Hierarchical Extension Operators and Local Multigrid Methods in Domain Decomposition Preconditioners

G. Haase*, U. Langer*, A. Meyer† and S.V. Nepomnyaschikh‡

East-West Journal of Numerical Mathematics:2, 1994, pp.173-193

Abstract

Domain Decomposition (DD) is not only the basic tool for data partitioning but also a successful technique for constructing new parallel pde solvers. The efficiency of the solver essentially depends on the preconditioner. In the present paper, we consider finite element (f.e.) schemes approximating plane, symmetric, second-order elliptic boundary value problems (b.v.p.). We derive and analyse so-called Dirichlet DD preconditioners based on the non-overlapping DD, modified Schur-complement preconditioners, local multigrid methods and hierarchical extension procedures. The crucial point is the combination of the almost norm-preserving and quite cheap hierarchical extension procedures with local multigrid. The symmetric multiplicative version of the preconditioning algorithm seems to be especially well suited for this approach. The numerical experiments carried out on various multiprocessor systems confirm the high efficiency of the parallel DD solvers proposed.

Keywords : Boundary value problems, finite element method, domain decomposition, preconditioners, parallel iterative solvers.

1991 Mathematics subject classifications : 65N55, 65N50, 65N30

1 Introduction

In some foregoing papers [16, 18, 19, 9, 20, 12, 22, 13, 15], the authors proposed the parallelization and the preconditioning of the Conjugate Gradient (CG) method on the basis of a non-overlapping Domain Decomposition (DD) approach. In Sections 2 and 3 we review some of the results of these papers. The DD preconditioner proposed contains three components which can be chosen in order to adapt the preconditioner to the problem under consideration as well as possible. One component is a (modified) Schur-complement preconditioner that has been studied by the DD communitiy very intensively [4, 6, 17, 19]. Another component is a preconditioner for the local homogeneous Dirichlet problems arising in each subdomain. The most sensitive part is the basis transformation matrix transforming the nodal f.e. basis on the interfaces into the approximate discrete harmonic basis [13]. In order to construct the two last components, we can use local multigrid methods. Local multigrid methods with zero initial guess have been already used for constructing the Dirichlet problem preconditioner as well as for the basis transformation. In the first case this is certainly sufficiently efficient. However, in the basis transformation case, the analysis shows that in general one has to carry out at least $O(\ln h^{-1})$ multigrid iterations in order to bound the term caused by the basis transformation in the condition number estimate unifomely in h. In [18, 19, 22], a norm preserving extension procedure was proposed that immediately provides a uniform bound. In Section 4 of this paper, we combine these ideas, i.e. the grid functions obtained by extension from the coupling boundaries (interfaces) are used as initial guesses for the local multigrid methods in the subdomains. So, we can impove the extented grid function very efficiently in direction of the harmonic

*Johannes Kepler University Linz, Inst. of Math., Altenberger Str. 69, A–4040 Linz, Austria
†Technical University Chemnitz–Zwickau, Dept. of Math., PSF 964, D–09009 Chemnitz, Germany
‡Sib. Branch of the Russian Academy of Sciences, Computing Centre, Lavrentier avenue 6, R-630090 Novosibirsk, Russia

extension without paying to much for it. On the other hand, we can weaken the extension procedure. In Section 5, we propose a hierarchical, nearly norm-preserving extension procedure which is much more cheaper and easier to implement than the extension procedure proposed earlier [18, 19, 22]. However, one has to pay by a logarithmic grow of the extension constent. This grow can be compensated by $O(\ln\ln h^{-1})$ (= 1 in the range of practical applications) multigrid iterations (Section 6). In Section 7, we present the results of the numerical experiments carried out on various multiprocessor systems. We use up to 128 processors and obtain asymptotical efficiencies of significantly more than 90% for an asymptotically almost optimal algorithm. The symmetric multiplicative version [10] of the preconditioning algorithm seems to be especially well suited for this approach. This follows from the analysis and is confirmed by the experiments at the same time.

2 The Nonoverlapping DD FEM

We consider the abstract symmetric, $\mathbf{V}_0$–elliptic and $\mathbf{V}_0$–bounded variational problem

$$\text{Find } u \in \mathbf{V}_0 \,:\, a(u,v) \;= < F, v > \;\; \forall v \in \mathbf{V}_0 \,, \tag{2.1}$$

arising from the weak formulation of a scalar second–order, symmetric and uniformly elliptic boundery value problem (b.v.p.) given in a plane bounded domain $\Omega \subset \mathbf{R}^2$ with a piecewise smooth boundary $\Gamma = \partial\Omega$.

As in the finite element substructuring technique, we decompose Ω into p non-overlapping subdomains Ω_i $(i = 1, 2, \ldots, p)$ such that $\overline{\Omega} = \bigcup_{i=1}^{p} \overline{\Omega}_i$, and each subdomain Ω_i into Courant's linear triangular finite elements δ_r such that this discretization process results in a conform triangulation of Ω . In the following, the indices "C" and "I" correspond to the nodes belonging to the coupling boundaries (interfaces) $\Gamma_C = \bigcup_{i=1}^{p} \partial\Omega_i \setminus \Gamma_D$ and to the interior $\Omega_I = \bigcup_{i=1}^{p} \Omega_i$ of the subdomains, respectively, where Γ_D is that part of $\partial\Omega$ where Dirichlet–type boundary conditions are given.

In Section 7, we consider the homogenious Dirichlet problem $(u = 0$ on $\Gamma_D = \Gamma)$ for the potential equation ($-\mathrm{div}\,(a\nabla u) = f$ in $\Omega = (0,2) \times (0,1)$) as a model problem for testing our algorithms. The variational formulation (2.1) of this problem consists in finding some function $u \in \mathbf{V}_0 = \overset{\circ}{\mathbf{H}}{}^1(\Omega)$ such that

$$\int_\Omega a(x)\, \nabla^T u(x)\, \nabla v(x)\, \mathrm{dx} \;=\; \int_\Omega f(x)\, v(x)\, \mathrm{dx} \qquad \forall v \in \mathbf{V}_0\,.$$

We decompose Ω into p = 8, 32 (see Fig. 1) and 128 subdomains Ω_i ($i = 1, 2, \ldots, p$) and each subdomain into three–node triangular finite elements as is shown in Fig.1. We suppose that $a(x) = a_i = \text{const} > 0 \quad \forall x \in \overline{\Omega}_i$.

Define the usual f.e. nodal basis

$$\Phi = \left[\varphi_1, \cdots, \varphi_{N_C}, \varphi_{N_C+1}, \cdots, \varphi_{N_C+N_{I,1}}, \cdots, \varphi_{N=N_C+N_I}\right] \,, \tag{2.2}$$

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 $N_I = N_{I,1} + N_{I,2} + \ldots + N_{I,p}$. The f.e. subspace

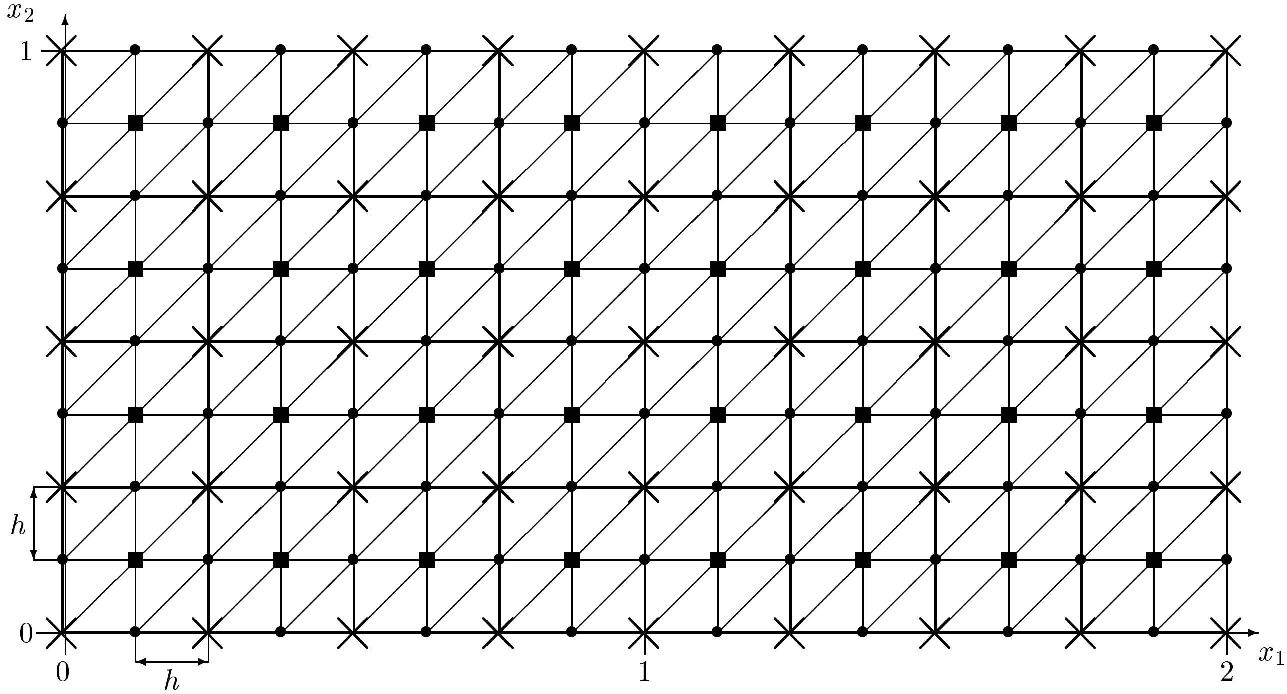
$$\mathbf{V} \;=\; \mathbf{V}_h \;=\; \mathrm{span}(\Phi) \;=\; \mathrm{span}(\Phi V) \;=\; \mathbf{V}_C + \mathbf{V}_I \subset \mathbf{V}_0 \tag{2.3}$$

can be represented as direct sum of the subspaces $\mathbf{V}_C = \mathrm{span}(\Phi V_C)$ and $\mathbf{V}_I = \mathrm{span}(\Phi V_I)$ with

$$V \;=\; (V_C\; V_I) \;=\; I \;=\; \begin{pmatrix} I_C & O \\ O & I_I \end{pmatrix}_{N\times N} , V_C \;=\; \begin{pmatrix} I_C \\ O \end{pmatrix}_{N\times N_C} \text{ and } V_I = \begin{pmatrix} O \\ I_I \end{pmatrix}_{N\times N_I} .$$

The f.e. isomorphism between the f.e. function $u \in \mathbf{V}$ and the corresponding vector $\underline{u} = (\underline{u}_C^T, \underline{u}_I^T)^T \in \mathbf{R}^N$ of the nodal parameters is given by

$$\mathbf{V} \ni u \;=\; \Phi V \underline{u} \;=\; \Phi \underline{u} \quad \overset{\Phi}{\longleftrightarrow} \quad \underline{u} \in \mathbf{R}^N \quad . \tag{2.4}$$



$\{\blacksquare\} \,\widehat{=}\, "I"$; $\{\bullet\} \cup \{\times\} \,\widehat{=}\, "C"$; $\{\times\}$ = cross points

$\Omega = \bigcup_{i=1}^{32} \Omega_i$, $p = 32$; $\Gamma_D = \Gamma = \partial\Omega$; $\Gamma_C = \bigcup_{i=1}^{32} \partial\Omega_i \setminus \Gamma_D$

Figure 1: 8×4 Domain decomposition and level-0-triangulation ($l = 0$) of our model problem

Once the basis Φ is chosen, the f.e. approximation

$$\text{(2.5)} \qquad \text{Find } u = \Phi V \underline{u} \in \mathbf{V} \ : \ a(\Phi V \underline{u}, \Phi V \underline{v}) = < F, \Phi V \underline{v} > \ \forall v = \Phi V \underline{v} \in \mathbf{V} \,,$$

leads to a large–scale sparse system

$$\text{(2.6)} \qquad K \, \underline{u} \ = \ \underline{f}$$

of finite element equations with the symmetric and positiv definite stiffness matrix K. Because of the arrangement of the basis functions made above, the system (2.6) can be rewritten in the block form

$$\text{(2.7)} \qquad \begin{pmatrix} K_C & K_{CI} \\ K_{IC} & K_I \end{pmatrix} \begin{pmatrix} \underline{u}_C \\ \underline{u}_I \end{pmatrix} = \begin{pmatrix} \underline{f}_C \\ \underline{f}_I \end{pmatrix} \quad ,$$

where $K_I = \text{blockdiag}\,(K_{I,i})_{i=1,2,\ldots,p}$ is blockdiagonal.

3 Some DD Preconditioners

Now we use the Parallel Preconditioned Conjugate Gradient (ParPCG) method for solving (2.6)–(2.7) on massively parallel computers. The data distribution and the parallelization of the CG–method is described in [13] in detail. The crucial point is the preconditioning equation

$$\text{(3.1)} \qquad C \, \underline{w} \ = \ \underline{r}$$

which must fit into the DD parallelization concept proposed earlier in [12, 13].

In [12, 13], the ASM–DD preconditioner

$$\text{(3.2)} \qquad C = \begin{pmatrix} I_C & K_{CI} B_I^{-T} \\ O & I_I \end{pmatrix} \begin{pmatrix} C_C & O \\ O & C_I \end{pmatrix} \begin{pmatrix} I_C & O \\ B_I^{-1} K_{IC} & I_I \end{pmatrix}$$

was derived on a purely algebraic basis. This preconditioner contains the three components C_C, $C_I = \operatorname{diag}(C_{I,i})_{i=1,2,\ldots,p}$ and $B_I = \operatorname{diag}(B_{I,i})_{i=1,2,\ldots,p}$, which can be freely chosen in order to adapt the preconditioner to the specialities of the problem under consideration.

In [12, 13], the first three authors proved the following result.

THEOREM 3.1 *Let the symmetric and positive definite block preconditioners C_C and C_I satisfy the spectral equivalence inequalities*

$$\underline{\gamma}_C C_C \le S_C + T_C \le \overline{\gamma}_C C_C \quad \text{and} \quad \underline{\gamma}_I C_I \le K_I \le \overline{\gamma}_I C_I \tag{3.3}$$

with some positive constants $\underline{\gamma}_C$, $\overline{\gamma}_C$, $\underline{\gamma}_I$ and $\overline{\gamma}_I$. Then the ASM-DD preconditioner (3.2) satisfies the spectral equivalence inequalities

$$\underline{\gamma} C \le K \le \overline{\gamma} C \tag{3.4}$$

with the spectral equivalence constants

$$\underline{\gamma} = \min\{\underline{\gamma}_C, \underline{\gamma}_I\} \left(1 - \sqrt{\tfrac{\mu}{1+\mu}}\right) \quad \text{and} \quad \overline{\gamma} = \max\{\overline{\gamma}_C, \overline{\gamma}_I\} \left(1 + \sqrt{\tfrac{\mu}{1+\mu}}\right) \tag{3.5}$$

where $\mu = \rho(S_C^{-1} T_C)$ denotes the spectral radius of $S_C^{-1} T_C$, with $S_C = K_C - K_{CI} K_I^{-1} K_{IC}$ and $T_C = K_{CI}(K_I^{-1} - B_I^{-T}) K_I (K_I^{-1} - B_I^{-1}) K_{IC}$. For the spectral condition number $\kappa(C^{-1}K)$ of $C^{-1}K$, the two–side estimate

$$\left(\underline{\gamma}_1 \,/\, \overline{\gamma}_1\right) (\sqrt{\mu} + \sqrt{1+\mu})^2 \le \kappa(C^{-1}K) \le \left(\overline{\gamma}_1 \,/\, \underline{\gamma}_1\right) (\sqrt{\mu} + \sqrt{1+\mu})^2 \tag{3.6}$$

holds with $\underline{\gamma}_1 = \min\{\underline{\gamma}_C, \underline{\gamma}_I\}$ and $\overline{\gamma}_1 = \max\{\overline{\gamma}_C, \overline{\gamma}_I\}$.

The matrix B_I, which is not supposed to be neither symmetric nor positive definite, can be interpreted as a part of the basis transformation matrix

$$\widetilde{V} = \left(\widetilde{V}_C \; \widetilde{V}_I\right) = \begin{pmatrix} I_C & O \\ -B_I^{-1} K_{IC} & I_I \end{pmatrix} \tag{3.7}$$

transforming the nodal basis Φ in the approximate discrete harmonic basis $\widetilde{\Phi} = \Phi \widetilde{V}$ and μ can be defined by the angle between $\widetilde{\mathbf{V}}_C = \operatorname{span}(\Phi \widetilde{\mathbf{V}}_C)$ and $\widetilde{\mathbf{V}}_I = \mathbf{V}_I = \operatorname{span}(\Phi \widetilde{V}_I)$, [12, 13]. More precisely,

$$\sqrt{\frac{\mu}{1+\mu}} = \cos \measuredangle \left(\widetilde{\mathbf{V}}_C, \widetilde{\mathbf{V}}_I\right) . \tag{3.8}$$

An alternative interpretation follows from the inequalities

$$\begin{aligned}
(S_C \underline{u}_C, \underline{u}_C) &= \inf_{\underline{v}_I \in \mathbf{R}^{N_I}} \left(K \begin{pmatrix} \underline{u}_C \\ \underline{v}_I \end{pmatrix}, \begin{pmatrix} \underline{u}_C \\ \underline{v}_I \end{pmatrix}\right) = \inf_{\substack{u \in \mathbf{V} \\ u = u_c \, on \, \Gamma_C}} a(u, u) \\
&= \left(K \begin{pmatrix} \underline{u}_C \\ -K_I^{-1} K_{IC} \underline{u}_C \end{pmatrix}, \begin{pmatrix} \underline{u}_C \\ -K_I^{-1} K_{IC} \underline{u}_C \end{pmatrix}\right) \\
&\le \left(K \begin{pmatrix} \underline{u}_C \\ -B_I^{-1} K_{IC} \underline{u}_C \end{pmatrix}, \begin{pmatrix} \underline{u}_C \\ -B_I^{-1} K_{IC} \underline{u}_C \end{pmatrix}\right) \\
&= ((S_C + T_C) \underline{u}_C, \underline{u}_C) \le (1+\mu)(S_C \underline{u}_C, \underline{u}_C) \qquad \forall \underline{u}_C \in \mathbf{R}^{N_C} ,
\end{aligned} \tag{3.9}$$

namely, that the function $\Phi\binom{\underline{u}_C}{-B_I^{-1} K_{IC} \underline{u}_C}$ should be a norm–preserving extension of the function $u_C = \Phi V_C \underline{u}_C$ on Γ_C in the sense of the energy norm induced by the Schur–complement S_C.

S.V. Nepomnyaschikh proved in [22] that one can construct a norm–preserving extension $\Phi\binom{\underline{u}_C}{E_{IC} \underline{u}_C}$ of u_C such that the inequality

$$\left\| \Phi \begin{pmatrix} \underline{u}_C \\ E_{IC} \underline{u}_C \end{pmatrix} \right\|_{H^1(\Omega)} \le c_E \; \| u_C \|_{H^{\frac{1}{2}}(\Gamma_C)} \tag{3.10}$$

holds for all $u_C = \Phi V_C \underline{u}_C$ and $\underline{u}_C \in \mathbf{R}^{N_C}$, where $E_{IC} : \mathbf{R}^{N_C} \to \mathbf{R}^{N_I}$ denotes the corresponding extension operator and c_E is an h–independent positive constant. Replacing $-B_I^{-1} K_{IC} \underline{u}_C$ by $E_{IC}\underline{u}_C$, we obtain an h–independent bound $1+\mu = \tilde{c}_E{}^2$, where $\tilde{c}_E$ is defined by c_E and by the norm equivalence constants between the $\mathbf{H}^1(\Omega)$– and the K–energy norm on the one hand and the $\mathbf{H}^{\frac{1}{2}}(\Gamma_C)$ and the S_C–energy norm on the other hand. In [18], the splitting $\mathbf{V} = \widetilde{\mathbf{V}}_C + \widetilde{\mathbf{V}}_I$, with $\widetilde{\mathbf{V}}_C = \text{span}(\Phi \widetilde{\mathbf{V}}_C)$, $\widetilde{\mathbf{V}}_I = \mathbf{V}_I = \text{span}(\Phi \widetilde{V}_I)$, and

$$\widetilde{V_C} = \begin{pmatrix} I_C \\ E_{IC} \end{pmatrix}, \tag{3.11}$$

was used in order to derive asymptotically optimal ASM–Preconditioners. In Section 4, we will use norm–preserving extensions of the form $E_{IC}\underline{u}_C$ as initual guesses in the local multigrid methods defining $B_{I,i}$ for $i = 1, 2, \ldots, p$.

Let us return to some algorithmic aspects for solving the preconditioning equation (3.1) and to some modifications of the basic preconditioning algorithm (Algorithm 1) given by the ASM–DD preconditioner (3.2). The basic preconditioning Algorithm 1 can be rewritten in the form $\underline{w} = C^{-1}\underline{r}$ as

Algorithm 1 : The ASM-DD Preconditioner [12, 13]
$\underline{\mathbf{w}}_C = C_C^{-1} \sum_{i=1}^{p} A_{C,i}^T \left(\underline{r}_{C,i} - K_{CI,i} B_{I,i}^{-T} \underline{r}_{I,i} \right)$ $\underline{\mathbf{w}}_{I,i} = C_{I,i}^{-1} \underline{r}_{I,i} - B_{I,i}^{-1} K_{IC,i} \underline{\mathbf{w}}_{C,i} \qquad ; \ i = 1, 2, \ldots, p$
Determined by the user : $C_C =?$, $C_I =?$, $B_I =?$

where $A_i = \begin{pmatrix} A_{C,i} & A_{CI,i} \\ A_{IC,i} & A_{I,i} \end{pmatrix}$ denotes the subdomain connectivity matrix which is used for a convenient notation only. The subdomain f.e. assembling process which is connected with nearest neighbour communication stands behind this notation [20, 12, 13].

The first modification consists in choosing $B_I = C_I$ which simplifies Algorithm 1. The resulting preconditioner is called simple DD preconditioner [2, 20, 12, 14].

Algorithm 2 : The simple DD Preconditioner [2, 14]
$\underline{\mathbf{w}}_C = C_C^{-1} \sum_{i=1}^{p} A_{C,i}^T \left(\underline{r}_{C,i} - K_{CI,i} C_{I,i}^{-T} \underline{r}_{I,i} \right)$ $\underline{\mathbf{w}}_{I,i} = C_{I,i}^{-1} \left(\underline{r}_{I,i} - K_{IC,i} \underline{\mathbf{w}}_{C,i} \right) \qquad ; \ i = 1, 2, \ldots, p$
Determined by the user : $C_C =?$, $C_I =?$

However, this choice does not care for the fundamental differences between B_I and C_I.

Two further modifications (i.e. special choices of B_I and C_I) of Algorithm 1 were discussed in [15, 10]. The last one is the symmetric multiplicative version. Besides the better convergence behaviour in comparison with the additive version, it is especially suited for the combination of the extension procedure with local multigrid und gives the best numerical results. These algorithms will be discussed in the next section.

4 Local Multigrid Methods with Non-Zero Initial Guesses in the DD Preconditioning

In this section we use local multigrid methods for defining C_I and B_I. Because of the obvious spectral equivalence inequalities

$$S_C \;\le\; S_C + T_C \;\le\; (1+\mu)\, S_C, \tag{4.1}$$

we can utilize Schur-Complement-Preconditioners in order to define C_C. In recent years various S_C-preconditioners have been developed by the DD community [4, 6, 17, 19]. In our numerical experiments presented in the final section we use the properly scaled BPS-preconditioner proposed in [4].

If one applies k symmetric multigrid steps to

$$K_{I,i}\, \underline{w}_{I,i} \;=\; \underline{r}_{I,i} \qquad ,\; i = 1,2,\ldots,p\;, \tag{4.2}$$

with the zero initial guess, then the symmetric multigrid preconditioner

$$C_I \;=\; K_I \left(I_I - M_I^k\right)^{-1} \tag{4.3}$$

is implicitly defined, where $M_I = \mathrm{diag}\,[M_{I,i}]_{i=1,2,\ldots,p}$ denotes the multigrid iteration operator of the multigrid method applied [12, 13]. In the same way, we can use non-necessarily symmetric local multigrid methods for the basis transformation, i.e.

$$B_I \;=\; K_I \left(I_I - \overline{M}_I^s\right)^{-1} \tag{4.4}$$

where $\overline{M}_I = \mathrm{diag}\left[\overline{M}_{I,i}\right]_{i=1,2,\ldots,p}$ and s denotes the corresponding multigrid iteration operator and the number of multigrid steps applied, respectively.

Let us now return to the estimate of the spectral constants $\underline{\gamma}_I$, $\overline{\gamma}_I$ and, especially, μ for second-order elliptic b.v.p. considered in Section 2 :

Lemma 4.1 *If* $M_I = \overset{*}{M}_I \ge 0$ *is selfadjoint and non-negative in the* K_I*-energy inner product, then the multigrid preconditioner (4.3) is symmetric and positive definite. Furthermore,* C_I *satisfies the spectral equivalence inequalities (3.3) with* $\underline{\gamma}_I = 1 - \eta^k$ *and* $\overline{\gamma}_I = 1$ *, where* $\eta = \rho(M_I) = \| M_I \|_{K_I} < 1$ *denotes the spectral radius of* M_I*, or some estimate of* $\rho(M_I)$ *from above.*

Lemma 4.2 *If* $\| \overline{M_I} \|_{K_I} \le \overline{\eta} < 1$ *, then* $\mu \le \overline{\eta}^{2s}(\frac{c}{h} - 1)$ *, where* h *denotes the usual discretization parameter and* c *is an* h*-independent positive constant.*

The proofs of both lemmas can be found in [12, 13]. ■

Lemma 4.2 implies an h-independent μ-estimate if and only if $\overline{\eta}^{2s} = \mathcal{O}(h)$, i.e. $s = \mathcal{O}(\ln h^{-1})$. Thus, an improvement of the basis transformation technique is required. Looking carefully on the right-hand side of the basis transformation equation

$$B_{I,i}\, \underline{\widetilde{w}}_{I,i} \;=\; -K_{IC,i}\, \underline{w}_{C,i} \qquad i = 1,2,\ldots,p\;, \tag{4.5}$$

we observe that the right-hand side belongs to the subspace $K_{IC}\mathbf{R}^{N_C}$. So, we must adapt the basis transformation technique to this subspace. This idea was discussed in [7, 8]. In particular, the frequency filtering technique of G. Wittum [24] was used for B_I. Another idea consists in using a non-zero initial guess in the local multigrid method defining B_I. There are, at least, two good candidates for such non-zero, adapted initial guesses, namely

1. that one obtained by special Full-Multigrid-Methods and

2. that one obtained by some norm-preserving extension technique (cf. Section 3).

The first one was discussed in [11]. Let us discuss here the second one. Suppose that

$$\underline{w}_I^s = -B_I^{-1} K_{IC}\, \underline{w}_C \tag{4.6}$$

results from the application of s multigrid steps to the equation

$$K_I\, \underline{w}_I = -K_{IC}\, \underline{w}_C \tag{4.7}$$

with the (non-zero) initial guess

$$\underline{w}_I^0 = E_{IC}\, \underline{w}_C \tag{4.8}$$

obtained by some extension procedure. For simplicity of representation, we assume that E_{IC} has the form

$$E_{IC} = -E_I K_{IC} \qquad \text{and} \qquad E_{CI} = E_{IC}^T = -K_{CI} E_I^T. \tag{4.9}$$

Then we have

$$\begin{aligned} \underline{w}_I^s &= \overline{M}_I^s \underline{w}_I^0 + \left(I_I - \overline{M}_I^s\right) K_I^{-1} \left(-K_{IC}\underline{w}_C\right) \\ &= \overline{M}_I^s\, E_{IC}\, \underline{w}_C + \left(I_I - \overline{M}_I^s\right) K_I^{-1} \left(-K_{IC}\underline{w}_C\right) \\ &= B_{IC}\underline{w}_C \;=\; -B_I^{-1} K_{IC}\underline{w}_C\, , \end{aligned} \tag{4.10}$$

with

$$B_{IC} = -B_I^{-1} K_{IC} = \overline{M}_I^s\, E_{IC} + \left(I_I - \overline{M}_I^s\right) K_I^{-1} \left(-K_{IC}\right), \tag{4.11}$$

$$B_I^{-1} = \overline{M}_I^s\, E_I + \left(I_I - \overline{M}_I^s\right) K_I^{-1} = \left(I_I - \overline{M}_I^s \left(I_I - E_I K_I\right)\right) K_I^{-1}. \tag{4.12}$$

The transposed operation

$$\underline{w}_C^s = -K_{CI} B_I^{-T}\, \underline{r}_I = B_{CI}\, \underline{r}_I, \tag{4.13}$$

with

$$B_{CI} = B_{IC}^T \;\; \textit{and} \;\; B_I^{-T} = \left(I_I - \left(I_I - E_I^T K_I\right) \overset{*}{\overline{M}}{}_I^s\right) K_I^{-1}, \tag{4.14}$$

means the following steps:

1. Apply s adjoint multigrid steps to $K_I \underline{w}^I = \underline{r}^I$ with the initial guess $\underline{w}_I^0 = 0$, obtaining $\underline{w}_I^s = (I_I - \overset{*}{\overline{M}}{}_I^s) K_I^{-1} \underline{r}_I$;
2. Compute the defect $\underline{d}_I^s = \underline{r}_I - K_I \underline{w}_I^s$;
3. Compute $\underline{w}_C^s = -K_{CI} \underline{w}_I^s + E_{CI} \underline{d}_I^s$,

where $E_{CI} = E_{IC}^T$. Mention that in (4.12) the term $(I_I - E_I K_I)$ and in (4.14) the term $(I_I - E_I^T K_I)$ can be interpreted as pseudo-pre-smoothing step in the first multigrid cycle and pseudo-post-smoothing step in the last multigrid cycle, respectively.

In [10], the first two authors showed that the MSM (Multiplicative Schwarz Method) DD Preconditioner corresponding to the ASM preconditioner of Algorithm 1 the components of which are defined by (4.3) and (4.4) is nothing else but an ASM preconditioner with the modified components $B_I = K_I \left(I_I - M_I^k \overline{M}_I^s (I_I - E_I K_I)\right)^{-1}$ and $C_I = K_I \left(I_I - M_I^{2k}\right)^{-1}$. Therefore, the preconditioning equation $\underline{w} = C^{-1}\underline{r}$ can be rewritten in the following form:

Algorithm 3 : The MSM-DD Preconditioner [10]				
1)	$\underline{y}_{I,i}$	$=$	$(I_{I,i} - M^k_{I,i})\, K^{-1}_{I,i} \underline{r}_{I,i}$	$i = 1, 2, \ldots, p$
2)	$\underline{\hat{w}}_{I,i}$	$=$	$\overset{*}{\overline{M}}{}^s_{I,i} \underline{y}_{I,i} + (I_{I,i} - \overset{*}{\overline{M}}{}^s_{I,i})\, K^{-1}_{I,i} \underline{r}_{I,i}$	$i = 1, 2, \ldots, p$
3)	$\underline{w}_C$	$=$	$C^{-1}_C \sum\limits_{i=1}^{p} A^T_{C,i} \left(\underline{r}_{C,i} - K_{CI,i} \underline{\hat{w}}_{I,i} + E_{CI,i}(\underline{r}_{I,i} - K_{I,i} \underline{\hat{w}}_{I,i})\right)$	$i = 1, 2, \ldots, p$
4)	$\underline{z}_{I,i}$	$=$	$\overline{M}^s_{I,i} E_{IC,i} \underline{w}_{C,i} + (I_{I,i} - \overline{M}^s_{I,i})\, K^{-1}_{I,i}\, (-K_{IC,i} \underline{w}_{C,i})$	$i = 1, 2, \ldots, p$
5)	$\underline{\tilde{r}}_{I,i}$	$=$	$\underline{r}_{I,i} - K_{IC,i} \underline{w}_{C,i}$	$i = 1, 2, \ldots, p$
6)	$\underline{\tilde{w}}_{I,i}$	$=$	$\underline{y}_{I,i} + \underline{z}_{I,i}$	$i = 1, 2, \ldots, p$
7)	$\underline{w}_{I,i}$	$=$	$M^k_{I,i} \underline{\tilde{w}}_{I,i} + (I_{I,i} - M^k_{I,i})\, K^{-1}_{I,i} \underline{\tilde{r}}_{I,i}$	$i = 1, 2, \ldots, p$
Determined by the user :			$C_C = ?,\ M_I = ?,\ \overline{M}_I = ?,\ k = ?,\ s = ?$	

Note that even in the case $s = 0$ the extented vector is improved by the k symmetric multigrid cycles defined by the iteration operator M_I. In this sense, the MSM-DD preconditioning algorithm is especially suited for the combination of the extension procedure with local multigrid. Indeed, the best numerical results were obtained by Algorithm 3 with $s = 0$ and $k = 1$, where M_I was defined by the cheapest V-cycle. We can even use non-symmetric cycles (e.g. V-cycle with 1 smoothing step only) for defining C_I, namely in the form $C_I = K_I \left(I_I - \overset{*}{M}{}^k_I M^k_I \right)^{-1}$ ensuring the symmetry of C_I.

The crucial point in the analysis of convergence rate of the ASM-DD- as well as the MSM-DD–ParPCG is the estimation of the spectral radius $\mu = \rho(S_C^{-1} T_C)$ defined by B_I. The following lemma provides an estimate of μ for the case that B_I, or B_{IC}, is given by (4.11), i.e. the basis transformation is defined by s multigrid iterations with a non-zero initial guess obtained from the extension of the coupling boundary data.

Lemma 4.3 *If*

$$\| \overline{M}_I \|_{K_I} \le \overline{\eta} < 1 \qquad \textit{and} \tag{4.15}$$

$$\left\| \begin{pmatrix} \underline{v}_C \\ E_{IC} \underline{v}_C \end{pmatrix} \right\|_K \le \tilde{c}_E \, \| \underline{v}_C \|_{S_C} \quad \forall \underline{v}_C \in \mathbf{R}^{N_C} , \tag{4.16}$$

then

$$\mu \equiv \rho(S_C^{-1} T_C) \le \overline{\eta}^{2s} \, (1 + \tilde{c}_E)^2 . \tag{4.17}$$

Proof. Using the identity

$$\left(K \left[\begin{pmatrix} \underline{v}_C \\ E_{IC} \underline{v}_C \end{pmatrix} - \begin{pmatrix} \underline{v}_C \\ -K_I^{-1} K_{IC} \underline{v}_C \end{pmatrix} \right] , \left[\begin{pmatrix} \underline{v}_C \\ E_{IC} \underline{v}_C \end{pmatrix} - \begin{pmatrix} \underline{v}_C \\ -K_I^{-1} K_{IC} \underline{v}_C \end{pmatrix} \right] \right)$$

$$= \| E_{IC} \underline{v}_C + K_I^{-1} K_{IC} \underline{v}_C \|^2_{K_I} \quad \forall \underline{v}_C \in \mathbf{R}^{N_C} , \tag{4.18}$$

we can estimate

$$\mu = \rho(S_C^{-1} T_C) = \max_{\underline{v}_C \in \mathbf{R}^{N_C} \setminus \{0\}} \frac{\| K_I^{-1} K_{IC} \underline{v}_C - B_I^{-1} K_{IC} \underline{v}_C \|^2_{K_I}}{\| \underline{v}_C \|^2_{S_C}}$$

$$= \max_{\underline{v}_C \in \mathbf{R}^{N_C} \setminus \{0\}} \frac{\left\| K_I^{-1} K_{IC} \underline{v}_C + \overline{M}^s_I E_{IC} \underline{v}_C + (I_I - \overline{M}^s_I) K_I^{-1} (-K_{IC} \underline{v}_C) \right\|^2_{K_I}}{\| \underline{v}_C \|^2_{S_C}}$$

$$
\begin{aligned}
&= \max_{\underline{v}_C \in \mathbf{R}^{N_C}\setminus\{0\}} \frac{\left\| \overline{M}_I^s (E_{IC}\underline{v}_C + K_I^{-1} K_{IC}\underline{v}_C)\right\|_{K_I}^2}{\| \underline{v}_C \|_{S_C}^2} \\
&\leq \| \overline{M}_I \|_{K_I}^{2s} \max_{\underline{v}_C \in \mathbf{R}^{N_C}\setminus\{0\}} \frac{\| E_{IC}\underline{v}_C + K_I^{-1} K_{IC}\underline{v}_C \|_{K_I}^2}{\| \underline{v}_C \|_{S_C}^2} \\
&= \| \overline{M}_I \|_{K_I}^{2s} \left[\max_{\underline{v}_C \in \mathbf{R}^{N_C}\setminus\{0\}} \frac{\left\| \binom{\underline{v}_C}{E_{IC}\underline{v}_C} - \binom{\underline{v}_C}{-K_I^{-1}K_{IC}\underline{v}_C} \right\|_K}{\| \underline{v}_C \|_{S_C}} \right]^2 \\
&\leq \overline{\eta}^{2s} \left[\max_{\underline{v}_C \in \mathbf{R}^{N_C}\setminus\{0\}} \frac{\left\| \binom{\underline{v}_C}{E_{IC}\underline{v}_C} \right\|_K + \left\| \binom{\underline{v}_C}{-K_I^{-1}K_{IC}\underline{v}_C} \right\|_K}{\| \underline{v}_C \|_{S_C}} \right]^2 \\
&\leq \overline{\eta}^{2s} \, (\tilde{c}_E + 1)^2 \; .
\end{aligned}
$$

■

This estimate of μ together with the spectral equivalence inequalities (3.3) for C_I and C_C, respectively, yield immediately rate estimates for the ASM-DD- as well as the MSM-DD–ParPCG in correspondence to the estimates given in Theorem 3.1 (ASM) and in [10] (MSM).

5 The Hierarchical Extension Operator

Let us now construct a very simple and easily implementable, almost norm preserving extension procedure $E_{IC} : \mathbf{R}^{N_C} \to \mathbf{R}^{N_I}$ on the basis of the hierarchical transformation technique proposed by H. Yserentant in [25] for preconditioning finite element equations. It is obviously sufficient to construct the extension for each subdomain $\overline{\Omega}_i$ separately. So, we omit the subindex i and describe the extension $E_{\overline{\Omega}\Gamma}$ of a piecewise linear function given on the boundary Γ $(= \Gamma_{C,i})$ of Ω to some piecewise linear function on $\overline{\Omega}$ $(= \overline{\Omega}_i)$. Emphasize that throughout this section Ω and Γ play the rule of the subdomain Ω_i and of the subdomain boundary $\Gamma_i = \partial\Omega_i$, respectively. For simplicity, we suppose that Ω is a polygonal plane domain and, of course, bounded.

Now we are forced to introduce explicitly a sequence of finer and finer triangulation. Starting with a coarse grid triangulation

$$\overline{\Omega}_0^h = \cup_{i=1}^{M_0} \overline{\tau}_i^{(0)}, \quad diam(\tau_i^{(0)}) = O\,(1)$$

of Ω, we refine Ω_0^h several times. This results in a sequence of nested triangulations $\Omega_0^h,\ \Omega_1^h,\ \ldots,\ \Omega_J^h$ such that

$$\overline{\Omega}_k^h = \cup_{i=1}^{M_k} \overline{\tau}_i^{(k)}, \quad k = 0, 1, \ldots, l,$$

where the triangles $\tau_i^{(k+1)}$ are generated by subdividing triangles $\tau_i^{(k)}$ into four congruent subtriangles connecting the midpoints of the edges (red subdivision) [1]. In the following, the indices "l" and "0" correspond to fine and coarse level quantities, respectively. Denote by $x_i^{(k)}, i = 1, 2, \ldots, L_k$ the nodes of the triangulation Ω_k^h.

Introduce now the spaces W_k and V_k of finite element functions. The space W_k consists of real-valued functions which are continuous on Ω and linear on the triangles in Ω_k^h. The space V_k is the space of traces on Γ of functions from W_k:

$$V_k = \{\varphi^h : \quad \varphi^h = u^h|_\Gamma, \text{ with } u^h \in W_k\}.$$

We will consider the usual norms of the Sobolev spaces $H^1(\Omega)$ and $H^{\frac{1}{2}}(\Gamma)$, respectively, in the finite element subspaces W_l and V_l, too:

$$
\begin{aligned}
\|u^h\|^2_{H^1(\Omega)} &= \|u^h\|^2_{L_2\Omega} + |u^h|^2_{H^1(\Omega)},\\
\|u^h\|^2_{L_2(\Omega)} &= \int_\Omega \Big((u^h(x)\Big)^2\,dx,\\
|u^h|^2_{H^1(\Omega)} &= \|\nabla u^h\|^2_{L_2(\Omega)},\\
\|\varphi^h\|^2_{H^{\frac12}(\Gamma)} &= \|\varphi^h\|^2_{L_2(\Gamma)} + |\varphi^h|^2_{H^{\frac12}(\Gamma)},\\
\|\varphi^h\|^2_{L_2(\Gamma)} &= \int_\Gamma \Big((\varphi^h(x)\Big)^2\,dx,\\
|\varphi^h|^2_{H^{\frac12}(\Gamma)} &= \int_\Gamma\int_\Gamma \frac{\Big(\varphi^h(x)-\varphi^h(y)\Big)^2}{|x-y|^2}dx\,dy.
\end{aligned}
$$

Our goal is the construction of some norm-preserving explicit extension operator $E_{\overline{\Omega}\Gamma}$ from V_l into W_l:

$$E_{\overline{\Omega}\Gamma} : V_l \to W_l.$$

As was mentioned above, the basis of our construction is the hierarchical decomposition of the space V_l which was suggested by Yserentant [25, 26]:

$$\varphi^h = I_0\varphi^h + \sum_{k=1}^{l}(I_k - I_{k-1})\varphi^h \quad \forall\varphi^h \in V_l,$$

where $I_k\varphi^h \in V_k$ denotes the finite element interpolant. Introduce the notation

$$
\begin{aligned}
\varphi_0^h &= I_0\varphi^h,\\
\varphi_k^h &= (I_k - I_{k-1})\varphi^h, \quad k = 1,\dots,l,
\end{aligned}
$$

and define the extension $u_k^h \in W_k$ of the function φ_k^h in the following way:

$$
u_0^h(x_i^{(0)}) = \begin{cases} \varphi_0^h(x_i^{(0)}), & x_i^{(0)} \in \Gamma,\\[2mm] \overline{\varphi}, & x_i^{(0)} \notin \Gamma,\end{cases}
$$

$$
u_k^h(x_i^{(k)}) = \begin{cases} \varphi_k^h(x_i^{(k)}), & x_i^{(k)} \in \Gamma,\\[2mm] 0, & x_i^{(k)} \notin \Gamma.\end{cases}
$$

Here $\overline{\varphi}$ is, for instance, the mean value of the function φ_0^h on Γ:

$$\overline{\varphi} = \frac{1}{N_0}\sum_{i=1}^{N_0}\varphi_0^h(x_i^{(0)}),$$

where $N_k, k = 0,1,\dots,l$ is the number of nodes $x_i^{(k)}$ on Γ. We assume that nodes $x_i^{(k)}$ are enumerated at first on Γ (in the natural order) and then inside Ω. Set

$$E_{\overline{\Omega}\Gamma}\varphi^h = u^h \equiv u_0^h + u_1^h + \cdots + u_l^h. \tag{5.1}$$

To study the extension operator (5.1), we need the following lemma.

Lemma 5.1 *Let I be the unit segment $I=[0,1]$ and*

$$I^h=\{x_i=ih \mid i=0,1,\ldots,n,\ h=1/n\}$$

be the uniform mesh on I. Further, let the space V consist of real-valued functions which are continuous on I and linear on $[x_i,x_{i+1}]$. Then there exists a positive constant c, independent of h, such that

$$\max_{x_i\in I^h}|\varphi^h(x_i)|\le c(\log h^{-1})^{\frac{1}{2}}\|\varphi^h\|_{H^{\frac{1}{2}}(I)}\qquad \forall\varphi^h\in V.$$

Proof. Let us introduce an auxiliary unit square Ω with the uniform mesh Ω^h :

$$\begin{aligned}\Omega &= \{(x,y)|0<x<1,0<y<1\},\\ \Omega^h &= \{x_{ij}|x_{ij}=(ih,jh),\ i,j=0,1,\ldots,n\}.\end{aligned}$$

Let W be the piecewise-linear finite element space on Ω^h. It follows from the trace theorem for mesh functions [21] that there exists a positive constant c_1, independent of h, such that for any $\varphi^h\in V$ there exists $u^h\in W$ such that

$$\begin{gathered}u^h|_I=\varphi^h,\\ \|u^h\|_{H^1(\Omega)}\le c_1\|\varphi^h\|_{H^{\frac{1}{2}}(I)}.\end{gathered}$$

Then we have

$$\max_{x_i\in I^h}|\varphi^h(x_i)|\ \le\ \max_{x_{ij}\in\Omega^h}|u^h(x_{ij})|\ \le$$

$$\le\ c_2(\log h^{-1})^{\frac{1}{2}}\|u^h\|_{H^1(\Omega)}\ \le\ c_1c_2(\log h^{-1})^{\frac{1}{2}}\|\varphi^h\|_{H^{\frac{1}{2}}(I)}.$$

where the constant c_2 is independent of h [3, 23]. ∎

Let us consider hierarchical meshes on the segment I with mesh sizes $h_k=2^{-k}, k=0,1,\ldots,l$ and corresponding hierarchical spaces $V_0,V_1,\ldots,V_l$. According to [26], define the following hierarchical norm:

$$|||\varphi^h|||^2=\|I_0\varphi^h\|^2_{H^{\frac{1}{2}}(I)}+\sum_{k=1}^{l}2^k\|(I_k-I_{k1})\varphi^h\|^2_{L_2(I)}.$$

Then the following lemma is valid:

Lemma 5.2 *There exists a positive constant c, independent of h, such that*

$$|||\varphi^h|||\le c(l)^{\frac{1}{2}}\|\varphi^h\|_{H^{\frac{1}{2}}(I)}.$$

Proof. It follows from Lemma 5.1 that there exists a positive constant c_1, independent of h, such that

$$\|I_0\varphi^h\|^2_{H^{\frac{1}{2}}(I)}\le c_1\cdot l\cdot\|\varphi^h\|^2_{H^{\frac{1}{2}}(I)}.$$

Consider the term $\|(I_k-I_{k-1})\varphi^h\|_{L_2}$. We have

$$\|(I_k-I_{k-1})\varphi^h\|^2_{L_2(I)}=\sum_{i=1}^{N_k-1}\|I_k\varphi^h-I_{k-1}\varphi^h\|^2_{L_2(x^{(k)}_{i-1},x^{(k)}_{i+1})}$$

$$=\sum_{i=1}^{N_k-1}\left(\varphi^h(x^{(k)}_i)-\frac{1}{2}\left(\varphi^h(x^{(k)}_{i-1})+\varphi^h(x^{(k)}_{i+1})\right)\right)^2\cdot\frac{2}{3}h_k\le$$

$$\le\sum_{i=1}^{N_k-1}\frac{1}{3}h_k\left(\left(\varphi^h(x^{(k)}_i)-\varphi^h(x^{(k)}_{i-1})\right)^2+\left(\varphi^h(x^{(k)}_{i+1})-\varphi^h(x^{(k)}_i)\right)^2\right).$$

Then there exists a positive constant c_2, independent of h, such that [21]:

$$\|(I_k - I_{k-1})\varphi^h\|^2_{L_2(I)} \leq c_2 \|\varphi^h\|^2_{H^{\frac{1}{2}}(I)} \cdot h_k,$$

i.e.

$$2^k \|(I_k - I_{k-1})\varphi^h\|_{L_2(I)} \leq c_2 \|\varphi^h\|^2_{H^{\frac{1}{2}}(I)}$$

Then

$$\sum_{k=1}^{l} 2^k \|(I_k - I_{k-1})\varphi^h\|^2_{L_2(I)} \leq c_2 l \|\varphi^h\|^2_{H^{\frac{1}{2}}(I)}.$$

■

Combining the Lemma 5.1 and Lemma 5.2, we have the following theorem.

THEOREM 5.1 *There exists a positive constant c, independent of h, such that*

$$\|E_{\overline{\Omega}\Gamma}\varphi^h\|_{H^1(\Omega)} \leq c \cdot l \cdot \|\varphi^h\|_{H^{\frac{1}{2}}(\Gamma)},$$

where the hierarchical extension operator $E_{\overline{\Omega}\Gamma}$ was defined in (5.1).

Proof. Let us first prove the following estimates:

$$\|u^h_k\|^2_{H^1(\Omega)} \leq c_1 2^k \|\varphi^h\|^2_{L_2(\Gamma)},$$

where the constant c_1 is independent of h. For $k = 0$, it is evident. Then, for $k \geq 1$, we have

$$\|u^h_k\|^2_{H^1(\Omega)} = \sum_{\overline{\tau}^{(k)}_i \cap \Gamma \neq 0} \|u^h_k\|^2_{H^1(\tau^{(k)}_i)} \leq$$

$$\leq c_2 \sum_{x^{(k)}_i \in \Gamma} \left(\varphi^h_k(x^{(k)}_i)\right)^2 \leq c_3 \cdot 2^k \|\varphi^h_k\|^2_{L_2(\Gamma)}.$$

Here c_2 and c_3 are independent of h. Now, in order to estimate

$$\|u^h_0 + u^h_1 + \cdots + u^h_k\|_{H^1(\Omega)},$$

we will estimate

$$\sum_{k=1}^{l-1} \sum_{j=k+1}^{l} |(\nabla u^h_k, \nabla u^h_j)_{L_2(\Omega)}|.$$

Then from the Cauchy-inequality we have

$$|(\nabla u^h_k, \nabla u^h_j)_{L_2(\Omega)}| \ \leq \ c_4 (2^k \|\varphi^h_k\|^2_{L_2(\Gamma)} + 2^j \|\varphi^h_j\|_{L_2(\Gamma)}). \tag{5.2}$$

Summing up the estimates (5.2) and using Lemma 5.2, we arrive at the statement of theorem. ■

6 Final Results

Let us summarize and discuss the results of Sections 3 - 5.

THEOREM 6.1 *Let us use the hierarchical extension procedure $E_{IC} : \mathbf{R}^{N_C} \to \mathbf{R}^{N_I}$ described in Section 5 to construct an initial guess for the local multigrid method defining the basis transformation (4.12). Further we assume that M_I defining C_I is given by a symmetric local multigrid cycle and that the modifed Schur complement $S_C + T_C$ is preconditioned by C_C.*
If there are h-independent constants $\underline{\gamma}_C$, $\overline{\gamma}_C$ (properly scaled for the MSM, i.e contained in $(0,2)$!) and η, $\overline{\eta} \in (0,1)$ such that

$$\begin{gathered}\underline{\gamma}_C C_C \ \le\ S_C + T_C \ \le\ \overline{\gamma}_C C_C, \\ \| M_I \|_{K_I} \le \eta < 1,\ k = O(1), \\ \| \overline{M_I} \|_{K_I} \le \overline{\eta} < 1,\ s = O(\ln\ln h^{-1}) \end{gathered} \tag{6.1}$$

and if the solution of the Schur-complement preconditioning equation does not require more than $O(h^{-2})$ operation (cf. Remark 6.1)), then the ASM-DD as well as the MSM-DD preconditioned ParPCG for solving the (2D) finite element equations (2.6) need at most $O(\ln \varepsilon^{-1})$ iterations and $O(h^{-2}\ \ln(\ln h^{-1})\ \ln \varepsilon^{-1})$ arithmetical operations (per processor) in order to reduce the initial error by the factor ε, where $\varepsilon \in (0,1)$ denotes the relative accuracy required, and h is the so-called local discretization parameter such that the number N_i of unknowns per subdomain $\overline{\Omega}_i$ is of the order $O(h^{-2})$.

Proof. The rate estimates for the ASM-DD (cf. Theorem 3.1 and [9]) and the MSM-DD (cf. [10]) preconditioned ParPCG are given in terms of $\underline{\gamma}_C$, $\overline{\gamma}_C$, η, $\overline{\eta}$ and μ. Since $\underline{\gamma}_C$, $\overline{\gamma}_C$, η, and $\overline{\eta}$ are independent of h, it remains to estimate μ. Combining the well-known norm equivalence inequalities

$$\underline{c}_1^2\ \| \Phi\underline{v} \|^2_{H^1(\Omega)} \le \| \underline{v} \|^2_K = a(\Phi\underline{v}, \Phi\underline{v}) \ \le\ \overline{c}_1^2\ \| \Phi\underline{v} \|^2_{H^1(\Omega)} \qquad \forall \underline{v} \in \mathbf{R}^N$$

and

$$\underline{c}_2^2\ \| \Phi V_C \underline{v}_C \|^2_{H^{\frac{1}{2}}(\Gamma_C)} \le \| \underline{v}_C \|^2_{S_C} \le\ \overline{c}_2^2\ \| \Phi V_C \underline{v}_C \|^2_{H^{\frac{1}{2}}(\Gamma_C)} \qquad \forall \underline{v}_C \in \mathbf{R}^{N_C}$$

with inequality

$$\left\| \Phi \begin{pmatrix} \underline{u}_C \\ E_{IC}\underline{u}_C \end{pmatrix} \right\|_{H^1(\Omega)} \ \le\ c_E\, l\ \ \| u_C \|_{H^{\frac{1}{2}}(\Gamma_C)} \tag{6.2}$$

of Theorem 5.1, we arrive at inequality (4.17) of Lemma 4.3., with

$$\tilde{c}_E \ =\ \overline{c}_1\, \underline{c}_2^{\,-1}\, c_E\, l\,,$$

where $l = O(\ln h^{-1})$. The constants $\underline{c}_1$, $\overline{c}_1$, $\underline{c}_2$ and $\overline{c}_2$ are independent of h. Lemma 4.3 yields

$$\mu \ \equiv\ \rho(S_C^{-1}T_C) \ \le\ \overline{\eta}^{2s}\, (1 + \overline{c}_1\, \underline{c}_2^{\,-1}\, c_E\, l)^2\ . \tag{6.3}$$

Therefore, since $s = O(\ln(l)) = O(\ln(\ln h^{-1}))$, μ is bounded independently of h. The complexity estimate follows now immediately from the assumption on C_C, from the $O(h^{-2})$ – complexity of one multigrid cycle and of the hierarchical extension procedure (cf. Section 5). ∎

Remark 6.1
1. If the more complicated extension procedure from [22] is used instead of the hierarchical extension proposed in Section 5, then we arrive at an asymptotically optimal method provided the other assumptions of Theorem 6.1 are valid. Moreover, since l disappears in the rigth-hand side of the μ-estimate (6.3), we can make this bound smaller than some given $\varepsilon \in (0,1)$ by $s = O(\ln \varepsilon^{-1})$ multigrid iterations.
2. In the numerical experiments presented in the next section, we use the so-called BPS-Schur-complement preconditioner proposed in [4]. Combining (4.1) with the results of [4], we arrive at (6.1), with $\underline{\gamma}_C = \underline{c}\, l^2$ and $\overline{\gamma}_C = \overline{c}\,(1+\mu)$, where the constents $\underline{c}$ and $\overline{c}$ are independent of h. The presents of l^2 causes a logarithmic grow in the number of the ParPCG iterations. The BPS-preconditioner costs $O(h^{-1}\ln h^{-1})$ operations for the so-called "egde" parts locally and the solution of the so-called "cross–point" system. The "cross–point" system can be solved on some underbalanced processor, or simultaneously on all processors, or again by some distributed solver in the case of several thousand processors. Note that the dimension of the "cross–point" system is proportional to the number of processors in our experiments (see Section 7).
3. The hierarchical extension technique can be easily adapted to non-uniform refinement procedures, e.g. to graded meshes studied in [11] for approximating singularities. The ASM- and MSM-DD preconditioning algorithms proposed in this paper can be generalized to higher-order triangular elements as well as to other finite elements.

7 Numerical Experiments on Massively Parallel Computers

Let us return to the example described in Section 2 (see also Fig. 1) and let us first study numerically the effects of the combination of the hierarchical extension procedure with local multigrid in the ASM-DD (Algorithm 1) and the MSM-DD (Algorithm 3) ParPCG for a fixed domain decomposition and for various triangulations (dependence on h).
Recall that we want to solve the Dirichlet problem for the Poisson-equation ($a(x) \equiv 1$) in the rectangle $(0,2) \times (0,1)$ which was decomposed into 32 subdomains ($p = 32$), and each subdomain was divided into three-nodes (linear) triangular elements. We use preconditioning Algorithm 1 (ASM) and Algorithm 3 (MSM) with an appropriately scaled BPS-Schur-complement preconditioner C_C ([4] and Remark 6.1.2) and the following local multigrid methods defining M_I and $\overline{M}_I$, respectively, C_I and B_I :

V11 : V-cycle with 1 pre- and 1 post-smoothing sweep on all grids;

V22 : V-cycle with 2 pre- and 2 post-smoothing sweeps on all grids;

3W22 : 3 W-cycles with 2 pre- and 2 post-smoothing sweeps on all grids, corresponding to the exact solution.

We use lexicographically forward and backward Gauss-Seidel sweeps in the pre- and post-smoothing, respectively. The interpolation and the restriction operators are defined by the bilinear interpolation $I^q_{I;k-1}$ and by the full weighting restriction $I^{k-1}_{I;k} = (I^k_{I;k-1})^T$ for all $k = 0, 1, \ldots, l$.

Tables 7.1 – 7.4 show the numbers $I(\varepsilon = 10^{-6})$ of iterations and the CPU-time in seconds for algorithm 1 (ASM) and Algorithm 3 (MSM). The first column of the tables contains the information whether the hierarchical extension E_{IC} was used or not. The ParPCG-iteration was stopped if the usual CG-relative accuracy $\varepsilon = 10^{-6}$, i.e.

$$(\underline{w}^j, \underline{r}^j) \;\le\; \varepsilon\,(\underline{w}^0, \underline{r}^0) \;,$$

had been attained. The numerical experiments presented in Tables 7.1 – 7.4 were carried out on a MULTICLUSTER–II-system with 32 T805 transputers and 8 MByte per node under the operating system PARIX 1.2. Recall that the 8-grid problem contains **2.100.225** unknowns !

components			$l = \#$ level - 1					
E_{IC}	$\overline{M}_I$	M_I	2	3	4	5	6	7
not used	$V11$	$V11$	16	20	27	36	54	84
used	$s = 0$	$V11$	37	47	60	70	83	100
used	$s = 0$	$V22$	36	46	58	68	80	97
used	$V11$	$V11$	14	14	16	17	18	20
used	$V11$	$V22$	13	14	15	17	18	20
not used	$3W22$	$3W22$	13	13	14	15	16	18

Table 7.1 Number of iterations for Algorithm 1 (ASM) (MultiCluster-II)

components			$l = \#$ level - 1					
E_{IC}	$\overline{M}_I$	M_I	2	3	4	5	6	7
not used	$V11$	$V11$	0.83	1.87	6.47	28.1	153.6	913.2
used	$s = 0$	$V11$	2.02	4.61	15.1	57.6	249.4	1130
used	$s = 0$	$V22$	2.01	4.81	16.0	62.8	273.4	1280
used	$V11$	$V11$	0.88	1.83	6.05	22.5	90.3	390.6
used	$V11$	$V22$	0.84	1.92	6.08	24.3	98.1	425.1
not used	$3W22$	$3W22$	1.06	3.01	11.8	49.5	211.3	950.4

Table 7.2 CPU-time in seconds for Algorithm 1 (ASM) (MultiCluster-II)

components			$l = \#$ level - 1					
E_{IC}	$\overline{M}_I$	M_I	2	3	4	5	6	7
not used	$V11$	$V11$	13	13	15	16	18	22
used	$s = 0$	$V11$	13	14	15	17	18	20
used	$s = 0$	$V22$	13	13	15	16	17	18
used	$V11$	$V11$	13	13	14	15	16	18
not used	$3W22$	$3W22$	13	13	14	15	16	18

Table 7.3 Number of iterations for Algorithm 3 (MSM) (MultiCluster-II)

components			$l = \#$ level - 1					
E_{IC}	$\overline{M}_I$	M_I	2	3	4	5	6	7
not used	$V11$	$V11$	0.90	2.50	7.13	24.4	102.5	484.8
used	$s = 0$	$V11$	0.88	1.75	5.04	19.4	76.5	326.0
used	$s = 0$	$V22$	0.87	2.16	5.78	21.7	86.6	357.1
used	$V11$	$V11$	0.98	2.73	6.39	25.1	98.5	431.3
not used	$3W22$	$3W22$	2.28	6.04	23.46	97.6	419.8	1900.0

Table 7.4 CPU-time in seconds for Algorithm 3 (MSM) (MultiCluster-II)

The fastest method is the preconditioning Algorithm 3 using the cheapest multigrid algorithm $V11$ for defining C_I and just applies the extension procedure E_{IC} without any multigrid step for defining B_I. The number of iterations differs from those where the subdomain problem with the system matrix $K_{I,i}$ is de facto solved exactly (cf. also [4]) by 1 or, at most, 2 iterations. In these cases the number of iterations grows like $\ln h^{-1}$ caused by the BPS interface preconditioner [4]. So, we constructed a fast and arithmetic cheap preconditioner the iteration numbers of which behave nearly like the iteration numbers using exact solvers per subdomain.

Now let us study the scale-up and the corresponding efficiency for the algorithm giving the best results in our first example, namely the Algorithm 3 (MSM) with E_{IC} (is used), $s = 0$ and M_I defined by V11. We use 16, 64 and 8, 32, 128 processors corresponding to the decomposition of $\Omega = (0,1) \times (0,1)$ into 16, 64 subdomains and to the decomposition of $\Omega = (0,2) \times (0,1)$ into 8, 32 (see Fig. 1) and 128 subdomains. We measure the scale-up $S(i,j)$ and the efficiency $E(i,j)$ for the 6-level ($l = 5$) case (Table 7.5) and the 7-level ($l = 6$) case (Table 7.6). These experiments were carried out on a GC-system with 192 T805 transputers and 4 MByte per node (up to 128 processors were used in our experiments). Note that the 7-level case for 128 subdomains has total **2.100.225** unknowns and **16.641** unknowns per subdomain $\overline{\Omega}_i$.

$\rightarrow$ j $\downarrow$ —— S(i,j) i E(j,i) ——	8	16	32	64	128
8	—	—	**3.79**	—	**12.56**
16	—	—	—	3.52	—
32	**0.95**	—	—	—	**3.34**
64	—	0.88	—	—	—
128	**0.79**	—	**0.84**	—	—

Table 7.5 Scale-up $S(i,j)$ and Efficiency $E(i,j)$ for the 6-level $l = 5$) case (only 15 % of the local storage was used)

$\rightarrow$ j $\downarrow$ —— S(i,j) i E(j,i) ——	8	16	32	64	128
8	—	—	**3.96**	—	**14.35**
16	—	—	—	3.92	—
32	**0.99**	—	—	—	**3.67**
64	—	0.98	—	—	—
128	**0.91**	—	**0.92**	—	—

Table 7.6 Scale-up $S(i,j)$ and Efficiency $E(i,j)$ for the 7-level ($l = 6$) case

In correspondence to tables 7.3 and 7.4, the tables 7.7 and 7.8 show the number $I(\varepsilon = 10^{-6})$ of iterations and the CPU-time in seconds for Algorithm 3 (MSM) in the case of 128 subdomains corresponding to 128 processors. Mention that the GC-system is approximately 10% slower than the MULTICLUSTER-II.

components			$l = \#$ level - 1				
E_{IC}	$\overline{M}_I$	M_I	2	3	4	5	6
used	$s=0$	$V11$	14	16	17	18	19
used	$s=0$	$V22$	13	14	15	16	17
used	$V11$	$V11$	12	13	14	16	16
not used	$3W22$	$3W22$	11	13	14	15	16

Table 7.7 Number of iterations for Algorithm 3 (MSM, GCel 128 processors)

components			$l = \#$ level - 1				
E_{IC}	$\overline{M}_I$	M_I	2	3	4	5	6
used	$s=0$	$V11$	6.11	7.13	11.0	27.0	96.7
used	$s=0$	$V22$	4.55	6.44	11.0	28.2	104.1
used	$V11$	$V11$	4.33	6.17	10.6	33.0	124.8

Table 7.8 CPU-time in seconds for Algorithm 3 (MSM, GCel 128 processors)

Similar experiments were carried out on the nCube2S parallel computer with 64 processors and 8 MByte per node. Table 7.9 contains the number of levels used, the total number of unknowns, the number $I(\varepsilon = 10^{-6})$ of iterations, the CPU–time in Seconds and the Efficiency of using 64 instead of 16 processors for Algotrithm 3 (MSM: E_{IC} used; $s = 0; M_I = V11, C_C = BPS$) in the case of $64(8 \times 8)$ subdomains ($\Omega = (0,1) \times (0,1)$) corresponding to 64 processors. Mention that the nCube2S is approximately 2 times faster than the transputer systems.

Algorithm 3 (MSM) : E_{IC} used, $s = 0$, M_I = V11, C_C = BPS				
$\ell\ = \#$ level-1	4	5	6	7
Total number of unknowns	65.025	262.144	1.046.529	4.190.209
Number I($\varepsilon = 10^{-6}$)	15	17	18	20
CPU-time [seconds]	1.99	6.19	22.1	93.1
Efficiency [$4 \times 4 \longrightarrow 8 \times 8$]	0.80	0.88	0.99	0.95

Table 7.9 Numerical experiments on the nCube2S (64 subdomains = 64 processors)

The MSM-DD ParPCG using the hierarchical extension procedure, local multigrid and an appropriate choosen (modified) Schur-complement preconditioner seems to be an algorithm of at least almost asymptotically optimal complexity and of high efficiency on massively parallel computers. There is a hope that this remains true for non-uniformly refined grids (see [11] for graded grid near singularities). In 3D, the extension procedure must be changed in order to get good initial guesses for the multigrid at least in some asymptotical sense (cf. [22] and BPX-ideas [5]) for extensions analogous to hierarchical extensions. However, in the range of practical applications, the zero initial guess should be good enough in 3D (see [11]).

References

[1] R. E. Bank, T. F. Dupont, and H. Yserentant. The Hierarchical Basis Multigrid Method. *Numerische Mathematik*, 52:427–458, 1988.

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

[3] J. H. Bramble. A second order finite difference analogue of the first biharmonic boundary value problem. *Numer. Math.*, 9:236–249, 1966.

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

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

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

[7] G. Haase. Die nichtüberlappende Gebietszerlegungsmethode zur Parallelisierung und Vorkonditionierung des CG-Verfahrens. Report 92-10, IWR Heidelberg, 1992.

[8] G. Haase. *Die nichtüberlappende Gebietszerlegungsmethode zur Parallelisierung und Vorkonditionierung iterativer Verfahren.* Dissertation,Fakultät Mathematik und Naturwissenschaften der TU Chemnitz–Zwickau, 1993.

[9] G. Haase and U. Langer. On the use of multigrid preconditioners in the domain decomposition method. In W. Hackbusch, editor, *Parallel Algorithms for PDEs*, pages 101–110, Braunschweig, 1990. Vieweg. Proc. of the 6th GAMM–Seminar, Kiel, 1990.

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

[11] G. Haase and U. Langer. Domain decomposition vs. adaptivity. In *Proceedings of the Converence on the Finite Element Method: Fifty Years of the Courant Element.* Marcel Dekker Publ. Inc., 1993. 30.8 - 3.9.93, appears.

[12] G. Haase, U. Langer, and A. Meyer. A new approach to the Dirichlet domain decomposition method. In S. Hengst, editor, *Proceedings of the "5-th Multigrid Seminar" held at Eberswalde, GDR, May 14-18, 1990*, pages 1–59, Berlin, 1990. Academy of Science. Report-Nr. R-MATH-09/90.

[13] G. Haase, U. Langer, and A. Meyer. 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), 1991.

[14] G. Haase, U. Langer, and A. Meyer. Domain decomposition preconditioners with inexact subdomain solvers. *J. of Num. Lin. Alg. with Appl.*, 1:27–42, 1992.

[15] G. Haase, U. Langer, and A. Meyer. Parallelisierung und Vorkonditionierung des CG-Verfahrens durch Gebietszerlegung. In *Parallele Algorithmen auf Transputersystemen*, Teubner-Scripten zur Numerik III, Stuttgart, 1992. Teubner. Tagungsbericht der GAMM-Tagung, 31. Mai- 1. Juni 1991, Heidelberg.

[16] A. M. Matsokin and S. V. Nepomnyaschikh. On the convergence of the non-overlapping schwarz subdomain alternating method. In *Metody Approksimatsii i Interpolyatsii (ed. Yu.A.Kuznetsov)*, pages 85–97, Novosibirsk, 1981. Comp. Cent. Sib. Branch, USSR Acad. Sci. (in Russian).

[17] A. M. Matsokin and S. V. Nepomnyaschikh. On using the bordering method for solving systems of mesh equations. In *Vychislitelnye Algoritmy v Zadachakh Mathematicheskoy Fiziki (ed. V.V.Smelov)*, pages 99–109, Novosibirsk, 1983. Comp. Cent. Sib. Branch, USSR Acad. Sci. (in Russian).

[18] A. M. Matsokin and S. V. Nepomnyaschikh. A Schwarz alternating method in a subspace. *Soviet Mathemathics*, 29(10):78–84, 1985.

[19] A. M. Matsokin and S. V. Nepomnyaschikh. Norms in the space of traces of mesh functions. *Sov. J. Numer. Anal. Math. Modelling*, 3:199–216, 1988.

[20] A. Meyer. A parallel preconditioned conjugate gradient method using domain decomposition and inexact solvers on each subdomain. *Computing*, 45:217–234, 1990.

[21] S. Nepomnyaschikh. Mesh theorems on traces, normalization of function traces and and their inversion. *Sov. J. Numer. Anal. Math. Modelling*, 6(3):223–242, 1991.

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

[23] A. A. Oganesyan and L. A. Rukhovets. *Variational–difference methods for the solution of elliptic equations.* Izd. Akad. Nauk Armianskoi SSR House, Erevan, 1979. (in Russian).

[24] G. Wittum. *Filternde Zerlegungen.* Teubner, Stuttgart, 1992.

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

[26] H. Yserentant. Two preconditioners based on the multi-level splitting of finite element spaces. *Numer. Math.*, 58:163–184, 1990.