

# Multigrid method for nonsmooth problems

Torsten Bosse, ANL

Received: October 6, 2015/ Accepted: date

## Abstract

Multigrid methods have been shown to be an efficient tool for solving partial differential equations. In this paper, the idea of a multigrid method for nonsmooth problems is presented based on techniques from piecewise linear differentiation. In detail, the original nonsmooth problem is approximated by a sequence of piecewise linear models, which can be written in abs-normal form by using additional switching variables. In certain cases, one can exploit the structure of the piecewise linearization and formulate an efficient modulus fixed-point iteration for these switching variables. Moreover, using the idea of multigrid methods, one can find a solution of the modulus fixed-point equation for the switching variables on a coarse discretization, which then serves as an initial guess for the next finer level. Here, the important aspect is the right choice for the prolongation operator in order to avoid undesirable smoothing effects as it will be shown. Numerical results indicate (almost) mesh-independent behavior of the resulting method if done in the right way.

## 1 Introduction

The standard prerequisite of algorithmic differentiation and numerical optimization in finite dimension is that functions  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  consists of several components (see e.g. [1, 22]) and can be represented by an evaluation procedure [13] that can be thought of as a sequence of intermediate assignments from a library of elemental functions such as  $\pm$ ,  $\sqrt{\cdot}$ ,  $\sin$ , and  $\cos$  on some intermediate variables. Assuming that the components of this evaluation procedure are sufficiently smooth for the considered input variables

$u$ , one can derive extended evaluation procedures [12, 13] that yield derivative information such as Jacobi-vector products, vector-Jacobi products, the Jacobian itself or even higher-order derivatives of  $F$ . The sensitivity information is usually used to form a linear or quadratic model of the original nonlinear problem, for example, to find a root of  $F(u) = 0$ , if  $m = n$ , or some other optimization tasks such as  $\min_{u \in \mathbb{R}^n} F(u)$ , if  $m = 1$ . The latter models then serve in a Newton, steepest-descent, trust-region, or some other method [23, 7] to define a search direction in order to find a better approximation of the real solution  $u^*$  for the original problem. Several of these algorithms have been successfully adopted in the function space setting [10, 16, 26], where  $u$  is not a real vector but a function in some adequate space and  $\mathcal{F}$  denotes an operator acting on this function.

In many cases, however, it is inadequate to use a smooth model to approximate  $\mathcal{F}$ , for example if  $\mathcal{F}$  is nonsmooth. Therefore, we focus in this paper on an efficient solution of the problem  $F(u) = 0$  for a nonsmooth operator  $F : \mathcal{U} \rightarrow \mathcal{U}$  that, for example, arises by reformulating (non-)linear complementarity problems using the minimap function [8, 21]. To do so, we extend several results from piecewise linear differentiation in finite dimensions [12, 18, 24] to the function space setting and only focus on operators that are a finite composition of smooth intermediate expressions and the absolute value. This approach allows us to formulate a more sophisticated piecewise linearization  $\mathcal{F}'(u; \delta u)$  of the original function  $\mathcal{F}$  at  $u$  for some increment  $\delta u$  that reflects the underlying problem structure more accurately by taking into account the nonsmoothness of the original problem.

The piecewise linear models  $\mathcal{G}_u(\delta u) := \mathcal{F}(u) + \mathcal{F}'(u; \delta u)$  are stated in terms of an “infinite” dimensional abs-normal form (ANF) using some additional switching variables  $\mathbf{z}$ , which was already introduced and investigated in [12, 19, 24]. The presented definitions are similar to the original ones from the finite dimensional setting [25] except that the variables  $v, u, z \dots$  are now functions over some appropriate domain instead of real scalars and vectors. Analogously, the quantities  $Z, J, L \dots$  in the abs-normal form and the products  $Zu, Ju, \dots$ , which used to be matrices and matrix-vector products, need to be understood as operators and their application on a function.

A solution of the piecewise linear equation  $\mathcal{G}_u(\delta u) = 0$  then yields a generalized Newton step, which either solves  $0 = \mathcal{F}(u + \delta u)$  in the piecewise linear case or serves as a search direction in the nonlinear scenario. Obviously, solving the piecewise linear equation is in general much harder than a simple linear solve that would be required for a simple linearization with

some active set method. However, we believe that the presented approach is much more intuitive since, for example, as a result we find that a nonlinear complementarity problem should be approximated by linear complementarity problems.

Depending on the structure these (piecewise) linearizations can still be solved efficiently. To this end, we propose a multigrid method for the solution of (discretized) piecewise linear equations in abs-normal form. The basic idea follows a successive refinement strategy, where the discretized ANF is iteratively solved by using the (approximate) solution of the additional switching variables  $\mathbf{z}$  on a coarser grid as an initial guess for the finer grid. The convergence of the resulting methods is mainly inherited by the convergence properties of the considered iterative scheme, for example, the proposed modulus fixed-point iteration [25]. An important aspect of this approach is the right choice for the prolongation operator to avoid undesirable smoothing effects in the refinement step of  $\mathbf{z}$  that is usually nonsmooth and discontinuous.

For simplicity, most of the results are exemplified only for the modulus fixed-point iteration on a simply switched complementarity problem, i.e. an operator whose evaluation procedure does not contain nested abs-value evaluations, although more complicated scenarios are possible. With this specific iteration, however, the presented approach is only partly applicable as shown for an extended complementarity problem at the end of Section 2.

The goal of the paper is not only to provide a “new” multigrid method, but also to inspire research on how piecewise linear algorithmic differentiation and its results can be used in a function space setting.

The paper is structured as follows: In Section 2, the basic ideas and results of piecewise linear differentiation are presented in the function space setting. This formalism can be used for a first-optimize-then-discretize approach to define a piecewise linear model in function space to approximate the original nonsmooth problem. The discretization of the latter approximation is efficiently solved by the multigrid/successive refinement strategy presented in Section 3. Some numerical experiments that validate this approach are given in Section 4. We conclude in Section 5 with a summary and suggestion for future work.

## 2 Piecewise differentiation (revised)

In the following, we extend the results from piecewise linear differentiation in the Euclidean space  $\mathbb{R}^n$  to infinite dimensional operators. We assume that  $(\mathcal{U}, \|\cdot\|_{\mathcal{U}})$  and  $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$  are real Hilbert spaces with a point-wise ordering and  $U$  is an open subset of  $\mathcal{U}$  with the basic operations extending the identity and absolute value function to the infinite dimensional scenario. Hence, we mainly follow standard notation and denote by  $\mathcal{I} : \mathcal{U} \rightarrow \mathcal{U}$  the identity operator that maps every element/function  $u \in U$  onto itself (i.e.  $\mathcal{I}(u) = u$ ) and we write  $\mathcal{L}(\mathcal{U}, \mathcal{W})$  for bounded linear operators from the space  $\mathcal{U}$  to  $\mathcal{W}$ . Furthermore, the projection operator  $\mathcal{I}_{\hat{u} < \check{u}} : \mathcal{U} \rightarrow \mathcal{U}$  refers to the operator<sup>1</sup> that point-wise satisfies

$$\mathcal{I}_{\hat{u}(\omega) < \check{u}(\omega)}(u(\omega)) = \begin{cases} u(\omega), & \text{wherever } \hat{u}(\omega) < \check{u}(\omega) \\ 0, & \text{else} \end{cases} \quad \forall \omega \in \Omega$$

for given functions  $\hat{u}, \check{u} \in \mathcal{U}$  over a domain  $\Omega$  such that the abs-operator  $\Sigma : \mathcal{U} \rightarrow \mathcal{U}$  applied on  $u$ ,

$$\Sigma(u) := \mathcal{I}_{u > 0}(u) - \mathcal{I}_{u \leq 0}(u),$$

yields the entry-wise absolute value function  $\Sigma(u) = (|u_1|, \dots, |u_n|)$  for all vectors  $u = (u_1, \dots, u_n) \in \mathcal{U}$  in the finite-dimensional case  $\mathcal{U} = \mathcal{W} = \mathbb{R}^n$ . Hence, we can abbreviate  $\Sigma(u) = |u|$  and use the corresponding symbols for all expressions that can be derived from these operators (e.g. min, max). In some cases, we also consider the modified abs-operator  $\Sigma_{\hat{u}}(u) : \mathcal{U} \times \mathcal{U} \rightarrow \mathcal{U}$  that is defined for a given  $\hat{u} \in \mathcal{U}$  by

$$\Sigma_{\hat{u}}(u) := \mathcal{I}_{\hat{u} > 0}(u) - \mathcal{I}_{\hat{u} \leq 0}(u).$$

The next definition is a generalization of the evaluation procedure for piecewise smooth functions [12, 25]. It uses the precedence relation  $j \prec i$  known from algorithmic differentiation [13], which indicates that an operator  $\mathcal{F}_i$  directly depends on the result of an operator  $\mathcal{F}_j$ . The key difference between the finite-dimensional and the following definition is that the range and the domain spaces of the intermediate operators  $\mathcal{F}_i : U_i \subseteq \mathcal{U}_i \rightarrow W_i \subseteq \mathcal{W}_i$  now need to be consistent; in other words, it is required that the subsets in the

---

<sup>1</sup> $\leq, >, \geq, \in,$  and  $=$  accordingly

range space  $(\mathcal{W}_j)_{j \prec i}$  of all operators  $\mathcal{F}_j$ , which directly precede the operator  $\mathcal{F}_i$ , must be able to be embedded in the open subset  $U_i$  of the domain space  $\mathcal{U}_i$  of  $\mathcal{F}_i$ .

**Definition 1.** Let  $(\mathcal{U}, \|\cdot\|_{\mathcal{U}})$  and  $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$  be real Hilbert spaces with a point-wise ordering, and let  $U$  be an open subset of  $\mathcal{U}$ . Then the operator  $\mathcal{F} : U \subset \mathcal{U} \rightarrow W \subset \mathcal{W}$  is called (abs-)decomposable if there are a finite number of point-wise ordered real Hilbert spaces  $\{(\mathcal{U}_i, \|\cdot\|_{\mathcal{U}_i}), (\mathcal{W}_i, \|\cdot\|_{\mathcal{W}_i})\}_{i=0}^k$  that are consistent and contain open subsets  $U_i \subset \mathcal{U}_i$  and  $W_i \subset \mathcal{W}_i$  with  $U \subseteq U_0$  and  $W_k \subseteq W$ , such that  $\mathcal{F}$  can equivalently be written by an infinite-dimensional evaluation procedure

$$v_i = \mathcal{F}_i(v_j)_{j \prec i}, \quad i = 0, \dots, k$$

of operators  $\mathcal{F}_i : \mathcal{U}_i \rightarrow \mathcal{W}_i$  that are either abs-operators  $\Sigma(u_i)$  with  $\mathcal{W}_i \subseteq \mathcal{U}_i$  or Frechet differentiable for any initial  $v_0 \in U$ . In other words, the result of applying the evaluation procedure<sup>2</sup> on any  $v_0 = u$  is identical to  $F$  applied on  $u \in U$ , namely,

$$\mathcal{F}(u) = v_k \in W \text{ for all } u \in U.$$

The triple  $(\{\mathcal{F}_i\}, \{\mathcal{U}_i\}, \{\mathcal{W}_i\})^{\prec_{\mathcal{F}}}$  is called (abs-)decomposition of  $\mathcal{F}$  for the precedence relation  $\prec_{\mathcal{F}}$  induced by  $\mathcal{F}$ .

The decomposition in Definition 1 is not unique, as can be seen in the next example, where  $\mathcal{F}$  can be decomposed in several other ways, for example, by switching the sign of the operator  $\mathcal{F}_2$  or exchanging  $\mathcal{F}_1$  and  $\mathcal{F}_3$ .

**Example 1.** Consider the linear complementarity problem for given  $f, \varphi \in \mathcal{H}_0^1(\Omega, \mathbb{R})$  to determine a function  $u \in \mathcal{U} := \mathcal{H}_0^1(\Omega, \mathbb{R})$  that satisfies

$$0 \leq u(x) - \varphi(x) \perp -\Delta u(x) + f(x) \geq 0 \quad \text{for almost all } x \in \Omega$$

over some domain  $\Omega \subset \mathbb{R}^n$ . The problem can be reformulated in terms of finding a root  $u^*$  with  $\mathcal{F}(u^*) = 0$  of the nonsmooth operator

$$\mathcal{F}(u) := u - \varphi - \Delta u + f - |u - \varphi + \Delta u - f|.$$

The operator  $\mathcal{F}(u) : \mathcal{U} \rightarrow \mathcal{W} := \mathcal{H}_0^1(\Omega, \mathbb{R})$  is decomposable since one can, for example, define the operations  $v_0 = F_0(u) := u$ ,  $v_1 = F_1(v_0) := v_0 -$

---

<sup>2</sup>For readability, we will usually omit the dependence of the  $v_i$ 's on their predecessors and drop the arguments if unambiguous

$\varphi - \Delta v_0 + f$ ,  $v_2 = F_2(v_0) := v_0 - \varphi + \Delta v_0 - f$ ,  $v_3 = F_3(v_0) := |v_2|$ , and  $v_4 = F_4(v_1, v_3) := v_1 - v_3$ . In detail, one can choose  $\mathcal{U}_i = \mathcal{W}_i = \mathcal{H}_0^1(\Omega, \mathbb{R})$  for all spaces except of  $\mathcal{U}_4$ , which has to be equal to  $\mathcal{H}_0^1(\Omega, \mathbb{R}) \times \mathcal{H}_0^1(\Omega, \mathbb{R})$  to allow for consistency. The corresponding precedence relation  $\prec_{\mathcal{F}}$  is simply  $(0 \prec 1)$ ,  $(0 \prec 2)$ ,  $(2 \prec 3)$ ,  $(1, 3 \prec 4)$ .

Analogous to before, one can now generalize the term piecewise linearization of an abs-decomposable function in terms of operators.

**Definition 2.** Let  $(\{\mathcal{F}_i\}, \{\mathcal{U}_i\}, \{\mathcal{W}_i\})^{\prec_{\mathcal{F}}}$  be the decomposition of a decomposable operator  $\mathcal{F}$ , and denote by  $\mathcal{F}'_i : U_i \rightarrow \mathcal{L}(\mathcal{U}_i, \mathcal{W}_i)$ ,  $v_i \rightarrow \mathcal{F}'_i(v_i)$ , the Frechet derivative at the argument  $u_i := (v_j)_{j \prec i} \in U_i$  for each intermediate Frechet differentiable operator  $\mathcal{F}_i$  from the original evaluation procedure. Assume that for any given input  $v_0 := u$  and  $\delta v_0 := \delta u \in U$  all increments  $\delta u_i := (\delta v_j)_{j \prec i} \in U_i$  and that intermediate spaces of the extended evaluation procedure

$$\delta v_i = \mathcal{F}'_i(u_i; \delta u_i), \quad i = 0, \dots, k,$$

are well-defined and consistent with

$$\mathcal{F}'_i(u_i; \delta u_i) := \begin{cases} \mathcal{F}'_i(u_i) \delta u_i & \text{if } \mathcal{F}_i \text{ is Frechet differentiable} \\ |u_i + \delta u_i| - |u_i| & \text{otherwise.} \end{cases}$$

The result  $\delta v_k$  of the modified evaluation procedure, represented by the quadruple  $(\{\mathcal{F}_i\}, \{\mathcal{F}'_i\}, \{\mathcal{U}_i\}, \{\mathcal{W}_i\})^{\prec_{\mathcal{F}}}$ , will be called the piecewise linearization  $\mathcal{F}'(u; \delta u)$  of  $\mathcal{F}$  at  $u$  for the direction  $\delta u$  and  $\mathcal{G}(u) := \mathcal{F}(u) + \mathcal{F}'(u; \delta u)$  is the piecewise linear model of the operator  $\mathcal{F}$  at  $u$ .

The latter definition yields the following results for the given example.

**Example (continued).** As can be easily seen, the operators  $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2$ , and  $\mathcal{F}_4$  are all linear and bounded, with their corresponding Frechet derivatives  $\mathcal{I}$ ,  $\mathcal{I} - \Delta$ ,  $\mathcal{I} + \Delta$ , and  $(\mathcal{I}, -\mathcal{I})$ , respectively. For these operators, the increments  $\delta v_i$  of the modified evaluation procedure are therefore given by

$$\begin{aligned} \mathcal{F}'_0(u; \delta u) &= \delta v_0, & \mathcal{F}'_1(u_1; \delta u_1) &= \delta v_1 - \Delta \delta v_1, \\ \mathcal{F}'_2(u_2; \delta u_2) &= \delta v_1 + \Delta \delta v_1, & \text{and } \mathcal{F}'_4(u_4; \delta u_4) &= \delta v_1 - \delta v_3, \end{aligned}$$

whereas  $\mathcal{F}'_3(u_3; \delta u_3) = |v_2 + \delta v_2| - |v_2|$  using  $v_2 = u - \varphi + \Delta u - f$ . Hence, the piecewise linear model turns out to be

$$\mathcal{F}(u) + \mathcal{F}'(u; \delta u) = 2 \min(u + \delta u - \varphi, -\Delta(u + \delta u) + f),$$

for the (element/point-wise) min-operator  $\min(a, b) := \frac{1}{2}(a + b - |a - b|)$ .

In fact, several parts of the concept for (piecewise linear) algorithmic differentiation can be adapted from finite dimensions besides the previous ones, such as the directed acyclic graph [13] for the precedence relation  $\prec$ , and it is straightforward to extend other definitions and results from [12] to the operator setting. For example, the following conclusions for the relation between any decomposable operator and its piecewise linearization hold true:

**Proposition 1.** *Suppose that  $\mathcal{F} : \mathcal{U} \rightarrow \mathcal{W}$  is decomposable and all intermediate Frechet derivatives  $\mathcal{F}'_i(u_i)$  of its piecewise linearization are Lipschitz continuous on some open subset  $U$  of a closed convex domain  $\mathcal{K}$ . Then there exists a non-negative constant  $c_1 \in \mathbb{R}$  such that for all  $\hat{u}, \check{u} \in \mathcal{K}$ ,*

$$\|\mathcal{F}(\hat{u}) - \mathcal{F}(\check{u}) - \mathcal{F}'(\check{u}; \hat{u} - \check{u})\|_{\mathcal{W}} \leq c_1 \|\hat{u} - \check{u}\|_{\mathcal{U}}^2. \quad (1)$$

Moreover, there exists a constant  $0 \leq c_2 \in \mathbb{R}$  such that for any pair  $\hat{u}, \check{u} \in \mathcal{K}$

$$\|\mathcal{F}'(\hat{u}; \delta u) - \mathcal{F}'(\check{u}; \delta u)\|_{\mathcal{W}} / (1 + \|\delta u\|_{\mathcal{U}}) \leq c_2 \|\hat{u} - \check{u}\|_{\mathcal{U}} \quad \forall \delta u \in \mathcal{U}. \quad (2)$$

*Proof.* Follows the same arguments from the finite-dimensional scenario.  $\square$

Obviously, the piecewise linear models  $\mathcal{G}$  of  $\mathcal{F}$  are of special interest for the generalized Newton method to find a sequence of iterates  $\{u^j\}_{j=0}^{\infty} \subset U$

$$u^{j+1} = u^j + \alpha^p \delta u^j, \quad \text{where } \delta u^j \text{ solves } 0 = \mathcal{F}(u^j) + \mathcal{F}'(u^j; \delta u^j) \quad (3)$$

from some initial guess  $u^0 \approx u^*$  that (at least locally) converge to a root  $u^*$  of  $\mathcal{F}$ , if existing. The global convergence is often improved by a step-size regulation, that is, by scaling the Newton direction  $\delta u$  with a non-negative step-size parameter  $\alpha \in \mathbb{R}$  derived from some line-search method [23] on an appropriate merit function [4]. For the considered example, one finds the following result in the Newton scenario.

**Example (continued).** *The previously considered operator  $\mathcal{F}$  is apparently itself piecewise linear since  $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2$ , and  $\mathcal{F}_4$  are all linear and bounded, and  $\mathcal{F}_4$  is an abs-operator. In fact,  $\mathcal{F}$  and its piecewise linear model coincide such that (1) is satisfied with  $c_1 = 0$  and the generalized Newton method (3) converges within one iteration for any initial value  $u^0$  if  $\delta u^0$  is a solution of*

$$0 \stackrel{\text{a.e.}}{=} 2 \min(u^0 + \delta u - \varphi, -\Delta(u^0 + \delta u) + f).$$

In the following, we use vectorial notation for the analysis and represent piecewise linear operators in terms of the standard abs-normal form (ANF) that was originally proposed in [19] and just recently rediscovered in [25]. Here, piecewise linear refers to operators  $\mathcal{G} : U \subset \mathcal{U} \rightarrow \mathcal{W}$  whose evaluation procedures contain only intermediate operators that are either linear and bounded or an abs-operator as is the case for the piecewise linear models of general abs-decomposable operators  $\mathcal{F} : U \subset \mathcal{U} \rightarrow \mathcal{W}$  at a fixed argument  $u \in U$ , namely  $\mathcal{G}(\delta u) := \mathcal{G}_u(\delta u)$  with  $\mathcal{G}_u(\delta u) := \mathcal{F}(u) + \mathcal{F}'(u; \delta u)$ . Basically, the only difference for the ANF is that  $a, b$  and  $Z, L, J, Y$  now denote functions and linear, bounded operators instead of vectors and matrices. Also, the former (sub-)matrix-vector products need to be thought of as applying an operator on the corresponding element instead of as a simple matrix-vector multiplication.

**Definition 3.** Let  $\mathcal{G} : \mathcal{U} \rightarrow \mathcal{W}$  be decomposable into  $(\{\mathcal{G}_i\}, \{\mathcal{U}_i\}, \{\mathcal{W}_i\})^{\prec \mathcal{F}}$  and denote by  $\mathbf{z} = (z_j)_{j=1}^s \in \mathcal{U}_S := \mathcal{U}_{i_1} \times \cdots \times \mathcal{U}_{i_s}$  the arguments of the  $s \in \mathbb{N}$  occurring abs-operators  $\mathcal{G}_{i_j} : \mathcal{U}_{i_j} \rightarrow \mathcal{W}_{i_j} = \mathcal{U}_{i_j}$ . The representation

$$\begin{bmatrix} \mathbf{z} \\ \mathcal{G}(\delta u) \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} Z & L \\ J & Y \end{bmatrix} \begin{bmatrix} \delta u \\ |\mathbf{z}| \end{bmatrix} \iff \begin{bmatrix} \mathbf{z} \\ \mathcal{G}(\delta u) \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} Z & L\Sigma_z \\ J & Y\Sigma_z \end{bmatrix} \begin{bmatrix} \delta u \\ \mathbf{z} \end{bmatrix} \quad (4)$$

of  $\mathcal{G}$  with  $Z \in \mathcal{L}(\mathcal{U}, \mathcal{U}_S)$ ,  $J \in \mathcal{L}(\mathcal{U}, \mathcal{W})$ , and  $Y \in \mathcal{L}(\mathcal{U}_S, \mathcal{W})$  will be referred to as an abs-normal form of  $\mathcal{G}$  if the operator  $L$  is strictly lower(-left) triangular:

$$L = \begin{bmatrix} 0 & \dots & 0 \\ * & & \\ \vdots & \ddots & \vdots \\ * & \dots & * & 0 \end{bmatrix}, \quad L \in \mathcal{L}(\mathcal{U}_S, \mathcal{U}_S).$$

More precisely, the components representing the derivatives  $\partial z_j / \partial |z_i|$  of the mapping  $|\mathbf{z}| \rightarrow \mathbf{z}$  must vanish for all indices  $i \geq j$ ,  $i, j = 1, \dots, s$ , which are represented by the upper right entries  $L_{ij} \equiv 0$  in the above matrix notation.

Analogous to the finite dimensional case, one can define the switching depth  $\eta \in \mathbb{N}$ , which is the smallest number such that  $(\eta + 1)$ -time repeated application  $L^{\eta+1} = L \circ L \circ \cdots \circ L$  of the operator  $L$  onto itself yields  $L^{\eta+1} \equiv 0$  and call ANFs with switching depth  $\eta = 0$  simply switched. Also, there always exist functions  $a, b$  and operators  $Z, L, J$ , and  $Y$  such that any piecewise linear function  $\mathcal{G}$  with a finite evaluation procedure can be represented

as an ANF. If  $\mathcal{G}$  is a piecewise linear model  $\mathcal{F}(u) + \mathcal{F}'(u; \delta u)$  of an operator  $\mathcal{F} : \mathcal{U} \rightarrow \mathcal{W}$  then the quantities of the ANF usually also depend on the argument  $u \in \mathcal{U}$ :

$$a = a(u) = (a_1(u), a_2(u), \dots, a_s(u)) : \mathcal{U} \rightarrow \mathcal{U}^s, \quad b = b(u) : \mathcal{U} \rightarrow \mathcal{W}$$

and the operators  $Z = Z(u)$ ,  $L = L(u)$ ,  $J = J(u)$ ,  $Y = Y(u)$  satisfy

$$\begin{aligned} u \in U &\rightarrow Z(u) \in \mathcal{L}(\mathcal{U}, \mathcal{U}_S), & u \in U &\rightarrow J(u) \in \mathcal{L}(\mathcal{U}, \mathcal{W}), \\ u \in U &\rightarrow L(u) \in \mathcal{L}(\mathcal{U}_S, \mathcal{U}_S), & u \in U &\rightarrow Y(u) \in \mathcal{L}(\mathcal{U}_S, \mathcal{W}). \end{aligned}$$

Although this dependence should always be kept in mind, it will not be explicitly stated in the following, in order to avoid cumbersome notation. Assuming that  $\mathcal{W} \equiv \mathcal{U}$  and that the operator  $J$  is invertible with a corresponding inverse operator  $J^{-1}$ , one can use the Schur-complement operator

$$\mathcal{S} = L - ZJ^{-1}Y \in \mathcal{L}(\mathcal{U}^s, \mathcal{U}^s)$$

to define a modulus fixed-point equation on the arguments  $\mathbf{z} \in \mathcal{U}^s$  of the abs-operators by

$$[\mathcal{I} - \mathcal{S}\Sigma_{\mathbf{z}}]\mathbf{z} = a - ZJ^{-1}b, \tag{5}$$

as was pointed out in [19, 25]. The latter equation is solved by a limit  $\mathbf{z}^* = \lim_{k \rightarrow \infty} \mathbf{z}^k$  of the modulus fixed-point iteration

$$\mathbf{z}^{k+1} := [\mathcal{I} - \mathcal{S}\Sigma_{\mathbf{z}^k}]^{-1}(a - ZJ^{-1}b) \tag{6}$$

if this limit exists and was found for some initial guess  $\mathbf{z}^0 \in \mathcal{U}_S$  (e.g.  $\mathbf{z}^0 = a$ ). The solution  $\mathbf{z}^*$  then provides a solution

$$\delta u = -J^{-1}[b + Y|z^*|] \tag{7}$$

of the ANF for the prescribed function value  $0 = \mathcal{G}(\delta u)$ . However, the convergence of the fixed-point iteration to a fixed point  $\mathbf{z}^*$  depends on the entries of the ANF at  $u$  and the initial value  $\mathbf{z}^0$ , as was observed in [25].

**Example (continued).** *An ANF for the piecewise linear model of the previously considered  $\mathcal{F}$  at  $u$  and some directional increment  $\delta u$  is given by*

$$\begin{bmatrix} \mathbf{z} \\ 2\mathcal{G}(\delta u) \end{bmatrix} = \begin{bmatrix} u - \varphi + \Delta u - f \\ u - \varphi - \Delta u + f \end{bmatrix} + \begin{bmatrix} \mathcal{I} + \Delta & 0 \\ \mathcal{I} - \Delta & -\mathcal{I} \end{bmatrix} \begin{bmatrix} \delta u \\ |\mathbf{z}| \end{bmatrix}.$$

Obviously, only  $a := u - \varphi + \Delta u - f$  and  $b := u - \varphi - \Delta u + f$  depend on  $u$ , whereas  $Z := \mathcal{I} + \Delta$ ,  $L := 0$ ,  $J := \mathcal{I} - \Delta$ , and  $Y := -\mathcal{I}$  are independent of  $u$ . Since the operator  $J = (\mathcal{I} + \Delta)$  is invertible, one can use its inverse mapping  $J^{-1} = (\mathcal{I} + \Delta)^{-1}$  to define the modulus fixed-point equation (6) with  $\mathcal{S} = 0 - (\mathcal{I} + \Delta) \circ (\mathcal{I} + \Delta)^{-1} \circ (-\mathcal{I}) \circ \Sigma_{\mathbf{z}}$  to find a direction  $\delta u$  with  $\mathcal{G}(\delta u) = 0$  for the generalized Newton method (3).

Clearly, the advantage of this approach, namely, first formulating the ANF in infinite dimensions instead of using a first-discretize-then-optimize approach, is that the structure of the considered problems can be exploited. In detail, the equations can now be treated in the correct spaces and solved by appropriate procedures, for example by some multigrid methods to improve the efficiency of the resulting algorithms. At least for the previous example, one can also benefit from the fact that  $Z$  and  $J^{-1}$  are commutative operators and reformulate the fixed-point equation in a more numerical efficient way

$$[J - Z\Sigma_{\mathbf{z}^k}]\mathbf{z}^{k+1} = Ja - Zb, \quad (8)$$

which can be iteratively solved and avoids dense matrix approximations of the inverse Laplace operator (or modifications) in the discrete formulation later on. However, there is no reason that  $J$  should be invertible at all. For this case, Griewank suggested in [12] using the simple identity

$$\delta u = |\delta u + |\delta u|| - |\delta u|, \quad (9)$$

or modifications of it, to define an ANF, where the corresponding lower left block is invertible, which inspires the definition of an extended ANF:

**Definition 4.** Let  $\mathcal{G} : U \subset \mathcal{U} \rightarrow \mathcal{U}$  be a decomposable operator that is represented by the ANF (4) and  $\Gamma = \Gamma(u) \in \mathcal{L}(\mathcal{U}, \mathcal{U})$  a given operator that might depend on  $u$ . Then the representation

$$\begin{bmatrix} \mathbf{z} \\ \hat{\mathbf{z}} \\ \check{\mathbf{z}} \\ \mathcal{G}(\delta u) \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \\ b \end{bmatrix} + \left[ \begin{array}{c|ccc} Z & L\Sigma_{\mathbf{z}} & 0 & 0 \\ J - \Gamma & 0 & 0 & 0 \\ J - \Gamma & 0 & \Sigma_{\hat{\mathbf{z}}} & 0 \\ \hline \Gamma & Y\Sigma_{\mathbf{z}} & -\Sigma_{\check{\mathbf{z}}} & \Sigma_{\check{\mathbf{z}}} \end{array} \right] \begin{bmatrix} \delta x \\ \mathbf{z} \\ \hat{\mathbf{z}} \\ \check{\mathbf{z}} \end{bmatrix} \quad (10)$$

is called an extended abs-normal form (of its original ANF w.r.t. to  $\Gamma$ ), where  $\Sigma_{\mathbf{z}}$  and  $\Sigma_{\hat{\mathbf{z}}}$ ,  $\Sigma_{\check{\mathbf{z}}}$  are the abs-operators for the original switching functions  $\mathbf{z} \in \mathcal{U}_S$  and the additional variables  $\hat{\mathbf{z}}, \check{\mathbf{z}}$  in some appropriate space  $\hat{\mathcal{U}}, \check{\mathcal{U}} \supseteq \mathcal{U}$ , respectively.

Using the same arguments as for the original ANF, one can now define a modulus fixed-point iteration for the extended ANF:

**Proposition 2.** *Denote by  $\Sigma_{\mathbf{z}}$ ,  $\Sigma_{\hat{\mathbf{z}}}$ , and  $\Sigma_{\check{\mathbf{z}}}$  the abs-operators of some current iterates  $(\mathbf{z}^k, \hat{\mathbf{z}}^k, \check{\mathbf{z}}^k)$  of the modulus fixed-point iteration (6) for the extended ANF (10) with prescribed  $0 = \mathcal{G}(\delta u)$ , and let  $\Gamma(u) \in \mathcal{L}(\mathcal{U}, \mathcal{U})$  be a given invertible operator. Then the next iterates  $(\mathbf{z}^{k+1}, \hat{\mathbf{z}}^{k+1}, \check{\mathbf{z}}^{k+1})$  are given by the solution of the system*

$$\begin{bmatrix} C & Z\Gamma^{-1}(\Sigma_{\check{\mathbf{z}}} - \Sigma_{\hat{\mathbf{z}}} + \Sigma_{\check{\mathbf{z}}}\Sigma_{\hat{\mathbf{z}}}) & 0 \\ 0 & B & 0 \\ 0 & -(\mathcal{I} + \Sigma_{\hat{\mathbf{z}}}) & \mathcal{I} \end{bmatrix} \begin{bmatrix} \mathbf{z}^{k+1} \\ \hat{\mathbf{z}}^{k+1} \\ \check{\mathbf{z}}^{k+1} \end{bmatrix} = \begin{bmatrix} a - Z\Gamma^{-1}b \\ (J\Gamma^{-1} - \mathcal{I})(Y\Sigma_{\mathbf{z}}C^{-1}[Z\Gamma^{-1}b - a] + b) \\ 0 \end{bmatrix}$$

if the two operators  $C := \mathcal{I} - L\Sigma_{\mathbf{z}} + Z\Gamma^{-1}Y\Sigma_{\mathbf{z}}$  and

$$B := \mathcal{I} + (J\Gamma^{-1} - \mathcal{I})(-\Sigma_{\hat{\mathbf{z}}} + \Sigma_{\check{\mathbf{z}}} + \Sigma_{\check{\mathbf{z}}}\Sigma_{\hat{\mathbf{z}}}) - (J\Gamma^{-1} - \mathcal{I})Y\Sigma_{\mathbf{z}}C^{-1}Z\Gamma^{-1}(-\Sigma_{\hat{\mathbf{z}}} + \Sigma_{\check{\mathbf{z}}} + \Sigma_{\check{\mathbf{z}}}\Sigma_{\hat{\mathbf{z}}})$$

are invertible.

*Proof.* Since  $\Gamma$  is invertible, one can compute the corresponding Schur complement  $\mathcal{S}$  for the extended ANF (10) and formulate the left-hand side  $\mathcal{I} - \mathcal{S}\Sigma$  of the modulus fixed point iteration (6) to determine the next switching functions  $(\mathbf{z}^{k+1}, \hat{\mathbf{z}}^{k+1}, \check{\mathbf{z}}^{k+1})$  for some currently given iterates  $(\mathbf{z}^k, \hat{\mathbf{z}}^k, \check{\mathbf{z}}^k)$  using the defined abs-operators  $\Sigma_{\mathbf{z}}$ ,  $\Sigma_{\hat{\mathbf{z}}}$ , and  $\Sigma_{\check{\mathbf{z}}}$ , namely,

$$\begin{aligned} \mathcal{I} - \mathcal{S}\Sigma &= \begin{bmatrix} \mathcal{I} - L\Sigma_{\mathbf{z}} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ 0 & -\Sigma_{\hat{\mathbf{z}}} & \mathcal{I} \end{bmatrix} + \begin{bmatrix} Z \\ J - \Gamma \\ J - \Gamma \end{bmatrix} \Gamma^{-1} \begin{bmatrix} Y\Sigma_{\mathbf{z}} & -\Sigma_{\hat{\mathbf{z}}} & \Sigma_{\check{\mathbf{z}}} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{I} - L\Sigma_{\mathbf{z}} + Z\Gamma^{-1}Y\Sigma_{\mathbf{z}} & -Z\Gamma^{-1}\Sigma_{\hat{\mathbf{z}}} & Z\Gamma^{-1}\Sigma_{\check{\mathbf{z}}} \\ (J - \Gamma)\Gamma^{-1}Y\Sigma_{\mathbf{z}} & \mathcal{I} - (J - \Gamma)\Gamma^{-1}\Sigma_{\hat{\mathbf{z}}} & (J - \Gamma)\Gamma^{-1}\Sigma_{\check{\mathbf{z}}} \\ (J - \Gamma)\Gamma^{-1}Y\Sigma_{\mathbf{z}} & -\Sigma_{\hat{\mathbf{z}}} - (J - \Gamma)\Gamma^{-1}\Sigma_{\hat{\mathbf{z}}} & \mathcal{I} + (J - \Gamma)\Gamma^{-1}\Sigma_{\check{\mathbf{z}}} \end{bmatrix}. \end{aligned}$$

This expression can be simplified by the transformation  $Q := Q_3Q_2Q_1$  that is given by the three elementary bijective operators

$$Q_1 = \begin{bmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ 0 & -\mathcal{I} & \mathcal{I} \end{bmatrix}, Q_2 = \begin{bmatrix} \mathcal{I} & 0 & -Z\Gamma^{-1}\Sigma_{\check{\mathbf{z}}} \\ 0 & \mathcal{I} & -(J\Gamma^{-1} - \mathcal{I})\Sigma_{\hat{\mathbf{z}}} \\ 0 & 0 & \mathcal{I} \end{bmatrix},$$

$$Q_3 = \begin{bmatrix} \mathcal{I} & 0 & 0 \\ -(J\Gamma^{-1} - \mathcal{I})Y\Sigma_{\mathbf{z}}C^{-1} & \mathcal{I} & 0 \\ 0 & 0 & \mathcal{I} \end{bmatrix},$$

where the last one exists only if  $C = \mathcal{I} - L\Sigma_{\mathbf{z}} + Z\Gamma^{-1}Y\Sigma_{\mathbf{z}}$  is invertible. Thus, applying the transformation  $Q$  on both sides of the modulus fixed point iteration for the extended ANF does not alter the solution and yields the desired result with the stated operator  $B$ .  $\square$

Note that the assumption “ $C$  is invertible” in the proposition can always be achieved by choosing  $\Gamma$  in such a way that  $\|\Gamma^{-1}\|$  is sufficiently small. However, the requirement of  $B$  being invertible cannot be always guaranteed since, for example, in the case  $J = 0$ ,  $\Sigma_{\mathbf{z}} = 0$  and  $-\Sigma_{\mathbf{z}} + \Sigma_{\mathbf{z}} + \Sigma_{\mathbf{z}}\Sigma_{\mathbf{z}} = \mathcal{I}$  one finds  $B = 0$ . Furthermore, by considering the extended ANF in order to resolve the issue of defining the modulus fixed-point iteration in case of singular  $J$ , one introduces a number of additional switching variables that might increase the likelihood of divergence of the modulus fixed point iteration in terms of cycling [25]. Moreover, the system might not be solvable at all, for example, under the previous assumptions with  $b \neq 0$ . Therefore, the algorithm in the next section is stated as general as possible to allow for different fallback options (see [2, 3, 20]) in case of problems like the following.

**Example 2.** For some prescribed functions  $f$ ,  $\varphi_l$ , and  $\varphi_u$  with  $\varphi_u \geq \varphi_l$ , consider the modified linear complementarity problem

$$\begin{aligned} 0 \leq +u(x) - \varphi_l(x) \perp -\Delta u(x) + f(x) \geq 0 \quad \text{and} \\ 0 \leq -u(x) + \varphi_u(x) \perp +\Delta u(x) - f(x) \geq 0 \quad \text{for almost all } x \in \Omega, \end{aligned}$$

which requires  $-\Delta u + f = 0$  to be satisfied whenever  $u$  is (strictly) between its lower or upper bound,  $\varphi_l$  and  $\varphi_u$ , respectively. It can be written as

$$0 \stackrel{a.e.}{=} \mathcal{F}(u) = \min [\min[u - \varphi_l, -\Delta u + f], \min[-u + \varphi_u, \Delta u - f]]$$

such that its piecewise linear model  $\mathcal{G}(\delta u) = \mathcal{F}(u) + \mathcal{F}'(u; \delta u)$  at  $u$ ,

$$\min [\min [u + \delta u - \varphi_l, -\Delta[u + \delta u] + f], \min[-u - \delta u + \varphi_u, \Delta[u + \delta u] - f]],$$

is represented by an ANF with a singular smooth part  $J = 0$ . The quantities of the extended ANF (10) for the piecewise linear model of the modified linear complementarity problem are given by

$$\begin{aligned} a = \begin{bmatrix} +u - \varphi_l + \Delta u - f \\ -u + \varphi_u - \Delta u + f \\ 2(u - \Delta u + f) - \varphi_l - \varphi_u \end{bmatrix}, \quad Z = \begin{bmatrix} +\mathcal{I} + \Delta \\ -\mathcal{I} - \Delta \\ 2(\mathcal{I} - \Delta) \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\mathcal{I} & \mathcal{I} & 0 \end{bmatrix} \\ b = [\varphi_u - \varphi_l], \quad J = [0], \quad \text{and} \quad Y = [-\mathcal{I} \quad -\mathcal{I} \quad -\mathcal{I}]. \end{aligned}$$

### 3 Nonsmooth multigrid method

This section contains a multigrid method for the numerically efficient solution of  $\mathcal{F}(u) = 0$  for some abs-factorizable operator  $\mathcal{F} : U \subset \mathcal{U} \rightarrow \mathcal{U}$  that represents some nonsmooth PDE. The proposed method follows an approach similar to the original multigrid method [14, 17], and thus its structure mainly coincides with the one in the smooth case. For simplicity, only piecewise-linear functions  $\mathcal{G} : U \subset \mathcal{U} \rightarrow \mathcal{U}$  are considered that arise as piecewise-linear models  $\mathcal{G}(\delta u) = \mathcal{F}(u) + \mathcal{F}'(u; \delta u)$  of a nonlinear operator  $\mathcal{F}$ . In the smooth case, the equality  $\mathcal{G}(\delta u) = 0$  is approximated by a discretized linear equation  $G(\delta u) = 0$ , which can be written as

$$A^l \delta u^l = q^l \quad (11)$$

for some appropriate discretization matrix  $A^l$  representing the Jacobian of  $\mathcal{F}$  at  $u$  and corresponding right-hand side  $q^l$ . Here, the superscripts  $l \in \mathbb{N}$  indicate the current discretization level  $l$  so that  $l - 1$  and  $l + 1$  represent the next coarser and finer discretization level, respectively. Consequently,

$$R_u^{l,l-1} : \mathcal{U}^l \rightarrow \mathcal{U}^{l-1} \quad (R_z^{l,l-1} : \mathcal{U}_S^l \rightarrow \mathcal{U}_S^{l-1})$$

denotes the restriction operator, which maps elements from the space of fine-grid functions  $\mathcal{U}^l$  ( $\mathcal{U}_S^l$ ) to the corresponding elements in the space of coarse grid functions  $\mathcal{U}^{l-1}$  ( $\mathcal{U}_S^{l-1}$ ); and, analogously, the prolongation operator is denoted by

$$P_u^{l-1,l} : \mathcal{U}^{l-1} \rightarrow \mathcal{U}^l \quad (P_z^{l-1,l} : \mathcal{U}_S^{l-1} \rightarrow \mathcal{U}_S^l).$$

In particular, the prolongation and restriction usually satisfy the equality  $R_u^{l,l-1}(P_u^{l-1,l}(u)) = u$  ( $R_z^{l,l-1}(P_z^{l-1,l}(z)) = z$ ), namely, the application of a restriction on a prolonged element from the coarse space is the element itself (in the coarse space). Using this notation, one can state the key ingredients of the nonsmooth multigrid algorithm, that is, the full multigrid method, the basic multigrid method, and the simple V-Cycle given by Algorithms 1, 2, and 3, respectively.

---

**Algorithm 1**  $[\mathbf{z}^l, u^l] = \mathbf{FullMG}[l_{max}, l, l_{min}, \mathbf{z}^l, u^l, \varepsilon, maxiter, maxmgv]$

---

```

1: for  $l = l, \dots, l_{min} + 1$  do
2:   Restrict  $\mathbf{z}^{l-1} := R_{\mathbf{z}}^{l,l-1}(\mathbf{z}^l)$  and  $u^{l-1} := R_u^{l,l-1}(u^l)$ 
3: end for
4: for  $l = l_{min}, \dots, l_{max}$  do
5:    $[\mathbf{z}^l, u^l, flag] := \mathbf{MGV}[l, l_{min}, \mathbf{z}^l, u^l, \varepsilon, maxiter, maxmgv]$ 
6:   if  $l < l_{max}$  then
7:     Project  $\mathbf{z}^l := D^l(\mathbf{z}^l)$ 
8:     Prolong  $\mathbf{z}^{l+1} := P_{\mathbf{z}}^{l,l+1}(\mathbf{z}^l)$  and  $u^{l+1} := P_u^{l,l+1}(u^l)$ 
9:   end if
10: end for

```

---

The key difference between the original full multigrid method for the smooth case (see [14, 17]) and the presented method **FullMG** is that instead of modifying  $u$  directly by solving the residual equations for (11) on the coarser grids to find a suitable correction of  $\delta u^l$ , one applies the V-cycle on the switching variables  $\mathbf{z}$  depicted in Figure 1, which is then used to compute a correction of  $\delta u^l$  at each level  $l$ .

---

**Algorithm 2**  $[\mathbf{z}^l, u^l, flag] = \mathbf{MGV}[l, l_{min}, \mathbf{z}^l, u^l, \varepsilon, maxiter, maxmgv]$

---

```

1:  $flag := false; iter = 1;$ 
2: while  $(flag == false) \&\& (iter \leq maxmgv)$  do
3:   Set  $\delta u^l := 0$ 
4:    $[\mathbf{z}^l, \delta u^l, flag] := \mathbf{VCycle}[l, l_{min}, \mathbf{z}^l, u^l, \delta u^l, \varepsilon, maxiter]$ 
5:   Update  $u^l := u^l + \delta u^l; iter = iter + 1;$ 
6: end while

```

---

In detail, for some given initial approximations  $u^l$  and  $z^l$  on the grid  $l$ , the nonsmooth **VCycle** used in the multigrid cycle **MGV** differs from the original method in that instead of evaluating the discretization matrix  $A^l$  for each level  $l$ , one evaluates in **evalANF** the abs-normal form (4) of  $G$  at  $u^l$  for each level  $l$  and performs the restriction, prolongation, and projection on the variables  $\mathbf{z}$ ,  $u$ ,  $\delta u$  (lines 5,7,8). The motivation for the projection operator

$$D^l : \mathcal{U}_S^l \rightarrow \mathcal{U}_S^l$$

will be explained in the next section, as well as an appropriate choice for the prolongation and restriction operators  $P^{l-1,l}$  and  $R^{l,l-1}$ , respectively.

---

**Algorithm 3**  $[\mathbf{z}^l, \delta u^l, flag] = \mathbf{VCycle}[l, l_{min}, \mathbf{z}^l, u^l, \delta u^l, \varepsilon, maxiter]$

---

```

1: Evaluate  $[a, b, Z, L, J, Y] := \mathbf{evalANF}[l, u^l, \delta u^l]$ 
2: if  $(l \leq l_{min})$  then  $maxiter = +\infty$ ;
3: else
4:    $[\mathbf{z}^l, \delta u^l, flag] := \mathbf{solveANF}[l, a, b, Z, L, J, Y, \mathbf{z}^l, \delta u^l, \varepsilon, maxiter]$ 
5:   Project  $\mathbf{z}^l := D^l(\mathbf{z}^l)$ 
6:   Restrict  $\mathbf{z}^{l-1} := R_z^{l,l-1}(\mathbf{z}^l)$ ,  $u^{l-1} := R_u^{l,l-1}(u^l)$ , and  $\delta u^{l-1} := R_u^{l,l-1}(\delta u^l)$ 
7:    $[\mathbf{z}^{l-1}, \delta u^{l-1}] := \mathbf{VCycle}[l-1, l_{min}, \mathbf{z}^{l-1}, u^{l-1}, \delta u^{l-1}, \varepsilon, maxiter]$ 
8:   Project  $\mathbf{z}^{l-1} := D^l(\mathbf{z}^{l-1})$ 
9:   Prolong  $\mathbf{z}^l := P_z^{l-1,l}(\mathbf{z}^{l-1})$  and  $\delta u^l := P_u^{l-1,l}(\delta u^{l-1})$ 
10: end if
11:  $[\mathbf{z}^l, \delta u^l, flag] := \mathbf{solveANF}[l, a, b, Z, L, J, Y, \mathbf{z}^l, \delta u^l, \varepsilon, maxiter]$ 

```

---

The method **solveANF** in the V-cycle given by Algorithm 4 corresponds to a smoothing operator on the current grid level that is usually applied in smooth multigrid methods (e.g., a Jacobi, Gauss-Seidel, or Richardson method). But instead of improving the linear residual equation of (11) as it is done in the smooth case, now one or more steps of the modulus fixed-point iteration **FPiter** (Eq. (6)) are performed to find a better approximation  $\mathbf{z}^l$ . The latter approximation  $\mathbf{z}^l$  is then used to compute an update  $\delta u^l$  for the multigrid algorithm 2 using Equation (7). In fact, the components of the multigrid algorithm are designed in such a way that any of the proposed iterations from [2, 3, 6, 25, 20] (e.g., also the Block-Seidel iteration) can be encoded in the method **FPiter** to solve the ANF and allow for alternatives in the case of singular  $J$ .

---

**Algorithm 4**  $[\mathbf{z}^l, \delta u^l, flag] = \mathbf{solveANF}[l, a, b, Z, L, J, Y, \mathbf{z}^l, \delta u^l, \varepsilon, maxiter]$

---

```

1:  $iter := 0$ ,  $flag := false$ 
2: while  $(flag == false)$  and  $(iter < maxiter)$  do
3:    $iter := iter + 1$ 
4:    $[\mathbf{z}_{new}^l, \delta u_{new}^l] := \mathbf{FPiter}[l, a, b, Z, L, J, Y, \mathbf{z}^l, \delta u^l]$ 
5:   if  $(\|\mathbf{z}^l - \mathbf{z}_{new}^l\|_{\mathcal{U}_S} < tol)$  then  $flag := true$ 
6:   end if
7:   Update  $\mathbf{z}^l := \mathbf{z}_{new}^l$  and  $\delta u^l := \delta u_{new}^l$ 
8: end while

```

---

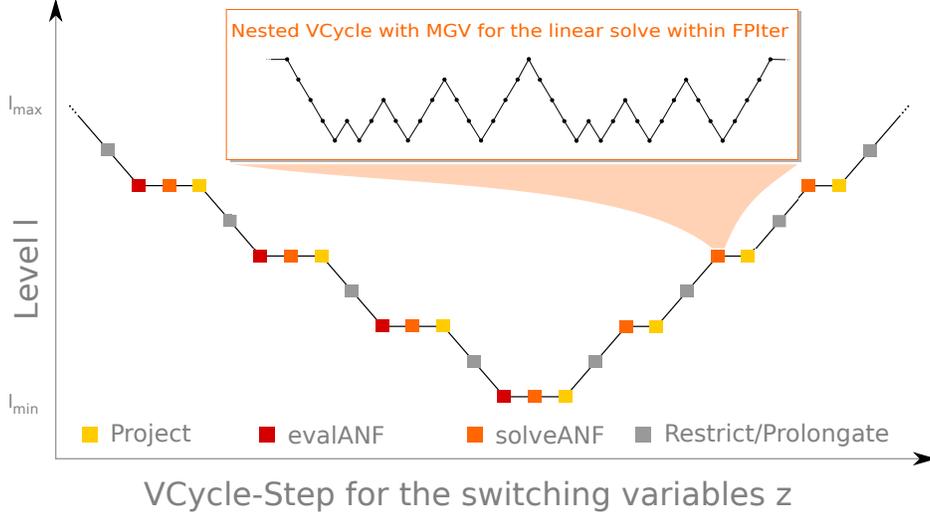


Figure 1: V-cycle of the proposed nonsmooth multigrid method and a more elaborate nested V-cycle, where each step of the modulus fixed-point iteration (6) and (7) is computed by a standard multigrid method to increase efficiency

### 3.1 Prolongation, Projection, and Restriction

The space  $\mathcal{U}_S$  of the switching variables is in general a product  $\mathcal{U}_1 \times \cdots \times \mathcal{U}_m \times \cdots \times \mathcal{U}_s$  of several spaces  $\mathcal{U}_m$ . Thus,  $\mathbf{z}$  can be thought of as a vector  $\mathbf{z} = (z_1, \dots, z_s)$  of several components/functions  $z_m$ . After the switching variable  $\mathbf{z}^{l-1}$  is computed on the coarse level  $l-1$  by the modulus fixed-point iteration within the previously described method, each of components  $z^{l-1} := z_m^{l-1} \in \mathcal{U}_m^l$  of  $\mathbf{z}$  needs to be prolonged to the next finer level  $l$ . However, the prolongation  $P_z^{l-1,l}$  for the component  $z^{l-1}$  should not be chosen simply to be one of the usual suspects that are used in smooth multigrid methods. In particular, the standard prolongation operator  $P_z^{l-1,l}$  defined as the weighted average over the direct neighbor values alone is not appropriate for the prolongation of each component  $z^{l-1}$  since it is prone to outliers and has an implicit smoothing effect that can eliminate precious information about  $z^{l-1}$ , which is usually nonsmooth and might even be discontinuous; for example, using only a smooth prolongation operator on  $z^{l-1}$  could result in an interpolated value that is positive although all but one of its neighbors are negative. Therefore, it is advisable to prolong the result of the projection

$$D_z^l(z^l(\omega_z)) = \mathcal{I}_{\{z^l(\omega_z) > 0\}}(z^l(\omega_z)) - \mathcal{I}_{\{z^l(\omega_z) < 0\}}(z^l(\omega_z)) \quad \forall \omega_z \in \Omega_z, \quad (12)$$

which yields the point-wise sign of  $z$ , namely,  $D_z^l(z(\omega_z)) \in \{-1, 0, 1\}$  over the appropriate domain  $\Omega_z \ni \omega_z$ . Alternatively, one can also use the  $\varepsilon$ -projection for sufficiently small  $\varepsilon \in \mathbb{R}_+$

$$D_{z,\varepsilon}^l(z^l(\omega_{z^l})) = \mathcal{I}_{\{z^l(\omega_z) > \varepsilon\}} z^l(\omega_z) - \mathcal{I}_{\{z^l(\omega_z) < -\varepsilon\}} z^l(\omega_{z^l}) \quad \forall \omega_z \in \Omega_z$$

to compensate for numerical floating-point errors, which make it hard to exactly verify  $z^{l-1} = 0$ . These two projection variants are based on the observation that only the sign of  $z^l$  over the corresponding domain  $\Omega_z \ni \omega_{z^l}$  and not its actual value is needed to define the first step of modulus fixed-point iteration at the next finer level. However, the correctness and the quality of the result from the  $\varepsilon$ -projection  $D_{z,\varepsilon}^l$  strongly depend on the scaling of  $z^l$  and the “right” choice of  $\varepsilon$ , which is usually unknown. Therefore, we suggest using the exact sign projection  $D_z^l = D_{z,0}^l$ , which often provides sufficiently accurate results and avoids the task of finding a good estimate for  $\varepsilon$  (as can be seen for the numerical examples later on). The prolongation from the coarse to the fine grid is achieved by some suitable prolongation operator  $P^{l-1,l}$  that leaves the value for every existing grid-point unchanged and assigns to every new point an average value of its neighbors. For example, in the 2D case for a square domain  $\Omega_z$ , which is discretized by an equidistant grid with  $N^{(l-1)}$  discretization points in one direction on the level  $l-1$ , one can use the nine-point prolongation [17] given by

$$\begin{aligned} z_{2i,2j}^l &= z_{i,j}^{l-1} & \forall i = 0 \dots N^{(l-1)}, & \quad \forall j = 0 \dots N^{(l-1)}, \\ z_{2i+1,2j}^l &= \frac{1}{2} (z_{i,j}^{l-1} + z_{i+1,j}^{l-1}) & \forall i = 0 \dots N^{(l-1)} - 1, & \quad \forall j = 0 \dots N^{(l-1)}, \\ z_{2i,2j+1}^l &= \frac{1}{2} (z_{i,j}^{l-1} + z_{i,j+1}^{l-1}) & \forall i = 0 \dots N^{(l-1)}, & \quad \forall j = 0 \dots N^{(l-1)} - 1, \\ z_{2i+1,2j+1}^l &= \frac{1}{4} (z_{i,j}^{l-1} + z_{i,j+1}^{l-1} + z_{i+1,j}^{l-1} + z_{i+1,j+1}^{l-1}) & \forall i, j = 0 \dots N^{(l-1)} - 1. \end{aligned} \tag{13}$$

This combination of projection and prolongation on  $z^{l-1}$  has the simple effect that the interpolated values  $z_{i,j}^l$  of the refined  $z^l$  are given by the average sign of its direct neighbors’  $\mathcal{N}_{i,j}$  values on the coarse grid, or more precisely

$$P_z^{l-1,l}(D^{l-1}(z_{i,j}^{l-1})) = \begin{cases} -1, & \text{if } \sum_{\omega \in \mathcal{N}_{i,j}} \text{sign}(z^{l-1}(\omega)) < 0 \\ 0, & \text{if } \sum_{\omega \in \mathcal{N}_{i,j}} \text{sign}(z^{l-1}(\omega)) = 0. \\ +1, & \text{if } \sum_{\omega \in \mathcal{N}_{i,j}} \text{sign}(z^{l-1}(\omega)) > 0 \end{cases}$$

On the one hand, this behavior is beneficial for problems with domains  $\Omega_z$  that consist of larger subsets, where  $z$  has the same sign, such that elements

in the relative interior of these subsets are assigned the same signature as the majority of it's neighbors and that elements with the same number of neighbors with positive and negative signature are set to the 'neutral' sign zero (cf. Figure 2). On the other hand, this effect might introduce wrong results, for example in the case of structures that resolve only on finer resolutions or have corners/isolated points. The latter artificially introduced errors then require additional correction steps in terms of the modulus fixed-point iteration on the finer grid. Nevertheless, at least for the numerical examples considered in Section 4, the first-project-then-prolong strategy dramatically reduces the overall number of fixed-point iterations and increases the efficiency of the proposed multigrid method.

In an analogous way, one can define the coarse approximations  $z^{l-1}$  of  $z^l$  on the previously projected values  $D^l(z^l)$  using either a (weighted version of the) nine-point restriction operator [17]

$$z_{i,j}^{l-1} = \frac{1}{4}z_{2i,2j}^l + \frac{1}{16} (z_{2i-1,2j-1}^l + z_{2i-1,2j+1}^l + z_{2i+1,2j-1}^l + z_{2i+1,2j+1}^l) + \frac{1}{8} (z_{2i,2j-1}^l + z_{2i,2j+1}^l + z_{2i-1,2j}^l + z_{2i+1,2j}^l), \forall i, j = 1 \dots N^{(l-1)} - 1, \quad (14)$$

or, alternatively, the trivial restriction  $z^{l-1} = R_z^{l-1}(z^l)$  that is given by

$$z_{i,j}^{l-1} = z_{2i,2j}^l, \quad \text{for all } i, j = 0 \dots N^{(l-1)}, \quad (15)$$

where every second element in the relative interior is simply dismissed.

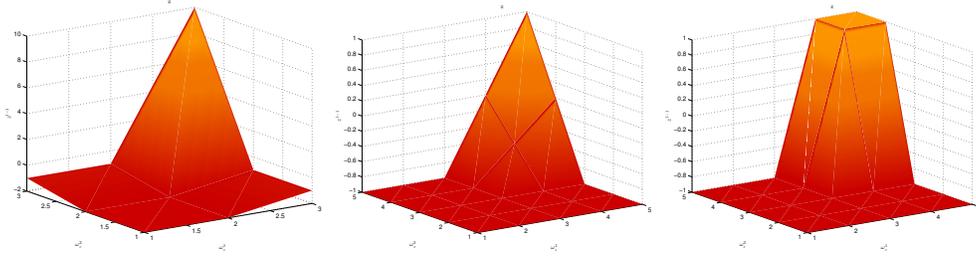


Figure 2: Visualization of a function  $z^{l-1}$  (left), its projected-prolonged-projection  $D_z^l(P_z^{l-1,l}(D_z^{l-1}(z^{l-1})))$  (middle), and its simply projected-prolongation  $D_z^l(P_z^{l-1,l}(z^{l-1}))$  (right). Here, the last projection  $D_z^l$  is due to the projection that is implicitly given by the definition of  $\Sigma$ .

### 3.2 Modified V-Cycle/Successive-Refinement Method

As was observed in the previous subsection, the restriction/prolongation in combination with the projection causes artificial errors that might lead to additional correction steps by the modulus fixed-point iteration on the next considered level  $l$ . Specifically, several redundant steps could be avoided and are needed only to undo the effects of the inappropriate smoothing effect introduced by (13) and (14). Therefore, instead of the full multigrid method, we propose to simply use the **modified MG $\mathbf{V}$**  method without the `solveANF` in line 4, where the number of maximum iterations is set to infinity and the maximum of MG $\mathbf{V}$  cycles equals one (i.e.,  $maxiter = \infty$  and  $maxmgv = 1$ ). In other words, one performs a V-cycle, where the modulus fixed-point iteration is fully converged to its limit  $\mathbf{z}_*$  on each level  $l$  before it is prolonged and projected on the next finer level  $l+1$  after eventually restricting the initial approximation to the coarsest level  $l_{min}$ , such that  $\delta u_* = \delta u(\mathbf{z}_*)$  then solves  $G(u + \delta u_*) = 0$  on the finest grid  $l_{max}$ .

Obviously, the most expensive part of this algorithm is the solution of the ANF in **solveANF**, which requires the solution of the linear equation

$$A^l z_{new}^l := [\mathcal{I} - \mathcal{S}\Sigma_{z_{old}^l}]z_{new}^l = (a - ZJ^{-1}b) := q^l$$

for every modulus fixed-point iteration step **FPIter** on each level  $l$ , in order to determine the next switching variables  $z_{new}^l$  based on the previous estimate  $z_{old}^l$  or the computation of  $\delta u^l$  according to (7). The latter linear equations can be solved again in a multigrid fashion, which then leads to the nested approach that was also depicted in Figure 1. However, the same objections concerning the undesired smoothing effect of the MG-cycle for the linear solve partly hold true. Another improvement that can be applied for all methods, which might pay off in the case of an expensive function evaluation, is that the ANF is evaluated only once on the finest level and then gets restricted to the coarser grids. Also, the ANF does not need to be reevaluated from scratch after updating  $u = u + \delta u_*$  after one **modified MG $\mathbf{V}$**  cycle; instead one can set  $a := a + Z\delta u_*$  and  $b := b + J\delta u_*$  to find the new residual approximation  $0 = G_u(\delta u_* + \delta u)$  for the next cycle<sup>3</sup> in the piecewise linear case.

---

<sup>3</sup>Note that in general  $G_u(\delta u_* + \delta u) \neq G_{u+\delta u_*}(\delta u)$  if  $G_x$  denotes the piecewise linearization of a nonlinear, nonsmooth function  $F$  at  $x$  such that the proposed update of  $a$  and  $b$  is, in fact, needed to find the exact residual equation of the ANF at  $u$ .

## 4 Numerical Experiments

### 4.1 Linear Complementary Problem

The first example is the linear complementarity problem in 2D,

$$0 \leq u(\omega) - \varphi(\omega) \perp -\Delta u(\omega) - f(\omega) \geq 0, \quad \text{for almost all } \omega \in \Omega, \quad (16)$$

where  $\omega = (\omega_1, \omega_2)$  is a vector in  $\Omega$ , which denotes the domain that is given by the unit square  $\Omega = [0, 1]^2$  in  $\mathbb{R}^2$ . The forcing term  $f = f_r$  for the negative Laplace operator  $-\Delta$  and the lower bound  $\varphi$  are defined for all  $\omega = (\omega_1, \omega_2) \in \Omega$  by

$$f_r(x) = \sin(r\pi\omega_1)\sin(r\pi\omega_2) \quad \text{and} \quad \varphi(\omega) = -\mathcal{I}_{\{\omega \in \Omega_B\}}(\omega),$$

where  $r \in \mathbb{N}$  and subsets  $\Omega_B \in \{\Omega_\circ, \Omega_\square, \Omega_\boxplus\}$  of  $\Omega$ . The three different choices  $\Omega_\circ, \Omega_\square, \Omega_\boxplus$  for the set  $\Omega_B$  and the corresponding induced lower bound functions  $\varphi$  are visualized in Figure 3 besides the forcing term  $f_3$ , which was used for the presented numerical results. The sets were specifically chosen to provide a small sample of different scenarios of lower bounds; that is, the set  $\Omega_\circ$  is the ball centered at 0 with radius  $\frac{1}{2}$  that has a smooth boundary within  $\Omega$ , in contrast to  $\Omega_\square$ , which is the closed square  $[\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{4}, \frac{1}{2}]$ . The set  $\Omega_\boxplus$  has a repeated checkerboard pattern and represents problems where the discretization needs to be fine enough to resolve the behavior of the lower bound. It is defined to be the subset of the unit square  $\Omega$ , where the function  $f_{15}$  has negative values.

### 4.2 Nonlinear Complementarity Problem

The proposed algorithms were also applied to the complementarity problem

$$0 \leq u(\omega) - \varphi(\omega) \perp H(u(\omega)) \geq 0, \quad \text{for almost all } \omega \in \Omega, \quad (17)$$

with the previously defined quantities, where the right part of the condition (16) was replaced by the nonlinear expression  $H(u) = -\nabla^2 u - f_3 - \Psi(u)$ . This involves the additional term  $\Psi(u)$ , which was chosen to be the operator

$$\Psi(u) := \sin(\pi\omega_1) \sin(2\pi\omega_2) * \left( \frac{u^3(\omega)}{3} - \frac{u^2(\omega)}{2} - 1 \right) \quad \forall \omega = (\omega_1, \omega_2) \in \Omega$$

such that  $\Psi$ , and thus  $f_3 + \Psi$  vanish on the boundaries of  $\Omega = [0, 1]^2$ .

### 4.3 Numerical Results

The numerical results were computed on a Lenovo Thinkpad X1 Carbon with Opensuse 13.1 using the simple Matlab 2013 implementation given in the Appendix. For all experiments, the same random (i.i.d.) function  $u_0$  (and  $z_0$ ) was used as an initial approximation<sup>4</sup> of the true solution  $u_*$  (and  $z_*$ ) on the finest grid level  $l_{max} = 11$ . Therefore, the partial differential equation was numerically approximated on an equidistant grid with  $N^{(l)} := 2^l + 1$  discretization points in each direction using the discrete finite-differences 2D Laplace operator matrix and zero Dirichlet boundary conditions. All problems were solved up to machine precision in the appropriate  $\mathcal{L}_2$ -norm for each discretization level  $l$ . The required number of inner modulus fixed-point iterations to solve the problems and the overall run-time in seconds are reported in Tables 1-7, which are summarized in Fig. 7.

In detail, Table 1 contains the results for the linear problem (16) and the choices  $\Omega_\circ$ ,  $\Omega_\square$ ,  $\Omega_\boxplus$  using the simple MFP (6) that was solved exactly on each of the stated discretization levels  $l$  for the random initialization. Table 2 shows the possible computational saving that can be obtained for the same setup by exploiting the given problem structure if the simple fixed-point iteration (6) is replaced by its efficient reformulation (8). Table 3 gives the result for one modified VCycle ( $maxmgv = 1$ ), where the modulus fixed-point iteration (8) for  $\mathbf{z}^l$  was converged ( $maxiter = \text{inf}$ ) on every level  $l$  starting from the solution on the coarser level  $l - 1$ , which was prolonged by (13) but without the projection (12). The results using the projection (12) can be found in Table 4. The results of the full multigrid method ( $maxiter = 2, maxmgv = 3$ ) with projection are given in Table 5. Here, the number of iterations represents the sum of all required modulus fixed-point iterations on each level needed during the full multigrid-method to achieve the same accuracy. Tables 6 and 7 summarize the results for the nonlinear problem that was solved by a sequence of modified MGV cycles and the full-multigrid method, respectively. On average, three modified MGV cycles (nonsmooth Newton-steps) are sufficient to achieve the desired residual accuracy. The required number  $maxmgv$  of MGV cycles for the full multigrid method turns out to be three, also. A visualization of the solution  $u_*$  and the final switching variables  $\mathbf{z}_*$  for the level  $l = 6$  can be found in Fig. 4, Fig. 5, and Fig. 6.

---

<sup>4</sup>if needed, the random initialization on the coarser grid was obtained by the restriction (14) to allow for comparable results

The nested multigrid method was not considered in the numerical experiments in order to avoid any interference, but it is likely to provide further speedup. Instead, all the occurring linear equations within the `solveANF` method were solved by Matlab's backslash operator.

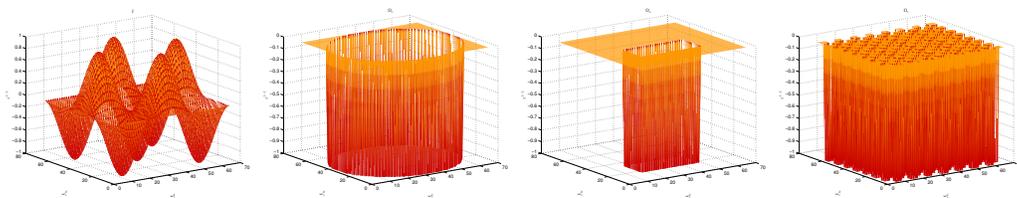


Figure 3: Right-hand-side  $f_3$ , and the indicator functions induced by  $\Omega_\circ$ ,  $\Omega_\square$ , and  $\Omega_\boxplus$  (from left to right).

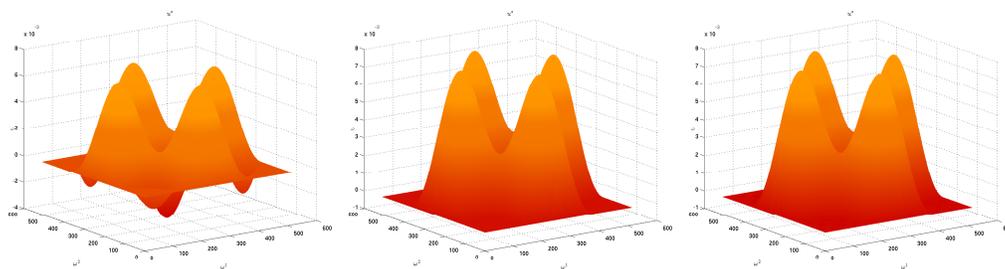


Figure 4: Solution  $u$  of the linear complementarity problem for the lower bounds induced by  $\Omega_\circ$ ,  $\Omega_\square$ , and  $\Omega_\boxplus$  (from left to right).

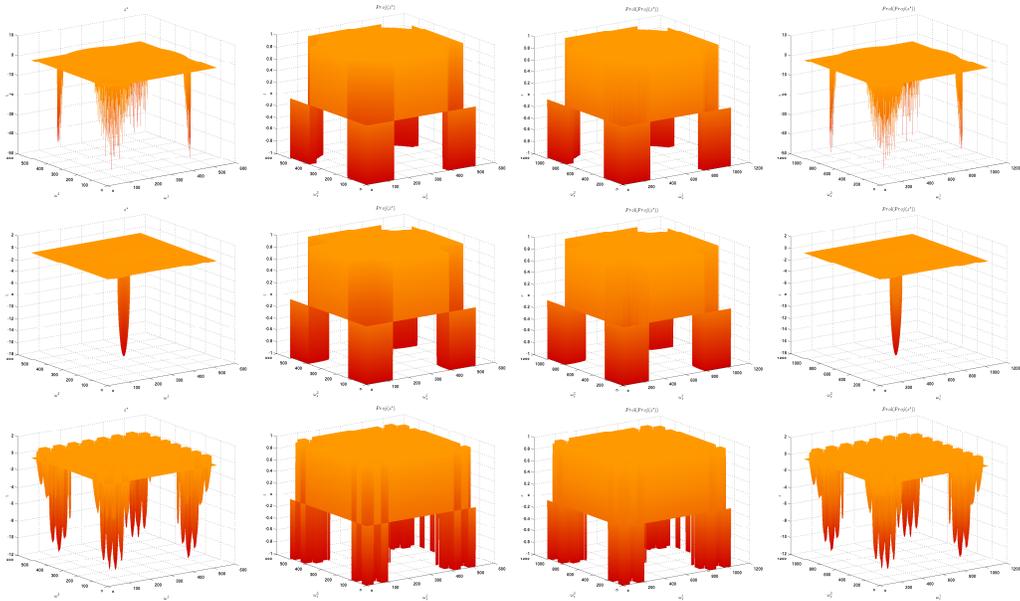


Figure 5: Visualization of the final switching function  $z$ ,  $sign(z)$ , prolongation of  $sign(z)$ , and the simple prolongation of  $z$  for the linear problem (from left to right) and the three choices  $\Omega_\circ$ (top),  $\Omega_\square$ (middle), and  $\Omega_\boxplus$ (bottom).

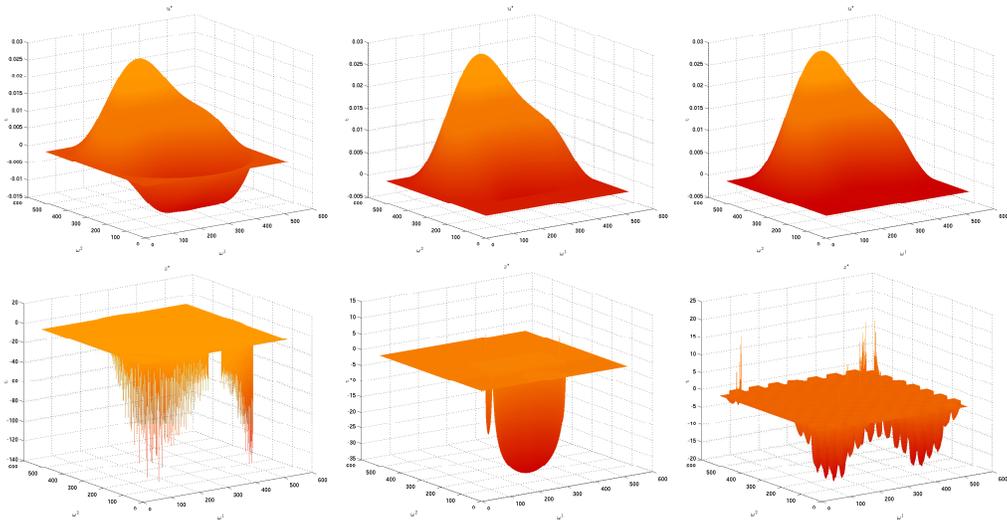


Figure 6: Visualization of the final solution  $u$  (top) and the final switching function  $z$  (bottom) for the nonlinear problem  $\Omega_\circ$ ,  $\Omega_\square$ , and  $\Omega_\boxplus$  (left to right).

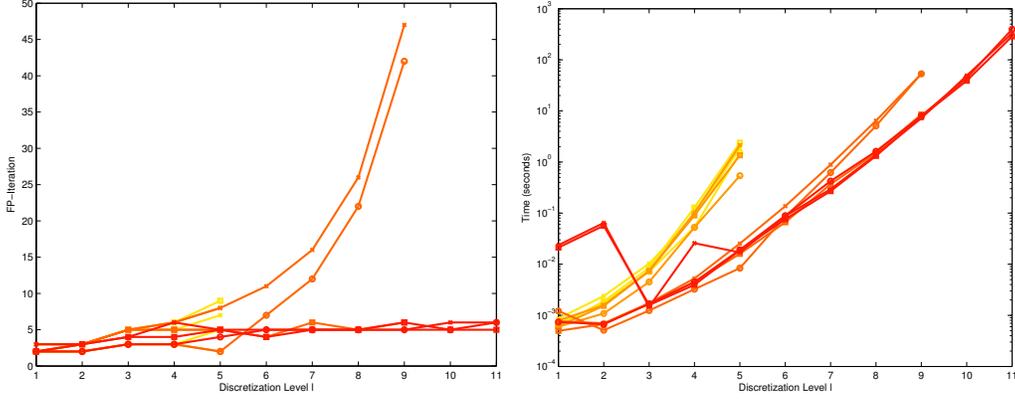


Figure 7: Visualization of the number of required FPIter (left) and overall run-time in seconds (right) for each discretization level  $l$  reported in Table 1 (yellow), Table 2 (light orange), Table 3 (dark orange), and Table 4 (red) for the three different lowerbounds  $\Omega_{\circ}$  ( $\circ$ ),  $\Omega_{\square}$  ( $\square$ ), and  $\Omega_{\boxplus}$  ( $\times$ ).

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	2	6.960000e-04	2	6.350000e-04	3	8.670000e-04
2	2	1.733000e-03	3	1.873000e-03	3	2.383000e-03
3	3	7.161000e-03	5	8.501000e-03	5	1.032100e-02
4	3	5.299700e-02	6	1.286270e-01	5	8.884200e-02
5	5	1.509989e+00	9	2.392125e+00	7	1.860127e+00

Table 1: Results for the linear problem (16) using the simple MFP (6), without projection, for the choices  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$  solved on each level  $l$ .

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	2	6.020000e-04	2	6.100000e-04	3	7.960000e-04
2	2	1.085000e-03	3	1.532000e-03	3	1.571000e-03
3	3	4.504000e-03	5	7.382000e-03	5	7.394000e-03
4	3	5.249300e-02	5	9.025300e-02	6	1.022870e-01
5	2	5.411540e-01	5	1.353975e+00	8	2.140963e+00

Table 2: Results for the linear problem (16) using the efficient MFP (8), without projection, standard prolongation (13), and simple restriction (15), for the choices  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$  solved on each level  $l$ .

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	2	1.216000e-03	2	4.970000e-04	3	7.640000e-04
2	2	5.120000e-04	3	6.750000e-04	3	7.010000e-04
3	3	1.240000e-03	5	1.685000e-03	5	1.738000e-03
4	3	3.251000e-03	5	4.509000e-03	6	5.278000e-03
5	2	8.380000e-03	5	1.596500e-02	8	2.536600e-02
6	7	8.506000e-02	4	6.641000e-02	11	1.369770e-01
7	12	6.216080e-01	6	3.692940e-01	16	8.856710e-01
8	22	5.112441e+00	5	1.510281e+00	26	6.447225e+00
9	42	5.328563e+01	6	8.533525e+00	47	5.336615e+01

Table 3: Results for the linear problem (16) using the efficient MFP (8) using one **modified VCycle**, without projection, standard prolongation (13), and simple restriction (15), for  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$ .

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	2	7.390000e-04	2	2.122500e-02	3	2.366400e-02
2	2	6.720000e-04	3	5.671400e-02	3	6.433000e-02
3	3	1.681000e-03	4	1.592000e-03	4	1.466000e-03
4	3	4.496000e-03	4	3.956000e-03	6	2.588600e-02
5	4	1.912300e-02	5	1.876200e-02	5	1.699400e-02
6	5	8.937600e-02	4	7.618000e-02	5	8.776500e-02
7	5	4.231790e-01	5	2.698110e-01	5	2.943600e-01
8	5	1.617928e+00	5	1.319262e+00	5	1.379170e+00
9	5	8.022691e+00	6	7.749272e+00	5	7.195976e+00
10	5	4.403858e+01	5	3.904835e+01	6	4.913688e+01
11	6	3.958369e+02	5	2.878728e+02	6	3.317204e+02

Table 4: Results for the linear problem (16) using the efficient MFP (8) using one **modified VCycle**, with projection, standard prolongation (13), and simple restriction (15), for  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$ .

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	31	1.324390e-01	33	2.058090e-01	40	7.987100e-02
2	55	1.672640e-01	59	1.668160e-01	71	1.495700e-01
3	47	1.224120e-01	52	1.150760e-01	64	1.383930e-01
4	39	2.049110e-01	44	3.029320e-01	56	2.337380e-01
5	32	8.836760e-01	36	4.422830e-01	48	5.185170e-01
6	24	7.157060e-01	28	1.186067e+00	40	1.576971e+00
7	20	3.216956e+00	20	3.502826e+00	32	5.414041e+00
8	16	1.055167e+01	16	8.553483e+00	24	1.576412e+01
9	12	3.098891e+01	12	2.963045e+01	16	4.225638e+01
10	8	1.139729e+02	8	1.176420e+02	8	1.208749e+02
11	4	4.012236e+02	4	4.682104e+02	4	3.886089e+02

Table 5: Results for the linear problem (16) using the efficient MFP (8) using the **Full multigrid**, with projection, standard prolongation (13), and simple restriction (15), for the choices  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$ .

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	(2,2,2)	3.648100e-02	(2,2,2)	7.651100e-02	(3,3,3)	3.660200e-02
2	(2,2,2)	1.039410e-01	(2,2,2)	1.501190e-01	(4,4,4)	1.378060e-01
3	(3,3,3)	4.527000e-03	(4,4,4)	6.986000e-03	(3,3,3)	3.965000e-03
4	(4,4,4)	5.646800e-02	(4,4,4)	1.094530e-01	(7,7,7)	8.923100e-02
5	(4,4,4)	5.401000e-02	(5,5,5)	4.828000e-02	(5,5,6)	6.527600e-02
6	(5,4,4)	3.606830e-01	(5,5,5)	1.839210e-01	(5,5,5)	8.170460e-01
7	(6,6,6)	1.082465e+00	(5,5,5)	8.430270e-01	(5,5,5)	9.230070e-01
8	(4,4,4)	4.097798e+00	(5,5,5)	4.181277e+00	(5,5,5)	4.203123e+00
9	(5,5,5)	2.617473e+01	(5,6,6)	2.201599e+01	(6,6,6)	2.343003e+01
10	(7,5,5)	1.584398e+02	(6,6,6)	1.361256e+02	(6,6,6)	1.321681e+02
11	(8,9,9)	1.588478e+03	(6,6,6)	9.912651e+02	(6,6,6)	9.558356e+02

Table 6: Results for the nonlinear problem (17) using the efficient MFP (8) using **modified V-cycle**, with projection (12), standard prolongation (13), and simple restriction(15), for  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$ , where the number of fixed-point iterations is given for each of the three MGCV cycles separately for all levels  $l$ .

$l$	# Iter	Time $\Omega_{\circ}$	# Iter	Time $\Omega_{\square}$	# Iter	Time $\Omega_{\boxplus}$
1	35	7.583000e-02	33	8.437100e-02	58	5.103100e-02
2	63	4.279350e-01	59	1.839830e-01	72	7.344700e-02
3	55	5.484400e-01	52	2.199040e-01	63	1.222720e-01
4	48	6.964250e-01	44	3.449400e-01	56	2.062820e-01
5	40	1.019395e+00	36	5.527570e-01	48	4.886610e-01
6	32	2.182906e+00	28	1.381449e+00	40	1.219615e+00
7	24	5.050670e+00	20	3.431927e+00	32	3.935502e+00
8	16	1.086553e+01	16	9.884764e+00	24	1.330366e+01
9	12	3.491688e+01	12	3.278858e+01	16	4.129180e+01
10	8	1.322932e+02	8	1.128040e+02	8	1.087870e+02
11	4	5.399108e+02	4	3.986173e+02	4	4.192130e+02

Table 7: Results for the nonlinear problem (17) using the efficient MFP (8) using the **Full multigrid**, with projection, standard prolongation (13), simple restriction(15), for the choices  $\Omega_{\circ}$ ,  $\Omega_{\square}$ ,  $\Omega_{\boxplus}$ .

From the results one can deduce (at least for the stated problem) that each of the proposed modifications increases the efficiency of the suggested methods. In particular, Tables 1 and 2 show that it (as usual) pays off to do some analysis and exploit the structure of the problem rather than blindly applying piecewise linear differentiation on a black-box model and the results for solving piecewise linear systems given in [25]. Table 3 indicates that using the prolonged solution of the coarser grid as an initial guess for a finer level helps decrease the number of solveANF cycles, which can be further reduced if the prolongation is done in a more appropriate way (see Table 4), namely, by using the projection (12) before the prolongation (13). In fact, the number of inner iterations seems to be constant, with an average around 6 indicating mesh-independent behavior independent of the lower bound function  $\varphi$ . A similar result can be seen for the nonlinear problem, where the computation of every nonsmooth Newton-direction by the modified VCycle requires the same number of fixed-point iterations to converge. The small fluctuations can be explained by the artificial error of the projection/prolongation, which is still not 100% appropriate for the nonsmooth/discontinuous functions  $\mathbf{z}$  and requires a small additional number of correction steps/fixed-point iterations. Comparing Table 4 and Table 5, we see that the modified V-cycle is competitive with the full multigrid method for the linear problem in contrast to the nonlinear case (Tables 6 and 7), where the **full multigrid** method

seems to perform better, mainly because of the smaller number of fixed point iterations on the finer grid.

## 5 Conclusion

In this paper, a multigrid method for nonsmooth problems was considered. It is based on ideas of the original multigrid method and techniques from piecewise linear differentiation in finite dimensions. We assumed that the considered problems were abs-decomposable and their piecewise linearization can be written in abs-normal form. The abs-normal form allows for various iterative schemes that can be applied to solve the resulting piecewise linear equation using some additional switching variables for each discretization level. The overall method is based on a successive refinement strategy, where (approximate) solutions of the additional switching variables on a coarser levels are used as initial approximations on a finer level. Since the switching variables themselves are usually nonsmooth or even discontinuous, their prolongation from the coarser to the finer level needs special treatment, as it was exemplified for a simple complementarity problem. It appears that a “correct” prolongation, restriction, and projection of these variables provide (almost) mesh-independent results allowing for an efficient solution of the problem with up to 4,2 million degrees of freedom on the finest discretization level in a reasonable time, which is comparable to the results<sup>5</sup> of the different approaches presented in [26, 9, 11, 15]. The resulting method and strategies could be used to solve time-dependent differential equations together with the generalized mid-point rule [5], for example if the time integration involves some nonsmooth UPWIND/QUICK scheme[27], which can be stated in terms of min and max. However, it is not clear whether the small average number of modulus fixed-point iterations observed for the considered example does hold true in general. If not, this would suggest further investigation of a more appropriate projection and prolongation strategy for nonsmooth/discontinuous functions. Also, different fallback options for singular  $J$  instead of using only the extended ANF should be investigated. Furthermore, different strategies for the nonlinear case should be part of future research directions, for example, comparing the presented Newton-MG with an MG-Newton scheme and an additional line-search for the globalization.

---

<sup>5</sup>if not better

## Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357. The author is grateful to Zichao (Wendy) Di for several comments that helped improve the notation and Gail Pieper for her patient and precious corrections.

## Appendix - Matlab Codes

This section contains parts of a simple MATLAB implementation for the proposed methods that were used to compute the presented results. The methods coincide with the algorithms except for the difference in **solveANF** and **FPIter**, where the computation of  $\delta u_{new}$  was detached from **FPIter** and is computed only once at the end of the fixed-point iteration, in order to avoid obsolete linear-solves and increase the overall efficiency.

Listing 1: V-Cycle method

---

```
function [ z, Du ] = VCycle(1, lmin, z, u, Du, tol, maxiter)
    [a, b, Z, L, J, Y] = evalANF(1, u, Du);
    if l<=lmin
        maxiter=1e3;
    else
        [z, Du] = solveANF(1, a, b, Z, L, J, Y, z, Du, tol, maxiter);
        z=projectz(z,l);
        z=restrictz(z,l);
        u=restrictu(u,l);
        [z, Du] = VCycle(1-1, lmin, z, u, Du, tol, maxiter);
        z=projectz(z,l-1);
        z=prolongz(z,l-1);
        Du=prolongu(Du,l-1);
    end
    [z, Du] = solveANF(1, a, b, Z, L, J, Y, z, Du, tol, maxiter);
return
```

---

Listing 2: Multigrid cycle

---

```
function [ z, u ] = MGV(1, lmin, z, u, tol, maxiter, maxmgv)
    flag=1; i=1;
    Du = 0.*u(:);
    while (flag~=0)&&(i<=maxmgv)
        [ z, Du] = VCycle(1, lmin, z, u, Du, tol, maxiter);
        u(:) = u(:) + Du(:);
        i=i+1;
        if (scalprod( 1, Du(:), Du(:)) <= tol)
            flag=0;
        end
    end
return
```

---

### Listing 3: Full multigrid method

---

```
function [ z, u ] = FullMG(lmax, lcur, lmin, z, u, tol, maxiter, maxmgv)
for l=lcur:-1:lmin+1
    z=projectz(z,l);
    z=restrictz(z,l);
    u=restrictu(u,l);
end
for l=lmin:1:lmax
    [ z, u ] = MGv(l, lmin, z, u, tol, maxiter, maxmgv);
    if l<lmax
        z=projectz(z,l);
        z=prolongz(z,l);
        u=prolongu(u,l);
    end
end
return
```

---

### Listing 4: Modified V-Cycle

---

```
function [ z, Du ] = modVCycle(l, lmin, z, u, Du, tol)
[a, b, Z, L, J, Y]=evalANF(l, u, Du);
if l>lmin
    z=projectz(z,l);
    z=restrictz(z,l);
    u=restrictu(u,l);
    [z, Du]= modVCycle(l-1, lmin, z, u, Du, tol);
    z=projectz(z,l-1);
    z=prolongz(z,l-1);
    Du=prolongu(Du,l-1);
end
[z, Du] = solveANF(l, a, b, Z, L, J, Y, z, Du, tol, inf);
return
```

---

### Listing 5: Modified Multigrid cycle

---

```
function [ z, u ] = modMGV(l, lmin, z, u, tol)
Du=0.*u(:);
[ z, Du ] = modVCycle(l, lmin, z, u, Du, tol);
u(:) = u(:) + Du(:);
return
```

---

### Listing 6: solveANF method

---

```
function [z, Du]= solveANF(l, a, b, Z, L, J, Y, z, Du, tol, maxiter)
iter=0; flag=1; tic;
while ((flag~=0)&&(iter<maxiter))
    iter=iter+1;
    [znew] = FPIter( a, b, Z, L, J, Y, z(:), Du(:));
    t = z(:)-znew(:); res=scalprod(1,t(:),t(:));
    if res<tol
        flag = 0;
    end
    z=znew;
end
Du=-J\b+Y*abs(znew(:)); %For efficiency reason
return
```

---

### Listing 7: Plain FPIter method

---

```
function [z, Du] = FPiter( a, b, Z, L, J, Y, z, Du)
    rhs=a(:)-Z*(J\b(:));
    Sigma=spdiags(sign(z(:)),0,length(a(:)),length(a(:)));
    I=spdiags(ones(length(a(:))),0,length(a(:)),length(a(:)));
    A=I-(L-Z*(J\Y))*Sigma;
    z=A\rhs(:);
return
```

---

### Listing 8: Efficient FPIter method

---

```
function [z, Du] = FPiter( a, b, Z, L, J, Y, z, Du)
    rhs=J*a(:)-Z*b(:);
    Sigma=spdiags(sign(z(:)),0,length(a(:)),length(a(:)));
    A=J-Z*Sigma;
    z=A\rhs(:);
return
```

---

### Listing 9: Projection z

---

```
function [ z ]= projectz(z,l)
    z=sign(z);
return
```

---

### Listing 10: Prolongation)

---

```
function [xnew] = prolong( x, l )
    n=2^l+1;
    nnew = 2*(n-1)+1;
    xnew = zeros(nnew,nnew);
    for j=1:n
        for i=1:n
            xnew(2*(i-1)+1,2*(j-1)+1) = x(i,j);
        end
    end
    for j=1:2:nnew
        for i=2:2:nnew-1
            xnew(i,j) = 0.5*(xnew(i-1,j) + xnew(i+1,j));
        end
    end
    for j=2:2:nnew-1
        for i=1:nnew
            xnew(i,j) = 0.5*(xnew(i,j-1) + xnew(i,j+1));
        end
    end
return
```

---

## References

- [1] J. Albersmeyer and M. Diehl. The lifted Newton method and its application in optimization. *SIAM J. on Optimization*, 20(3):1655–1684, Jan. 2010.

- [2] Z.-Z. Bai and L.-L. Zhang. Modulus-based synchronous multisplitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications*, 20(3):425–439, 2013.
- [3] Z.-Z. Bai and L.-L. Zhang. Modulus-based synchronous two-stage multisplitting iteration methods for linear complementarity problems. *Numer. Algorithms*, 62(1):59–77, Jan. 2013.
- [4] D. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific Series in Optimization and Neural Computation. Athena Scientific, 1996.
- [5] P. Boeck, B. Gompil, A. Griewank, R. Hasenfelder, and N. Strogies. Experiments with generalized midpoint and trapezoidal rules on two nonsmooth ODEs. 2015.
- [6] L. Brugnano and V. Casulli. Iterative solution of piecewise linear systems. *SIAM Journal on Scientific Computing*, 30(1):463–472, 2008.
- [7] A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000.
- [8] R. Cottle, J. Pang, and R. Stone. *The Linear Complementarity Problem*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 1992.
- [9] T. De Luca, F. Facchinei, and C. Kanzow. A semismooth equation approach to the solution of nonlinear complementarity problems. *Mathematical Programming*, 75(3):407–439, 1996.
- [10] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente. Trust-region interior-point sqp algorithms for a class of nonlinear programming problems. *SIAM Journal on Control and Optimization*, 36(5):1750–1794, 1998.
- [11] A. Fischer. Solution of monotone complementarity problems with locally lipschitzian functions. *Mathematical Programming*, 76(3):513–532, 1997.
- [12] A. Griewank. On stable piecewise linearization and generalized algorithmic differentiation. *Optimization Methods and Software*, 28(6):1139–1178, 2013.

- [13] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*. SIAM e-books. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2008.
- [14] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2003.
- [15] P. T. Harker and J.-S. Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Math. Program.*, 48(2):161–220, Sept. 1990.
- [16] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.
- [17] C. Kanzow. *Numerik linearer Gleichungssysteme: Direkte und iterative Verfahren*. Springer-Lehrbuch. Springer, 2004.
- [18] K. A. Khan and P. I. Barton. Evaluating an element of the Clarke generalized Jacobian of a composite piecewise differentiable function. *ACM Trans. Math. Softw.*, 39(4):23:1–23:28, July 2013.
- [19] D. M. W. Leenaerts and W. M. V. Bokhoven. *Piecewise Linear Modeling and Analysis*. Kluwer Academic Publishers, Norwell, MA, US, 1998.
- [20] O. Mangasarian. Linear complementarity problems solvable by a single linear program. *Mathematical Programming*, 10(1):263–270, 1976.
- [21] O. L. Mangasarian and J. S. Pang. Exact penalty functions for mathematical programs with linear complementarity constraints. *Optimization*, 42:96–06.
- [22] J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 2013.
- [23] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.
- [24] S. Scholtes. *Introduction to Piecewise Differentiable Equations*. Springer-Briefs in Optimization. Springer, Dordrecht, 2012.

- [25] T. Streubel, A. Griewank, M. Radons, and J.-U. Bernt. Representation and analysis of piecewise linear functions in abs-normal form. In C. Pötzsche, C. Heuberger, B. Kaltenbacher, and F. Rendl, editors, *System Modeling and Optimization*, volume 443 of *IFIP Advances in Information and Communication Technology*, pages 327–336. Springer Berlin Heidelberg, 2014.
- [26] M. Ulbrich. *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*. Society for Industrial and Applied Mathematics, 2011.
- [27] H. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007.

**Government License:**

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.