ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Deep Solution Operators for Variational Inequalities via Proximal Neural Networks

C. Schwab and A. Stein

Research Report No. 2021-37
November 2021

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Deep Solution Operators for Variational Inequalities via Proximal Neural Networks

Christoph Schwab * Andreas Stein *†

November 15, 2021

Abstract

We introduce ProxNet, a collection of deep neural networks with ReLU activation which emulate numerical solution operators of variational inequalities (VIs). We analyze the expression rates of ProxNets in emulating solution operators for variational inequality problems posed on closed, convex cones in separable Hilbert spaces, covering the classical contact problems in mechanics, and early exercise problems as arise, e.g. in valuation of American-style contracts in Black-Scholes financial market models. In the finite-dimensional setting, the VIs reduce to matrix VIs in Euclidean space, and ProxNets emulate classical projected matrix iterations, such as PSOR and semi-smooth Newton iterations which are realized as primal-dual active set strategies, which we encode in the novel PDASNet.

1 Introduction

Variational Inequalities (VIs for short) in infinite-dimensional spaces arise in variational formulations of numerous models in the sciences. We refer only to [17, 7] and the references there for models of contact problems in continuum mechanics, [20] and the references there for applications from optimal stopping in finance (mainly option pricing with "American-style", early exercise features), contact problems in mechanics (e.g. [26] and the references there), and [4] and the references there for ressource allocation and game theoretic models. Two broad classes of approaches toward numerical solution of VIs can be identified: *deterministic* approaches, which are based on discretization of the VI in function space, and *probabilistic* approaches, which exploit stochastic numerical simulation and an interpretation of the solution of the VI as conditional expectations of optimally stopped sample paths. The latter approach has been used to design ML algorithms for the approximation of the solution of one instance of the VI in [3].

Deep neural network structures arise naturally in abstract variational inequality problems (VIs) posed on the product of (possibly infinite-dimensional) Hilbert spaces in [5]. Therein, the activation functions correspond to proximity operators of certain potentials that define the constraints of the VI. Weak convergence of this recurrent NN structure in the limit of infinite depth to feasible solutions of the VI is shown under suitable assumptions. An independent, but related, development in recent years has been the advent of DNN-based numerical approximations which are based on encoding known, iterative solvers for discretized partial differential equations, and certain fixed point iterations for nonlinear operator equations. We mention only [9], that developed DNNs which emulate the ISTA iteration of [6], or the more recently proposed generalization of "deep unrolling/unfolding" methodology [22]. Closer to PDE numerics, recently [11] proposed MGNet, being neural network emulation of multilevel, iterative solver for linear, elliptic PDEs.

The general idea behind these approaches is to emulate by a DNN the contractive map, say Φ, which is assumed to satisfy the conditions of Banach's Fixed Point Theorem (BFPT). Let us denote the approximate map realized by the DNN $\tilde{\Phi}$. It follows from the universality theorem for

*ETH Zürich, Seminar for Applied Mathematics
†andreas.stein@sam.math.ethz.ch

DNNs in various function classes (see, e.g., [16, 25] and the references there) that for any $\varepsilon > 0$ a DNN surrogate $\tilde{\Phi}$ to the contraction map exists, which is ε-close to Φ, uniformly on the domain of attraction of Φ.

Iteration of the DNN $\tilde{\Phi}$ being realized by composition, *any finite number K of steps of the fixed point iteration can be realized by K-fold composition of the DNN surrogate $\tilde{\Phi}$*. Iterating $\tilde{\Phi}$, instead of Φ, induces an error of order $\mathcal{O}(\varepsilon/(1-L))$, *uniformly* in the number of iterations K, where $L \in (0,1)$ denotes the contraction constant of Φ. Due to the contraction property of Φ, K may be chosen as $O(|\log(\varepsilon)|)$ in order to output an approximate fixed point with accuracy ε upon termination. The K-fold composition of the surrogate DNN $\tilde{\Phi}$ is, in turn, itself a DNN of depth $O(\mathrm{depth}(\tilde{\Phi})|\log(\varepsilon)|)$. This reasoning is valid also in metric spaces, since the notions of continuity and contractivity of the map Φ do not rely on availability of a norm. Hence, a (sufficiently large) DNN $\tilde{\Phi}$ exists which may be used likewise for the iterative solution of VIs in metric spaces. Furthermore, the *resulting fixed-point-iteration Nets obtained in this manner naturally exhibit a recurrent structure*, in the case (considered here) that the surrogate $\tilde{\Phi}$ is fixed throughout the K-fold composition (more refined constructions with stage-dependent approximations $\{\tilde{\Phi}^{(k)}\}_{k=1}^{K}$ of increasing emulation accuracy could be considered, but shall not be addressed here).

In summary, with the geometric error reduction of FPIs which is implied by the contraction condition, finite truncation at a prescribed emulation precision $\varepsilon > 0$ will imply $O(|\log(\varepsilon)|)$ iterations, and exact solution representation (of the fixed point of $\tilde{\Phi}$) in the infinite depth limit. In DNN calculus, finitely terminated FPIs can be realized via finite concatenation of the DNN approximation $\tilde{\Phi}$ of the contraction map Φ. The corresponding DNNs exhibit logarithmic in $|\varepsilon|$ depth, and naturally a recurrent structure due to the repetition of the Net $\tilde{\Phi}$ their construction. Thereby, recurrent DNNs can be built which encode numerical solution maps of fixed point iterations. This idea has appeared in various incarnations in recent work; we refer to, e.g., MGNet for the realization of Multi-grid iterative solvers of discretized elliptic PDEs [11]. The presently proposed ProxNet and PDASNet architectures are, in fact, DNN emulations of corresponding fixed point iterations of (discretized) variational inequalities. To analyze expression rates of deep neural networks (DNNs) for *emulating solution operators* to VIs is the purpose of the present paper. In line with recent work (e.g. [19, 21] and the references there), we take the perspective of *infinite-dimensional* VIs, which are set on closed cones in separable Hilbert spaces. The task at hand is then the analysis of *rates of expression of the approximate data-to-solution map*, which relates the input data (i.e. operator, cone, etc.) to the unique solution of the VI.

1.1 Layout

The structure of this paper is as follows. In Section 2, we recapitulate basic notions and definitions of proximal neural networks in infinite-dimensional, separable Hilbert spaces. A particular role is taken by so-called proximal activations, and a calculus of ProxNets, which we shall use throughout the rest of the paper to build solution operators of VIs. Section 3 addresses the conceptual use of ProxNets in the constructive solution of VIs. We build in particular ProxNet emulators of convergent fixed point iterations to construct solutions of VIs. Section 3.2 introduces quantitative bounds for perturbations of ProxNets. Section 4 emphasizes that ProxNets may be regarded as (approximate) solution operators to unilateral obstacle problems in infinite-dimensional Hilbert spaces. Section 5 presents DNN emulations of iterative solvers of matrix LCPs which arise from discretization of unilateral problems for PDEs. Section 6.1 introduces PDASNet, which emulate a class of primal-dual active set strategies for the numerical solution of VIs. Section 7 presents several numerical experiments, which illustrate the foregoing developments. More precisely, we consider the numerical solution of free boundary value problems arising in the valuation of American-style options. Section 8 provides a brief summary of the main results and indicates possible directions for further research.

1.2 Notation

We use standard notation. By $\mathcal{L}(\mathcal{H}, \mathcal{K})$ we denote the Banach space of bounded, linear operators from the Banach space $\mathcal{H}$ into $\mathcal{K}$ (surjectivity will not be required). Unless explicitly stated otherwise, all Hilbert and Banach-spaces are infinite-dimensional. By bold symbols, we denote matrices resp. linear maps between finite-dimensional spaces. Vectors in finite-dimensional, euclidean space are always understood as column vectors, with $\top$ denoting transposition of matrices and vectors.
Acknowledgement: The preparation of this work benefited from the participation of ChS in the thematic period "Mathematics of Deep Learning (MDL)" from 1 July to 17 December 2021, at the Isaac Newton Institute, Cambridge, UK.

2 Proximal Neural Networks – ProxNets

We consider the following *model for an artificial neural network*: For finite $m \in \mathbb{N}$, let $\mathcal{H}$ and $(\mathcal{H}_i)_{0 \leq i \leq m}$ be real, separable Hilbert spaces. For every $i \in \{1, \dots, m\}$ let $W_i \in \mathcal{L}(\mathcal{H}_{i-1}, \mathcal{H}_i)$ be a bounded linear operator, let $b_i \in \mathcal{H}_i$, let $R_i : \mathcal{H}_i \to \mathcal{H}_i$ be a nonlinear, continuous operator, and define

$$T_i : \mathcal{H}_{i-1} \to \mathcal{H}_i, \quad x \mapsto R_i(W_i x + b_i). \tag{1}$$

Moreover, let $W_0 \in \mathcal{L}(\mathcal{H}_0, \mathcal{H})$, $W_{m+1} \in \mathcal{L}(\mathcal{H}_m, \mathcal{H})$, $b_{m+1} \in \mathcal{H}$ and consider the neural network (NN) model

$$\Psi : \mathcal{H}_0 \to \mathcal{H}, \quad x \mapsto W_0 x + W_{m+1}(T_m \circ \cdots \circ T_1)(x) + b_{m+1}. \tag{2}$$

The operator $W_0 \in \mathcal{L}(\mathcal{H}_0, \mathcal{H})$ allows to include skip connections in the model, similar to *deep residual neural networks* as proposed in [12, 13]. This article focuses in particular on NNs with identical input and output spaces as in [5, Model 1.1], that arise as special case of model (2) with $\mathcal{H}_0 = \mathcal{H}_m = \mathcal{H}$ and are of the form

$$\Phi : \mathcal{H} \to \mathcal{H}, \quad x \mapsto (1 - \lambda)x + \lambda(T_m \circ \cdots \circ T_1)(x), \tag{3}$$

for a relaxation parameter $\lambda > 0$ to be adjusted for each application. The relation $\mathcal{H}_0 = \mathcal{H}_m = \mathcal{H}$ allows us to investigate fixed points of $\Phi : \mathcal{H} \to \mathcal{H}$, which are in turn solutions to variational inequalities.

The nonlinear operators R_i act as *activation operators* of the NNs and are subsequently given by suitable *proximity operators* on $\mathcal{H}_i$. We refer to Ψ and Φ as *proximal neural networks or ProxNets for short*, and derive sufficient conditions on the operators T_i, resp. W_i and R_i, so that Φ defines a contraction on $\mathcal{H}$. Hence, the unique fixed point $x^* = \Phi(x^*) \in \mathcal{H}$ solves a variational inequality, that is turn uniquely determined by the network parameters W_i, b_i and R_i for $i \in \{1, \dots, m\}$. On the other hand, any well-posed variational inequality on $\mathcal{H}$ may be recast as fixed-point problem for a suitable contractive ProxNet $\Phi : \mathcal{H} \to \mathcal{H}$. This enables us to approximate solutions to variational inequality problems as fixed-point iterations of ProxNets and derive convergence rates. Due to the contraction property of Φ, the fixed-point iteration $x_n = \Phi(x_{n-1}), n \in \mathbb{N}$ converges to $x^* = \Phi(x^*)$ for any $x_0 \in \mathcal{H}$ at linear rate. Moreover, the iteration is stable under small perturbations of the network parameters. As we show in Subsection 5.3 below, the latter property allows us to solve *entire classes* of variational inequality problems using only *one* ProxNet with fixed parameters.

2.1 Proximal Activations

Definition 2.1. Let $i \in \{0, \dots, m\}$ be a fixed index, $\psi_i : \mathcal{H}_i \to \mathbb{R} \cup \{\infty\}$ and $\mathrm{dom}(\psi_i) := \{x \in \mathcal{H}_i | \psi_i(x) < \infty\}$. We denote by $\Gamma_0(\mathcal{H}_i)$ the set of all proper, lower semi-continuous functions on $\mathcal{H}_i$, that is

$$\Gamma_0(\mathcal{H}_i) := \left\{ \psi_i : \mathcal{H}_i \to \mathbb{R} \cup \{\infty\} \middle| \ \liminf_{y \to x} \psi_i(y) \geq \psi_i(x) \text{ for all } x \in \mathcal{H}_i \text{ and } \mathrm{dom}(\psi_i) \neq \emptyset. \right\}$$

For any $\psi_i \in \Gamma_0(\mathcal{H}_i)$, the *subdifferential* of ψ_i at $x \in \mathcal{H}_i$ is

$$\partial\psi_i(x) := \{v \in \mathcal{H}_i | \, (y - x, v) + f(x) \leq f(y) \;\text{ for all } y \in \mathcal{H}_i\} \subset \mathcal{H}_i, \quad x \in \mathcal{H}_i,$$

and the *proximity operator* of ψ_i is

$$\text{prox}_{\psi_i} : \mathcal{H}_i \to \mathcal{H}_i, \quad x \mapsto \underset{y \in \mathcal{H}_i}{\text{argmin}} \, \psi_i(y) + \frac{\|x - y\|^2_{\mathcal{H}_i}}{2}. \tag{4}$$

It is well-known that prox_{ψ_i} is a firmly nonexpansive operator, i.e., $2\text{prox}_{\psi_i} - \text{id}$ is nonexpansive, see, e.g., [2, Proposition 12.28]. As outlined in [5, Section 2], there is a natural relation between proximity operators and activation functions in neural networks: Virtually any commonly used activation function such as rectified linear unit, tanh, softmax, etc. may be expressed as proximity operator on $\mathcal{H}_i = \mathbb{R}^d$, $d \in \mathbb{N}$, for an appropriate $\psi_i \in \Gamma_0(\mathcal{H}_i)$ (see [5, Section 2] for examples). We consider a set of particular proximity operators given by

$$\mathcal{A}(\mathcal{H}_i) := \{R_i = \text{prox}_{\psi_i} | \, \psi_i \in \Gamma_0(\mathcal{H}_i) \text{ such that } \psi_i \text{ is minimal at } 0 \in \mathcal{H}_i\}, \tag{5}$$

cf. [5, Definition 2.20]. Besides being continuous and nonexpansive, any $R_i \in \mathcal{A}(\mathcal{H}_i)$ satisfies $R_i(0) = 0$ ([5, Proposition 2.21]). Therefore, in the case $\mathcal{H}_i = \mathbb{R}$, the elements in $\mathcal{A}(\mathbb{R})$ are also referred to as *stable activation functions*, cf. [10, Lemma 5.1]. With this in mind, we formally define proximal neural networks, or ProxNets.

Definition 2.2. Let $\Psi : \mathcal{H}_0 \to \mathcal{H}$ be the m-layer neural network model in (2). If $R_i \in \mathcal{A}(\mathcal{H}_i)$ holds for any $i \in \{1, \ldots, m\}$, Ψ is called a *proximal neural network or ProxNet.*

2.2 ProxNet Calculus

Before investigating the relation of Φ in (3) to variational inequality models, we record several useful definitions and results for NN calculus in the more general model Ψ from Equation (2).

Definition 2.3. Let $j \in \{1, 2\}$, $m_j \in \mathbb{N}$, let $\mathcal{H}^{(j)}, \mathcal{H}^{(j)}_0, \ldots, \mathcal{H}^{(j)}_{m_j}$ be seperable Hilbert spaces such that $\mathcal{H}^{(2)} = \mathcal{H}^{(1)}_0$, and let Ψ_j be m_j-layer ProxNets as in (2) given by

$$\Psi_j : \mathcal{H}^{(j)}_0 \to \mathcal{H}^{(j)}, \quad x \mapsto W^{(j)}_{m_j+1} \left(T^{(j)}_{m_j} \circ \cdots \circ T^{(j)}_1\right)(x) + b^{(j)}_{m+1}.$$

The *concatenation* of Ψ_1 and Ψ_2 is defined by the map

$$\Psi_1 \bullet \Psi_2 : \mathcal{H}^{(2)}_0 \to \mathcal{H}^{(1)}, \quad x \mapsto (\Psi_1 \circ \Psi_2)(x). \tag{6}$$

Remark 2.4. Due to $W^{(j)}_0 \equiv 0$ there are no skip connections after the last proximal activation in Ψ_j, hence $\Psi_1 \bullet \Psi_2$ is in fact a ProxNet as in (2) with $2m$ layers and no skip connection.

Definition 2.5. Let $m \in \mathbb{N}$, $j \in \{1, 2\}$, let $\mathcal{H}^{(j)}, \mathcal{H}^{(j)}_0, \ldots, \mathcal{H}^{(j)}_{m_j}$ be seperable Hilbert spaces such that $\mathcal{H}^{(1)}_0 = \mathcal{H}^{(2)}_0$, and let Ψ_j be m-layer ProxNets as in (2) given by

$$\Psi_j : \mathcal{H}^{(j)}_0 \to \mathcal{H}^{(j)}, \quad x \mapsto W^{(j)}_0 x + W^{(j)}_{m+1} \left(T^{(j)}_{m_j} \circ \cdots \circ T^{(j)}_1\right)(x) + b^{(j)}_{m+1}.$$

The parallelization of Ψ_1 and Ψ_2 is given for $\mathcal{H}_0 := \mathcal{H}^{(1)}_0 = \mathcal{H}^{(2)}_0$ by

$$P(\Psi_1, \Psi_2) : \mathcal{H}_0 \to \mathcal{H}^{(1)} \oplus \mathcal{H}^{(2)}, \quad x \mapsto (\Psi_1(x), \Psi_2(x)).$$

Proposition 2.6. *The parallelization $P(\Psi_1, \Psi_2)$ of two ProxNets Ψ_1 and Ψ_2 as in Definition 2.5 is a ProxNet.*

Proof. We set $\mathcal{H}_{m+1}^{(j)} := \mathcal{H}^{(j)}$ for $j \in \{1,2\}$, fix $i \in \{1,\dots,m\}$ and observe that $\mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)}$ equipped with the scalar product $(\cdot,\cdot)_{\mathcal{H}_i^{(1)}\oplus\mathcal{H}_i^{(2)}} := (\cdot,\cdot)_{\mathcal{H}_i^{(1)}} + (\cdot,\cdot)_{\mathcal{H}_i^{(2)}}$ is again a separable Hilbert space. We define

$$
\begin{aligned}
&W_0 : \mathcal{H}_0 \mapsto \mathcal{H}^{(1)} \oplus \mathcal{H}^{(2)}, \quad x \mapsto (W_0^{(1)}x, W_0^{(2)}y),\\
&W_1 : \mathcal{H}_0 \mapsto \mathcal{H}_1^{(1)} \oplus \mathcal{H}_1^{(2)}, \quad x \mapsto (W_1^{(1)}x, W_1^{(2)}y),\\
&W_i : \mathcal{H}_{i-1}^{(1)} \oplus \mathcal{H}_{i-1}^{(2)} \mapsto \mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)}, \quad (x,y) \mapsto (W_i^{(1)}x, W_i^{(2)}y), \quad i \in \{2,\dots,m+1\},\\
&b_i := (b_i^{(1)}, b_i^{(2)}) \in \mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)}, \quad i \in \{1,\dots,m+1\},\\
&R_i : \mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)} \mapsto \mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)}, \quad (x,y) \mapsto (R_i^{(1)}x, R_i^{(2)}y), \quad i \in \{0,1,\dots,m\}.
\end{aligned}
$$

Note that all W_i are bounded, linear operators. Moreover, if $R_i^{(j)} = \operatorname{prox}_{\psi_i^{(j)}} \in \mathcal{A}(\mathcal{H}_i^{(j)})$ holds for $\psi_i^{(j)} \in \Gamma_0(\mathcal{H}_i^{(j)})$ and $j \in \{1,2\}$, then $R_i = \operatorname{prox}_{\psi_i}$, where $\psi_i \in \Gamma_0(\mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)})$ is defined by $\psi_i(x,y) := \psi_i^{(1)}(x) + \psi_i^{(2)}(y)$. Hence, $R_i \in \mathcal{A}(\mathcal{H}_i^{(1)} \oplus \mathcal{H}_i^{(2)})$ and it holds that

$$
P(\Psi_1,\Psi_2) : \mathcal{H}_0 \to \mathcal{H}^{(1)} \oplus \mathcal{H}^{(2)}, \quad x \mapsto W_0 x + W_{m+1}(T_m \circ \cdots \circ T_1)(x) + b_{m+1},
$$

with $T_i := R_i(W_i \cdot + b_i)$ for $i \in \{1,\dots,m\}$, which shows the claim. □

3 ProxNets and Variational Inequalities

3.1 Contractive ProxNets

We formulate sufficient conditions on the neural network model in (3) so that $\Phi : \mathcal{H} \to \mathcal{H}$ is a contraction. The associated fixed-point iteration converges to the unique solution of a variational inequality, which is characterized in the following.

Assumption 3.1. Let Φ be a ProxNet as in (3) with $m \in \mathbb{N}$ layers such that $W_i \in \mathcal{L}(\mathcal{H}_{i-1}, \mathcal{H}_i)$, $b_i \in \mathcal{H}_i$, and $R_i \in \mathcal{A}(\mathcal{H}_i)$ for all $i \in \{1,\dots,m\}$. It holds that $\lambda \in (0,2)$ and the operators W_i satisfy

$$
L_\Phi := \prod_{i=1}^{m} \|W_i\|_{\mathcal{L}(\mathcal{H}_{i-1},\mathcal{H}_i)} < \min(1, 2/\lambda - 1).
$$

Theorem 3.2. *Let Φ be as in (3), let $x^0 \in \mathcal{H}$ and define the iteration $x^{k+1} := \Phi(x^k)$, $k \in \mathbb{N}_0$. Under Assumption 3.1, the sequence $(x^k, k \in \mathbb{N}_0)$ converges for any $x^0 \in \mathcal{H}$ to the unique fixed-point $x^* \in \mathcal{H}$. For any finite number $k \in \mathbb{N}$ the error is bounded by*

$$
\|x^* - x^k\|_{\mathcal{H}} \le \frac{\|\Phi(x^0) - x^0\|}{1 - L_{\Phi,\lambda}} L_{\Phi,\lambda}^k, \quad L_{\Phi,\lambda} := |1-\lambda| + \lambda L_\Phi \in [0,1). \tag{7}
$$

It holds that

$$
(x_1^*,\dots,x_m^*) := (T_1 x^*, (T_2 \circ T_1)x^*, \dots, (T_{m-1} \circ \cdots \circ T_1)x^*, x^*) \in \mathcal{H}_1 \times \cdots \times \mathcal{H}_m
$$

is the unique solution to the variational inequality problem: find $x_1 \in \mathcal{H}_1, \dots, x_0 = x_m \in \mathcal{H}_m$, such that

$$
W_i x_{i-1} + b_i - x_i \in \partial\psi_i(x_i), \quad i \in \{1,\dots,m\}. \tag{8}
$$

Moreover, x^* is bounded by

$$
\|x^*\|_{\mathcal{H}} \le C^* \sum_{i=1}^{m} \left(\prod_{j=i+1}^{m} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1},\mathcal{H}_j)} \right) \|b_i\|_{\mathcal{H}_i}, \quad C^* := \begin{cases} \frac{1}{1-L_\Phi} & < \infty, \quad \lambda \in (0,1] \\ \frac{\lambda}{2-\lambda(1+L_\Phi)} & < \infty, \quad \lambda \in (1,2) \end{cases}.
$$

Proof. By the non-expansiveness of $R_i : \mathcal{H}_i \to \mathcal{H}_i$ for $i \in \{1, \dots, m\}$ it follows for any $x, y \in \mathcal{H}$ that

$$\begin{aligned}
\|\Phi(x) - \Phi(y)\|_{\mathcal{H}} &\leq |1-\lambda|\|x-y\|_{\mathcal{H}} + \lambda\|(T_m \circ \cdots \circ T_1)x - (T_m \circ \cdots \circ T_1)y\|_{\mathcal{H}_m} \\
&\leq |1-\lambda|\|x-y\|_{\mathcal{H}} \\
&\quad + \lambda\|(W_m \circ (T_{m-1} \circ \cdots \circ T_1))x - (W_m \circ (T_{m-1} \circ \cdots \circ T_1))y\|_{\mathcal{H}_m} \\
&\leq |1-\lambda|\|x-y\|_{\mathcal{H}} \\
&\quad + \lambda\|W_m\|_{\mathcal{L}(\mathcal{H}_{m-1},\mathcal{H}_m)}\|(T_{m-1} \circ \cdots \circ T_1)x - (T_{m-1} \circ \cdots \circ T_1)y\|_{\mathcal{H}_{m-1}} \\
&\leq |1-\lambda|\|x-y\|_{\mathcal{H}} + \lambda\left(\prod_{i=1}^{m}\|W_i\|_{\mathcal{L}(\mathcal{H}_{i-1},\mathcal{H}_i)}\right)\|x-y\|_{\mathcal{H}_0} \\
&= \underbrace{(|1-\lambda| + \lambda L_\Phi)}_{:=L_{\Phi,\lambda}}\|x-y\|_{\mathcal{H}}.
\end{aligned}$$

As $\lambda \in (0,2)$ and $L_\Phi < \min(1, 2/\lambda - 1)$ by Assumption 3.1, it follows that $L_{\Phi,\lambda} < 1$, hence $\Phi : \mathcal{H} \to \mathcal{H}$ is a contraction. Existence and uniqueness of $x^* \in \mathcal{H}$ and the first part of the claim then follow by Banach's fixed-point theorem for any initial value $x^0 \in \mathcal{H}$.

By [2, Proposition 16.44], it holds for any $i \in \{1, \dots, m\}$, $x_i, y_i \in \mathcal{H}_i$ and $\psi_i \in \Gamma_0(\mathcal{H}_i)$ that

$$x_i = \operatorname{prox}_{\psi_i}(y_i) \quad \Leftrightarrow \quad y_i - x_i \in \partial\psi_i(x_i).$$

Now let $x_0^* := x^*$ and $x_i^* := (T_i \circ \cdots \circ T_1)(x^*)$ for $i \in \{1, \dots, m\}$. This yields $\Phi(x_0^*) = (1-\lambda)x^* + \lambda x_m^* = x^*$ and hence $x_m^* = x^*$. Recalling that $R_i = \operatorname{prox}_{\psi_i}$ with $\psi_i \in \Gamma_0(\mathcal{H}_i)$ for all $i \in \{1, \dots, m\}$, it hence follows that

$$W_i x_{i-1}^* + b_i - x_i^* \in \partial\psi_i(x_i^*),$$

cf. [5, Propostion 4.3]. Finally, to bound x^*, we use that

$$\|x^*\|_{\mathcal{H}} \leq \|\Phi(x^*) - \Phi(0)\|_{\mathcal{H}} + \|\Phi(0)\|_{\mathcal{H}} \leq L_{\Phi,\lambda}\|x^*\|_{\mathcal{H}} + \lambda\|(T_m \circ \cdots \circ T_1)(0)\|_{\mathcal{H}_m}.$$

As $R_i \in \mathcal{A}(\mathcal{H}_i)$, it holds $R_i(0) = 0$ and therefore $\|R_i(x)\|_{\mathcal{H}_i} \leq \|x\|_{\mathcal{H}_i}$ for all $x \in \mathcal{H}_i$, which in turn shows

$$\begin{aligned}
\|(T_m \circ \cdots \circ T_1)(0)\|_{\mathcal{H}_m} &\leq \|W_m\|_{\mathcal{L}(\mathcal{H}_{m_1},\mathcal{H}_m)}\|(T_{m-1} \circ \cdots \circ T_1)(0)\|_{\mathcal{H}_{m-1}} + \|b_m\|_{\mathcal{H}_m} \\
&\leq \|W_m\|_{\mathcal{L}(\mathcal{H}_{m_1},\mathcal{H}_m)} \\
&\quad \cdot \left(\|W_{m-1}\|_{\mathcal{L}(\mathcal{H}_{m-2},\mathcal{H}_{m-1})}\|(T_{m-2} \circ \cdots \circ T_1)(0)\|_{\mathcal{H}_{m-2}} + \|b_{m-1}\|_{\mathcal{H}_{m-1}}\right) \\
&\quad + \|b_m\|_{\mathcal{H}_m} \\
&\leq \sum_{i=1}^{m}\left(\prod_{j=i+1}^{m}\|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1},\mathcal{H}_j)}\right)\|b_i\|_{\mathcal{H}_i}.
\end{aligned}$$

The claim follows with $L_\Phi < \min(1, 2/\lambda - 1)$, since

$$1 - L_{\Phi,\lambda} = \begin{cases} \lambda(1 - L_\Phi) & > 0, \quad \lambda \in (0,1] \\ 2 - \lambda(1 + L_\Phi) & > 0, \quad \lambda \in (1,2) \end{cases}.$$

□

3.2 Perturbation Estimates for ProxNets

We introduce a perturbed version of the ProxNet Φ in (3) in this subsection. Besides changing the network parameters W_i, b_i and R_i, we also augment the input space $\mathcal{H}$ and allow an architecture that approximates each nonlinear operator T_i itself by a multilayer network. These changes

allow us to consider ProxNet as an approximate data-to-solution operator for infinite-dimensional variational inequalities and to control perturbations of the network parameters. For instance, we show in Example 3.4 that augmented ProxNets mimic the solution operator to Problem (8), that maps the bias vectors $b_1, \dots, b_m$ to the solution $x_1, \dots, x_m$.

Let $\widetilde{\mathcal{H}}_0, \dots, \widetilde{\mathcal{H}}_{m-1}$ be arbitrary separable Hilbert spaces and let $\widetilde{\mathcal{H}} := \widetilde{\mathcal{H}}_0$. Then, for $i \in \{0, \dots, m-1\}$ the direct sum $\mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i$ equipped with the inner product $(\cdot,\cdot)_{\mathcal{H}_i} + (\cdot,\cdot)_{\widetilde{\mathcal{H}}_i}$ is again a separable Hilbert space. For notational convenience, we set $\widetilde{\mathcal{H}}_m := \{0 \in \mathcal{H}_m\}$ and use the identification $\mathcal{H}_m \oplus \widetilde{\mathcal{H}}_m = \mathcal{H}_m = \mathcal{H}$. We consider the ProxNet

$$\widetilde{\Phi} : \mathcal{H} \oplus \widetilde{\mathcal{H}} \to \mathcal{H}, \quad (x, \widetilde{x}) \mapsto (1-\lambda)x + \lambda(\widetilde{T}_m \circ \cdots \circ \widetilde{T}_1)(x, \widetilde{x}), \tag{9}$$

where we allow that the operators $\widetilde{T}_i$ are itself multi-layer ProxNets: For any $i \in \{1, \dots, m\}$ let $m_i \in \mathbb{N}$ and let $\mathcal{H}_0^{(i)} := \mathcal{H}_{i-1} \oplus \widetilde{\mathcal{H}}_{i-1}$, $\mathcal{H}_1^{(i)}, \dots, \mathcal{H}_{m_i-1}^{(i)}, \mathcal{H}_{m_i}^{(i)} := \mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i$ be separable Hilbert spaces. For $j_i \in \{1, \dots, m_i\}$ consider the operators $\widetilde{T}_{j_i}^{(i)}(\cdot) = R_{j_i}^{(i)}(W_{j_i}^{(i)} \cdot + b_{j_i}^{(i)})$ given by

$$R_{j_i}^{(i)} \in \mathcal{A}(\mathcal{H}_{j_i}^{(i)}), \quad W_{j_i}^{(i)} \in \mathcal{L}(\mathcal{H}_{j_i-1}^{(i)}, \mathcal{H}_{j_i}^{(i)}), \quad b_{j_i}^{(i)} \in \mathcal{H}_{j_i}^{(i)}.$$

We then define $\widetilde{T}_i$ as

$$\widetilde{T}_i : \mathcal{H}_{i-1} \oplus \widetilde{\mathcal{H}}_{i-1} \to \mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i, \quad (x_{i-1}, \widetilde{x}_{i-1}) \mapsto (\widetilde{T}_{m_i}^{(i)} \circ \cdots \circ \widetilde{T}_1^{(i)})(x_{i-1}, \widetilde{x}_{i-1}),$$

which in turn determines $\widetilde{\Phi}$ in (9). By construction, $\widetilde{\Phi}$ is a ProxNet of the form (2) with $\sum_{i=1}^m m_i \geq m$ layers. As compared to Φ, we augmented the input and intermediate spaces by $\widetilde{\mathcal{H}}_i$. The composite structure of the maps $\widetilde{T}_i$ allows to choose input vectors $\widetilde{x}_{i-1} \in \widetilde{\mathcal{H}}_{i-1}$ such that the first component of $\widetilde{T}_i(x_{i-1}, \widetilde{x}_{i-1})$ approximates $T_i(x_{i-1})$ uniformly on a subset of $\mathcal{H}_{i-1}$. As we show in Subsection 5.3 below, this enables us to solve large classes of variational inequalities with *only one* fixed ProxNet $\widetilde{\Phi}$, that in turn approximates a *data-to-solution operator*, instead of employing different fixed maps $\Phi : \mathcal{H} \to \mathcal{H}$ for every problem.

To formulate reasonable assumptions on $\widetilde{\Phi}$ we denote for any $i \in \{1, \dots, m-1\}$ by

$$P_{\mathcal{H}_i} : \mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i \mapsto \mathcal{H}_i, \quad (x_i, \widetilde{x}_i) \mapsto x_i,$$
$$P_{\widetilde{\mathcal{H}}_i} : \mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i \mapsto \widetilde{\mathcal{H}}_i, \quad (x_i, \widetilde{x}_i) \mapsto \widetilde{x}_i$$

the projections to the first and second component for an element in $\mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i$, respectively. Moreover, we define the closed ball $B_r^{(i)} := \{x_i \in \mathcal{H}_i \,|\, \|x_i\|_{\mathcal{H}_i} \leq r\} \subset \mathcal{H}_i$ with radius $r > 0$.

Assumption 3.3. Let Φ and $\widetilde{\Phi}$ be proximal neural networks defined as in Equations (3) and (9), respectively. There are constants $\widetilde{L} \in (0,1)$, $\delta \geq 0$ and $\Theta_1 \geq \Theta_0 \geq \Theta_2 > 0$ such that

1. Φ satisfies Assumption 3.1 with $\lambda \in (0,1]$ and $L_\Phi \leq \widetilde{L} \in (0,1)$.
2. It holds that

$$\left(\max_{i \in \{0,1,\dots,m\}} \prod_{j=1}^{i} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right) \Theta_0 + \sum_{i=1}^{m} \left(\prod_{j=i+1}^{m} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right) (\|b_i\|_{\mathcal{H}_m} + \delta) \leq \Theta_1,$$

$$\sum_{i=1}^{m} \left(\prod_{j=i+1}^{m} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right) \|b_i\|_{\mathcal{H}_i} \leq (1-\widetilde{L})\Theta_2,$$

as well as

$$\Theta_2 + \frac{\delta}{(1-\widetilde{L})} \sum_{i=1}^{m} \left(\prod_{j=i+1}^{m} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right) \leq \Theta_0.$$

3. There is a vector $\widetilde{x}_0 \in \widetilde{\mathcal{H}}_0$, such that for $i \in \{1, \dots, m\}$, any $x_{i-1} \in B_{\Theta_1}^{(i-1)} \subset \mathcal{H}_{i-1}$ and $\widetilde{x}_i := P_{\widetilde{\mathcal{H}}_i} \widetilde{T}_i(x_{i-1}, \widetilde{x}_{i-1})$ it holds

$$\|T_i(x_{i-1}) - P_{\mathcal{H}_i} \widetilde{T}_i(x_{i-1}, \widetilde{x}_{i-1})\|_{\mathcal{H}_i} \leq \delta.$$

Before we derive error bounds, we provide an example to motivate the construction of $\widetilde{\Phi}$ and Assumption 3.3.

Example 3.4 (Bias-to-solution operator)**.** Let Φ be as in Assumption 3.1 with $m = 2$ layers and network parameters R_i, W_i, b_i for $i \in \{1, 2\}$. We construct a ProxNet $\widetilde{\Phi}$ that takes the bias vectors b_1, b_2 of Φ as inputs to represent Φ for any choice of $b_i \in \mathcal{H}_i$, and therefore may be concatenated to map any choice of b_1, b_2 to the respective solution (x_1, x_2) of (8). In other words, we approximate the *bias-to-solution operator*

$$O_{b_1,b_2} : \mathcal{H}_1 \oplus \mathcal{H}_2 \mapsto \mathcal{H}_1 \oplus \mathcal{H}_2, \ (b_1, b_2) \mapsto (x_1, x_2).$$

To this end, we set $\widetilde{\mathcal{H}}_0 = \mathcal{H}_1 \oplus \mathcal{H}_2$, $\widetilde{\mathcal{H}}_1 = \mathcal{H}_2$, $n_1 = n_2 = 1$, $b_{i,1} = 0 \in \mathcal{H}_i \oplus \widetilde{\mathcal{H}}_i$ and

$$\begin{aligned}
W_1^{(1)} &: \mathcal{H} \oplus \mathcal{H}_1 \oplus \mathcal{H}_2 \to \mathcal{H}_1 \oplus \mathcal{H}_2, && (x, x_1, x_2) \mapsto (W_1 x + x_1, x_2) \\
W_1^{(2)} &: \mathcal{H}_1 \oplus \mathcal{H}_2 \to \mathcal{H}_2, && (x_1, x_2) \mapsto W_2 x_1 + x_2, \\
R_1^{(1)} &: \mathcal{H}_1 \oplus \mathcal{H}_2 \to \mathcal{H}_1 \oplus \mathcal{H}_2, && (x_1, x_2) \mapsto R_1(x_1) + x_2, \\
R_1^{(2)} &: \mathcal{H}_2 \to \mathcal{H}_2, && x_2 \mapsto R_2(x_2).
\end{aligned}$$

Note that $R_1^{(1)} = \operatorname{prox}_{\psi_1^{(1)}}$ with $\psi_1^{(1)}(x_1, x_2) := \psi_1(x_1)$ for any $(x_1, x_2) \in \mathcal{H}_1 \oplus \mathcal{H}_2$, where ψ_1 determines $R_1 = \operatorname{prox}_{\psi_1}$. Hence, $R_1^{(1)} \in \mathcal{A}(\mathcal{H}_1 \oplus \widetilde{\mathcal{H}}_1)$, and it follows with $\widetilde{x}_0 := (b_1, b_2) \in \mathcal{H}_1 \oplus \mathcal{H}_2$ for any $x \in \mathcal{H}$ and $x_1 \in \mathcal{H}_1$ that

$$\begin{aligned}
T_1(x) &= R_1(W_1 x + b_1) = P_{\mathcal{H}_1}(R_1(W_1 x + b_1), b_2) = P_{\mathcal{H}_1} R_1^{(1)}(W_1^{(1)}(x, \widetilde{x}_0)) = P_{\mathcal{H}_1} \widetilde{T}_1(x, \widetilde{x}_0) \\
T_2(x) &= R_2(W_2 x_1 + b_2) = R_1^{(2)}(W_1^{(2)}(x_1, b_2)) = P_{\mathcal{H}_2} R_1^{(2)}(W_1^{(2)}(x_1, P_{\widetilde{\mathcal{H}}_1} \widetilde{T}_1(x_1, \widetilde{x}_0))).
\end{aligned}$$

Therefore, the last part of Assumption 3.3 holds with $\delta = 0$ for arbitrary large $\Theta_1 > 0$ and hence the constants $\Theta_0, \Theta_1, \Theta_2$ do not play any role in this example. The generalization to $m > 2$ layers follows by a similar construction of Φ.

Now let (x_1, x_2) be the solution to (8) for any choice $b_1 \in \mathcal{H}_1, b_2 \in \mathcal{H}_2$. It follows from Theorem 3.2 that the operator

$$\widetilde{O}_{b_1,b_2} : \mathcal{H} \oplus \mathcal{H}_1 \oplus \mathcal{H}_2 \to \mathcal{H}, \quad (x, b_1, b_2) \mapsto \underbrace{\widetilde{\Phi}(\cdot, b_1, b_2) \bullet \cdots \bullet \widetilde{\Phi}(\cdot, b_1, b_2)}_{k \text{ times}}(x)$$

satisfies $x_2 \approx \widetilde{O}_{b_1,b_2}(x^0, b_1, b_2)$ and $x_1 \approx T_1(\widetilde{O}_{b_1,b_2}(x^0, b_1, b_2))$ for any choice of $(x^0, b_1, b_2) \in \mathcal{H} \oplus \mathcal{H}_1 \oplus \mathcal{H}_2$, for a sufficiently large number k of concatenations of $\widetilde{\Phi}(\cdot, b_1, b_2)$.

The augmented ProxNet $\widetilde{\Phi}$ may also be utilized to consider families of obstacle problems, as shown in Example 4.4 below. Therein, the parametrization is with respect to the proximity operators R_i instead of the bias vectors b_i, and we construct an approximate *obstacle-to-solution operator* in the fashion of Example 3.4. In the finite-dimensional case (where the linear operators W_i correspond to matrices) the input of $\widetilde{\Phi}$ may even be augmented by a suitable space of operators, see Subsection 5.3 below for a detailed discussion. We conclude this section with a perturbation estimate that allows us to approximate the fixed-point of Φ by the augmented NN $\widetilde{\Phi}$.

Theorem 3.5. *Let Φ and $\widetilde{\Phi}$ be proximal neural networks as in Equations* (3) *and* (9) *that satisfy Assumption 3.3, and denote by $x^* \in \mathcal{H}$ the unique fixed-point of Φ from Theorem 3.2. Let $x^0 \in B_{\Theta_2}^{(0)}$ be arbitrary, let $\widetilde{x}_0$ be as in Assumption 3.3 and define the sequence $\widetilde{x}^{k+1} := \widetilde{\Phi}(\widetilde{x}^k, \widetilde{x}_0)$ for $k \in \mathbb{N}_0$, where $\widetilde{x}^0 := x^0$. Then there is a constant $C > 0$ which is independent of $\delta > 0$ and $\widetilde{x}_0$, such that for any $k \in \mathbb{N}$ it holds*

$$\|x^* - \widetilde{x}^k\|_{\mathcal{H}} \leq C\left(\widetilde{L}_\lambda^k + \delta\right),$$

where $\widetilde{L}_\lambda := (1-\lambda) + \lambda\widetilde{L} < 1$.

Proof. Let $x \in B_{\Theta_0}^{(0)}$ and let $\widetilde{x}_0 \in \widetilde{\mathcal{H}}_0$ be as in Assumption 3.3. We define $v_0 = x$, $v_i := P_{\mathcal{H}_i}(\widetilde{T}_i \circ \cdots \circ \widetilde{T}_1)(x, \widetilde{x}_0) \in \mathcal{H}_i$ for $i \in \{1, \ldots, m-1\}$, and $v_m := (\widetilde{T}_m \circ \cdots \circ \widetilde{T}_1)(x, \widetilde{x}_0) \in \mathcal{H}$. With $\widetilde{x}_i := P_{\widetilde{\mathcal{H}}_i}\widetilde{T}_i(x_{i-1}, \widetilde{x}_{i-1})$ and the convention that $P_{\mathcal{H}_m} = \mathrm{id}$, we obtain the recursion formula

$$v_i = P_{\mathcal{H}_i}\widetilde{T}_i(v_{i-1}, \widetilde{x}_{i-1}), \quad i \in \{1, \ldots, m\}. \tag{10}$$

We now show by induction that $\|v_i\|_{\mathcal{H}_i} \leq \Theta_1$ for $i \in \{0, \ldots, m\}$. By Assumption 3.3 it holds

$$\begin{aligned}
\|v_0\|_{\mathcal{H}_0} &= \|x\|_{\mathcal{H}} \\
&\leq \Theta_0 = \left(\prod_{j=1}^{0} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right)\Theta_0 + \sum_{j=1}^{0}\left(\prod_{\ell=j+1}^{0}\|W_\ell\|_{\mathcal{L}(\mathcal{H}_{\ell-1}, \mathcal{H}_\ell)}\right)(\|b_j\|_{\mathcal{H}_j} + \delta) \\
&\leq \Theta_1.
\end{aligned}$$

Now let

$$\|v_i\|_{\mathcal{H}_i} \leq \left(\prod_{j=1}^{i} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right)\Theta_0 + \sum_{j=1}^{i}\left(\prod_{\ell=j+1}^{i}\|W_\ell\|_{\mathcal{L}(\mathcal{H}_{\ell-1}, \mathcal{H}_\ell)}\right)(\|b_j\|_{\mathcal{H}_j} + \delta)$$

hold for a fixed $i \in \{0, \ldots, m-1\}$. Assumption 3.3 yields with Equation (10)

$$\|T_{i+1}(v_i) - v_{i+1}\|_{\mathcal{H}_{i+1}} = \|T_{i+1}(v_i) - P_{\mathcal{H}_{i+1}}\widetilde{T}_{i+1}(v_i, \widetilde{x}_0)\|_{\mathcal{H}_{i+1}} \leq \delta.$$

Using $\|R_{i+1}(x)\|_{\mathcal{H}_{i+1}} \leq \|x\|_{\mathcal{H}_{i+1}}$ for $x \in \mathcal{H}_{i+1}$ then yields together with the triangle inequality and the induction hypothesis

$$\begin{aligned}
\|v_{i+1}\|_{\mathcal{H}_{i+1}} &\leq \delta + \|T_{i+1}(v_i)\|_{\mathcal{H}_{i+1}} \\
&\leq \delta + \|W_{i+1}\|_{\mathcal{L}(\mathcal{H}_i, \mathcal{H}_{i+1})}\|v_i\|_{\mathcal{H}_i} + \|b_{i+1}\|_{\mathcal{H}_{i+1}} \\
&\leq \left(\prod_{j=1}^{i+1} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right)\Theta_0 + \sum_{l=1}^{i+1}\left(\prod_{j=l+1}^{i+1}\|W_\ell\|_{\mathcal{L}(\mathcal{H}_{\ell-1}, \mathcal{H}_\ell)}\right)(\|b_j\|_{\mathcal{H}_j} + \delta) \\
&\leq \Theta_1,
\end{aligned}$$

and hence $v_i \in B_{\Theta_1}^{(i)}$ for all $i \in \{0, \ldots, m\}$. With Assumption 3.3 and Equation (10) we further obtain for each $x \in B_{\Theta_0}^{(0)}$

$$\begin{aligned}
\frac{1}{\lambda}\|\Phi(x) - \widetilde{\Phi}(x, \widetilde{x}_0)\|_{\mathcal{H}_m} &= \|(T_m \circ \cdots \circ T_1)(x) - v_m\|_{\mathcal{H}} \\
&\leq \|(T_m \circ \cdots \circ T_1)(x) - T_m(v_{m-1})\|_{\mathcal{H}} + \|T_m(v_{m-1}) - \widetilde{T}_m(v_{m-1}, \widetilde{x}_m)\|_{\mathcal{H}} \\
&\leq \|W_m\|_{\mathcal{L}(\mathcal{H}_{m-1}, \mathcal{H}_m)}\|(T_{m-1} \circ \cdots \circ T_1)(x) - v_{m-1}\|_{\mathcal{H}_{m-1}} + \delta,
\end{aligned}$$

and by iterating this estimate over $i \in \{1, \ldots, m\}$

$$\|\Phi(x) - \widetilde{\Phi}(x, \widetilde{x}_0)\|_{\mathcal{H}_m} \leq \lambda\delta\sum_{i=1}^{m}\left(\prod_{j=i+1}^{m}\|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1}, \mathcal{H}_j)}\right) =: \lambda\delta C_\Phi. \tag{11}$$

Now let $x^* \in \mathcal{H}$ be the unique fixed-point of Φ as in Theorem 3.2, let $x^k = \Phi(x^{k-1})$ and $\widetilde{x}^k = \widetilde{\Phi}(\widetilde{x}^{k-1}, \widetilde{x}_0)$ for any $k \in \mathbb{N}$ and a given initial value $x^0 = \widetilde{x}^0 \in \mathcal{H}$ with $\|x^0\|_{\mathcal{H}} \leq \Theta_2$. We obtain as in the proof of Theorem 3.2

$$\begin{aligned}
\|x^1\|_{\mathcal{H}} &\leq \|\Phi(x^0) - \Phi(0)\|_{\mathcal{H}} + \|\Phi(0)\|_{\mathcal{H}} \\
&\leq L_{\Phi,\lambda}\|x^0\|_{\mathcal{H}} + \lambda \sum_{i=1}^{m} \left(\prod_{j=i+1}^{m} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1},\mathcal{H}_j)} \right) \|b_i\|_{\mathcal{H}_i} \\
&\leq (1-\lambda)\Theta_2 + \lambda \left(\widetilde{L}\Theta_2 + \sum_{i=1}^{m} \left(\prod_{j=i+1}^{m} \|W_j\|_{\mathcal{L}(\mathcal{H}_{j-1},\mathcal{H}_j)} \right) \|b_i\|_{\mathcal{H}_i} \right) \\
&\leq \Theta_2,
\end{aligned} \tag{12}$$

where we have used that $L_{\Phi,\lambda} = (1-\lambda) + \lambda L_\Phi \leq (1-\lambda) + \lambda\widetilde{L}$ and Assumption 3.3. Hence, we have $\|x^k\|_{\mathcal{H}} \leq \Theta_2$ inductively for all $k \in \mathbb{N}$. In the next step, we show that $\|\widetilde{x}^k\|_{\mathcal{H}} \leq \Theta_0$ by induction over k. First, we obtain with $\|x^0\| \leq \Theta_2 \leq \Theta_0$, (11) and (12) that

$$\|\widetilde{x}^1\|_{\mathcal{H}} = \|\widetilde{\Phi}(x^0, \widetilde{x}_0)\|_{\mathcal{H}} \leq \|\widetilde{\Phi}(x^0, \widetilde{x}_0) - \Phi(x^0)\|_{\mathcal{H}} + \|\Phi(x^0)\|_{\mathcal{H}} \leq \lambda\delta C_\Phi + \Theta_2.$$

Thus, $\|\widetilde{x}^1\|_{\mathcal{H}} \leq \Theta_0$ follows with Assumption 3.3 on the relation of Θ_0 and Θ_2 as $\lambda(1-\widetilde{L}) < 1$. Using the induction hypothesis $\|\widetilde{x}^k - x^k\|_{\mathcal{H}} \leq \lambda\delta C_\Phi \sum_{j=0}^{k-1} \widetilde{L}_{\Phi,\lambda}^j$ for a fixed $k \in \mathbb{N}$, $\|x^k\|_{\mathcal{H}} \leq \Theta_2$, and $L_{\Phi,\lambda} \leq \widetilde{L}_\lambda := (1-\lambda) + \lambda\widetilde{L} < 1$ yields similarly

$$\begin{aligned}
\|\widetilde{x}^{k+1}\|_{\mathcal{H}} &\leq \|\widetilde{\Phi}(\widetilde{x}^k, \widetilde{x}_0) - \Phi(\widetilde{x}^k)\|_{\mathcal{H}} + \|\Phi(\widetilde{x}^k) - \Phi(x^k)\|_{\mathcal{H}} + \|\Phi(x^k)\|_{\mathcal{H}} \\
&\leq \lambda\delta C_\Phi + L_{\Phi,\lambda}\|\widetilde{x}^k - x^k\|_{\mathcal{H}} + \Theta_2 \\
&\leq \lambda\delta C_\Phi \sum_{j=0}^{k} \widetilde{L}_\lambda^j + \Theta_2,
\end{aligned}$$

and hence $\|\widetilde{x}^k\|_{\mathcal{H}} \leq \lambda\delta C_\Phi/(\lambda(1-\widetilde{L})) + \Theta_2 \leq \Theta_0$ holds by induction for all $k \in \mathbb{N}$. We apply the bounds from Theorem 3.2 and (11) and conclude the proof by deriving

$$\begin{aligned}
\|x^* - \widetilde{x}^k\| &\leq \|x^* - x^k\| + \|\Phi(x^{k-1}) - \Phi(\widetilde{x}^{k-1})\| + \|\Phi(\widetilde{x}^{k-1}) - \widetilde{\Phi}(\widetilde{x}^{k-1}, \widetilde{x}_0)\| \\
&\leq \frac{\|x^1 - x^0\|}{1 - L_{\Phi,\lambda}} L_{\Phi,\lambda}^k + L_{\Phi,\lambda}\|x^{k-1} - \widetilde{x}^{k-1}\|_{\mathcal{H}} + \lambda\delta C_\Phi \\
&\leq \frac{\|\Phi(x^0) - x^0\|}{1 - \widetilde{L}_\lambda} \widetilde{L}_\lambda^k + \lambda\delta C_\Phi \sum_{j=0}^{k-1} \widetilde{L}_\lambda^j \\
&\leq \frac{\max(2\Theta_0, \lambda C_\Phi)}{1 - \widetilde{L}_\lambda} \left(\widetilde{L}_\lambda^k + \delta \right).
\end{aligned}$$

□

4 Variational Inequalities in Hilbert Spaces

In the previous sections we have considered a ProxNet model and derived the associated variational inequalities. Now we use the variational inequality as starting point derive suitable ProxNets for its (numerical) solution. Let $(\mathcal{H}, (\cdot,\cdot)_{\mathcal{H}})$ be a separable Hilbert space with topological dual space denoted by $\mathcal{H}'$ and let ${}_{\mathcal{H}'}\langle\cdot,\cdot\rangle_{\mathcal{H}}$ be the associated dual pairing. Let $a : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ be a bilinear form, let $f : \mathcal{H} \to \mathbb{R}$ be a functional and let $\mathcal{K} \subset \mathcal{H}$ be a subset of $\mathcal{H}$. We consider the variational inequality problem

$$\text{find } u \in \mathcal{K}\text{:} \quad a(u, v-u) \geq f(v-u), \quad \forall v \in \mathcal{K}. \tag{13}$$

Assumption 4.1. The bilinear form $a : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is bounded and coercive on $\mathcal{H}$, i.e. there exists constants $C_-, C_+ > 0$ such that for any $v, w \in \mathcal{H}$ it holds

$$a(v, w) \leq C_+ \|v\|_{\mathcal{H}} \|w\|_{\mathcal{H}} \quad \text{and} \quad a(v, v) \geq C_- \|v\|_{\mathcal{H}}^2.$$

Moreover, $f \in \mathcal{H}'$ and $\mathcal{K} \subset \mathcal{H}$ is nonempty, closed and convex.

Problem (13) arises in various applications in the natural sciences, engineering and finance. It is well-known that there exists a unique solution $u \in \mathcal{K}$ under Assumption 4.1, see, e.g., [14, Theorem A.3.3] for a proof. We also mention that well-posedness of Problem (13) is ensured under weaker conditions as Assumption 4.1, in particular, the coercivity requirement may be relaxed as shown in [8]. For this article, however, we focus on the bounded and coercive case in order to obtain numerical convergence rates for ProxNet approximations.

4.1 Fixed-Point Approximation by ProxNets

Theorem 4.2. *Let Assumption 4.1 hold, and define $\mathcal{H}_1 := \mathcal{H}_0 := \mathcal{H}$. Then, there exists a one-layer ProxNet Φ as in Equation 3 such that $u \in \mathcal{K}$ is the unique fixed-point of Φ. Moreover, for a given $u^0 \in \mathcal{H}$ define the iteration $u^k := \Phi(u^{k-1})$, $k \in \mathbb{N}$. There are constants $L_{\Phi,\lambda} \in (0, 1)$ and $C = C(u^0) > 0$, independent of k, such that*

$$\|u - u^k\| \leq C L_{\Phi,\lambda}^k, \quad k \in \mathbb{N}. \tag{14}$$

Proof. We recall the fixed-point argument, e.g. in [14, Theorem A.3.3], for proving existence and uniqueness of u as a starting point, since it is the base for the ensuing ProxNet construction: Assumption 4.1 ensures that $a(v, \cdot), f \in \mathcal{H}'$ for any $v \in \mathcal{H}$. The Riesz representation theorem yields the existence of $A \in \mathcal{L}(\mathcal{H})$ and $F \in \mathcal{H}$ such that for all $v, w \in \mathcal{H}$

$$(Av, w)_{\mathcal{H}} = a(v, w) \quad \text{and} \quad (F, v)_{\mathcal{H}} = f(v).$$

Since $\mathcal{K}$ is closed convex, the $\mathcal{H}$-orthogonal projection $P_{\mathcal{K}} : \mathcal{H} \to \mathcal{K}$ onto $\mathcal{K}$ is well-defined and for any $\omega > 0$ there holds

$$u \text{ solves (13)} \qquad \Longleftrightarrow \qquad u = P_{\mathcal{K}}(\omega(F - Au) + u).$$

Hence, u is a fixed-point of the mapping

$$T_\omega : \mathcal{H} \to \mathcal{H}, \quad v \mapsto P_{\mathcal{K}}(\omega(F - Av) + v).$$

By Assumption 4.1 it is now possible to choose $\omega > 0$ sufficiently small, so that T_ω is a contraction on $\mathcal{H}$, which proves existence and uniqueness of u. The optimal choice in terms of the bounds C_-, C_+ is $\omega^* = C_-/C_+^2$, leading to $\|T_{\omega^*}\|_{\mathcal{L}(\mathcal{H})}^2 = (1 - C_1^2/C_2^2) < 1$, see e.g. [14, Theorem A.3.3].

To transfer this proof of in the ProxNet setting, we denote by $\iota_{\mathcal{K}}$ the *indicator function* of $\mathcal{K}$ given by

$$\iota_{\mathcal{K}} : \mathcal{H} \to (-\infty, \infty], \quad v \mapsto \begin{cases} 0, & \text{if } v \in \mathcal{K}, \\ \infty, & \text{otherwise.} \end{cases}$$

Since $\mathcal{K}$ is closed convex, it holds that $\iota_{\mathcal{K}} \in \Gamma_0(\mathcal{H})$ and $\operatorname{prox}_{\iota_{\mathcal{K}}} = P_{\mathcal{K}}$ (cf. [2, Examples 1.25 and 12.25]). Now let $m = 1$, $\mathcal{H}_1 = \mathcal{H}$, $W_1 := I - \omega A \in \mathcal{L}(\mathcal{H})$, $b_1 := \omega F \in \mathcal{H}$, and $R_1 := \operatorname{prox}_{\iota_{\mathcal{K}}}$, where $\omega > 0$ is such that $I - \omega A$ is a contraction. Define the ProxNet

$$\Phi : \mathcal{H} \to \mathcal{H}, \quad v \mapsto (1 - \lambda)v + \lambda \underbrace{R_1(W_1 v + b_1)}_{:= T_1(v)}.$$

Since $\|W_1\|_{\mathcal{L}(\mathcal{H})} < 1$, Assumption 3.1 is satisfied for $\lambda \in (0, 1]$ and Theorem 3.2 yields that the iteration $u^k := \Phi(u^{k-1})$ converges for any $u^0 \in \mathcal{H}$ to a unique fixed-point $u^* \in \mathcal{H}$ with error bounded by (14) and $L_{\Phi,\lambda} := (1 - \lambda) + \lambda \|W_1\|_{\mathcal{L}(\mathcal{H})} \in (0, 1)$. Since $\Phi(v) = (1 - \lambda)v + \lambda T_1(v)$, it follows that u^* is in turn the unique fixed-point of T_1, hence $u = u^*$, which proves the claim. □

Remark 4.3. In the fashion of Example 3.4, we may construct an augmented ProxNet $\widetilde{\Phi} : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$ such that $\widetilde{\Phi}(v, F) = \Phi(v)$ for any $v \in \mathcal{H}$, where $F \in \mathcal{H}$ is the Riesz representative of $f \in \mathcal{H}'$ in Problem (13). The only difference is that F has to be multiplied with ω in the first linear transform to obtain $b_1 = \omega F$ instead of F as bias vector. The parameters of $\widetilde{\Phi}$ in this construction are independent of F, hence Theorem 3.5 yields that for any $f \in \mathcal{H}'$ (resp. $F \in \mathcal{H}$) and $x^0 \in \mathcal{H}$ it holds

$$\|u - \widetilde{u}^k\| \leq C L^k_{\Phi,\lambda}, \quad k \in \mathbb{N},$$

where $\widetilde{u}^k := \widetilde{\Phi}(u^{k-1}, F)$.

The previous remark shows that one fixed ProxNet is sufficient to solve Problem (13) for any $f \in \mathcal{H}'$. A similar result is achieved if the set $\mathcal{K} \subset \mathcal{H}$ associated Problem (13) is parameterized by a suitable family of functions:

Example 4.4 (Obstacle-to-solution operator)**.** Let $\mathcal{H}$ be a Hilbert space of real-valued functions over a domain $\mathcal{D} \subset \mathbb{R}^d$ such that $C(\mathcal{D}) \cap \mathcal{H}$ is a dense subset, e.g., $\mathcal{H} = L^2(\mathcal{D})$ or $\mathcal{H} = H^1(\mathcal{D})$, and let $\mathcal{K} := \{v \in \mathcal{H} | \, v \geq g \text{ almost everywhere}\}$ for a sufficiently smooth function $g : \mathcal{D} \to \mathbb{R}$. With this choice of $\mathcal{K}$, (13) is an *obstacle problem* and $P_{\mathcal{K}}(v) = \max(v, g)$ holds for any $v \in \mathcal{H} \cap C(\mathcal{D})$. We construct a ProxNet approximation to the *obstacle-to-solution operator* $O_g : \mathcal{H} \to \mathcal{H}, \; g \mapsto u$ corresponding to Problem (13) with $\mathcal{K} = \{v \in \mathcal{H} | \, v \geq g \text{ almost everywhere}\}$.

Assume $\Phi(v) = P_{\mathcal{K}}(W_1 v + b_1)$ for $W_1 \in \mathcal{L}(\mathcal{H})$ and $b_1 \in \mathcal{H}$ are as in Theorem 4.2 and let $\mathcal{K}_0 := \{v \in \mathcal{H} | \, v \geq 0 \text{ almost everywhere}\}$. To obtain a ProxNet that uses the obstacle $g \in \mathcal{H}$ as input, we define

$$\widetilde{\Phi} : \mathcal{H} \oplus \mathcal{H} \to \mathcal{H}, \quad (v, g) \mapsto \widetilde{T}_1(v, g) = (\widetilde{T}_2^{(1)} \circ \widetilde{T}_1^{(1)})(v, g)$$

via $\widetilde{T}_{j_1}^{(1)}(v, g) := R_{j_1}^{(1)}(W_{j_1}^{(1)}(v, g) + b_{j_1}^{(1)})$ which are, for $j_1 \in \{1, 2\}$, defined by

$$\begin{aligned}
&W_1^{(1)} : \mathcal{H} \oplus \mathcal{H} \to \mathcal{H} \oplus \mathcal{H}, \; (v_1, v_2) \mapsto (W_1 v_1 - v_2, v_2), \\
&b_1^{(1)} := (b_1, 0) \in \mathcal{H} \oplus \mathcal{H}, \quad R_1^{(1)} := \operatorname{prox}_{\psi_1^{(1)}}, \quad \psi_1^{(1)}(v, g) := \iota_{\mathcal{K}_0}(v), \\
&W_2^{(1)} : \mathcal{H} \oplus \mathcal{H} \to \mathcal{H}, \; (v_1, v_2) \mapsto v_1 + v_2, \quad b_2^{(1)} := 0 \in \mathcal{H}, \quad R_2^{(1)} := \mathrm{id} \in \mathcal{A}(\mathcal{H}).
\end{aligned}$$

Note that this yields $W_1^{(1)} \in \mathcal{L}(\mathcal{H} \oplus \mathcal{H})$, $W_2^{(1)} \in \mathcal{L}(\mathcal{H})$, and $R_1^{(1)}(v_1, v_2) = (P_{\mathcal{K}_0} v_1, v_2)$ for all $v_1, v_2 \in \mathcal{H}$. It now follows for any given $v, g \in \mathcal{H}$ and $\mathcal{K} := \{v \in \mathcal{H} | \, v \geq g \text{ almost everywhere}\}$

$$\begin{aligned}
\Phi(v) &= P_{\mathcal{K}}(W_1 v + b_1) \\
&= P_{\mathcal{K}_0}(W_1 v + b_1 - g) + g \\
&= R_2^{(1)}(W_2^{(1)}(P_{\mathcal{K}_0}(W_1 v + b_1 - g), g) + b_2^{(1)}) \\
&= \widetilde{T}_2^{(1)}\left((P_{\mathcal{K}_0}(W_1 v + b_1 - g), g)\right) \\
&= \widetilde{T}_2^{(1)} \circ (R_1^{(1)}(W_1^{(1)}(v, g) + b_1^{(1)})) \\
&= \widetilde{\Phi}(v, g).
\end{aligned}$$

As in Example 3.4 we concatenate $\widetilde{\Phi}$ to obtain the operator

$$\widetilde{O}_g : \mathcal{H} \oplus \mathcal{H} \to \mathcal{H}, \quad (x, g) \mapsto \left[\widetilde{\Phi}(\cdot, g) \bullet \cdots \bullet \widetilde{\Phi}(\cdot, g)\right](x).$$

Convergence of $\widetilde{O}_g(x^0, g)$ to u for any choice of $(x^0, g) \in \mathcal{H} \oplus \mathcal{H}$ is again guaranteed as the number of concatenations tends to infinity. Therefore, as in Example 3.4, we are able to solve a family of obstacle problems with parameter $g \in \mathcal{H}$ with only one ProxNet $\widetilde{\Phi}$.

A combination of the ProxNets from Remark 4.3 and Example 4.4 enables us to consider both, f and $\mathcal{K}$ in (13), as input variables of a suitable NN $\widetilde{\Phi} : \mathcal{H} \oplus \mathcal{H} \oplus \mathcal{H} \to \mathcal{H}$. This allows, in particular, to construct an approximation of the data-to-solution operator to Problem (13) that maps $(F, g) \in \mathcal{H} \oplus \mathcal{H}$ to u.

5 Example: Linear Matrix Complementarity Problems

Common examples for Problem (13) arise in financial and engineering applications, where the bilinear form $a : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ stems from a second order elliptic or parabolic differential operator. In this case, $\mathcal{H} \subset H^s(\mathcal{D})$, where $H^s(\mathcal{D})$ is the Sobolev space of smoothness $s > 0$ with respect to the spatial domain $\mathcal{D} \subset \mathbb{R}^n$, $n \in \mathbb{N}$. Coercivity and boundedness of a as in Assumption 4.1 often arises naturally in this setting. To obtain a computationally tractable problem it is necessary to discretize (13), for instance by a Galerkin approximation with respect to a finite dimensional subspace $\mathcal{H}_d \subset \mathcal{H}$. To illustrate this, we assume that $\dim(\mathcal{H}_d) = d \in \mathbb{N}$ is a suitable finite-dimensional subspace with basis $\{v_1, \dots, v_d\}$ and consider an obstacle problem with $\mathcal{K} = \{v \in \mathcal{H} |\, v \geq g \text{ almost everywhere}\}$ for a smooth function $g \in \mathcal{H}$.

Following Example 4.4 we introduce the set $\mathcal{K}_0 := \{v \in \mathcal{H} |\, v \geq 0 \text{ almost everywhere}\}$ and note that Problem (13) is equivalent to finding $u = u_0 + g \in \mathcal{K}$

$$\text{with } u_0 \in \mathcal{K}_0 \text{ such that:} \quad a(u_0, v - u_0) \geq f(v - u_0) - a(g, v - u_0), \quad \forall v \in \mathcal{K}_0. \tag{15}$$

5.1 Discretization and Matrix LCP

Any element $v \in \mathcal{H}_d$ may be expanded as $v = \sum_{i=1}^d w_i v_i$ for a coefficient vector $w \in \mathbb{R}^d$. To preserve non-negativity of the discrete approximation to (15), we assume that $v \in \mathcal{K}_0$ if and only if the basis coordinates are nonnegative, i.e., if $w \in \mathbb{R}^d_{\geq 0}$. This property holds, for instance, in finite element approaches. We write the discrete solution as $u_d = \sum_{i=1}^d x_i v_i$. Then $u_d \in \mathcal{K}_0$ if and only if $x \in \mathbb{R}^d_{\geq 0}$. Consequently, the discrete version of (15) is to

$$\text{find } x \in \mathbb{R}^d_{\geq 0}\colon \quad (y - x)^\top \mathbf{A} x \geq (y - x)^\top c, \quad \forall y \in \mathbb{R}^d_{\geq 0}, \tag{16}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ and the vector $c \in \mathbb{R}^d$ are given by

$$\mathbf{A}_{ij} := a(v_j, v_i) \quad \text{and} \quad c_i := {}_{\mathcal{H}'}\langle f, v_i \rangle_{\mathcal{H}} - a(g, v_i), \quad i, j \in \{1, \dots, d\}. \tag{17}$$

Problem (16) is equivalent to the *linear complementary problem* (LCP) to find $x \in \mathbb{R}^d$ such that for $\mathbf{A} \in \mathbb{R}^{d \times d}$ and $c \in \mathbb{R}^d$ as in (17) it holds

$$\mathbf{A} x \geq c, \quad x \geq 0, \quad x^\top (\mathbf{A} x - c) = 0, \tag{18}$$

see, e.g., [14, Lemma 5.1.3]. If $a : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is bounded and coercive as in Assumption 4.1, it readily follows that

$$C_- \|x\|_2^2 \leq x^\top \mathbf{A} x \leq C_+ \|x\|_2^2, \quad x \in \mathbb{R}^d, \tag{19}$$

where the constants $C_+ \geq C_- > 0$ stem from Assumption 4.1 and $\|\cdot\|_2$ is the Euclidean norm on $\mathbb{R}^d$. This implies in particular that the LCP (18) has a unique solution $x \in \mathbb{R}^d$, see [23, Theorem 4.2]. Equivalently, we may regard Problem (16), resp. (18), as variational inequality on the finite-dimensional Hilbert space $\mathbb{R}^d$ equipped with the Euclidean scalar product $(\cdot, \cdot)_2$. Well-posedness then follows directly from Assumption 4.1 with the identification $\mathcal{H} = \mathbb{R}^d$ and the discrete bilinear form $a : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $(x, y) \mapsto x^\top \mathbf{A} y$.

5.2 Solution of Matrix LCPs by ProxNets

The purpose of this section is to show that several well-known iterative algorithms to solve (finite-dimensional) LCPs may be recovered as particular cases of ProxNets in the setting of Section 2. To this end, we fix $d \in \mathbb{N}$ and use the notation $\mathcal{H} := \mathbb{R}^d$ for convenience. We denote by $\{e_1, \dots, e_d\} \subset \mathbb{R}^d$ the canonical basis of $\mathcal{H}$. To approximately solve LCPs by ProxNets, and to introduce a numerical LCP solution map, we introduce the scalar and vector-valued Rectified Linear Unit (ReLU) activation function.

Definition 5.1. The scalar ReLU activation function ϱ is defined as $\varrho : \mathbb{R} \to \mathbb{R}, x \mapsto \max(x, 0)$. The component-wise ReLU activation in $\mathbb{R}^d$ is given by

$$\varrho^{(d)} : \mathbb{R}^d \to \mathbb{R}^d, \quad x \mapsto \sum_{i=1}^{d} \varrho((x, e_i)_{\mathcal{H}})e_i. \tag{20}$$

Remark 5.2. The scalar ReLU activation function ϱ satisfies $\varrho = \operatorname{prox}_{\iota_{[0,\infty)}}$ with $\iota_{[0,\infty)} \in \Gamma_0(\mathbb{R})$ (see [5, Example 2.6]). This in turn yields that $\varrho^{(d)} \in \mathcal{A}(\mathbb{R}^d)$ for any $d \in \mathbb{N}$ by [5, Proposition 2.24].

Example 5.3 (PJORNet)**.** Consider the LCP (18) with matrix $\mathbf{A}$ and the triangular decomposition

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}, \tag{21}$$

where $\mathbf{D} \in \mathbb{R}^{d\times d}$ contains the diagonal entries of $\mathbf{A}$, and $\mathbf{L}, \mathbf{U} \in \mathbb{R}^{d\times d}$ are the (strict) lower and upper triangular parts of $\mathbf{A}$, respectively. The *projected Jacobi* (PJOR) overrelaxation method to solve LCP (18) is given as:

Algorithm 1 Projected Jacobi overrelaxation method

Given: initial guess $x^0 \in \mathbb{R}^d$, relaxation parameter $\omega > 0$ and tolerance $\varepsilon > 0$.

for $k = 0, 1, 2, \dots$ **do**
  $x^{k+1} = \max\big((\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})x^k + \omega\mathbf{D}^{-1}c, 0\big)$
  if $\|x^{k+1} - x^k\|_2 < \varepsilon$ **then**
    return x^{k+1}
  end if
end for

The max-function in Algorithm 1 acts component-wise on each entry of a vector in $\mathbb{R}^d$. Hence, one iteration of the PJOR may be expressed as a ProxNet in Model (3) with $m = 1$, $\lambda = 1$ and $\varrho^{(d)}$ from Equation (20) as

$$\Phi_{PJOR} : \mathbb{R}^d \to \mathbb{R}^d, \quad x \mapsto T_1(x) := \varrho^{(d)}(\underbrace{(\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})}_{=:W_1} x + \underbrace{\omega\mathbf{D}^{-1}c}_{:=b_1}).$$

If $\mathbf{A}$ satisfies (19) for constants $C_+ \geq C_- > 0$, it holds that

$$\begin{aligned}
\|W_1\|^2_{\mathcal{L}(\mathcal{H})} &= \|\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A}\|_2^2 \\
&= \sup_{x\in\mathbb{R}^d, \|x\|_2=1} x^\top x - \omega x^\top \mathbf{D}^{-1}(\mathbf{A}^\top + \mathbf{A})x + \omega^2 (x\mathbf{D}^{-1}\mathbf{A})^\top \mathbf{D}^{-1}\mathbf{A}x \\
&\leq 1 - 2\omega \min_{i\in\{1,\dots,d\}} \frac{1}{\mathbf{A}_{ii}} C_- + \omega^2 \max_{i\in\{1,\dots,d\}} \frac{1}{\mathbf{A}_{ii}^2} \|\mathbf{A}\|_2^2 \\
&\leq 1 - 2\omega \frac{C_-}{C_+} + \omega^2 \frac{\|\mathbf{A}\|_2^2}{C_-^2} =: \Lambda(\omega).
\end{aligned}$$

The choice $\omega^* := C_-^3/(C_+\|\mathbf{A}\|_2^2)$ minimizes Λ such that $\Lambda(\omega^*) < 1$. Moreover, $\Lambda(0) = 1$, Λ is strictly decreasing on $[0, \omega^*]$, and increasing for $\omega > \omega^*$. Hence, there exists $\overline{\omega} > 0$ such that for any $\omega \in (0, \overline{\omega})$ the mapping $\Phi_{PJOR} : \mathbb{R}^d \to \mathbb{R}^d$ is a contraction. An application of Theorem 3.2 then shows that Algorithm (1) converges linearly for suitable $\omega > 0$ and any initial guess x^0. In the special case that $\mathbf{A}$ is strictly diagonally dominant, choosing $\omega = 1$ is sufficient to ensure convergence, i.e., no relaxation before the activation is necessary.

Example 5.4 (PSORNet)**.** Another popular algorithm to numerically solve LCPs is the *projected successive overrelaxation* (PSOR) method in Algorithm 2.

Algorithm 2 Projected successive overrelaxation algorithm

Given: initial guess $x^0 \in \mathbb{R}^d$, relaxation parameter $\omega > 0$ and tolerance $\varepsilon > 0$.

for $k = 0, 1, 2, \dots$ **do**
  for $i = 1, 2, \dots, d$ **do**
    $y_i^{k+1} = \frac{1}{\mathbf{A}_{ii}} \left(c_i - \sum_{j=0}^{i-1} \mathbf{A}_{ij} x_j^{k+1} - \sum_{j=i+1}^{d} \mathbf{A}_{ij} x_j^k \right)$
    $x_i^{k+1} = \max((1-\omega) x_i^k + \omega y_i^{k+1}, 0)$
  end for
  if $\|x^{k+1} - x^k\|_2 < \varepsilon$ **then**
    return x^{k+1}
  end if
end for

To represent the PSOR-iteration by a ProxNet as in (3), we use the scalar ReLU activation ϱ from Definition 5.1 and and define for $i \in \{1, \dots, d\}$

$$R_i : \mathbb{R}^d \to \mathbb{R}^d, \quad x \mapsto \varrho((x, e_i)_{\mathcal{H}}) e_i + \sum_{j=1,\, j \neq i}^{d} x_j e_j. \tag{22}$$

In contrast to $\varrho^{(d)}$ in Equation (20), the activation operator R_i takes the maximum only with respect to the i-th entry of the input vector. Nevertheless, $R_i \in \mathcal{A}(\mathbb{R}^d)$ holds again by [5, Proposition 2.24]. Now define $b_i \in \mathbb{R}^d$ and $W_i \in \mathbb{R}^{d \times d}$ by

$$b_i = (0, \dots, 0, \underbrace{\omega \frac{c_i}{\mathbf{A}_{ii}}}_{i\text{-th entry}}, 0, \dots, 0), \quad (W_i)_{lj} = \begin{cases} 1 - \omega & l = j = i, \\ 1 & l = j \in \{1, \dots, d\} \setminus \{i\}, \\ -\omega \frac{\mathbf{A}_{ij}}{\mathbf{A}_{ii}}, & l = i, j \in \{1, \dots, d\} \setminus \{i\}, \\ 0, & \text{elsewhere}, \end{cases}$$

and let $T_i(x) := R_i(W_i x + b_i)$ for $x \in \mathbb{R}^d$. Given the k-th iterate x^k and $x_1^{k+1}, \dots, x_{i-1}^{k+1}$ from the inner loop of Algorithm 2, it follows for $z^{k,i-1} := (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^k, \dots, x_d^k)^\top$ that

$$x_i^{k+1} = z_i^{k,i}, \quad z^{k,i} = T_i(z^{k,i-1}), \quad i \in \{1, \dots, d\},\, k \in \mathbb{N}. \tag{23}$$

As $z^{k-1,d} = z^{k,0} = x^k$ for $k \in \mathbb{N}$, this shows $x^{k+1} = \Phi_{PSOR}(x^k)$ for

$$\Phi_{PSOR} : \mathbb{R}^d \to \mathbb{R}^d, \quad x \mapsto (T_d \circ \cdots \circ T_1)(x). \tag{24}$$

Provided (19) holds, we derive similarly to Example 5.3

$$\begin{aligned} \|W_i\|_2^2 &= \sup_{x \in \mathbb{R}^d, \|x\|_2 = 1} x^\top x - 2 \frac{\omega}{\mathbf{A}_{ii}} x^\top \mathbf{A}_{[i]} x_i + \frac{\omega^2}{\mathbf{A}_{ii}^2} (x^\top \mathbf{A}_{[i]})^2 \\ &\leq 1 - 2\omega \frac{1}{\mathbf{A}_{ii}} C_- + \frac{\omega^2}{\mathbf{A}_{ii}^2} \|\mathbf{A}\|^2, \end{aligned}$$

where $\mathbf{A}_{[i]}$ denotes the i-th row of $\mathbf{A}$. Hence, $\omega^* := C_-^3 / (C_+ \|\mathbf{A}\|_2^2)$ is sufficient to ensure that Φ_{PSOR} is a contraction, and convergence to a unique fixed-point follows as in Theorem 3.2.

Remark 5.5. Both, the PJORNet and PSORNet from Examples 5.3 and 5.4 may be augmented as in 3.4 to take $c \in \mathbb{R}^d$ as additional input vector, and therefore to solve the LCP (18) for varying c. That is, concatenation of the PJORNet/PSORNet again yields an approximation to the solution operator $O_c : \mathbb{R}^d \to \mathbb{R}^d,\ c \mapsto x$ associated to the LCP (18). This is of particular interest, for instance, in the valuation of American options, where a collection of LCPs with varying model parameters has to be solved, see [14, Chapter 5] and the numerical examples in Section 7. Recall that $c_i := {}_{\mathcal{H}'}\langle f, v_i \rangle_{\mathcal{H}} - a(g, v_i)$ if the matrix LCP stems from a discretized obstacle problem as introduced in the beginning of this section. Hence, by varying c it is possible to modify the right hand side f, as well as the obstacle g, of the underlying variational inequality.

5.3 Solution of Parametric Matrix LCPs by ProxNets

In this section we construct ProxNets that take *arbitrary* LCPs $(\mathbf{A}, c)$ in finite-dimensional, Euclidean space as input, and output approximations of the solution x to (18) with any prescribed accuracy. Consequently, these ProxNets realize approximate *data-to-solution operators*

$$O_{\mathbf{A},c} : \{\mathbf{A} \in \mathbb{R}^{d^2} \mid \text{there are } C_-, C_+ > 0 \text{ s.t. } \mathbf{A} \text{ satisfies (19)}\} \otimes \mathbb{R}^d \to \mathbb{R}^d,\ (\mathbf{A}, c) \mapsto x.$$

The key step is to realize Algorithm (1) for any given matrix $\mathbf{A}$, meaning the weights of the NN may not depend on $\mathbf{A}$ as in the previous section. To this end, we use that the multiplication of real numbers may be emulated by ReLU-NNs with controlled error and growth bounds on the layers and size of the network. This was first shown in [27], and then extended to the multiplication of $n \in \mathbb{N}$ real numbers in [24].

Proposition 5.6. *[24, Proposition 2.6] For any $\delta_0 \in (0,1)$, $n \in \mathbb{N}$ and $\Theta \geq 1$, there exists a ProxNet $\widetilde{\prod}^n_{\delta_0,\Theta} : \mathbb{R}^n \to \mathbb{R}$ of the form (2) such that*

$$\begin{aligned} \sup_{(x_1,\dots,x_n)\in[-\Theta,\Theta]^n} \left| \prod_{i=1}^n x_i - \widetilde{\prod}^n_{\delta_0,\Theta}(x_1,\dots,x_n) \right| &\leq \delta_0, \\ \operatorname*{ess\,sup}_{(x_1,\dots,x_n)\in[-\Theta,\Theta]^n} \sup_{j\in\{1,\dots,n\}} \left| \partial_{x_j} \prod_{i=1}^n x_i - \partial_{x_j} \widetilde{\prod}^n_{\delta_0,\Theta}(x_1,\dots,x_n) \right| &\leq \delta_0, \end{aligned} \tag{25}$$

where ∂_{x_j} is the weak derivative with respect to x_j. The neural network $\widetilde{\prod}^n_{\delta_0,\Theta}$ uses only ReLUs as in Definition 5.1 as proximal activations. There exists a constant C, independent of $\delta_0 \in (0,1)$, $n \in \mathbb{N}$ and $\Theta \geq 1$, such that the number of layers $m_{n,\delta_0,\Theta} \in \mathbb{N}$ of $\widetilde{\prod}^n_{\delta_0,\Theta}$ is bounded by

$$m_{n,\delta_0,\Theta} \leq C \left(1 + \log(n) \log\left(\frac{n\Theta^n}{\delta}\right)\right).$$

Remark 5.7. For our purposes, it is sufficient to consider the cases $n \in \{2,3\}$, therefore we assume without loss of generality that there is a constant C, independent of $\delta_0 \in (0,1)$ and $\Theta \geq 1$, such that for $n \in \{2,3\}$ it holds

$$m_{n,\delta_0,\Theta} \leq C \left(1 + \log\left(\frac{\Theta}{\delta_0}\right)\right).$$

Moreover, we may assume without loss of generality that $m_{2,\delta_0,\Theta} = m_{3,\delta_0,\Theta}$, as it is always possible to add ReLU-layers that emulate the identity function to the shallower network (see [24, Section 2] for details).

With this at hand, we are ready to proof the main result of this section.

Theorem 5.8. *Let $\Theta \geq 2$ be a fixed constant, $d \geq 2$ and let $(\mathbf{A}, c) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d$ be any tuplet such that $\|c\|_\infty \leq \Theta$ and $\mathbf{A} \in \mathbb{R}^{d\times d}$ satisfies (19) with $\Theta \geq C_+ \geq C_- \geq \Theta^{-1} > 0$. For the triangular decomposition $\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}$ as in (21), define $z_{\mathbf{A}} := \operatorname{vec}(\mathbf{D}^{-1} + \mathbf{L} + \mathbf{U}) \in \mathbb{R}^{d^2}$, where $\operatorname{vec} : \mathbb{R}^{d\times d} \to \mathbb{R}^{d^2}$ is the row-wise vectorization of a $\mathbb{R}^{d\times d}$-matrix. Let x^* be the unique solution to the LCP $(\mathbf{A}, c)$ and let $\widetilde{x}^0 \in \mathbb{R}^d$ be arbitrary such that $\|\widetilde{x}^0\|_2 \leq \Theta$.*

For any $\varepsilon > 0$ there exists a ProxNet

$$\widetilde{\Phi} : \mathbb{R}^d \oplus \mathbb{R}^{d^2} \oplus \mathbb{R}^d \to \mathbb{R}^d \tag{26}$$

as in (9) and a $k_\varepsilon \in \mathbb{N}$ such that

$$\|x^* - \widetilde{x}^{k_\varepsilon}\|_2 \leq \varepsilon$$

holds for the sequence $\widetilde{x}^k := \widetilde{\Phi}(\widetilde{x}^{k-1}, z_{\mathbf{A}}, c)$ generated by $\widetilde{\Phi}$ and any *tuplet $(\mathbf{A}, c)$ as above. Moreover, $k_\varepsilon \leq C_1(1+\log(|\varepsilon|))$, where $C_1 > 0$ only depends on Θ and $\widetilde{\Phi}$ has $m \leq C_2(1+\log(|\varepsilon|)+\log(d))$ layers, where $C_2 > 0$ is independent of Θ.*

Proof. Our strategy is to approximate Φ_{PJOR} for given $(\mathbf{A}, c)$ from Example 5.3 by $\widetilde{\Phi}(\cdot, z_{\mathbf{A}}, c)$. We achieve this by constructing $\widetilde{\Phi}$ based on the approximate multiplication NNs from Proposition 5.6 and show that Φ_{PJOR} and $\widetilde{\Phi}$ satisfy Assumption 3.3 to apply the error estimate from Theorem 3.5.

We start by defining the mapping $\widetilde{\Phi} : \mathbb{R}^d \oplus \mathbb{R}^{d^2} \oplus \mathbb{R}^d \to \mathbb{R}^d$ via

$$\widetilde{\Phi}(x, z_{\mathbf{A}}, c)_i = \max\left((1-\omega)x_i - \omega \sum_{j=1, j\neq i} \widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j, \frac{1}{\mathbf{A}_{ii}}, \mathbf{A}_{ij}\right) + \omega \widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}_{ii}}, c_i\right), 0\right),$$

for $i \in \{1, \dots, d\}$ and $0 < \omega := \Theta^{-6} \leq \frac{C_-^3}{C_+ \|\mathbf{A}\|_2^2} = \omega^*$ and $\delta_0 \in (0, d^{-3/2}]$.

We show in the following that $\widetilde{\Phi}$ is indeed a ProxNet. To bring the input into the correct order for multiplication, we define for $i \in \{1, \dots, d\}$ the binary matrix $W^{(i)} \in \mathbb{R}^{(2d+1)\times(d^2+2d)}$ by

$$\left(W^{(i)}\right)_{lj} := \begin{cases} 1 & l = j \in \{1, \dots, d\}, \\ 1 & l \in \{d+1, \dots, 2d\},\ j = d + d(i-1) + (l-d), \\ 1 & l = 2d+1,\ j = d + d^2 + i, \\ 0 & \text{elsewhere.} \end{cases}$$

Hence, we obtain

$$W^{(i)} \begin{pmatrix} x \\ z_{\mathbf{A}} \\ c \end{pmatrix} = \left(x^\top, (\mathbf{A}_{ij})_{j<i}, \frac{1}{\mathbf{A}_{ii}}, (\mathbf{A}_{ij})_{j>i}, c_i\right)^\top.$$

Now let $e_1, \dots, e_{d+2} \subset \mathbb{R}^{2d+1}$ be the canonical basis of $\mathbb{R}^{2d+1}$ and define furthermore $E_i^{(i)} := e_i^\top \in \mathbb{R}^{1\times(2d+1)}$, $E_j^{(i)} := [e_j \ e_{d+i} \ e_{d+j}]^\top \in \mathbb{R}^{3\times(2d+1)}$ for $j \in \{1, \dots, d\} \setminus \{i\}$ and $E_{d+1}^{(i)} := [e_{d+i} \ e_{2d+1}]^\top \in \mathbb{R}^{2\times(2d+1)}$. By Remark 5.7, we may assume that $\widetilde{\prod}^3_{\delta_0,\Theta}$ and $\widetilde{\prod}^2_{\delta_0,\Theta}$ have an identical number of layers, denoted by $m_{\delta_0,\Theta} \in \mathbb{N}$. Moreover, it is straightforward to construct a ProxNet $\mathrm{Id}_{m_{\delta_0,\Theta}} : \mathbb{R} \to \mathbb{R}$ with $m_{\delta_0,\Theta}$ layers that corresponds to the identity map, i.e. $\mathrm{Id}_{m_{\delta_0,\Theta}}(x) = x$ for all $x \in \mathbb{R}$. We use the concatenation from Definition 2.3 to define

$$\begin{aligned}
\widetilde{\Phi}_i^{(i)} &:= \mathrm{Id}_{m_{\delta_0,\Theta}} \bullet (E_i^{(i)} W^{(i)}) : \mathbb{R}^{d^2+2d} \to \mathbb{R} \\
\widetilde{\Phi}_j^{(i)} &:= \widetilde{\prod}^2_{\delta_0,\Theta} \bullet (E_j^{(i)} W^{(i)}) : \mathbb{R}^{d^2+2d} \to \mathbb{R}, \quad j \in \{1, \dots, d\} \setminus \{i\}, \\
\widetilde{\Phi}_{d+1}^{(i)} &:= \widetilde{\prod}^3_{\delta_0,\Theta} \bullet (E_{d+1}^{(i)} W^{(i)}) : \mathbb{R}^{d^2+2d} \to \mathbb{R}.
\end{aligned}$$

Note that this yields

$$\widetilde{\Phi}_i^{(i)}(x, z_{\mathbf{A}}, c) = x_i, \quad \widetilde{\Phi}_j^{(i)}(x, z_{\mathbf{A}}, c) = \widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j, \frac{1}{\mathbf{A}_{ii}}, \mathbf{A}_{ij}\right), \quad \widetilde{\Phi}_{d+1}^{(i)}(x, z_{\mathbf{A}}, c) = \widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}_{ii}}, c_i\right).$$

Furthermore, we set $n_1 := m_{\delta_0,\Theta} + 1$ and define $\widetilde{T}_{n_1}^{(+,i)} : \mathbb{R}^{d^2+d} \to \mathbb{R}$, $x \mapsto \varrho(W_{n_1}^{(+,i)} x)$, where $\varrho : \mathbb{R} \to \mathbb{R}$ is the (scalar) ReLU activation and $W_{n_1}^{(+,i)} \in \mathbb{R}^{1\times(d+1)}$ is given by

$$\left(W_{n_1}^{(+,i)}\right)_j := \begin{cases} 1-\omega & j = i, \\ -\omega & j \in \{1, \dots, d\} \setminus \{i\}, \\ \omega & j = d+1. \end{cases}$$

As $\widetilde{\Phi}_1^{(i)}, \dots, \widetilde{\Phi}_{d+1}^{(i)}$ have the same input dimension, the same number of $m_{\delta_0,\Theta}$ layers, and no skip connections, we may parallelize as in Definition 2.5 to ensure

$$\begin{aligned}
\widetilde{\Phi}(x, z_{\mathbf{A}}, c)_i &= \max\left((1-\omega)x_i - \omega \sum_{j=1, j\neq i} \widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j, \frac{1}{\mathbf{A}_{ii}}, \mathbf{A}_{ij}\right) + \omega \widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}_{ii}}, c_i\right), 0\right) \\
&= \left(\widetilde{T}_{n_1}^{(+)} \bullet P\left(\widetilde{\Phi}_1^{(i)}, \dots, \widetilde{\Phi}_{d+1}^{(i)}\right)\right)(x, z_{\mathbf{A}}, c).
\end{aligned}$$

It holds that $\widetilde{\Phi}_i := \widetilde{T}_{n_1}^{(+)} \bullet P\left(\widetilde{\Phi}_1^{(i)}, \dots, \widetilde{\Phi}_{d+1}^{(i)}\right)$ is a ProxNet as in Equation (9) with $\widetilde{\Phi}_i : \mathbb{R}^{d^2+2d} \to \mathbb{R}$ and $n_1 = m_{\delta_0,\Theta} + 1$ layers for any $i \in \{1, \dots, d\}$. We parallelize once more and obtain that $\widetilde{\Phi} := P(\widetilde{\Phi}_1, \dots, \widetilde{\Phi}_d)$ is a ProxNet with $m_{\delta_0,\Theta}+1$ layers that may be written as $\widetilde{\Phi} = \widetilde{T}_1^{(1)} \circ \cdots \circ \widetilde{T}_{n_1}^{(1)}$ for suitable one-layer networks $\widetilde{T}_1^{(1)} : \mathbb{R}^{d_{j-1}} \to \mathbb{R}^{d_j}$ and dimensions $d_j \in \mathbb{N}$ for $j \in \{1, \dots, n_1\}$ such that $d_0 = d^2 + 2d$ and $d_{n_1} = d$.

We now fix $(\mathbf{A}, c)$ and let $\Phi_{PJOR} := R(W_1 \cdot + b_1)$ be as in Example 5.3 with $\omega = \Theta^{-6}$, $W_1 = \mathbf{I}_d - \omega \mathbf{D}^{-1}\mathbf{A}$ and $b_1 := \omega \mathbf{D}^{-1} c$. This shows that Φ_{PJOR} has Lipschitz constant $L_\Phi = \|W_1\|_2 \leq \sqrt{1 - 2\Theta^{-4} + \Theta^{-8}} = 1 - \Theta^{-4} < 1$ and $\|b_1\|_2 \leq \omega\Theta^2 \leq \Theta^{-4}$.

Note that $|c_i|, 1/\mathbf{A}_{ii}, |\mathbf{A}_{ij}| \leq \Theta$ for any $i, j \in \{1, \dots, d\}$. Therefore, Proposition 5.6 yields for $\widetilde{x}_0 := (z_{\mathbf{A}}, c)$ and any $x \in \mathbb{R}^d$ with $\|x\|_\infty \leq \Theta$ that

$$\begin{aligned}
\|\Phi(x) - \widetilde{\Phi}(x, \widetilde{x}_0)\|_2^2 &= \|T_1(x) - \widetilde{T}_1(x, \widetilde{x}_0)\|_2^2 \\
&= \omega^2 \sum_{i=1}^{d} \left(\frac{c_i}{\mathbf{A}_{ii}} - \widetilde{\prod}_{\delta_0,\Theta}^2 \left(c_i, \frac{1}{\mathbf{A}_{ii}} \right) - \sum_{j=1, j\neq i}^{d} \frac{\mathbf{A}_{ij}}{\mathbf{A}_{ii}} x_j - \widetilde{\prod}_{\delta_0,\Theta}^3 \left(\mathbf{A}_{ij}, \frac{1}{\mathbf{A}_{ii}}, x_j \right) \right)^2 \\
&\leq \omega^2 d^3 \delta_0^2.
\end{aligned}$$

Hence, since $\delta_0 \in (0, d^{-3/2}]$ and $\omega = \Theta^{-6}$, Φ_{PSOR} and $\widetilde{\Phi}$ satisfy Assumption 3.3 with

$$\begin{aligned}
\widetilde{L} &:= 1 - \Theta^{-4} \in (0,1), \quad \delta := \omega d^{3/2} \delta_0 \geq 0, \quad \Theta_1 := \Theta \geq 2, \\
\Theta_0 &:= \Theta_1 - \|b_1\|_2 - \delta \geq \Theta - \Theta^{-4} - \omega d^{3/2} \delta_0 \geq \frac{123}{64}, \\
\Theta_2 &:= \Theta_0 - \delta/(1 - \widetilde{L}) \geq \Theta_0 - \frac{\Theta^{-6}}{\Theta^{-4}} \geq \frac{123}{64} - \frac{1}{4} > 0.
\end{aligned}$$

Theorem 3.5 then yields

$$\|x^* - \widetilde{x}^k\|_{\mathcal{H}} \leq C \left(\widetilde{L}^k + \delta \right),$$

where $C \leq \max(2\Theta_0, 1)/(1 - \widetilde{L}) \leq 2\Theta^5$ is independent of k. Given $\varepsilon > 0$, we choose

$$k_\varepsilon =: \left\lceil \frac{\log(\varepsilon) - \log(2C)}{\log(\widetilde{L})} \right\rceil, \quad \delta_0 := \frac{\min\left(1, \frac{\varepsilon}{2C\omega}\right)}{d^{3/2}} \geq \frac{\min\left(1, \frac{\varepsilon\Theta}{4}\right)}{d^{3/2}}$$

to ensure $\|x^* - \widetilde{x}^{k_\varepsilon}\| \leq \varepsilon$. Hence, $k_\varepsilon \leq C_1(1 + \log(|\varepsilon|))$, where $C_1 = C_1(\Theta) > 0$ is independent of d. Moreover, Proposition 5.6 and the choice of δ_0 show that $m_{\delta_0,\Theta} \leq C_2(1 + \log(|\varepsilon|) + \log(d))$, where $C_2 > 0$ is independent of Θ. The claim follows since $\widetilde{\Phi}$ has $n_1 = m_{\delta_0,\Theta} + 1$ layers by construction. □

As before, we may now concatenate $\widetilde{\Phi}$ to obtain the approximate data-to-solution operator

$$\widetilde{O}_{\mathbf{A},c}(x, \mathbf{A}, c) := \left[\widetilde{\Phi}(\cdot, z_{\mathbf{A}}, c) \bullet \cdots \bullet \widetilde{\Phi}(\cdot, z_{\mathbf{A}}, c) \right](x)$$

and obtain that $\widetilde{O}_{\mathbf{A},c}(x^0, \mathbf{A}, c) \approx O_{\mathbf{A},c}(\mathbf{A}, c) = x$ holds for any $(x^0, \mathbf{A}, c)$, satisfying the assumptions of Theorem 5.8.

Furthermore, the construction of $\widetilde{\Phi}$ by approximate ReLU-multiplications even allows to derive Lipschitz continuity of $\widetilde{O}_{\mathbf{A},c}$. This is established by the following version of Strang's Lemma for the approximate solutions of variational inequalities.

Theorem 5.9. *Let $(\mathbf{A}^{(1)}, c^{(1)})$ and $(\mathbf{A}^{(2)}, c^{(2)})$ be any two LCPs that satisfy the assumptions of Theorem 5.8 for some $\Theta \geq 2$. For $l \in \{1, 2\}$, let $A^{(l)} = \mathbf{D}^{(l)} + \mathbf{L}^{(l)} + \mathbf{U}^{(l)}$ be the decomposition of $A^{(l)}$ as in (21) and define $z_{\mathbf{A}^{(l)}} := \mathrm{vec}((\mathbf{D}^{(l)})^{-1} + \mathbf{L}^{(l)} + \mathbf{U}^{(l)}) \in \mathbb{R}^{d^2}$. For arbitrary $\varepsilon > 0$ let $\widetilde{\Phi}$ be the ProxNet as in (26), let $\widetilde{x}^0 \in \mathbb{R}^d$ be such that $\|\widetilde{x}^0\|_2 \leq \Theta$, and define the sequences*

$$\widetilde{x}^{(l),k} := \widetilde{\Phi}(\widetilde{x}^{(l),k-1}, z_{\mathbf{A}^{(l)}}, c^{(l)}),\ k \in \mathbb{N}, \quad \widetilde{x}^{(l),0} := \widetilde{x}^0, \quad l \in \{1, 2\}. \tag{27}$$

Furthermore, let $\|\cdot\|_F$ denote the Frobenius norm on $\mathbb{R}^{d\times d}$.

Then there is a constant $C>0$ depending only on Θ and d, such that for any $k\in\mathbb{N}_0$ and arbitrary, fixed $\varepsilon>0$ it holds that

$$\|\widetilde{x}^{(l),k}-\widetilde{x}^{(l),k}\|_2\leq \widetilde{C}\left(\|A^{(1)}-A^{(2)}\|_F+\|c^{(1)}-c^{(2)}\|_2\right). \tag{28}$$

Proof. By the construction of $\widetilde{\Phi}$ in Theorem 5.8 we have for any $x\in\mathbb{R}^d$, $l\in\{1,2\}$, and $i\in\{1,\dots,d\}$ that

$$\widetilde{\Phi}(x,z_{\mathbf{A}^{(l)}},c^{(l)})_i=\max\left((1-\omega)x_i-\omega\sum_{j=1,j\neq i}^{d}\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(l)}_{ii}},\mathbf{A}^{(l)}_{ij}\right)+\omega\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(l)}_{ii}},c^{(l)}_i\right),0\right).$$

Therefore, we estimate by the triangle inequality

$$\begin{aligned}
&|\widetilde{\Phi}(x,z_{\mathbf{A}^{(1)}},c^{(1)})_i-\widetilde{\Phi}(x,z_{\mathbf{A}^{(2)}},c^{(2)})_i|\\
\leq&\omega\sum_{j=1,j\neq i}^{d}\left|\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(1)}_{ii}},\mathbf{A}^{(1)}_{ij}\right)-\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(2)}_{ii}},\mathbf{A}^{(2)}_{ij}\right)\right|\\
&+\omega\left|\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(1)}_{ii}},c^{(1)}_i\right)-\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(2)}_{ii}},c^{(2)}_i\right)\right|\\
\leq&\omega\sum_{j=1,j\neq i}^{d}\left|\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(1)}_{ii}},\mathbf{A}^{(1)}_{ij}\right)-\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(1)}_{ii}},\mathbf{A}^{(2)}_{ij}\right)\right|\\
&\omega\sum_{j=1,j\neq i}^{d}\left|\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(1)}_{ii}},\mathbf{A}^{(2)}_{ij}\right)-\widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j,\frac{1}{\mathbf{A}^{(2)}_{ii}},\mathbf{A}^{(2)}_{ij}\right)\right|\\
&+\omega\left|\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(1)}_{ii}},c^{(1)}_i\right)-\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(1)}_{ii}},c^{(2)}_i\right)\right|\\
&+\omega\left|\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(1)}_{ii}},c^{(2)}_i\right)-\widetilde{\prod}^2_{\delta_0,\Theta}\left(\frac{1}{\mathbf{A}^{(2)}_{ii}},c^{(2)}_i\right)\right|.
\end{aligned}$$

By the assumptions on $(A^{(l)},c^{(l)})$, $l\in\{1,2\}$, it holds for any $i,j\in\{1,\dots,d\}$ that $1/\mathbf{A}^{(l)}_{ii}$, $\mathbf{A}^{(l)}_{ij}$, $c^{(l)}_i\in[-\Theta,\Theta]$. Hence, for any x with $\|x\|_\infty\leq\Theta$ we obtain by $\Theta\geq 2$ and the second estimate from Proposition 5.6

$$\begin{aligned}
&|\widetilde{\Phi}(x,z_{\mathbf{A}^{(1)}},c^{(2)})_i-\widetilde{\Phi}(x,z_{\mathbf{A}^{(2)}},c^{(2)})_i|\\
\leq&\omega\sum_{j=1,j\neq i}^{d}\left(\delta_0+\left|\frac{x_j}{\mathbf{A}^{(1)}_{ii}}\right|\right)\left|\mathbf{A}^{(1)}_{ij}-\mathbf{A}^{(2)}_{ij}\right|+\omega\left(\delta_0+|x_j\mathbf{A}^{(2)}_{ij}|\right)\left|\frac{1}{\mathbf{A}^{(1)}_{ii}}-\frac{1}{\mathbf{A}^{(2)}_{ii}}\right|\\
&+\omega\left(\delta_0+\frac{1}{\mathbf{A}^{(1)}_{ij}}\right)\left|c^{(1)}_i-c^{(2)}_i\right|+\omega\left(\delta_0+|c^{(2)}_i|\right)\left|\frac{1}{\mathbf{A}^{(1)}_{ii}}-\frac{1}{\mathbf{A}^{(2)}_{ii}}\right|\\
\leq&\omega 2(\delta_0\Theta^2+\Theta^4)\left(\sum_{j=1}^{d}\left|\mathbf{A}^{(1)}_{ij}-\mathbf{A}^{(2)}_{ij}\right|+\left|c^{(1)}_i-c^{(2)}_i\right|\right)\\
\leq&\omega(\delta_0\Theta^2+\Theta^4)\left(d^{1/2}\left(\sum_{j=1}^{d}\left|\mathbf{A}^{(1)}_{ij}-\mathbf{A}^{(2)}_{ij}\right|^2\right)^{1/2}+\left|c^{(1)}_i-c^{(2)}_i\right|\right).
\end{aligned}$$

We have used the mean-value theorem to obtain the bound

$$\left|\frac{1}{\mathbf{A}^{(1)}_{ii}} - \frac{1}{\mathbf{A}^{(2)}_{ii}}\right| \leq \Theta^2 \left|\mathbf{A}^{(1)}_{ii} - \mathbf{A}^{(2)}_{ii}\right|$$

in the second last inequality and the Cauchy-Schwarz inequality in the last step. We recall from the proof of Theorem 5.8 that $\omega = \Theta^{-6}$ and $\delta_0 \leq d^{-3/2}$, hence, there is a constant $C = C(\Theta, d) > 0$, depending only on the indicated parameters, such that for any $x \in \mathbb{R}^d$ with $\|x\|_\infty \leq \Theta$ it holds

$$\|\widetilde{\Phi}(x, z_{\mathbf{A}^{(1)}}, c^{(1)}) - \widetilde{\Phi}(x, z_{\mathbf{A}^{(2)}}, c^{(2)})\|_2 \leq C\left(\|\mathbf{A}^{(1)} - \mathbf{A}^{(2)}\|_F + \|c^{(1)} - c^{(2)}\|_2\right). \tag{29}$$

Moreover, for any $x, y \in \mathbb{R}$ such that $\|x\|_\infty, \|y\|_\infty \leq \Theta$, it holds by the mean-value theorem and the second estimate from Proposition 5.6

$$\begin{aligned}
&|\widetilde{\Phi}(x, z_{\mathbf{A}^{(1)}}, c^{(1)})_i - \widetilde{\Phi}(y, z_{\mathbf{A}^{(1)}}, c^{(1)})_i| \\
\leq &\left|\widetilde{\Phi}(x, z_{\mathbf{A}^{(1)}}, c^{(1)})_i - \widetilde{\Phi}(y, z_{\mathbf{A}^{(1)}}, c^{(1)})_i - ((\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})(x-y))_i\right| \\
&+ \left|((\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})(x-y))_i\right| \\
= &\omega\left|\sum_{j=1, j\neq i}^{d} \widetilde{\prod}^3_{\delta_0,\Theta}\left(x_j, \frac{1}{\mathbf{A}^{(1)}_{ii}}, \mathbf{A}^{(1)}_{ij}\right) - \widetilde{\prod}^3_{\delta_0,\Theta}\left(y_j, \frac{1}{\mathbf{A}^{(1)}_{ii}}, \mathbf{A}^{(1)}_{ij}\right) - \frac{\mathbf{A}^{(1)}_{ij}}{\mathbf{A}^{(1)}_{ii}}(x_j - y_j)\right| \\
&+ \left|((\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})(x-y))_i\right| \\
\leq &\omega\delta_0 \sum_{j=1, j\neq i}^{d} |x_j - y_j| + \left|((\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})(x-y))_i\right|
\end{aligned}$$

Hence, Young's inequality yields for any $\varepsilon > 0$ that

$$\begin{aligned}
&\|\widetilde{\Phi}(x, z_{\mathbf{A}^{(1)}}, c^{(1)}) - \widetilde{\Phi}(y, z_{\mathbf{A}^{(1)}}, c^{(1)})\|_2^2 \\
\leq &\sum_{i=1}^{d}\left(1 + \frac{1}{4\varepsilon}\right)\omega^2\delta_0^2\left(\sum_{j=1, j\neq i}^{d}|x_j - y_j|\right)^2 + (1+\varepsilon)\|(\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A})(x-y)\|_2^2 \\
\leq &\left(\left(1 + \frac{1}{4\varepsilon}\right)\omega^2\delta_0^2 d(d-1) + (1+\varepsilon)\|\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A}\|_2^2\right)\|x - y\|_2^2
\end{aligned} \tag{30}$$

where we have used the Cauchy-Schwarz inequality in the last step. From the proof of Theorem 5.8 we have as before that $\omega = \Theta^{-6}$, $\delta_0 \leq d^{-3/2}$, and, furthermore $\|\mathbf{I}_d - \omega\mathbf{D}^{-1}\mathbf{A}\|_2 \leq 1 - \Theta^4$. Setting $\varepsilon := \Theta^{-4}$ and using $\Theta \geq 2, d \geq 1$ therefore shows that $\widetilde{\Phi}(\cdot, z_{\mathbf{A}^{(1)}}, c^{(1)}) : \mathbb{R}^d \to \mathbb{R}^d$ is a contraction on $(\mathbb{R}^d, \|\cdot\|_2)$ with Lipschitz constant $\widetilde{L}_1 > 0$ bounded by

$$\widetilde{L}_1 \leq \left(\left(\Theta^{-12} + \frac{\Theta^{-8}}{4}\right)d^{-1} + (1 - \Theta^{-8})\right)^{1/2} \leq \sqrt{1 - \frac{11}{16}\Theta^{-8}} \in (0, 1). \tag{31}$$

Now let $(\widetilde{x}^{(l),k})$ for $l \in \{1, 2\}$ and $k \in \mathbb{N}_0$ denote the iterates as defined in (27) and recall from the proof of Theorem 3.5 that $\|\widetilde{x}^{(l),k}\|_\infty \leq \|\widetilde{x}^{(l),k}\|_2 \leq \Theta$. Therefore, we may apply the estimates in (29) and (30), and to obtain

$$\begin{aligned}
\|\widetilde{x}^{(1),k} - \widetilde{x}^{(2),k}\|_2 &\leq \|\widetilde{x}^{(1),k} - \widetilde{\Phi}(\widetilde{x}^{(2),k-1}, z_{\mathbf{A}^{(1)}}, c^{(1)})\|_2 + \|\widetilde{\Phi}(\widetilde{x}^{(2),k}, z_{\mathbf{A}^{(1)}}, c^{(1)}) - \widetilde{x}^{(2),k}\|_2 \\
&\leq \widetilde{L}_1\|\widetilde{x}^{(1),k-1} - \widetilde{x}^{(2),k-1}\|_2 + C\left(\|\mathbf{A}^{(1)} - \mathbf{A}^{(2)}\|_F + \|c^{(1)} - c^{(2)}\|_2\right) \\
&\leq C\left(\|\mathbf{A}^{(1)} - \mathbf{A}^{(2)}\|_F + \|c^{(1)} - c^{(2)}\|_2\right)\sum_{j=1}^{k-1}\widetilde{L}_1^j \\
&\leq \frac{C}{1 - \widetilde{L}_1}\left(\|\mathbf{A}^{(1)} - \mathbf{A}^{(2)}\|_F + \|c^{(1)} - c^{(2)}\|_2\right).
\end{aligned}$$

The claim follows since $C = C(\Theta, d)$ and $\widetilde{L}_1$ is bounded independently in ε and k by (31). □

6 PDASNet: From Linear to Superlinear Convergence

We have shown in the previous two sections that solutions to variational inequalities and the associated discrete LCPs (18) may be approximated by fixed-point iterations of ProxNets. The contraction property of the NNs yields linear convergence rates, both in finite and infinite-dimensional Hilbert spaces. In practice, one always works in the finite-dimensional setting, and we show in this section that even superlinear convergence is possible for a different ProxNet architecture. The key is to switch the analysis from fixed-point iterations to a primal-dual active set strategy, and to approximate the binary decision for the active/inactive sets by ReLU activation functions.

6.1 Primal-dual active set (PDAS) strategy

Besides PJOR and PSOR, to solve (18) the *primal-dual active set* (PDAS) method has been proposed in [15]. We construct corresponding DNN emulations. To introduce the PDAS, we first note that (18) may be reformulated as an LCP problem which is to find $x, \mu \in \mathbb{R}^d$ such that

$$\mathbf{A}x - \mu = c, \quad x \geq 0, \quad \mu \geq 0, \quad x^\top \mu = 0. \tag{32}$$

Let $\eta > 0$ be an arbitrary positive constant and define

$$\mathcal{C} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d, \quad (x, \mu) \mapsto \mu - \max(\mu - \eta x, 0),$$

so that Problem (32) is equivalent to find $(x, \mu) \in \mathbb{R}^d \times \mathbb{R}^d$ such that

$$\mathbf{A}x - \mu = c, \quad \mathcal{C}(x, \mu) = 0. \tag{33}$$

The equivalent formulation of (18) to (33) motivates the PDAS method given in Algorithm 3. By interpreting the PDAS as a semi-smooth Newton method, the authors show in [15] that Algorithm 3 converges globally for any initial value in finite time, and locally at a superlinear rate to the unique solution (x, μ) of Equation (33) under mild assumptions.

Algorithm 3 Primal-dual active set algorithm

Given: initial guesses $x^0, \mu^0 \in \mathbb{R}^d$, $\eta > 0$ and tolerance $\varepsilon > 0$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Set

$$\mathcal{I}_k := \{i \in \{1, \ldots, d\} : \mu_i^k - \eta x_i^k \leq 0\},$$
$$\mathcal{A}_k := \{i \in \{1, \ldots, d\} : \mu_i^k - \eta x_i^k > 0\}.$$

3: Solve $\mathbf{A}x^{k+1} - \mu^{k+1} = c$ such that $x_i^{k+1} = 0$ for $i \in \mathcal{A}_k$ and $\mu_i^{k+1} = 0$ for $i \in \mathcal{I}_k$.
4: **if** $\|x^{k+1} - x^k\|_2 < \varepsilon$ **then**
5: **return** x^{k+1}
6: **end if**
7: **end for**

Theorem 6.1. *[15, Theorem 3.1] Let* $\mathbf{A}$ *be a P-matrix, i.e., all principal minors of* $\mathbf{A}$ *are positive, and let* (x, μ) *be the solution of Equation* (33)*. Then, provided that* $\|x^0 - x\|_2 + \|\mu^0 - \mu\|_2$ *is sufficiently small, the PDAS converges to the solution* (x, μ) *in* finitely many steps*. Furthermore, the convergence is locally* superlinear*: for sufficiently large* $k \in \mathbb{N}$ *it holds that*

$$\|x - x^k\|_2 + \|\mu - \mu^k\|_2 \leq \widetilde{C}_k \left(\|x - x^{k-1}\|_2 + \|\mu - \mu^{k-1}\|_2 \right),$$

where $\widetilde{C}_k \geq 0$ *for all* $k \in \mathbb{N}$ *and* $\lim_{k\to\infty} \widetilde{C}_k = 0$.

Global of the PDAS requires stricter assumptions on $\mathbf{A}$ and is established in the next theorem.

Theorem 6.2. *[15, Theorem 3.3/3.4] Let* $\mathbf{A}$ *satisfy one of the following assumptions:*

- $\mathbf{A}$ *is a* M*-matrix, i.e.,* $\mathbf{A}$ *is regular,* $\mathbf{A}_{ij} \leq 0$ *for any* $i \neq j$ *and* $\mathbf{A}^{-1} \geq 0$*.*
- $\mathbf{A}$ *is a* P*-matrix and for every partitioning of the index set* $\{1, \dots, d\}$ *into disjoint sets* $\mathcal{I}$ *a* $\mathcal{A}$ *it holds*

$$\|(\mathbf{A}_{\mathcal{I},\mathcal{I}}^{-1}\mathbf{A}_{\mathcal{I},\mathcal{A}})_+\|_1 < 1, \quad \textit{and,} \quad \sum_{i\in\mathcal{I}}(\mathbf{A}_{\mathcal{I},\mathcal{I}}y_{\mathcal{I}})_i \geq 0$$

for any $y_{\mathcal{I}} \geq 0$*, where* $(\mathbf{B})_+$ *contains the positive parts of the elements of a matrix* $\mathbf{B}$*.*

Then, the PDAS converges for any initial guess (x^0, μ^0) *to the solution* (x, μ) *of Equation* (33) *in* finitely many steps *and the convergence is* locally superlinear*.*

Remark 6.3. The bounds (19) imply that $\mathbf{A}$ is a P-matrix, and therefore guarantee local convergence in Theorem 6.1. Interpreting the PDAS as a Newton method yields locally superlinear convergence, in contrast to the fixed-point iterations in Algorithms 1 and 2, which converge (globally) at linear speed. In most applications, this results in a significantly better performance of the PDAS algorithm. We emphasize, however, that the fixed-point approach works for abstract variational inequality problems in arbitrary Hilbert spaces as introduced in Section 4. Convergence for the PDAS, on the other hand, is only ensured for finite-dimensional LCPs and it cannot be expected to find a straightforward analogue of the PDAS for infinite-dimensional variational inequalities. For further details, we refer to examples and discussion in [15, Section 4].

By introducing the auxiliary variable μ, we now consider the solution operator

$$O_{PDAS,c} : \mathbb{R}^d \to \mathbb{R}^d \oplus \mathbb{R}^d, \quad c \mapsto (x, \mu) \tag{34}$$

associated to Problem (33), rather than $O_c : \mathbb{R}^d \to \mathbb{R}^d$ $c \mapsto x$ as the fixed point methods to solve LCP (18).

6.2 Merging PDAS and ProxNets: PDASNet

The structure of Algorithm 3 suggest that one iteration of the PDAS may be realized by a two-layer neural network with a discontinuous activation layer based on the *binary step unit* (BiSU) activation function $\varrho^{BS}(x) := \mathbb{1}_{\{x>0\}}$. This approach, however, is not desirable as discontinuous activation functions do not correspond to proximal operators, thus lack stability and entail difficulties in the training process of the network. Fortunately, we may circumvent this issue by approximation the BiSU by ReLU activation functions.

Lemma 6.4. *Let* $\varrho^{BS} : \mathbb{R} \to \{0, 1\}$, $x \mapsto \mathbb{1}_{\{x>0\}}$ *be the BiSU and let* $\varrho : \mathbb{R} \to [0, \infty)$, $x \mapsto \max(x, 0)$ *be the scalar ReLU as in Definition 5.1. For any* $\gamma > 0$ *define*

$$\varrho_\gamma : \mathbb{R} \to \mathbb{R}_{\geq 0}, \quad x \mapsto \varrho(x/\gamma) - \varrho(x/\gamma - 1). \tag{35}$$

It holds that $\varrho_\gamma(x) = \varrho^{BS}(x)$ *for any* $x \in \mathbb{R} \setminus (0, \gamma)$*.*

The proof of Lemma 6.4 is immediate, but this simple observation enables us to realize one iteration of the PDAS as ProxNet and achieve superlinear convergence. As the construction of this network slightly differs from the fixed-point architectures, we provide the proof directly for an augmented network as in setting (2). Hence, this allows us again to approximate the solution operator $O_{PDAS,c}$ from (34) by using c in (18) as input variable.

Proposition 6.5. *Let* $\beta, \gamma, \xi > 0$ *be fixed constants, let* $(\mathbf{A}, c) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d$ *be any LCP* (18) *such that* $\mathbf{A}$ *satisfies* (19)*,* $c_i \notin (-\gamma, 0)$ *and* $(\mathbf{A}c)_i \notin (-\beta, 0)$ *holds for all* $i \in \{1, \dots, d\}$*, and such that* $\|c\|_\infty, \|\mathbf{A}^{-1}c\|_\infty < \xi$*. Furthermore, let* $x^0, \mu^0 \in \mathbb{R}^d$ *and* $\eta \geq \gamma/\beta > 0$ *be such that* $\mu_i^0 - \eta x_i^0 \notin (0, \gamma)$ *holds for all* $i \in \{1, \dots, d\}$*, and such that* $\|x^0\|_\infty, \|\mu^0\|_\infty < \xi$*. Let* $k \in \mathbb{N}$*, and let* (x^k, μ^k) *be the associated* k*-th iterate of the PDAS (Algorithm 3) with parameter* $\eta \geq \gamma/\beta > 0$ *for the LCP* $(\mathbf{A}, c)$*.*

There is a two-layer ProxNet $\Psi_{PDAS} : \mathbb{R}^{3d} \to \mathbb{R}^{2d}$ as in (2) such that for any pair $(\mathbf{A}, c)$ and any initial value (x^0, μ^0) as above it holds that

$$\begin{pmatrix} x^{k+1} \\ \mu^{k+1} \end{pmatrix} = \Psi_{PDAS} \begin{pmatrix} x^k \\ \mu^k \\ c \end{pmatrix}, \quad k \in \mathbb{N}_0.$$

The weights and biases of Ψ_{PDAS} depend on γ, ξ, η and $\mathbf{A}$, but are independent of c.

Proof. For any $k \in \mathbb{N}_0$ and (x^k, μ^k), let $\mathcal{A}_k$ and $\mathcal{I}_k$ be the active and inactive set, respectively, as defined in Algorithm 3. Let $g^k \in \mathbb{R}^d$ be the binary vector given for $i \in \{1, \dots, d\}$ by

$$g_i^k := \varrho^{BS}(\mu_i^k - \eta x_i^k) = \begin{cases} 1, & i \in \mathcal{A}_k \\ 0, & i \in \mathcal{I}_k \end{cases}.$$

Moreover, let $\mathbf{G}_k := \operatorname{diag}(g^k)$ and note that the update in Algorithm 3 is the solution to the linear system

$$\begin{pmatrix} \mathbf{A} & -\mathbf{I}_d \\ \mathbf{G}_k & \mathbf{I}_d - \mathbf{G}_k \end{pmatrix} \begin{pmatrix} x^{k+1} \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}.$$

As $\mathbf{G}_k x^{k+1} = (\mathbf{I}_d - \mathbf{G}_k)\mu^{k+1} = 0$ and $\mathbf{A}$ is regular, this is equivalent to solving

$$\begin{pmatrix} \mathbf{A}(\mathbf{I}_d - \mathbf{G}_k) & -\mathbf{G}_k \\ \mathbf{G}_k & \mathbf{I}_d - \mathbf{G}_k \end{pmatrix} \begin{pmatrix} x^{k+1} \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}.$$

Observing that $\mathbf{G}_k^2 = \mathbf{G}_k$, $(\mathbf{I}_d - \mathbf{G}_k)^2 = \mathbf{I}_d - \mathbf{G}_k$, $\mathbf{G}_k(\mathbf{I}_d - \mathbf{G}_k) = \mathbf{0}_d$, where $\mathbf{0}_d \in \mathbb{R}^{d \times d}$ has only zero entries, together with $\mathbf{G}_k \mathbf{A} = \mathbf{A}^\top \mathbf{G}_k$ then yields

$$\begin{pmatrix} x^{k+1} \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} (\mathbf{I}_d - \mathbf{G}_k)\mathbf{A}^{-1} & \mathbf{G}_k \\ -\mathbf{G}_k & \mathbf{I}_d - \mathbf{G}_k \end{pmatrix} \begin{pmatrix} c \\ 0 \end{pmatrix} = \begin{pmatrix} (\mathbf{I}_d - \mathbf{G}_k)\mathbf{A}^{-1} c \\ -\mathbf{G}_k c \end{pmatrix}. \tag{36}$$

Since $\mu_i^0 - \eta x_i^0 \in \mathbb{R} \setminus (0, \gamma)$ holds by assumption for every $i \in \{1, \dots, d\}$, Lemma 6.4 yields that

$$\varrho_\gamma(\mu_i^0 - \eta x_i^0) = \varrho^{BS}(\mu_i^0 - \eta x_i^0),$$

and therefore

$$g^0 = \varrho^{(d)}((\mu^0 - \eta x^0)/\gamma) - \varrho^{(d)}((\mu^0 - \eta x^0)/\gamma - \mathbf{1}_d),$$

where $\varrho^{(d)} : \mathbb{R}^d \to \mathbb{R}^d$ is the component-wise ReLU activation from (20). We denote by $\varrho^{(4d)} : \mathbb{R}^{4d} \to \mathbb{R}^{4d}$ the corresponding ReLU in $\mathbb{R}^{4d}$ and define the ProxNet

$$\Psi_1 : \mathbb{R}^{3d} \to \mathbb{R}^{2d}, \quad \begin{pmatrix} x \\ \mu \\ c \end{pmatrix} \mapsto \widetilde{W}_2 \varrho^{(4d)} \left(W_1 \begin{pmatrix} x \\ \mu \\ c \end{pmatrix} + b_1 \right),$$

where

$$W_1 := \begin{pmatrix} -\frac{\eta}{\gamma}\mathbf{I}_d & \frac{1}{\gamma}\mathbf{I}_d & \mathbf{0}_d \\ -\frac{\eta}{\gamma}\mathbf{I}_d & \frac{1}{\gamma}\mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{0}_d & \mathbf{I}_d \\ \mathbf{0}_d & \mathbf{0}_d & -\mathbf{I}_d \end{pmatrix}, \quad b_1 := \begin{pmatrix} 0 \\ -\mathbf{1}_d \\ 0 \\ 0 \end{pmatrix}, \quad \widetilde{W}_2 := \begin{pmatrix} \mathbf{I}_d & -\mathbf{I}_d & \mathbf{0}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{0}_d & \mathbf{I}_d & -\mathbf{I}_d \end{pmatrix}. \tag{37}$$

It hence follows that

$$\Psi_1 \left(\begin{pmatrix} x^0 \\ \mu^0 \\ c \end{pmatrix} \right) = \begin{pmatrix} g^0 \\ c \end{pmatrix}.$$

Now recall that $\mathbf{G}_k := \operatorname{diag}(g^k)$, and note that for any $y \in \mathbb{R}^d$ such that $\|y\|_\infty \leq \xi$ it holds

$$\begin{aligned}\mathbf{G}_k y &= \operatorname{diag}(g^k)y = \max(y + \xi(2g^k - \mathbf{1}_d), 0) - \xi g^k \\ (\mathbf{I}_d - \mathbf{G}_k)y &= \operatorname{diag}(\mathbf{1}_d - g^k)y = \max(y + \xi(-2g^k + \mathbf{1}_d), 0) - \xi(\mathbf{1}_d - g^k).\end{aligned}$$

We define the ProxNet

$$\Psi_2 : \mathbb{R}^{2d} \to \mathbb{R}^{2d}, \quad \begin{pmatrix} g \\ c \end{pmatrix} \mapsto W_3 \varrho^{(3d)} \left(\widehat{W}_2 \begin{pmatrix} g \\ c \end{pmatrix} + b_2 \right) + b_3, \tag{38}$$

where $\varrho^{(3d)} : \mathbb{R}^{3d} \to \mathbb{R}^{3d}$ is the component-wise ReLU in $\mathbb{R}^{3d}$ and with

$$\widehat{W}_2 := \begin{pmatrix} -2\xi\mathbf{I}_d & \mathbf{A}^{-1} \\ 2\xi\mathbf{I}_d & -\mathbf{I}_d \\ \mathbf{I}_d & \mathbf{0}_d \end{pmatrix}, \; b_2 := \begin{pmatrix} \xi\mathbf{1}_d \\ -\xi\mathbf{1}_d \\ 0 \end{pmatrix}, \; W_3 := \begin{pmatrix} \mathbf{I}_d & \mathbf{0}_d & \xi\mathbf{I}_d \\ \mathbf{0}_d & \mathbf{I}_d & -\xi\mathbf{I}_d \end{pmatrix}, \; b_3 := \begin{pmatrix} -\xi\mathbf{1}_d \\ 0 \end{pmatrix}. \tag{39}$$

Equation (36) then shows together with $\varrho^{(d)}(g^0) = g^0$ that

$$\Psi_2 \left(\begin{pmatrix} g^0 \\ c \end{pmatrix} \right) = \begin{pmatrix} (\mathbf{I}_d - \mathbf{G}_0)\mathbf{A}^{-1}c \\ -\mathbf{G}_0 c \end{pmatrix} = \begin{pmatrix} x^1 \\ \mu^1 \end{pmatrix},$$

and since Ψ_1 and Ψ_2 have no skip connections, we may concatenate to obtain

$$\Psi_2 \bullet \Psi_1 \left(\begin{pmatrix} x^0 \\ \mu^0 \\ c \end{pmatrix} \right) = \begin{pmatrix} x^1 \\ \mu^1 \end{pmatrix}.$$

Note that the second linear transform in $\Psi_2 \bullet \Psi_1 : \mathbb{R}^{3d} \to \mathbb{R}^{2d}$ is given by the matrix $W_2 := \widehat{W}_2 \widetilde{W}_2 \in \mathbb{R}^{3d \times 4d}$ and the vector b_2 from Equations (37) and (39).

The claim follows inductively if we can ensure that $\mu_i^k - \eta x_i^k \notin (0, \gamma)$ holds for all $k \in \mathbb{N}_0$. By (36) we have for any $k \in \mathbb{N}_0$

$$\mu_i^{k+1} - \eta x_i^{k+1} = \left(\mathbf{G}_k(\eta\mathbf{A}^{-1} - \mathbf{I}_d)c - \eta\mathbf{A}^{-1}c \right)_i = \begin{cases} -c_i, & \text{if } g_i^k = 1, \\ -\eta(\mathbf{A}^{-1}c)_i, & \text{if } g_i^k = 0, \end{cases}$$

and, by assumption, it holds that $c_i \notin (-\gamma, 0)$, $(\mathbf{A}c)_i \notin (-\beta, 0)$. The claim now follows since $\eta \geq \frac{\gamma}{\beta} > 0$ and therefore $\mu_i^k - \eta x_i^k \notin (0, \gamma)$. □

The approximate solution operator built on the PDASNet is given as

$$\widetilde{O}_{PDAS,c} : \mathbb{R}^d \oplus \mathbb{R}^d \oplus \mathbb{R}^d \to \mathbb{R}^d \oplus \mathbb{R}^d, \quad (x, \mu, c) \mapsto \left[\Psi_{PDAS}(\cdot, \cdot, c) \bullet \cdots \bullet \Psi_{PDAS}(\cdot, \cdot, c) \right](x, \mu).$$

Combining Proposition 6.5 and Theorem 6.1 show that $\widetilde{O}_{PDAS,c}(x^0, \mu^0, c) \to O_{PDAS,c}(c)$ for suitable (x^0, μ^0) and any c at superlinear rate.

Theorem 6.6. *Let $\beta, \gamma, \xi > 0$ be fixed constants, let $(\mathbf{A}, c) \in \mathbb{R}^{d \times d} \times \mathbb{R}^d$ be any LCP (18) such that $\mathbf{A}$ satisfies (19), $c_i \notin (-\gamma, 0)$ and $(\mathbf{A}c)_i \notin (-\beta, 0)$ holds for all $i \in \{1, \dots, d\}$, and such that $\|c\|_\infty, \|\mathbf{A}^{-1}c\|_\infty < \xi$. Furthermore, let $x^0, \mu^0 \in \mathbb{R}^d$ and $\eta \geq \gamma/\beta > 0$ be such that $\mu_i^0 - \eta x_i^0 \notin (0, \gamma)$ holds for all $i \in \{1, \dots, d\}$, and such that $\|x^0\|_\infty, \|\mu^0\|_\infty < \xi$. Let Ψ_{PDAS} be as in Proposition 6.5, so that (x^k, μ^k) is the associated k-th iterate generated by Ψ_{PDAS} for any $k \in \mathbb{N}$, and let (x, μ) be the unique solution to (32).*

Then, there is a decreasing function $\mathrm{k} : (0, 1) \to \mathbb{N}$, depending only on $\mathbf{A}$ and ξ, such that for any $\varepsilon \in (0, 1)$, and any x^0, μ^0, c as above such that $\|x^0 - x\|_2 + \|\mu^0 - \mu\|_2$ is sufficiently small it holds

$$\|x - x^{\mathrm{k}(\varepsilon)}\|_2 + \|\mu - \mu^{\mathrm{k}(\varepsilon)}\|_2 \leq \varepsilon \quad \textit{and} \quad \lim_{\varepsilon \to 0} \frac{\mathrm{k}(\varepsilon)}{\log(1/\varepsilon)} = 0.$$

Proof. Proposition 6.5 shows that the PDAS with initial values x^0 and μ^0 is realized by the sequence $((x^k, \mu^k), k \in \mathbb{N})$ under the given Assumptions. By Theorem 6.1, $(x^k, \mu^k), k \in \mathbb{N})$ converges to (x, μ) for a sufficiently close initial guess (x^0, μ^0), hence for any $\varepsilon > 0$, there is a $k_\varepsilon \in \mathbb{N}$ such that

$$\|x - x^{k_\varepsilon}\|_2 + \|\mu - \mu^{k_\varepsilon}\|_2 \leq \varepsilon.$$

It remains to prove the asymptotic behavior of k_ε. By Theorem 6.1, there is a nonnegative sequence $(\widetilde{C}_k, k \in \mathbb{N})$ decreasing to zero and an integer $k_0 \in \mathbb{N}_0$, such that for any $k > k_0$

$$\|x - x^k\|_2 + \|\mu - \mu^k\|_2 \leq \widetilde{C}_k \left(\|x - x^{k-1}\|_2 + \|\mu - \mu^{k-1}\|_2\right).$$

We iterate this estimate until k_0 to obtain

$$\|x - x^k\|_2 + \|\mu - \mu^k\|_2 \leq \left(\prod_{l=k_0+1}^{k} \widetilde{C}_l\right) \left(\|x - x^{k_0}\|_2 + \|\mu - \mu^{k_0}\|_2\right), \tag{40}$$

and note that

$$\|x - x^{k_0}\|_2 + \|\mu - \mu^{k_0}\|_2 \leq \|x - x^0\|_2 + \|\mu - \mu^0\|_2 =: C_0.$$

Note that from $(\mathbf{A}x - c)^\top x = 0$ and $\mu = \mathbf{A}x - c$ it follows with (19) that $\|x\|_2 \leq \xi/C_-$ and $\|\mu\|_2 \leq \xi(C_+/C_- + 1)$. Hence, $C_0 < \infty$ is bounded uniformly with respect to x^0, μ^0 and c. We may assume without loss of generality that $(\widetilde{C}_k, k \in \mathbb{N}_0)$ is monotone decreasing and $\widetilde{C}_k \leq 1$ for all any $k \in \mathbb{N}$. For any given $\varepsilon > 0$ this yields

$$\varepsilon \overset{!}{\geq} \left(\prod_{l=k_0+1}^{k} \widetilde{C}_l\right) C_0 \quad \Longleftrightarrow \quad \log(\varepsilon) - \log(C_0) \geq \sum_{l=k_0+1}^{k} \log(\widetilde{C}_l) \geq k \log(\widetilde{C}_k).$$

Now define

$$\mathrm{k} : (0,1) \to \mathbb{N}, \quad \varepsilon \mapsto \min\{k \in \mathbb{N} : k \log(\widetilde{C}_k) \leq \log(\varepsilon) - \log(C_0)\}.$$

As $k \log(\widetilde{C}_k) \to -\infty$ for $k \to \infty$, k is well-defined and decreasing on $(0,1)$. Moreover, for $\varepsilon \to 0$ it follows that $\mathrm{k}(\varepsilon) \to \infty$ and hence $\widetilde{C}_{\mathrm{k}(\varepsilon)} \to 0$. This yields

$$0 \leq \lim_{\varepsilon \to 0} \frac{\mathrm{k}(\varepsilon)}{\log(1/\varepsilon)} = \lim_{\varepsilon \to 0} \frac{\mathrm{k}(\varepsilon) \log(\widetilde{C}_{\mathrm{k}(\varepsilon)})}{\log(1/\varepsilon) \log(\widetilde{C}_{\mathrm{k}(\varepsilon)})} \leq \lim_{\varepsilon \to 0} \frac{\log(\varepsilon) - \log(C_0)}{\log(1/\varepsilon) \log(\widetilde{C}_{\mathrm{k}(\varepsilon)})} = 0.$$

□

7 Numerical Experiments – Valuation of American Options

7.1 Black-Scholes Model

To illustrate an application for ProxNets, we consider the valuation of an American option in the Black-Scholes model. The associated payoff function of the American option is denoted by $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and we assume a time horizon $\mathbb{T} = [0, T]$ for $T > 0$. In any time $t \in \mathbb{T}$ and for any spot price $x_0 \geq 0$ of the underlying stock, the value of the option is denoted by $V(t, x)$ and defines a mapping $V : \mathbb{T} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Changing to time-to-maturity and log-price yields the map $v : \mathbb{T} \times \mathbb{R} \to \mathbb{R}_{\geq 0},\ (t, x) \mapsto V(T - t, e^x)$, which is the solution to the free boundary value problem

$$\begin{aligned}
\partial_t v - \frac{\sigma^2}{2} \partial_{xx} v - \left(r - \frac{\sigma^2}{2}\right) \partial_x v + rv &\geq 0 && \text{in } (0, T] \times \mathbb{R}, \\
v(t, x) &\geq g(e^x) && \text{in } (0, T] \times \mathbb{R}, \\
\left(\partial_t v - \frac{\sigma^2}{2} \partial_{xx} v - \left(r - \frac{\sigma^2}{2}\right) \partial_x v + rv\right)(g - v) &= 0 && \text{in } (0, T] \times \mathbb{R}, \\
v(0, e^x) &= g(e^x) && \text{in } \mathbb{R},
\end{aligned} \tag{41}$$

see, e.g., [14, Chapter 5.1]. The parameters $\sigma > 0$ and $r \in \mathbb{R}$ are the volatility of the underlying stock and the interest rate, respectively. We assume that $g \in H^1(\mathbb{R}_{\geq 0})$ and construct in the following a ProxNet-approximation to the *payoff-to-solution operator* at time $t \in \mathbb{T}$ given by

$$O_{g,t} : H^1(\mathbb{R}_{\geq 0}) \to H^1(\mathbb{R}), \quad g \mapsto v(t, \cdot). \tag{42}$$

As V and v, and therefore O_g, are in general not known in closed-form, a common approach to approximate v for a given payoff function g is to restrict Problem (41) to a bounded domain $\mathcal{D} \subset \mathbb{R}$ and to discretize $\mathcal{D}$ by linear finite elements based on d equidistant nodal points. The payoff function is interpolated with respect to the nodal basis and we collect the respective interpolation coefficients of g in the vector $\underline{g} \in \mathbb{R}^d$. The time domain $[0, T]$ is split by $M \in \mathbb{N}$ equidistant time steps and step size $\Delta t = T/M$, the temporal derivative is approximated by a backward Euler approach. This space-time discretization of the free boundary problem (41) leads to a sequence of discrete variational inequalities: Given $\underline{g} \in \mathbb{R}^d$ and $u_0 := 0 \in \mathbb{R}^d$ find $u_m \in \mathbb{R}^d$ such that for $m \in \{1, \dots, M\}$ it holds

$$\mathbf{A}u_{m+1} \geq F_m, \quad u_{m+1} \geq 0, \quad (\mathbf{A}u_{m+1} - F_m)^\top u_{m+1} = 0. \tag{43}$$

The LCP (43) is defined by the matrices $\mathbf{A} := \mathbf{M} + \Delta t \mathbf{A}^{BS} \in \mathbb{R}^{d \times d}$, $\mathbf{A}^{BS} := \frac{\sigma^2}{2}\mathbf{S} + (\frac{\sigma^2}{2} - r)\mathbf{B} + r\mathbf{M} \in \mathbb{R}^{d \times d}$ and right hand side $F_m := -\Delta t(\mathbf{A}^{BS})^\top \underline{g} + \mathbf{M}u_m \in \mathbb{R}^d$. The matrices $\mathbf{S}, \mathbf{B}, \mathbf{M} \in \mathbb{R}^{d \times d}$ represent the finite element stiffness, advection and mass matrices, hence $\mathbf{A}$ is tri-diagonal and asymmetric if $\frac{\sigma^2}{2} \neq r$. The true value of the options at time km is approximated at the nodal points via $v(\Delta t m, \cdot) \approx u_m + \underline{g}$. This yields the *discrete payoff-to-solution operator* at time $\Delta t m$ defined by

$$\overline{O}_{g,\Delta t m} : \mathbb{R}^d \mapsto \mathbb{R}^d, \quad \underline{g} \mapsto u_m + \underline{g}, \qquad m \in \{1, \dots, M\}. \tag{44}$$

Problem (43) may be solved for all m using a shallow ProxNet

$$\Phi : \mathbb{R}^d \oplus \mathbb{R}^d \oplus \mathbb{R}^d \to \mathbb{R}^d, \quad x \mapsto R(W_1 x + b_1),$$

with ReLU-activation $R = \varrho^{(d)} : \mathbb{R}^d \to \mathbb{R}^d$. The architecture of Φ allows to take $\underline{g}$ and u_m as additional inputs in each step, therefore *we train only one shallow ProxNet* that may be used for any payoff function g and every time horizon $\mathbb{T}$. Therefore, we learn the payoff-to-solution operator O_g associated to Problem (41) by concatenating Φ. The parameters $W_1 \in \mathbb{R}^{d \times 3d}$ and $b_1 \in \mathbb{R}^d$ are learned in the training process and shall emulate one step of the PJOR Algorithm 1, as well as the linear transformation $(\underline{g}, u_m) \mapsto F_m$ to obtain the the right hand side in (43). Therefore, a total of $3d^2 + d$ parameters have to be learned in each example.

For our experiments we use the Python-based machine learning package PyTorch[1]. All experiments are run on a notebook with 8 CPUs, each with 1.80 GHz, and 16 GB memory. To train Φ, we sample $N_s \in \mathbb{N}$ input data points $x^{(i)} := (x_0^{(i)}, \underline{g}^{(i)}, u^{(i)}) \in \mathbb{R}^{3d}$, $i \in \{1, \dots, N_s\}$, from a $3d$-dimensional standard-normal distribution. The output-training data samples $y^{(i)}$ consist of one iteration of Algorithm 1 with $\omega = 1$, initial value $x^0 := x_0^{(i)}$, with $\mathbf{A}$ as in (43) and right hand side given by $c := -\Delta t(\mathbf{A}^{BS})^\top \underline{g}^{(i)} + \mathbf{M}u^{(i)} \in \mathbb{R}^d$. We draw a total of $N_s = 2 \cdot 10^4$ input-output samples, use half of the data for training, and the other half for validation. In the training process, we use mini-batches of size $N_{batch} = 100$ and the *Adam Optimizer* [18] with initial learning rate 10^{-3}, which is reduced by 50% every 20 epochs. As error criterion we use the mean-squared error (MSE) loss function, which is for each batch of inputs $((x^{(i_j)}, \underline{g}^{(i_j)}, u^{(i_j)}), j = 1, \dots, N_{batch})$ and outputs $(y^{(i_j)}, j = 1, \dots, N_{batch})$ given by

$$\begin{aligned} &Loss\left((x^{(i_1)}, \underline{g}^{(i_1)}, u^{(i_1)}), \dots, (x^{(i_{N_{batch}})}, \underline{g}^{(i_{N_{batch}})}, u^{(i_{N_{batch}})})\right) \\ &:= \frac{1}{N_{batch}} \sum_{j=1}^{N_{batch}} \|\Phi(x^{(i_j)}, \underline{g}^{(i_j)}, u^{(i_j)}) - y^{(i_j)}\|_2^2. \end{aligned}$$

[1] https://pytorch.org/

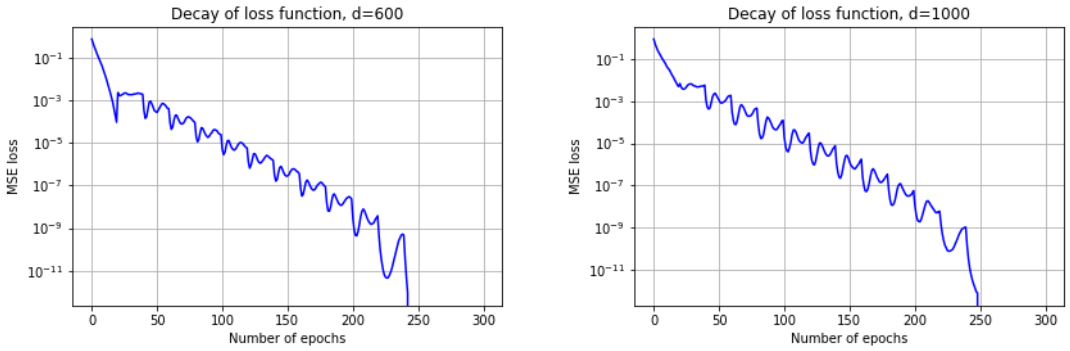

Figure 1: Decay of the loss function for $d = 600$ (left) and $d = 1000$ (right). In all of our experiments the training loss falls below the threshold of 10^{-12} before the 250-th epoch, and training is stopped early.

d	200	400	600	800	1000
training time in sec.	19.80	83.71	182.42	330.74	519.95
err_{val}	$9.88 \cdot 10^{-7}$	$9.79 \cdot 10^{-7}$	$7.29 \cdot 10^{-7}$	$1.29 \cdot 10^{-6}$	$1.40 \cdot 10^{-6}$

Table 1: Training times and validation errors for the ProxNets in the Black-Scholes model in several dimensions, as estimated in (45) based on $N_{val} = 10^4$ samples. The relative error remains stable with increasing problem dimension.

We stop the training process if the loss function falls below the tolerance 10^{-12} or after a maximum of 300 epochs. The number of spatial nodal points d that determines the size of the matrix LCPs are varied throughout our experiments in $d \in \{200, 400, \dots, 1000\}$. We choose the Black-Scholes parameters $\sigma = 0.1$, $r = 0.01$ and $T = 1$. Spatial and temporal refinement are balanced by using $M = d$ time steps of size $\Delta t = T/M = 1/d$. The decay of the loss-curves throughout in dimension d is depicted in Figure 1. The reduction of the learning rate every 20 epochs explains the characteristic "steps" in the decay. This stabilizes the training procedure, and we reached a loss of $\mathcal{O}(10^{-12})$ for each d before the 250-th epoch. Once training is terminated, we compress the resulting weight matrix of the trained single-layer ProxNet by setting all entries with absolute value lower than 10^{-8} to zero. This speeds up evaluation of the trained network, while the resulting error is negligible. As the matrix W_1 in the trained ProxNet is close to the "true" tri-diagonal matrix $\mathbf{A}$ from (43), this procedure eliminates most of the ProxNet's $\mathcal{O}(d^2)$ parameters, and only $\mathcal{O}(d)$ non-trivial entries remain.

The relative validation error is estimated based on the $N_{val} := 10^4$ validation samples via

$$err_{val}^2 := \frac{\sum_{j=1}^{N_{val}} \|\Phi(x^{(i_j)}, \underline{g}^{(i_j)}, u^{(i_j)}) - y^{(i_j)}\|_2^2}{\sum_{j=1}^{N_{val}} \|y^{(i_j)}\|_2^2}. \tag{45}$$

The validation errors and training times for each dimension are found in Table 1, and confirm the successful training of the ProxNet. Naturally, training time increases in d, while the validation error is small of order $\mathcal{O}(10^{-6})$ for all d.

To test the trained neural networks on Problem (43) for the valuation of an American option, we consider a basket of 20 put options with payoff function $g_i(x) := \max(K_i - x, 0)$, and strikes $K_i = 10 + 90\frac{i}{20}$ for $i \in \{1, \dots, 20\}$. Hence, we use the same ProxNet for 20 different payoff vectors $\underline{g}_i$. Note that *we did not train our networks on payoff functions*, but on random samples, and thus we could in principle consider an arbitrary basket containing different types of payoffs. The restriction to put options is for the sake of brevity only. We denote by $u_{m,i}$ for $m \in \{0, \dots, M\}$ the sequence of solutions to (43) with payoff vector $\underline{g}_i$ and $u_{0,i} = 0 \in \mathbb{R}^d$ for each i.

Concatenating Φ k times yields an approximation to the discrete operator in (44) for any

d	200	400	600	800	1000
err_{rel}	$1.84 \cdot 10^{-4}$	$7.43 \cdot 10^{-4}$	$1.48 \cdot 10^{-3}$	$2.35 \cdot 10^{-3}$	$3.80 \cdot 10^{-3}$
time ProxNet in sec.	0.32	1.93	5.85	15.06	34.07
time reference in sec.	1.88	6.40	23.86	66.41	134.44

Table 2: Relative errors and computational times of a ProxNet solver for a basket of American put options in the Black-Scholes model. ProxNets significantly reduce computational time, while their relative error remains sufficiently small for all d.

$m \in \{1, \dots, M\}$ via

$$\widetilde{O}_{g,\Delta tm} : \mathbb{R}^d \oplus \mathbb{R}^d \oplus \mathbb{R}^d \to \mathbb{R}^d, \quad (x, \widetilde{u}_m, \underline{g}) \mapsto \left[\underbrace{\Phi(\cdot, \underline{g}, \widetilde{u}_m) \bullet \cdots \bullet \Phi(\cdot, \underline{g}, \widetilde{u}_m)}_{k\text{-fold concatenation}} \right](x).$$

An approximating sequence of $(u_{m,i}, m \in \{0, \dots, M\})$ is then in turn generated by

$$\widetilde{u}_{m+1,i} := \widetilde{O}_{g,km}(\widetilde{u}_{m,i}, \widetilde{u}_{m,i}, \underline{g}), \quad \widetilde{u}_{0,i} := u_{0,i} = 0 \in \mathbb{R}^d.$$

That is, $\widetilde{u}_{m+1,i}$ is given by iterating Φ k times with initial input $x^0 = \widetilde{u}_{m,i} \in \mathbb{R}^d$ and fixed inputs and $\underline{g}_i$ and $\widetilde{u}_{m,i}$. We stop for each m after k iterations if two subsequent iterates x^k and x^{k-1} satisfy $\|x^k - x^{k-1}\|_2 < 10^{-3}$.

The reference solution $u_{M,i}$ is calculated by a Python-implementation that uses the PDAS Algorithm 3 to solve (43) with tolerance $\varepsilon = 10^{-6}$ in every time step. Compared to a fixed-point iteration, the standard PDAS implementation converges (locally) superlinear according to Theorem 6.1, but has to be called seperately for each payoff function g_i. In contrast, the ProxNet Φ may be iterated for the entire batch of 20 payoffs at once in PyTorch. We measure the relative error

$$err_{i,rel} := \|\widetilde{u}_{M,i} - u_{M,i}\|_2 / \|u_{M,i}\|_2$$

for each payoff vector $\underline{g}_i$ at the end point $T = \Delta t M = 1$ and report the sample mean error

$$err_{rel} := \frac{1}{20} \sum_{i=1}^{20} err_{i,rel}. \tag{46}$$

Sample mean errors and computational times are depicted for $d \in \{200, 400, \dots, 1000\}$ in Table 2. The results clearly show that ProxNets significantly accelerate the valuation of American option baskets, if compared to the standard, PDAS-based implementation. This holds true for any spatial resolution, i.e., the number of grid points d, while the relative error is small of magnitude $\mathcal{O}(10^{-3})$ or $\mathcal{O}(10^{-4})$. We observe that computational times scale similarly for both, ProxNet and reference solution, in d. Hence, in our experiments, ProxNets are computationally advantageous even for a very fine resolution of $d = 1000$ nodal points.

7.2 Jump-Diffusion Model

We generalize the setting of the previous subsection from the Black-Scholes market to an *exponential Lévy model*. That is, the log-price of the stock evolves as a Lévy process, with jumps distributed with respect to the Lévy measure $\nu : \mathcal{B}(\mathbb{R}) \to [0, \infty)$. The option value v (in log-price

d	200	400	600	800	1000
training time in sec.	23.92	80.79	182.86	332.02	515.56
err_{val}	$1.14 \cdot 10^{-6}$	$1.09 \cdot 10^{-6}$	$8.78 \cdot 10^{-7}$	$1.12 \cdot 10^{-6}$	$1.38 \cdot 10^{-6}$

Table 3: Training times and validation errors for the ProxNets in the jump-diffusion model, as estimated in (45) based on $N_{val} = 10^4$ samples. The relative error remains stable with increasing problem dimension.

and time-to-maturity) is now the solution of a *partial integro-differential inequality* given by

$$\begin{aligned} \partial_t v - \frac{\sigma^2}{2}\partial_{xx} v - \gamma \partial_x v + \int_{\mathbb{R}} v(\cdot + z) - v - \partial_x v \nu(dz) + rv \geq 0 \quad & \text{in } (0,T] \times \mathbb{R}, \\ v(t,x) \geq g(e^x) \quad & \text{in } (0,T] \times \mathbb{R}, \\ \left(\partial_t v - \frac{\sigma^2}{2}\partial_{xx} v - \gamma \partial_x v + \int_{\mathbb{R}} v(\cdot + z) - v - \partial_x v \nu(dz) + rv \right)(g - v) = 0 \quad & \text{in } (0,T] \times \mathbb{R}, \\ v(0, e^x) = g(e^x) \quad & \text{in } \mathbb{R}. \end{aligned} \tag{47}$$

Introducing jumps in the model hence adds a non-local integral term to Equation (41). The drift is set to $\gamma := -\sigma^2/2 - \int_{\mathbb{R}}(e^z - 1 - z)\nu(dz) \in \mathbb{R}$ in order to eliminate arbitrage in the market. We discretize Problem (47) by an equidistant grid in space and time as in the previous subsection, for details, e.g., integration with respect to ν, we refer to [14, Chapter 10]. The space-time approximation yields again a sequence of LCPs of the form

$$\mathbf{A}^L u_{m+1} \geq F_m, \quad u_{m+1} \geq 0, \quad (\mathbf{A}^L u_{m+1} - F_m)^\top u_{m+1} = 0, \tag{48}$$

where $\mathbf{A}^L := \mathbf{M} + \Delta t \mathbf{A}^{Levy} \in \mathbb{R}^{d \times d}$ with $\mathbf{A}^{Levy} := \frac{\sigma^2}{2}\mathbf{S} + \mathbf{A}^J$, and where the matrix $\mathbf{A}^J$ stems from the integration of ν. A crucial difference to (43) is that $\mathbf{A}^L$ is not anymore tri-diagonal, but a *dense matrix*, due to the non-local integral term caused by the jumps. Moreover, $\mathbf{A}^L$ does not necessarily satisfy the assumptions for global convergence of the PDAS in Theorem 6.2, which has to be taken into account when calculating the reference solutions. The drift γ and interest rate r are transformed into the right hand side, such that $F_m := -\Delta t (\mathbf{A}^{Levy})^\top \underline{g}_m + \mathbf{M} u_m \in \mathbb{R}^d$, where $\underline{g}_m$ is the nodal interpolation of the transformed payoff $g_m(x) := g e^{rkm}(x - (\gamma + r)km)$. The inverse transformation gives an approximation to the solution v of (47) at the nodal points via $v(km, \cdot - (\gamma + r)T) \approx e^{-rT} u_M$. We refer to [14, Chapter 10.6] for further details on the discretization of American options in Lévy models.

The jumps are distributed according to the Lévy measure

$$\nu(dz) = \lambda p \beta_+ e^{-\beta_+ z} \mathbb{1}_{\{z>0\}}(z) + \lambda(1-p)\beta_- e^{-\beta_- z} \mathbb{1}_{\{z<0\}}(z), \quad z \in \mathbb{R}. \tag{49}$$

That is, the jumps follow an asymmetric, double-sided exponential distribution with jump intensity $\lambda = \nu(\mathbb{R}) \in (0, \infty)$. We choose the parameters $p = 0.7$, $\beta_+ = 25$, $\beta_- = 20$ to characterize the tails of ν and set jump intensity to $\lambda = 1$. We further use $\sigma = 0.1$ and $r = 0.01$ as in the Black-Scholes example.

We use the same training procedure and parameters as in the previous subsection to train the shallow ProxNets. Training times and validation errors are depicted in Table 3, and indicate again a successful training. The decay of the training loss is for each d very similar to Figure 1, and training is again stopped in each case before the 300-th epoch.

After training, we again concatenate the shallow nets to approximate the operator $O_{g,t}$ in (42), that maps the payoff function g to the corresponding option value $v(t, \cdot)$ at any (discrete) point in time. We repeat the test from Subsection 7.1 in the jump-diffusion model with the identical basket of put options to test the trained ProxNets. The reference solution is again computed by a standard, PDAS-based implementation. The results for American options in the jump-diffusion model are depicted in Table 4. Again, we see that the trained ProxNets approximated the solution v to (47) for any g to an error of magnitude $\mathcal{O}(10^{-3})$ or less. While keeping the relative error small, ProxNets again significantly reduce computational time, and are therefore a valid alternative even in more involved financial market models.

d	200	400	600	800	1000
err_{rel}	$9.76 \cdot 10^{-5}$	$4.96 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$	$1.61 \cdot 10^{-3}$	$2.13 \cdot 10^{-3}$
time ProxNet in sec.	0.29	1.64	6.71	12.89	27.81
time reference in sec.	1.90	6.99	27.05	72.72	160.18

Table 4: Relative errors and computational times of a ProxNet solver for a basket of American put options in the jump-diffusion model. ProxNets significantly reduce computational time, while their relative error remains sufficiently small for all d.

8 Conclusions

We proposed deep neural networks which realize approximate input-to-solution operators for unilateral, inequality problems in separable Hilbert spaces. Their construction was based on realizing approximate solution constructions in the continuous (infinite dimensional) setting, via proximinal and contractive maps. As particular cases, several classes of finite-dimensional projection maps (PSOR, PJOR, primal-dual active set strategies) were shown to be representable by the proposed DNN architectures, ProxNet and PDASNet. The general construction principle behind ProxNet and PDASNet introduced in the present paper can be employed to realize further DNN architectures, also in more general settings. We refer to [1] for multilevel and multigrid methods to solve (discretized) variational inequality problems. The algorithms in this reference may also be realized as concatenation of ProxNets, similarly to the PJOR-Net and PSOR-Net from Examples 5.3 and 5.4. However, we leave the further analysis and representation of multigrid methods as ProxNets for future research.

Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

[1] L. Badea. Convergence rate of some hybrid multigrid methods for variational inequalities. *J. Numer. Math.*, 23(3):195–210, 2015.

[2] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, Cham, second edition, 2017. With a foreword by Hédy Attouch.

[3] S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *JMLR*, 20(74), 2019.

[4] J. M. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization*, volume 3 of *CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC*. Springer, New York, second edition, 2006. Theory and examples.

[5] P. L. Combettes and J.-C. Pesquet. Deep neural network structures solving variational inequalities. *Set-Valued Var. Anal.*, 28(3):491–518, 2020.

[6] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.

[7] G. Duvaut and J.-L. Lions. *Inequalities in mechanics and physics*, volume 219 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, Berlin-New York, 1976. Translated from the French by C. W. John.

[8] S. Glas and K. Urban. On noncoercive variational inequalities. *SIAM Journal on Numerical Analysis*, 52(5):2250–2271, 2014.

[9] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *International Conference on Machine Learning*, pages 1–8. PMLR, 2010.

[10] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl. Parseval proximal neural networks. *J. Fourier Anal. Appl.*, 26(4):Paper No. 59, 31, 2020.

[11] J. He and J. Xu. MgNet: a unified framework of multigrid and convolutional neural network. *Sci. China Math.*, 62(7):1331–1354, 2019.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[14] N. Hilber, O. Reichmann, C. Schwab, and C. Winter. *Computational methods for quantitative finance: Finite element methods for derivative pricing*. Springer Science & Business Media, 2013.

[15] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM J. Optim.*, 13(3):865–888 (2003), 2002.

[16] K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

[17] D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and their applications*, volume 31 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1980 original.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces, 2021.

[20] D. Lamberton and B. Lapeyre. *Introduction to stochastic calculus applied to finance*. Chapman & Hall/CRC Financial Mathematics Series. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2008.

[21] L. Lu, P. Jin, and G. E. Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, 2020.

[22] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

[23] K. G. Murty. On the number of solutions to the complementarity problem and spanning properties of complementary cones. *Linear Algebra and its Applications*, 5(1):65–108, 1972.

[24] J. A. A. Opschoor, C. Schwab, and J. Zech. Exponential ReLU DNN expression of holomorphic maps in high dimension. *Constructive Approximation*, 2019. Report SAM 2019-35 (revised).

[25] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numer.*, 8:143–195, 1999.

[26] B. Wohlmuth. Variationally consistent discretization schemes and numerical algorithms for contact problems. *Acta Numer.*, 20:569–734, 2011.

[27] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.