Solving the Signorini Problem on the Basis of Domain Decomposition Techniques

Joachim Schöberl

August 21, 1996

Abstract

In recent years a couple of very efficient domain decomposition preconditioners have been developed. One of the basic principle is the approximative decoupling of the global boundary value problem into inner Dirichlet subproblems and the Schur-complement problem on the boundary.

The Finite Elemente Discretization of the Signorini Problem leads to a finite dimensional constrained minimization problem. This problem can be solved by the projection method, which is very similar to Richardson iteration. Because of the slow convergence rate on fine meshes preconditioning should be applied. To be effective, the relative condition number of the system matrix with respect to the preconditioning matrix has to be small, and the applications of the preconditioner as well as the projection onto the set of feasible elements have to be fast computable. In this paper it is shown how to build and analyse such preconditioners on the basis of Domain Decomposition techniques.

The results can be extended to elastic contact problems easily. The numerical results obtained for the Signorini problem as well as for the plane elasticity problem confirm the theoretical analysis quite well. Additional examples show the combination with nested and adaptive techniques.

1 Introduction

The contact problem is an important problem in computational mechanics. An elastic body is deformed due to volume and surface forces, but the body must not penetrate into a given, rigid obstacle (see Fig. 1). This leads to unilateral boundary conditions, called contact conditions. For modelling and analysis see [25], [20], [19], [6], [15]. Discretization methods are studied in [7], [4], [9], [8].

In this paper we are interested in numerical algorithms to solve the finite dimensional constrained minimization problem. There are classical iterative methods like point projection methods and point overrelaxation methods ([7]). These methods suffer from slow

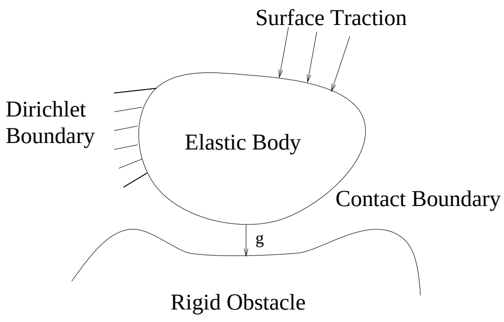

Figure 1: Body with obstacle

convergence rates on fine meshes. Multigrid methods have been successfully applied to obstacle problems with inequality constraints in the whole domain by [13], [23], [18], [17]. Domain decomposition methods for variational inequalities have been investigated in [28], [16], [1]. But these domain decompostion methods differ from our method based on domain decomposition operators. Boundary element methods have been applied for contact problems in [14], [27].

For the sake of simplicity, we will consider the Signorini Problem for the Poisson equation to develop and analyze the algorithms, but we also provide numerical results for the contact problem in elasticity.

For the Signorini problem, the unknown function u is restricted from below on the Signorini part of the boundary. If the restriction is active in a given point, we may have a positive flux into the domain, otherwise the flux vanishes. In classical form, the Signorini problem reads as:

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \\ u &= 0 && \text{on } \Gamma_D \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \Gamma_N \\ \left.\begin{aligned} u &\geq g \\ \tfrac{\partial u}{\partial n} &\geq 0 \\ \tfrac{\partial u}{\partial n}(u-g) &= 0 \end{aligned}\right\} && && \text{on } \Gamma_C. \end{aligned} \tag{1}$$

The domain Ω is bounded in $\mathbf{R}^d$, $d = 2$ or 3 with a smooth boundary $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_C$ such that $meas(\Gamma_D) \neq 0$ and $meas(\Gamma_C) \neq 0$. The right hand side f is called source term, and g bounds the unknown function u from below.

The problem is transformed into its weak form, where the analysis in the framework of variational inequalities can be carried out. The weak form is also the basis for the finite element (FE) discretization. To solve the finite dimensional constrained minimization problem, the projection method is applied. This method is not very popular in practice, because its convergence rate is very slow on fine meshes, unless good preconditioning is applied. The goal of this paper is to present preconditioning techniques satisfying the following properties:

- The relative condition number of the stiffness matrix A with respect to C is small,
- the operation $C^{-1} \times \underline{v}$ is fast computable,
- the projection P with respect to the C inner product onto the convex set defined by the restrictions is fast computable,

where the preconditioning matrix C is supposed to be symmetric and positive definite (spd). Multilevel (ML) preconditioners as well as domain decomposition (DD) preconditioners are designed to meet the first two requirements very well. The basic concept of Dirichlet DD is the approximative decoupling of the global FE space into inner and extended boundary subspaces. The projection involves only the boundary subspace. To implement this projection, the dual problem is introduced. This enables us to use well known boundary preconditioners. To avoid two cascaded iterations until convergence, we can do a step similar to the transition from Uzawa's Algorithm to the Arrow Hurwitz Algorithm (see [21]). The convergence of this final algorithm is analyzed by means of the Bramble Pasciak transformation, introduced in [3].

If the underlying components are optimal and a regular refined mesh is used, then the complexity is of optimal order for the two as well as three dimensional case.

In Section 2 the weak form and the FE discretization are given. The projection algorithm and an approximative variant are presented in Section 3. Convergence in enegy norm is proved. The concepts of additive DD preconditioning is shortly repeated in Section 4. In Section 5 both concepts are brought together to develop an optimal preconditioner for the projection algorithm. The truncated variant is analyzed in Section 6. Finally, numerical results are presented in Section 7.

2 Weak Forms and Discretization

Similar to the case of linear partial differential equations, weak forms are much better suited for analysis as well as for Finite Element (FE) discretization. Instead of dealing with variational equations, we have to consider variational inequalities, and the minimization problem for the Ritz functional is replaced by a constrained minimization problem.

Let us define the subspace

$$V := \left\{ v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D \right\}$$

of the Sobolev space $H^1(\Omega)$ equipped with the inner product $(.,.)_1$ and the duality product $\langle .,. \rangle : V^* \times V \to \mathbf{R}$, and the convex, closed, non-empty subset of feasible functions

$$K := \{ v \in V : v \geq g \quad \text{a.e. on } \Gamma_C \} .$$

On V, we define the symmetric, continuous and elliptic bilinearform a and the continuous functional f:

$$a(u,v) := \int_\Omega \nabla u \nabla v \mathrm{dx} \qquad \langle f, v\rangle := \int_\Omega f v \mathrm{dx}$$

If additional smoothness assumptions are fulfilled, then a solution of the classical problem is a solution of both of the following two, equivalent problems and vica verse:

Constrained Minimization Problem (CMP):

$$\text{Find } u \in K : \quad J(u) = \inf_{v \in K} J(v)$$

$$J(v) := \frac{1}{2} a(v,v) - \langle f, v\rangle \tag{2}$$

Variational Inequality (VI):

$$\text{Find } u \in K : a(u, v-u) \geq \langle f, v-u\rangle \quad \forall v \in K$$

Now we will discretize the weak forms by the Finite Element (FE) method (see [5]). We need a regular triangulation $\mathcal{T}_h$ in the sense of [5] of our domain Ω, where h is the usual discretization parameter, N denotes the total number of mesh nodes and N_C denotes the number of nodes on the Signorini boundary Γ_C. We set $V_h = span\{p_i\}$ with the piecewise linear nodal basis functions p_i. With the help of this basis we can identify a finite element function $v_h \in V_h$ with its counterpart $\underline{v} = (v_i)_{i=1\dots N}$ in $\mathbf{R}^N$ by $v_h = \sum\limits_{i=1}^{N} v_i p_i$. The bilinearform a and the linear functional f define the matrix A and the right hand side vector $\underline{f}$, respectively:

$$\begin{aligned} A &= (a_{ij})_{i,j=1\dots N}, & a_{ij} &= a(p_i, p_j) \\ \underline{f} &= (f_i)_{i=1\dots N}, & f_i &= \langle f, p_i\rangle \end{aligned}$$

The lower bound function g on Γ_C is approximated by a finite element function g_h in $V_{C_h} = \mathrm{tr} V_h$ and corresponds to the vector $\underline{g}_C \in \mathbf{R}^{N_C}$. According to ([4] it is enough to demand the inequalities only in the mesh nodes. This leads to the set

$$K_h := \left\{ v_h \in V_h : \underline{v}_C \geq \underline{g}_C \right\}$$

of feasible finite element functions. We denote the part of the vector $\underline{v}$ according to the nodes on the Signorini boundary Γ_C by $\underline{v}_C$. $\underline{v}_C \geq \underline{g}_C$ is understood as $\underline{v}_{C_i} \geq \underline{g}_{C_i}$, $i = 1 \dots N_C$. Therefore, we arrive at the finite dimensional, constrained minimization problem:

$$\text{Find } \underline{u} \in \mathbf{R}^N, \underline{u}_C \geq \underline{g}_C : \quad J(\underline{u}) = \inf_{\substack{\underline{v} \in \mathbf{R}^N \\ \underline{v}_C \geq \underline{g}_C}} J(\underline{v}) \tag{3}$$

with

$$J(\underline{v}) := \frac{1}{2} \underline{v}^t A \underline{v} - \underline{f}^t \underline{v}.$$

From now on, we will skip the indices h for FE functions and underscores for vectors in $\mathbf{R}^N$.

3 The Projection Method

The finite dimensional, constrained optimization problem can be solved by the projection method which reads as:

Algorithm 1 (Projection Method)

Choose an arbitrary initial guess $u^1 \in K$,
For $k = 1, 2, \ldots$ *do*

$$u^{k+1} = P\left(u^k + \tau C^{-1}\left(f - Au^k\right)\right)$$

In Algorithm 1, C is the symmetric and positive definite (spd) preconditioning matrix. We assume the spectral equivalence inequalities

$$\underline{\alpha} C \leq A \leq \overline{\alpha} C \tag{4}$$

in the sense of Euklidean inner product, with the positive spectral equivalence constants $\underline{\alpha}$ and $\overline{\alpha}$. In general, these bounds depend on the mesh parameter h. If they are independent of h, then we call the preconditioner C asymptotically optimal. Asymptotically optimal preconditioners can be constructed by multilevel techniques as well as by multilevel - domain decomposition techniques. This ML-DD preconditioner will be presented in Section 4.

We assume, that the real, positve relaxation parameter τ is chosen such that $\tau\overline{\alpha} < 1$. P is the projection opererator onto K with respect to the matrix norm induced by C:

$$P : w \in V \rightarrow P(w) \in K : \quad \|P(w) - w\|_C \leq \|v - w\|_C, \ \forall v \in K \tag{5}$$

It is straight forward to show that u^k converges to its limit u in C-energy norm with convergence factor $1 - \tau\underline{\alpha}$ (see e.g. [8]).

A convergence estimation in A-energy norm is necessary for the composite algorithm in Section 6. However, such A-energy norm convergence estimates were not available from the literature. It could not be shown that u^k converges to u in A-norm, but the following theorem provides a monotone decay of the quadratic functional J:

Theorem 1 (Energy convergence)
u^k *is the sequence generated by Algorithm 1. Then*

$$J(u^{k+1}) \leq \rho J\left(u^k\right) + (1 - \rho)\, J(u) \tag{6}$$

hold for all $k \in \mathbf{N}$ *with the convergence rate*

$$\rho = 1 - \frac{\tau\underline{\alpha}}{2}. \tag{7}$$

Proof: We define the Richardson step

$$\tilde{u}^k := u^k + \tau C^{-1}\left(f - Au^k\right)$$

and the approximation

$$J^k(v) := \frac{1}{2\tau}\|v - \tilde{u}^k\|_C^2 - \frac{1}{2\tau}\|u^k - \tilde{u}^k\|_C^2 + J(u^k)$$

of J. Because of the definition of τ we have $J^k(v) \geq J(v)\ \forall v \in V$, with equality in u^k.

Using the last inequality for $v := u^{k+1}$, the convexity of K for the inclusion of the stright line $[u, u^k] \subset K$, and the definition of the projection P applied to $u^{k+1} = P(\tilde{u}^k)$, we get:

$$\begin{aligned}
J(u^{k+1}) &\leq J^k(u^{k+1}) \\
&= \frac{1}{2\tau}\|u^{k+1} - \tilde{u}^k\|_C^2 - \frac{1}{2\tau}\|u^k - \tilde{u}^k\|_C^2 + J(u^k) \\
&\leq \frac{1}{2\tau}\|P_{[u,u^k]}(\tilde{u}^k) - \tilde{u}^k\|_C^2 - \frac{1}{2\tau}\|u^k - \tilde{u}^k\|_C^2 + J(u^k) \\
&= -\frac{1}{2\tau}\underbrace{\left(P_{[u,u^k]}(\tilde{u}^k) - \tilde{u}^k, u^k - P_{[u,u^k]}(\tilde{u}^k)\right)_C}_{\geq 0} \\
&\quad -\frac{1}{2\tau}\left(\tilde{u}^k - u^k, P_{[u,u^k]}(\tilde{u}^k) - u^k\right)_C + J(u^k) \\
&\leq -\frac{1}{2\tau}\left(\tilde{u}^k - u^k, P_{[u,u^k]}(\tilde{u}^k) - u^k\right)_C + J(u^k)
\end{aligned}$$

$P_{[u,u^k]}(\tilde{u}^k) - u^k$ is a scalar multiple of $u - u^k$, and so we can write

$$\left(\tilde{u}^k - u^k, P_{[u,u^k]}(\tilde{u}^k) - u^k\right)_C = \left(\tilde{u}^k - u^k, u - u^k\right)_C \frac{\left(P_{[u,u^k]}(\tilde{u}^k) - u^k, u - u^k\right)_C}{(u - u^k, u - u^k)_C}.$$

We distinguish the three possible cases

$$\left(\tilde{u}^k - u^k, u - u^k\right)\begin{cases} \leq 0 \\ \geq \|u - u^k\|_C^2 \\ \textit{else} \end{cases}$$

and see that

$$\left(P_{[u,u^k]}(\tilde{u}^k) - u^k, u - u^k\right)_C = \max\left\{0, \min\left\{\|u - u^k\|_C^2, \left(\tilde{u}^k - u^k, u - u^k\right)_C\right\}\right\}$$

Using the definition of $\tilde{u}^k$, the variational specification of the solution u and the spectral equivalence constant $\underline{\alpha}$, we get

$$\begin{aligned}
&\left(\tilde{u}^k - u^k, u - u^k\right)_C = \tau\left(f - Au^k, u - u^k\right)_{\mathbf{R}^N} \\
&= \tau\left(f - Au, u - u^k\right)_{\mathbf{R}^N} + \tau\left(A(u - u^k), u - u^k\right)_{\mathbf{R}^N} \\
&\geq \tau\underline{\alpha}\|u - u^k\|_C^2.
\end{aligned}$$

Using $0 \le \tau\underline{\alpha} \le 1$ we get

$$\left(\tilde{u}^k - u^k, P_{[u,u^k]}(\tilde{u}^k) - u^k\right)_C \ge \tau\underline{\alpha}\left(\tilde{u}^k - u^k, u - u^k\right)_C.$$

We go on with the original quantity

$$\begin{aligned}
J(u^{k+1}) &\le -\frac{\underline{\alpha}}{2}\left(\tilde{u}^k - u^k, u - u^k\right)_C + J(u^k) \\
&= -\frac{\tau\underline{\alpha}}{2}\left(f - Au^k, u - u^k\right)_{\mathbf{R}^N} + J(u^k) \\
&= \left(1 - \frac{\tau\underline{\alpha}}{2}\right) J(u^k) + \frac{\tau\underline{\alpha}}{2} J(u) \\
&\quad + \frac{\tau\underline{\alpha}}{2}\left(\frac{1}{2}\left(Au^k, u^k\right)_{\mathbf{R}^N} - \left(f, u^k\right)_{\mathbf{R}^N} - \frac{1}{2}(Au, u)_{\mathbf{R}^N} + (f, u)_{\mathbf{R}^N}\right) \\
&\quad - \frac{\tau\underline{\alpha}}{2}\left(f - Au^k, u - u^k\right)_{\mathbf{R}^N} \\
&= \rho J(u^k) + (1-\rho)\, J(u) - \frac{1}{2}\|u - u^k\|_A^2 \\
&\le \rho J(u^k) + (1-\rho)\, J(u)
\end{aligned}$$

□

For our application the exact projection P is too expensive to compute, and so it is replaced by an approximative projection $\tilde{P}$. This leads us to the approximative projection method:

Algorithm 2 (Approximative Projection Method)
Choose an arbitrary $u^1 \in K$
For $k = 1, 2, \ldots$ *do*

$$\begin{aligned}
\tilde{u}^k &= u^k + \tau C^{-1}\left(f - Au^k\right) \\
u^{k+1} &= \tilde{P}\left(\tilde{u}^k\right)
\end{aligned}$$

Theorem 2 (Convergence of approximative projection method)
If the approximative projection $\tilde{P}$ *fulfills*

$$\|\tilde{P}\left(\tilde{u}^k\right) - \tilde{u}^k\|_C^2 \le \rho_2\|u^k - \tilde{u}^k\|_C^2 + (1-\rho_2)\,\|P\left(\tilde{u}^k\right) - \tilde{u}^k\|_C^2, \tag{8}$$

with $\rho_2 \in (0,1)$ *then the estimates*

$$J\left(u^{k+1}\right) \le \rho J\left(u^k\right) + (1-\rho) J\left(u\right) \tag{9}$$

hold for all $k \in \mathbf{N}$ *with the setting*

$$\begin{aligned}
\rho_1 &= 1 - \frac{\tau\underline{\alpha}}{2} \\
\rho &= \rho_1 + \rho_2 - \rho_1\rho_2
\end{aligned}$$

Proof: Similar to the proof of Theorem 1 we have

$$
\begin{aligned}
J(u^{k+1}) &\leq \frac{1}{2\tau}\|u^{k+1} - \tilde{u}^k\|_C^2 - \frac{1}{2\tau}\|u^k - \tilde{u}^k\|_C^2 + J\left(u^k\right) \\
&\leq \frac{1}{2\tau}\left(\rho_2\|u^k - \tilde{u}^k\|_C^2 + (1-\rho_2)\|P(\tilde{u}^k) - \tilde{u}^k\|_C^2\right) - \frac{1}{2\tau}\|u^k - \tilde{u}^k\|_C^2 + J\left(u^k\right) \\
&= (1-\rho_2)\underbrace{\left[\frac{1}{2\tau}\|P(\tilde{u}^k) - \tilde{u}^k\|_C^2 - \frac{1}{2\tau}\|u^k - \tilde{u}^k\|_C^2 + J\left(u^k\right)\right]}_{\leq \rho_1 J(u^k) + (1-\rho_1)J(u)} + \rho_2 J(u^k) \\
&\leq \rho_1 + \rho_2 - \rho_1\rho_2 J(u^k) + (1-\rho_1)(1-\rho_2)J(u) \\
&= \rho J(u^k) + (1-\rho)J(u)
\end{aligned}
$$

□

For practical computation we need a computable estimation for the iteration error. We get the same error estimator as for iterativ methods for linear systems.

Corollary 1
The sequence u^k is generated by Algorithm 1 or by Algorithm 2. Then the error of the iteration value u^{k+1} is bounded by

$$
\|u - u^{k+1}\|_A^2 \leq \frac{2\rho}{1-\rho}\left(u^{k+1} - u^k, f - Au^k\right)_{\mathbf{R}^N}, \tag{10}
$$

with ρ from Theorem 1 or Theorem 2, respectively.

Proof: By the variational definition of u we have

$$
\frac{1}{2}\|u - v\|_A^2 \leq J(v) - J(u) \quad \forall v \in K,
$$

and by Theorem 1 or Theorem 2 we have

$$
\begin{aligned}
&(1-\rho)(J(u^{k+1}) - J(u)) \leq \rho(J(u^k) - J(u^{k+1})) \\
&\quad = \rho\left(-\frac{1}{2}\|u^{k+1} - u^k\|_A^2 + \left(u^{k+1} - u^k, f - Au^k\right)_{\mathbf{R}^N}\right) \\
&\quad \leq \rho\left(u^{k+1} - u^k, f - Au^k\right)_{\mathbf{R}^N},
\end{aligned}
$$

which finishes the proof. □

4 Domain Decomposition Preconditioning

A lot of work has been spent to build fast preconditioners with close spectral equivalence constants $\underline{\alpha}$ and $\overline{\alpha}$, such that the spectral equivalence inequalities $\underline{\alpha}C \leq A \leq \overline{\alpha}C$ hold. We are interested in two types of modern preconditioners, in multigrid and in domain decomposition ones. Multigrid ensures condition numbers independent of the discretization parameter h. The domain decomposition strategy provides us with a splitting of the FE - space V into inner (V_I) and extended boundary (V_C) subspaces, which is useful for parallel computing. We will present the approximative DD preconditioner introduced in [10].

We reorder our basis to split vectors and matrices into contact boundary and inner parts:

$$A = \begin{pmatrix} A_C & A_{CI} \\ A_{IC} & A_I \end{pmatrix} \qquad u = \begin{pmatrix} u_C \\ u_I \end{pmatrix} \qquad f = \begin{pmatrix} f_C \\ f_I \end{pmatrix}$$

These two subspaces of V are far away from being orthogonal with respect to energy norm. To get an idea about orthogonality we have to go back to the original equation stated in the Sobolev space $H^1(\Omega)$. The two subspaces $H^1_0(\Omega)$ and the space of the harmonic functions are orthogonal with respect to the H^1-half norm. The space V_I is an approximation to $H^1_0(\Omega)$. To approximate the harmonic subspace one takes

$$\tilde{V}_C := \{Ev_C : v_C \in V_C\},$$

where $E = \begin{pmatrix} I_C \\ E_I \end{pmatrix}$ is a computable operator which approximates the discrete harmonic extension of a given boundary function. E is bounded by c_E in the sense of

$$\|Ev_C\|_A \leq c_E \inf_{\substack{w \in V \\ w_C = v_C}} \|w\|_A = c_E \|v_C\|_{S_C}$$

with the boundary - Schur complement

$$S_C = A_{CC} - A_{CI} A_{II}^{-1} A_{IC}.$$

For the two subspaces $V_1 = \tilde{V}_C$ and $V_2 = V_I$ we have the strengthend Cauchy-Schwarz inequality

$$(v_1, v_2)_A \leq \gamma \|v_1\|_A \|v_2\|_A \qquad \forall v_1 \in V_1, \forall v_2 \in V_2$$

with

$$\gamma = \sqrt{1 - \frac{1}{c_E^2}}.$$

We define the preconditioner of the exact Additive Schwarz Method (ASM) by the following procedure:

$$\boxed{\begin{array}{rl} \text{Given:} & d \in V^* \\ \text{Calculate:} & w = C^{-1} d \in V \\ & w = w_1 + w_2 \\ & (w_i, v_i)_A = \langle d, v_i \rangle \quad \forall v_i \in V_i, i = 1, 2 \end{array}}$$

For the exact ASM preconditioner we get the spectral equivalence inequalities

$$(1 - \gamma) C \leq A \leq (1 + \gamma) C. \tag{11}$$

It would be too expensive, to solve the two subproblems exactly, so the bilinearforms are replaced by preconditioners C_i on the individual subspaces. They fulfill the inequalities

$$\underline{\alpha}_i C_i \leq A \leq \overline{\alpha}_i C_i \quad \text{on} V_i.$$

This leads us to the inexact ASM preconditioner defined by

Given:	$d \in V^*$
Calculate:	$w = C^{-1} d \in V$
	$w = w_1 + w_2$
	$(w_i, v_i)_{C_i} = \langle d, v_i \rangle \quad \forall v_i \in V_i, i = 1, 2$

fulfilling

$$(1-\gamma) \min_{i=1,2} \{\underline{\alpha}_i\} C \leq A \leq (1+\gamma) \max_{i=1,2} \{\overline{\alpha}_i\} C.$$

The bilinearfrom $(w_1, v_1)_A$ on V_1 can be expressed by the bilinearform $(w_C, v_C)_{E^t AE}$ living only on the boundary. We need a preconditioner C_C approxiamting $E^T AE$. For the space V_2 we need a preconditioner for the Dirichlet problem. If we assume the spectral bounds

$$\begin{array}{rcll} \underline{\alpha}_C C_C & \leq & E^t AE & \leq \overline{\alpha}_C C_C \quad \text{on } V_C \\ \underline{\alpha}_I C_I & \leq & A & \leq \overline{\alpha}_I C_I \quad \text{on } V_I \end{array}$$

for these two components, then the overall preconditioner

$$C^{-1} = \begin{pmatrix} I_C & 0 \\ E_I & I_I \end{pmatrix} \begin{pmatrix} C_C^{-1} & 0 \\ 0 & C_I^{-1} \end{pmatrix} \begin{pmatrix} I_C & E_I^t \\ 0 & I_I \end{pmatrix} \tag{12}$$

holds

$$\underline{\alpha} C \leq A \leq \overline{\alpha} C$$

with the constants

$$\begin{aligned} \underline{\alpha} &= \left(1 - \sqrt{1 - \frac{1}{c_E^2}}\right) \min\{\underline{\alpha}_C, \underline{\alpha}_I\} \\ \overline{\alpha} &= \left(1 + \sqrt{1 - \frac{1}{c_E^2}}\right) \max\{\overline{\alpha}_C, \overline{\alpha}_I\} \end{aligned}$$

The preconditioner can be used in conjugate gradient iteration.

If the components E, C_I and C_C are of assymtotical optimal, then the preconditioner C is it as well. Possible components are the following:

- In [11] the hierarchical basis extension is introduced. The boundary function is represented in the hierarchical basis, and the extension of u_C is the sum of the natural extension of the basis functions at the corresponding grid level. This extension is optimal up to a logarithmic factor. In [24] an extension based on multilevel splitting of the boundary function is given. This extension is of assymtotical optimal. A hierarchical basis extension with smoothing steps is introduced in [12].

- A symmetric multigrid operator ([2]) can be used to apply the preconditioning action $C_I^{-1} \times d$
- The restriction of the BPX preconditioner onto the boundary can be implemented directly. This is a choice of optimal order for the preconditioner C_C ([29]). A nearly optimal boundary operator based on hierarchical basis is given in [26].

All these components can be implemented in optimal time complexity, i.e. the operations E, E^t and C_I^{-1} take $O(N)$ time, while the application of C_C^{-1} takes $O(N_C)$ time. The precise analysis of the overall operator C is given in [10]. There is also shown, how a symmetric multiplicative Schwarz preconditioner fits into the framework of additive Schwarz preconditioners.

5 Inner Products for the Projection Method

Now we are well prepared to apply the Projection Method within different inner products generated by preconditioners C.

5.1 First Attempt: No preconditioner

The simplest possible case is to set $C = I$. With respect to this inner product the nodal basis is an orthogonal basis. The set K is an orthogonal product of intervals. Therefore the projection onto K splits into the one dimensional projections at the boundary nodes

$$\begin{aligned} (Pu)_C &= \max\{u_C, g\}, \\ (Pu)_I &= u_I. \end{aligned}$$

The disadvantage of this choice consists in an increasing condition number proportional to h^{-2}.

5.2 Second Attempt: Inner preconditioner

The projection involves only boundary nodes. The whole subspace V_I is kept unchanged. Therefore the componentwise projection at the boundary nodes is also equivalent to the projection in the inner product induced by the matrix

$$C = \begin{pmatrix} I_C & 0 \\ 0 & C_I \end{pmatrix}.$$

This means, that in every iteration step one has to apply a preconditioner for the Dirichlet problem. The condition number now grows proportional to h^{-1}.

5.3 Third Attempt: Optimal preconditioner

Iteration numbers independent of h can only be achieved, if one uses an optimal preconditioner. We wand to use the DD preconditioner of Section 4. But now it is more difficult to build the projection. We introduce the basis transformation

$$T = \begin{pmatrix} I_C & 0 \\ E_I & I_I \end{pmatrix} \tag{13}$$

and express the solution u within this basis:

$$u = T\hat{u}$$

Therefore $\hat{u}$ is the solution of the CMP

$$\inf_{\hat{v}\in\hat{K}} \hat{J}(\hat{v})$$

with

$$\hat{K} = T^{-1}K$$

and

$$\hat{J}(\hat{v}) = J(T\hat{v}) = \frac{1}{2}\hat{v}^t \underbrace{T^t A T}_{\hat{A}} \hat{v} - \underbrace{f^t T}_{\hat{f}^t} v.$$

If C is a preconditioner for A, then also $\hat{C} = T^t C T$ is one for $\hat{A}$ with the same bounds. If we use the DD-preconditioner (12) than $\hat{C}$ has the structure

$$\hat{C} = \begin{pmatrix} C_C & 0 \\ 0 & C_I \end{pmatrix},$$

i.e. is block diagonal with respect to inner and boundary blocks. Because v_I can be chosen arbitrary in a linear space, the set $\hat{K}$ reduces to K:

$$\hat{K} = T^{-1}K = \left\{ \begin{pmatrix} I_C & 0 \\ -E_I & I_I \end{pmatrix} \begin{pmatrix} v_C \\ v_I \end{pmatrix} : v_C \geq g \right\} = K$$

Now we apply the projection method to the transformed system:

$$\hat{u}^{k+1} = \hat{P}\left(\hat{u}^k + \tau\hat{C}^{-1}\left(\hat{f} - \hat{A}\hat{u}^k\right)\right)$$

Using the original quantities A, u and f this iteration can be rewritten as

$$u^{k+1} = T\hat{P}\underbrace{\left[T^{-1}u^k + \tau\hat{C}^{-1}T^t\left(Au^k - f\right)\right]}_{\tilde{u}^k}. \tag{14}$$

The projection involves only boundary components, which are decoupled from the inner components by the scalar product $\hat{C}$. Therefore it can be applied just on the boundary, the inner components stay unchanged. Appliing the projection means to solve the problem

$$\text{find}\, u_C \geq g_C : \ \|u_C - \tilde{u}_C^k\|_{C_C}^2 = \inf_{v_C \geq g_C} \|v_C - \tilde{u}_C^k\|_{C_C}^2. \tag{15}$$

This CMP can be solved by the projection method with Euclidean inner product and pointwise projection

$$u_C^{k,i+1} = P\left(u_C^{k,i} + \tau_2 C_C \left(\tilde{u}_C^k - u_C^{k,i}\right)\right). \tag{16}$$

The computational effort is now $n_o c_o + n_o n_i c_i$, with the iteration numbers n_o and n_i for outer and inner iterations and the costs per iteration step c_o and c_i. n_o is now optimal $O(1)$, n_i is $O(\kappa(C_C))$, which is $O(h^{-1})$ for two as well as three dimensional problems. The costs in the outer iteration are optimal $O(N)$. The costs in the inner iteration are at least $O(N_C)$, but they depend on the application of C_C. In DD, we need a fast implementation of the action $C_C^{-1} \times v$. That is ensured by the multilevel Schur complement preconditioner mentioned above, however the operation $C_C \times v$ cannot be performed explicitly.

5.4 Fourth Attempt: Dual problem for inner projection

By dualizing the boundary CMP (15) we get the operator C_C into the computable direction. In this chapter we skip the index C for boundary components as well as the outer iteration index k. By writing down the Kuhn Tucker conditions for the CMP we get the equivalent complementary problem ([7])

$$\begin{aligned} C_C u + p &= C_C \tilde{u} \\ u &\geq g \\ p &\leq 0 \\ p^t (u - g) &= 0. \end{aligned}$$

After defining $v := u - g$ and exchanging primal and dual variables one can write

$$\begin{aligned} C_C^{-1}(-p) + (-v) &= g - \tilde{u} \\ (-p) &\geq 0 \\ (-v) &\leq 0 \\ v^t p &= 0. \end{aligned}$$

Now this complementary problem can be rewritten as a CMP in the dual variables:

$$\min_{q \leq 0} \frac{1}{2} q^t C_C^{-1} q - q^t (g - \tilde{u}) \tag{17}$$

After calculating p, one gets u by $u = \tilde{u} - C_C^{-1} p$. The costs c_i of an inner step are now of optimal order $O(N_C)$.

5.5 Algorithm

Now we can state the whole algorithm developed so far:

Algorithm 3 (Inner-outer projection algorithm)

$u^0 = 0$
for $k = 0, 1, 2, ...$

$$
\begin{aligned}
d^k &= f - Au^k \\
w^k &= C^{-1} d^k \\
\tilde{u}^k &= u^k + \tau w^k \\
p^0 &= 0
\end{aligned}
$$

for $i = 0, 1, 2, ...$

$$p^{i+1} = P\left(p^i + \tau_2\left(g - \tilde{u}^k - C_C^{-1} p^i\right)\right)$$

$$u^{k+1} = \tilde{u}^{k+1} - E C_C^{-1} p^{\infty}$$

Unfortunately, the inner iteration has to be performed until a convergence criteria is fulfilled. To overcome this disadvantage we look for a better initial value p^0 and perform just a fixed number of iterations. The idea is similar to the step from Uzawa's algorithm to the Arrow Hurwitz one. To analyze that we need the Augmented Lagrangian formulation.

6 Augmented Lagrangian Formulation

Similar to Section 5.4, we dualize the whole CMP (3):

$$\min_{\substack{q \in \mathbf{R}^{N_C} \\ q \leq 0}} \frac{1}{2} q^t \mathcal{I}_C^t A^{-1} \mathcal{I}_C q - q^t \mathcal{I}_C^t \left(A^{-1} f + \mathcal{I}_C g\right). \tag{18}$$

$\mathcal{I}_C$ is the injection $\begin{pmatrix} I_C \\ 0 \end{pmatrix}$ of R^{N_C} into R^N. This functional and its derivative are not fast computable, because both need the inverse of A. By the Augmented Lagrangian technique, we can change it into something computable. We may add a convex functional in v and q, which minimum in v is 0 for every fixed q:

$$\inf_{q \leq 0, v} \frac{1}{2} \|Av + \mathcal{I}_C q - f\|^2_{\underline{\gamma}^{-1} C^{-1} - A^{-1}} + \frac{1}{2} q^t \mathcal{I}_C^t A^{-1} \mathcal{I}_C q - q^t \mathcal{I}_C^t \left(A^{-1} f + \mathcal{I}_C g\right) \tag{19}$$

C is a preconditioner to A scaled such that

$$C < A,$$

because then the matrix $C^{-1} - A^{-1}$ is positive definite.

6.1 The Bramble Pasciak Transformation

The discrete weak formulation of Dirichlet boundary conditions results in a saddle point problem of the form

$$\begin{pmatrix} A & \mathcal{I}_C \\ \mathcal{I}_C^t & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{20}$$

In [3] it is shown how to change (20) into a positive definite system. C is a preconditioner to A fulfilling

$$0 \le A - C \le \alpha A. \tag{21}$$

C may be the DD preconditioner with a proper scaling factor. By multiplication from the left with the matrices

$$\begin{pmatrix} A - C & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \mathcal{I}_C^t & -I \end{pmatrix} \begin{pmatrix} C^{-1} & 0 \\ 0 & I \end{pmatrix} \tag{22}$$

one gets

$$\underbrace{\begin{pmatrix} AC^{-1}A - A & (AC^{-1} - I)\,\mathcal{I}_C \\ \mathcal{I}_C^t\,(C^{-1}A - I) & \mathcal{I}_C^t C^{-1} \mathcal{I}_C \end{pmatrix}}_{\mathcal{A}} \begin{pmatrix} u \\ p \end{pmatrix} = \underbrace{\begin{pmatrix} (AC^{-1} - I)\,f \\ \mathcal{I}_C^t C^{-1} f - g \end{pmatrix}}_{\mathcal{F}} \tag{23}$$

In [3] the third multiplication is not applied explicitly, there it is contained in the inner product. In [3] the spectral equivalence to the block diagonal matrix

$$\begin{pmatrix} A - C & 0 \\ 0 & \mathcal{I}_C^t A^{-1} \mathcal{I}_C \end{pmatrix},$$

with constants

$$\begin{aligned} \underline{\alpha} &= \left(1 + \frac{\alpha}{2} + \sqrt{\alpha + \frac{\alpha^2}{4}}\right)^{-1} \\ \overline{\alpha} &= \frac{1 + \sqrt{\alpha}}{1 - \alpha}. \end{aligned}$$

is shown. If we replace the inverse of the Schur complement $\mathcal{I}_C^t A^{-1} \mathcal{I}_C$ by the preconditioner C_C^{-1} we get some additional factors into the spectral equivalence constants. The following theorem provides a direct estimation:

Theorem 3
Let C be a preconditioner to A fulfilling (21). Let C_C^{-1} denote the block $\mathcal{I}_C^t C^{-1} \mathcal{I}_C$. Then the block diagonal matrix

$$\mathcal{C} = \begin{pmatrix} \frac{1}{1-\alpha}(A - C) & 0 \\ 0 & C_C^{-1} \end{pmatrix} \tag{24}$$

is spectrally equivalent to the block matrix $\mathcal{A}$ of (23) with spectral equivalence constants

$$\underline{\alpha} = 1 - \sqrt{\alpha} \tag{25}$$
$$\overline{\alpha} = 1 + \sqrt{\alpha} \tag{26}$$

Proof: First we show $(A-C)C^{-1}A \leq \frac{1}{1-\alpha}(A-C)$. By (21) we have $A \leq \frac{1}{1-\alpha}C$. Because of symmetry, the spectral radius ρ is equal to the Reighlay quotient:

$$\sup_u \frac{u^t(A-C)C^{-1}Au}{u^t(A-C)u} = \rho(C^{-1}A) = \sup_u \frac{u^tAu}{u^tCu} \leq \frac{1}{1-\alpha}$$

We use the inequality $2(u,v) \leq \gamma\|u\|^2 + \gamma^{-1}\|u\|$, $\forall\gamma > 0$ applied to C^{-1}-energy norm:

$$\begin{aligned}
&\left(\mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} u \\ p \end{pmatrix}\right) = \\
&= ((A-C)C^{-1}Au,u) + 2(C^{-1}(A-C)u,\mathcal{I}_C p) + (C_C^{-1}p,p) \\
&\leq ((A-C)C^{-1}Au,u) + \gamma^{-1}((A-C)C^{-1}(A-C)u,u) + (1+\gamma)(C_C^{-1}p,p) \\
&= (1+\gamma^{-1})((A-C)C^{-1}Au,u) - \gamma^{-1}((A-C)u,u) + (1+\gamma)(C_C^{-1}p,p) \\
&\leq ((1-\alpha)^{-1}(1+\gamma^{-1}) - \gamma^{-1})((A-C)u,u) + (1+\gamma)(C_C^{-1}p,p)
\end{aligned}$$

Now we set $\gamma = \gamma^*$, the positiv solution of

$$(1-\alpha)^{-1}(1+\gamma^{-1}) - \gamma^{-1} = (1+\gamma)(1-\alpha)^{-1},$$

which is

$$\gamma^* = \sqrt{\alpha}.$$

Summing up we get the upper bound $\overline{\alpha}$

$$\begin{aligned}
\left(\mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} u \\ p \end{pmatrix}\right) &\leq (1+\gamma^*)(1-\alpha)^{-1}((A-C)u,u) + (1+\gamma^*)(C_C^{-1}p,p) \\
&= (1+\sqrt{\alpha})\left(\mathcal{C}\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} u \\ p \end{pmatrix}\right).
\end{aligned}$$

For the lower bound we use the same inequality with $0 < \gamma < 1$.

$$\begin{aligned}
&\left(\mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} u \\ p \end{pmatrix}\right) \geq \\
&\geq ((A-C)C^{-1}Au,u) - \gamma^{-1}((A-C)C^{-1}(A-C)u,u) + (1-\gamma)(C_C^{-1}p,p) \\
&= \underbrace{(1-\gamma^{-1})}_{<0}\underbrace{((A-C)C^{-1}Au,u)}_{\leq(1-\alpha)^{-1}((A-C)u,u)} + \gamma^{-1}((A-C)u,u) + (1-\gamma)(C_C^{-1}p,p) \\
&\geq ((1-\alpha)^{-1}(1-\gamma^{-1}) + \gamma^{-1})((A-C)u,u) + (1-\gamma)(C_C^{-1}p,p)
\end{aligned}$$

We set $\gamma = \gamma^* = \sqrt{\alpha}$, the positive solution of

$$(1-\alpha)^{-1}(1-\gamma^{-1}) + \gamma^{-1} = (1-\alpha)^{-1}(1-\gamma)$$

and get

$$\left(\mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} u \\ p \end{pmatrix}\right) \geq (1-\sqrt{\alpha})\left(\mathcal{C}\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} u \\ p \end{pmatrix}\right).$$

$\square$

We want to use a general preconditioner C fulfilling the spectral equivalence inequalities

$$\underline{\gamma} C \leq A \leq \overline{\gamma} C.$$

Then the rescaled preconditioner $C_1 = \underline{\gamma} C$ fulfills (21) with

$$\alpha = 1 - \frac{\underline{\gamma}}{\overline{\gamma}}.$$

The blockmatrix

$$\mathcal{A} = \begin{pmatrix} \underline{\gamma}^{-1} A C^{-1} A - A & \left(\underline{\gamma}^{-1} A C^{-1} - I\right) \mathcal{I}_C \\ \mathcal{I}_C^t \left(\underline{\gamma}^{-1} C^{-1} A - I\right) & \underline{\gamma}^{-1} \mathcal{I}_C^t C^{-1} \mathcal{I}_C \end{pmatrix} \tag{27}$$

is spectrally equivalent to the preconditioner

$$\mathcal{C} = \begin{pmatrix} \frac{\overline{\gamma}}{\underline{\gamma}}(A - \underline{\gamma} C) & 0 \\ 0 & \underline{\gamma}^{-1} C_C^{-1} \end{pmatrix} \tag{28}$$

with constants

$$\underline{\alpha} = 1 - \sqrt{1 - \frac{\underline{\gamma}}{\overline{\gamma}}} \tag{29}$$

$$\overline{\alpha} = 1 + \sqrt{1 - \frac{\underline{\gamma}}{\overline{\gamma}}}. \tag{30}$$

Mention, that $\overline{\alpha} \leq 2$ and $\underline{\alpha} \geq \frac{\underline{\gamma}}{2\overline{\gamma}}$, and therefore the relative condition number of the block systems is less then four times as worse as the relative condition number of A and C. The upper bound for $\overline{\alpha} \leq 2$ also provides us with a simple choice of the needed relaxation parameter.

6.2 The Approximative Augmented Projection Algorithm

The matrix of the quadratic term of the functional (19) is exactly the matrix $\mathcal{A}$ in (27). The linear term is the vector

$$\mathcal{F} = \begin{pmatrix} \left(\underline{\gamma}^{-1} A C^{-1} - I\right) f \\ \underline{\gamma}^{-1} \mathcal{I}_C^t C^{-1} f - g \end{pmatrix} \tag{31}$$

One can try to use the projection algorithm with inner product (28). We apply one preconditioned Richardson step for the linear system $\mathcal{A} \begin{pmatrix} u \\ p \end{pmatrix} = \mathcal{F}$,

$$\begin{pmatrix} \tilde{u}^k \\ \tilde{p}^k \end{pmatrix} = \begin{pmatrix} u^k \\ p^k \end{pmatrix} + \tau \mathcal{C}^{-1} \left[\mathcal{F} - \mathcal{A} \begin{pmatrix} u^k \\ p^k \end{pmatrix} \right].$$

With the notations

$$
\begin{aligned}
d_u^k &= f - Au^k - \mathcal{I}_C p^k && (32)\\
w_u^k &= C^{-1} d_u^k && (33)\\
w_p^k &= \mathcal{I}_C w_u^k - \underline{\gamma}(g - \mathcal{I}_C^t u^k), && (34)
\end{aligned}
$$

the Richardson step simplifies to

$$\begin{pmatrix} \tilde{u}^k \\ \tilde{p}^k \end{pmatrix} = \begin{pmatrix} u^k \\ p^k \end{pmatrix} + \tau \begin{pmatrix} \overline{\gamma}^{-1} w_u^k \\ C_C w_p^k \end{pmatrix}.$$

The restrictions involve only the dual component p and the inner product matrix $\mathcal{C}$ decouples the primal and the dual components, therefore the projection needs only the dual component:

$$\text{find } p_{ex}^{k+1} \leq 0 : \ \|p_{ex}^{k+1} - \tilde{p}^k\|_{C_C^{-1}}^2 = \inf_{q \leq 0} \|q - \tilde{p}^k\|_{C_C^{-1}}^2$$

By Theorem 2 it is enough to have an approximative projection fulfilling

$$\|p^{k+1} - \tilde{p}^k\|_{C_C^{-1}}^2 \leq \rho_2 \|p^k - \tilde{p}^k\|_{C_C^{-1}}^2 + (1 - \rho_2)\|p_{ex}^{k+1} - \tilde{p}^k\|_{C_C^{-1}}^2$$

The approximative projection is implemented by appliing n steps of the projection algorithm with Euklidean inner produkt

$$
\begin{aligned}
q^0 &= p^k \\
q^{i+1} &= P\left(q^i + \tau_2 C_C^{-1}(\tilde{p}^k - q^i)\right) \\
p^{k+1} &= q^n.
\end{aligned}
$$

We sustitute for $\tilde{p}^k$, and we see, that we never have to apply C_C directly:

$$q^{i+1} = P\left(q^i + \tau_2\left(\tau w_p^k + C_C^{-1}(p^k - q^i)\right)\right).$$

The spectrum of the Schur complement preconditioning operation C_C^{-1} is included in $[\underline{\lambda}, \overline{\lambda} h^{-1}]$ with h-independent constants $\underline{\lambda}$ and $\overline{\lambda}$ for the two dimensional case. In the three dimensional case there is an additional factor h^{-1} in both bounds. The relaxation parameter τ_2 has to be chosen such that $\tau_2 \overline{\lambda} \leq h$. By Theorem 1 q^i converges in energy-norm, i.e. C_C^{-1} norm, with rate $\rho_i = 1 - \frac{\tau_2 \underline{\lambda}}{2}$. We get the error estimation after n steps as

$$\|p^{k+1} - \tilde{p}^k\|_{C_C^{-1}}^2 \leq \rho_i^n \|p^k - \tilde{p}^k\|_{C_C^{-1}}^2 + (1 - \rho_i^n)\|p_{ex}^{k+1} - \tilde{p}^k\|_{C_C^{-1}}^2.$$

We set $\rho_2 := \rho_i^n$ and demand $n \geq c\frac{2}{\tau_2 \underline{\lambda}}$, for some positive constant c. This expression grows proportional to h^{-1}. So we get

$$\rho_2 \leq \left(1 - \frac{\tau_2 \underline{\lambda}}{2}\right)^{c\frac{2}{\tau_2 \underline{\lambda}}} \leq e^{-c} \tag{35}$$

By Theorem 2 we have a h independent convergence rate. The final algorithm looks like:

Algorithm 4 (Approximative Augmented Projection Method)

$u^0 = g, p^0 = 0$
for $k = 0, 1, 2, ...$

$$
\begin{aligned}
&d_u^k = f - Au^k - \mathcal{I}_C p^k \\
&w_u^k = C^{-1} d_u^k \\
&w_p^k = \mathcal{I}_C^t w_u^k - \underline{\gamma}(g_c - \mathcal{I}_C^t u_k) \\
&\textit{for } i = 0 \textit{ to } n-1 \\
&\qquad p^{k+\frac{i+1}{n}} = P\left(p^{k+\frac{i}{n}} + \tau_2 \left(C_C^{-1} p^k + \tau w_p^k - C_C^{-1} p^{k+\frac{i}{n}}\right)\right) \\
&u^{k+1} = u^k + \tau \overline{\gamma}^{-1} w_u^k
\end{aligned}
$$

To apply the algorithm we have to choose some parameters, namely

- find $\underline{\gamma}$ and $\overline{\gamma}$ as close as possible fulfilling the spectral equivalence inequalities
$$\underline{\gamma} C \leq A \leq \overline{\gamma},$$
calculate τ via
$$\tau = \left(1 + \sqrt{1 - \frac{\underline{\gamma}}{\overline{\gamma}}}\right)^{-1},$$
- find τ_2 as large as possible such that
$$\tau_2 C_C^{-1} \leq I,$$
- and set n proportional to the condition number od C_C, e.g.
$$n = 2^{Multigridlevels}$$

The iteration error can be estimated by Corollary 1, which leads after some simple calculations to

$$
\begin{aligned}
\left\| \begin{pmatrix} u - u^{k+1} \\ p - p^{k+1} \end{pmatrix} \right\|_{\mathcal{A}}^2 &\leq \frac{2\rho}{1-\rho} \left(\begin{pmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{pmatrix}, \mathcal{F} - \mathcal{A} \begin{pmatrix} u^k \\ p^k \end{pmatrix} \right)_{\mathbf{R}^N} \\
&\leq \frac{2\rho}{1-\rho} \left[\tau \frac{\overline{\gamma} - \underline{\gamma}}{\overline{\gamma}\underline{\gamma}} (w_u^k, d_u^k)_{\mathbf{R}^N} + (p^{k+1} - p^k, w_p^k)_{\mathbf{R}^{N_C}} \right] \\
&\leq c \left[(w_u^k, d_u^k)_{\mathbf{R}^N} + (p^{k+1} - p^k, w_p^k)_{\mathbf{R}^{N_C}} \right], \qquad (36)
\end{aligned}
$$

with a constant c only depending on ρ and the spectral equivalence constants of the preconditioner C.

The complexity of the final algorithm is $O(n_o c_o + n_o n_i c_i)$. The costs per iteration step c_o and c_i are of optimal order, this means $O(N)$ and $O(N_C)$, respectively. The number of inner iterations n_i is $O(h^{-1})$ for the two dimensional as well as three dimensional case. The number of outer iterations n_o is fixed for a given error ε. This gives the complexity $O((N + h^{-1} N_C)\, ln\, \varepsilon)$. On a regular refined mesh we have $N_C = O(hN)$ and therefore the complexity reduces to the optimal term $O(N\, ln\, \varepsilon)$.

6.3 The multiplicative version

Algorithm 4 is an approximative additive Schwarz method with respect to the splitting

$$V_1 = \emptyset \times \mathbf{R}^{N_C}, \qquad V_2 = \mathbf{R}^N \times \emptyset.$$

According to [22] we can use a multiplicative Schwarz method with respect to this splitting. Without proving, we claim, that the multiplicative method converges at least as fast as the additive method.

With the approximative Projection $\tilde{P}$ the multiplicative method is given by

$$\begin{aligned} \tilde{p}^k &= p^k + (0\ I)\,\mathcal{C}^{-1}\left[\mathcal{F} - \mathcal{A}\begin{pmatrix} u^k \\ p^k \end{pmatrix}\right] \\ p^{k+1} &= \tilde{P}(\tilde{p}^k) \\ u^{k+1} &= u^k + (I\ 0)\,\mathcal{C}^{-1}\left[\mathcal{F} - \mathcal{A}\begin{pmatrix} u^k \\ p^{k+1} \end{pmatrix}\right]. \end{aligned}$$

Because $\mathcal{A} \leq \mathcal{C}$ on both subspaces V_1 and V_2 the relaxation parameters could be set to 1. We see that the update for p^{k+1} is equal to the update given by the additive version, the difference in u^{k+1} due to the different versions of p can be expressed by

$$\begin{aligned} u^{k+1} - u^{k+1}_{ASM} &= (I\ 0)\,\mathcal{C}^{-1}\mathcal{A}\begin{pmatrix} 0 \\ p^k - p^{k+1} \end{pmatrix} \\ &= \frac{1}{\overline{\gamma}} C^{-1}\mathcal{I}_C(p^{k+1} - p^k) \\ &= \frac{1}{\overline{\gamma}} E C_C^{-1}(p^{k+1} - p^k) \end{aligned}$$

Now we can state the multiplicative algorithm, which is very similar to the Inner-outer projection algorithm (Algorithm 3), but the inner iteration is now finite:

Algorithm 5 (Approximative Multiplicative Augmented Projection Method)

$u^0 = g, p^0 = 0$
for $k = 0, 1, 2, \ldots$

$d_u^k = f - Au^k - \mathcal{I}_C p^k$
$w_u^k = C^{-1} d_u^k$
$w_p^k = \mathcal{I}_C^t w_u^k - \underline{\gamma}(g_c - \mathcal{I}_C^t u_k)$
for $i = 0$ *to* $n - 1$

$$p^{k+\frac{i+1}{n}} = P\left(p^{k+\frac{i}{n}} + \tau_2\left(C_C^{-1}p^k + w_p^k - C_C^{-1}p^{k+\frac{i}{n}}\right)\right)$$

$$u^{k+1} = u^k + \overline{\gamma}^{-1}\left(w_u^k + EC_C^{-1}(p^k - p^{k+1})\right)$$

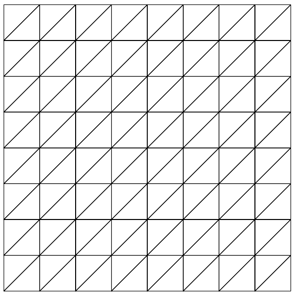

Figure 2: Level 4 mesh

7 Numerical Results

To give a numerical verification of theory the algorithms above have been implemented to solve the equations

$$\begin{aligned} -\Delta u &= -1 \quad on \quad (0,1)^2 \\ u &= 0 \quad on \quad [0,1]\times\{1\} \\ \frac{\partial u}{\partial n} &= 0 \quad on \quad \{0,1\}\times[0,1] \end{aligned}$$

$$\left.\begin{aligned} u &\geq g \\ \tfrac{\partial u}{\partial n} &\geq 0 \\ \tfrac{\partial u}{\partial n}(u-g) &= 0 \end{aligned}\right\} \quad on \quad [0,1]\times\{0\}$$

with

$$g(x,y) := \begin{cases} 1 & \text{for } x\in[0.25,0.75] \\ 0 & \text{else} \end{cases}.$$

The numerical examples have been carried out within the C++ finite element code FE++ on a Sun Ultra 1/170 MHz. We used a sequence of hierarchical refined triangular meshes for FE discretization. The level 4 mesh is given in Figure 2.

For the preconditioner C, a symmetric, multiplicative DD preconditioner with a V-2-2 multigrid cycle, a hierarchical extension with 2 smoothing step and a boundary BPX preconditioner for the Schur complement has been used. The spectral equivalence constants $\underline{\gamma}$ and $\overline{\gamma}$ calculated by the Lanczos method are shown in Table 1. The bounds $\underline{\gamma}_C$ and $\overline{\gamma}_C$ for the eigenvalues of the matrix C_C^{-1} are shown in Table 2. Based on this calculations we have chosen a priori the values

$$\underline{\gamma} = 0.65, \qquad \overline{\gamma} = 2.2, \qquad \tau_2 = 2^{-level}, \qquad n = 2^{level},$$

and applied Algorithm 4 and Algorithm 5 to reduce the squared error according to the estimator (36) by a factor of 10^{-8}. The iteration numbers and the CPU time are shown in

level	N	$\underline{\gamma}$	$\overline{\gamma}$
2	9	0.969	1.458
3	25	0.864	1.732
4	81	0.814	1.863
5	289	0.747	1.956
6	1089	0.693	2.018
7	4225	0.666	2.065
8	16641	0.653	2.101

Table 1: preconditioner values

level	N_C	$\underline{\gamma}_C$	$\overline{\gamma}_C$
2	3	0.649	3.851
3	5	0.552	7.031
4	9	0.500	13.084
5	17	0.5	25.077
6	33	0.5	49.055
7	65	0.5	97.035
8	129	0.5	193.02

Table 2: Schur complement preconditioner values

level	N	N_C	its	time[sec]
2	9	3	73	0.08
3	25	5	91	0.22
4	81	9	84	0.47
5	289	17	80	1.38
6	1089	33	76	4.70
7	4225	65	75	18.5
8	16641	129	74	81.7

Table 3: Algorithm 4 (additive)

level	N	N_C	its	time[sec]
2	9	3	26	0.05
3	25	5	36	0.13
4	81	9	31	0.27
5	289	17	29	0.86
6	1089	33	26	2.91
7	4225	65	23	10.5
8	16641	129	23	47.7

Table 4: Algorithm 5 (multiplicative)

Table 3 and Table 4, respectively. The iteration numbers are bounded by a constant, and the CPU time grows proportional to the unknowns. The solution on the level 5 mesh is drawn in Figure 3.

If the FE solution u_h fulfills the restrictions of the continuous problem, then the discretization error in energy norm may be estimated by $\|u - u_h\|^2 \leq \frac{1}{2}\left(J(u_h) - J(u)\right)$. For our example, we have $K_h \subset K$ up from level 3. Table 5 contains the minima of the Ritz functional and the difference to the estimated limit $\widetilde{J(u)}$. The estimated error shows the expected convergence rate $\|u - u_h\|^2 \leq ch$ for a solution u in $H^{1.5}(\Omega)$.

Next we implemented a nested iterative scheme. On level 3 the problem was solved nearly exactly. On the refined levels we used the prolongated approximative solution of the next coarser level as initial guess. On each level we did just a fixed number of three steps of the multiplicative method. The prolongation for the primal component u was done naturally, for the dual component p in $H^{-1/2}(\Gamma_C)$ the $L_2(\Gamma_C)$ representant in V_{hC} was calculated and extended naturally. By this nested method we got nearly the same estimated error within the accumulated solver time and total time shown in Table 6.

Because of the singularities in the solution an adaptive refined mesh has large advantages with respect to memory requirements and CPU time. Therefore an error estimator

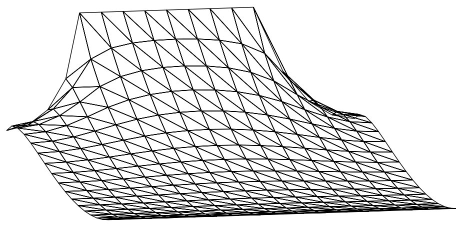

Figure 3: Solution on level 5 mesh

level	N	N_C	$J(u_h)$	$J(u_h) - \widetilde{J(u)}$
3	25	5	0.9180	0.0714
4	81	9	0.8851	0.0385
5	289	17	0.8663	0.0197
6	1089	33	0.8566	0.0100
7	4225	65	0.8516	0.0050
8	16641	129	0.8491	0.0025
∞			~ 0.8466	

Table 5: Minimum of Ritz functional

level	N	N_C	$J(u_h)$	$J(u_h) - \widetilde{J(u)}$	solver time[sec]	total time[s]
3	25	5	0.9180	0.0714	0.11	0.12
4	81	9	0.8909	0.0443	0.13	0.17
5	289	17	0.8706	0.0240	0.23	0.41
6	1089	33	0.8588	0.0122	0.58	1.25
7	4225	65	0.8527	0.0061	2.01	4.56
8	16641	129	0.8496	0.0030	8.42	18.5
9	66049	257	0.8481	0.0015	35.2	75.3
10	263169	513	0.8473	0.0007	144.2	305
∞			~ 0.8466			

Table 6: Nested iteration

level	N	N_C	$J(u_h)$	$J(u_h) - \widetilde{J(u)}$	solve [sec]	estimate[sec]	total[sec]
3	25	5	0.9180	0.0714	0.11	0.02	0.15
4	75	9	0.8890	0.0424	0.14	0.06	0.27
5	188	17	0.8699	0.0233	0.22	0.17	0.60
6	494	33	0.8588	0.0122	0.41	0.48	1.49
7	1165	61	0.8529	0.0065	0.90	1.22	3.66
8	2588	109	0.8498	0.0032	2.13	2.88	8.61
9	5087	187	0.8482	0.0016	5.10	6.29	19.1
10	8516	297	0.84745	0.00088	11.9	12.5	39.6
11	14440	439	0.84705	0.00048	27.3	23.2	78.5
12	22909	605	0.84683	0.00026	60.4	40.8	149.6
13	39866	881	0.84671	0.00014	140.8	73.1	297.2
14	69198	1187	0.84664	0.00007	330.7	130.9	606.3
∞			~ 0.84657				

Table 7: adaptive refinement

based on local problems (see [30]) was implemented. The refinement was done using hanging nodes. As preconditioner we used $PC^{-1}P^t$, where C is the DD-preconditioner from above and P is the natural projection into the conforming subspace. In Table 7 you can see the growing of the total unknows of about 2^{levels}, the growing of the boundary unknows of about $2^{levels/2}$, the estimated discretization error and the accumulated times for the solver, the time for the error estimation and the total time. The discretization error is of the same order as an the full grid. Although the complexity is not $O(N)$ anymore, the overall CPU time is much less. The adaptive refined mesh at level 8 you can see in Figure 4.

As promised in the introduction, we have also numerical results for the plane elasticity problem. For the first example we set

$$\Omega = [0,1] \times [0.05, 1.05], \qquad V = \left\{ v \in \left[H^1(\Omega)\right]^2 : v = 0 \text{ on } \{0\} \times [0.05, 1.05] \right\},$$

for the second one we use the circle

$$\Omega = C((0.5, 0.5), \sqrt{0.5}), \qquad V = \left[H^1(\Omega)\right]^2.$$

For both examples we want to find the displacement field minimizing the functional

$$J(v) = \frac{1}{2} a(v,v) - \langle f, v \rangle$$

with

$$a(u,v) = \int_\Omega 2\mu\, \varepsilon(u) : \varepsilon(v) + \lambda \operatorname{div} u \operatorname{div} v \, \mathrm{dx}, \qquad \langle f, v \rangle = -g \int_\Omega v_2 \, \mathrm{dx}$$

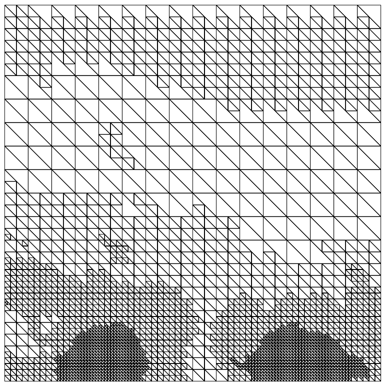

Figure 4: Adaptive mesh at level 8

with the linearized strain tensor $\varepsilon_{ij}(u) = \frac{1}{2}(u_{i,j} + u_{j,i})$ and the parameters $\lambda = \mu = 1$ and $g = 0.2$, over the set

$$K = \{v \in V : x_2 + v_2(x) \geq 0 \; \forall x \in \partial\Omega\}\,.$$

The bilinearform a is continuous and thanks to Korn's inequalities also elliptic modulo the rigid body motions. Therefore a block diagonal preconditioner consisting of two copies of preconditioners for the Laplace equation can be used. We used two times the DD - preconditioner.

The injection of the boundary space into the global space is now replaced by the injection in normal direction. In the contact zone the normal direction is $(0, -1)$.

We applied the nested multiplicative approximative algorithm with the parameters $\underline{\lambda} = 0.4$, $\overline{\lambda} = 4$ and 12 iterations per level. For the first example we got the error reduction shown in Table 8. We see, the squared error $\|u - u_h\|^2 \leq ch^2$, which is the optimal convergence rate for smooth solutions. Therefore a uniform refined mesh meets all requirements. The deformed body is drawn in Figure 5. For the dual component p the $L_2(\Gamma)$ representant is calculated. It is the normal traction, which is also drawn in Figure 5. The result for the second elasticity example is given in Figure 6.

8 Concluding Remarks

We have developed and analysed an algorithm to solve Signorini's problem for the Poisson equation. It was generalized to solve the elastic contact problem for plane obstacles. We have seen, that the algorithm is of optimal complexity for uniform refined meshes. Both algorithms can be implemented very fast, if an according domain decomposition preconditioner is at hand. In the future, the algorithm will be extended to body - body contacts and to three dimensional problems.

I would like to thank Prof. Ulrich Langer for his inspiration to this work and for his clear presentations of domain decomposition techniques.

level	N	N_C	$J(u_h)$	$J(u_h) - \widetilde{J(u)}$	total time[sec]
2	9	3	-0.0059705	1.23E-3	0.27
3	25	5	-0.0066809	5.25E-4	0.38
4	81	9	-0.0070115	1.94E-4	0.64
5	289	17	-0.0071444	6.10E-5	1.56
6	1089	33	-0.0071879	1.75E-5	5.13
7	4225	65	-0.0072006	4.8E-6	20.1
8	16641	129	-0.0072041	1.3E-6	87.2
9	66049	257	-0.0072051	3.0E-7	365.8
∞			$\sim$ -0.0072054		

Table 8: Nested iteration for elasticity problem

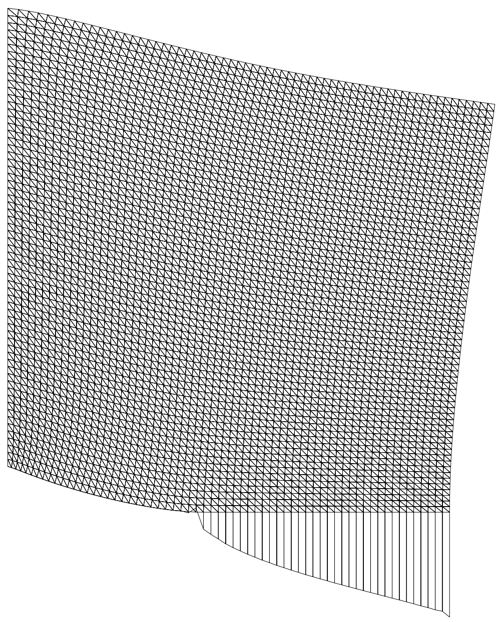

Figure 5: Elasticity problem 1

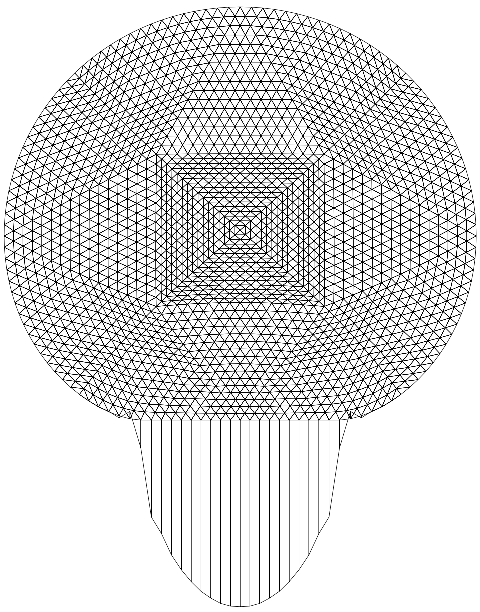

Figure 6: Elasticity problem 2

References

[1] M. Benassi and R. E. White. Parallel numerical solution of variational inequalities. *SIAM J. Numer. Anal.*, 31(3):813–830, 1994.

[2] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the v-cycle. *SIAM J. Numer. Anal.*, 20(5):967–975, 1983.

[3] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation*, 50(181):1–17, 1988.

[4] Franco Brezzi, William W. Hager, and P. A. Raviart. Error estimates for the finite elment solution of variational inequalities. *Numer. Math.*, 28:431–443, 1977.

[5] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1978.

[6] G. Fichera. Boundary value problems of elasticity with unilateral constraints. In *Handbuch der Physik - Encyclopedia of Physics*, volume Band VI a/2 Festkörpermechanik II. Springer, Berlin, 1972.

[7] Roland Glowinski, Jacques-Louis Lions, and Raymond Trémolières. *Numerical analysis of variational inequalities.* North-Holland Publishing Company, Amsterdam - New York - Oxford, 1981.

[8] Ch. Großmann and H. G. Roos. *Numerik partieller Differentialgleichungen.* Teubner, Stuttgart, 1994.

[9] Joachim Gwinner. Finite-element convergence for contact problems in plane linear elastostatics. *Quart. of appl. Math.*, L(1):11–25, March 1992.

[10] G. Haase, U. Langer, and A. Meyer. The approximate dirichlet domain decomposition method. part I: An algebraic approach. partII: Applications to 2nd-order elliptic boundary value problems. *Computing*, 47:137–151, 153–167, 1991.

[11] G. Haase, U. Langer, A. Meyer, and S. V. Nepomnyaschikh. Hierarchical extension operators and local multigrid methods in domain decomposition preconditioners. *East-West J. Numer. Math.*, 2(3):173–193, 1994.

[12] Gundolf Haase. An improvement of the hierarchical extension technique by smoothing and its use in a domain decomposition preconditioner. *Appl. Num. Math.*, (to appear).

[13] W. Hackbusch and H.D. Mittelmann. On multi-grid methods for variational inequalities. *Numer. Math.*, 42:65–76, 1983.

[14] Houde Han. A direct boundary element method for signorini problems. *Math. Comp.*, 55(191):115–128, 1990.

[15] J. Haslinger, I. Hlaváček, and Nečas. Numerical methods for unilateral problems in solid mechanics. In *Handbook of Numerical Analysis*, volume 2. North Holland, 1996.

[16] K.-H. Hoffmann and J. Zou. Parallel algorithms of schwarz variant for variational inequalities. *Numer. Funct. Anal. and Optimiz.*, 13:449–462, 1992.

[17] R. H. W. Hoppe and R. Kornhuber. Adaptive multilevel methods for obstacle problems. *SIAM J. Numer. Anal.*, 31(2):301–323, 1994.

[18] Ronald H. W. Hoppe. Multigrid algorithms for variational inequalities. *SIAM J. Numer. Anal.*, 24(5):1046–1065, 1987.

[19] N. Kikuchi and J. T. Oden. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods.* SIAM, Philadelphia, 1988.

[20] D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and theri applications.* Academic Press, New York, 1980.

[21] U. Langer and W. Queck. Preconditioned uzawa-type iterative methods for solving mixed finite elemente equations. Wissenschaftliche Schriftenreihe 3/1987, Technische Universität Karl-Marx-Stadt (now Chemnitz) , 1987.

[22] P.L. Lions. On the schwarz alternating method I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations.* SIAM, 1988.

[23] Jan Mandel. A multilevel iterative method for symmetric, positive definite linear complementarity problems. *Appl. Math. Optim.*, 11:77 – 95, 1984.

[24] S. V. Nepomnyaschikh. Optimal multilevel extension operators. *Preprint SPC 95_3, Technische Universität Chemnitz-Zwickau, Fakultät für Mathematik*, 1995.

[25] A. Signorini. Sopra alcune questioni di elastostatica. *Atti Soc. Ial. Progr. Sci.*, 1933.

[26] B. F. Smith and O. B. Widlund. A domain decomposition algorithm using a hierarchical basis. *SIAM J. Sci. Stat. Comput.*, 11(6):1212–1220, 1990.

[27] W. Spann. On the boundary element method for the signorini problem of the laplacian. *Numer. Math.*, 65:337 – 356, 1993.

[28] Lü Tao, Chin-Bo Liem, and Tsi-Min Shih. Parallel algorithms for variational inequalities based on domain decomposition. *System Science and Mathematical Sciences*, 4(4):341–348, 1991.

[29] C. H. Tong, T. F. Chan, and C. C. Jay Kuo. A domain decomposition preconditioner based on a change to a multilevel nodal basis. *SIAM J. Sci. Stat. Comp.*, 12(6):1486–1495, 1991.

[30] R. Verfuerth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Wiley-Teubner, 1996.