implications for the neural network architecture analyzed in the following chapters.

Algorithm 1: Multigrid V-Cycle with monotone restriction: $\mathrm{VCMR}_{k,\ell}$

Input: $\mathbf{u}, \mathbf{f}, A_{\mathbf{y}}, \boldsymbol{\varphi}$
for k *pre-smoothing steps* **do**
  $\mathbf{u} \leftarrow \max\{\mathbf{u} + \omega(\mathbf{f} - A_{\mathbf{y}}\mathbf{u}), \boldsymbol{\varphi}\}$ ▷ Perform smoothing steps
end
if $\ell = 1$ **then**
  solve $A_{\mathbf{y}}\mathbf{u} = \mathbf{f}$ for $\mathbf{u}$ on coarsest level ▷ E.g. by smoothing until converged
end
else
  $\overline{\boldsymbol{\varphi}} \leftarrow R_\ell^{\max}(\boldsymbol{\varphi} - \mathbf{u})$ ▷ Compute monotone restricted obstacle
  $\overline{\mathbf{r}} \leftarrow P_{\ell-1}^{\mathsf{T}}(\mathbf{f} - A_{\mathbf{y}}\mathbf{u})$ ▷ Compute restricted residual
  $\overline{A_{\mathbf{y}}} \leftarrow P_{\ell-1}^{\mathsf{T}} A_{\mathbf{y}} P_{\ell-1}$ ▷ Compute restricted operator
  $\overline{\mathbf{e}} \leftarrow \mathrm{VCMR}_{k,\ell-1}(0, \overline{\mathbf{r}}, \overline{A_{\mathbf{y}}}, \overline{\boldsymbol{\varphi}})$ ▷ Use V-Cycle with monotone restriction on coarser grid
  $\mathbf{u} \leftarrow \mathbf{u} + P_{\ell-1}\overline{\mathbf{e}}$ ▷ Add coarse correction
end
for k *post-smoothing steps* **do**
  $\mathbf{u} \leftarrow \max\{\mathbf{u} + \omega(\mathbf{f} - A_{\mathbf{y}}\mathbf{u}), \boldsymbol{\varphi}\}$ ▷ Perform smoothing steps
end
return: $\mathbf{u}$

5 Convolutional neural network

Convolutional neural networks (CNNs) have been proven to be an efficient tool for approximating solutions to parametric PDEs, see [21, 32, 34]. For the application of this architecture to the parametric obstacle problem, some additional steps have to be taken. For the analysis, the following conditions are assumed for the activation function throughout this paper.

Assumption 5.1 (Activation function)**.** *Let $\tau : \mathbb{R} \to \mathbb{R}$ satisfy the following conditions:*

1 *There exists $x_0 \in \mathbb{R}$ and an open interval $I \subseteq \mathbb{R}$ with $x_0 \in I$ such that τ is three times differentiable on I and $\tau''(x_0) \neq 0$.*

2 *For any $\varepsilon > 0$ there exist two affine-linear mappings $\varrho_1, \varrho_2 : \mathbb{R} \to \mathbb{R}$ such that for all $x \in \mathbb{R}$*

$$|(\varrho_2 \circ \tau \circ \varrho_1)(x) - \max\{x, 0\}| \leq \varepsilon.$$

These conditions are fulfilled by a family of soft ReLU variants such as Softplus, SeLU, or ELU, see [29]. The first condition is needed in the expressivity analysis to approximate multiplication, while the second condition is applied to approximate the maximum in the projected Richardson iteration. In the expressivity theory in [21], only the first condition is necessary since the maximum does not need to be approximated. The number of trainable parameters of a given CNN Ψ is denoted by $M(\Psi)$.

5.1 Expressivity

Similar to [21, Theorem 6] an approximation of a multigrid V-Cycle can be can be imitated by a CNN. Here the $\mathrm{VCMR}^m_{k,k_0,\ell}$ is approximated with projected Richardson iteration instead of a plain Richardson iteration and newly including the monotone restriction operator. For the expressivity analysis, it is first shown that the projected Richardson iteration can be approximated by a CNN of size growing linearly in the number of iterations.

Theorem 5.2 (CNN for the projected Richardson iteration)**.** *There exists a constant $C > 0$ such that for any $\omega, M, \varepsilon > 0$ and $m \in \mathbb{N}$ there exists a CNN $\Psi : \mathbb{R}^{4\times n\times n} \to \mathbb{R}^{1\times n\times n}$ such that for all discretized coefficients $\boldsymbol{\kappa} \in [-M, M]^{n\times n}$, initial guesses $\mathbf{u} \in [-M, M]^{n\times n}$, obstacles $\boldsymbol{\varphi} \in [-M, M]^{n\times n}$ and tested right-hand sides $\mathbf{f} \in [-M, M]^{n\times n}$ it holds*

(i) $\left\|\Psi(\mathbf{u}, \boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi}) - \mathbf{u}^{(m)}\right\|_{H^1(D)} \leq \varepsilon$,

(ii) number of parameters bounded by $M(\Psi) \leq Cm$.

The proof can be found in Appendix A. As derived in Section 4.1, the projected Richardson iteration can approximate the true solution up to arbitrary accuracy if the smoothing coefficient $\omega > 0$ is chosen appropriately. Then, the CNN approximating the Richardson iteration also approximates the solution of problem Theorem 3.2 as shown in the next corollary.

Corollary 5.3 (CNN for parametric obstacle problem)**.** *Assume that $\mathfrak{c} \leq \kappa(x, \mathbf{y}) \leq \mathfrak{C}$ is uniformly bounded by some $\mathfrak{c}, \mathfrak{C} > 0$ over all $\mathbf{y} \in \Gamma, x \in D$. Let $\mathbf{u}(\cdot, \mathbf{y}, \boldsymbol{\varphi})$ denote the solution of Theorem 3.2 for the parameter vector $\mathbf{y} \in \Gamma$ and obstacle $\boldsymbol{\varphi}$. For every $\varepsilon > 0$, there exists a CNN Ψ such that*

(i) for all $\mathbf{y} \in \Gamma$ it holds that

$$\left\|\Psi(\boldsymbol{\varphi}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}, \boldsymbol{\varphi}) - \mathbf{u}(\cdot, \mathbf{y}, \boldsymbol{\varphi})\right\|_{H^1} \leq \varepsilon \left(C^2_{H^1} \left\|f\right\|_{H^{-1}} + \frac{C_{H^1}}{c_{H^1}} \left\|\boldsymbol{\varphi}\right\|_{H^1} \right),$$

(ii) the number of parameters is bounded by $\#\Psi \leq C \left\lceil \log\left(\varepsilon^{-1}\right) \log\left(1 - \left(\mathfrak{C} C_{V_h}\right)^{-1}\right)^{-1} \right\rceil$, where $C > 0$ is the independent constant from Theorem 5.2.

The proof can be found in Appendix A. The upper bound for number of parameters is grid-dependent through the constant C_{V_h}. In [21] the dependence on the size h is circumvented by approximating a multigrid algorithm based on the Richardson iteration, which inspires a specific CNN architecture. Here, a fitting multigrid is approximated as well. The approximation of the Richardson iteration in Theorem 5.2 with the approximation of the prolongation and monotone restriction operator shows that the whole Algorithm 1 can be approximated by a U-net based CNN similar to the construction in [21].

The main differences lie in the additional projection in each step of the Richardson iteration and in the monotone restriction operator. The approximation of the monotone restriction operator is proven in the appendix in Theorem A.2.

Remark 5.4 (CNN for multigrid algorithm)**.** *Let $V_h \subseteq H_0^1(D)$ be the P1 FE space on a uniform square mesh. Then there exists a constant $C > 0$ such that for any $M, \varepsilon > 0$ and $k, m, \ell \in \mathbb{N}$ there exists a CNN $\Psi : \mathbb{R}^{4\times \dim V_h} \to \mathbb{R}^{\dim V_h}$ with*

(i) $\left\|\Psi(\mathbf{u}_0, \boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi}) - \mathrm{VCMR}^m_{k,k_0,\ell}(\mathbf{u}_0, \boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi})\right\|_{H^1(D)} \leq \varepsilon$ *for all* $\mathbf{u}_0, \boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi} \in [-M, M]^{\dim V_h}$

(ii) number of weights bounded by $M(\Psi) \leq Cm\ell$.

The proof can be found in Appendix A. The analysis works similarly to the analysis carried out in [21]. The main difference lies in the additional application of the maximum of the solution approximation and the obstacle in every step of the projected Richardson iteration and in the restriction operator. The architecture of the CNN used in the numerical experiments in this work is inspired by the constructive proof of Theorem 5.4. A description of the architecture can also be found in [21]. To the best of our knowledge, a convergence speed-up of the $\mathrm{VCMR}^m_{k,k_0,\ell}$ compared to the projected Richardson iteration has not been shown theoretically despite a considerable numerical speed up. Therefore, another approximation of the solution based on this algorithm is not included here.

5.2 Multi-level advantage

Multi-level machine learning algorithms include training neural networks on efficient decompositions of the data to improve performance. In [28] the generalization error for such a decomposition was analyzed. With the grids introduced in Section 4.2, lower-level models approximate a coarse grid solution and higher-level models approximate high fidelity corrections. The implemented network introduced in [21] is based on this multi-level decomposition. Here, a discretized solution operator $u_L : \Gamma \to V_L$ of the parametric obstacle problem is decomposed into components $u_\ell : \Gamma \to V_\ell$ on the individual spaces $V_1, \dots, V_L$ by

$$u_L = \sum_{\ell=1}^{L} u_\ell - u_{\ell-1} = \sum_{\ell=1}^{L} v_\ell,$$

where u_0 is set to zero and $v_\ell : \Gamma \to V_\ell$ are additive corrections on each level. This decomposition is visualized in Figure 6.1. The individual parts of the solution are approximated separately by CNNs. First, a normalized solution on a coarse grid is approximated with a CNN Ψ_1. Then, individual networks $\Psi_2 \dots, \Psi_L$ are trained to approximate normalized corrections with some normalization constant $b_\ell > 0$ on finer grids with some accuracy $\varepsilon_\ell > 0$ with

$$\left\|\Psi_\ell - \frac{v_\ell}{b_\ell}\right\| \leq \varepsilon_\ell.$$

Then, the weighted sum $\Psi := \sum_{\ell=1}^{L} b_\ell \Psi_\ell$ approximates the whole solution by

$$\left\|\Psi - u^L\right\| = \left\|\sum_{\ell=1}^{L} b_\ell \left(\Psi_\ell - \frac{v_\ell}{b_\ell}\right)\right\| \leq \sum_{\ell=1}^{L} b_\ell \left\|\Psi_\ell - \frac{v_\ell}{b_\ell}\right\|.$$

If the normalization constant is chosen as an operator norm of v_ℓ and setting $b_\ell \coloneqq \|v_\ell\|_{L^p(\Gamma,L^2(D))}$, the inequalities [4, Equation 6,10, Equation 7.6] imply that

$$\left\|\Psi - u^L\right\| = \sum_{\ell=1}^{L} \varepsilon_\ell h_\ell (\|f\|_{L^2(\Gamma,L^2(D))} + \|\varphi\|_{L^2(\Gamma,H^2(D))}) \leq C_{f,\varphi} \sum_{\ell=1}^{L} \varepsilon_\ell 2^{-\ell}$$

holds, when considering that the multi-level decomposition applied in the given architecture yields $h_\ell \leq C2^{-\ell}$ for the maximal side length h_ℓ of triangles in $\mathcal{T}_\ell$. To achieve an overall accuracy

$$C_{f,\varphi} \sum_{\ell=1}^{L} \varepsilon_\ell 2^{-\ell} \leq \varepsilon,$$

the accuracy of each sub-model on each level only has to satisfy

$$\varepsilon_\ell \leq \frac{\varepsilon 2^\ell}{L C_{f,\varphi}}. \tag{10}$$

Since functions in high fidelity spaces have more degrees of freedom, they are in general more difficult to approximate. The advantage of the decomposition then lies in the fact that the approximation in higher dimensional spaces is allowed to be exponentially worse. This can be either translated to less trainable parameters of the CNN or fewer expensive high fidelity solutions for training.

6 Numerical experiments

For the numerical tests, the U-net based architecture described in [21] and supported by the result in Theorem 5.4 was used. Here, $L = 7$ nested FE spaces are considered. A first CNN then approximates the solution of the obstacle problem on the coarse grid FE space V_1. Further individual networks then approximate the corrections of the solution on finer meshes in $V_2, \dots, V_7$ as visualized in Figure 6.1 for the first four spaces.

The following test cases are considered.

1 **Deterministic obstacle.** To solve Theorem 3.1 for a constant obstacle $\varphi(x, \mathbf{y}) = -0.035$ and $f(x) = 1$ for all $x \in D = [0,1]^2$, a sample $\mathbf{y} \in \Gamma$ was drawn. As e.g. in [13, 21], the coefficient field is assumed to have the representation

$$\kappa(x, \mathbf{y}) \coloneqq a_0(x) + \sum_{m=1}^{p} \mathbf{y}_m a_m(x),$$

where $a_m(x) \coloneqq m^{-2} \sin(\lfloor \frac{m+2}{2} \rfloor \pi x_1) \sin(\lceil \frac{m+2}{2} \rceil \pi x_2)$ and y is chosen uniformly in $\Gamma = [-1,1]^p$. A realization of the coefficient, the obstacle and the corresponding solution as well as the contact domain are visualized in Figure 3.1.

2 **Stochastic constant obstacle.** The problem above is now implemented with the additional variation of the obstacle with φ chosen as a constant function with value distributed uniformly in $[-0.045, -0.025]$. Since the value of the obstacle is the p-th entry of the $\mathbf{y}$, the sum above defining κ only goes to $p-1$.

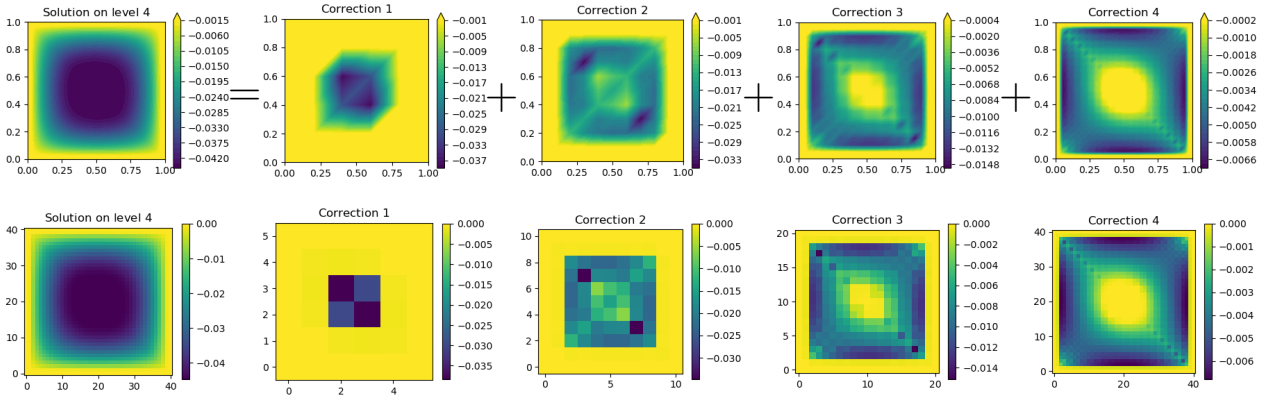

Figure 6.1: The solution to the obstacle problem in V_4 in the first row on the left-hand side can be decomposed into corrections in V_1, V_2, V_3, V_4 of decreasing values on different grids as seen in the first row. The sum of the corrections equals the full solution. The FE coefficients of the solution in $\mathbb{R}^{N_4}$ and the corrections in $\mathbb{R}^{N_1}, \mathbb{R}^{N_2}, \mathbb{R}^{N_3}, \mathbb{R}^{N_4}$ are visualized in images underneath each function.

3 **Rough surface obstacle.** Similar to [4], problem Theorem 3.1 is solved for a constant coefficient $\kappa \equiv 1$. The domain is chosen as $D = [0,1]^2$ and the constant forcing is set to $f \equiv 25$. The obstacle, which is used to model rough surfaces [30], is given by

$$\varphi(x, \mathbf{y}) = \sum_q B_q(H) \cos(q \cdot x + \mathbf{y}_q),$$

where the sum is taken over all q with components, which are multiples of π such that $1 \leq \|q\|_2 \leq 26$. The phase shifts $\mathbf{y}_q \sim \mathcal{U}([0, 2\pi])$ and the Hurst exponent $H \sim \mathcal{U}([0,1])$ are mutually independent and the amplitudes are defined as

$$B_q(H) = \pi(2\pi \max(\|q\|_2, 10))^{-(H+1)}/25.$$

The parameter vector $\mathbf{y}$ is set to the collection of H and $\mathbf{y}_q$ for all considered q. A realization of the rough surface and corresponding solution can be found in Figure 6.2.

In each setting the problem was solved on $L = 7$ levels with function spaces of size $(5 \times 2^{\ell-1} + 1) \times (5 \times 2^{\ell-1} + 1)$ for $\ell = 1, \ldots, L$. The networks was trained with 10.000 training samples and 1024 validation samples. The number of parameters of the network on each level were selected as shown in Table 1. Note that the network architecture is chosen to approximate a coarse grid solution on the first level and finer grid corrections on higher levels. On finer grids, more FE coefficients need to be estimated than on coarser grids, but the contributions of the finer corrections to the full solutions are smaller than later corrective terms, see Section 5.2. Since the exponentially increasing number of FE coefficients to approximate on each level is countered by the exponential increase of required accuracy as derived in Eq. (10), approximately the same number of parameters on each level was chosen for the CNN. Furthermore, note that on lower levels ℓ, the $\mathrm{VCMR}^m_{k,k_0,\ell}$ has fewer recursive calls than on higher levels. Since one U-net with parameters on every considered level corresponds to one full call of the $\mathrm{VCMR}^m_{k,k_0,\ell}$, U-nets for lower levels need less parameters for a full downsampling

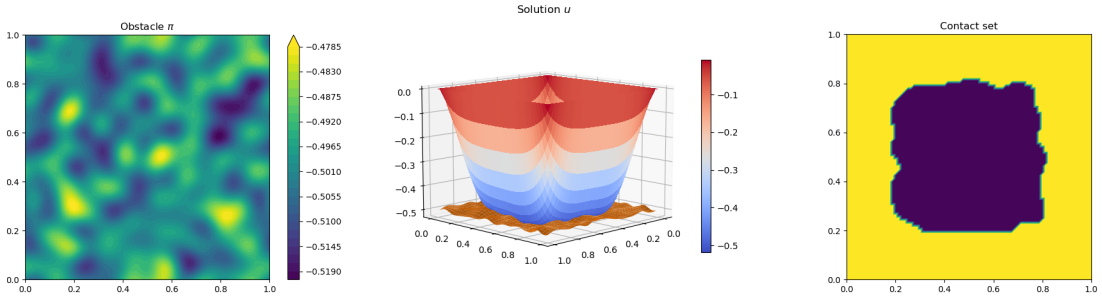

Figure 6.2: The first image depicts a realization of the rough surface model [30]. In the second and third images, the corresponding solution of the obstacle problem and the resulting contact set are shown, where the solution is equal to the obstacle. The contact set is colored in purple.

Table 1: Collection of parameters of the used CNN architecture CNN, including the network output dimensions of the FE solutions of the obstacle problem on different levels. For the complete solution, all outputs and therefore all parameters on all levels are needed (and summed up). Moreover, the number of parameters used in the form of concatenated U-nets on each level is displayed.

level	1	2	3	4	5	6	7
# params	1073248	1069984	1032992	1014496	838048	996000	1153952
# U-nets	11	6	4	3	2	2	2

and upsampling scheme. Therefore, on lower levels more U-nets are applied assuming the same number of trainable parameters as on higher levels.

For each test case, the *mean relative* $H^1(D)$ error and *mean relative* $L^2(D)$ error are calculated with respect to a finite element solution on the same grid as the neural network output and with respect to a (reference) fine grid solution. For $N = 1024$ test samples $\mathbf{y}_1, \dots, \mathbf{y}_N \in \Gamma$, let $u_1, \dots, u_N$ be the output of the neural network in V_L and let $v_L : \Gamma \to V_L$ be the finite element solution operator on the same grid. Furthermore, let $v_{\text{ref}} : \Gamma \to H^1$ be the finite element solution operator on a grid two times finer than the grid of v_L. Then, define the same grid network error and the reference error by

$$\mathcal{E}_{\text{MR}_*} = \sqrt{\frac{\sum_{i=1}^N \|u_i - v_L(\mathbf{y}_i)\|_*^2}{\sum_{i=1}^N \|v_L(\mathbf{y}_i)\|_*^2}}, \qquad \mathcal{E}^{\text{ref}}_{\text{MR}_*} = \sqrt{\frac{\sum_{i=1}^N \|u_i - v_{\text{ref}}(\mathbf{y}_i)\|_*^2}{\sum_{i=1}^N \|v_{\text{ref}}(\mathbf{y}_i)\|_*^2}}$$

with $* \in \{H^1, L^2\}$.

The training of the CNN was repeated 5 times and the mean results as well as the variances over the training procedures are recorded in Table 2 and Table 3. Table 2 shows the the errors in the H^1 norm. It can be observed that in all test cases the network error is significantly smaller than the reference error, in some cases it is even a magnitude smaller. Therefore, the overall (prediction or approximation) error could only efficiently be reduced by refining the grid. Furthermore, note that the error does not increase with respect to the parameter dimension p despite the problem becoming more challenging.

For the L^2 error in Table 3, the domination of the FE approximation error is not as pronounced. This

Table 2: The mean relative H^1 error is reported for the error of the network to Galerkin solutions on the same grid $\mathcal{E}_{\mathrm{MR}H^1}$ and with respect to the reference Galerkin solutions on a twice uniformly refined grid $\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}H^1}$.

problem	parameter dimension p	$\mathcal{E}_{\mathrm{MR}H^1}$	$\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}H^1}$
deterministic obstacle	10	$4.76 \times 10^{-4} \pm 2 \times 10^{-4}$	$6.82 \times 10^{-3} \pm 1 \times 10^{-5}$
	50	$4.35 \times 10^{-4} \pm 1 \times 10^{-4}$	$6.82 \times 10^{-3} \pm 8 \times 10^{-6}$
stochastic obstacle	11	$1.32 \times 10^{-3} \pm 4 \times 10^{-4}$	$6.92 \times 10^{-3} \pm 7 \times 10^{-5}$
	51	$1.91 \times 10^{-3} \pm 1 \times 10^{-3}$	$7.17 \times 10^{-3} \pm 4 \times 10^{-4}$
rough surface	100	$2.13 \times 10^{-3} \pm 6 \times 10^{-4}$	$8.59 \times 10^{-3} \pm 6 \times 10^{-4}$
	220	$2.07 \times 10^{-3} \pm 2 \times 10^{-4}$	$9.04 \times 10^{-3} \pm 3 \times 10^{-5}$

Table 3: The mean relative L^2 error is reported for the error of the network to Galerkin solutions on the same grid $\mathcal{E}_{\mathrm{MR}L^2}$ and to reference Galerkin solutions on a twice uniformly refined grid $\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}L^2}$.

problem	parameter dimension p	$\mathcal{E}_{\mathrm{MR}L^2}$	$\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}L^2}$
deterministic obstacle	10	$1.85 \times 10^{-4} \pm 5 \times 10^{-5}$	$1.96 \times 10^{-4} \pm 5 \times 10^{-5}$
	50	$1.41 \times 10^{-4} \pm 4 \times 10^{-5}$	$1.52 \times 10^{-4} \pm 4 \times 10^{-5}$
stochastic obstacle	11	$6.39 \times 10^{-4} \pm 2 \times 10^{-4}$	$6.43 \times 10^{-4} \pm 2 \times 10^{-4}$
	51	$1.01 \times 10^{-3} \pm 7 \times 10^{-4}$	$1.01 \times 10^{-3} \pm 6 \times 10^{-4}$
rough surface	100	$7.00 \times 10^{-4} \pm 4 \times 10^{-4}$	$7.09 \times 10^{-4} \pm 4 \times 10^{-4}$
	220	$5.40 \times 10^{-4} \pm 3 \times 10^{-5}$	$5.47 \pm 10^{-4} \pm 3 \times 10^{-5}$

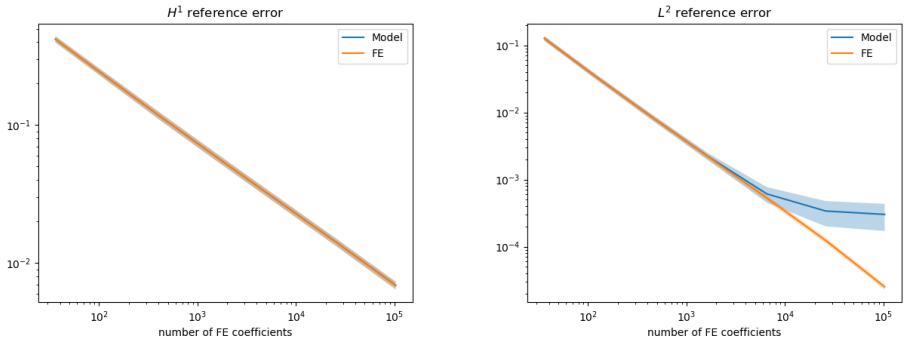

Figure 6.3: Error plots for the stochastic constant obstacle problem with parameter dimension $p = 11$ are shown for a trained CNN. Errors of the CNN output compared to a reference solution are plotted in blue and errors of the finite element solution on the same grid as the CNN output to the reference solution are plotted in orange. A line indicates the mean of the relative errors over a test set and the area visualizes its variance. The left plot shows H^1 errors and the right plot shows L^2 errors.

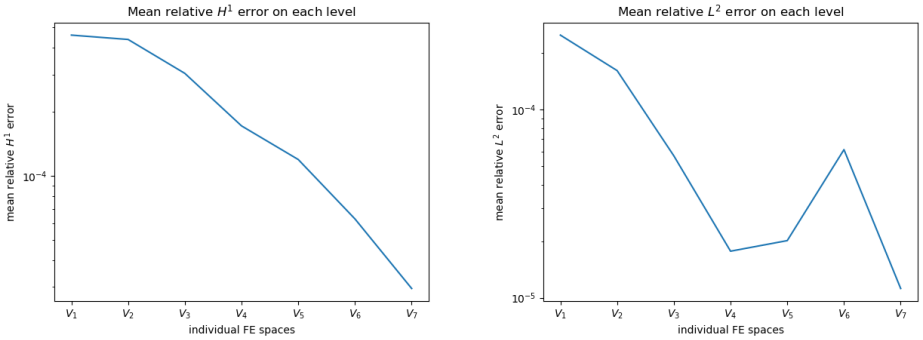

Figure 6.4: Mean relative H^1 (left) and L^2 (right) errors for a CNN trained for the stochastic constant obstacle problem with parameter dimension $p = 11$ are shown. The errors are plotted over the outputs of the network on each level in the multi-level decomposition. It can be seen that the multigrid corrections on fine grids are well approximated.

could be due to the fact that the network is trained with respect to the H^1 error. The difference in error convergence can also be seen in Figure 6.3, where the decay of the H^1 and L^2 errors for the obstacle problem with variable obstacle and parameter dimension 11 over the degrees of freedom is visualized. Both the error of the network and the reference solutions ($\|u_i - v_{\text{ref}}(\mathbf{y}_i)\|_*$) and the error of the FE solution on the same grids as the network output and the reference solution ($\|v_L(\mathbf{y}_i) - v_{\text{ref}}(\mathbf{y}_i)\|_*$) are considered. While the H^1 error achieves the same convergence rate as the true Galerkin solution on each refined grid, the L^2 error suffers from a bias in the last few levels.

The error contribution of the individual levels is illustrated in Figure 6.4. The mean relative H^1 and L^2 errors are visualized for the approximation of the correction in each discrete FE space. The approximated corrections are visualized in Figure 6.1. It can be seen that a high relative accuracy is achieved for the coefficients of corrections on high fidelity grids despite approximately the same number of parameters being used for each sub-network. This underlines the effectiveness of the multigrid decomposition.

7 Conclusion

A multi-level NN architecture was analyzed in the context of a parametric obstacle problem, a highly nonlinear and highly challenging extension to the studies in [21]. Here, it could be shown analytically and numerically that their proposed architecture is well suited to solve the considered parametric obstacle problem. An expressivity result for the architecture applied to the obstacle problem was derived, stating an upper bound on the number of trainable parameters, which depends only logarithmically on the required accuracy. The architecture was successfully applied to the parametric obstacle problem for multiple parameter-setups with different dimensions, with variable coefficient and with constant as well as variable and rough obstacles.

Acknowledgments

ME & JS acknowledge funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in the priority programme SPP 2298 ”Theoretical Foundations of Deep Learning”. ME acknowledges support by the ANR-DFG project COFNET: Compositional functions networks - adaptive learning for high-dimensional approximation and uncertainty quantification. This study does not have any conflicts to disclose.

References

[1] A. Alphonse, M. Hintermüller, A. Kister, C. H. Lun, and C. Sirotenko. A neural network approach to learning solutions of a class of elliptic variational inequalities. 2024.

[2] H. E. Bahja, J. C. Hauffen, P. Jung, B. Bah, and I. Karambal. A physics-informed neural network framework for modeling obstacle-related equations. *Nonlinear Dynamics*, 2025.

[3] A. J. Bernoff and C. M. Topaz. A primer of swarm equilibria. *SIAM Journal on Applied Dynamical Systems*, 10(1):212–250, 2011.

[4] C. Bierig and A. V. Chernov. Convergence analysis of multilevel monte carlo variance estimators and application for random obstacle problems. *Numerische Mathematik*, 130:579–613, 2015.

[5] A. Blanchet. On the regularity of the free boundary in the parabolic obstacle problem. application to american options. *Nonlinear Analysis: Theory, Methods & Applications*, 65(7):1362–1378, 2006.

[6] A. Blanchet, J. Dolbeault, and R. Monneau. On the continuity of the time derivative of the solution to the parabolic obstacle problem with variable coefficients. *Journal de Mathématiques Pures et Appliquées*, 85(3):371–414, 2006.

[7] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the V-cycle. *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, 20:967–975, 10 1983.

[8] J. Carrillo, M. Delgadino, and A. Mellet. Regularity of local minimizers of the interaction energy via obstacle problems. *Communications in Mathematical Physics*, 343, 05 2016.

[9] X. Cheng, X. Shen, X. Wang, and K. Liang. A deep neural network-based method for solving obstacle problems. *Nonlinear Analysis: Real World Applications*, 72:103864, 2023.

[10] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.

[11] M. R. D'Orsogna, Y. L. Chuang, A. L. Bertozzi, and L. S. Chayes. Self-propelled particles with soft-core interactions: Patterns, stability, and collapse. *Phys. Rev. Lett.*, 96:104302, Mar 2006.

[12] G. Duvaut. R´esolution d'un probl'eme de stefan (fusion d'un bloc de glace a zero degr´ees). *C. R. Acad. Sci. Paris*, 276:1461–1463, 1973.

[13] M. Eigel, R. Schneider, P. Trunschke, and S. Wolf. Variational monte carlo—bridging concepts of machine learning and high-dimensional partial differential equations. *Advances in Computational Mathematics*, 45(5–6):2503–2532, Oct. 2019.

[14] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 06 2014.

[15] R. Fetecau, Y. Huang, and T. Kolokolnikov. Swarm dynamics and equilibria for a nonlocal aggregation model. *Nonlinearity*, 24:2681, 08 2011.

[16] R. Forster and R. Kornhuber. A polynomial chaos approach to stochastic variational inequalities. *Journal of Numerical Mathematics*, 18(4):235–255, 2010.

[17] A. Friedman. Variational principles and free-boundary problems. In *Variational and Free Boundary Problems*, Pure and applied mathematics, New York u.a., 1982. Wiley.

[18] B. Geshkovski. Obstacle problems: theory and applications. Master's thesis, Université de Bordeaux, 2018.

[19] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer Berlin, Heidelberg, 1 edition, 1984.

[20] C. Gräser and R. Kornhuber. Multigrid methods for obstacle problems. *Journal of Computational Mathematics*, 27(1):1–44, 2009.

[21] C. Heiß, I. Gühring, and M. Eigel. Multilevel cnns for parametric pdes. *Journal of Machine Learning Research*, 24(373):1–42, 2023.

[22] D. D. Holm and V. Putkaradze. Aggregation of finite-size particles with variable mobility. *Physical review letters*, 95 22:226106, 2005.

[23] D. D. Holm and V. Putkaradze. Formation of clumps and patches in self-aggregation of finite-size particles. *Physica D: Nonlinear Phenomena*, 220(2):183–196, 2006.

[24] R. Kornhuber. On constrained newton linearization and multigrid for variational inequalities. *Numerische Mathematik*, 91:699–721, 2002.

[25] R. Kornhuber, C. Schwab, and M.-W. Wolf. Multilevel monte carlo finite element methods for stochastic elliptic variational inequalities. *SIAM Journal on Numerical Analysis*, 52(3):1243–1268, 2014.

[26] R. Kornhuber and E. Youett. Adaptive multilevel monte carlo methods for stochastic variational inequalities. *SIAM Journal on Numerical Analysis*, 56(4):1987–2007, 2018.

[27] P. Laurence and S. Salsa. Regularity of the free boundary of an american option on several assets. *Communications on Pure and Applied Mathematics*, 62(7):969–994, 2009.

[28] K. O. LYE, S. MISHRA, and R. MOLINARO. A multi-level procedure for enhancing accuracy of machine learning algorithms. *European Journal of Applied Mathematics*, 32(3):436–469, 2021.

[29] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. 2018.

[30] B. N. J. Persson, O. Albohr, U. Tartaglino, A. I. Volokitin, and E. Tosatti. On the nature of surface roughness with application to contact mechanics, sealing, rubber friction and adhesion. *Journal of Physics: Condensed Matter*, 17(1):R1, dec 2004.

[31] X. Ros-Oton. Obstacle problems and free boundaries: an overview. *SeMA*, 2017.

[32] J. E. Schütte and M. Eigel. Adaptive multilevel neural networks for parametric PDEs with error estimation. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024.

[33] C. Schwab and A. Stein. Deep solution operators for variational inequalities via proximal neural networks. *Research in the Mathematical Sciences*, 9, 06 2022.

[34] J. E. Schütte and M. Eigel. Multilevel cnns for parametric pdes based on adaptive finite elements, 2024.

[35] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*. Springer Berlin, Heidelberg, 2 edition, 2006.

[36] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Oxford University Press, 04 2013.

[37] Y. Zhang. Multilevel projection algorithm for solving obstacle problems. *Computers & Mathematics with Applications*, 41(12):1505–1513, 2001.

[38] X. E. Zhao, W. Hao, and B. Hu. Two neural-network-based methods for solving elliptic obstacle problems. *Chaos, Solitons & Fractals*, 161(C), 2022.

A Proofs of expressivity theory

Proof of Theorem 4.1. First, assume some $\lambda : D \to \mathbb{R}$ with $\lambda = 0$ on ∂D that is continuous and Lebesgue-almost-everywhere differentiable. If λ is not constant zero this implies that there exists a point $x_0 \in D$ and $\varepsilon > 0$ such that λ is differentiable and $\nabla\lambda \neq 0$ on an ε neighborhood of x_0 denoted b $U_\varepsilon(x_0)$. Then, due to $\kappa(\cdot, \mathbf{y}) > 0$ everywhere, we obtain that

$$a_{\mathbf{y}}(\lambda, \lambda) = \int_D \kappa(\cdot, \mathbf{y}) \langle \nabla\lambda, \nabla\lambda \rangle \, \mathrm{d}x \geq \int_{U_\varepsilon(x_0)} \kappa(\cdot, \mathbf{y}) \langle \nabla\lambda, \nabla\lambda \rangle \, \mathrm{d}x > 0.$$

Therefore, for any nonzero $\mathbf{w} \in \mathbb{R}^N$ we deduce that

$$\mathbf{w}^\intercal A_{\mathbf{y}} \mathbf{w} = a_{\mathbf{y},k} \left(\sum_{i=1}^N \mathbf{w}_i \lambda_i, \sum_{i=1}^N \mathbf{w}_i \lambda_i \right) > 0$$

and hence $A_{\mathbf{y}}$ is positive definite. Denote the eigenvalues and eigenvectors of $A_{\mathbf{y}}$ by $\sigma_i, \mathbf{v}^i$ for $i = 1, \dots, N$ with

$$\begin{aligned} A_{\mathbf{y}} \mathbf{v}^i &= \sigma_i \mathbf{v}^i \qquad \text{such that} \\ \delta_{i,j} &= \left\langle \mathbf{v}^i, \mathbf{v}^j \right\rangle_{\ell^2} \text{ for all } i, j = 1, \dots, N. \end{aligned}$$

Furthermore, for $\mathbf{w} \in \mathbb{R}^N$, let

$$J_{\mathbf{y}} \mathbf{w} \coloneqq (I - \omega_{\mathbf{y}} A_{\mathbf{y}}) \mathbf{w}.$$

Then, with $\mathbf{w} = \sum_{i=1}^N c_i \mathbf{v}^i$

$$J_{\mathbf{y}} \mathbf{w} = \sum_{i=1}^N c_i (I - \omega_{\mathbf{y}} A_{\mathbf{y}}) \mathbf{v}^i = \sum_{i=1}^N c_i (1 - \omega_{\mathbf{y}} \sigma_i) \mathbf{v}^i.$$

Additionally,

$$|\mathbf{w}|^2 \coloneqq \sum_{i=1}^N \sigma_i (1 - \sigma_i \omega_{\mathbf{y}}) c_i^2$$

defines a semi-norm due to $0 < \omega_{\mathbf{y}} \leq \sigma_{\max}(A_{\mathbf{y}})^{-1}$. It then follows that

$$|\mathbf{w}|^2 = \sum_{i,j=1}^N (1 - \sigma_i \omega_{\mathbf{y}}) \sigma_i \left\langle \mathbf{v}^i, \mathbf{v}^j \right\rangle_{\ell^2} c_i c_j = \left\langle \sum_{i=1}^N c_i (1 - \sigma_i \omega_{\mathbf{y}}) \mathbf{v}^i, \sum_{j=1}^N c_j \mathbf{v}^j \right\rangle_{A_{\mathbf{y}}} = \langle J_{\mathbf{y}} \mathbf{w}, \mathbf{w} \rangle_{A_{\mathbf{y}}},$$

$$\|\mathbf{w}\|_{A_{\mathbf{y}}}^2 = \langle A_{\mathbf{y}} \mathbf{w}, \mathbf{w} \rangle_{\ell^2} = \sum_{i,j=1}^N \sigma_i c_i c_j \left\langle \mathbf{v}^i, \mathbf{v}^j \right\rangle_{\ell^2} = \sum_{i=1}^N c_i^2 \sigma_i,$$

$$|\mathbf{w}|^2 = \sum_{i=1}^N \sigma_i c_i^2 - \sum_{i=1}^N \sigma_i \omega_{\mathbf{y}} c_i^2 = (1 - \omega_{\mathbf{y}}) \sum_{i=1}^N \sigma_i c_i^2 = (1 - \omega_{\mathbf{y}}) \|\mathbf{w}\|_{A_{\mathbf{y}}}^2.$$

With the Hölder inequality,

$$\begin{aligned}\|J_{\mathbf{y}}\mathbf{w}\|_{A_{\mathbf{y}}}^2 &= \sum_{i=1}^{N}\sigma_i(c_i(1-\sigma_i\omega_{\mathbf{y}}))^2 = \sum_{i=1}^{N}(\sigma_i^{1/3}|c_i|^{2/3})(\sigma_i^{2/3}|c_i|^{4/3}(1-\sigma_i\omega_{\mathbf{y}})^2)\\ &\leq \left(\sum_{i=1}^{N}(\sigma_i^{1/3}|c_i|^{2/3})^3\right)^{1/3}\left(\sum_{i=1}^{N}(\sigma_i^{2/3}|c_i|^{4/3}(1-\sigma_i\omega_{\mathbf{y}})^2)^{3/2}\right)^{2/3}\\ &= \left(\sum_{i=1}^{N}\sigma_i|c_i|^2\right)^{1/3}\left(\sum_{i=1}^{N}\sigma_i|c_i|^2(1-\sigma_i\omega_{\mathbf{y}})^3\right)^{2/3}.\end{aligned}$$

This yields the result by estimating

$$\begin{aligned}\|J_{\mathbf{y}}\mathbf{w}\|_{A_{\mathbf{y}}}^3 &= \left(\|J_{\mathbf{y}}\mathbf{w}\|_{A_{\mathbf{y}}}^2\right)^{3/2}\\ &\leq \left(\sum_{i=1}^{N}\sigma_i c_i^2\right)^{1/2}\left(\sum_{i=1}^{N}\sigma_i c_i^2(1-\sigma_i\omega_{\mathbf{y}})^3\right)\\ &= \|\mathbf{w}\|_{A_{\mathbf{y}}}|J_{\mathbf{y}}\mathbf{w}|^2\\ &= \|\mathbf{w}\|_{A_{\mathbf{y}}}(1-\omega_{\mathbf{y}})\|J_{\mathbf{y}}\mathbf{w}\|_{A_{\mathbf{y}}}^2\end{aligned}$$

and dividing by $\|J_{\mathbf{y}}\mathbf{w}\|_{A_{\mathbf{y}}}^2$. □

Lemma A.1 (Maxima approximation)**.** *Let $\tau:\mathbb{R}\to\mathbb{R}$ satisfy Theorem 5.1 $\varepsilon, M>0$ and $n\in\mathbb{N}$. There exist convolutional $(1,1)$-kernels $K_1\in\mathbb{R}^{2\times2\times1\times1}$ and $K_2\in\mathbb{R}^{2\times1\times1\times1}$ and biases $B_1\in\mathbb{R}^2, B_2\in\mathbb{R}$ such that*

$$\|(\psi_{K_2,B_2}\circ\tau\circ\psi_{K_1,B_1})(\mathbf{u},\boldsymbol{\varphi})-\max\{\mathbf{u},\boldsymbol{\varphi}\}\|_{L^\infty([-M,M]^{2\times n\times n})}\leq\varepsilon,$$

where the maximum is defined component-wise.

Proof. Note that $\max\{\mathbf{u},\boldsymbol{\varphi}\} = \max\{\mathbf{u}-\boldsymbol{\varphi},0\}+\boldsymbol{\varphi}$ and $\boldsymbol{\varphi} = \max\{\boldsymbol{\varphi},0\}+\max\{-\boldsymbol{\varphi},0\}$. Therefore, an approximation of the mapping

$$\begin{pmatrix}\mathbf{u}\\ \boldsymbol{\varphi}\end{pmatrix}\mapsto\begin{pmatrix}\mathbf{u}-\boldsymbol{\varphi}\\ \boldsymbol{\varphi}\end{pmatrix}\mapsto\begin{pmatrix}\max\{\mathbf{u}-\boldsymbol{\varphi},0\}\\ \max\{\boldsymbol{\varphi},0\}\\ \max\{-\boldsymbol{\varphi},0\}\end{pmatrix}\mapsto\big(\max\{\mathbf{u}-\boldsymbol{\varphi},0\}+\boldsymbol{\varphi}\big)$$

yields the claim. The addition of channels in the first and last step of the flow can be represented by kernels of width and height 1 with the appropriate number of input and output channels. According to Theorem 5.1, there exist affine linear mappings $\varrho_1,\varrho_2:\mathbb{R}\to\mathbb{R}$ such that the maximum of the input and 0 can be approximated on $[-2M,2M]$. Accounting for $\mathbf{u}-\boldsymbol{\varphi}\in[-2M,2M]$ for $\mathbf{u},\boldsymbol{\varphi}\in[-M,M]$ implies that applying these maps to the appropriate channels provides an approximation of the second step. Concatenating these maps shows the claim since the kernels have width and height 1 and can therefore be concatenated to one kernel of width and height 1. □

Proof of Theorem 5.2. Let $\mathcal{T}_h$ be a uniform triangulation of D, where each node i is in 6 triangles $T_i^1, \dots, T_i^6$ as depicted in [21, Figure 4] and let V_h be the P1 finite element space over the triangulation. As in [21, Definition 14], let $\boldsymbol{\Upsilon}(\kappa_h, k, i) \coloneqq \int_{T_i^k} \kappa_h \,\mathrm{d}x$ and $\boldsymbol{\Upsilon}(\kappa_h) \coloneqq [\boldsymbol{\Upsilon}(\kappa_h, k, i)]_{k\in[6], i\in[\dim V_h]} \in \mathbb{R}^{6\times \dim V_h}$. Moreover, let $F : \mathbb{R}^{7\times \dim V_h} \to \mathbb{R}^{\dim V_h}$ be defined as in [21, Theorem 16]. with $F(\mathbf{u}, \boldsymbol{\Upsilon}(\kappa_h)) = A_\kappa \mathbf{u}$. We consider a CNN approximating the following steps.

$$\begin{pmatrix} \mathbf{u}^{(0)} \\ \boldsymbol{\kappa} \\ \mathbf{f} \\ \boldsymbol{\varphi} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{u}^{(0)} \\ \boldsymbol{\Upsilon}(\kappa) \\ \mathbf{f} \\ \boldsymbol{\varphi} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{u}^{(1)} \coloneqq \max\left\{\mathbf{u}^{(0)} + \omega\left[\mathbf{f} - F(\mathbf{u}^{(0)}, \boldsymbol{\Upsilon}(\kappa))\right], \boldsymbol{\varphi}\right\} \\ \boldsymbol{\Upsilon}(\kappa) \\ \mathbf{f} \\ \boldsymbol{\varphi} \end{pmatrix}$$
$$\mapsto \dots \mapsto \begin{pmatrix} \mathbf{u}^{(m)} \\ \boldsymbol{\Upsilon}(\kappa) \\ \mathbf{f} \\ \boldsymbol{\varphi} \end{pmatrix} \mapsto \left(\mathbf{u}^{(m)}\right)$$

According to [21, Lemma 15(i)], there exists a convolutional kernel $K_1 \in \mathbb{R}^{1\times 6\times 3\times 3}$ such that $\boldsymbol{\kappa} * K_1 = \boldsymbol{\Upsilon}(\kappa_h)$. Trivially extending the kernel to unused channels and parallelizing with identity kernels yields a CNN representation of the first step. According to [21, Theorem 18], for any $\tilde{\varepsilon}, \tilde{M} > 0$ there exists a CNN $\Psi_{\tilde{\varepsilon},\tilde{M}} : \mathbb{R}^{7\times \dim V_h} \to \mathbb{R}^{\dim V_h}$ of size independent of $\tilde{\varepsilon}, \tilde{M}$ such that

$$\left\| \Psi_{\tilde{\varepsilon},\tilde{M}} - F \right\|_{L^\infty([-\tilde{M},\tilde{M}]^{7\times \dim V_h})} \leq \tilde{\varepsilon}.$$

Furthermore, Theorem A.1 shows, that the maximum can be approximated by a CNN. Again extending the CNN to unaltered channels and parallelizing with an identity CNN yields approximations of every intermediate step. The last step is realized by the convolution with a kernel $K_2 \in \mathbb{R}^{9\times 1\times 1\times 1}$, which is 1 in the first and 0 in all other input channels.

According to [21, Lemma 20] and since all steps are continuous operations, their concatenation can be approximated by the concatenation $\tilde{\Psi}$ of the approximating CNNs $\Psi_{\tilde{\varepsilon},\tilde{M}}$, where different approximation accuracies $\tilde{\varepsilon}$ and domains $\tilde{M}$ are expected for the different steps. Concatenating a one layer CNN with kernel K_1 and $\tilde{\Psi}$ and another one layer CNN with kernel K_2 yields a CNN Ψ. Then, for each $\mathbf{u}^{(0)}, \boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi} \in [-M, M]^{n\times n}$ it holds that

$$\left\| \Psi(\mathbf{u}^{(0)}, \boldsymbol{\kappa}, \mathbf{f}, \boldsymbol{\varphi}) - \mathbf{u}^{(m)} \right\|_\infty \leq \varepsilon.$$

Since the space of FE coefficients is a finite dimensional real vector space, the $\|\cdot\|_\infty$-norm is equivalent to the $\|\cdot\|_{H^1}$-norm. Since the size of the approximating CNNs does not depend on $\tilde{\varepsilon}, \tilde{M}$, the size of the concatenated CNN also does not depend on it and the overall number of parameters only grows linearly with the number of intermediate steps m. □

Proof of Theorem 5.3. Let $\omega \coloneqq (\mathfrak{C} C_{V_h})^{-1}$. Then Theorem 4.2 yields $\|I - \omega A_{\mathbf{y}}\|_{A_{\mathbf{y}}} < 1 - \omega$ for all $\mathbf{y} \in \Gamma$. For $\mathbf{y} \in \Gamma$, (9) implies for $A \coloneqq A_{\mathbf{y}}$ and for $\mathbf{e}^{(m)} \coloneqq \mathbf{u}^{(m)} - \mathbf{u}$ (where $\mathbf{u}$ solves Theorem 3.2) that for $\gamma \coloneqq 1 - \omega$,

$$\left\| \mathbf{e}^{(m)} \right\|_{A_{\mathbf{y}}} \leq \gamma^m \left\| \boldsymbol{\varphi} - \mathbf{u} \right\|_{A_{\mathbf{y}}}. \tag{11}$$

Note that the second term can be bounded by the following consideration.

$$\begin{aligned}\|\boldsymbol{\varphi}-\mathbf{u}\|_{A_\mathbf{y}}^2 &= \int_D \kappa_h(\cdot,\mathbf{y})\,\langle\nabla(\varphi_h-u_h),\nabla(\varphi_h-u_h)\rangle\,\mathrm{d}x\\ &= -\int_D \kappa_h(\cdot,\mathbf{y})\,\langle\nabla u_h,\nabla(\varphi_h-u_h)\rangle\,\mathrm{d}x+\int_D \kappa_h(\cdot,\mathbf{y})\,\langle\nabla\varphi_h,\nabla(\varphi_h-u_h)\rangle\,\mathrm{d}x.\end{aligned}$$

Since u_h solves Theorem 3.1, this implies

$$\begin{aligned}\|\varphi_h-u_h\|_{A_\mathbf{y}}^2 &= \int_D \kappa(\cdot,\mathbf{y})\,\langle\nabla(\varphi_h-u_h),\nabla(\varphi_h-u_h)\rangle\,\mathrm{d}x\\ &= -\int_D \kappa(\cdot,\mathbf{y})\,\langle\nabla u_h,\nabla(\varphi_h-u_h)\rangle\,\mathrm{d}x+\int_D \kappa(\cdot,\mathbf{y})\,\langle\nabla\varphi_h,\nabla(\varphi_h-u_h)\rangle\,\mathrm{d}x\\ &\le -\int_D f(x)(\varphi_h-u_h)(x)\,\mathrm{d}x+\|\varphi_h\|_{A_\mathbf{y}}\,\|\varphi_h-u_h\|_{A_\mathbf{y}}\\ &\le \|f\|_{H^{-1}}\,\|\varphi_h-u_h\|_{H^1}+\frac{1}{c_{H^1}}\,\|\varphi_h\|_{H^1}\,\|\varphi_h-u_h\|_{A_\mathbf{y}}\\ &\le C_{H^1}\,\|f\|_{H^{-1}}\,\|\varphi_h-u_h\|_{A_\mathbf{y}}+\frac{1}{c_{H^1}}\,\|\varphi_h\|_{H^1}\,\|\varphi_h-u_h\|_{A_\mathbf{y}},\end{aligned}$$

where the fist inequality follows with the definition of the dual norm (2) and the Cauchy-Schwarz inequality. The second and last inequality follow from (3). With (7) the second term is bounded by

$$\|\boldsymbol{\varphi}-\mathbf{u}\|_{A_\mathbf{y}}=\|\varphi_h-u_h\|_{A_\mathbf{y}}\le C_{H^1}\,\|f\|_{H^{-1}}+\frac{1}{c_{H^1}}\,\|\varphi_h\|_{H^1}. \tag{12}$$

To bound γ^m, m can be chosen as

$$m\ge \log\left(\varepsilon/2\right)\log\left(\gamma\right)^{-1},$$

such that (11) and (12) lead to

$$\left\|\mathbf{u}^{(m)}-\mathbf{u}\right\|_{H^1}\le\frac{1}{c_{H^1}}\left\|\mathbf{u}^{(m)}-\mathbf{u}\right\|_{A_\mathbf{y}}\le\frac{\varepsilon}{2c_{H^1}}\left(C_{H^1}\,\|f\|_{H^{-1}}+\frac{1}{c_{H^1}}\,\|\varphi_h\|_{H^1}\right).$$

According to Theorem 5.2 there exists a CNN Ψ such that the number of parameters grows linearly with m and

$$\left\|\Psi(\boldsymbol{\varphi},\boldsymbol{\kappa},\mathbf{f},\boldsymbol{\varphi})-\mathbf{u}^{(m)}\right\|_{H^1}\le\frac{\varepsilon}{2c_{H^1}}\left(C_{H^1}\,\|f\|_{H^{-1}}+\frac{1}{c_{H^1}}\,\|\varphi_h\|_{H^1}\right).$$

This yields the claim with

$$\begin{aligned}\|\Psi(\boldsymbol{\varphi},\boldsymbol{\kappa},\mathbf{f},\boldsymbol{\varphi})-\mathbf{u}\|_{H^1} &\le \left\|\Psi(\boldsymbol{\varphi},\boldsymbol{\kappa},\mathbf{f},\boldsymbol{\varphi})-\mathbf{u}^{(m)}\right\|_{H^1}+\left\|\mathbf{u}^{(m)}-\mathbf{u}\right\|_{H^1}\\ &\le\frac{\varepsilon}{c_{H^1}}\left(C_{H^1}\,\|f\|_{H^{-1}}+\frac{1}{c_{H^1}}\,\|\varphi_h\|_{H^1}\right).\end{aligned}$$

□

Theorem A.2 (CNN for monotone restriction operator)**.** *For $\ell = 1, \dots, L-1$, let $V_\ell \subseteq V_\ell + 1 \subseteq H_0^1([0,1]^2)$ be two nested P1 finite element spaces as in Section 2. Let the activation function τ satisfy Theorem 5.1 and $R_\ell^{\max}$ be the monotone restriction operator defined as in Theorem 4.4. There exists a $C > 0$ such that for every $\varepsilon > 0$ there exists a CNN Ψ and kernel $K \in \mathbb{R}^{1\times 9\times 3\times 3}$ such that*

(i) $\|(\Psi \circ *_K^{2s})(\boldsymbol{\varphi}) - R_\ell^{\max}\boldsymbol{\varphi}\|_{L^\infty([-M,M]^{n\times n})} \leq \varepsilon$,

(ii) the number of parameters is bounded by $M(\Psi) + M(K) \leq C$.

Proof. Let $K \in \mathbb{R}^{1\times 9\times 3\times 3}$ be defined such that for each output channel $i = 1, \dots, 9$ one kernel entry is one and all other are zero. Applying this kernel in a two-strided convolution, a FE function on a fine grid as the input image gets mapped to nine FE functions on the coarse grid, i.e. an output tensor with nine channels. For each grid point in the coarse grid, these nine channels contain the values of the grid points in the fine grid surrounding the same grid point. By the definition of the monotone restriction, the maximum needs to be taken over all channels in each pixel. This can be achieved by applying Theorem A.1. Denote the maximum of $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$ by $\boldsymbol{\varphi}_{1,2}$. Let $\Psi_1, \dots, \Psi_4$ be CNNs approximating the steps

$$\begin{pmatrix} \boldsymbol{\varphi}_1 \\ \dots \\ \boldsymbol{\varphi}_9 \end{pmatrix} \mapsto \begin{pmatrix} \max\{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2\} \\ \max\{\boldsymbol{\varphi}_3, \boldsymbol{\varphi}_4\} \\ \dots \end{pmatrix} =: \begin{pmatrix} \boldsymbol{\varphi}_{1,2} \\ \dots \\ \boldsymbol{\varphi}_{7,8} \\ \boldsymbol{\varphi}_9 \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{\varphi}_{1,2,3,4} \\ \boldsymbol{\varphi}_{5,6,7,8} \\ \boldsymbol{\varphi}_9 \end{pmatrix}$$
$$\mapsto \begin{pmatrix} \boldsymbol{\varphi}_{1,2,3,4} \\ \boldsymbol{\varphi}_{5,6,7,8,9} \end{pmatrix} \mapsto \left(\boldsymbol{\varphi}_{1,2,3,4,5,6,7,8,9}\right).$$

Since the concatenation of functions can be approximated by the concatenation of approximate functions [21, Lemma 20] and the size of the kernels does not depend on the required accuracy, the size of the concatenated CNN does not depend on the ε. Since each step can be approximated with two kernels of height and width one and one application of the activation function, the secondly applied kernel of Ψ_k and the firstly applied kernel of Ψ_{k+1} can be concatenated to one kernel performing both convolutions. This leads to a CNN Ψ with 4 applications of the activation functions and 5 kernels of height and width one. This yields the claim. □

Proof of Theorem 5.4. The proof can be carried out similarly to [21, Proof of Theorem 6]. Here, the obstacle has to be taken into account in every step. For each $\ell = 1, \dots, L$ and $g : \mathbb{R}^{n\times \dim V_\ell} \to \mathbb{R}^{m\times \dim V_\ell}$, let the function also considering the obstacle φ be defined by $\tilde{g} : \mathbb{R}^{(n+1)\times \dim V_\ell} \to \mathbb{R}^{(m+1)\times \dim V_\ell}$ with $(\mathbf{w}, \boldsymbol{\varphi}) \mapsto (g(\mathbf{w}), \boldsymbol{\varphi})$. Adapting the proof to use the extended functions for $f_{\mathsf{in}}, f_{\mathsf{sm}}^\ell, f_{\mathsf{resi}}^\ell$, only a few changes in the smoothing iterations and projections have to be made in the proof [21, Proof of theorem 6] as detailed in the following.

(ii) **Smoothing iteration:** For $\ell = 1, \dots, L$ a function f_{sm}^ℓ is defined as a mapping from the current discrete solution $\mathbf{u}$, the integrated coefficient $\boldsymbol{\Upsilon}$ and the discretized right-hand side $\mathbf{f}$ to an updated discrete solution. The other inputs are unaltered $f_{\mathsf{sm}}^\ell : \mathbb{R}^{8\times \dim V_\ell} \to \mathbb{R}^{8\times \dim V_\ell}$,

$$f_{\mathsf{sm}}^\ell(\mathbf{u}, \boldsymbol{\Upsilon}, \mathbf{f}) = [\mathbf{u} + \omega(\mathbf{f} - A_{\mathbf{y}}^\ell \mathbf{u}), \boldsymbol{\Upsilon}, \mathbf{f}].$$

In [21, Theorem 18] it is shown that this function can be approximated up to an L^∞ error bounded by some $\varepsilon > 0$ on a compact subset of D, where the number of parameters is independent of the the size of the compact set and the accuracy ε. To use the projected Richardson method for the smoothing iterations, this function has to be adjusted to map another input to itself. For this, define $\tilde{f}^\ell_{\mathsf{sm}} : \mathbb{R}^{9\times \dim V_\ell} \to \mathbb{R}^{9\times \dim V_\ell}$ with

$$\tilde{f}^\ell_{\mathsf{sm}}(\mathbf{u}, \boldsymbol{\Upsilon}, \mathbf{f}, \boldsymbol{\varphi}) = [\mathbf{u} + \omega(\mathbf{f} - A^\ell_{\mathbf{y}}\mathbf{u}), \boldsymbol{\Upsilon}, \mathbf{f}, \boldsymbol{\varphi}].$$

Next, the maximum of the first and last input has to be taken. Hence, let $f^\ell_{\max} : \mathbb{R}^{9\times \dim V_\ell} \to \mathbb{R}^{9\times \dim V_\ell}$ be defined by

$$f^\ell_{\max}(\mathbf{u}, \boldsymbol{\Upsilon}, \mathbf{f}, \boldsymbol{\varphi}) = [\max\{\mathbf{u}, \boldsymbol{\varphi}\}, \boldsymbol{\Upsilon}, \mathbf{f}, \boldsymbol{\varphi}].$$

Then, Theorem A.1 implies that $f^\ell_{\max}$ can be approximated up to accuracy $\varepsilon > 0$ on a compact set in the L^∞ norm with the number of parameters independent of the accuracy ε and the size of the compact set. Concatenating the last two functions yields an update of the projected Richardson iteration (8) of the form

$$(f^\ell_{\max} \circ \tilde{f}^\ell_{\mathsf{sm}})(\mathbf{u}, \boldsymbol{\Upsilon}, \mathbf{f}, \boldsymbol{\varphi}) = [\max\{\mathbf{u} + \omega(\mathbf{f} - A^\ell_{\mathbf{y}}\mathbf{u}), \boldsymbol{\varphi}\}, \boldsymbol{\Upsilon}, \mathbf{f}, \boldsymbol{\varphi}]$$

and can be approximated similarly to [21, Lemma 20].

(iii) **Restricted residual:** The residual can be calculated as in [21, Proof of Theorem 6 (iii)] by altering the function to include the obstacle $\tilde{f}^\ell_{\mathsf{resi}}$. The restriction function f_{rest} is adjusted to also restrict the obstacle using the monotone restriction operator for the obstacle input. For each $\mathbf{u}, \mathbf{r}, \mathbf{f}, \boldsymbol{\varphi} \in \mathbb{R}^{\dim V_\ell}$, $\bar{\boldsymbol{\kappa}} \in \mathbb{R}^{6\times \dim V_\ell}$, define

$$\tilde{f}^\ell_{\mathsf{rest}} : \mathbb{R}^{10\times \dim V_\ell} \to \mathbb{R}^{8\times \dim V_\ell} \times \mathbb{R}^{8\times \dim V_{\ell-1}}, \begin{pmatrix} \mathbf{u} \\ \mathbf{r} \\ \bar{\boldsymbol{\kappa}} \\ \mathbf{f} \\ \boldsymbol{\varphi} \end{pmatrix} \mapsto \left[\begin{pmatrix} \mathbf{u} \\ \bar{\boldsymbol{\kappa}} \\ \mathbf{f} \\ \boldsymbol{\varphi} \end{pmatrix} \times \begin{pmatrix} \mathbf{0} \\ \bar{\boldsymbol{\kappa}} * K \\ P^T_{\ell-1}\mathbf{r} \\ R^{\max}_\ell \boldsymbol{\varphi} \end{pmatrix} \right],$$

where $K = [K_1, \dots, K_6] \in \mathbb{R}^{6\times 6\times 3\times 3}$ is defined as in the proof mentioned above. There, it is shown that $\boldsymbol{\Upsilon}(\kappa_h, \mathcal{T}^\ell, k) * K_k = \boldsymbol{\Upsilon}(\kappa_h, \mathcal{T}^{\ell-1}, k)$ for each $k = 1, \dots, 6$ and except for $R^{\max}_\ell \boldsymbol{\varphi}$ each part can be represented by a convolutional kernel. The monotone restriction can be represented due to Theorem A.2.

As in [21, Proof of Theorem 6], the operations of one V-Cycle on level $\ell = 2, \dots, L$ can then be written as the concatenation of the introduced functions

$$\mathrm{VC}^\ell_{k_0,k} := \left(\bigcirc_{i=1}^k \left(f^\ell_{\max} \circ \tilde{f}^\ell_{\mathsf{sm}}\right)\right) \circ \tilde{f}^\ell_{\mathsf{prol}} \circ \left(\mathrm{Id}, \mathrm{VC}^{\ell-1}_{k_0,k}\right) \circ \tilde{f}^\ell_{\mathsf{rest}} \circ \tilde{f}^\ell_{\mathsf{resi}} \circ \left(\bigcirc_{i=1}^k (f^\ell_{\max} \circ \tilde{f}^\ell_{\mathsf{sm}})\right),$$
$$\mathrm{VC}^1_{k_0,k} := \bigcirc_{i=1}^{k_0} \tilde{f}^1_{\mathsf{sm}}.$$

To represent the whole algorithm, the input and output are adjusted like

$$\mathrm{VCMR}^m_{k,k_0,\ell} = \tilde{f}_{\mathsf{out}} \circ \left(\bigcirc_{i=1}^m \mathrm{VC}^L_{k_0,k}\right) \circ \tilde{f}_{\mathsf{in}}.$$

□