

Monotone multigrid methods based on element agglomeration coarsening away from the contact boundary for the Signorini's problem

Ana H. Iontcheva and Panayot S. Vassilevski^{*,†}

Center for Applied Scientific Computing, UC Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551, U.S.A.

SUMMARY

Two multilevel schemes for solving inequality constrained finite element second-order elliptic problems, such as the Signorini's contact problem, are proposed and studied. The main ingredients of the schemes are that first they utilize element agglomeration coarsening away from the constraint set (boundary), which allows for easy construction of coarse level approximations that straightforwardly satisfy the fine-grid constraints. Second important feature of the schemes is that they provide monotone reduction of the energy functional throughout the multilevel cycles. This is achieved by using monotone smoothers (such as the projected Gauss–Seidel method) and due to the fact that the recursive application of the two-grid schemes is also monotone. The performance of the resulting methods is illustrated by numerical experiments on a model 2D Signorini's problem. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: contact problem; algebraic multigrid; FAS; subspace minimization; algebraic coarsening

1. INTRODUCTION

The solution of finite element second-order elliptic problems with inequality constraints is the topic of the present paper. This problem is closely related and is a natural extension of the solution of linear (unconstrained) second-order finite element problems. For the latter class of problems efficient multigrid methods exist for geometrically refined meshes as well as algebraic multigrid methods (AMG) suitable for unstructured meshes. For AMG, see, e.g. References [1, 2], and the long list of references in Reference [3]. Extensions of multigrid methods to inequality constrained problems are found as early as in Reference [4]. They considered extensions of FAS multilevel solvers applied to the linear complementarity problems arising from free boundary problems. The difference between the linear complementarity

*Correspondence to: Panayot S. Vassilevski, Center for Applied Scientific Computing, UC Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551, U.S.A.

†E-mail: panayot@llnl.gov

Contract/grant sponsor: University of California Lawrence Livermore National Laboratory; contract/grant number: W-7405-Eng-48

problem and the contact (Signorini's) problem, the latter we target in the present paper, is in the constrained set. Whereas in the former (considered in Reference [4], see also Reference [5]) it is the whole set of indices, for the contact problem the constrained set involves degrees of freedom associated with a certain boundary C . That is, the dimension of the constrained set of degrees of freedom is $\mathcal{O}(h^{-(d-1)})$ for contact problem, versus $\mathcal{O}(h^{-d})$ for linear complementarity problems, that is, one dimension less. Here $d=2$ or $d=3$ is the dimension of the problem domain Ω where the second order elliptic pde is posed, and $h \mapsto 0$ is the mesh-size.

Mathematically, the problem under consideration can be stated as a quadratic functional constrained minimization problem. One seeks a function u from a finite element space V which minimizes the quadratic functional $J(u) = \frac{1}{2}a(u, u) - (f, u)$ subject to $u \in K = \{v \in V : v(x) \leq g(x) \text{ for all } x \in C\}$.

Here $a(u, v) = \int_{\Omega} k(x) \nabla u \cdot \nabla v dx$ is a model second-order elliptic bilinear form, Ω is a polygonal domain (or \mathbb{R}^3 polytope) covered exactly by the elements T from a given finite element triangulation \mathcal{T}_h . To be specific, the space V is a conforming, Lagrangian finite element space of piecewise linear basis functions. In practice, the convex set K is considered consisting of all vectors $\mathbf{v} = (v_i)$ such that for degrees of freedom $x_i \in C$ satisfy $v_i \leq g_i$. Here, $\mathbf{g} = (g_i)$ is a given vector defined for indices from C . One can relate the contact problems to linear complementarity problem, by considering a reduced form of the contact problem. This reduced problem involves vectors of size of the constrained set C . That is, then all degrees of freedom of the reduced problem are constrained, which is the case in linear complementarity problems. In a matrix–vector form the above problem reads

$$J(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T A \mathbf{v} - \mathbf{b}^T \mathbf{v} \mapsto \min, \quad \mathbf{v} \in \mathbb{R}^n \quad (1)$$

subject to the inequality constraints

$$v_i \leq g_i \quad \text{for all } i \in \Delta$$

Here Δ is a given subset of the index set $\{0, 1, 2, \dots, n-1\}$ and $\mathbf{g} = (g_i)$ is a given vector defined for indices in Δ . Finally, A is a given symmetric positive definite matrix, the stiffness matrix corresponding to the given bilinear form $a(\cdot, \cdot)$ and the finite element space $V = V_h$.

In the present paper we modify the element agglomeration AMGe method from Reference [6] (which is a special version of Reference [7]) to define a sequence of coarse approximations to the model finite element contact problem formulated above. The main feature of the modification is that the coarse finite element spaces utilize elements (agglomerates of fine-grid elements) which are coarse away from the contact boundary C . That is, the degrees of freedom on C stay on all coarse grids, which allows for a straightforward transfer of data between the grids without violating the inequality constraints.

On the basis of the coarsened away algebraically constructed coarse finite element spaces, we are able to formulate two multilevel schemes to solve the fine-grid constrained minimization contact problem. Our first method is based on the classical FAS [8] and is very similar to the projected full approximation method (FPAS) from Reference [4]. With our construction of the coarse spaces we are able to prove that our FAS as well as a related multilevel subspace minimization algorithm are monotone. This is possible for smoothers which are monotone (such as the projected Gauss–Seidel method), that is, smoothers that decrease the energy functional monotonically. The proposed extension of the AMGe method can be applied to bodies with complex geometry.

Geometric multigrid methods for solving constrained minimization problems that are monotone are found in References [9–11]. There is a variety of conventional methods developed for the solution of the Signorini's problem. Those methods (similar to conventional methods applied to linear second-order finite element elliptic equations) exhibit convergence that deteriorates as the mesh size $h \rightarrow 0$. We can mention methods based on active set strategies [12–14] where one alternates between approximating the contact set and solving a linear problem with fixed contact set. The conventional methods can be used as coarse-grid solvers and those that are monotone are suitable for smoothing in a multilevel scheme. Other class of methods is the penalty methods [15–17] where one ends up with an unconstrained minimization problem for a functional which depends on a penalty parameter. In general, the thus obtained solution is only an approximation to the problem of interest. Other methods that can have a proven mesh independent convergence are the so-called projection methods (cf. References [17, 18]). Roughly speaking these methods are based on an iterative process applied to the related linear (unconstrained) problem and after every iteration the current intermediate iterate is projected onto the constrained set. To achieve good convergence efficient preconditioning is needed for the (unconstrained) linear problem. Typically, the projection methods lead to an inner–outer process which involves (approximate) evaluation of the projection [18]. This is a viable approach that is left for further research.

The remainder of the paper is organized as follows. We begin with a general two-grid iteration scheme, which under certain condition is proved to be monotone. This is found in Section 2. The specific element agglomeration coarsening procedure is introduced in Section 3. A recursive application of the two-level method is discussed in the following two sections where the Multilevel Subspace Minimization Algorithm and the FAS Constrained Optimization Algorithm are summarized, respectively, in Sections 4 and 5. Finally, numerical experiments for a model 2D Signorini's problem are presented in Section 6 and some conclusions are drawn at the end.

2. A MONOTONE TWO-GRID SCHEME

To define a two-grid scheme we need a coarse space and a smoothing procedure. The coarse space we will consider will satisfy an important (special) property. The smoothing procedure will be monotone (to be described in what follows).

Consider a coarse space $V_H \subset V = V_h$ and let P be an interpolation matrix that transfers the coarse coefficient vector \mathbf{v}_c of a function $v_H \in V_H$ to a fine-grid coefficient vector $\mathbf{v} = P\mathbf{v}_c$ of v_H now as an element of V . Our main assumption is that

$$P = \begin{bmatrix} P_{fc} \\ I \end{bmatrix}$$

that is, the values of \mathbf{v}_c at coarse degrees of freedom (dofs) are interpolated as identity on the fine grid. We also assume that constrained dofs $x_i \in C$ (or equivalently $i \in \Delta$) are all present on the coarse grid. This implies that

$$P\mathbf{v}_c|_{\Delta} = \mathbf{v}_c|_{\Delta} \quad (2)$$

2.1. Projected Gauss–Seidel

We next describe the projected Gauss–Seidel method. Mathematically, it can be described as a sequence of one-dimensional minimization problems. Consider the functional $J(\mathbf{v})$. Given a current iterate $\mathbf{v}=(v_i)$ which satisfies the constraints, one varies only a single component v_i (the remaining ones $v_j, j \neq i$ are kept fixed). This leads to a scalar quadratic function $\varphi(v_i)=J(v_i\mathbf{e}_i+\mathbf{v}^0)$ where $\mathbf{v}^0=\sum_{j \neq i} v_j\mathbf{e}_j$, and $\{\mathbf{e}_i\}$ are the unit co-ordinate vectors. If $i \in \Delta$ then we have to satisfy the constraint $v_i \leq g_i$. Thus we obtain a problem of finding the minimum of a quadratic function subject to a simple inequality constraint, that is, with $x=v_i, a=\mathbf{e}_i^T A \mathbf{e}_i > 0, b=\mathbf{e}_i^T(\mathbf{b}-A\mathbf{v}^0)$, a constant c and $d=g_i$, we have to solve

$$\varphi(x)=\frac{1}{2}ax^2 - bx + c \mapsto \min$$

subject to

$$x \leq d$$

The solution is $x=b/a$ if $b/a \leq d$, or $x=d$ otherwise. The new iterate then is $\mathbf{v}:=\mathbf{v}^0 + v_i\mathbf{e}_i$ with $v_i=x$. After a loop over all indices i one completes the projected Gauss–Seidel cycle. This procedure used iteratively is referred to as the Projected Gauss–Seidel method. One can also develop block-versions of this method (see Reference [16]) or even using overlapping blocks, by solving small dimensional constrained minimization problems for every block. An important property of the projected Gauss–Seidel is that every intermediate iterate decreases the functional, and hence after a complete cycle one has that $J(\mathbf{v}) \leq J(\mathbf{v}^{\text{initial}})$, that is, it is a monotone method.

2.2. Coarse-grid solve

Let $\mathbf{v}^{m-1}, m \geq 1$, be a current iterate for solving our fine-grid problem (1). After performing a few steps of the projected Gauss–Seidel (or any other monotone smoothing scheme) we end up with an intermediate iterate $\mathbf{v}^{m-1/2}=(v_i^{m-1/2})$ which satisfies the constraints $v_i^{m-1/2} \leq g_i, i \in \Delta$, and we also have

$$J(\mathbf{v}^{m-1/2}) \leq J(\mathbf{v}^{m-1})$$

The next iterate \mathbf{v}^m is sought as $\mathbf{v}^m=\mathbf{v}^{m-1/2} + P_C\mathbf{y}_C$ where \mathbf{y}_C is a coarse-grid vector such that the resulting coarse-grid energy functional is minimized. More specifically, since Δ is a subset of the set of coarse-grid degrees of freedom, one can solve the following coarse-grid minimization problem:

Find \mathbf{y}_C such that

$$\frac{1}{2}(\mathbf{v}^{m-1/2} + P_C\mathbf{y}_C)^T A(\mathbf{v}^{m-1/2} + P_C\mathbf{y}_C) - \mathbf{b}^T(\mathbf{v}^{m-1/2} + P_C\mathbf{y}_C) \mapsto \min \quad (3)$$

subject to the constraints $(\mathbf{y}_C)_i \leq g_i - v_i^{m-1/2}$ for $i \in \Delta$.

Let $A_C=P^T A P$ and $\mathbf{b}_C=P^T(\mathbf{b}-A\mathbf{v}^{m-1/2})$. Then (3) is equivalent to the following coarse quadratic constrained minimization problem:

$$J_C(\mathbf{y}_C)=\frac{1}{2}\mathbf{y}_C^T A_C \mathbf{y}_C - \mathbf{b}_C^T \mathbf{y}_C \mapsto \min$$

subject to the inequality constraints

$$(\mathbf{y}_C)_i \leq g_i - (v^{m-1/2})_i \quad \text{for all } i \in \Delta$$

We summarize the two-grid iteration method:

Algorithm 2.1 (*Two-grid minimization method*). Given an iterate \mathbf{v}^{m-1} , compute the next iterate \mathbf{v}^m performing the following steps:

Step 1: $\mathbf{v}^{m-1/2} = \mathbf{v}^{m-1} + \widetilde{\mathbf{y}}^1 + \widetilde{\mathbf{y}}^2 + \cdots + \widetilde{\mathbf{y}}^n$. Here $\widetilde{\mathbf{y}}^i$, $1 \leq i \leq n$, are corrections spanned by the unit co-ordinate vectors \mathbf{e}_i , produced by the Projected Gauss–Seidel algorithm with initial approximation \mathbf{v}^{m-1} .

Step 2: $\mathbf{v}^m = \mathbf{v}^{m-1/2} + P\mathbf{y}_C$. Here $\mathbf{y}_C \in V_C$ —the coarse-grid vector space, solves the quadratic minimization problem,

$$J_C(\mathbf{y}_C) = \frac{1}{2} \mathbf{y}_C^T A_C \mathbf{y}_C - \mathbf{b}_C^T \mathbf{y}_C \mapsto \min$$

subject to the inequality constraints

$$(\mathbf{y}_C)_i \leq g_i - (\mathbf{v}^{m-1/2})_i \quad \text{for all } i \in \Delta$$

The following main result holds.

Theorem 2.2

Algorithm 2.1 provides a monotone scheme, i.e. for any two consecutive iterates, \mathbf{v}^{m-1} and \mathbf{v}^m , produced by the algorithm we have

$$J(\mathbf{v}^m) \leq J(\mathbf{v}^{m-1})$$

Proof

Given the iterate \mathbf{v}^{m-1} and applying the Projected Gauss–Seidel algorithm in Step 1 we get a new intermediate iterate $\mathbf{v}^{m-1/2}$ for which we have:

- (1) The new intermediate iterate satisfies the inequality constraints due to the projection operation in the Projected Gauss–Seidel algorithm:

$$(\mathbf{v}^{m-1/2})_i \leq g_i \quad \text{for all } i \in \Delta \quad (4)$$

- (2) The value of the functional at the new intermediate iterate is less than one at the previous one because the Projected Gauss–Seidel algorithm provides a monotone scheme

$$J(\mathbf{v}^{m-1/2}) \leq J(\mathbf{v}^{m-1})$$

At the next step, we seek for a correction $\mathbf{y}_C \in V_C$ such that

$$(\mathbf{v}^{m-1/2} + P\mathbf{y}_C)_i \leq g_i \quad \text{for all } i \in \Delta$$

and

$$J(\mathbf{v}^{m-1/2} + P\mathbf{y}_C) \leq J(\mathbf{v}^{m-1/2})$$

We simplify the expression

$$\begin{aligned}
 & J(\mathbf{v}^{m-1/2} + P\mathbf{y}_C) \\
 &= \frac{1}{2}(\mathbf{v}^{m-1/2})^T A \mathbf{v}^{m-1/2} + \frac{1}{2} \mathbf{y}_C^T P^T A P \mathbf{y}_C + \mathbf{y}_C^T P^T A \mathbf{v}^{m-1/2} - \mathbf{b}^T (\mathbf{v}^{m-1/2} + P\mathbf{y}_C) \\
 &= J(\mathbf{v}^{m-1/2}) + J_C(\mathbf{y}_C)
 \end{aligned} \tag{5}$$

It is clear that it is equivalent to solve the coarse grid constraint minimization problem

$$J_C(\mathbf{y}_C) \equiv \frac{1}{2} \mathbf{y}_C^T A_C \mathbf{y}_C - \mathbf{b}_C^T \mathbf{y}_C \mapsto \min$$

subject to

$$(\mathbf{y}_C)_i \leq g_i - (\mathbf{v}^{m-1/2})_i \quad \text{for all } i \in \Delta$$

Note that we used the fact that the constraints are exactly present on the coarse level by our assumption on P , namely, that, $(P\mathbf{y}_C)|_\Delta = \mathbf{y}_C|_\Delta$. It is clear then that if we choose the correction $\mathbf{y}_C = \mathbf{y}_C^{\text{opt}}$ where $\mathbf{y}_C^{\text{opt}}$ is the solution of the above constraint minimization problem we have that

$$J(\mathbf{v}^{m-1/2} + P\mathbf{y}_C^{\text{opt}}) \leq J(\mathbf{v}^{m-1/2}) \tag{6}$$

The latter is true since one may choose $\mathbf{y}_C = 0$ and satisfy the constraints due to inequality (4). We then have

$$J_C(\mathbf{y}_C^{\text{opt}}) \leq J_C(0) = 0$$

Therefore, from (5) we have $J_C(\mathbf{y}_C^{\text{opt}}) = J(\mathbf{v}^{m-1/2} + P\mathbf{y}_C^{\text{opt}}) - J(\mathbf{v}^{m-1/2})$ which implies (6). Thus from (2.2) and (6) since $\mathbf{v}^m = \mathbf{v}^{m-1/2} + P\mathbf{y}_C^{\text{opt}}$ the proof is complete, i.e. one has,

$$J(\mathbf{v}^m) \leq J(\mathbf{v}^{m-1/2}) \leq J(\mathbf{v}^{m-1}) \quad \square$$

3. COARSENING AWAY FROM THE CONTACT BOUNDARY

We generate coarse spaces V_k , $k=0, \dots, \ell$, where $V_0 = V$ is the original fine-grid space by element agglomeration. Here we use the fact that the problem under consideration comes from a finite element discretization. Hence, one has access to elements and their topology (on the fine grid). Agglomeration algorithms were proposed in Reference [6]. They utilize certain element topological relations and create the same relations on coarse levels recursively. Details about specific implementation of the algorithms can be found in Reference [19]. Here we use a modification of the agglomeration algorithm in a way that dof associated with the constraint boundary C are not coarsened. The original agglomeration algorithm from Reference [6] allowed for barriers, that is, some faces of elements are labelled as unacceptable and the elements that share such faces are kept on coarse levels (without being agglomerated with other elements). A principal step of the thus modified algorithm is as follows: The faces of elements that are on the boundary C are labelled as unacceptable. Then one labels as unacceptable the faces of all elements that touch the initial set of unacceptable faces. Thus, at least one layer of elements near the contact boundary C is kept on the next coarse level (i.e. those elements have not been agglomerated with other elements). On the next coarsening

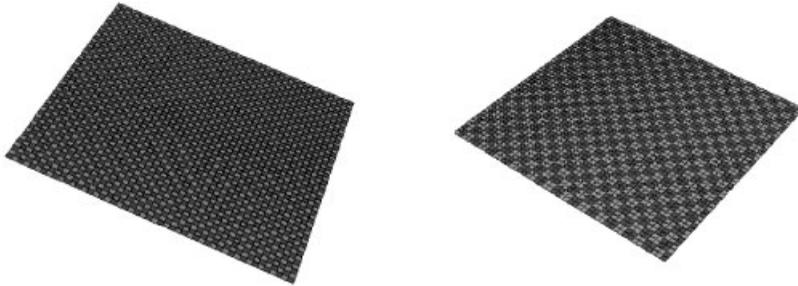


Figure 1. Levels 0 and 1 of the agglomeration.

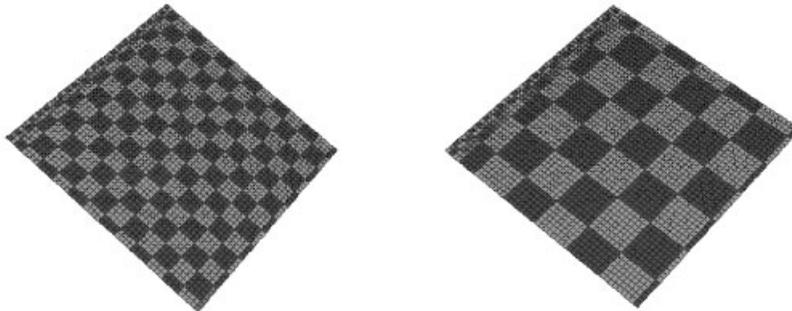


Figure 2. Levels 2 and 3 of the agglomeration.

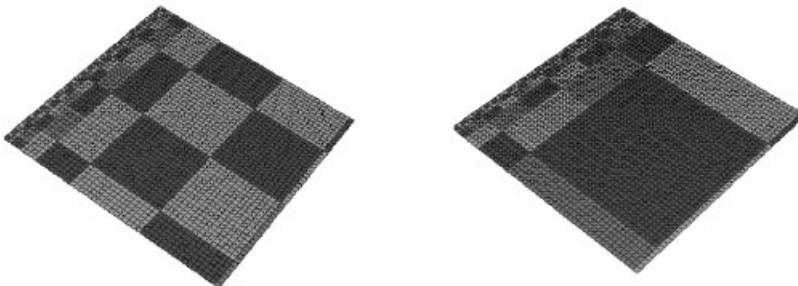


Figure 3. Levels 4 and 5 of the agglomeration.

levels, one increases the number of unacceptable faces by adding the faces one more layer of (current coarse) elements on the list of unacceptable faces. The resulting agglomerated elements are shown in Figures 1–3.

The second part of the AMGe coarsening is to select coarse dofs. In the present paper, we have selected the vertices of the agglomerated elements as coarse dofs. The interpolation matrix $P_k = I_{k+1}^k$ is defined by the AMGe principle in the form used in Reference [6]. It

Table I. Agglomeration information.

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
dofs	5000	1500	614	398	332	326
Elements	2401	674	242	135	106	103
dofs on Γ_C	100	100	100	100	100	100

requires element matrices on a given discretization level and creates element matrices on the next coarse level. Then the coarse operator is defined variationally, that is, $A_{k+1} = P_k^T A_k P_k$.

The main reason for the kind of coarsening we chose in the present paper, is that if a fine-grid vector $\mathbf{v} \in V_k$ satisfies the constraints, its (point-wise) restriction $\mathbf{v}|_{\text{coarse nodes}}$ also satisfies the constraints, and more importantly, if a coarse vector satisfies the coarse level constraints its AMGe interpolant satisfies the fine-grid constraints (trivially) since the vector does not change on C . This is the case, since the elements near the contact boundary have not been changed. Computationally, this strategy is acceptable since the constraint set C is a boundary of one-dimension less than the whole index set (see Table I).

4. MULTILEVEL SUBSPACE MINIMIZATION ALGORITHM

Let $V = V_0$ be the fine-grid vector space and V_k , $k \geq 0$ be a sequence of coarse vector spaces. The transfer operators are denoted by $P_k = I_{k+1}^k : V_{k+1} \mapsto V_k$. Their transposes I_k^{k+1} are used as restriction operators. Finally, let $A_{k+1} = P_k^T A_k P_k$ be the coarse matrix obtained variationally from the next level fine-grid matrix.

Algorithm 4.1 (*Multilevel subspace minimization algorithm*). Consider problem (1) with $\mathbf{b}_0 = \mathbf{b}$ and $\mathbf{g}^0 = \mathbf{g}$ given. Let ℓ be the coarsest level.

0. For $k \geq 0$ let $\mathbf{v}_k \in V_k$, $(\mathbf{v}_k)_i \leq (g^k)_i$, $i \in \Delta$, be a current iterate at level k and let $J_k(\mathbf{y}) = \frac{1}{2} \mathbf{y}^T A_k \mathbf{y} - \mathbf{b}_k^T \mathbf{y}$ be the k th level quadratic functional.
1. If $k < \ell$ apply $v_1 \geq 1$ Projected Gauss–Seidel iterations with initial iterate \mathbf{v}_k . The resulting iterate we also denote by \mathbf{v}_k . Go to Step 3.
2. Else (i.e. if $k = \ell$), then solve the corresponding constrained minimization problem exactly. The resulting solution denote by \mathbf{v}_k . Set $k := k - 1$ and go to Step 4.
3. Seek for a coarse-grid correction \mathbf{v}_{k+1} .
 - Set $\mathbf{g}^{k+1} = \mathbf{g}^k - \mathbf{v}^k|_{\Delta}$ and choose as initial approximation at level $k + 1$ $\mathbf{v}_{k+1} = 0$.
 - Set $\mathbf{b}_{k+1} = I_k^{k+1}(\mathbf{b}_k - A_k \mathbf{v}_k)$.
 Set $k := k + 1$ and go to Step 1.
4. Update level k iterate \mathbf{v}_k

$$\mathbf{v}_k^{\text{new}} = \mathbf{v}_k + I_{k+1}^k \mathbf{v}_{k+1}$$

5. Apply $v_2 \geq 1$ Projected Gauss–Seidel iterations with initial iterate $\mathbf{v}_k^{\text{new}}$. The resulting iterate we denote by \mathbf{v}_k . Set $k := k - 1$. If $k \geq 0$, go to Step 4, else one V -cycle is completed.

We first show that the above algorithm is well-defined, namely, that all the intermediate iterates in Algorithm 4.1 satisfy the appropriate constraints.

The resulting iterate \mathbf{v}_k after the application of the Projected Gauss–Seidel algorithm in Step 1 of Algorithm 4.1 satisfies the following constraints $(\mathbf{v}_k)_i \leq (\mathbf{g}^k)_i$, $i \in \Delta$. This is true due to the projection procedure in the Projected Gauss–Seidel algorithm.

In Step 2, the resulting iterate satisfies the same constraints since we use exact coarse grid solve.

In Step 3, the constraint set changes and we have that the initial approximation at level $k + 1$, $\mathbf{v}_{k+1} = 0$ satisfies the new constraints $(\mathbf{v}_{k+1})_i \leq (\mathbf{g}^{k+1})_i$, $i \in \Delta$. This is true because $(\mathbf{v}_k)_i \leq (\mathbf{g}^k)_i$, $i \in \Delta$.

In Step 4, one should note that $(\mathbf{v}_k^{\text{new}})_i \leq (\mathbf{g}^k)_i$, $i \in \Delta$. Indeed, due to the main property of the interpolation matrix $P \equiv I_{k+1}^k$, (2),

$$(\mathbf{v}_k^{\text{new}})|_{\Delta} = (\mathbf{v}_k)|_{\Delta} + (I_{k+1}^k \mathbf{v}_{k+1})|_{\Delta} = (\mathbf{v}_k)|_{\Delta} + \mathbf{v}_{k+1}|_{\Delta} \leq (\mathbf{v}_k)|_{\Delta} + \mathbf{g}^{k+1} = \mathbf{g}^k$$

In Step 5, the resulting iterate \mathbf{v}_k satisfies the appropriate constraints by construction due to the properties of the Projected Gauss–Seidel.

Corollary 4.2

Algorithm 4.1 provides a monotone scheme.

Proof

This fact follows directly from Theorem 2.2. □

We remark at the end that a proof of mesh-independent convergence rate of Algorithm 4.1 seems possible along the lines presented in Reference [20]. A corresponding proof in the present context will simplify substantially since our spaces are coarsened away from the constraint boundary. Details will be provided elsewhere.

5. FAS CONSTRAINED OPTIMIZATION ALGORITHM

Since one can treat problem (1) as a non-linear one, one could attempt to solve it by applying the classical FAS method [8]. A corresponding FAS algorithm in the present context takes the form.

Algorithm 5.1 (*FAS constrained optimization algorithm*). Consider problem (1) with $\mathbf{b}_0 = \mathbf{b}$ and $\mathbf{g}^0 = \mathbf{g}$ given. Let ℓ be the coarsest level. Finally, let \mathcal{N}_k be the set of nodes (dofs) at level k .

0. For $k \geq 0$ let $\mathbf{v}_k^0 \in V_k$, $(\mathbf{v}_k^0)_i \leq (\mathbf{g}^k)_i$, $i \in \Delta$, be a current iterate at level k and let $J_k(\mathbf{y}) = \frac{1}{2} \mathbf{y}^T A_k \mathbf{y} - \mathbf{b}_k^T \mathbf{y}$ be the k th level quadratic functional.
1. If $k < \ell$ apply $v_1 \geq 1$ Projected Gauss–Seidel iterations with initial iterate \mathbf{v}_k^0 . The resulting iterate we denote by \mathbf{v}_k . Go to Step 3.
2. Else (i.e. if $k = \ell$), then solve the corresponding constrained minimization problem exactly. The resulting solution denote by \mathbf{v}_k . Set $k := k - 1$ and go to Step 4.

3. Based on a coarse-grid constrained minimization problem correct the k th level iterate \mathbf{v}_k . We define \mathbf{g}_{k+1} and \mathbf{b}_{k+1} for the coarse-grid problem as follows:
 - Set $\mathbf{g}^{k+1} = \mathbf{g}^k = \dots = \mathbf{g}$ and choose as initial approximation at level $k+1$ $\mathbf{v}_{k+1}^0 = \mathbf{v}_k|_{\mathcal{N}_{k+1}}$.
 - Set $\mathbf{b}_{k+1} = I_{k+1}^{k+1}(\mathbf{b}_k - A_k \mathbf{v}_k) + A_{k+1} \mathbf{v}_{k+1}^0$.
Set $k := k+1$ and go to Step 1.
4. Update level k iterate \mathbf{v}_k

$$\mathbf{v}_k^{\text{new}} = \mathbf{v}_k + I_{k+1}^k(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0)$$

5. Apply $v_2 \geq 1$ Projected Gauss–Seidel iterations with initial iterate \mathbf{v}_k . The resulting iterate we also denote by \mathbf{v}_k . Set $k := k-1$. If $k \geq 0$, go to Step (4), else one V-cycle is completed.

We next show that the above algorithm is well-defined, that is, that all the intermediate iterates in Algorithm 5.1 satisfy the appropriate constraints.

The resulting iterate \mathbf{v}_k after the application of the Projected Gauss–Seidel algorithm in Step 1 of Algorithm 5.1 satisfies the following constraints $(\mathbf{v}_k)_i \leq (\mathbf{g}^k)_i$, $i \in \Delta$. This is true due to the projection procedure in the Projected Gauss–Seidel algorithm.

In Step 2, we have again that the resulting iterate satisfies the same constraints since we use exact coarse-grid solve.

In Step 3, the constraint set does not change (from level k to level $k+1$) thus, \mathbf{v}_{k+1}^0 satisfies the constraints since it is a restriction of \mathbf{v}_k and the latter satisfies the constraints.

In Step 4, one has to show that $(\mathbf{v}_k^{\text{new}})_i \leq (\mathbf{g}^k)_i$, $i \in \Delta$. Indeed, due to the main property of the interpolation matrix $P \equiv I_{k+1}^k$, (2),

$$\begin{aligned} (\mathbf{v}_k^{\text{new}})|_{\Delta} &= \mathbf{v}_k|_{\Delta} + (I_{k+1}^k(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0))|_{\Delta} \\ &= \mathbf{v}_k|_{\Delta} + \mathbf{v}_{k+1}|_{\Delta} - \mathbf{v}_{k+1}^0|_{\Delta} \\ &= \mathbf{v}_k|_{\Delta} + \mathbf{v}_{k+1}|_{\Delta} - \mathbf{v}_k|_{\Delta} \\ &= \mathbf{v}_{k+1}|_{\Delta} \\ &\leq \mathbf{g}^{k+1} = \mathbf{g}^k \end{aligned}$$

In Step 5, the resulting iterate \mathbf{v}_k satisfies the appropriate constraints again due to the properties of the Projected Gauss–Seidel.

The following, main fact easily follows from the construction of the FAS iterates.

Theorem 5.2

Algorithm 5.1 provides a monotone scheme.

Proof

It is sufficient to prove that

$$J_k(\mathbf{v}_k^{\text{new}}) \leq J_k(\mathbf{v}_k)$$

Denote for brevity $P = I_{k+1}^k$. Based on the definition of $A_{k+1} = P^T A_k P$, J_k and J_{k+1} , and $\mathbf{b}_{k+1} = P^T(\mathbf{b}_k - A_k \mathbf{v}_k) + A_{k+1} \mathbf{v}_{k+1}^0$, one can derive the identity,

$$\begin{aligned}
J_k(\mathbf{v}_k^{\text{new}}) &= J_k(\mathbf{v}_k + P(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0)) \\
&= \frac{1}{2}(\mathbf{v}_k + P(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0))^T A_k (\mathbf{v}_k + P(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0)) - (\mathbf{b}_k)^T (\mathbf{v}_k + P(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0)) \\
&= J_k(\mathbf{v}_k) + \frac{1}{2}(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0)^T A_{k+1} (\mathbf{v}_{k+1}^0 - \mathbf{v}_{k+1}) - (\mathbf{b}_k - A_k \mathbf{v}_k)^T P(\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^0) \\
&= J_k(\mathbf{v}_k) + \frac{1}{2}(\mathbf{v}_{k+1})^T A_{k+1} \mathbf{v}_{k+1} - (P^T(\mathbf{b}_k - A_k \mathbf{v}_k) + A_{k+1} \mathbf{v}_{k+1}^0)^T \mathbf{v}_{k+1} \\
&\quad + \frac{1}{2}(\mathbf{v}_{k+1}^0)^T A_{k+1} \mathbf{v}_{k+1}^0 + (\mathbf{b}_k - A_k \mathbf{v}_k)^T P \mathbf{v}_{k+1}^0 \\
&= J_k(\mathbf{v}_k) + J_{k+1}(\mathbf{v}_{k+1}) + \frac{1}{2}(\mathbf{v}_{k+1}^0)^T A_{k+1} \mathbf{v}_{k+1}^0 + (\mathbf{b}_k - A_k \mathbf{v}_k)^T P \mathbf{v}_{k+1}^0
\end{aligned}$$

One also has,

$$\begin{aligned}
-J_{k+1}(\mathbf{v}_{k+1}^0) &= -\frac{1}{2}(\mathbf{v}_{k+1}^0)^T A_{k+1} \mathbf{v}_{k+1}^0 + (\mathbf{b}_{k+1})^T \mathbf{v}_{k+1}^0 \\
&= -\frac{1}{2}(\mathbf{v}_{k+1}^0)^T A_{k+1} \mathbf{v}_{k+1}^0 + (P^T(\mathbf{b}_k - A_k \mathbf{v}_k) + A_{k+1} \mathbf{v}_{k+1}^0)^T \mathbf{v}_{k+1}^0 \\
&= \frac{1}{2}(\mathbf{v}_{k+1}^0)^T A_{k+1} \mathbf{v}_{k+1}^0 + (\mathbf{b}_k - A_k \mathbf{v}_k)^T P \mathbf{v}_{k+1}^0
\end{aligned}$$

The latter two identities imply the following main one:

$$J_k(\mathbf{v}_k^{\text{new}}) = J_k(\mathbf{v}_k) + J_{k+1}(\mathbf{v}_{k+1}) - J_{k+1}(\mathbf{v}_{k+1}^0)$$

Now, having in mind that in Algorithm 5.1 the vector \mathbf{v}_{k+1} reduces the functional J_{k+1} (assumed by induction, true at the coarsest level, and since we use monotone smoother), that is,

$$J_{k+1}(\mathbf{v}_{k+1}) \leq J_{k+1}(\mathbf{v}_{k+1}^0)$$

we arrive at the final desired inequality,

$$J_k(\mathbf{v}_k^{\text{new}}) \leq J_k(\mathbf{v}_k) \quad \square$$

6. NUMERICAL EXPERIMENTS FOR SIGNORINI'S PROBLEM

6.1. Signorini's problem

We consider the so-called Signorini's problem, which models a linearly elastic body, deformed due to volume and surface forces, which should not penetrate a rigid frictionless foundation.

Find the displacement field $\mathbf{u}=\mathbf{u}(\mathbf{x})$ (a vector valued function) such that

$$\begin{aligned} -\sigma_{ij,j}(\mathbf{u}) &= f_i && \text{in } \Omega \\ u_i &= 0 && \text{on } \Gamma_D \\ \sigma_{ij}(\mathbf{u})n_j &= t_i && \text{on } \Gamma_F \\ \sigma_{T_i}(\mathbf{u}) &= 0 && \text{on } \Gamma_C \\ \sigma_n(\mathbf{u}) &\leq 0 && \text{on } \Gamma_C \\ u_n - g &\leq 0 && \text{on } \Gamma_C \\ \sigma_n(\mathbf{u})(u_n - g) &= 0 && \text{on } \Gamma_C \end{aligned}$$

$\Omega \subset \mathbb{R}^d$, $d=2,3$, $i,j,k, l=1,\dots,d$.

\mathbf{n} outward unit vector normal to Γ_C ,

$$u_n = \mathbf{u} \cdot \mathbf{n} = u_i n_i$$

g is the initial gap between the body and the foundation, $g(\mathbf{x}) \geq 0$; \mathbf{f} the body forces; \mathbf{t} the surface tractions applied to a portion of the body surface Γ_F ; Γ_D a portion of the boundary along which the body is fixed; Γ_C the candidate contact surface (the actual surface on which the body comes in contact is not known in advance but is contained in the candidate contact surface); \mathbf{E} the Hooke's tensor having the following symmetry properties $E_{ijkl} = E_{jikl} = E_{ijlk} = E_{klij}$; σ the stress tensor $\sigma_{ij}(\mathbf{u}) = E_{ijkl} u_{k,l}$ —Hooke's law; σ_n the normal component of the stress vector, $\sigma_n = \sigma_{ij} n_i n_j$ and σ_{T_i} the tangential component of the stress vector, $\sigma_{T_i} = \sigma_{ij} n_j - \sigma_n n_i$.

We are using the Einstein summation notation above i.e. repeated indices are implicitly summed over. Also we use the notation, $_j = \partial/\partial x_j$ for a derivative.

The finite element discretization of the weak form of the Signorini's problem can be equivalently formulated as the following constrained minimization problem:

$$\begin{aligned} J(\mathbf{v}) &= \frac{1}{2} \mathbf{v}^T A \mathbf{v} - \mathbf{b}^T \mathbf{v} \mapsto \min, \quad \mathbf{v} \in K \\ K &= \{ \mathbf{v} \in \mathbb{R}^{dN_P} \mid \mathbf{n}_P \cdot \mathbf{v}_P \leq g_P, P \in \Gamma_C \} \end{aligned} \tag{7}$$

Note that there is only one inequality imposed per node P on Γ_C and that each node P is associated with $d > 1$ dof. In general, the set K is not of the canonical form $v_i \leq g_i$ but it can be transformed to one (see, e.g. Reference [11] or below).

\mathcal{T}_h consists of simplex elements, over which each component of the displacement is approximated by linear polynomials. N_P is the total number of nodal points in \mathcal{T}_h . Again, $d=2$ or $d=3$ is dimension of the domain; A the symmetric positive definite matrix; \mathbf{n} (or \mathbf{n}_P) the unit length outward normal on Γ_C (associated with node $P \in \Gamma_C$); g_i the length of a vector beginning at the node corresponding to dofs i with direction parallel to the normal vector at that node, ending at the crossing point with the rigid foundation and Δ the set of indices of all constraint degrees of freedom on Γ_C .

We will assume that all normal vectors \mathbf{n}_P , associated with nodes $P \in \Gamma_C$ are equal to a co-ordinate unit vector. This can be achieved by appropriate transformation (change of variables) at the nodes on Γ_C of the unknown displacement vector. Then problem (7) takes the standard form with simple inequality constraints as in (1).

Table II. Numerical results for both FAS and MSMA on a sequence of six grids. The performance of the two methods is practically identical; they differ only by the number of iterations on coarsest grid (not shown in the table).

	Grid1	Grid2	Grid3	Grid4	Grid5	Grid6
No. dofs	200	800	1800	5000	20000	45000
No. Grid levels	5	6	6	7	8	9
Grid complexity	2.57	2.11	1.84	1.69	1.54	1.49
Operator complexity	2.47	2.03	1.78	1.66	1.52	1.47
No. MG iter 5 pre- and post-PGS sweeps	4	4	4	5	7	9
No. MG iter 1 pre- and post-PGS sweeps	6	5	6	7	12	22

The algorithm given above has been implemented in a C++ code and have been applied to a Signorini's problem in 2D with $\Omega = (0, 4)^2$, $\Gamma_D = \{1\} \times [0, 4]$, $\Gamma_F = \{4\} \times [0, 4] \cup [0, 4] \times \{4\}$, $\Gamma_C = [0, 4] \times \{1\}$, with body force $\mathbf{f} = (0, -1)$, surface traction $\mathbf{t} = 0$, initial gap $g = 2$, Lamé constants $\lambda = 11.3$, $\mu = 8.1$. The continuous problem is discretized by bilinear finite elements on quadrilaterals. In each iteration of the Multilevel Subspace Minimization Algorithm (MSMA) and the FAS Constrained Optimization Algorithm (FAS) we apply either 1 or 5 pre- and post-smoothing sweeps of Projected Gauss–Seidel and the Dostal's algorithm as a coarse grid solver on the coarsest level. The latter algorithm is found in Reference [12]. It falls in the category of active set strategies. Depending on a parameter, one alternates between searching for the contact set and solving a complementary unconstrained minimization problem with Conjugate gradient like method.

The algorithms MSMA and FAS terminate the iterations when $|\mathbf{J}(\mathbf{v}_{m+1}) - \mathbf{J}(\mathbf{v}_m)|/|\mathbf{J}(\mathbf{v}_0)| \leq 10^{-5}$. The stopping criterion for the Dostal's algorithm is $|\mathbf{J}(\mathbf{v}_{m+1}) - \mathbf{J}(\mathbf{v}_m)|/|\mathbf{J}(\mathbf{v}_0)| \leq 10^{-10}$. In the latter case, J and \mathbf{v}_i stand for the functional and the iterates at the coarsest level.

The numerical results are shown in Table II.

Both algorithms give very similar results—the number of multigrid iterations for both algorithms are the same, the only difference is in the number of Conjugate gradient steps in the Dostal's algorithm.

One can clearly see the almost mesh-independent number of the iterations especially if the number of smoothing steps is sufficiently large. The deterioration of the overall iterations for small numbers of smoothing steps is most likely due to the fact that we use the simplest choice of coarse dofs, namely the vertices of the agglomerated elements. An alternative would be to use richer coarse grids (obtained by compatible relaxation [21] or spectral AMGe [22]), but this is left for further study. Another fact that possibly contributes to the need of high number of projected Gauss–Seidel iterations is that the smoothing property of Gauss–Seidel for the constrained problem is probably not as good as in the case of standard Gauss–Seidel applied to the unconstrained functional.

We finally mention that the cost of the coarse-grid algorithm is typically of order $\mathcal{O}(|\Delta| \times (\text{number of iterations}))$, that is, proportional to the number of dofs on the contact boundary times the number of iterations used in the Dostal's algorithm. The latter can be bounded by the condition number of the respective matrix involved. In our model case it is of order h^{-1} . That is overall, the cost is bounded by the total number of dofs (on the fine mesh).

In Figure 4 is shown the difference $\log(J(\mathbf{v}_k) - J(\mathbf{u}))$. Here \mathbf{v}_k is the k th iterate and \mathbf{u} is the vector at which the functional reaches its minimal value, with respect to the number of

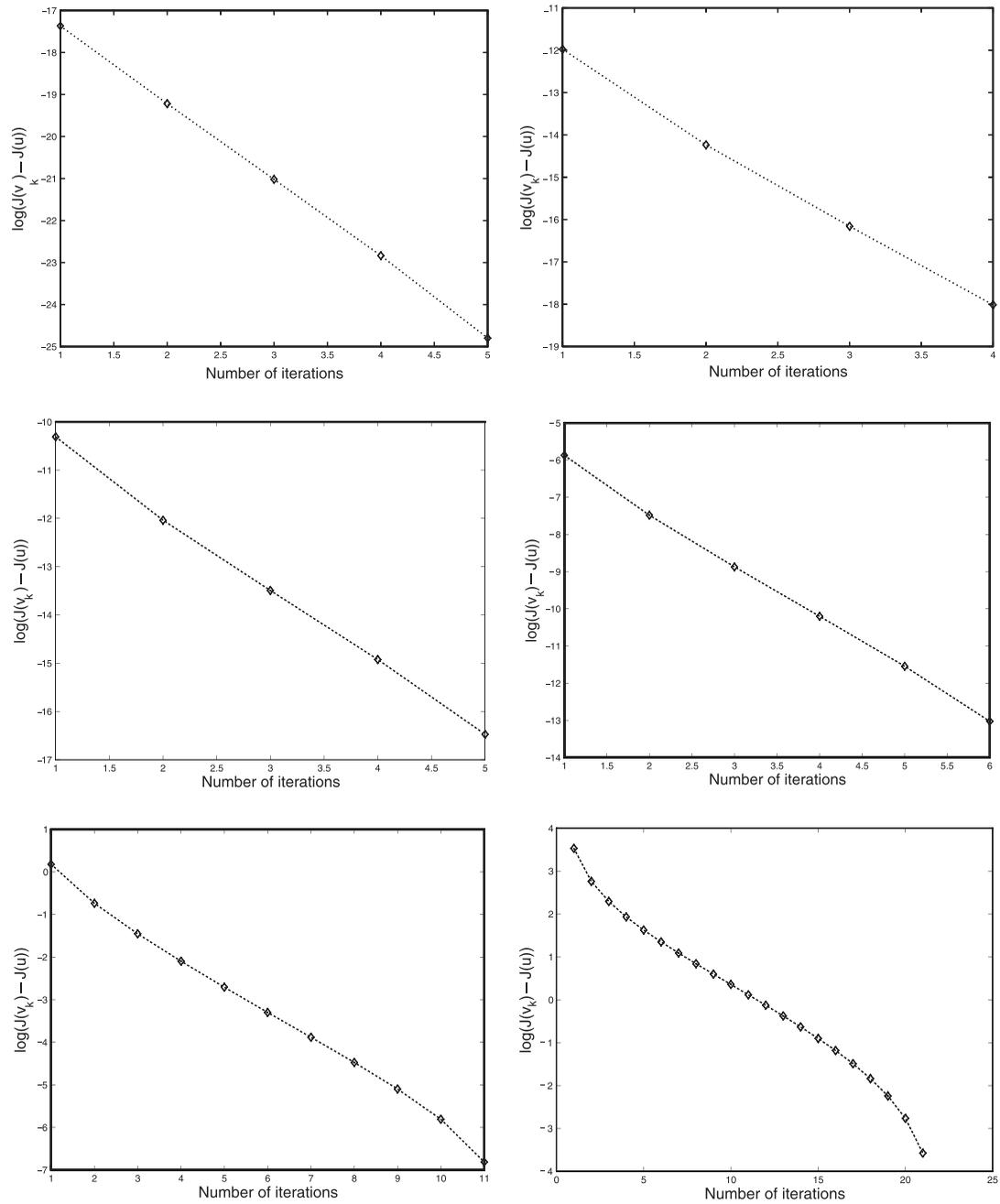


Figure 4. Numerical results for grids 1–6.

iterations on six different grids. The observation is that this quantity decreases almost linearly on all the grids.

7. CONCLUSIONS

Two multigrid algorithms for the solution of the constrained minimization problem arising from the Signorini's problem are presented—Multilevel Subspace Minimization Algorithm and a classical FAS Constrained Optimization Algorithm. Both algorithms provide monotone convergence due to the special element agglomeration AMGe coarsening away from the constrained set we employed. Thus the constraint set is present on all levels and transfer of data between the grids is straightforward.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract; contract/grant number: W-7405-Eng-48.

REFERENCES

1. Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparcity and its Applications*, Evans DJ (ed.). Cambridge University Press: Cambridge, 1984; 257–284.
2. Ruge J, Stüben K. Algebraic Multigrid (AMG). *Multigrid Methods, Frontiers in Applied Mathematics*, McCormick S (ed.). SIAM: Philadelphia, 1987; 73–130.
3. Trottenberg U, Oosterlee C, Schüller A. *Multigrid*. Academic Press: San Diego, 2001.
4. Brandt A, Cryer C. Multi-grid algorithms for the solution of linear complementarity problems arising from free boundary problems. *SIAM Journal on Scientific and Statistical Computing* 1983; **4**:655–684.
5. Oosterlee C. On multigrid for linear complementarity problems with application to American-style options. *Electronic Transactions on Numerical Analysis* 2003; **15**:165–185.
6. Jones J, Vassilevski P. AMGe based on element agglomeration. *SIAM Journal on Scientific Computing* 2001; **23**:109–133.
7. Brezina M, Cleary AJ, Falgout RD, Henson VE, Jones JE, Manteuffel TA, McCormick SF, Ruge JW. Algebraic multigrid based on element interpolation (AMGe). *SIAM Journal on Scientific Computing* 2000; **22**: 1570–1592.
8. Brandt A. Multilevel-adaptive solutions to boundary value problems. *Mathematics of Computation* 1977; **31**: 333–390.
9. Kornhuber R. Monotone multigrid methods for elliptic variational inequalities I. *Numerische Mathematik* 1994; **69**:167–184.
10. Kornhuber R. Monotone multigrid methods for elliptic variational inequalities II. *Numerische Mathematik* 1996; **72**:481–499.
11. Kornhuber R, Krause R. Adaptive multigrid methods for Signorini's problem in linear elasticity. *Computing and Visualization in Science* 2001; **4**:9–20.
12. Dostal Z. Box constrained quadratic programming with proportioning and projections. *SIAM Journal on Optimization* 1997; **7**:871–887.
13. Hackbusch W, Mittelman H. On multigrid methods for variational inequalities. *Numerische Mathematik* 1983; **42**:65–76.
14. Hoppe R. Multigrid algorithms for variational inequalities. *SIAM Journal on Numerical Analysis* 1987; **24**: 1046–1065.
15. Carstensen C, Scherf O, Wriggers P. Adaptive finite elements for elastic bodies in contact. *SIAM Journal on Scientific Computing* 1999; **20**:1605–1626.
16. Glowinski R. *Numerical Methods for Nonlinear Variational Problems*. Springer: New York, 1984.
17. Kikuchi N, Oden J. *Contact Problems in Elasticity*. SIAM: Philadelphia, 1988.
18. Schöberl J. Solving the Signorini problem on the basis of domain decomposition techniques. *Computing* 1998; **60**:323–344.

19. Vassilevski P. Sparse matrix element topology with application to AMG and preconditioning. *Numerical Linear Algebra with Applications* 2002; **9**:429–444.
20. Tai X. Rate of convergence of some constraint decomposition methods for nonlinear variational inequalities. *Numerische Mathematik* 2003; **93**:755–786.
21. Brandt A. General highly accurate algebraic coarsening. *Electronic Transactions on Numerical Analysis* 2000; **10**:1–20.
22. Chartier T, Falgout R, Henson V, Jones J, Manteuffel T, McCormick S, Ruge J, Vassilevski PS. Spectral AMG. *SIAM Journal on Scientific Computing* 2003; **25**:1–26.