Computing and Visualization in Science manuscript No.
(will be inserted by the editor)

A Cascadic Multigrid Algorithm for Variational Inequalities

H. Blum[2], **D. Braess**[1], **and F.T. Suttmeier**[2]

1) Faculty of Mathematics, Ruhr-University, D-44780 Bochum, Germany
2) Universität Dortmund, Vogelpothsweg 87, D-44221 Dortmund, Germany

Received: date / Revised version: date

Abstract. When classical multigrid methods are applied to discretizations of variational inequalities, several complications are frequently encountered mainly due to the lack of simple feasible restriction operators. These difficulties vanish in the application of the cascadic version of the multigrid method which in this sense yields greater advantages than in the linear case. Furthermore, a cg-method is proposed as smoother and as solver on coarse meshes. The efficiency of the new algorithm is elucidated by test calculations for an obstacle problem and for a Signorini problem.

Key words multigrid method, contact problem, obstacle problem, finite element method.

1 Introduction

Contact problems, obstacle problems, Signorini's problem, and the dam problem are examples of problems that lead to variational inequalities. A typical example is

$$\int_{\Omega} [\frac{1}{2}(\nabla v)^2 - fv]dx \to \min! \tag{1}$$

$$v(x) = g(x) \quad \text{on} \quad \Gamma_D \subset \partial\Omega$$

with the one-sided restrictions

$$v(x) \geq \psi(x) \quad \text{for} \quad x \in \Omega_{\geq}.$$

Here $\Omega_{\geq}$ is either a subset of Ω or of $\partial\Omega$. Formally we are looking for a solution in a closed convex cone in $H^1(\Omega)$ and not in a linear space.

There is a natural interest to solve the inequalities which arise from a finite element discretization by the multigrid method. Unfortunately, there is a complication that is not encountered in linear problems. Let v_h^0 be the actual approximation in our multigrid iteration at the level with mesh size h. Then we are looking for the solution of a minimization problem in a set $v_h^0 + K_h$, where K_h is a convex cone whose boundary is given by a finite element function on the mesh with mesh size h. When we proceed to a coarser grid, we have also to coarsen the finite element function that specifies the cone. As a consequence, we get an approximation of the cone (in addition to the reduction of the dimension).

In the eighties (of the last century) there were used either inner approximations of the cone, cf. Hackbusch and Mittelmann [8], or outer approximations, cf. Brandt and Cryer [5] or Bollrath [2]. The inner approximation can lead to so small cones that the multigrid efficiency gets lost, and the (outer) approximation by a larger cone requires that a projection onto the feasible set is added, and multigrid efficiency can also not be guaranteed. These disadvantages were avoided by Kornhuber [11] who modified the basis functions at the frontier of the active set, cf. also [9].

We will take a different approach and use a *cascadic multigrid algorithm* avoiding the return to coarser grids in the iteration. Cascadic algorithms have been used successfully for linear problems [3,4,6], and the advantage to employ transfer operators between the grids in one direction only, is here even clearer than in the application to linear elliptic problems.

The crucial point of a successful usage of cascadic codes is that the number of smoothing steps is sufficiently large and well balanced on the coarser grids [6]. This feature is also essential when treating variational inequalities and makes the difference to the code in [1] where the computations on coarse grids were restricted to the subdomain of inactive points.

The cascadic iteration starts on the coarsest mesh, and there an iteration of cg-type is applied. We note that a different cg code was proposed by Dostál and Schöberl [7]. The iteration step in their cg-code as well as in our one is more expensive than in the linear case since a possible change of the set of active points implies that the update of the gradients requires an extra matrix-vector-multiplication. Nevertheless, we recommend cg-steps as solver on the coarsest level as well as in the smoothing procedures at the higher levels. The efficiency of this approach and the whole cascadic multigrid iteration will be documented by several numerical examples.

2 A Smoothing Procedure

The solution u of the variational problem (1) is characterized by the inequalities

$$(\nabla u, \nabla(v-u)) \geq (f, v-u) \qquad \forall v \in K, \tag{2}$$

where

$$K := \{v \in H^1(\Omega);\ v \geq \psi \text{ for } x \in \Omega_{\geq} \text{ and } v = g \text{ on } \Gamma_D\},$$

and $(.,.)$ denotes the inner product in $L_2(\Omega)$.

In the following, we apply the finite element method on a triangulation $\mathcal{T}_h$ of Ω. Let V_h be the space of (bi-)linear finite element functions on $\mathcal{T}_h$. The finite element solution u_h belongs to the induced cone

$$K_h := \{v \in V_h;\ v(x) \geq \psi_h \text{ for } x \in \Omega_{\geq}\}$$

and is characterized by

$$(\nabla u_h, \nabla(v-u_h)) \geq (f, v-u_h) \qquad \forall v \in K_h. \tag{3}$$

Here ψ_h denotes a suitable interpolant of the obstacle ψ.

Since (3) describes the minimum of an elliptic expression (function) on a closed convex set, there exists a unique solution of (3).

When the solution is determined, a nodal basis is used. For convenience, we identify a finite element function $v \in V_h$ with the vector v whose components are the nodal values

$$v_i := v(x_i).$$

Similarly, we have $\psi_i := \psi(x_i)$, defining the interpolant $\psi_I \in V_h$ of ψ.

Let A be the associated stiffness matrix. Then (2) is equivalent to the complementary relations

$$\begin{aligned} Av &\geq b, \\ v &\geq \psi_I, \\ (Av-b)'(v-\psi_I) &= 0. \end{aligned}$$

Here, we understand that $\psi_i = -\infty$ if there is no restriction at the point x_i.

Therefore the solution is a fixed point of the projected symmetrical Gauss–Seidel relaxation.

Projected Gauss–Seidel relaxation $S_h(v)$
for $i = 1, 2, \ldots, \dim(A)$ and for $i = \dim(A), \ldots, 2, 1$

$$\{$$
$$v_i \leftarrow \max\{\psi_i,\ v_i + \frac{\omega}{a_{ii}}(b_i - \sum_j a_{ij} v_j)\} \tag{4}$$
$$\}$$

Here, it is understood that on the right-hand side of (4) always the current value of v_j is inserted. Of course, the relaxation can be applied with a relaxation factor ω between 0 and 2.

The computing time for m sweeps of the symmetric relaxation is roughly the same as for $m + 1/2$ matrix-vector multiplications. When evaluating (4), we can use the sum with the terms $i < j$ from the previous evaluation in the backward sweep and those with $i > j$ in the forward sweep.

In each relaxation step the quadratic functional given in (1) is reduced. Therefore, a compactness argument yields that the iteration with the projected Gauss–Seidel relaxation (as a one-level algorithm) converges to the solution [5]. It is however also known that it converges as slowly as the standard Gauss–Seidel relaxation for linear elliptic problems.

3 The Multilevel Procedure

For the setup of a multigrid algorithm we assume that we have a sequence of mesh sizes

$$h_0 > h_1 > \ldots > h_L \tag{5}$$

and $h_{\ell-1} = 2h_\ell$ for $\ell = 0, 1, \ldots, L$. The associated finite element spaces are assumed to be nested

$$V_{h_0} \subset V_{h_1} \subset \ldots \subset V_{h_L}, \tag{6}$$

and often we will write V_ℓ instead of V_{h_ℓ}.

As already mentioned, an essential feature of the cascadic algorithm is that the number m_ℓ of smoothing steps at the level ℓ increases for small ℓ. In the case of full regularity, it is sufficient to have $m_{\ell-1} = (2+\varepsilon)m_\ell$; cf. [3]. Here we will be conservative for good reasons and choose

$$m_\ell = 3^{L-\ell} m_L\,. \tag{7}$$

Now we are ready to define our basic multilevel algorithm. Some improvements of the algorithm will be added later.

Cascadic Multigrid Algorithm.
1) Compute a good approximation v_0 of the solution for the coarsest level $\ell = 0$. [Here v_0 is considered as good if the residue is not larger than the expected residue for the final numerical solution on the highest level ($\ell = L$).] — It can be computed by $S_h(v)$ with a suitable overrelaxation factor ω.

2) for $\ell = 1, 2, \ldots, L$
{
prolongate $v_{\ell-1} \to v_\ell$,
apply $m_\ell = 3^{L-\ell} m_L$ steps of S_h,
improve the current solution
on the set of inactive points,
}

The smoothing procedure in the algorithm above may consist only of the relaxations as described before. An improvement by cg steps will be described below. In the test examples below, suitable values for ω for the solution on the coarsest level and for the smoothing step are determined experimentally and are chosen as $\omega = 1.5$ and $\omega \approx 0.8$ respectively.

Remark. Actually, it was not necessary on the levels with very coarse grids to execute all $m_\ell = 3^{L-\ell} m_L$ relaxation steps. The accuracy measured in terms of the residues was good enough earlier. A strategy for deciding whether an iterate is good enough, will be discussed below for the numerical examples.

4 A Conjugate Gradient Algorithm

The cascadic algorithms for linear problems usually have conjugate gradient iterations in their smoothing procedures [3,6,13]. Moreover, a cg-iteration can accelerate the computation on the coarsest level. For these purposes we suggest a cg-method whose efficiency will become apparent from the numerical results.

Although our algorithm differs from the algorithm of Dostál and Schöberl [7], its concept is based on similar ideas:

1. The acceleration by steps of cg-type is restricted to the inactive set since such a restriction makes the problem linear. — In the case that restrictions are never active, the computation becomes the same as in the linear case.

2. When a computation differs for the active and the inactive set, the distinction between the two categories cannot be based on the question whether $v_i = \psi_i$ or $v_i > \psi_i$. Points with $v_i > \psi_i$ and $v_i - \psi_i$ being small must be treated like points of the active set.

3. Although the complete history of a cg-iteration enters into its analysis, the information of the last two steps is sufficient for the computations. The current step produces the optimum in the two-dimensional affine space that is parallel to the span of the current preconditioned gradient and the correction from the preceding step.

For convenience, we write the cg-procedure for the side conditions

$$u_i \geq 0 \qquad \text{for all} \quad i.$$

The generalization to $u_i \geq \psi_i$ (for some i) is obvious. Since we are discussing a one-level procedure, there is no specification of a mesh size.

Conjugate Gradient Algorithm for the Obstacle Problem (cg-PSSOR).
Let u^0 be an initial guess.
Compute u^1 by applying one step of the restricted symmetrical Gauss–Seidel relaxation $u^1 \leftarrow S(u^0)$,
for $\nu = 1, 2, 3, \ldots,$
{
$\tilde{u}^{\nu+1} \leftarrow S(u^\nu)$,
$d^{\nu-1} \leftarrow u^\nu - u^{\nu-1}$,
$g^\nu \leftarrow \tilde{u}^{\nu+1} - u^\nu$,
for $= 1, 2, \ldots, \dim(A)$
{
$d_i^{\nu-1} \leftarrow g_i^\nu \leftarrow 0$ if $\max\{|d_i^{\nu-1}|, |g_i^\nu|\} \geq \tilde{u}_i^{\nu+1}$,
}
determine d^ν such that $\tilde{u}^{\nu+1} + d^\nu$ minimizes the quadratic function on the set $\tilde{u}^{\nu+1}+$ span $\{d^{\nu-1}, g^\nu\}$,
$u^{\nu+1} \leftarrow \tilde{u}^{\nu+1} + d^\nu$,
if $(u^{\nu+1} \not\geq 0)$
{
determine the largest number α
such that $\tilde{u}^{\nu+1} + \alpha d^\nu \geq 0$,
$u^{\nu+1} \leftarrow \tilde{u}^{\nu+1} + \alpha d^\nu$,
}
}

The 2-dimensional space span $\{d^{\nu-1}, g^\nu\}$ is attached to the provisional update $\tilde{u}^{\nu+1}$ (and not to u^ν as in the linear case). Therefore, the quadratic functional at $u^{\nu+1}$ is not larger than at $\tilde{u}^{\nu+1}$. Moreover, the step length factor α will not be small, since by construction

$$\tilde{u}^{\nu+1} + \alpha_1 d^{\nu+1} + \alpha_2 g^\nu \geq 0$$

whenever $|\alpha_1| \leq \frac{1}{2}$, $|\alpha_2| \leq \frac{1}{2}$. The components of $d^{\nu-1}$ and g^ν which could have spoiled the inequality above, have been set to zero.

5 Numerical Results

5.1 Example 1

As a first test example, we consider (2) on $\Omega := (0,1)^2$ with $f = 0$. The obstacle is given by

$$\psi(x) := \exp\left((y^2 - 1)^{-1}\right) ,$$

with $y = (x - x_m)/r$, $x_m = (0.5, 0.5)$, $r = 0.4$.

In Table 1, we compare results with the projected Gauss–Seidel iteration (PSSOR) and with its conjugate gradient version (cg-PSSOR) when they are applied as one-level iterations. For simplicity, the relaxation parameter is kept fixed for all meshes. One observes the cg version to be significantly superior to the simple iteration. Reducing the mesh size by a factor of two by global refinement leads to only a doubling of the iteration steps. This indicates that the condition number κ of the problem enters with $\mathcal{O}(\sqrt{\kappa})$, as it is expected from the linear case.

	cg-PSSOR		PSSOR	
DOF	iter.	res.	iter.	res.
1089	20	4.432-05	139	5.124-05
4225	36	7.488-05	448	8.335-05
16641	59	1.433-04	1439	1.647-04
66049	93	3.223-04	4393	3.276-04
263169	197	6.399-04	12106	6.497-04

Table 1. Example 1: Comparison of PSSOR and cg-PSSOR as one-level methods ($\omega = 1.5$). The finite element spaces have between 1089 and 263169 degrees of freedom.

In Table 2, we show the results for our cascadic iteration, comparing the performance of cg-PSSOR and PSSOR as smoothing procedures. The number of smoothing steps (iter.), the initial residual (start-res.) and the final residual (end-res.) are depicted for each level. On the coarsest level (with 25 DOF), the solution is computed nearly exactly by cg-PSSOR with $\omega = 1.5$ for both smoothing procedures.

At the start of the iteration at the level $\ell \geq 1$ the error is about $u_{2h} - u_h$ (with $h = h_\ell$). The discretization error of u_h (or $u_h - u_{h/2}$) is about $1/2$ of the number above; cf. the residues at the start on all levels. So a reduction of the residue by a factor of 10 at the finest level will lead to an approximate solution with an (algebraic) error about $1/5$ of the discretization error. This is achieved by a few smoothing steps. Thus the algorithm shows multigrid efficiency.

From these arguments we also derive a rule for the termination of the smoothing iterations at the lower levels. The idea is that we are allowed to stop when the residue on the level ℓ is about $1/10$ of the residue at the start at the final level L. This number is not known a priori, but we observe that the start-residue decreases by about a factor of about $1/2$ when going to a finer grid. For being conservative we terminate the iteration at the level ℓ when

$$\text{residue}_\ell \leq \frac{1}{10} 0.4^{L-\ell} \text{start-residue}_\ell .$$

In this example and in the next one the maximal number of smoothing steps occurs at the levels $\ell \approx L - 4$. This differs from the experience in [4].

We observe the PSSOR smoother to require about twice as many smoothing iterations to yield approximately the same accuracy as the cg-PSSOR version. — For a comparison of the computing time we refer to the end of the paper.

In Figure 1 the dependence of the cascadic iteration on the relaxation parameter ω is depicted, when the smoothing is performed with PSSOR. The dependency suggests to choose $\omega = 0.75$ in the following test calculations. Moreover, the value of ω smaller than 1 makes clear that the iterations here serve as smoothers and not as solvers.

	cg-PSSOR		
DOF	iter.	start-res.	end-res.
25	4	8.301-02	1.266-06
81	5	4.981-02	1.287-05
289	9	2.795-02	2.659-05
1089	15	8.804-03	1.607-05
4225	19	4.228-03	2.406-05
16641	24	2.130-03	3.276-05
66049	6	9.876-04	3.378-05
263169	3	4.889-04	3.041-05

	PSSOR				
DOF	iter.	start-res.	end-res.	iter.	end-res.
25	4	8.301-02	1.266-06	4	1.266-06
81	12	4.981-02	1.684-05	12	1.684-05
289	36	2.795-02	2.471-05	36	2.471-05
1089	92	8.805-03	2.207-05	92	2.207-05
4225	81	4.229-03	8.248-05	162	3.206-05
16641	27	2.131-03	9.838-05	54	5.366-05
66049	9	9.910-04	7.309-05	18	4.192-05
263169	3	4.926-04	6.279-05	4	4.365-05

Table 2. Example 1: Results for cascadic iteration, comparing the performance of cg-PSSOR and PSSOR as smoothing procedure ($\omega = 0.75$). — In all cases the cg-version with $\omega = 1.5$ was used as a solver on the coarsest level.

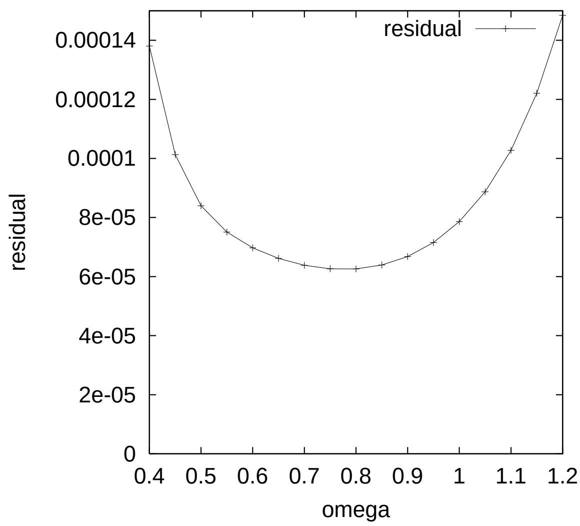

Fig. 1. Example 1: Residual on final grid with 263169 degrees of freedom obtained by the cascadic iteration with relaxation parameter ω on each level.

5.2 Example 2

Now we apply our ideas to Signorini's problem, a fundamental model situation for contact problems in elasticity.

The corresponding classical notation reads (c.f. Kikuchi and Oden [10])

$$-\operatorname{div}\sigma = f, \quad A\sigma = \varepsilon(u) \text{ in } \Omega,$$

with the boundary conditions

$$\left.\begin{array}{r} \sigma_T = 0, \quad (u_n - g)\sigma_n = 0 \\ u_n - g \leq 0, \quad \sigma_n \leq 0 \end{array}\right\} \text{ on } \Gamma_C,$$

and $u = 0$ on $\Gamma_D, \sigma \cdot n = t$ on Γ_N.

This idealized model describes the deformation of an elastic body occupying the domain $\Omega \subset \mathbb{R}^3$, which is unilaterally supported by a frictionless rigid foundation. The displacement u and the corresponding stress tensor σ are caused by a body force f and a surface traction t along Γ_N. Along the portion Γ_D of the boundary the body is fixed and $\Gamma_C \subset \partial\Omega$ denotes the part which is a candidate contact surface. We use the notation $u_n = u{\cdot}n$, $\sigma_n = \sigma_{ij}n_in_j$ and $\sigma_T = \sigma \cdot n - \sigma_n n$, where n is the outward normal of $\partial\Omega$, and g denotes the gap between Γ_C and the foundation.

Further, the deformation is supposed to be small, so that the strain tensor can be written as $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)$. The compliance tensor A is assumed to be symmetric and positive definite.

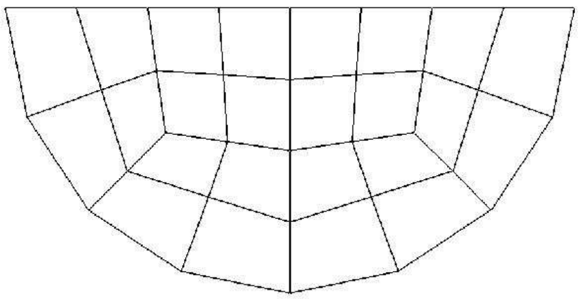

Fig. 2. Coarse grid for Hertz contact problem

	cg-PSSOR		PSSOR	
DOF	iter.	res.	iter.	res.
834	47	6.476+00	543	8.265+00
3202	80	1.196+01	1651	1.204+01
12546	145	1.682+01	4971	1.726+01
49666	265	2.450+01	15123	2.459+01
197634	480	3.477+01	45221	3.489+01

Table 3. Example 2: Comparison between PSSOR and cg-PSSOR on one level, i.e., as solvers of the discrete problems. ($\omega = 1.5$)

Following Kornhuber and Krause [12], we consider as a test example, a plane strain problem for a half circle with radius 0.4, centered at the point (0; 0.4) in contact with a rigid plane (Figure 2). The material constants are given by $E = 270269 N/mm^2$ for Young's modulus and $\nu = 0.248$ for Poisson's ratio. For our test calculations we choose a displacement of (0; −0.05).

The results are presented the same way as for the example above.

Again, the superiority of the cg-PSSOR-method can be observed for a one level iteration; see Table 3. Within a cascadic procedure; see Table 4, again we observe the PSSOR smoother to require about twice as many smoothing iterations to reach approximately the same accuracy as the cg-PSSOR version.

	cg-PSSOR		
DOF	iter.	start-res.	end-res.
22	6	1.454+04	2.178+00
66	10	1.409+03	2.299-01
226	15	2.237+03	9.693-01
834	23	1.111+03	2.782+00
3202	30	4.883+02	3.061+00
12546	13	2.378+02	3.643+00
49666	7	1.152+02	3.939+00
197634	3	5.700+01	4.308+00

	PSSOR				
DOF	iter.	start-res.	end-res.	iter.	end-res.
22	15	1.454+04	1.631+00	15	1.631+00
66	35	1.409+03	5.176-01	35	5.176-01
226	106	2.237+03	2.251+00	106	2.251+00
834	243	1.111+03	4.602+00	291	2.831+00
3202	81	4.890+02	7.801+00	162	4.684+00
12546	27	2.380+02	7.818+00	54	4.236+00
49666	9	1.152+02	8.768+00	18	4.895+00
197634	3	5.696+01	1.082+01	6	4.920+00

Table 4. Example 2: Results for cascadic iteration, comparing the performance of cg-PSSOR and PSSOR as smoothing procedure. ($\omega = 0.85$)

The numerical results for Signorini's problem show a similar behavior as in Example 1. In particular, a few smoothing steps are sufficient to reduce the residue by a factor of about 10. The iteration yields a numerical solution with an error that is smaller than the discretization error.

Remark: Taking into account that one step of cg-PSSOR requires as many operations as two to three steps of PSSOR, the performance of both schemes can be regarded as equal for our examples. For more involved problems, for example anisotropic material behavior, the cg-PSSOR is expected to be more robust as known from the linear case.

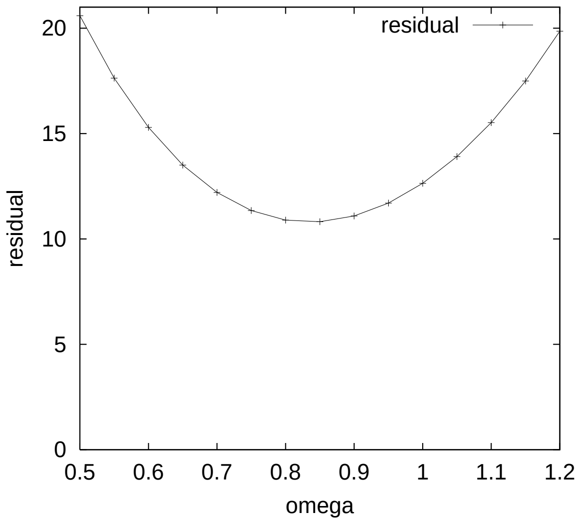

Fig. 3. Example 2: Residual on final grid with 197634 degrees of freedom obtained by the cascadic iteration with relaxation parameter ω on each level.

References

1. H. Blum and F.T. Suttmeier. An adaptive finite element discretisation for a simplified Signorini problem. *Calcolo*, 37(2):65–77, 2000.
2. C. Bollrath. Two multi-level algorithms for the dam problem. In D. Braess, W. Hackbusch, and U. Trottenberg, editors, *Advances in multi-grid methods*. Vieweg & Sohn, Braunschweig/Wiesbaden, 1984.
3. F.A. Bornemann and P. Deuflhard. The cascadic multigrid method for elliptic problems. *Numer. Math.*, 75:135–152, 1996.
4. D. Braess, P. Deuflhard, and K. Lipnikov. A subspace cascadic multigrid method for mortar elements. *Computing* 69, 205–225, 2002.
5. A. Brandt and C.W. Cryer. Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems. *SIAM J. Sci. Stat. Comput.*, 4:655–684, 1980.
6. P. Deuflhard. Cascadic conjugate gradient methods for elliptic partial differential equations: algorithm and numerical results. In D. Keyes and J. Xu, editors, *Domain Decomposition Methods in Scientific and Engineering Computing*, volume 180 of *AMS Series*, pages 29–42, 1994.
7. Z. Dostál and J. Schöberl. Minimizing quadratic functions subject to bound constraints. Technical report, Tech. Univers. Ostrava, 2002.
8. W. Hackbusch and H.D. Mittelmann. On multi-grid methods for variational inequalities. *Numer. Math.*, 42:65–76, 1983.
9. R.H.W. Hoppe and R. Kornhuber. Adaptive multilevel methods for obstacle problems. *SIAM J. Numer. Anal.*, 31(2):301–323, 1994.
10. N. Kikuchi and J.T. Oden. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. Studies in Applied Mathematics 8. SIAM, 1988.
11. R. Kornhuber. Monotone multigrid methods for variational inequalities I. *Numer. Math.* 69, 167–184, 1994
12. R. Kornhuber and R. Krause. Adaptive multigrid methods for Signorini's problem in linear elasticity. *Comp. Visual. Sci.*, 4:9–20, 2001.
13. R. Stevenson. Nonconforming finite elements and the cascadic iteration. *Numer. Math.*, 91:351–387, 2002.