42

Multilevel Extension Techniques in Domain Decomposition Preconditioners

Gundolf Haase

One component in Additive Schwarz Method (ASM) Domain Decomposition (DD) preconditioners [BPS89, SBG96] using inexact subdomain solvers [Boe89, HLM91] consists of an operator extending the boundary data into the interior of each subdomain, i.e., a homogeneous extension with respect to the differential operator given in that subdomain. This paper is concerned with the construction of cheap extension operators using multilevel nodal bases [Yse86, Xu89, BPX90, Osw94] from an implementation viewpoint. Additional smoothing sweeps in the extension operators further improve the condition number of the preconditioned system. The paper summarizes and improves results given in [HLMN94, Nep95, Haa97].

1 The ASM-DD-Preconditioner

Consider the symmetric, $\mathbb{V}_0 = \overset{\circ}{\mathbb{H}}{}^1(\Omega)$–elliptic and $\mathbb{V}_0$–bounded variational problem

$$\text{find } u \in \mathbb{V}_0 \ : \int_\Omega \lambda(x)\, \nabla^T u(x)\, \nabla v(x)\, dx \ = \int_\Omega f(x)\, v(x)\, dx \qquad \forall v \in \mathbb{V}_0\,, \tag{1.1}$$

arising from the weak formulation of a scalar second–order, symmetric and uniformly bounded elliptic boundary value problem given in a plane bounded domain $\Omega \subset \mathbb{R}^2$ with a piecewise smooth boundary $\Gamma = \partial\Omega$. The material coefficients $\lambda(x) \geq \lambda_0 > 0 \quad \forall x \in \overline{\Omega}$ have to be restricted for certain extension techniques.

The domain Ω will be decomposed into p non-overlapping subdomains Ω_i $(i = 1, \dots, p)$ such that $\overline{\Omega} = \bigcup_{i=1}^p \overline{\Omega}_i$. The discretization process using Courant's linear triangular finite elements in each subdomain Ω_i results in a conforming triangulation of Ω . In the following, the indices "C" and "I" correspond to nodes belonging to the coupling boundaries $\Gamma_C = \bigcup_{i=1}^p \partial\Omega_i \setminus \Gamma_D$ and to the interior $\Omega_I = \bigcup_{i=1}^p \Omega_i$ of the subdomains, respectively. Γ_D is that part of $\partial\Omega$ where Dirichlet–type boundary conditions are given whereas Neumann boundary conditions will be handled as coupling boundaries.

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

Define the usual finite elements (FE) nodal basis

$$\Phi = [\Phi_C, \Phi_I] = \left[\psi_1, \cdots, \psi_{N_C}, \psi_{N_C+1}, \cdots, \psi_{N_C+N_{I,1}}, \cdots, \psi_{N=N_C+N_I}\right] , \tag{1.2}$$

where the first N_C basis functions belong to Γ_C, the next $N_{I,1}$ to Ω_1, the next $N_{I,2}$ to Ω_2 and so on such that $N_I = \sum_{i=1}^{p} N_{I,i}$. Then the FE isomorphism leads to the symmetric and positive definite system of equations

$$K\,\underline{u} \;:=\; \begin{pmatrix} K_C & K_{CI} \\ K_{IC} & K_I \end{pmatrix} \begin{pmatrix} \underline{u}_C \\ \underline{u}_I \end{pmatrix} \;=\; \begin{pmatrix} \underline{f}_C \\ \underline{f}_I \end{pmatrix} \;=:\; \underline{f} \ , \tag{1.3}$$

where $K_I = \operatorname{blockdiag}\left(K_{I,i}\right)_{i=1,\dots,p}$ is block diagonal and symmetric, positive definite.

Solving system (1.3) with some parallelized iterative method, e.g., CG-method, we use the ASM–DD preconditioner

$$C = \begin{pmatrix} I_C & -B_{IC}^{-T} \\ O & I_I \end{pmatrix} \begin{pmatrix} C_C & O \\ O & C_I \end{pmatrix} \begin{pmatrix} I_C & O \\ -B_{IC} & I_I \end{pmatrix} . \tag{1.4}$$

This preconditioner contains the three components C_C, $C_I = \operatorname{diag}\left(C_{I,i}\right)_{i=1,\dots,p}$ and $B_{IC} = \operatorname{blockmatrix}\left(B_{IC,i}\right)_{i=1,\dots,p}$, which can freely be chosen in order to adapt the preconditioner to the particulars of the problem under consideration. For the choice $B_{IC,i} = -B_{I,i}K_{IC,i}$ see [HLM91]. As preconditioner C_C for the Schur complement $S_C = K_C - K_{CI}K_I^{-1}K_{IC}$ the BPS [BPS89] and the S(chur)-BPX [TCK92] are used.

The preconditioning step $\underline{w} = C^{-1}\underline{r}$ can be rewritten in the form

Algorithm 1 : The ASM-DD Preconditioner [HLM91]			
$\underline{\mathbf{w}}_C$	$=$	$C_C^{-1} \sum\limits_{i=1}^{p} A_{C,i}^T \left(\underline{r}_{C,i} + B_{IC,i}^T \underline{r}_{I,i}\right)$	
$\underline{\mathbf{w}}_{I,i}$	$=$	$C_{I,i}^{-1}\underline{r}_{I,i} + B_{IC,i}\underline{\mathbf{w}}_{C,i}$	$;\ i = 1, 2, \dots, p$

where $A_i = \left(\begin{smallmatrix} A_{C,i} & A_{CI,i} \\ A_{IC,i} & A_{I,i} \end{smallmatrix}\right)$ denotes the subdomain connectivity matrix which is used for a convenient notation only. The subdomain FE assembly process which is connected with nearest neighbour communication stands behind this notation. Other DD-preconditioners and modifications of Algorithm 1 can be found in [HL92].

Assume positive, h-independent spectral equivalence constants $\underline{\gamma}_C, \overline{\gamma}_C, \underline{\gamma}_I, \overline{\gamma}_I$ fulfilling the spectral equivalence inequalities

$$\underline{\gamma}_C\, C_C \;\leq\; S_C \;\leq\; \overline{\gamma}_C\, C_C \qquad \text{and} \qquad \underline{\gamma}_I\, C_I \;\leq\; K_I \;\leq\; \overline{\gamma}_I\, C_I \ . \tag{1.5}$$

If we have a constant c_E so that

$$\left\| \begin{pmatrix} \underline{v}_C \\ B_{IC}\underline{v}_C \end{pmatrix} \right\|_K \;\leq\; c_E \;\|\, \underline{v}_C \,\|_{S_C} \qquad \forall \underline{v}_C \in \mathbb{R}^{N_c} \tag{1.6}$$

holds then the upper and lower bounds of the condition number $\kappa(C^{-1}K)$ [HLM91, Che93] can be estimated as

$$\mathcal{O}(c_E^2) \;\leq\; \kappa(C^{-1}K) \;\leq\; \mathcal{O}(c_E^4) \ . \tag{1.7}$$

In the remaining chapter we construct extension techniques defining $B_{IC,i}$ which are cheap to implement and result in a constant c_E independent of or slightly dependent on the discretization parameter h and the number of levels ℓ.

In the following the subscript i denoting the subdomain number will be omitted.

2 Multilevel Extension Operators

Let space $\mathbb{W}_k$ consist of real-valued functions which are continuous on Ω and linear on the triangles in Ω_k^h. The space $\mathbb{V}_k$ is the space of traces on Γ of functions from $\mathbb{W}_k$.

The multilevel extension of a FE function $\varphi^h \in \mathbb{V}_\ell$ on the boundary $\partial\Omega$ into a function $u^h \in \mathbb{W}_\ell$ in the interior of the domain Ω via the extension operator $\mathcal{E}_{\overline{\Omega}\Gamma}$ consists of three steps:

1. Choose the projection $\mathcal{Q}_k$ from $\mathbb{V}_\ell$ into $\mathbb{V}_k$ $(k = 0, \ldots, \ell)$.
2. Split the function φ^h into a multilevel nodal basis according to the projections $\mathcal{Q}_k$

$$\begin{aligned} \chi_0^h &= \mathcal{Q}_0 \varphi^h, && (2.8a) \\ \chi_k^h &= (\mathcal{Q}_k - \mathcal{Q}_{k-1}) \varphi^h \qquad k = 1, \ldots, \ell\,. && (2.8b) \end{aligned}$$

3. Define the extensions $u_k^h \in \mathbb{W}_k \setminus \mathbb{W}_{k-1}$ of the function χ_k^h

$$u_0^h(x_i^{(0)}) = \begin{cases} \chi_0^h(x_i^{(0)}) & , x_i^{(0)} \in \Gamma, \\ \overline{\chi} & , x_i^{(0)} \notin \Gamma, \end{cases} \qquad (2.9a)$$

$$u_k^h(x_i^{(k)}) = \begin{cases} \chi_k^h(x_i^{(k)}) & , x_i^{(k)} \in \Gamma, \\ 0 & , x_i^{(k)} \notin \Gamma, \end{cases} \qquad k = 1, \ldots, \ell\,. \qquad (2.9b)$$

For $\overline{\chi}$ we choose either the mean value of the boundary function χ_0^h or the solution of the proper PDE on the coarsest grid with Dirichlet boundary conditions χ_0^h.

Now, the extension u^h is simply

$$\mathcal{E}_{\overline{\Omega}\Gamma} \varphi^h \;:=\; u^h \;=\; \sum_{k=0}^{\ell} u_k^h\,. \qquad (2.10)$$

In Algorithm 1 the discrete representation B_{IC}^T of the transposed operator $\mathcal{E}_{\overline{\Omega}\Gamma}^T$ is needed so we prefer the recursive version of definition (2.10):

$$\begin{aligned} v_0 &:= u_0^h && (2.11a) \\ v_k &:= v_{k-1} + u_k^h \qquad k = 1, \ldots, \ell\,, && (2.11b) \end{aligned}$$

and set $\mathcal{E}_{\overline{\Omega}\Gamma} \varphi^h := v_\ell$. Note that $v_k \in \mathbb{W}_k$ is the extension of $\mathcal{Q}_k \varphi^h$ on level $k = 0, \ldots, \ell$.

3 Matrix Representation of Multilevel Extension Operators

Using the FE isomorphism we change from the operator description of the extension $\mathcal{E}_{\overline{\Omega}\Gamma}$ to the matrix representation. The bilinear FE basis $\Psi^{(k)}$ will be defined similarly to (1.2) and the matrices $I_{C,k}$ denote the proper identities on level k.

The multilevel extension of a function $’^h = \sum_i^{N_{C,\ell}} ’_i \psi_i^{(\ell)} \in \mathbb{V}_\ell$ represented by the vector $\underline{’} \in \mathbb{R}^{N_{C,\ell}}$ into a function $u^h = \sum_i^{N_\ell} u_i \psi_i^{(\ell)} \in \mathbb{W}_\ell$ represented by the vector $\underline{u} \in \mathbb{R}^{N_\ell}$ consists of three steps:

1. Determine the rectangular $N_{C,k} \times N_{C,\ell}$ <u>projection matrix</u> Q_k and define the coefficients of the projection $\mathcal{Q}_k ’^h$ in the FE nodal basis of level k
$$\underline{\beta}_k := Q_k \underline{’} \qquad k = 0, \dots, \ell\ . \tag{3.12}$$
2. According to (2.8) <u>split the vectors</u> $\underline{\beta}_k$ into the coefficients of the multilevel nodal basis presentation of $’^h = \sum_{k=0}^{\ell} \sum_i^{N_{C,k}} \alpha_i^{(k)} \psi_i^{(k)}$. Denoting by $P_{C,k}^{k+1}$, $P_{I,k}^{k+1}$, $P_{IC,k}^{k+1}$ the usual linear FE interpolation matrices on the proper subsets of nodes we can determine the coefficient vectors $\underline{\alpha}_k$
$$\underline{\alpha}_0 \ := \ \underline{\beta}_0 \tag{3.13a}$$
$$\underline{\alpha}_k \ := \ \begin{pmatrix} -P_{C,k-1}^k & I_{C,k} \end{pmatrix} \begin{pmatrix} \underline{\beta}_{k-1} \\ \underline{\beta}_k \end{pmatrix} \qquad k = 1, \dots, \ell. \tag{3.13b}$$
3. The coefficients $\underline{v}_k$ of the <u>extensions</u> $v_k = \sum_{i=1}^{N_k} v_i^{(k)} \psi_i^{(k)}$ are determined by
$$\underline{v}_0 = \begin{pmatrix} \underline{v}_{C,0} \\ \underline{v}_{I,0} \end{pmatrix} \ := \ \begin{pmatrix} I_{C,0} \\ B_{IC,0} \end{pmatrix} \underline{\alpha}_0 \tag{3.14a}$$
$$\underline{v}_k = \begin{pmatrix} \underline{v}_{C,k} \\ \underline{v}_{I,k} \end{pmatrix} \ := \ \begin{pmatrix} I_{C,k} & P_{C,k-1}^k & 0 \\ 0 & P_{IC,k-1}^k & P_{I,k-1}^k \end{pmatrix} \begin{pmatrix} \underline{\alpha}_k \\ \underline{v}_{C,k-1} \\ \underline{v}_{I,k-1} \end{pmatrix} \tag{3.14b}$$
Set $E_{IC} \underline{’} := \underline{v}_\ell$.
The matrix $B_{IC,0}$ can be chosen as $\frac{1}{N_{C,0}} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix}_{N_{I,0} \times N_{C,0}}$, mapping the mean value of the boundary data into the interior. Another approach is the discrete harmonic extension on the coarsest grid with respect to the PDE, i.e., $B_{IC,0} = -K_{I,0}^{-1} K_{IC,0}$.

4 Various Multilevel Extension Techniques

Although having in mind the operator representation from Section 2 we give the following extension techniques mostly in the matrix representation from Section 3. Note, that the theoretical results for the first two techniques below require material coefficients $\lambda(x) = \lambda_i > 0 \quad \forall x \in \overline{\Omega}_i \quad (i = 1, \dots, p)$ whereas the third technique is also available for more general coefficients.

The Hierarchical Extension Technique

Using just the injection from level ℓ to level k for defining the projections Q_k we get exactly the hierarchical extension technique $\widehat{E}_{IC}$ which was first proposed in [HLMN94]. The proofs of the following statements can be found therein.

Setting $B_{IC} := \widehat{E}_{IC}$ the constant c_E in (1.6) behaves like

$$\text{in 2D}: \quad c_E(\widehat{E}_{IC}) = \mathcal{O}(\ln h^{-1}) = \mathcal{O}(\ell) \qquad \text{in 3D}: \quad c_E(\widehat{E}_{IC}) = \mathcal{O}(h^{-1}) \,. \tag{4.15}$$

This hierarchical extension can also be used as initial guess in an iteration approximating the extension, i.e., $B_{IC} := M_I^s \widehat{E}_{IC} + (I_I - M_I^s)K_I^{-1}(-K_{IC})$ with some iteration operator M_I. If there exists an h-independent positive constant such that $\| M_I \|_{K_I} \leq \eta < 1$ holds, e.g., using multigrid to define M_I, then in the 2D case $s = \mathcal{O}(\ln(\ln h^{-1}))$ iterations are sufficient to achieve an h-independent constant c_E.

The BPX-like Extension Technique

When approaching a BPX splitting with the L_2 ortho-projections of the boundary data ’^{h} the projection matrix will be expressed by $Q_k = M_{k,k}^{-1} M_{k,\ell}$, where the entries in the mass matrix $M_{k,p}$ are defined via the L_2 inner product $m_{i,j}^{(k,p)} = (\psi_i^{(k)}, \psi_j^{(p)})_{L_2}$. Using bilinear FE functions in the 2D case the matrix $M_{k,k}$ possesses exactly 3 non-zero entries per row. Whereas this matrix is rather easy to invert in the 3D case this will require too much arithmetic work.

Therefore a mass lumping or the proposal in [Nep95] for defining the projection operator $\mathcal{Q}_k$ is used. This results in an easily invertible diagonal matrix $\overline{M}_{k,k}$ defining $Q_k := \overline{M}_{k,k}^{-1} M_{k,\ell}$ and leads to the BPX-like extension technique $\overline{E}_{IC}$ for which

$$c_E(\overline{E}_{IC}) = \mathcal{O}(1) \tag{4.16}$$

was proved in [Nep95] for the 2D as well as for the 3D case.

Multilevel Extension Techniques Plus Smoothing

The recursive definition of the extension in (3.14) leads to the idea of an additional improvement of the extensions given above via some linear smoothing procedure $S_{I,k} : \mathbb{R}^{N_{I,k}} \to \mathbb{R}^{N_{I,k}}$, $k = 1, \ldots, \ell$ with the properties

$$\begin{array}{llll} \text{High frequencies}: & \| S_{I,k}\, \underline{v}^h \|_{K_I} \leq \varrho_k \| \underline{v}^h \|_{K_I} & \forall \Phi_I \underline{v}^h \in \mathbb{W}_k \setminus \mathbb{W}_{k-1} \\ \text{Low frequencies}: & \| S_{I,k}\, \underline{v}^h \|_{K_I} \leq \sigma_k \| \underline{v}^h \|_{K_I} & \forall \Phi_I \underline{v}^h \in \mathbb{W}_k \;, \end{array} \tag{4.17}$$

where the number of smoothing sweeps is denoted by ν_k. We require smoothing factors $\varrho_k \leq \sigma_k \leq 1$ independent of $h = 2^{-l}$. So, definition (3.14b) changes into

$$\begin{pmatrix} \underline{v}_{C,k} \\ \underline{v}_{I,k} \end{pmatrix} := \begin{pmatrix} I_{C,k} & 0 \\ -(I_{I,k} - S_{I,k}^{\nu_k}) K_{I,k}^{-1} K_{IC,k} & S_{I,k}^{\nu_k} \end{pmatrix} \begin{pmatrix} I_{C,k} & P_{C,k-1}^k & 0 \\ 0 & P_{IC,k-1}^k & P_{I,k-1}^k \end{pmatrix} \begin{pmatrix} \underline{\alpha}_k \\ \underline{v}_{C,k-1} \\ \underline{v}_{I,k-1} \end{pmatrix} \tag{4.18}$$

Using the hierarchical extension together with the smoothing, i.e., setting $B_{IC} = \widehat{E}_{IC}(\nu)$, it was shown in [Haa97] that in the 2D case

$$c_E(\widehat{E}_{IC}(\nu)) \leq c \left(1 + \sqrt{\ell} \cdot \sqrt{\sum_{k=1}^{\ell} \varrho_k^{2\nu_k} \prod_{j=k+1}^{\ell} \sigma_j^{2\nu_j}} \right) = \mathcal{O}(\ell) \tag{4.19}$$

holds with a positive and h-independent constant c. In comparison to the hierarchical extension the order remains the same but the constants hidden in that order statement are partially controlled by the smoothing. Now, $\nu_k = \mathcal{O}(\ln(\ln h^{-1}))$ iterations of the smoothing procedure $S_{I,k}$ are sufficient to achieve an h-independent constant c_E so that an additional iteration, as in the hierarchical extension technique, is no longer needed.

In the BPX-like extension additional smoothing sweeps are also feasible but the theory for this is still open. In that case the parameter σ_k will have more influence on the behavior of the constant c_E.

5 An Algorithmic Improvement of the Preconditioner

The improvement of Algorithm 1 is based upon three observations:

A) In Algorithm 1 the matrices B_{IC}^T and C_I^{-1} are applied to the same vector $\underline{r}_I$.

B) In the transposed operation to (3.14b) used in B_{IC}^T,

$$\begin{pmatrix} \underline{\alpha}_k \\ \underline{v}_{C,k-1} \\ \underline{v}_{I,k-1} \end{pmatrix} := \begin{pmatrix} I_{C,k} & 0 \\ \left(P_{C,k-1}^k\right)^T & \left(P_{IC,k-1}^k\right)^T \\ 0 & \left(P_{I,k-1}^k\right)^T \end{pmatrix} \begin{pmatrix} \underline{v}_{C,k} \\ \underline{v}_{I,k} \end{pmatrix} ,$$

the last row is simply the usual linear restriction from the finer to the coarser grid. A similar observation for the extension technique plus smoothing in (4.18) leads to

$$\underline{v}_{I,k-1} := \left(P_{I,k-1}^k\right)^T \left(S_{I,k}^T\right)^{\nu_k} \underline{v}_{I,k} . \tag{5.20}$$

C) Perform in a multigrid algorithm at level k ν_k smoothing sweeps with the iteration matrix $\overset{*}{S}_{I,k}$, ($\overset{*}{S}_{I,k}$ means the adjoint matrix to $S_{I,k}$ in the $K_{I,k}$-energy inner product) together with calculation and restriction of the defect. Then that brief algorithm,

$$\underline{w}_{I,k} := (I_{I,k} - (\overset{*}{S}_{I,k})^{\nu_k}) K_{I,k}^{-1} \underline{f}_{I,k} \quad \text{and} \quad \underline{d}_{I,k-1} := \left(P_{I,k-1}^k\right)^T (\underline{f}_{I,k} - K_{I,k} \underline{w}_{I,k}) ,$$

can be simplified into

$$\underline{d}_{I,k-1} := \left(P_{I,k-1}^k\right)^T \left(S_{I,k}^T\right)^{\nu_k} \underline{f}_{I,k} . \tag{5.21}$$

Now, choosing $\overset{*}{S}_{I,k}$ as pre-smoother and $S_{I,k}$ as post-smoother in a multigrid algorithm defining C_I we preserve the required symmetry and may use restriction and pre-smoothing from the implementation of the transposed extension B_{IC}^T. When, additionally, in that extension a coarse grid solver is used more than half of the algorithmic work can be saved when applying C_I^{-1}. This algorithmic improvement does not depend on the choice of the projections Q_k !

Numerical tests using this improved algorithm together with the hierarchical extension technique can be found in [Haa97].

6 A test example

For checking the theoretical results concerning the behavior of the constant c_E it is necessary that the spectral equivalence constants in (1.5) are independent of h. Therefore, the test example consists of the PDE

$$-\Delta\, u(x) \;=\; 1 \quad \text{in } \Omega = (0,1)\times(0,0.5) \qquad \text{and} \qquad u(x) = 0 \quad \text{on } \partial\Omega \;,$$

where the domain $\overline{\Omega}$ was decomposed into two squares. Using in the preconditioner C (1.4) Dryja's approach [Dry82] as Schur complement preconditioner C_C and exact solvers for $C_{I,i}^{-1}$, we achieve h-independent spectral equivalence constants in (1.5), so the condition number of the preconditioned system $\kappa(C^{-1}K)$ is influenced only by the extension B_{IC}.

The discrete system (1.3) was solved with a preconditioned parallelized CG until a relative accuracy of 10^{-6} measured in the $\|\cdot\|_{KC^{-1}K}$-norm of the error was reached. Due to estimates (1.7) the number of CG iterations behaves like $\mathcal{O}(c_E^2)$.

On level 0 with the discretization parameter $h = 0.25$, i.e., just one node on the interface between the two subdomains, the automatic mesh generator produces a triangular mesh with 4 inner nodes per subdomain. All finer meshes were produced by simply subdividing each triangle into 4 congruent ones. The Gauß-Seidel smoother $S_{I,k}$ applied $\nu = \nu_k$-times $(k = 1, \ldots, \ell)$ and a coarse grid solver were used in the extension B_{IC}. Whereas in Table 1 the iteration numbers connected with the

Table 1 Number of CG iterations for the test example using 2 processors

Splitting	ν	$\ell=0$	$\ell=1$	$\ell=2$	$\ell=3$	$\ell=4$	$\ell=5$	$\ell=6$
hier.	0	2	7	11	17	24	30	36
hier.	1	2	6	8	11	13	16	19
hier.	2	2	5	7	8	10	13	14
BPX-like	0	2	7	8	11	12	13	13

hierarchical splitting grow linearly with the number of levels ℓ the iteration numbers of the BPX-like splitting tend to an upper bound. This confirms the theoretical results from (4.15), (4.16) and (4.19). Obviously the additional smoothing sweeps decrease the number of iterations and also the condition number $\kappa(C^{-1}K)$ rapidly.

7 Final Remarks

Although the development was done on rather simple operators, the results carry over to systems of symmetric elliptic second order PDE s resulting in symmetric $\mathbb{V}_0$-elliptic and $\mathbb{V}_0$-bounded bilinear forms. The extension techniques used in the test example work also very well in more challenging examples and result in a faster solver when the algorithmic improvement from Paragraph 4 is used; see also [Haa97].

In the latter paper, the full proof for estimate (4.19) can be found and it was shown that this estimate holds for the whole range from extension without smoothing to the exact harmonic extension. Especially smoothing controls similar to the multigrid slash cycle and generalized slash cycle were investigated, so that $O(\ln \ell)$ smoothing sweeps are sufficient to achieve a condition number $\kappa(C^{-1}K) = O(1)$. The algorithmic improvement from Paragraph 5 was derived in a more heuristic way. The electrical machine with jumping coefficients and a complicated geometry proved that the new extension method is also successfully applicable to practical problems.

The implementation and theoretical analysis of the BPX-like extension together with smoothing sweeps will be done in a forthcoming paper. Also, a comparison to other extension techniques proposed in [Nep91, Che93, BPV96] should be done on a more challenging example, with respect to the CPU time needed to solve (1.1).

REFERENCES

[Boe89] Boergers M. (1989) The Neumann–Dirichlet domain decomposition method with inexact solvers on the subdomains. *Numerische Mathematik* 55(2): 123–136.

[BPS89] Bramble J., Pasciak J., and Schatz A. (1986, 1987, 1988, 1989) The construction of preconditioners for elliptic problems by substructuring I – IV. *Mathematics of Computation* 47, 103–134, 49, 1–16, 51, 415–430, 53, 1–24.

[BPV96] Bramble J., Pasciak J., and Vassilev A. (1996) Analysis of non-overlapping domain decomposition algorithms with inexact solves. Technical report. available via http://www.math.tamu.edu/~james.bramble/papers.html.

[BPX90] Bramble J., Pasciak J., and Xu J. (1990) Parallel multilevel preconditioners. *Mathematics of Computation* 55(191): 1–22.

[Che93] Cheng H. (1993) *Iterative Solution of Elliptic Finite Element Problems on Partially Refined Meshes and the Effect of Using Inexact Solvers.* PhD thesis, Courant Institute of Mathematical Science, New York University.

[Dry82] Dryja M. (1982) A capacitance matrix method for Dirichlet problems on polygonal regions. *Numerische Mathematik* 39(1): 51–64.

[Haa97] Haase G. (May 1997) Hierarchical extension operators plus smoothing in domain decomposition preconditioners. *Applied Numerical Mathematics* 23(3).

[HL92] Haase G. and Langer U. (1992) The non-overlapping domain decomposition multiplicative Schwarz method. *International Journal of Computer Mathemathics* 44: 223–242.

[HLM91] Haase G., Langer U., and Meyer A. (1991) The approximate Dirichlet domain decomposition method. Part I: An algebraic approach. Part II: Applications to 2nd-order elliptic boundary value problems. *Computing* 47: 137–151 (Part I), 153–167 (Part II).

[HLMN94] Haase G., Langer U., Meyer A., and Nepomnyaschikh S. (1994) Hierarchical extension operators and local multigrid methods in domain decomposition preconditioners. *East-West Journal of Numerical Mathematics* 2: 173–193.

[Nep91] Nepomnyaschikh S. (1991) Method of splitting into subspaces for solving elliptic boundary value problems in complex-form domains. *Sov. J. Numer. Anal. Math. Modelling* 6(2).

[Nep95] Nepomnyaschikh S. (1995) Optimal multilevel extension operators. Report 95-3, TU Chemnitz.

[Osw94] Oswald P. (1994) *Multilevel Finite Element Approximation.* Teubner, Stuttgart.

[SBG96] Smith B., Bjorstad P., and Gropp W. (1996) *Domain Decomposition : parallel methods for elliptic partial differential equations.* Cambridge University Press.

[TCK92] Tong C., Chan T., and Kuo C. J. (1992) Multilevel filtering preconditioners: Extensions to more general elliptic problems. *SIAM J. Sci. Stat. Comput.* 13: 227–242.

[Xu89] Xu J. (1989) Theory of multilevel methods. Technical Report AM48, Department of Mathematics, Penn State University.

[Yse86] Yserentant H. (1986) On the multi-level splitting of finite element spaces. *Numer. Math.* 49(4): 379–412.