Technische Universität Chemnitz-Zwickau

DFG-Forschergruppe "SPC" · Fakultät für Mathematik

Holger Matthes

Parallel Preconditioners for Plate Problems

Abstract

This paper concerns the solution of plate bending problems in domains composed of rectangles. Domain decomposition (DD) is the basic tool used for both the parallelization of the conjugate gradient method and the construction of efficient parallel preconditioners. A so-called Dirichlet DD preconditioner for systems of linear equations arising from the finite element approximation by non-conforming Adini elements is derived. It is based on the non-overlapping DD, a multilevel preconditioner for the Schur-complement and a fast, almost direct solution method for the Dirichlet problem in rectangular domains based on fast Fourier transform. Making use of Xu's theory of the auxiliary space method we construct an optimal preconditioner for plate problems discretized by conforming Bogner-Fox-Schmidt rectangles.

Results of numerical experiments carried out on a multiprocessor system are given. For the test problems considered the number of iterations is bounded independent of the mesh sizes <u>and</u> independent of the number of subdomains. The resulting parallel preconditioned conjugate gradient method requires $O(h^{-2}\ln h^{-1}\ln\varepsilon^{-1})$ arithmetical operations per processor in order to solve the finite element equations with the relative accuracy ε.

Preprint-Reihe der Chemnitzer DFG-Forschergruppe "Scientific Parallel Computing"

SPC 95_31 **November 1995**

Contents

Authors' address:

Holger Matthes
TU Chemnitz-Zwickau
Fakultät für Mathematik
D-09107 Chemnitz
Germany

e-mail: hmatthes@mathematik.tu-chemnitz.de

1 Introduction

This paper concerns the solution of bending problems for thin elastic plates. The Kirchhoff plate theory formulates boundary value problems for fourth-order partial differential equations. Our topic is the construction of efficient iterative methods for the solution of finite element equations arising from the discretization of these plate problems by rectangular finite elements. To take advantage of modern parallel computing environments it is insufficient to transfer commonly used *sequential* algorithms. Instead, we need *parallel* algorithms adapted to the specific classes of parallel machines. For the solution of boundary value problems approximated by finite element schemes domain decomposition (DD) methods are of increasing interest. They are basic tools for both the parallelization of iterative methods and the construction of efficient parallel preconditioners. For instance in [11] - [15] the parallelization of the conjugate gradient (CG) method on the basis of a non-overlapping DD approach and aspects of the algorithmic realization have been studied in detail. The theory of the DD-preconditioner has been derived on a purely algebraic basis. We briefly review some of the main results which are valid in our context too.

Three components of the DD-preconditioner have to be chosen in order to adapt it as well as possible to the boundary value problem considered. A preconditioner for the subdomain problems (in many cases the homogeneous Dirichlet problem) and a basis transformation algorithm extending the data at the coupling boundary to the inner nodes biharmonically have to be constructed. That's why in section 3 we generalize the so-called *boundary potential method* developed in [19] and [20] to the case of non-homogeneous boundary conditions. It allows us to solve the clamped plate problem in rectangular domains with almost optimal arithmetical effort of $O(N \ln N)$ operations, where N denotes the number of unknowns.

The third component is a multilevel Schur-complement preconditioner for conforming Bogner-Fox-Schmidt elements proposed in [24]. The auxiliary space method (see [28]) allows us to make use of this preconditioner also in the case of the non-conforming Adini elements (section 5). In section 6 the role of both discretizations is switched. There a parallel preconditioner for the conforming discretization is constructed using Adini elements to define the auxiliary finite element space.

The parallel preconditioned CG (ParPCG) method is well suited for the implementation on multiprocessor systems with local memory and message passing principle. In section 7 the results of numerical tests are presented. They confirm the theoretical results and show that the preconditioners developed are applicable to plate problems with various types of boundary conditions.

2 Plate problems, finite elements and domain decomposition

2.1 Plate problems

Let us consider the homogeneous Dirichlet problem (1) for the biharmonic operator $\Delta^2 = \partial_1^4 + 2 \cdot \partial_{12}^2 + \partial_2^4$ as a prototype of an elliptic, fourth-order, partial differential equation.

$$\begin{aligned} \Delta^2 u(x) &= f(x), \quad x \in \Omega, & (1)\\ u(x) &= 0, \quad x \in \Gamma = \partial\Omega, \\ \partial_\nu u(x) &= 0, \quad x \in \Gamma = \partial\Omega. \end{aligned}$$

It can be interpreted as the classical formulation of the clamped plate problem (2) concerning the equilibrium position of a originally plane, thin plate of constant thickness under the action of a transverse force. Its solution minimizes the total potential energy J in V_{00} (see [1] - [3]).

$$\begin{aligned} u \in V_{00} : J(u) &= \min_{v \in V_{00}} J(v), & (2)\\ V_{00} &= H_0^2(\Omega) = \left\{ v \in H^2(\Omega) : v = \partial_\nu v = 0 \text{ on } \Gamma = \partial\Omega \right\}, \\ J(v) &= \frac{1}{2} a(v,v) - f(v), & (3)\\ \text{where} & \\ a(u,v) &= \int_\Omega \{\Delta u \Delta v + (1-\sigma)(2\partial_{12}u\partial_{12}v - \partial_{11}u\partial_{22}v - \partial_{22}u\partial_{11}v)\}\, dx, \\ f(v) &= \int_\Omega f v dx, \quad f \in L^2(\Omega). \end{aligned}$$

For physical reasons the Poisson coefficient σ of the plate material lies in $(0, \frac{1}{2})$. As we will see later, the boundary potential method reduces the numerical solution of the clamped plate problem in a rectangle to the solution of the simply supported plate problem (4).

$$\begin{aligned} \bar{u} \in V_0 : J(\bar{u}) &= \min_{v \in V_0} J(v), & (4)\\ V_0 &= H^2(\Omega) \cap H_0^1(\Omega) = \left\{ v \in H^2(\Omega) : v = 0 \text{ on } \Gamma = \partial\Omega \right\}. \end{aligned}$$

2.2 Finite element discretization

Let Ω composed of a finite number of rectangles, such that it may be covered by a triangulation $\mathcal{R}_h$ made of rectangles R. Let $\mathcal{N}_h$ be the set of nodes of $\mathcal{R}_h$.

Using non-conforming Adini elements we define the finite element space $\tilde{V}_h$.

$$\tilde{V}_h = \left\{ v_h \in C^0(\bar{\Omega}) : v_h|_R \in \tilde{Q}_3(R) \quad \forall R \in \mathcal{R}_h \right\}, \tag{5}$$

$$\tilde{Q}_3(R) = \left\{ p(x) = \sum_{k_1+k_2 \leq 3} \gamma_{k_1 k_2} x_1^{k_1} x_2^{k_2} + \gamma_{13} x_1 x_2^3 + \gamma_{31} x_1^3 x_2 \right\}.$$

Using the usual finite element nodal basis $\tilde{\Phi}_h = \left\{ \tilde{\varphi}_{(\alpha,P)} \right\}$, $(\alpha, P) \in \tilde{\tau} \times \mathcal{N}_h$, $\tilde{\tau} = \{(0,0),(1,0),(0,1)\} = \{\mathbf{0}, e_1, e_2\}$, the functions $v_h = \tilde{\Phi}_h \underline{v} \in \tilde{V}_h$ are uniquely defined by the vector $\underline{v}$ of the nodal values $v_{(\alpha,P)}$.

$$\partial_1^{\alpha_1} \partial_2^{\alpha_2} v_h|_P = v_{(\alpha,P)}, \quad (\alpha, P) \in \tilde{\tau} \times \mathcal{N}_h. \tag{6}$$

Approximate solutions u_h for (2) and $\bar{u}_h$ for (4) minimize the discrete energy functional $\tilde{J}_h$ in $\tilde{V}_{00,h}$ respectively $\tilde{V}_{0,h}$.

$$\begin{aligned}
\tilde{V}_{00,h} &= \left\{ v_h \in \tilde{V}_h : v_h(P) = \partial_\tau v_h(P) = \partial_\nu v_h(P) = 0, \quad \forall P \in \mathcal{N}_h \cap \Gamma \right\}, \\
\tilde{V}_{0,h} &= \left\{ v_h \in \tilde{V}_h : v_h(P) = \partial_\tau v_h(P) = 0, \quad \forall P \in \mathcal{N}_h \cap \Gamma \right\}, \\
\tilde{J}_h(v_h) &= \frac{1}{2} \tilde{a}_h(v_h, v_h) - f(v_h), \\
\tilde{a}_h(u_h, v_h) &= \sum_{R \in \mathcal{R}_h} \int_R \{ \Delta u_h \Delta v_h + \\
&\qquad (1-\sigma)\left(2\partial_{12} u_h \partial_{12} v_h - \partial_{11} u_h \partial_{22} v_h - \partial_{22} u_h \partial_{11} v_h\right)\} \, dx.
\end{aligned}$$

The minimum condition

$$\frac{\partial}{\partial u_{(\alpha,P)}} \tilde{J}_h(u_h) = 0, \quad \forall\, (\alpha, P) \in \tilde{\tau} \times \mathcal{N}_h,$$

leads us to systems of linear equations determining the associated nodal vectors $\underline{u}$ and $\underline{\bar{u}}$.

$$\begin{aligned}
\tilde{L} \underline{u} &= \underline{f}, \\
\bar{L} \underline{\bar{u}} &= \underline{\bar{f}}.
\end{aligned}$$

The discretization of plate problems using Adini rectangles is a non-conforming finite element method in the sense that $\tilde{V}_h$ is not a subspace of $H^2(\Omega)$. However, the following inclusions are valid

$$\tilde{V}_{00,h} \subset \tilde{V}_{0,h} \subset H_0^1(\Omega) \cap C^0(\bar{\Omega}), \quad v_h|_R \in H^2(R), \quad \forall R \in \mathcal{R}_h, \quad \forall v_h \in \tilde{V}_h.$$

Functions $v_h \in \tilde{V}_{00,h}$ satisfy the Dirichlet boundary condition for the normal derivatives at all boundary nodes $P \in \mathcal{N}_h \cap \partial\Omega$. For more details and error estimates see e.g. [2] and [3].

A conforming finite element discretization can be constructed using rectangular, bicubic Bogner-Fox-Schmidt elements.

$$\begin{aligned} V_h &= \left\{ v_h \in C^1(\bar{\Omega}) : v_h|_R \in Q_3(R), \quad \forall R \in \mathcal{R}_h \right\}, \qquad (7) \\ Q_3(R) &= \left\{ p(x) = \sum_{k_1,k_2 \leq 3} \gamma_{k_1 k_2} x_1^{k_1} x_2^{k_2} \right\}. \end{aligned}$$

The finite element functions $v_h \in V_h$ are uniquely defined by their nodal values, gradients and mixed derivatives, i.e. $\tau = \{(0,0),(1,0),(0,1),(1,1)\} = \{\mathbf{0}, e_1, e_2, \mathbf{1}\}$.

2.3 Non-overlapping Domain Decomposition

We consider a plate problem in a domain Ω composed of p rectangles $\bar{\Omega} = \cup_{s=1}^{p} \bar{\Pi}_s$ discretized either by Adini or Bogner-Fox-Schmidt elements.

Let index "C" correspond to those degrees of freedom belonging to the coupling boundary $\Gamma_C = \cup_{s=1}^{p} \partial\Pi_s$ which are not given by an essential boundary condition. Let "I" correspond the interior $\Omega_I = \cup_{s=1}^{p} \Pi_s$ of the subdomains. Ordering the nodal basis functions by these two groups, the resulting system of finite element equations $L\underline{u} = \underline{f}$ can be written in the block form

$$\begin{pmatrix} L_C & L_{CI} \\ L_{IC} & L_I \end{pmatrix} \begin{pmatrix} \underline{u}_C \\ \underline{u}_I \end{pmatrix} = \begin{pmatrix} \underline{f}_C \\ \underline{f}_I \end{pmatrix}. \qquad (8)$$

There L_I is the block-diagonal matrix which consists of the stiffness matrices related to the clamped plate problems in Π_s.

$$L_I = \mathrm{blockdiag}\,(L_{I,s})_{s=1,2,\ldots,p}\,.$$

We solve (8) using the parallel preconditioned conjugate gradient (ParPCG) method. It is well suited for the implementation on multiprocessor systems with local memory and message passing principle. An advantage of DD parallelized iterative methods is the low demand of communication between processors. Details about the parallelization and implementation of the ParPCG method are described for instance in [10], [11] and [15].

The preconditioning step in the ParPCG method should fit into the parallelization concept of the non-overlapping DD. The idea of preconditioning is to transform the original problem $L\underline{u} = \underline{f}$ in a equivalent one $C^{-1}L\underline{u} = C^{-1}\underline{f}$ in order to obtain a significantly lower condition number $\kappa(C^{-1}L) \ll \kappa(L)$. A preconditioner C must have the following (sometimes contrary) properties:

1. C is symmetric and positive definite,

$$C = C^T, \quad (C\underline{x}, \underline{x}) > 0 \quad \forall \underline{x} \in \mathbb{R}^N \setminus \{0\}.$$

2. C is spectrally equivalent to L,

$$\exists \overline{\gamma}, \underline{\gamma} > 0 : \underline{\gamma}(C\underline{x}, \underline{x}) \leq (L\underline{x}, \underline{x}) \leq \overline{\gamma}(C\underline{x}, \underline{x}) \quad \forall \underline{x} \in \mathbb{R}^N,$$

   where $\kappa(C^{-1}L) \leq \frac{\overline{\gamma}}{\underline{\gamma}} \longrightarrow \min$.

3. The preconditioning system

$$C\underline{w} = \underline{r}$$

   can be solved much more efficiently than the original problem $L\underline{u} = \underline{f}$. Additionally we demand a high degree of parallelism in the preconditioning algorithm.

Considering the well known factorization of the matrix L from (8)

$$L = \begin{pmatrix} I_C & L_{CI}L_I^{-1} \\ 0 & I_I \end{pmatrix} \begin{pmatrix} S_C & 0 \\ 0 & L_I \end{pmatrix} \begin{pmatrix} I_C & 0 \\ L_I^{-1}L_{IC} & I_I \end{pmatrix}, \tag{9}$$

where $S_C = L_C - L_{CI}L_I^{-1}L_{IC}$ is the Schur-complement, the so-called ASM-DD-preconditioner was derived on a purely algebraic basis.

$$C = \begin{pmatrix} I_C & L_{CI}B_I^{-T} \\ 0 & I_I \end{pmatrix} \begin{pmatrix} C_C & 0 \\ 0 & C_I \end{pmatrix} \begin{pmatrix} I_C & 0 \\ B_I^{-1}L_{IC} & I_I \end{pmatrix}. \tag{10}$$

The three components C_C, C_I and B_I have to be chosen in order to adapt the preconditioner as well as possible to the problem under consideration (partial differential equation, discretization method, etc.). The following theorem proved in [10], [12] gives general criteria for the choice of the components of (10).

Theorem 2.1 *Let the symmetric and positive definite block preconditioners C_C and C_I satisfy the spectral equivalence inequalities*

$$\exists \overline{\gamma}_I, \underline{\gamma}_I > 0 \quad : \quad \underline{\gamma}_I C_I \leq L_I \leq \overline{\gamma}_I C_I, \tag{11}$$

$$\exists \overline{\gamma}_C, \underline{\gamma}_C > 0 \quad : \quad \underline{\gamma}_C C_C \leq S_C + T_C \leq \overline{\gamma}_C C_C, \tag{12}$$

where $S_C + T_C$ is the modified Schur-complement, $S_C = L_C - L_{CI}L_I^{-1}L_{IC}$, $T_C = L_{CI}\left(L_I^{-1} - B_I^{-T}\right)L_I\left(L_I^{-1} - B_I^{-1}\right)L_{IC}$. Let B_I nonsingular.
Then the ASM-DD preconditioner (10) satisfies the spectral equivalence inequalities

$$\underline{\gamma}\left(1 - \sqrt{\frac{\mu}{1+\mu}}\right) C \leq L \leq \overline{\gamma}\left(1 - \sqrt{\frac{\mu}{1+\mu}}\right) C. \tag{13}$$

The condition number $\kappa(C^{-1}L)$ satisfies the estimates

$$\left(\underline{\gamma}/\overline{\gamma}\right)\left(\sqrt{\mu} + \sqrt{1+\mu}\right)^2 \leq \kappa(C^{-1}L) \leq \left(\overline{\gamma}/\underline{\gamma}\right)\left(\sqrt{\mu} + \sqrt{1+\mu}\right)^2, \tag{14}$$

where $\overline{\gamma} = \max\{\overline{\gamma}_I, \overline{\gamma}_C\}$, $\underline{\gamma} = \min\left\{\underline{\gamma}_I, \underline{\gamma}_C\right\}$ and $\mu = \rho\left(S_C^{-1}T_C\right)$ denotes the spectral radius of $S_C^{-1}T_C$. ■

Because of the obvious spectral equivalence inequalities

$$S_C \le S_C + T_C \le (1+\mu) S_C,$$

we can use a Schur-complement preconditioner to define C_C (section 5).

3 Fast solution methods in a rectangle

In [19] and [20] a fast solution method for the biharmonic Dirichlet problem (1) in rectangular domains, the so-called *boundary potential method* has been proposed. It was developed for a special five-point variational finite difference scheme, but the method can be applied to the clamped plate problem discretized by Adini element in principle. We briefly describe this method, but in view of its use in the domain decomposition context we have to generalize it to the case of non-homogeneous Dirichlet boundary conditions.

3.1 Statement of the problem

We consider the inhomogeneous clamped plate problem in $\Pi = (0, l_1) \times (0, l_2)$

$$\begin{aligned} u \in V_g : J(u) &= \min_{v \in V_g} J(v), \\ V_g &= \left\{ v \in H^2(\Pi) : v = g_D, \partial_\nu v = g_N \text{ on } \Gamma = \partial\Pi \right\}. \end{aligned} \tag{15}$$

Let it discretized by uniform Adini [1] elements of size $h_k = \frac{l_k}{N_k}$, $N_k = 2^{n_k}$ in x_k-direction $k = 1, 2$. The node $P = (i_1 h_1, i_2 h_2)$ is uniquely defined by the index $i = (i_1, i_2), 0 \le i_1 \le N_1, 0 \le i_2 \le N_2$. Further we distinguish several sub-vectors of the coefficient vector $\underline{u} \in \mathbb{R}^{3(N_1+1)(N_2+1)}$.

- Nodal values at inner nodes of Π:

$$\begin{aligned} \underline{u}_I &= \underline{u}|_{\omega_I} \in \mathbb{R}^{N_I}, \\ \omega_I &= \{(\alpha, i) : \alpha \in \tau, 0 < i_k < N_k, k = 1, 2\}, \end{aligned}$$

- nodal values of the function and tangential derivatives on $\partial\Pi$:

$$\begin{aligned} \underline{u}_D &= \underline{u}|_{\gamma_D} \in \mathbb{R}^{N_D}, \\ \gamma_D &= \{(\alpha, i) : \alpha \in \{o, e_{3-k}\} \wedge i_k \in \{0, N_k\} \wedge 0 < i_{3-k} < N_{3-k}, k = 1, 2\}, \end{aligned}$$

[1] The tilde symbol on matrices and vectors will be omitted in this section.

- nodal values of normal derivatives on $\partial\Pi$:

$$\begin{aligned} \underline{u}_N &= \underline{u}|_{\gamma_N} \in \mathbb{R}^{N_N}, \\ \gamma_N &= \{(\alpha, i) : \alpha = e_k \wedge i_k \in \{0, N_k\} \wedge 0 < i_{3-k} < N_{3-k}, k = 1, 2\} \end{aligned}$$

  and

- $\underline{u}_F = \underline{u}|_{\omega_F} \in \mathbb{R}^{N_F}$, $\omega_F = \omega_I \cup \gamma_N$.

After the homogenization of the essential boundary conditions on γ_D the discretized boundary value problem (15) looks like

$$\underline{u}_I \in \mathbb{R}^{N_I}, \underline{u}_N \in \mathbb{R}^{N_N} : \begin{pmatrix} L_I & L_{IN} \\ 0 & I_N \end{pmatrix} \begin{pmatrix} \underline{u}_I \\ \underline{u}_N \end{pmatrix} = \begin{pmatrix} \underline{f}_I - L_{ID}\underline{g}_D \\ \underline{g}_N \end{pmatrix}. \tag{16}$$

The corresponding simply supported plate problem has the homogenized form

$$\underline{\bar{u}}_F \in \mathbb{R}^{N_F} : \bar{L}_F \underline{\bar{u}}_F = \begin{pmatrix} L_I & L_{IN} \\ L_{NI} & L_N \end{pmatrix} \begin{pmatrix} \underline{\bar{u}}_I \\ \underline{\bar{u}}_N \end{pmatrix} = \begin{pmatrix} \underline{f}_I - L_{ID}\underline{g}_D \\ -L_{ND}\underline{g}_D \end{pmatrix}. \tag{17}$$

The solution $\underline{\bar{u}}$ of (17) is easy to obtain via the Fourier method. It satisfies the Dirichlet boundary conditions on γ_D and the first row of equation (16). In contrast the direct solution of the clamped plate problem via the Fourier method is not possible because the equations do not decouple. The main idea of the boundary potential method is to find some correction $\underline{v}$, such that $\underline{u} = \underline{\bar{u}} + \underline{v}$ satisfies (16). At least this leads us to an additional system of linear equations which can be solved efficiently using the Jacobi iterative method.

3.2 The Fourier method

We look for the solution $\underline{\bar{u}}_F$ of the discrete boundary value problem (17) in $\mathbb{R}^{N_F}$. For $\mathbb{R}^{N_F}$ we introduce the unitary basis M.

$$\begin{aligned} M &= \left(\mu_{(\alpha,i)(\beta,j)}\right)_{\omega_F \times \omega_F}, \qquad (18) \\ \mu_{(\alpha,i)(\beta,j)} &= \delta_{\alpha\beta} \cdot \xi_{i_1}(\alpha_1, j_1) \cdot \xi_{i_2}(\alpha_2, j_2), \\ \xi_{i_k}(0, j_k) &= \sqrt{\frac{2}{N_k}} \sin \frac{i_k j_k \pi}{N_k}, \\ \xi_{i_k}(1, j_k) &= \sqrt{\frac{2}{N_k}\kappa_k(i_k)} \cos \frac{i_k j_k \pi}{N_k}, \kappa_k(i_k) = \begin{cases} \frac{1}{2} & : \quad i_k = 0, N_k \\ 1 & : \quad 0 < i_k < N_k \end{cases}. \end{aligned}$$

In terms of the basis M matrix $\bar{L}_F$ is block-diagonal. Thus we have

$$\bar{L}_F^{-1} = M\Lambda^{-1}M^T \tag{19}$$

$$
\begin{aligned}
&= M \quad \mathrm{blockdiag}(\Lambda_j^{-1}) M^T, \\
\Lambda_j &= \begin{cases} \begin{pmatrix} \lambda_{j,o,o} & \lambda_{j,o,e_1} & \lambda_{j,o,e_2} \\ \lambda_{j,o,e_1} & \lambda_{j,e_1,e_1} & 0 \\ \lambda_{j,o,e_2} & 0 & \lambda_{j,e_2,e_2} \end{pmatrix} & : \quad \tau \times j \subset \omega_I, \\ \lambda_{j,\alpha,\alpha} & : \quad (\alpha, j) \in \gamma_N, \end{cases} \\
\lambda_{j,o,o} &= \sum_{k=1,2} \frac{8h_{3-k}}{h_k^3}(1-c_k)(2+c_{3-k}) + \frac{8\,(7-2\sigma)}{5h_1h_2}(1-c_1)(1-c_2), \\
\lambda_{j,o,e_k} &= -4s_k \left(\frac{h_{3-k}}{h_k^2}(2+c_{3-k}) + \frac{(1-\sigma)}{5h_{3-k}}(1-c_{3-k}) \right), \\
\lambda_{j,e_k,e_k} &= \frac{4h_{3-k}}{3h_k}(2+c_k)(2+c_{3-k}) + \frac{4\,(1-\sigma)\,h_k}{15h_{3-k}}(4-c_k)(1-c_{3-k}), \\
c_k &= \cos\frac{j_k\pi}{N_k}, \quad s_k \;=\; \sin\frac{j_k\pi}{N_k}.
\end{aligned}
$$

Multiplications of vectors from $\mathbb{R}^{N_F}$ by M respectively M^T via fast Fourier transform [5] demand $O(N_1N_2\ln(N_1N_2))$ arithmetical operations. The solution of $\frac{1}{3}N_I = (N_1-1)(N_2-1)$ systems of three linear equations and $N_N = 2(N_1+N_2-2)$ single linear equations demands $O(N_1N_2)$ operations. So we have a suboptimal solution method for the non-homogeneous boundary value problem (17) in a rectangle.

3.3 The boundary potential method

Let $\underline{\bar{u}}$ the solution of (17). In order to find the solution $\underline{u}$ of (16) we consider the difference $\underline{v} = \underline{u} - \underline{\bar{u}}$. Therefore we have the following boundary conditions on $\underline{v}$:

$$
\begin{aligned}
\underline{v}_D &= 0_D, && (20) \\
\underline{v}_N &= \underline{g}_N - \underline{\bar{u}}_N. && (21)
\end{aligned}
$$

From the identity between the first rows of (16) and (17) we get

$$
\bar{L}_F \underline{v}_F = \begin{pmatrix} 0_I \\ \underline{b}_N \end{pmatrix}, \qquad (22)
$$

where the part $\underline{b}_N$ of the right-hand side is unknown at the moment. However, we represent $\underline{b}_N$ in terms of the basis N of $\mathbb{R}^{N_N}$.

$$
\begin{aligned}
N &= \left(\nu_{(\alpha,i)(\beta,j)} \right)_{\gamma_N \times \gamma_N}, && (23) \\
\nu_{(\alpha,i)(\beta,j)} &= \delta_{\alpha\beta} \cdot \begin{cases} \xi_{i_{3-k}}(0, j_{3-k}) & : \quad (i_k, j_k) \in \{0, N_k\} \times \{0, N_k\} \setminus (N_k, N_k), \\ -\xi_{i_{3-k}}(0, j_{3-k}) & : \quad i_k = j_k = N_k. \end{cases}
\end{aligned}
$$

We have

$$\underline{b}_N = N\underline{b}_\nu \text{ respectively } \begin{pmatrix} 0_I \\ \underline{b}_N \end{pmatrix} = \hat{N}\underline{b}_\nu, \tag{24}$$

where $\hat{N}$ denotes the extension of N by zero for the rows corresponding to ω_I. In order to find the coefficient vector $\underline{b}_\nu \in \mathbb{R}^{N_N}$ we pay attention to the boundary condition (21). Multiplying by N^T we get

$$\begin{aligned} N^T\underline{v}_N &= N^T\underline{g}_N - N^T\underline{\bar{u}}_N \\ \text{respectively} \quad \hat{N}^T\underline{v}_F &= N^T\underline{g}_N - \hat{N}^T\underline{\bar{u}}_F. \end{aligned} \tag{25}$$

Using (22), (24) and (25) the coefficient vector $\underline{b}_\nu$ is the solution of a system of linear equations.

$$\begin{aligned} B\underline{b}_\nu &= \underline{d}, \\ \text{where} \quad B &= \hat{N}^T\bar{L}_F^{-1}\hat{N}, \\ \underline{d} &= N^T\underline{g}_N - \hat{N}^T\underline{\bar{u}}_F. \end{aligned} \tag{26}$$

Using (22), (24) and (26) the solution $\underline{u}_I$ of (16) has the form

$$\underline{\bar{f}}_F = \begin{pmatrix} \underline{f}_I - L_{ID}\underline{g}_D \\ -L_{ND}\underline{g}_D \end{pmatrix}, \tag{27}$$

$$\underline{u}_I = \left[\bar{L}_F^{-1}\underline{\bar{f}}_F + \bar{L}_F^{-1}\hat{N}B^{-1}\left(N^T\underline{g}_N - \hat{N}^T\bar{L}_F^{-1}\underline{\bar{f}}_F\right)\right]_{\omega_I}. \tag{28}$$

3.4 The algorithmic realization

From (19), denoting $N_\mu = M^T\hat{N}$, we get

$$\underline{u}_I = \left[M\left(\Lambda^{-1}M^T\underline{\bar{f}}_F + \Lambda^{-1}N_\mu B^{-1}\left(N^T\underline{g}_N - N_\mu^T\Lambda^{-1}M^T\underline{\bar{f}}_F\right)\right)\right]_{\omega_I}. \tag{29}$$

Thus, the algorithm of the boundary potential method consists of the following steps.

1. Homogenize the Dirichlet data on γ_D according to (27).
2. Represent the solution $\underline{\bar{u}}$ of the auxiliary problem (17) in terms of its Fourier coefficients
$$\underline{\bar{u}}_\mu = \Lambda^{-1}M^T\underline{\bar{f}}_F.$$
3. Represent the Dirichlet data on γ_N in terms of the basis N.
$$\underline{g}_\nu = N^T\underline{g}_N.$$

4. Generate system (26).

$$\begin{aligned} B &= N_\mu^T \Lambda^{-1} N_\mu, \\ \underline{d} &= \underline{g}_\nu - N_\mu^T \underline{\bar{u}}_\mu. \end{aligned}$$

5. Solve system (26).

6. Represent the solution $\underline{u}$ of (16) in terms of the Fourier coefficients $\underline{u}_\mu$.

$$\underline{u}_\mu = \underline{\bar{u}}_\mu + \Lambda^{-1} N_\mu \underline{b}_\nu$$

7. Synthesize $\underline{u}$.

$$\underline{u}_I = \left[M \underline{u}_\mu\right]_{\omega_I}$$

- Steps 1,2 and 7 are known from the Fourier method solving (17) and demand $O(N_1 N_2 \ln(N_1 N_2))$ arithmetical operations.

- At least step 3 demands $O(N_1 \ln N_1) + O(N_2 \ln N_2)$ arithmetical operations to perform the fast Fourier transform for two vectors of length $N_1 - 1$ and two vectors of length $N_2 - 1$.

- The matrix $N_\mu = \left(n_{(\alpha,i)(\beta,j)}\right)_{\omega_F \times \gamma_N}$ can be described more explicitly. We have

$$\begin{aligned} n_{(\alpha,i)(\beta,j)} &= \sum_{(\gamma,k)\in\omega_F} \mu_{(\gamma,k)(\alpha,i)} \nu_{(\gamma,k)(\beta,j)} \\ &= \begin{cases} 2\sqrt{\frac{1}{N_l}}\delta_{\alpha\beta\gamma e_l}\delta_{i_{3-l} j_{3-l}} & : \quad \frac{j_l}{N_l} + i_l \text{ even}, j_l = 0, N_l, l = 1,2, \\ 0 & : \quad \text{else.} \end{cases} \end{aligned}$$

  Thus, steps 4 and 6 demand $O(N_1 N_2)$ operations.

- System (26) can be solved via the Jacobi method (30) where $D = \text{diag} B$.

$$\begin{aligned} \underline{b}_\nu^{(k+1)} &= \underline{b}_\nu^{(k)} + D^{-1}\left(\underline{d} - B \underline{b}_\nu^{(k)}\right), \\ \underline{b}_\nu^{(0)} &= 0. \end{aligned} \tag{30}$$

  Considering the structure of the matrix B we see

$$B_{(\alpha,i)(\beta,j)} = 0 \text{ if } (\alpha = \beta) \wedge (i \neq j)$$

  and therefor one iteration step demands $O(N_1 N_2)$ operations.

The matrix B is independent of h_1 and h_2 diagonal dominant, that means for the iteration operator $S = (I - D^{-1}B)$ of (30)

$$\|S\|_\infty = \max_{(\alpha,i)} \left| \frac{\sum_{(\beta,j)\in\gamma_N\setminus(\alpha,i)} B_{(\alpha,i)(\beta,j)}}{B_{(\alpha,i)(\alpha,i)}} \right| \le c_0 < 1.$$

In the case of Adini finite element schemes we get $c_0 = 0.7$. Thus, in order to obtain an approximate solution $\underline{b}_\nu^{(n)}$ of the relative accuracy ε

$$\left\| \underline{b}_\nu - \underline{b}_\nu^{(n)} \right\|_B \le \varepsilon \left\| \underline{b}_\nu \right\|_B ,$$

we have to perform $n \ge I(\varepsilon) = \frac{\ln \varepsilon^{-1}}{\ln c_0}$ iteration steps (30). From

$$(B\underline{b}_\nu, \underline{b}_\nu) = (\hat{N}^T \bar{L}_F^{-1} \hat{N} \underline{b}_\nu, \underline{b}_\nu) = (\bar{L}_F \underline{v}_F, \underline{v}_F)$$

for the approximate solution $\underline{u}_F^{(n)}$ corresponding to $\underline{b}_\nu^{(n)}$ follows

$$\left\| \begin{pmatrix} \underline{u}_I - \underline{u}_I^{(n)} \\ \underline{g}_N - \underline{u}_N^{(n)} \end{pmatrix} \right\|_{\bar{L}_F} \le \varepsilon \left\| \underline{v}_F \right\|_{\bar{L}_F} .$$

The above facts show that the solution of (16) using the boundary potential method with relative accuracy ε demands $O(N_1 N_2 \ln(N_1 N_2) \ln \varepsilon^{-1})$ arithmetical operations. The asymptotic behavior of the arithmetical cost is dominated by the fast Fourier transform in steps 1 and 7.

3.5 An example

The following example illustrates the theoretical results concerning the boundary potential method. The purely sequential computations where performed on a *single* processor PowerPC-601 of a multiprocessor system. Later it will be a subroutine of a parallel code.

We consider the clamped plate problem in the unit square $\Pi = (0,1) \times (0,1)$. Choosing the right-hand side f, the solution u is known.

$$\begin{aligned} f(x,y) &= 6144 \cdot \left(x^2(1-x)^2 + y^2(1-y)^2 \right) \\ &\quad +2048 \cdot (6x^2 - 6x + 1) \cdot (6y^2 - 6y + 1), \\ u(x,y) &= 256 \cdot x^2(1-x)^2 y^2 (1-y)^2. \end{aligned} \tag{31}$$

The inner iteration was stopped at the relative accuracy

$$\frac{\|d - Bb_\nu^n\|}{\|d\|} < \varepsilon = 10^{-6}.$$

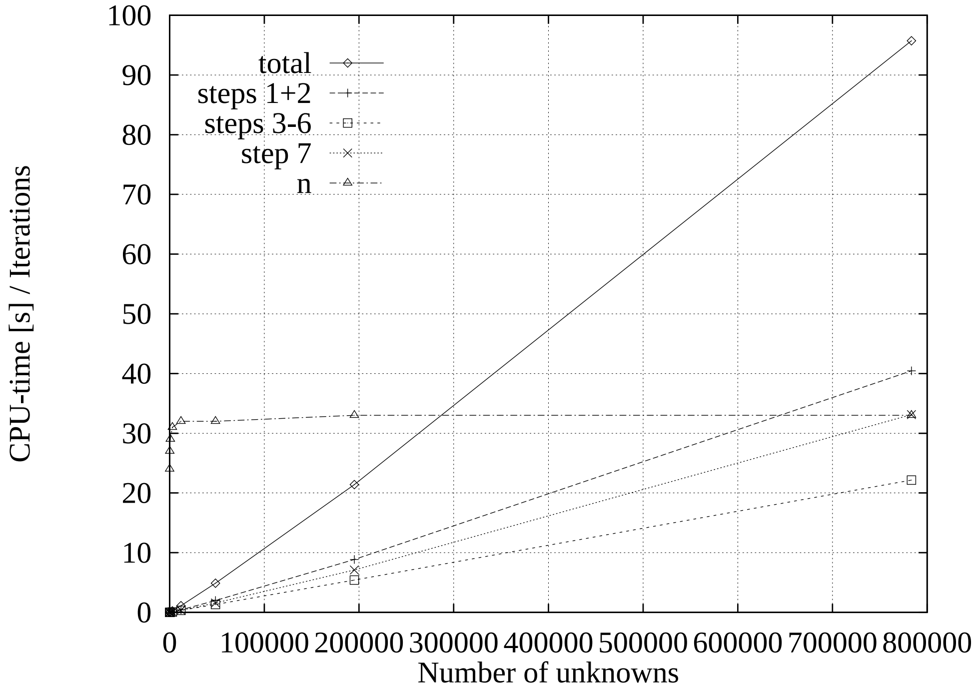

Figure 1: Arithmetical cost of the boundary potential method.

If the mesh size decreases the number n of inner iterations approaches a constant value.

In comparison to the CPU-time required to solve the auxiliary problem using the Fourier method (steps 1,2,7), the CPU-time for the additional steps 3 -6 of the boundary potential method increases slower. Especially the asymptotic behavior of the total CPU-time is not changed for the worse by these additional steps (see figure 1). The finest discretization considered is a 512×512-mesh.

4 The boundary potential method in DD preconditioning

This section concerns the choice of the preconditioner C_I for the local subproblems and the basis transformation in the ASM-DD-preconditioner for non-conforming Adini element discretizations[2].

In view of the spectral equivalence constants $\underline{\gamma}_I$ and $\overline{\gamma}_I$ (c.f. (11)) the choice $C_I = L_I$ would be optimal. But, if the mesh size tends to zero classical direct

[2]Again the tilde symbol will be omitted.

methods lead to an unacceptable growth of the number of arithmetical operations required . The boundary potential method offers a good compromise at this point. If the inner system (26) is solved quite exactly, the spectral equivalence constants are expected to be close to 1. On the other hand we saw, that the number of arithmetical operations is asymptotically suboptimal, if the number of inner iterations is fixed.

We choose

$$\begin{aligned} C_I &= \operatorname{blockdiag}(C_{I,s}), \\ C_{I,s}^{-1}\underline{r}_{I,s} &= \left[\left(\bar{L}_{F,s}^{-1} - \bar{L}_{F,s}^{-1}\hat{N}_s\,(I - S_s^{n_C})\,B_s^{-1}\hat{N}_s^T\bar{L}_{F,s}^{-1}\right)\begin{pmatrix} \underline{r}_{I,s} \\ 0_{N,s} \end{pmatrix}\right]_{\omega_{I,s}}, \end{aligned} \tag{32}$$

where n_C is the number of iteration steps (30) performed in order to solve (26) approximately.

Lemma 4.1 *C_I satisfies the requirements (11) of theorem 2.1, where*

$$\begin{aligned} \underline{\gamma}_I &= \inf_{\underline{x}_I} \frac{\left(C_I^{-1}\underline{x}_I, \underline{x}_I\right)}{\left(L_I^{-1}\underline{x}_I, \underline{x}_I\right)} = 1 - c_0^{n_C}, \\ \overline{\gamma}_I &= \sup_{\underline{x}_I} \frac{\left(C_I^{-1}\underline{x}_I, \underline{x}_I\right)}{\left(L_I^{-1}\underline{x}_I, \underline{x}_I\right)} = 1 + c_0^{n_C}. \end{aligned}$$

Proof:

A $C_I = C_I^T$ is positive definite. From (32) we have

$$\begin{aligned} \left(C_{I,s}^{-1}\underline{x}_{I,s}, \underline{y}_{I,s}\right) &= \left(L_{I,s}^{-1}\underline{x}_{I,s}, \underline{y}_{I,s}\right) \\ &+ \left(S_s^{n_C}B_s^{-1}\hat{N}_s^T\bar{L}_{F,s}^{-1}\begin{pmatrix} \underline{x}_{I,s} \\ 0_{N,s} \end{pmatrix}, B_s^{-1}\hat{N}_s^T\bar{L}_{F,s}^{-1}\begin{pmatrix} \underline{y}_{I,s} \\ 0_{N,s} \end{pmatrix}\right)_{B_s}. \end{aligned}$$

$L_{I,s}$ is symmetric, S_s self-adjoint in the inner product $(.,.)_{B_s}$. Therefore the symmetry of C_I is obvious. Further $L_{I,s}$ and $\bar{L}_{F,s}$, consequently B_s and $B_sS_s^{n_C}$ are positive definite.

B Denoting $b_{\nu,s} = B_s^{-1}\hat{N}_s^T\bar{L}_{F,s}^{-1}\begin{pmatrix} \underline{x}_{I,s} \\ 0_{N,s} \end{pmatrix}$, we get

$$\begin{aligned} \left(C_{I,s}^{-1}\underline{x}_{I,s}, \underline{x}_{I,s}\right) &= \underbrace{\left(\bar{L}_{F,s}^{-1}\begin{pmatrix} \underline{x}_{I,s} \\ 0_{N,s} \end{pmatrix}, \begin{pmatrix} \underline{x}_{I,s} \\ 0_{N,s} \end{pmatrix}\right) + (b_{\nu,s}, b_{\nu,s})_{B_s}}_{\left(L_{I,s}^{-1}\underline{x}_{I,s}, \underline{x}_{I,s}\right)} \\ &\quad + (S_s^{n_C}b_{\nu,s}, b_{\nu,s})_{B_s}. \end{aligned}$$

Using the norm estimate of the iteration operator

$$(S_s b_{\nu,s}, b_{\nu,s})_{B_s} \le c_0 \, (b_{\nu,s}, b_{\nu,s})_{B_s} ,$$

we arrive at the statement of the lemma.

$$(1 - c_0^{n_C}) \left(L_{I,s}^{-1} \underline{x}_{I,s}, \underline{x}_{I,s} \right) \le \left(C_{I,s}^{-1} \underline{x}_{I,s}, \underline{x}_{I,s} \right) \le (1 + c_0^{n_C}) \left(L_{I,s}^{-1} \underline{x}_{I,s}, \underline{x}_{I,s} \right) .$$

■

How to choose the basis transformation? In view of the spectral radius $\mu = \rho(S_C^{-1} T_C)$ the choice $B_I = L_I$ is optimal again. That means solving non-homogeneous Dirichlet problems on each subdomain Π_s, where the right-hand side $f_{I,s}$ is zero. In other words we are searching for the biharmonic extension of the Dirichlet data $\underline{w}_{C,s} = \begin{pmatrix} \underline{w}_{D,s} \\ \underline{w}_{N,s} \end{pmatrix}$ to the interior nodes of the subdomain Π_s.

Using the modified boundary potential method developed in section 3 this can be done exactly when the inner systems (26) are solved directly.

$$C_*^{-1} = V_*^{-T} \begin{pmatrix} C_C^{-1} & 0 \\ 0 & C_I^{-1} \end{pmatrix} V_*^{-1}, \tag{33}$$

$$V_{*,s}^{-1} \underline{w}_s = \begin{pmatrix} \underline{w}_{D,s} - L_{FD,s}^T \bar{L}_{F,s}^{-1} \underline{w}_{F,s} \\ 0_{N,s} \\ \underline{w}_{I,s} \end{pmatrix} + \begin{pmatrix} L_{FD,s}^T \bar{L}_{F,s}^{-1} \hat{N}_s B_s^{-1} \hat{N}_s^T \bar{L}_{F,s}^{-1} \underline{w}_{F,s} \\ N_s B_s^{-1} \hat{N}_s^T \bar{L}_{F,s}^{-1} \underline{w}_{F,s} \\ 0_{I,s} \end{pmatrix} \tag{34}$$

Unfortunately, when the inner systems are solved iteratively, the resulting DD-preconditioner

$$C^{-1} = V^{-T} \begin{pmatrix} C_C^{-1} & 0 \\ 0 & C_I^{-1} \end{pmatrix} V^{-1}, \tag{35}$$

$$V_s^{-1} \underline{w}_s = V_{*,s}^{-1} \underline{w}_s - \begin{pmatrix} L_{FD,s}^T \bar{L}_{F,s}^{-1} \hat{N}_s S_s^{n_B} B_s^{-1} \hat{N}_s^T \bar{L}_{F,s}^{-1} \underline{w}_{F,s} \\ N_s S_s^{n_B} B_s^{-1} \hat{N}_s^T \bar{L}_{F,s}^{-1} \underline{w}_{F,s} \\ 0_{I,s} \end{pmatrix} . \tag{36}$$

does not fit into the framework of theorem 2.1.

Lemma 4.2 *Let us use the boundary potential method for the basis transformation and let us use the Jacobi iterative method (30) to solve the inner system (26). Then, performing a the DD-preconditioner C spectrally equivalent to C_*, i.e.*

$$\exists \mu_1, \mu_2 > 0 : \mu_1 \left(C_*^{-1} \underline{x}, \underline{x} \right) \le \left(C^{-1} \underline{x}, \underline{x} \right) \le \mu_2 \left(C_*^{-1} \underline{x}, \underline{x} \right) , \quad \forall \underline{x}$$

requires not more than $n_B = O(\ln h^{-1})$ iteration steps (30).

Proof: In a first step, let us assume

$$\exists\, \delta > 0 : \left\| \left(V_*^{-1} - V^{-1} \right) \underline{x} \right\|_{D^{-1}} \leq \delta \left\| V_*^{-1} \underline{x} \right\|_{D^{-1}} , \tag{37}$$

where $D = \begin{pmatrix} C_C & 0 \\ 0 & C_I \end{pmatrix}$. Then the following estimations hold.

$$\begin{aligned} \left| \left(\left(C_*^{-1} - C^{-1} \right) \underline{x}, \underline{x} \right) \right| &= \left| \left\| V_*^{-1} \underline{x} \right\|_{D^{-1}}^2 - \left\| V^{-1} \underline{x} \right\|_{D^{-1}}^2 \right| \\ &= \left| \left(\left(V_*^{-1} + V^{-1} \right) \underline{x}, \left(V_*^{-1} - V^{-1} \right) \underline{x} \right)_{D^{-1}} \right| \\ &\leq \left\| \left(V_*^{-1} + V^{-1} \right) \underline{x} \right\|_{D^{-1}} \left\| \left(V_*^{-1} - V^{-1} \right) \underline{x} \right\|_{D^{-1}} \\ &\leq \left\{ 2 \left\| V_*^{-1} \underline{x} \right\|_{D^{-1}} + \left\| \left(V_*^{-1} - V^{-1} \right) \underline{x} \right\|_{D^{-1}} \right\} \cdot \\ &\quad \cdot \left\| \left(V_*^{-1} - V^{-1} \right) \underline{x} \right\|_{D^{-1}} \\ &\leq (2 + \delta)\, \delta \left(C_*^{-1} \underline{x}, \underline{x} \right) . \end{aligned} \tag{38}$$

From (38) we have $\mu_1 = 1 - (2 + \delta)\, \delta$ and $\mu_2 = 1 + (2 + \delta)\delta$.

In the second step we will state estimation (37). Introducing some notations

$$\left(V_*^{-1} - V^{-1} \right) \underline{x} = \begin{pmatrix} L_{FD}^T \bar{L}_F^{-1} \hat{N} S^{n_B} \underline{b}_\nu^* \\ N S^{n_B} \underline{b}_\nu^* \\ 0_I \end{pmatrix} = \begin{pmatrix} J S^{n_B} \underline{b}_\nu^* \\ 0_I \end{pmatrix} , \tag{39}$$

where $\underline{b}_\nu^* = B^{-1} \hat{N}^T \bar{L}_F^{-1} \underline{x}_F$, we get

$$\begin{aligned} \left\| \left(V_*^{-1} - V^{-1} \right) \underline{x} \right\|_{D^{-1}}^2 &= \| S^{n_B} \underline{b}_\nu^* \|_{J^T C_C^{-1} J}^2 \\ &\leq \| S^{n_B} \|_{J^T C_C^{-1} J}^2 \, (J \underline{b}_\nu^*, J \underline{b}_\nu^*)_{C_C^{-1}} \\ &\leq \| S \|_{J^T C_C^{-1} J}^{2 n_B} \left\| V_*^{-1} \underline{x} \right\|_{D^{-1}}^2 \\ &\leq c^2 h^{-2p} \, \| S \|_\infty^{2 n_B} \left\| V_*^{-1} \underline{x} \right\|_{D^{-1}}^2 . \end{aligned} \tag{40}$$

From (40) we see: To bound δ by a constant ϵ independent of h we have to perform

$$n_B \geq \frac{\ln c + \ln \epsilon^{-1} + p \ln h^{-1}}{\ln c_0^{-1}}$$

inner iterations (30). ∎

5 The Schur-complement preconditioner

In [24] certain multilevel preconditioners for discretizations of the biharmonic equation by rectangular finite elements were proposed. In particular a BPX-like preconditioner C_C for the Schur-complement S_C of a conforming finite element discretization using bicubic Bogner-Fox-Schmidt rectangles (7) is considered there. The author proves that the condition number $\kappa(C_C^{-1}S_C)$ is bounded by a constant depending only on the initial partition $\mathcal{R}_{j_0}$. A different approach to construct multilevel preconditioners for the Schur-complement system is used in [26] and [27]. There multilevel Schur-complement preconditioners are derived from multilevel preconditioners for the underlying global stiffness matrices using algebraic arguments.

In some sense Adini elements can be interpreted as a non-conforming counterpart of Bogner-Fox-Schmidt elements. The degree of the polynomials is reduced such that we have one degree of freedom less per node. Oswald suggests to construct multilevel preconditioners for non-conforming elements by the strategy of switching to a reference multilevel method defined for a conforming finite element scheme (see [24], [25] and references therein).

Here we make use of Xu's abstract theory of the auxiliary space method. The result is an optimal multilevel preconditioner for the Schur-complement $\tilde{S}_C$ resulting from the non-conforming Adini element discretization. Further we discuss the algorithmic realization of the resulting preconditioner and in order to obtain an optimal condition number we introduce two scaling factors and find their nearly optimal values.

5.1 Multilevel Schur-complement preconditioners

We assume that the subdomain pattern of $\Omega = \cup_{s=1}^{p}\Pi_s$ can be interpreted as a conform initial partition $\mathcal{R}_{j_0} = \{\Pi_1, \ldots, \Pi_p\}$. Refining dyadically we generate partitions $\mathcal{R}_l, l = j_0+1, \ldots, j$ and corresponding sets of nodes $\mathcal{N}_l$.

On the finest level j we construct the finite element space of Adini elements $\tilde{V}_h = \operatorname{span}\left\{\tilde{\varphi}_{(\alpha,P)}\right\}_{(\alpha,P)\in\tilde{\tau}\times\mathcal{N}_j}$. By the above described construction the local level j triangulation of the subdomains Π_s is uniform as assumed in section 3. The notations introduced there will keep their sense locally on the subdomains.

Using Bogner-Fox-Schmidt elements we construct a nested sequence of finite element spaces

$$\begin{aligned} V^{j_0} \subset V^{j_0+1} \subset \ldots \subset V^j = V_h, \\ V^l = \operatorname{span}\{\varphi^l_{(\alpha,P)}\}_{(\alpha,P)\in\tau\times\mathcal{N}_l}. \end{aligned}$$

On each level we distinguish different sets of degrees of freedom according to their sense (see also section 3). The full information of the trace of the finite element function $u^l = \Phi^l \underline{u}^l$ on the coupling boundary is contained in the pair

$$u^l_C = \left(u^l_D, u^l_N\right) = \left(\Phi^l_D \underline{u}^l_D, \Phi^l_N \underline{u}^l_N\right) \in V^l_C = V^l_D \times V^l_N. \tag{41}$$

The space V^l_D of traces of level l f.e. functions on Γ_C is spanned by Φ^l_D, the normal derivatives V^l_N are spanned by Φ^l_N.

$$\Phi^l_D = \left\{\phi^l_{(\alpha,P)}\right\}_{(\alpha,P)\in\gamma^l_D}, \tag{42}$$

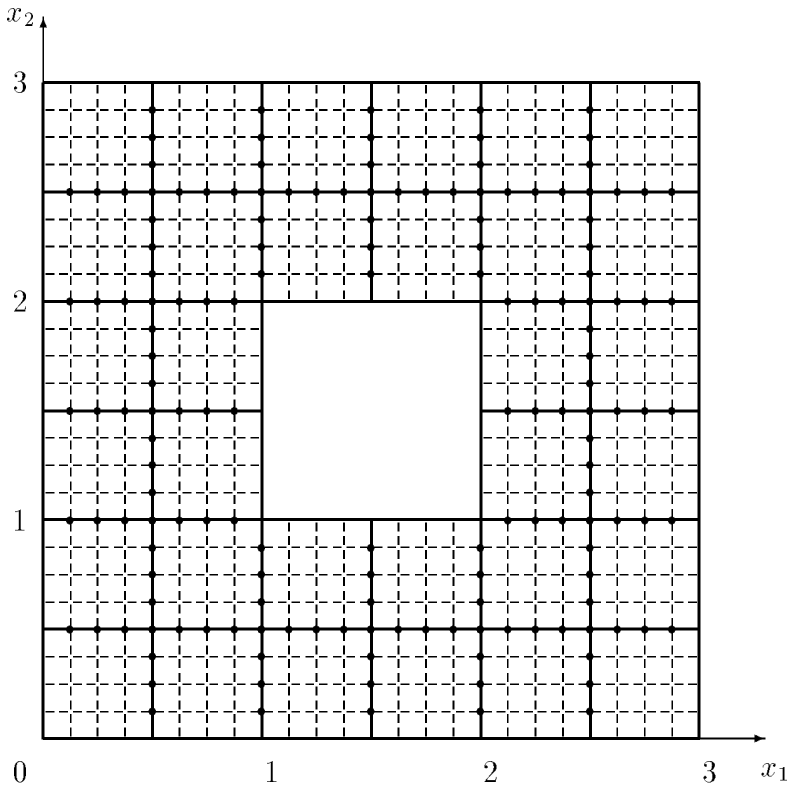

Figure 2: $\Omega = (0,3) \times (0,3) \setminus [1,2] \times [1,2], p = 32, j_0 = 1, j = 3$

$$\begin{aligned} \Phi_N^l &= \left\{ \phi_{(\alpha,P)}^l \right\}_{(\alpha,P)\in\gamma_N^l}, \\ \phi_{(\alpha,P)}^l &= \begin{cases} \left.\varphi_{(\alpha,P)}^l\right|_{\Gamma_C} & : \quad (\alpha,P) \in \gamma_D^l, \\ \left.\partial_\nu \varphi_{(\alpha,P)}^l\right|_{\Gamma_C} & : \quad (\alpha,P) \in \gamma_N^l. \end{cases} \end{aligned} \tag{43}$$

On V_C^j a weighted L_2-inner product is introduced by

$$\left(u_C^j, v_C^j\right)_C = 2^{3j_0} \left(u_D^j, v_D^j\right)_0 + 2^{j_0} \left(u_N^j, v_N^j\right)_0, \quad \forall u_C^j, v_C^j \in V_C^j. \tag{44}$$

The mapping $\mathcal{M}_{k,C} : V_C^j \longrightarrow V_C^k$

$$\mathcal{M}_{k,C} v_C^j = \sum_{(\alpha,P)\in\gamma_D^k\cup\gamma_N^k} \frac{\left(v_C^j, \phi_{(\alpha,P)}^k\right)}{\left(\phi_{(\alpha,P)}^k, \phi_{(\alpha,P)}^k\right)} \cdot \phi_{(\alpha,P)}^k, \quad v_C^j \in V_C^j, \tag{45}$$

which is spectrally equivalent to the orthogonal projection from V_C^j onto V_C^k splits into parts $\mathcal{M}_{k,D} : V_D^j \longrightarrow V_D^k$ and $\mathcal{M}_{k,N} : V_N^j \longrightarrow V_N^k$. Using the above definitions the multilevel Schur-complement preconditioning operator $\mathcal{C}_C$ is defined by

$$\mathcal{C}_C^{-1} = \mathcal{L}_{j_0,C}^{-1}\mathcal{M}_{j_0,C} + \sum_{k=j_0+1}^{j}\left(2^{3(j_0-k)}\mathcal{M}_{k,D} + 2^{(j_0-k)}\mathcal{M}_{k,N}\right), \tag{46}$$

where $\mathcal{L}_{j_0,C}$ corresponds to the underlying boundary value problem discretized on level j_0.

Now, using Xu's abstract framework of the auxiliary space method, we construct a Schur-complement preconditioner for the non-conforming Adini scheme. We define mappings $\mathbf{\Pi}_C : V_C \to \tilde{V}_C$ and $\mathcal{P}_C : \tilde{V}_C \to V_C$. For a given $v_C \in V_C$ $\tilde{v}_C = \mathbf{\Pi}_C v_C$ is defined by

$$\begin{aligned} \tilde{v}_C(P) &= v_C(P), \\ \partial_\tau \tilde{v}_D(P) &= \partial_\tau v_D(P), \quad \forall P \in \mathcal{N}_h \cap \Gamma_C, \end{aligned} \tag{47}$$

given $\tilde{v}_C \in \tilde{V}_C$, $v_C = \mathcal{P}_C \tilde{v}_C$ is defined by

$$\begin{aligned} v_C(P) &= \tilde{v}_C(P), \\ \partial_\tau v_D(P) &= \partial_\tau \tilde{v}_D(P), \\ \partial_\tau v_N(P) &= \begin{cases} \sum_{i=1,2} \frac{\tilde{v}_N(P+e_i h_i) - \tilde{v}_N(P-e_i h_i)}{4\, h_i} & : \ P \text{ crosspoint}, \\ \frac{\tilde{v}_N(P+e_\tau h_\tau) - \tilde{v}_N(P-e_\tau h_\tau)}{2\, h_\tau} & : \ \text{else}, \end{cases} \quad \forall P \in \mathcal{N}_h \cap \Gamma_C. \end{aligned} \tag{48}$$

Lemma 5.1 *Let $\tilde{\mathcal{C}}_C^{-1} = \mathbf{\Pi}_C \mathcal{C}_C^{-1} \mathbf{\Pi}_C^T$. There exists a constant $c \neq c(h)$:*

$$\kappa(\tilde{\mathcal{C}}_C^{-1}\tilde{\mathcal{S}}_C) \leq c. \tag{49}$$

Proof: The statement of the lemma is a direct consequence of the main theorem of [28] applied to our situation. It is obvious that $\mathcal{P}_C$ is a right inverse of $\mathbf{\Pi}_C$, i.e. $\mathbf{\Pi}_C \mathcal{P}_C \tilde{v}_C = \tilde{v}_C$, $\forall \tilde{v}_C \in \tilde{V}_C$, therefore no smoother is required. Using Oswald's result $\kappa\left(\mathcal{C}_C^{-1}\mathcal{S}_C\right) \leq \bar{c}$, $\bar{c} \neq \bar{c}(h)$ the proof is finished. ■

5.2 Optimal scaling

We reformulate the Schur-complement preconditioner and introduce two scaling factors $\alpha, \beta \in \mathbb{R}$.

$$\tilde{\mathcal{C}}_C^{-1} = \beta\, \mathbf{\Pi}_C \left[\mathcal{L}_{j_0,C}^{-1}\mathcal{M}_{j_0,C} + \alpha \sum_{k=j_0+1}^{j}\left(2^{3(j_0-k)}\mathcal{M}_{k,D} + 2^{(j_0-k)}\mathcal{M}_{k,N}\right)\right] \mathbf{\Pi}_C^T, \tag{50}$$

Factor β is caused by the use of the Schur-complement preconditioner within the domain decomposition preconditioner. It is used to get an optimal scaling between the block-preconditioners $\tilde{C}_I$ and $\tilde{C}_C$. Its correct choice results in a minimum value of the quotient $\overline{\gamma}/\underline{\gamma}$ (see theorem 2.1).

The introduction of α is intended to get optimal weighting between the solution of the coarse level problem ($k = j_0$) and parts arising from higher levels ($k = j_0+1, \ldots, j$). The appropriate choice of α becomes more and more important in the many subdomain case. A similar scaling factor was already mentioned in [26]. Using a preconditioned gradient method for the Schur-complement system approximates of the minimum and maximum eigenvalues of $\tilde{C}_C^{-1}\tilde{S}_C$ have been computed numerically (see [6]) using different values of the scaling factor α. Two characteristic examples of the resulting graphs for $\kappa(\tilde{C}_C^{-1}\tilde{S}_C)$ are shown in figure 3 and figure 4. From the knowledge of the spectral bounds we chose β.

5.3 The implementation of the algorithm

The implementation of the mapping $\mathbf{\Pi}_C$ respectively $\mathbf{\Pi}_C^T$ is quite simple. In order to describe the implementation of $\mathcal{C}_C^{-1}$ we consider the preconditioning step in a DD parallelized iterative process for the Schur-complement system

$$S_C \underline{u}_C^j = \underline{d}_C^j.$$

We have to distinguish carefully between several kinds of representation of data. The finite element function u_C^j is represented by a vector of nodal values $\underline{u}_C^j$. In parallel iterative methods these nodal values are of the so-called *overlapping* type of distribution. That means processor s stores the full nodal values of $\gamma_{C,s}^j$ in the local memory.

$$\underline{u}_{C,s}^j = A_{C,j,s}\underline{u}_C^j.$$

The right-hand side vector $\underline{d}_C^j$ and the residual $\underline{r}_C^j = \underline{d}_C^j - S_C\underline{u}_C^j$ contain inner products $\left(d_C^j, \phi_{(\alpha,P)}^j\right)$ respectively $\left(r_C^j, \phi_{(\alpha,P)}^j\right)$. These values are naturally of *adding* type of distribution. The full values corresponding to the coupling nodes are result of a sum including data transfer between processors.

$$\underline{r}_C^j = \sum_{s=1}^{p} A_{C,j,s}^T \underline{r}_{C,s}^j.$$

$A_{C,j,s}$ is the Boolean matrix which maps the global vector of nodal values to the local vector of nodal values corresponding to $\gamma_{C,s}^j$. The usual element by element assembling procedure results in a stiffness matrix of adding type. Therefor the multiplication of the stiffness matrix by a vector of overlapping type results in a vector of adding type and the preconditioning step

$$\underline{w}_C^j = C_C^{-1}\underline{r}_C^j \tag{51}$$

demands a transfer operation from adding type ($\underline{r}_C^j$) to overlapping type ($\underline{w}_C^j$).

Let $S \in \mathcal{N}_l \setminus \mathcal{N}_{l-1} \cap \Gamma_C$ a new coupling node of level l. Then $P_1 \in \mathcal{N}_{l-1} \cap \Gamma_C$ and $P_2 \in \mathcal{N}_{l-1} \cap \Gamma_C$ denote the parent nodes of S lying to the left respectively

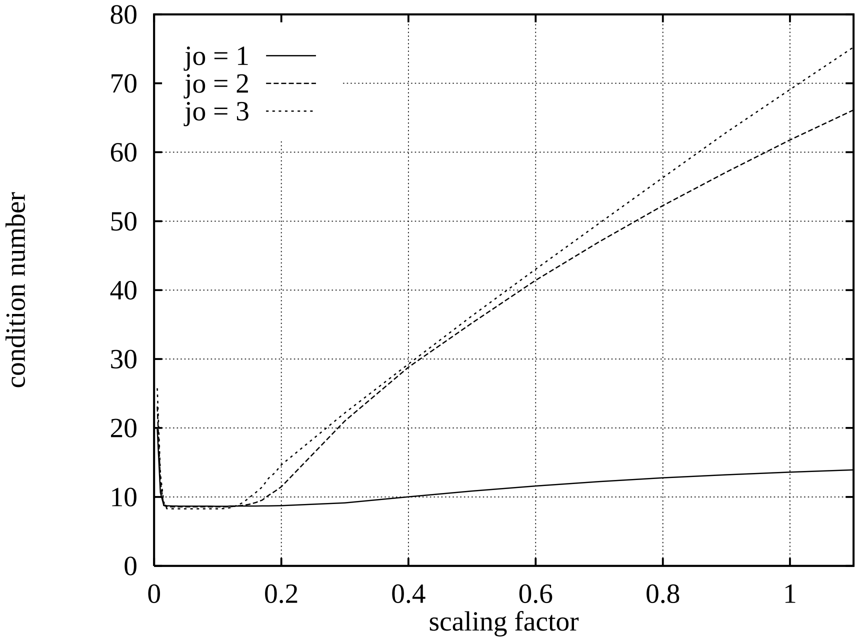

Figure 3: Clamped plate problem in $(0,1)^2$, $\kappa(C_C^{-1}S_C)$ - α, level $j = 6$.

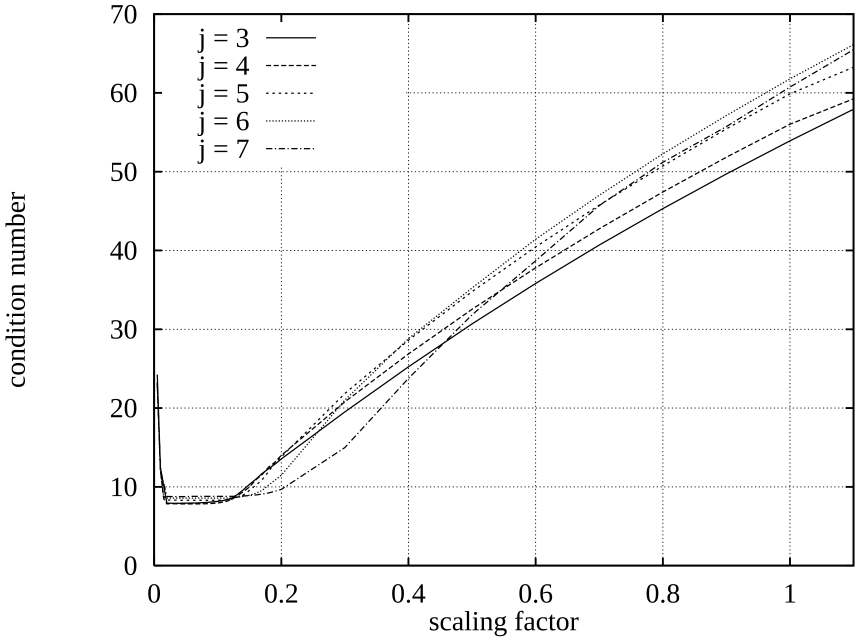

Figure 4: Clamped plate problem in $(0,1)^2$, $\kappa(C_C^{-1}S_C)$ - α, $j_0 = 2$.

to the right of S relatively to the coordinate axis parallel to the part of the boundary under consideration. The preconditioning step (51) could be realized by the following V-cycle.

1. Compute residual scalar products recursively.

for $s = 1$ to p do in parallel
  for $P \in \mathcal{N}_j \cap \Gamma_{C,s}$ do $r^j_{(\mathbf{1},P),s} = 0$ next P
  for $l = j$ down to $j_0 + 1$ do
    for $S \in \mathcal{N}_l \setminus \mathcal{N}_{l-1} \cap \Gamma_{C,s}$ do

$$
\begin{aligned}
r^{l-1}_{(\mathbf{0},P_1),s} &= r^l_{(\mathbf{0},P_1),s} + \tfrac{1}{2}\, r^l_{(\mathbf{0},S),s} + \tfrac{3}{4\,h^l_\tau}\, r^l_{(e_\tau,S),s},\\
r^{l-1}_{(\mathbf{0},P_2),s} &= r^l_{(\mathbf{0},P_2),s} + \tfrac{1}{2}\, r^l_{(\mathbf{0},S),s} - \tfrac{3}{4\,h^l_\tau}\, r^l_{(e_\tau,S),s},\\
r^{l-1}_{(e_\tau,P_1),s} &= r^l_{(e_\tau,P_1),s} - \tfrac{1}{4}\, r^l_{(e_\tau,S),s} - \tfrac{h^l_\tau}{4}\, r^l_{(\mathbf{0},S),s},\\
r^{l-1}_{(e_\tau,P_2),s} &= r^l_{(e_\tau,P_2),s} - \tfrac{1}{4}\, r^l_{(e_\tau,S),s} + \tfrac{h^l_\tau}{4}\, r^l_{(\mathbf{0},S),s},\\
r^{l-1}_{(e_\nu,P_1),s} &= r^l_{(e_\nu,P_1),s} + \tfrac{1}{2}\, r^l_{(e_\nu,S),s} + \tfrac{3}{4\,h^l_\tau}\, r^l_{(\mathbf{1},S),s},\\
r^{l-1}_{(e_\nu,P_2),s} &= r^l_{(e_\nu,P_2),s} + \tfrac{1}{2}\, r^l_{(e_\nu,S),s} - \tfrac{3}{4\,h^l_\tau}\, r^l_{(\mathbf{1},S),s},\\
r^{l-1}_{(\mathbf{1},P_1),s} &= r^l_{(\mathbf{1},P_1),s} - \tfrac{1}{4}\, r^l_{(\mathbf{1},S),s} - \tfrac{h^l_\tau}{4}\, r^l_{(e_\nu,S),s},\\
r^{l-1}_{(\mathbf{1},P_2),s} &= r^l_{(\mathbf{1},P_2),s} - \tfrac{1}{4}\, r^l_{(\mathbf{1},S),s} + \tfrac{h^l_\tau}{4}\, r^l_{(e_\nu,S),s},
\end{aligned}
$$

    next S
  next l
end parallel

2. Change the type of distribution of the residual vectors and solve the coarse grid problem.

$$
\begin{aligned}
\underline{w}^{(j_0)}_{C,s} &= A_{C,j_0,s} L^{-1}_{C,j_0} \sum_{s=1}^{p} A^T_{C,j_0,s} \underline{r}^{(j_0)}_{C,s}\\
\underline{v}^l_{C,s} &= A_{C,l,s} \sum_{s=1}^{p} A^T_{C,l,s} \underline{r}^l_{C,s}, \quad \forall l = j_0+1,\dots,j
\end{aligned}
$$

3. Sum the preconditioned residuals recursively.

for $s = 1$ to p do in parallel
  for $l = j_0 + 1$ to j do
    for $P \in \mathcal{N}_{l-1} \cap \Gamma_{C,s}$ do

$$
\begin{aligned}
w^l_{(\mathbf{0},P),s} &= w^{l-1}_{(\mathbf{0},P),s} + \alpha\, h^l_\tau\, h^l_\nu\, v^l_{(\mathbf{0},P),s},\\
w^l_{(e_\tau,P),s} &= w^{l-1}_{(e_\tau,P),s} + \alpha\, v^l_{(e_\tau,P),s},
\end{aligned}
$$

$$
\begin{aligned}
w^l_{(e_\nu,P),s} &= w^{l-1}_{(e_\nu,P),s} + \alpha\, v^l_{(e_\nu,P),s},\\
w^l_{(\mathbf{1},P),s} &= w^{l-1}_{(\mathbf{1},P),s} + \tfrac{\alpha}{h^l_\tau\, h^l_\nu}\, v^l_{(\mathbf{1},P),s},
\end{aligned}
$$

next P

for $S \in \mathcal{N}_l \setminus \mathcal{N}_{l-1} \cap \Gamma_{C,s}$ do

$$
\begin{aligned}
w^l_{(\mathbf{0},S),s} &= \alpha\, h^l_\tau\, h^l_\nu\, v^l_{(\mathbf{0},S),s} + \tfrac{1}{2}\left(w^{l-1}_{(\mathbf{0},P_1),s} + w^{l-1}_{(\mathbf{0},P_2),s}\right)\\
&\quad + \tfrac{h^l_\tau}{4}\left(w^{l-1}_{(e_\tau,P_1),s} - w^{l-1}_{(e_\tau,P_2),s}\right),\\
w^l_{(e_\tau,S),s} &= \alpha\, v^l_{(e_\tau,S),s} - \tfrac{1}{4}\left(w^{l-1}_{(e_\tau,P_1),s} + w^{l-1}_{(e_\tau,P_1),s}\right)\\
&\quad - \tfrac{3}{4\, h^l_\tau}\left(w^{l-1}_{(\mathbf{0},P_1),s} - w^{l-1}_{(\mathbf{0},P_2),s}\right),\\
w^l_{(e_\nu,S),s} &= \alpha\, v^l_{(e_\nu,S),s} + \tfrac{1}{2}\left(w^{l-1}_{(e_\nu,P_1),s} + w^{l-1}_{(e_\nu,P_2),s}\right)\\
&\quad + \tfrac{h^l_\tau}{4}\left(w^{l-1}_{(\mathbf{1},P_1),s} - w^{l-1}_{(\mathbf{1},P_2),s}\right),\\
w^l_{(\mathbf{1},S),s} &= \tfrac{\alpha}{h^l_\tau\, h^l_\nu}\, v^l_{(\mathbf{1},S),s} - \tfrac{1}{4}\left(w^{l-1}_{(\mathbf{1},P_1),s} + w^{l-1}_{(\mathbf{1},P_1),s}\right)\\
&\quad - \tfrac{3}{4\, h^l_\tau}\left(w^{l-1}_{(e_\nu,P_1),s} - w^{l-1}_{(e_\nu,P_2),s}\right),
\end{aligned}
$$

next S

next l

end parallel

The above described algorithm demands $O(2^{j-j_0})$ arithmetical operations per processor to realize steps 1 and 3. The direct solution of the coarse level problem requires $O(8^{j_0})$ arithmetical operation during each iteration step (L_{C,j_0} is stored in factorized form).

In practice the inter-processor data exchange is organized as follows. Before step 1 we assemble all residual values corresponding to edge points (not the vertices) of the subdomains via nearest neighbour communication. During step 2 a global data exchange is required to assemble the right-hand side of the level j_0 problem. The exchange of values corresponding to vertices at level $j_0+1,\ldots,j$ is enclosed there. The length of data packages in the global exchange is proportional to $j - j_0$.

At least the number of communication steps required is the same compared to the non-preconditioned CG method.

6 A parallel preconditioner for BFS elements

In the previous sections we defined a domain decomposition preconditioner for finite element equations resulting from plate problems discretized by Adini rectangles. The boundary potential method for the subdomain problems was one of its main components. One could try to construct an efficient subdomain solver for the case of the Bogner-Fox-Schmidt elements for instance using the idea of the boundary potential method or using multigrid algorithms. Here we choose an alternate way based on the close relations between both types of rectangular finite elements.

6.1 Application of the auxiliary space method

In order to construct a preconditioner for the discretized plate problem in V_h we consider the auxiliary finite element space $\tilde{V}_h$ where an optimal preconditioner $\tilde{\mathcal{C}}$ is known. From the theory presented in [28] the construction of the preconditioner is clear.

First we define mappings $\mathbf{\Pi} : \tilde{V}_h \to V_h$ and $\mathcal{P} : V_h \to \tilde{V}_h$. The image $v_h = \mathbf{\Pi}\tilde{v}_h \in V_h$ of $\tilde{v}_h \in \tilde{V}_h$ is uniquely defined by

$$\begin{aligned} v_h|_P &= \tilde{v}_h|_P\,, \\ \nabla v_h|_P &= \nabla\tilde{v}_h|_P\,, \\ \partial_{12} v_h|_P &= \frac{1}{m_P}\sum_{R:P\in R} \partial_{12,R}\tilde{v}_h|_P\,, \forall P \in \mathcal{N}_h, \end{aligned} \tag{52}$$

where m_P is the number of elements R with vertex P, $\partial_{12,R}\tilde{v}_h$ is the mixed derivative of $\tilde{v}_h$ evaluated on element R. The mapping $\mathcal{P}$ is the natural interpolant. Given $v_h \in V_h$, $\tilde{v}_h = \mathcal{P}v_h$ is defined by

$$\begin{aligned} \tilde{v}_h|_P &= v_h|_P\,, \\ \nabla\tilde{v}_h|_P &= \nabla v_h|_P\,, \forall P \in \mathcal{N}_h. \end{aligned} \tag{53}$$

Further we choose the smoothing operator of Jacobi type

$$\mathcal{R}v_h = \sum_{(\alpha,P)\in\tau\times\mathcal{N}_h} \frac{\left(v_h, \varphi_{(\alpha,P)}\right)}{a\left(\varphi_{(\alpha,P)}, \varphi_{(\alpha,P)}\right)}\,\varphi_{(\alpha,P)}. \tag{54}$$

Lemma 6.1 *Let $\tilde{\mathcal{C}}$ an optimal preconditioner for the plate problem discretized by Adini elements. Then the preconditioner*

$$\mathcal{C}^{-1} = \mathcal{R} + \mathbf{\Pi}\,\tilde{\mathcal{C}}^{-1}\,\mathbf{\Pi}^T \tag{55}$$

for the plate problem discretized by Bogner-Fox-Schmidt elements leads to a condition number $\kappa\,(\mathcal{C}^{-1}\mathcal{L})$ bounded by a constant independent of h.

Proof: The mappings $\mathbf{\Pi}$ and $\mathcal{P}$ defined in (52) respectively (53) and the smoothing operator $\mathcal{R}$ defined in (54) fulfill the requirements (56 - 59) of the auxiliary space theory presented in [28]. There exist non-negative constants $\alpha_0, \alpha_1, \beta_1$ and positive constants β_0 and γ_0 independent of the mesh size h such that

$$\alpha_0 \, h^4 \, (v_h, v_h) \le (\mathcal{R} v_h, v_h) \le \alpha_1 \, h^4 \, (v_h, v_h) \, , \forall v_h \in V_h, \tag{56}$$

$$\begin{aligned} a\left(\mathbf{\Pi}\tilde{v}_h, \mathbf{\Pi}\tilde{v}_h\right) &\le \beta_1 \, \tilde{a}\left(\tilde{v}_h, \tilde{v}_h\right), \forall \tilde{v}_h \in \tilde{V}_h, && (57)\\ \tilde{a}\left(\mathcal{P} v_h, \mathcal{P} v_h\right) &\le \frac{1}{\beta_0} \, a\left(v_h, v_h\right), \forall v_h \in V_h, && (58)\\ \left\|v_h - \mathbf{\Pi}\mathcal{P} v_h\right\|^2 &\le \frac{1}{\gamma_0} \, h^4 \, a\left(v_h, v_h\right), \forall v_h \in V_h. && (59) \end{aligned}$$

■

6.2 A remark on the parallel implementation

The implementation of the mapping $w_h = \mathbf{\Pi}\tilde{w}_h$ (respectively its matrix representation $\underline{w} = \Pi\underline{\tilde{w}}$) requires the evaluation of the nine-point formula (60) at each point $P \in \mathcal{N}_h \cap \Omega$ and corresponding formulas for the boundary points $P \in \mathcal{N}_h \cap \partial\Omega$.

$$\begin{aligned} w_{(\alpha,P)} &= \tilde{w}_{(\alpha,P)}, \; \alpha = \mathbf{0}, e_1, e_2, && (60)\\ w_{(\mathbf{1},P)} &= \frac{1}{2\,h_2}\left(\tilde{w}_{(e_1,P+e_2)} - \tilde{w}_{(e_1,P-e_2)}\right) + \frac{1}{2\,h_1}\left(\tilde{w}_{(e_2,P+e_1)} - \tilde{w}_{(e_2,P-e_1)}\right)\\ &- \frac{1}{4\,h_1\,h_2}\left(\tilde{w}_{(\mathbf{0},P+e_1+e_2)} + \tilde{w}_{(\mathbf{0},P-e_1-e_2)} - \tilde{w}_{(\mathbf{0},P-e_1+e_2)} - \tilde{w}_{(\mathbf{0},P+e_1-e_2)}\right). \end{aligned}$$

Therefore, in addition to the auxiliary preconditioner $\tilde{\mathcal{C}}$ and the smoother $\mathcal{R}$ the mappings $\mathbf{\Pi}$ respectively $\mathbf{\Pi}^T$ require data exchange between processors.

Let the residual vector $\underline{d}$ of adding type. Assembling those components which correspond to mixed derivatives ($\alpha = \mathbf{1}$) and evaluating $\Pi^T\underline{d}$ locally on the subdomains results in a vector $\underline{\tilde{d}}$ of adding type. Let $\underline{\tilde{w}} = \tilde{C}^{-1}\underline{\tilde{d}}$ of overlapping type, then the local evaluation of Π on each subdomain results in a vector $\underline{\bar{w}}$ of overlapping type for $\alpha = \mathbf{0}, e_1, e_2$ and adding type for $\alpha = \mathbf{1}$. The smoothing algorithm simply consists of the division by the main diagonal D of the stiffness matrix L and the assembly of the resulting vector $\underline{s} = D^{-1}\underline{d}$. For the adding components of $\bar{w}$ the sum $\underline{w} = \underline{s} + \underline{\bar{w}}$ should be evaluated before assembling $\underline{s}$.

Three communication steps are contained in the algorithm. In the first step one number per coupling node is exchanged, the auxiliary preconditioner exchanges three numbers per coupling node and assembling the smoothed residual $\underline{s}$ requires the exchange of four numbers per coupling node. At least compared to the iterative method using no preconditioner or a DD preconditioner designed for Bogner-Fox-Schmidt elements we have twice as much data exchange between processors.

7 Numerical tests

The following numerical tests have been performed on a multiprocessor system GC/Power Plus 128 equipped by 64 nodes of 2 processors PowerPC-601 and 4 transputers T805 for communication. We introduce some notations used in this section.

- Three types of boundary conditions are considered. The clamped part of the boundary is denoted by Γ_0, the simply supported part by Γ_1 and the free part by Γ_2. Essential boundary conditions are considered to be homogeneous.
- The number of subdomains p is proportional to 4^{j_0}, where j_0 is the number of the coarse level.
- N_* denotes the number of unknown degrees of freedom on the entire domain at discretization level j .
- $I(\varepsilon)$ is the number of CG iterations required to arrive at the relative accuracy ε:
$$\left\| \underline{u} - \underline{u}^{I(\varepsilon)} \right\|_{LC^{-1}L} \leq \varepsilon \left\| \underline{u} - \underline{u}^{0} \right\|_{LC^{-1}L} .$$
- T_T is the total time required to solve the considered problem and
- T_A is the time of pure arithmetics without interprocessor communication.

The largest subdomain problems we could fit into memory consider a local 256×256-mesh ($j - j_0 = 8$). Times marked by a star were measured when our program was using so called *virtual processors*. That means for instance 128 physical processors solve a 256 subdomain problem.

Our domain decomposition preconditioner uses four control parameters:

- The number of inner iterations performed during basis transformation (n_B) and subdomain preconditioning (n_C) via the boundary potential method and
- scaling factors α and β for the multilevel Schur-complement preconditioner.

In order to investigate the parallelization properties of the ParCG method we compare speedup $S(p,q)$ and efficiency $E(p,q)$ between p and q processores at equal discretization levels (j =constant) or equal size of the subdomain problems ($j - j_0$ =constant).

$$\begin{aligned} S(p,q) &= \frac{T_T(p)}{N_*(p)} \cdot \frac{N_*(q)}{T_T(q)}, \qquad (61) \\ E(p,q) &= \frac{p}{q} \cdot S(p,q). \end{aligned}$$

7.1 Adini elements

Let us give short descriptions of the model problems we solved in order to demonstrate the behaviour of the DD preconditioned ParCG method for solving finite element equations arising from Adini discretizations of plate problems.

Example 1	model domain:	$\Omega = (0,1) \times (0,1)$,
		$p = 4, 16, 64, 256$,
	boundary conditions:	$\Gamma_0 = \partial\Omega$,
	discretization:	Adini elements,
	parameter:	$\alpha = 0.05, \beta = 1$,
		$n_C = n_B = 20$,
	see tables 1,2,3.	

Example 2	model domain:	$\Omega_L = (0,3) \times (0,3) \setminus [1,2] \times [1,2]$,
		$p = 32, 128$,
	boundary conditions:	$\Gamma_0 = \partial\Omega$,
	discretization:	Adini elements
	parameter:	$\alpha = 0.02, \beta = 1$,
		$n_C = n_B = 20$,
	see table 4.	

Example 3	model domain:	$\Omega_L = (0,3) \times (0,3) \setminus [1,2] \times [1,2]$,
		$p = 32, 128$,
	boundary conditions:	$\Gamma_0 = \{0,3\} \times (0,3) \cup [0,3] \times \{0,3\}$,
		$\Gamma_1 = \{1,2\} \times (1,2) \cup [1,2] \times \{1,2\}$,
	discretization:	Adini elements
	parameter:	$\alpha = 0.02, \beta = 1$,
		$n_C = n_B = 20$,
	see table 5.	

Example 4	model domain:	$\Omega_L = (0,3) \times (0,3) \setminus [1,2] \times [1,2]$,
		$p = 32, 128$,
	boundary conditions:	$\Gamma_0 = \{0,3\} \times (0,3) \cup [0,3] \times \{0,3\}$,
		$\Gamma_2 = \{1,2\} \times (1,2) \cup [1,2] \times \{1,2\}$,
	discretization:	Adini elements
	parameter:	$\alpha = 0.02, \beta = 1$,
		$n_C = n_B = 20$,
	see table 6.	

		$I(10^{-5})$				T_T[s]			
N_*	$j \mid j_0$	1	2	3	4	1	2	3	4
27	2	4	—	—	—	0.09	—	—	—
147	3	10	13	—	—	0.18	0.45	—	—
675	4	14	19	14	—	0.56	0.74	1.89	—
2 883	5	15	18	18	—	2.17	1.15	2.89	—
11 907	6	16	19	17	16	8.75	3.22	3.58	31.16*
48 387	7	16	18	17	16	36.17	10.54	5.18	34.97*
195 075	8	16	18	15	14	157.73	43.13	10.97	36.32*
783 363	9	16	17	14	12	685.00	173.79	35.62	43.66*
3 139 587	10	—	16	13	10	—	709.00	134.13	74.80*
12 570 627	11	—	—	13	10	—	—	572.09	228.58*

Table 1: Square clamped plate (Adini elements): iteration counts and time.

j	$S(4,16)$	$S(4,64)$	$S(16,64)$	$E(4,16)$	$E(4,64)$	$E(16,64)$
7	3.43	6.98	2.03	0.86	0.44	0.51
8	3.66	14.38	3.93	0.91	0.90	0.98
9	3.96	19.23	4.86	0.99	1.20	1.22
10	—	—	5.29	—	—	1.32

Table 2: Square clamped plate (Adini elements): speedup and efficiency.

$j - j_0$	$S(4,16)$	$S(4,64)$	$S(16,64)$	$E(4,16)$	$E(4,64)$	$E(16,64)$
4	2.78	7.03	2.53	0.70	0.44	0.63
5	3.37	13.07	3.87	0.84	0.82	0.97
6	3.38	16.44	4.86	0.85	1.03	1.22
7	3.66	18.93	5.17	0.91	1.18	1.29
8	3.87	19.21	4.96	0.97	1.20	1.24

Table 3: Square clamped plate (Adini elements): speedup and efficiency.

		$I(10^{-5})$		T_T[s]	T_A[s]	T_T[s]	T_A[s]
j	N_*	$j_0 = 1$	$j_0 = 2$	$j_0 = 1$		$j_0 = 2$	
3	1 344	23	—	1.59	0.61	—	
4	5 760	24	25	2.31	1.08	10.78	5.10
5	23 808	25	25	4.71	3.16	12.92	5.58
6	96 768	25	26	14.01	12.16	17.39	8.01
7	390 144	25	26	54.23	52.00	28.76	17.52
8	1 566 720	26	27	237.73	234.79	74.66	61.17
9	6 279 168	25	27	1000.90	995.59	268.83	250.00
10	25 141 248	—	27	—		1121.92	1080.14

Table 4: Clamped plate Ω_L, Adini elements.

		$I(10^{-5})$		T_T[s]	T_A[s]	T_T[s]	T_A[s]
j	N_*	$j_0 = 1$	$j_0 = 2$	$j_0 = 1$		$j_0 = 2$	
3	1 372	32	—	2.19	0.87	—	
4	5 820	33	33	3.16	1.48	14.12	6.71
5	23 932	34	33	6.36	4.24	16.95	7.35
6	97 020	36	35	19.97	17.39	23.22	10.72
7	390 652	37	35	79.12	76.05	38.34	23.45
8	1 567 740	37	36	335.89	331.44	98.51	81.02
9	6 281 212	38	37	1502.66	1497.73	366.89	340.44
10	25 145 340	—	37	—		1519.51	1466.60

Table 5: Clamped/simply supported plate Ω_L, Adini elements.

		$I(10^{-5})$		T_T[s]	T_A[s]	T_T[s]	T_A[s]
j	N_*	$j_0 = 1$	$j_0 = 2$	$j_0 = 1$		$j_0 = 2$	
3	1 440	27	—	1.87	0.74	—	
4	5 952	30	26	2.90	1.35	11.23	5.36
5	24 192	29	28	5.45	3.66	14.48	6.29
6	97 536	30	27	16.73	14.56	18.07	8.30
7	391 680	30	28	64.54	62.02	30.87	18.90
8	1 569 792	30	29	273.11	270.03	80.03	65.57
9	6 285 312	30	29	1198.49	1193.91	284.85	266.72
10	25 153 536	—	29	—		1184.31	1155.86

Table 6: Clamped/free plate Ω_L, Adini elements.

7.2 Bogner-Fox-Schmidt elements

Now we consider conforming discretizations of plate problems using Bogner-Fox-Schmidt elements. The resulting finite element equations are solved by the conjugate gradient method preconditioned as described in section 6.

Example 5 model domain: $\Omega = (0,1) \times (0,1)$,
$p = 1, 4, 16, 64, 256$,
boundary conditions: $\Gamma_0 = \partial\Omega$,
discretization: Bogner-Fox-Schmidt elements,
parameter: $\alpha = 0.05, \beta = 1$,
$n_C = n_B = 20$,
see table 7.

Example 6 model domain: $\Omega = (0,1) \times (0,1)$,
$p = 1, 4, 16, 64$,
boundary conditions: $\Gamma_1 = [0,1] \times \{0,1\}$,
$\Gamma_2 = \{0,1\} \times (0,1)$,
discretization: Bogner-Fox-Schmidt elements,
parameter: $\alpha = 0.05, \beta = 1$,
$n_C = n_B = 20$,
see table 8.

		$I(10^{-5})$					T_T[s]			
N_*	$j \mid j_0$	0	1	2	3	4	0	1	2	3
4	1	1	—	—	—	—	0.02	—	—	—
36	2	6	6	—	—	—	0.09	0.18	—	—
196	3	9	14	12	—	—	0.80	0.42	0.81	—
900	4	8	15	13	11	—	1.64	0.75	0.87	2.17
3 844	5	5	14	15	13	8	3.52	2.23	1.42	2.98
15 876	6	5	14	15	13	11	14.34	8.47	3.11	3.65
64 516	7	5	14	15	13	12	60.76	35.49	10.09	5.36
260 100	8	5	14	16	13	11	262.57	151.97	42.35	11.56
1 044 484	9	—	14	16	13	10	—	659.22	178.87	37.13
4 168 116	10	—	—	16	13	10	—	—	768.46	146.81
16 760 836	11	—	—	—	13	10	—	—	—	617.24

Table 7: Square clamped plate (BFS elements): iteration counts and time.

		$I(10^{-5})$				T_T[s]			
N_*	$j\,\vert\,j_0$	0	1	2	3	0	1	2	3
24	1	6	—	—	—	0.03	—	—	—
80	2	15	15	—	—	0.20	0.55	—	—
288	3	18	20	13	—	0.63	0.73	1.35	—
1 088	4	20	20	16	10	2.68	1.21	1.98	1.96
4 224	5	22	21	17	14	12.36	3.38	2.46	3.21
16 640	6	25	23	18	15	59.83	15.85	4.51	4.14
66 048	7	27	26	20	15	278.13	63.98	13.82	6.03
263 168	8	28	27	21	15	1249.19	283.34	54.17	13.12
1 050 624	9	—	28	21	15	—	1273.64	225.06	42.47
4 198 400	10	—	—	22	16	—	—	1013.10	177.85
16 785 408	11	—	—	—	16	—	—	—	754.26

Table 8: Square simply supported/free plate (BFS-elements): iteration counts and time.

8 Conclusions

We developed a domain decomposition preconditioner $\tilde{C}$ for the finite element equations $\tilde{L}\underline{\tilde{u}} = \underline{\tilde{f}}$ arising from the discretization of plate problems by non-conforming Adini elements. Let us formulate its main properties and the properties of the resulting parallel preconditioned conjugate gradient method within two theorems.

Theorem 8.1 (Spectrally equivalence of $\tilde{C}$ and $\tilde{L}$) *Let $\tilde{C}_*$ defined in (33), where $\tilde{C}_C$ is the modified multilevel Schur-complement preconditioner described in section 5 and $\tilde{C}_I$ is defined by the boundary potential method. Then there are constants $\underline{c}_1$ and $\overline{c}_1$ independent of the mesh size and independent of the number of subdomains, such that*

$$\underline{c}_1\, \tilde{C}_* \le \tilde{L} \le \overline{c}_1\, \tilde{C}_*.$$

Using the Boundary Potential method to define the basis transformation (35) it is sufficient to perform $O(\ln h^{-1})$ inner iterations, to have constants $\underline{c}$ and $\overline{c}$ independent of the mesh size and independent of the number of subdomains, such that

$$\underline{c}\,\tilde{C} \le \tilde{L} \le \overline{c}\,\tilde{C}.$$

Therefor the preconditioned ParPCG method for solving the finite element equations needs at most $O(\ln \varepsilon^{-1})$ iterations to achieve the relative accuracy ε.

Proof: The statement follows directly from theorem 2.1 using lemmata 4.1, 5.1 and 4.2. ■

We claim that the estimations in lemma 4.2 can be improved. Numerical results show, that at least a constant number n_B of inner iterations is sufficient.

Theorem 8.2 (Arithmetical cost of the algorithm) *Let us solve the coarse level equations via the Cholesky method and let the coarse grid matrix be stored in factorized form. Then one ParPCG step requires*

$$O\left((j-j_0)\cdot 4^{j-j_0}\right)+O\left(8^{j_0}\right)$$

arithmetical operations per processor.

Proof: The statement is clear from the properties of the well known methods used herein and from the considerations of section 3 concerning the arithmetical cost of the boundary potential method and section 5 concerning the algorithm of the Schur-complement preconditioner. ■

In view of both discretization parameters j (discretization level) and j_0 (subdomain pattern) an asymptotically suboptimal method for the approximate solution of plate problems in domains composed of rectangles has been constructed.

The numerical results confirm the above statements. In particular we see that (at least using our hardware) high efficiency and speedup of the parallel method are obtained for large subdomain problems. This fact becomes more important when the computational performance of the single nodes is high compared to the performance of the communication network.

Further, we constructed an asymptotically optimal preconditioner C for the finite element equations $L\underline{u}=\underline{f}$ arising from the conforming finite element discretization of the plate problem by Bogner-Fox-Schmidt rectangles. At least we obtain the same iteration counts as for the non-conforming discretization preconditioned by the above mentioned DD preconditioner (compare tables 1 and 7). The higher iteration time is caused by additional arithmetical work, but mostly by the additional inter-processor communication.

References

[1] L. D. Landau, E. M. Lifschitz. Lehrbuch der theoretischen Physik, Bd.VII Elastizitätstheorie. Akademie-Verlag, Berlin, 1989.

[2] P. G. Ciarlet. The finite element method for elliptic problems. North-Holland, Amsterdam, 1978.

[3] P. G. Ciarlet. Basic error estimates for elliptic problems, in Handbook of Numerical Analysis, Vol.II: Finite Elements (P. G. Ciarlet and J. L. Lions, Editors). North-Holland, Amsterdam, 1990.

[4] I. Babuška, B. Szabo. Finite Element Analysis. John Wiley & Sons, 1991.

[5] A. A. Samarskij, E. N. Nikolajew. Numerical methods for grid equations. Vol.I: Direct methods. Birkhäuser, 1989.

[6] A. A. Samarskij, E. N. Nikolajew. Numerical methods for grid equations. Vol.II: Iterative methods. Birkhäuser, 1989.

[7] W. Hackbusch. Theorie und Numerik elliptischer Differentialgleichungen. Teubner, Stuttgart, 1986.

[8] W. Hackbusch. Iterative Lösung großer schwachbesetzter Gleichungssysteme. Teubner, Stuttgart, 1991.

[9] J. H. Bramble, J. E. Pasciak, A. H. Schatz. The Construction of Preconditioners for Elliptic Problems by Substructuring. Part I, in Math. of Computation, Vol. 47, No. 175, pp. 103–134, 1986.

[10] G. Haase, U. Langer, 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, pp. 1–59, Berlin, 1990. Academy of Science. Report-Nr. R-MATH-09/90.

[11] G. Haase, U. Langer, A. Meyer. The Approximate Dirichlet Domain Decomposition Method, Part I and Part II, Computing, Vol. 47, pp. 137–151, pp. 153–167, 1991.

[12] G. Haase, U. Langer, A. Meyer. Domain decomposition preconditioners with inexakt subdomain solvers. J. of Num. Lin. Alg. with Appl., Vol. 1, pp. 27–42, 1992.

[13] G. Haase, U. Langer. The non-overlapping domain decomposition multiplicative Schwarz method, Intern. J. Comp. Math., Vol. 44, pp. 223–242.

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

[15] G. Haase. Die nichtüberlappende Gebietszerlegungsmethode zur Parallelisierung und Vorkonditionierung iterativer Verfahren. Dissertation, TU Chemnitz – Zwickau, 1993.

[16] M. Pester. Zur Implementation paralleler Algorithmen. Kommunikation im Hypercube. Preprint 152, TU Karl-Marx-Stadt, 1990.

[17] H. Matthes. Gebietszerlegungstechniken für die biharmonische Gleichung. Diplomarbeit. TU Chemnitz – Zwickau, 1993.

[18] H. Matthes. A DD Preconditioner for the clamped plate problem. Preprint SPC95_9, TU Chemnitz-Zwickau, 1995.

[19] D. Bahlmann. Bystrye metody čislennogo rešenija sadazi dirichle dlia bigarmoničeskogo operatora. Dissertation. Leningrad 1990.

[20] D. Bahlmann, V. G. Korneev. The method of boundary potentials for solving the clamped plate bending problem in a rectangle. Preprint 236, TU Chemnitz, 1993.

[21] U. Langer. Ob iteratzionnom rešenii nekotorych schem metoda koečnych elementov dlia elliptičeskich uravnenii poriadka $2n, n \geq 2$. Jurnal vyčislitelnoi matematiki i matemetičeskoi phyziki, Vol. 23, No. 4, pp. 881–891, 1983.

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

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

[24] P. Oswald. Preconditioners for discretizations of the biharmonic equation by rectangular finite elements. Forschungsergebnisse, FSU Jena, Math./91/3, 1991.

[25] P. Oswald. Multilevel finite element approximation: theory and applications. Teubner, Stuttgart, 1994.

[26] B. F. Smith, O. B. Widlund. A domain decomposition algorithm using a hierarchical basis. SIAM J. Sci. Stat. Comput., Vol. 11, No. 6, pp. 1212–1220, 1990.

[27] C. H. Tong, T. F. Chan, C. C. J. Kuo. A domain decomposition preconditioner based on a change to a multilevel nodal basis. SIAM J. Sci. Stat. Comput., Vol. 12, No. 6, pp. 1486–1495, 1991.

[28] J. Xu. The auxiliary space method and optimal multigrid preconditioning techniques for unstructered grids. To appear in: Computing, special issue for the International Workshop on Multilevel Methods, 1994 at Meisdorf, Germany.