

# NONLINEAR AMGE WITH COARSENING AWAY FROM THE CONTACT BOUNDARY FOR THE SIGNORINI'S PROBLEM

ANA H. IONTCHEVA AND PANAYOT S. VASSILEVSKI

ABSTRACT. The finite element discretization of the Signorini's problem leads to inequality constrained minimization problem. In this paper we present a nonlinear element based algebraic multigrid method with a special coarsening away from the contact boundary for the solution of this problem. As a smoothing procedure we use the Projected Gauss–Seidel algorithm and for the coarse grid solver—a modification of the Dostal's algorithm ([2]). The performance of the resulting method is illustrated by numerical experiments.

## 1. INTRODUCTION

This paper deals with a standard FAS–type multigrid algorithm for solving a constrained minimization problem that arises in the finite element discretization of a contact problem. The resulting fine–grid problem can be large and conventional algorithms do not usually converge fast. In the literature one can find a variety of methods developed for the solution of the Signorini's problem. Dual methods [3], [7] are based on reformulation of the constrained minimization problem into a saddle point problem using Lagrange multipliers. Penalty methods [1], [3], [7] define a functional, which depends on a penalty parameter and an unconstrained minimization problem is solved which is an approximation to the problem of interest. Projection methods suffer from slow convergence rates, so an efficient preconditioning is needed. Typically they lead to an inner–outer schemes in order to (approximately) evaluate the projection [9]. Active set strategies [2], [4], [5] alternate between approximating the contact set and solving a linear problem with fixed contact set. Monotone multigrid methods [8] successively minimize the functional of total energy in direction of suitably chosen subspaces. Our approach can be viewed as a monotone multigrid scheme using algebraically constructed coarse spaces. More specifically, in the present paper, we present a full approximation scheme (FAS) with interpolation operators constructed algebraically following the algorithm in [6]. The difference here is that a special coarsening away from the initial guess of the contact boundary is adopted. The main advantage of this coarsening is that on all levels the nodes on the contact boundary remain unchanged, thus avoiding the problems of violating the inpenetrability condition by applying the interpolation operators. Another advantage is that by construction the mesh along the contact boundary on all levels is fine (and

---

*Date:* Jan 02, 2003–beginning, today is: 13th January 2003.

*1991 Mathematics Subject Classification.* 74S05, 74B05, 65N22, 65N55.

*Key words and phrases.* contact problem, algebraic multigrid, FAS, algebraic coarsening.

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

coarsened gradually away from it) which increases the accuracy of the solution in the region producing the largest error.

The remainder of the paper is organized as follows. First we review the two algorithms that will be used in the FAS algorithm as a smoother and as a solver of the coarsest level. The smoothing procedure is the Projected Gauss-Seidel and the coarsest grid solver is the Dostal's algorithm. A monotone subspace minimization algorithm is presented in the next section. The specific element agglomeration coarsening procedure is discussed in Section 4. The FAS algorithm is summarized in Section 5. Finally, numerical experiments for a model 2-d Signorini's problem are presented.

## 2. CONSTRAINED MINIMIZATION PROBLEMS

In the present paper we consider the following model constrained minimization problem

$$(2.1) \quad J(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T A \mathbf{v} - \mathbf{b}^T \mathbf{v} \mapsto \min,$$

subject to the inequality constraints

$$n_i v_i \leq g_i \quad \text{for all } i \in \Sigma_C.$$

Here  $\Sigma$  and  $\Sigma_C$ ,  $\Sigma_C \subset \Sigma$  are given sets of indices and  $\mathbf{n}$  and  $\mathbf{g}$  are given vectors defined for indices from  $\Sigma$ . Typically  $n_i = 1$ , or  $n_i = -1$ .

We describe two algorithms for generating vectors  $\mathbf{v}$  minimizing the functional  $J(\mathbf{v})$ .

**2.1. Projected Gauss-Seidel.** Consider the functional  $J(\mathbf{v})$ . For every single component  $v_i$  it is simply a scalar quadratic function  $J(\mathbf{v}) = J(v_i)$ , if we fix all the values of the unknown vector at all degrees of freedom except the  $i$ th. If  $i \in \Sigma_C$  then we have the constraint  $n_i v_i \leq g_i$ . Assume for example that  $n_i = 1$ . Thus we obtain a problem of finding the minimum of a second order polynomial subject to a simple inequality constraint, that is, with  $x = v_i$ , and  $a > 0$ , we have to solve

$$\begin{aligned} ax^2 + bx + c &\mapsto \min \\ \text{subject to } x &\leq d. \end{aligned}$$

The solution is  $x = -\frac{b}{2a}$  if  $-\frac{b}{2a} \leq d$ , or  $x = d$  otherwise.

This method used iteratively is referred to as the Projected Gauss-Seidel. One can also develop block-versions of this method (see [3]).

**2.2. Dostal's algorithm.** Dostal's algorithm is presented in [2]. It falls in the category of active set strategies. Depending on a parameter  $\Gamma > 0$ , one alternates between searching for the contact set and solving the unconstrained minimization problem. It was developed for scalar minimization problems with box constraints. Let  $\mathbf{K}$  be the set of admissible vectors, satisfying the constraints  $\{n_i v_i \leq g_i, \quad \text{for all } i \in \Sigma_C\}$ . We will use the notation  $v_i$ , for the  $i$ th component of  $\mathbf{v}$ .

Define  $\mathbf{r}(\mathbf{v}) = \mathbf{b} - \mathbf{A}\mathbf{v}$ ,  $\mathcal{A}(\mathbf{v}) \subset \Sigma_C$ , the set of all indices for which the constraints are satisfied as equalities, called active index set. The free index set is then defined as  $\mathcal{F}(\mathbf{v}) = \Sigma \setminus \mathcal{A}(\mathbf{v})$ .

Define also for  $\mathbf{v} \in \mathbf{K}$ :

$$\varphi_i(\mathbf{v}) = \begin{cases} r_i(\mathbf{v}) & , i \in \mathcal{F}(\mathbf{v}) \\ 0 & , i \in \mathcal{A}(\mathbf{v}) \end{cases}$$

$$\beta_i(\mathbf{v}) = \begin{cases} 0 & , i \in \mathcal{F}(\mathbf{v}) \\ \min(r_i(\mathbf{v}), 0) & , i \in \mathcal{A}(\mathbf{v}) \end{cases}$$

$\nu(\mathbf{v}) = \varphi(\mathbf{v}) + \beta(\mathbf{v})$  - projected gradient

$\varphi(\mathbf{v})$  - error in  $\mathcal{F}(\mathbf{v})$

$\beta(\mathbf{v})$  - error in  $\mathcal{A}(\mathbf{v})$

To reduce  $\varphi(\mathbf{v})$  - one solves  $\mathbf{A}\mathbf{v} = \mathbf{b}$  restricted to  $\mathcal{F}(\mathbf{v})$

To reduce  $\beta(\mathbf{v})$  - one removes indices from  $\mathcal{A}(\mathbf{v})$ .

For the decomposition  $R = \mathcal{F}(\mathbf{v})$ ,  $S = \mathcal{A}(\mathbf{v})$  of the set of indices  $\Sigma$ , we can partition and rearrange any matrix  $\mathbf{D}$  and any vector  $\mathbf{d}$  in the following way

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{RR} & \mathbf{D}_{RS} \\ \mathbf{D}_{SR} & \mathbf{D}_{SS} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_R \\ \mathbf{d}_S \end{bmatrix}$$

The projection operator  $P_{\mathbf{K}}$  on the set of admissible vectors  $\mathbf{K}$  is defined as:

$$P_{\mathbf{K}}\mathbf{y} = \begin{cases} \mathbf{y} & \text{if } \mathbf{y} \in \mathbf{K} \\ \mathbf{q} & \text{if } \mathbf{y} \notin \mathbf{K} \end{cases}, \quad \text{and } \mathbf{q} = (q_i), \text{ where } q_i = \begin{cases} y_i & \text{if } i \notin \Sigma_C \\ g_i/n_i & \text{if } i \in \Sigma_C \end{cases}$$

**Proposition.**  $\mathbf{v}$  - solution  $\Leftrightarrow \mathbf{v} \in \mathbf{K}$ ,  $\nu(\mathbf{v}) = 0$ , i.e.,  $\varphi(\mathbf{v}) = \beta(\mathbf{v}) = 0$ .

The detailed algorithm is presented next.

ALGORITHM (PROPORTIONING WITH CONJUGATE GRADIENTS):

The algorithm generates sequence of iterates  $\mathbf{v}^m$ ,  $m \geq 0$ .

Let  $m = 0$ , and  $\mathbf{v}^0 \in \mathbf{K}$ ,  $\delta > 0$ ,  $\Gamma > 0$  be given.

while  $\|\nu(\mathbf{v}^m)\| > 0$

if  $\|\beta(\mathbf{v}^m)\|_{\infty} > \Gamma \|\varphi(\mathbf{v}^m)\|$  (Disproportional  $\mathbf{v}^m$ . Proportioning.)

$\mathbf{v}^{m+1} = \mathbf{v}^m + \alpha_m \beta(\mathbf{v}^m)$ ,  $\alpha_m = \min\left\{\frac{r^t \beta(\mathbf{v}^m)}{\beta^t(\mathbf{v}^m) \mathbf{A} \beta(\mathbf{v}^m)}, \frac{\delta}{\|\beta(\mathbf{v}^m)\|_{\infty}}\right\}$

$\mathbf{r} = \mathbf{r} + \alpha_m \mathbf{A} \beta(\mathbf{v}^m)$

$m = m + 1$

end if

if  $\|\beta(\mathbf{v}^m)\|_\infty \leq \Gamma \|\varphi(\mathbf{v}^m)\|$  (Proportional  $\mathbf{v}^m$ .)

*{Initialization of the conjugate gradient loop.}*

- (1)  $\mathbf{y} = \mathbf{v}^m$ ,  $R = \mathcal{F}(\mathbf{v}^m)$  – index set,  $\mathbf{p} = \mathbf{r}_R$ ,  $\alpha = 0$ ,  $\alpha_{CG} = 0$   
while  $\|\nu(\mathbf{y})\| > \varepsilon$  and  $\alpha = \alpha_{CG}$  and  $\|\beta(\mathbf{v}^m)\|_\infty \leq \Gamma \|\varphi(\mathbf{v}^m)\|$

*{Set steplength.}*

- (2)  $\alpha_{CG} = \frac{\mathbf{r}_R^t \mathbf{p}_R}{\mathbf{p}_R^t \mathbf{A}_{RR} \mathbf{p}_R}$   
if  $\mathbf{y} + \alpha_{CG} \mathbf{p} \in \mathbf{K}$   
(a)  $\alpha = \alpha_{CG}$   
else  
(b) Choose  $\alpha$  so that  $\mathbf{J}(P_{\mathbf{K}}(\mathbf{y} + \alpha \mathbf{p})) \leq \mathbf{J}(\mathbf{v}^m)$  and  $\mathcal{A}(\mathbf{v}^m) \subseteq \mathcal{A}(P_{\mathbf{K}}(\mathbf{y} + \alpha \mathbf{p}))$   
end if

*{Conjugate gradient update}*

- (3)  $\mathbf{y} = \mathbf{y} + \alpha \mathbf{p}$   
 $\mathbf{r} = \mathbf{r} + \alpha \mathbf{A} \mathbf{p}$   
 $\beta = \frac{\mathbf{r}_R^t \mathbf{A}_{RR} \mathbf{p}_R}{\mathbf{p}_R^t \mathbf{A}_{RR} \mathbf{p}_R}$   
 $\mathbf{p}_R = \mathbf{r}_R - \beta \mathbf{p}_R$

end while

- (4)  $\mathbf{v}^{m+1} = P_{\mathbf{K}}(\mathbf{y})$   
 $\mathbf{r} = \mathbf{b} - \mathbf{A} \mathbf{v}^{m+1}$   
 $m = m + 1$

end if

There are several strategies to choose  $\alpha$  in Step 2 b.

- (1) Feasible strategy:

$$\alpha = \{\max\{\mu : \mathbf{y} + \mu \mathbf{p} \in \mathbf{K}\} \text{ if } \mathbf{y} \in \mathbf{K} \text{ else } 0\} = f_s(\mathbf{y}, \mathbf{p})$$

- (2) Monotone strategy:

$$\alpha = \begin{cases} \alpha_{CG} & \text{if } \mathbf{J}(P_{\mathbf{K}}(\mathbf{y} + \alpha_{CG} \mathbf{p})) \leq \mathbf{J}(P_{\mathbf{K}}(\mathbf{y})) \\ f_s(\mathbf{y}, \mathbf{p}) & \text{else} \end{cases}$$

- (3) As long as possible strategy:

$$\alpha = \begin{cases} \alpha_{CG} & \text{if } \mathbf{J}(P_{\mathbf{K}}(\mathbf{y} + \alpha_{CG} \mathbf{p})) \leq \mathbf{J}(\mathbf{v}^m) \\ f_s(\mathbf{y}, \mathbf{p}) & \text{else} \end{cases}$$

## 3. A SUBSPACE MINIMIZATION METHOD

In this section we outline our monotone scheme in general terms.

We consider the following constrained minimization problem:

$$J(\mathbf{v}) = \frac{1}{2} (A\mathbf{v}, \mathbf{v}) - (\mathbf{b}, \mathbf{v}) \mapsto \min,$$

subject to the linear inequalities

$$n_i v_i \leq g_i, \quad \text{for all } i \in \Sigma_C.$$

Here  $\Sigma$  and  $\Sigma_C$ ,  $\Sigma_C \subset \Sigma$  are given sets of indices;  $\mathbf{n}$  and  $\mathbf{g}$  are given vectors defined for indices from  $\Sigma$ . Typically,  $n_i = 1$ , or  $n_i = -1$ .

We consider the following algorithm for solving the above constrained minimization problem.

Let  $\mathbf{V}_{\Sigma_*}$  be a subspace of  $\mathbf{V}$  which has all degrees of freedom associated with  $\Sigma_C$  and perhaps others. Let  $\mathbf{V}_0$  be another subspace of  $\mathbf{V}$  such that for any  $\mathbf{v}^0 \in \mathbf{V}_0$ ,  $\mathbf{v}_i = 0$  for  $i \in \Sigma_C$ . One has  $\mathbf{V}_{\Sigma_*} + \mathbf{V}_0 \subset \mathbf{V}$ . Note that the decomposition does not have to be direct and it, in general, may only be a proper subset of  $\mathbf{V}$ .

Let  $A_0$  be the restriction of  $A$  to the subspace  $\mathbf{V}_0$ . That is,  $A_0 = I_0^T A I_0$  for a rectangular matrix  $I_0$ . Finally let  $M_0^{-1}$  be an approximate inverse (preconditioner) to  $A_0$  such that

$$(3.1) \quad M_0 + M_0^T - A_0$$

be symmetric positive semi-definite.

**Algorithm 3.1** (Subspace minimization). *Given an iterate  $\mathbf{v}^{m-1}$ , compute the next iterate  $\mathbf{v}^m = \mathbf{v}^{m-1} + \mathbf{y}^0 + \mathbf{y}^{\Sigma_*}$ ,  $\mathbf{y}^0 = I_0(M_0)^{-1}I_0^T(\mathbf{r}^{m-1} - A\mathbf{y}^{\Sigma_*})$  such that  $\mathbf{y}^{\Sigma_*} \in \mathbf{V}_{\Sigma_*}$  solves the minimization problem*

$$J_{\Sigma_*}(\mathbf{y}^{\Sigma_*}) \equiv J(\mathbf{v}^{m-1} + I_0(M_0)^{-1}I_0^T\mathbf{r}^{m-1} + E_0\mathbf{y}^{\Sigma_*}) \mapsto \min.$$

Here,  $E_0$  is the subspace iteration matrix  $I - I_0(M_0)^{-1}I_0^T A$ . Note that the constraints do not involve  $\mathbf{y}^0$  (since it vanishes on  $\Sigma_*$ ). Therefore the constrained minimization problem only affects  $\mathbf{y}^{\Sigma_*}$  in the sense that we must satisfy

$$n_i y_i^{\Sigma_*} \leq g_i^{m-1} \equiv g_i - n_i v_i^{m-1}.$$

The following expression for the functional is obtained,

$$\begin{aligned} J_{\Sigma_*}(\mathbf{y}^{\Sigma_*}) &= J(\mathbf{v}^{m-1}) - \left( (M_0 - \frac{1}{2}A_0)M_0^{-1}I_0^T\mathbf{r}^{m-1}, M_0^{-1}I_0^T\mathbf{r}^{m-1} \right) \\ &\quad + \frac{1}{2}(E_0^T A E_0 \mathbf{y}^{\Sigma_*}, \mathbf{y}^{\Sigma_*}) - (E_0^T E_0^T \mathbf{r}^{m-1}, \mathbf{y}^{\Sigma_*}). \end{aligned}$$

It is clear that if one optimizes with respect to  $\mathbf{y}^{\Sigma_*}$  and defines  $\mathbf{v}^m = \mathbf{v}^{m-1} + \mathbf{y}^0 + \mathbf{y}_{\text{opt}}^{\Sigma_*}$  then

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

The latter is true since one may choose  $\mathbf{y}^{\Sigma_*} = 0$  and satisfy the constraints, therefore due to the optimization, one has

$$\frac{1}{2}(E_0^T A E_0 \mathbf{y}_{\text{opt}}^{\Sigma_*}, \mathbf{y}_{\text{opt}}^{\Sigma_*}) - (E_0^T E_0^T \mathbf{r}^{m-1}, \mathbf{y}_{\text{opt}}^{\Sigma_*}) \leq 0,$$

and due to the property (3.1) of  $M_0$ ,

$$\left( (M_0 - \frac{1}{2}A_0)M_0^{-1}I_0^T \mathbf{r}^{m-1}, M_0^{-1}I_0^T \mathbf{r}^{m-1} \right) \geq 0.$$

That is, the above algorithm provides a monotone scheme.

We mention at the end that one can devise a number of algorithms by choosing subspaces  $\mathbf{V}_k \subset \mathbf{V}$  and varying the decompositions  $\mathbf{V}_k = \mathbf{V}_{\Sigma_*}^k + \mathbf{V}_0^k$ . In our application  $\mathbf{V}^k$  will be coarse subspaces of  $\mathbf{V}$  and  $\mathbf{V}_{\Sigma_*}^k$  and  $\mathbf{V}_0^k$  will correspond to splittings of the degrees of freedom of  $\mathbf{V}^k$  into dofs on  $\Sigma_*$  and interior dofs. In general  $M_0$  will correspond to Gauss–Seidel sweeps over the interior dofs, whereas the optimization for the corresponding  $\mathbf{y}^{\Sigma_*}$  will be based either on the Projected Gauss–Seidel or on the Dostal’s algorithm.

#### 4. COARSENING AWAY FROM THE CONTACT BOUNDARY

We generate coarse spaces  $\mathbf{V}_k$   $k = 0, \dots, \ell$ , where  $\mathbf{V}^0 = \mathbf{V}$  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 [6]. They utilize certain element topological relations and create the same relations on coarse levels recursively. Details can be found in [10]. Here we use a modification of the agglomeration algorithm in a way that degrees of freedom associated with  $\Sigma_C$  are not coarsened. The original agglomeration algorithm from [6] allowed for barriers, that is, some faces of elements are labeled 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  $\Sigma_C$  are labeled as unacceptable. Then one labels all elements that touch such faces. Finally the faces of all such labeled elements are labeled as unacceptable. Thus at least one layer of elements near the contact boundary  $\Sigma_C$  are kept on the initial coarse level. On the next coarsening 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 on Fig. 4.1–4.3. The second part of the coarsening is to choose the coarse degrees of freedom (dofs). In this 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 [6]. It 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 this kind of coarsening is that if a fine–grid vector  $\mathbf{v} \in \mathbf{v}_k$  satisfies the constraints, its (pointwise) restriction  $\mathbf{v}|_{\text{coarse nodes}}$  also satisfies the constraints, since the elements near the contact boundary have not been changed.

#### 5. ALGORITHM

Here we present our FAS cycle. It is based on a smoothing procedure – the projected Gauss–Seidel algorithm and on a coarse grid solver – based on Dostal’s algorithm presented earlier.

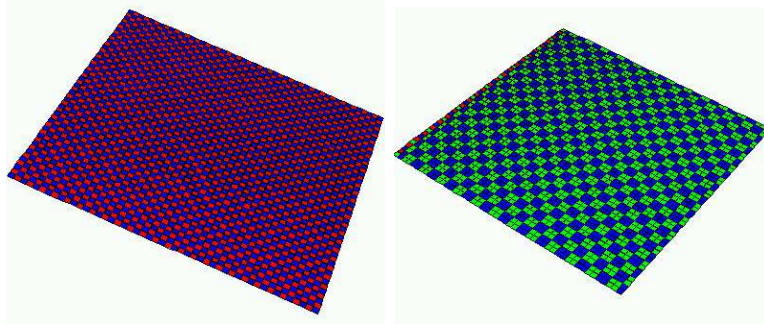


FIGURE 4.1. Level 0 and Level 1 of the agglomeration.

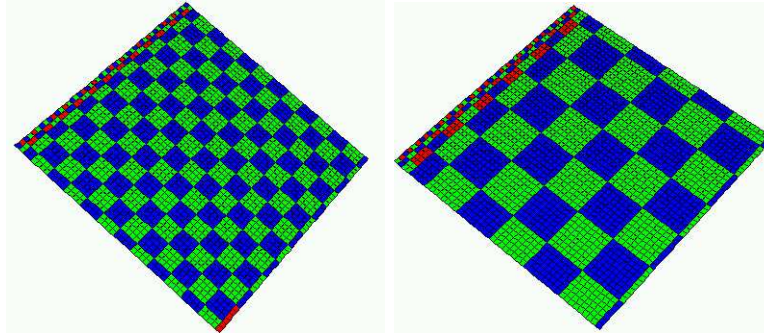


FIGURE 4.2. Level 2 and Level 3 of the agglomeration.

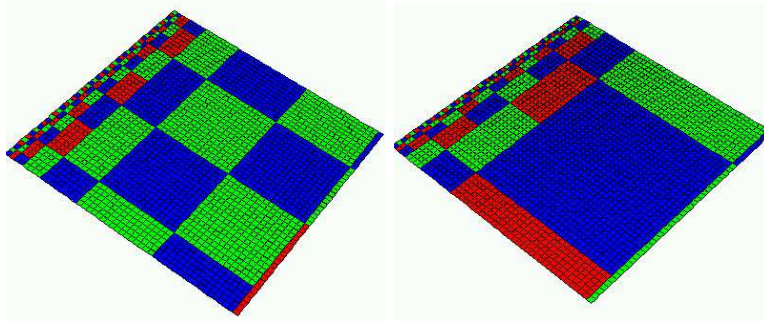


FIGURE 4.3. Level 4 and Level 5 of the agglomeration.

	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 $\Gamma_C$	100	100	100	100	100	100

TABLE 1. Agglomeration information.

**Algorithm 5.1.** FAS MULTIGRID CYCLE:

$$\mathbf{u}_k^{m+1} = \mathbf{FAS}(k, \mathbf{u}_k^m, \mathbf{A}_k, \mathbf{b}_k, \nu_1, \nu_2)$$

PRESMOOTHING - apply  $\nu_1$  times Projected Gauss–Seidel.

$$\bar{\mathbf{u}}_k^m = \mathbf{GS}^{\nu_1}(\mathbf{u}_k^m, \mathbf{A}_k, \mathbf{b}_k)$$

COARSE-GRID CORRECTION, (CGC)

- Take the restriction of  $\bar{\mathbf{u}}_k^m$  on the coarse mesh,  $\bar{\mathbf{u}}_{k+1}^m = \bar{\mathbf{u}}_k^m|_{\text{coarse nodes}}$
- Apply the restriction operator to the right-hand side  $\mathbf{b}_{k+1} = \mathbf{I}_k^{k+1} \mathbf{b}_k$   
if  $k+1 = \ell$  - apply Dostal's algorithm  $\mathbf{w}_{k+1}^m = \mathbf{Dostal}(\mathbf{u}_{k+1}^m, \mathbf{A}_{k+1}, \mathbf{b}_{k+1})$   
if  $k+1 < \ell$ , then perform recursion, that is, one computes

$$\mathbf{w}_{k+1}^m = \mathbf{FAS}(k+1, \mathbf{u}_{k+1}^m, \mathbf{A}_{k+1}, \mathbf{b}_{k+1}, \nu_1, \nu_2)$$

- Compute the correction  $\mathbf{v}_{k+1}^m = \mathbf{w}_{k+1}^m - \bar{\mathbf{u}}_{k+1}^m$
- Interpolate the correction  $\mathbf{v}_k^m = \mathbf{I}_{k+1}^k \mathbf{v}_{k+1}^m$
- Compute the corrected approximation  $\mathbf{u}_k^{m, \text{after CGC}} = \bar{\mathbf{u}}_k^m + \mathbf{v}_k^m$

POSTSMOOTHING- apply  $\nu_2$  times Projected Gauss–Seidel.

$$\mathbf{u}_k^{m+1} = \mathbf{GS}^{\nu_2}(\mathbf{u}_k^{m, \text{after CGC}}, \mathbf{A}_k, \mathbf{b}_k)$$

Based on the discussion in Section 3, it is clear that the above FAS algorithm provides a monotone algorithm. Its practically mesh-independent convergence is illustrated in the next section.

## 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  $\mathbf{u}$  such that:

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

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

$\mathbf{g}$ : initial gap between the body and the foundation.

$\mathbf{f}$ : body forces.

$\mathbf{t}$ : surface tractions applied to a portion of the body surface  $\Gamma_F$ .

$\Gamma_D$ : a portion of the boundary along which the body is fixed.

$\Gamma_C$ : 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).



$\boldsymbol{\sigma}$ : stress vector  $\sigma_{ij}(\mathbf{u}) = E_{ijkl}u_{k,l}$  - Hooke's law,  $E_{ijkl}$  - elasticities characterizing the material,  $E_{ijkl} \in L^\infty(\Omega)$ .

$\sigma_{T_i}$ : tangential component of the stress vector.

$\sigma_n$ : normal component of the stress vector.

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

$$J(\mathbf{v}) = \frac{1}{2}\mathbf{v}^t \mathbf{A} \mathbf{v} - \mathbf{v}^t \mathbf{b} \mapsto \min, \quad \mathbf{v} \in \mathbf{K},$$

$$\mathbf{K} = \{\mathbf{v} \in \mathbb{R}^{N.P} \mid n_i \cdot v_i \leq g_i, \quad i \in \Sigma_C\}.$$

$\Omega_h$  consists of simplex elements, over which each component of the displacement is approximated by linear polynomials.

P - total number of nodal points in  $\overline{\Omega}_h$ . N - dimension of the domain.

$\mathbf{A}$  - symmetric positive definite matrix.

$\mathbf{n}$  - unit length outward normal on  $\Gamma_C$ .

$g_i$  - the length of a vector beginning at the node corresponding to degree of freedom  $i$  with direction parallel to the normal vector at that node, ending at the crossing point with the rigid foundation.

$\Sigma_C$  -the set of indices of all degrees of freedom on  $\Gamma_C$ .

We will assume that all normal vectors  $\mathbf{n}^\theta$ ,  $\theta \in \Sigma_C$  are equal to a coordinate unit vector. This can be achieved by appropriate transformation (change of variables) at the nodes on  $\Sigma_C$  of the unknown displacement vector.

The algorithm given above has been implemented in a C++ code and have been applied to a Signorini's problem in 2-d 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 FAS iteration we apply 4 pre- and 4 postsmoothing steps of Projected Gauss-Seidel and Dostal's algorithm on the coarsest level. The algorithm terminates the iterations when  $\frac{|\mathbf{J}_{m+1} - \mathbf{J}_m|}{|\mathbf{J}_0|} \leq 10^{-5}$ . The numerical results are shown on Table 2. One can clearly see the almost mesh-independent number of FAS iterations. We finally mention that the cost of the coarse-grid algorithm is typically  $\mathcal{O}(|\Sigma_c| \times (\text{number of iterations}))$ , that is, proportional to the number of degrees of 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 the it is of order  $h^{-1}$ . That is overall, the cost is bounded by the total number of degrees of freedom (on the fine mesh).

## REFERENCES

- [1] C. Carstensen, O. Scherf, and P. Wriggers. Adaptive finite elements for elastic bodies in contact. *SIAM J. Sci. Comput.*, 20:1605-1626, 1999.
- [2] Z. Dostal. Box constrained quadratic programming with proportioning and projections. *SIAM J. Optim.*, 7:871-887,1997.

	Grid1	Grid2	Grid3	Grid4
# Dofs	200	800	1800	5000
# FAS iterations	3	4	3	5
# Grid levels	4	5	5	6
Grid complexity	2.57	2.12	1.84	1.69
Operator complexity	2.48	2.04	1.79	1.66

TABLE 2. Results of the numerical experiments.

- [3] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer, New York, 1984.
- [4] W. Hackbusch and H. D. Mittelmann. On multigrid methods for variational inequalities. *Numer. Math.*, 42:65-76, 1983.
- [5] R. H. W. Hoppe. Multigrid algorithms for variational inequalities. *SIAM J. Numer. Anal.*, 24:1046-1065, 1987.
- [6] Jones, J. E. and P. S. Vassilevski. AMGe based on element agglomeration. *SIAM Journal of Scientific Computing*, **23** (2001), pp. 109-133.
- [7] N. Kikuchi and J. T. Oden. *Contact problems in elasticity*. SIAM. 1988.
- [8] R. Kornhuber, R. Krause. Adaptive Multigrid Methods for Signorini's Problem in Linear Elasticity, *Comp. Visual. Sci.*, **4**, 9-20.
- [9] J. Schoberl. Solving the Signorini problem on the basis of domain decomposition techniques, *Computing*, 1998.
- [10] P. S. Vassilevski. Sparse matrix element topology with application to AMG and preconditioning, *Numerical Linear Algebra with Applications*, **9**(2002), pp. 429-444.

CENTER FOR APPLIED SCIENTIFIC COMPUTING, LAWRENCE LIVERMORE NATIONAL LABORATORY, P.O. BOX 808, L-561, LIVERMORE, CA 94551, U.S.A.