Technische Universität Chemnitz-Zwickau

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Michael Thess

Parallel Multilevel Preconditioners for Problems of Thin Smooth Shells

Preprint SFB393/97_13

Abstract

In the last years multilevel preconditioners like BPX became more and more popular for solving second-order elliptic finite element discretizations by iterative methods. P. Oswald has adapted these methods for discretizations of the fourth order biharmonic problem by rectangular conforming Bogner-Fox-Schmidt elements and nonconforming Adini elements and has derived optimal estimates for the condition numbers of the preconditioned linear systems. In this paper we generalize the results from Oswald to the construction of BPX and Multilevel Diagonal Scaling (MDS-BPX) preconditioners for the elasticity problem of thin smooth shells of arbitrary forms where we use Koiter's equations of equilibrium for an homogeneous and isotropic thin shell, clamped on a part of its boundary and loaded by a resultant on its middle surface. We use the two discretizations mentioned above and the preconditioned conjugate gradient method as iterative method. The parallelization concept is based on a non-overlapping domain decomposition data structure. We describe the implementations of the multilevel preconditioners. Finally, we show numerical results for some classes of shells like plates, cylinders, and hyperboloids.

Key words: thin shell problems, linear partial differential equations, parallel computing, multilevel methods, additive splittings, finite element methods, cooling towers

Preprint-Reihe des Chemnitzer SFB 393

SFB393/97_13 **March 1997**

Contents

Authors' address:

Michael Thess
Faculty for Mathematics
Technical University Chemnitz-Zwickau
D - 09107 Chemnitz, Germany

e-mail: `m.thess@mathematik.tu-chemnitz.de`

1 Introduction

The BPX-like multilevel preconditioners are very efficient for second-order elliptic finite element (FE) discretizations, since they have a convergence rate which is independent of the discretization parameter, and the cost of arithmetical work per iteration step is proportional to the number of unknowns. In [14] Oswald has adapted these methods for discretizations of the fourth order biharmonic problem by rectangular conforming Bogner-Fox-Schmidt (BFS) elements and nonconforming Adini elements and has derived optimal estimates for the condition numbers of the preconditioned linear systems. Using these results, in [10] Matthes developed BPX and Multilevel Diagonal Scaling (MDS-BPX) preconditioners for discretizations of problems of cylindrical shells by BFS elements.

In this paper we generalize the results of [14], [10] to the construction of BPX-like preconditioners for the elasticity problem of smooth, thin shells of arbitrary form where we use Koiter's equations of equilibrium for an homogeneous and isotropic thin shell, clamped on a part of its boundary and loaded by a resultant on its middle surface. We also use BFS and Adini elements for the discretization.

We discuss the implementation of the multilevel preconditioners on parallel computers with MIMD architecture and a message passing communication handling. The implementation of all algorithms is based on a non-overlapping domain decomposition (DD) data structure (cf. [7]).

The paper is organized as follows: In Section 2 we describe the geometry of an arbitrary smooth, thin shell in the tensorial form following [6] and mention some basic properties of the tensors used. In accordance to [3], in Section 3 we give the variational formulation of Koiter's model, using the two basic hypotheses of this model for reducing the study of the strain of a thin shell to the determination of the displacement field of the middle surface. In Section 4 we introduce the finite element discretization of the variational problem by BFS and Adini elements. Section 5 is devoted to the construction of multilevel precondioners by stable subspace splittings of the BFS element space and the transformation of the preconditioner to the Adini element space. For this, we use Oswald's theory of [13]. The parallelization and implementation of the algorithms is described in Section 6, which contains three subsections. In the first subsection we present the DD parallelization concept which is used in the second and third subsection to describe the algorithms for the assemblation of the stiffness matrix and for the parallelization of the preconditioned conjugate gradient (PPCG) method. In Section 7, the results of the MDS-BPX preconditioner for plates, arches rsp. cylinders, and hyperbolids are presented by means of different numerical examples. Finally, we give some conclusions.

2 Description of the geometry of a thin shell

2.1 Definition of the middle surface

First we introduce some notations. Vectors are always denoted by bold typed letters. In the three-dimensional Euclidian space $\mathbf{R}^3$ with the orthonormal basis $(\mathbf{e}_i)$ the Euclidian product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbf{R^3}$ is denoted by $\mathbf{u} \cdot \mathbf{v}$, $|\cdot|$ denotes the associated Euclidian norm, and the vector product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbf{R^3}$ is denoted by $\mathbf{u} \times \mathbf{v}$. We consider the bounded open connected domain Ω in the two-dimensional Euclidian space, its boundary Γ is assumed to be Lipschitz-continuous. The points of $\bar{\Omega}$ are denoted by $(x^1,\, x^2)$. Greek indices always belong to the set $\{1,\, 2\}$, Latin indices always belong to the set $\{1,\, 2,\, 3\}$, and we use the summation convention. Finally, we write the partial derivations in the form $\partial_\alpha = \partial/\partial x^\alpha$, $\partial_{\alpha\beta} = \partial^2/\partial x^\alpha \partial x^\beta$.

Let S be the *middle surface* of the *shell* $\mathcal{C}$. Then S is the image of the set $\bar{\Omega}$ through a mapping $\boldsymbol{\phi} = \phi^i(x^1, x^2)\mathbf{e}_i : \bar{\Omega} \to \mathbf{R}^3$ where the mapping $\boldsymbol{\phi}$ is of the class $\mathcal{C}^3$.

Now we define the basis vectors

$$\mathbf{a}_\alpha = \partial_\alpha \boldsymbol{\phi} = \frac{\partial \boldsymbol{\phi}}{\partial x^\alpha}, \ \alpha = 1, 2 \tag{2.1}$$

which span the *tangent plane* to the surface $S = \boldsymbol{\phi}(\bar{\Omega})$ at the point $\boldsymbol{\phi}(x^1,\, x^2)$ (cf. Fig.1), if they are linearly independent at each point $(x^1,\, x^2) \in \bar{\Omega}$. The last, *regularity* condition we take as a further assumption. The *normal vector* to the tangent plane is given by

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|} \ . \tag{2.2}$$

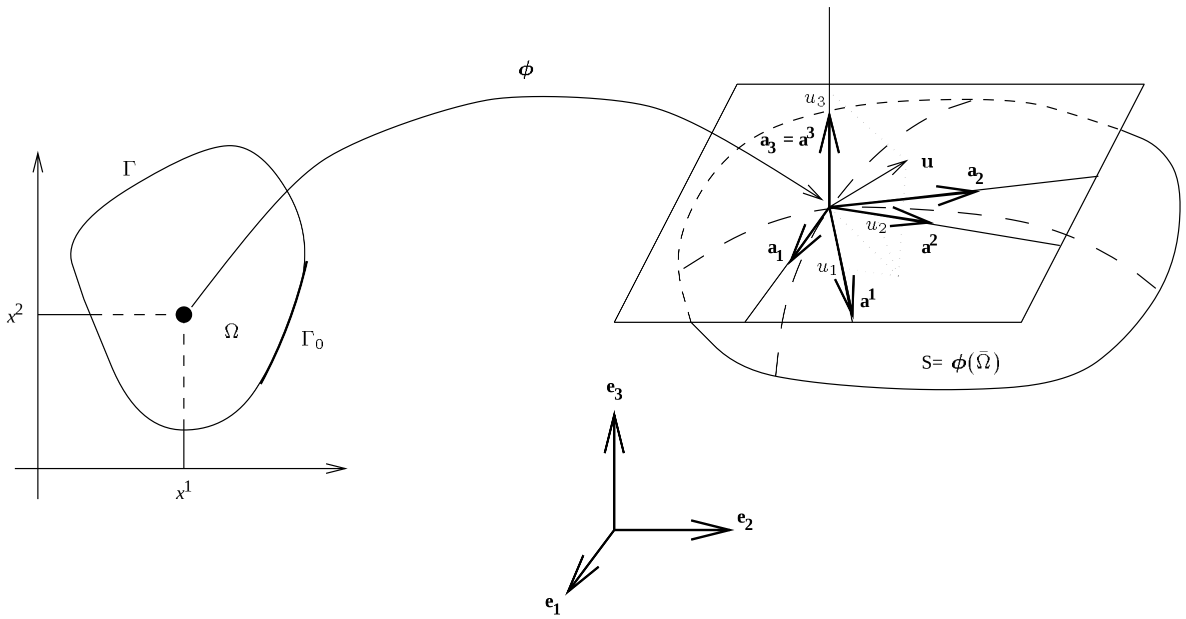


Figure 1: Definition of the middle surface S.

Thus the three vectors $\mathbf{a}_i$ define the *covariant basis* at the point $\boldsymbol{\phi}(x^1,\, x^2)$. We also define the *contravariant basis* $(\mathbf{a}^i)$ at the same point by the relations

$$\mathbf{a}^i \cdot \mathbf{a}_j = \delta^i_j \ , \tag{2.3}$$

from which follows that $\mathbf{a}^3 = \mathbf{a}_3$ (cf. Fig.1).

The *first fundamental form* $(a_{\alpha\beta})$ of the surface S is defined by

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \ . \tag{2.4}$$

Using the first fundamental form, or *metric tensor*, we can give the area element dS along S by

$$dS = \sqrt{a}dx^1 dx^2 \ ,$$

where due to the regularity assumption the determinant

$$a = \det(a_{\alpha\beta}) \tag{2.5}$$

does not vanish in the domain $\bar{\Omega}$. Now we define the fundamental metric tensor $(a^{\alpha\beta})$ by

$$a^{\alpha\beta} = \mathbf{a}^\alpha \cdot \mathbf{a}^\beta \tag{2.6}$$

Thus the vectors $\mathbf{a}^\alpha$ are related to the vectors $\mathbf{a}_\alpha$ by

$$\mathbf{a}_\alpha = a_{\alpha\beta}\mathbf{a}^\beta, \ \mathbf{a}^\alpha = a^{\alpha\beta}\mathbf{a}_\beta \ .$$

By the use of the fundamental metric tensors $(a_{\alpha\beta})$ and $(a^{\alpha\beta})$ we can associate the *covariant, contravariant and mixed components* of a given surface tensor. Therefore, with the covariant components $t_{\alpha\beta}$ of an arbitrary surface tensor of the order 2, we can associate the corresponding mixed and contravariant components

$$t_\alpha^\beta = a^{\beta\lambda}t_{\lambda\alpha} \ , \ t^{\alpha\beta} = a^{\alpha\lambda}a^{\beta\nu}t_{\lambda\nu} \ , \tag{2.7}$$

and, conversely,

$$t_{\alpha\beta} = a_{\alpha\lambda}t_\beta^\lambda = a_{\alpha\lambda}a_{\beta\nu}t^{\lambda\nu} \ . \tag{2.8}$$

The *covariant* components of the *second fundamental form* $(b_{\alpha\beta})$ of the surface S are defined by

$$b_{\alpha\beta} = \mathbf{a}_3 \cdot \partial_\beta \mathbf{a}_\alpha \ , \tag{2.9}$$

and with (2.7) its *mixed* components are

$$b_\alpha^\beta = a^{\beta\lambda}b_{\lambda\alpha} \ . \tag{2.10}$$

The *covariant* components of the *third fundamental form* $(c_{\alpha\beta})$ of the surface S are defined by

$$c_{\alpha\beta} = b_{\alpha\lambda}b_\beta^\lambda \ . \tag{2.11}$$

Remark. The three fundamental forms are *symmetric* with respect to their indices.

In general the bases $(\mathbf{a}_1, \ \mathbf{a}_2, \ \mathbf{a}_3)$ and $(\mathbf{a}^1, \ \mathbf{a}^2, \ \mathbf{a}^3)$ are neither normalized nor orthogonal. In order to calculate the derivatives of the basis vectors, we introduce the *Christoffel symbols of the second type* $\Gamma_{\alpha\beta}^\varrho$:

$$\Gamma_{\alpha\beta}^\varrho = \Gamma_{\beta\alpha}^\varrho = \mathbf{a}^\varrho \cdot \partial_\beta \mathbf{a}_\alpha \ . \tag{2.12}$$

Using the Christoffel symbols we can give the rules of *covariant differentiation* for a surface tensor. For a surface tensor of order 1, we obtain the covariant derivatives (denoted by a vertical bar) from their ordinary derivatives (denoted by a comma) as follows:

$$\left\{ \begin{array}{lcl} t_{\alpha|\gamma} &=& t_{\alpha,\gamma} - \Gamma_{\alpha\gamma}^\lambda t_\lambda \ , \\ t^\alpha|_\gamma &=& t_{,\gamma}^\alpha + \Gamma_{\lambda\gamma}^\alpha t^\lambda \ , \end{array} \right. \tag{2.13}$$

and for a tensor of order 2 we get:

$$\left\{ \begin{array}{lcl} t_{\alpha\beta|\gamma} &=& t_{\alpha\beta,\gamma} - \Gamma_{\alpha\gamma}^\lambda t_{\lambda\beta} - \Gamma_{\beta\gamma}^\lambda t_{\alpha\lambda} \ , \\ t_{\alpha|\gamma}^\beta &=& t_{\alpha,\gamma}^\beta + \Gamma_{\gamma\lambda}^\beta t_\alpha^\lambda - \Gamma_{\alpha\gamma}^\lambda t_\lambda^\beta \ , \\ t^{\alpha\beta}|_\gamma &=& t_{,\gamma}^{\alpha\beta} + \Gamma_{\lambda\gamma}^\alpha t^{\lambda\beta} + \Gamma_{\lambda\gamma}^\beta t^{\alpha\lambda} \ . \end{array} \right. \tag{2.14}$$

We also need the following rule of covariant differentiation of the normal component t_3:

$$\left\{ \ t_{3|\gamma\theta} \ = \ t_{3,\gamma\theta} - \Gamma_{\gamma\theta}^\lambda t_{3,\lambda} \ . \right. \tag{2.15}$$

2.2 Geometrical definition of the unstrained shell

We now define a thin shell $\mathcal{C}$, with the middle surface S, and with a constant *thickness* $\varepsilon > 0$, as a closed subset of $\mathbf{R}^3$ by:

$$\mathcal{C} = \{\mathbf{r} \in \mathbf{R}^3;\ \mathbf{r} = \boldsymbol{\phi}(x^1, x^2) + x^3\mathbf{a}_3(x^1, x^2),\ (x^1, x^2) \in \bar{\Omega},\ |x^3| \le \frac{\varepsilon}{2}\}\ .$$

The derivatives of the vector $\mathbf{r} = \boldsymbol{\phi}(x^1, x^2) + x^3\mathbf{a}_3(x^1, x^2)$ define the basis vectors of the continous medium $\mathbf{g}_i$ and satisfy

$$\mathbf{g}_\alpha = \partial_\alpha \mathbf{r} = (\delta_\alpha^\lambda - x^3 b_\alpha^\lambda)\mathbf{a}_\lambda\ ,\ \mathbf{g}_3 = \partial_3 \mathbf{r} = \mathbf{a}_3\ . \tag{2.16}$$

While the vectors $\mathbf{g}_1$ and $\mathbf{g}_2$ are parallel to the tangent plane to the middle surface at the point $\boldsymbol{\phi}(x^1, x^2)$, the vector $\mathbf{g}_3$ is normal to this plane. We now define the *metric tensor of the continuous medium* (g_{ij})

$$g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j \tag{2.17}$$

and the inverse (g^{ij}). Because of (2.16) we get in particular:

$$g^{\alpha 3} = g^{3\alpha} = 0,\ g^{33} = 1\ . \tag{2.18}$$

We assume that the shell $\mathcal{C}$ is *clamped* at the part $G_0 = \boldsymbol{\phi}(\Gamma_0) \times [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$ of its lateral surface, where $meas(\Gamma_0) > 0$. We further assume that there are no applied surface forces on the remaining part of the lateral surface. Finally, we assume that there are surface forces applying on the upper and lower faces of the shell and body forces applying in its interior.

3 The linear model of Koiter

In the following we assume that the displacements and deformations are "small" so that we can use linearized equations.

Koiter's approach for the thin shell theory [8] is based on the reduction of investigations of the complete displacement field to the determination of a displacement field $\mathbf{u} = u_i\mathbf{a}^i$ of the points of the middle surface S (this means that $\mathbf{u}(x^1, x^2)$ is the displacement of the point $\boldsymbol{\phi}(x^1, x^2)$ for all points $(x^1, x^2) \in \bar{\Omega}$; cf. Fig. 1). For this, Koiter uses two *basic hypotheses*:

(i) any point on a normal to the unstrained middle surface remains, after deformation, on the normal to the deformed middle surface;

(ii) the stresses are approximately plane and the stresses parallel to the the middle surface vary approximately linear across the thickness.

We use these hypotheses to get an approximation to the *strain tensor* ϵ_{ij} of the three-dimensional medium which is defined by $\epsilon_{ij} = \frac{1}{2}(\bar{g}_{ij} - g_{ij})$. Here $\bar{g}_{ij}$ and g_{ij} are the metric tensors of the continuous medium (2.17) in the strained and unstrained configurations for the same parametrization (x^1, x^2, x^3).

We now give a possible modelling according to [3]: The hypotheses of thin shells, small displacements, and small strains together with the first hypothesis (i) lead to:

$$\begin{aligned} \epsilon_{\alpha\beta} &= \frac{1}{2}(\bar{a}_{\alpha\beta} - a_{\alpha\beta}) - x^3(\bar{b}_{\alpha\beta} - b_{\alpha\beta})\ , \\ \epsilon_{\alpha 3} &= \epsilon_{3\alpha} = 0\ . \end{aligned}$$

Now we use *Hooke's law*: for an isotropic homogeneous elastic material we can write the relation of the three-dimensional *stress tensor* σ^{ij} to the strain tensor as

$$\sigma^{ij} = C^{ijkl}\epsilon_{kl}(\mathbf{U}) \ , \tag{3.1}$$

where the *coefficients of elasticity* are given by

$$C^{ijkl} = \frac{E}{2(1+\nu)}(g^{ik}g^{jl} + g^{il}g^{jk} + \frac{2\nu}{1-2\nu}g^{ij}g^{kl}) \ , \tag{3.2}$$

with *Young's modulus* E and the *Poisson coefficient* ν of the material. Using the equations (2.18), expression (3.2) gives

$$\begin{aligned}
C^{333\alpha} &= C^{3\alpha\beta\lambda} = 0 \ , \\
C^{33\alpha\beta} &= \frac{E\nu}{(1+\nu)(1-2\nu)}g^{\alpha\beta} \ , \\
C^{3\alpha3\beta} &= \frac{E}{2(1+\nu)}g^{\alpha\beta} \ , \\
C^{3333} &= \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}.
\end{aligned}$$

Hypothesis (ii) on *plane stresses* ($\sigma^{33} \equiv 0$) then leads to:

$$\epsilon_{33} = -\frac{\nu}{1-\nu}g^{\alpha\beta}\epsilon_{\alpha\beta} \ .$$

As these equations show, the shell strain tensor ϵ_{ij} is completely determined through the following two surface tensors:

(i) the *middle surface tensor* $\gamma_{\alpha\beta} = \frac{1}{2}(\bar{a}_{\alpha\beta} - a_{\alpha\beta})$;

(ii) the *middle surface change of curvature tensor* $\bar{\rho}_{\alpha\beta} = \bar{b}_{\alpha\beta} - b_{\alpha\beta}$.

On simplifying, we get the following expression for the tensor $\gamma_{\alpha\beta}$:

$$\gamma_{\alpha\beta}(\mathbf{u}) = \frac{1}{2}(u_{\alpha|\beta} + u_{\beta|\alpha}) - b_{\alpha\beta}u_3 \ ,$$

and for the tensor $\bar{\rho}_{\alpha\beta}$ we obtain:

$$\bar{\rho}_{\alpha\beta}(\mathbf{u}) = u_{3|\alpha\beta} + b^{\lambda}_{\alpha}u_{\lambda|\beta} + b^{\lambda}_{\beta}u_{\lambda|\alpha} + b^{\lambda}_{\alpha|\beta}u_{\lambda} - c_{\alpha\beta}u_3 \ .$$

Using (2.13),(2.14) and (2.15) we finally get the following expressions:

$$\gamma_{\alpha\beta}(\mathbf{u}) = \frac{1}{2}(\partial_\alpha u_\beta + \partial_\beta u_\alpha) - \Gamma^{\lambda}_{\alpha\beta}u_\lambda - b_{\alpha\beta}u_3 \tag{3.3}$$

$$\begin{aligned}
\bar{\rho}_{\alpha\beta}(\mathbf{u}) &= \partial_{\alpha\beta}u_3 - \Gamma^{\lambda}_{\alpha\beta}\partial_\lambda u_3 + b^{\lambda}_{\alpha}(\partial_\beta u_\lambda - \Gamma^{\mu}_{\lambda\beta}u_\mu) + b^{\lambda}_{\beta}(\partial_\alpha u_\lambda - \Gamma^{\mu}_{\lambda\alpha}u_\mu) + \\
&\quad (\partial_\alpha b^{\lambda}_{\beta} + \Gamma^{\lambda}_{\alpha\mu}b^{\mu}_{\beta} - \Gamma^{\mu}_{\alpha\beta}b^{\lambda}_{\mu})u_\lambda - c_{\alpha\beta}u_3 \ .
\end{aligned} \tag{3.4}$$

Putting the equations (3.1) and (3.2) into the formula of the *strain energy* (cf. [9]) and using the properties of C^{ijkl} mentioned above, the strain energy of the shell associated with the field of displacement $\mathbf{U}$ is

$$\begin{aligned}
F(\mathbf{U}) &= \frac{1}{2}\int_{\mathcal{C}} \sigma^{ij}\epsilon_{ij}(\mathbf{U})d\mathcal{C} \\
&= \frac{1}{2}\int_{\mathcal{C}} C^{ijkl}\epsilon_{ij}(\mathbf{U})\epsilon_{kl}(\mathbf{U})d\mathcal{C} \\
F(\mathbf{U}) &= \frac{1}{2}\int_{\mathcal{C}} B^{\alpha\beta\lambda\mu}\epsilon_{\alpha\beta}(\mathbf{U})\epsilon_{\lambda\mu}(\mathbf{U})d\mathcal{C} \ ,
\end{aligned}$$

where

$$B^{\alpha\beta\lambda\mu} = \frac{E}{2(1+\nu)}(g^{\alpha\lambda}g^{\beta\mu} + g^{\alpha\mu}g^{\beta\lambda} + \frac{2\nu}{1-\nu}g^{\alpha\beta}g^{\lambda\mu}) .$$

We integrate over the thickness and obtain the following approximation:

$$F(\mathbf{U}) \simeq \frac{1}{2}\int_\Omega \varepsilon E^{\alpha\beta\lambda\mu}\{\gamma_{\alpha\beta}(\mathbf{u})\gamma_{\lambda\mu}(\mathbf{u}) + \frac{\varepsilon^2}{12}\bar{\rho}_{\alpha\beta}(\mathbf{u})\bar{\rho}_{\lambda\mu}(\mathbf{u})\}\sqrt{a}dx^1dx^2 ,$$

where the elasticity coefficients are

$$E^{\alpha\beta\lambda\mu} = \frac{E}{2(1+\nu)}(a^{\alpha\lambda}a^{\beta\mu} + a^{\alpha\mu}a^{\beta\lambda} + \frac{2\nu}{1-\nu}a^{\alpha\beta}a^{\lambda\mu}) . \tag{3.5}$$

We associate the following bilinear form with this approximation:

$$a(\mathbf{u},\mathbf{v}) = \int_\Omega \varepsilon E^{\alpha\beta\lambda\mu}\{\gamma_{\alpha\beta}(\mathbf{u})\gamma_{\lambda\mu}(\mathbf{v}) + \frac{\varepsilon^2}{12}\bar{\rho}_{\alpha\beta}(\mathbf{u})\bar{\rho}_{\lambda\mu}(\mathbf{v})\}\sqrt{a}dx^1dx^2 , \tag{3.6}$$

The linear form f is given by

$$f(\mathbf{u}) = \int_\Omega \mathbf{p}\cdot\mathbf{u}\sqrt{a}dx^1dx^2 , \tag{3.7}$$

where $\mathbf{p} : \Omega \to \mathbf{R}^3$ is the vector field that we can calculate from the applied surface forces and by integration of the body forces over the thickness of the shell.

Then we obtain the following variational formulation for the unknown $\mathbf{u}$:

$$Find\, \mathbf{u} \in \mathbf{V}_\Gamma \; such\, that\; a(\mathbf{u},\mathbf{v}) = f(\mathbf{v}) ,\forall \mathbf{v} \in \mathbf{V}_\Gamma , \tag{3.8}$$

where the space $\mathbf{V}_\Gamma$ is defined as

$$\begin{aligned} \mathbf{V}_\Gamma &= \{\mathbf{v}=(v_\alpha, v_3) \in \mathbf{V} \;:\; \mathbf{v}|_{\Gamma_0} = \mathbf{0}, \frac{\partial v_3}{\partial\eta}|_{\Gamma_0} = 0\}, \qquad (3.9)\\ \mathbf{V} &= (H^1(\Omega))^2 \times H^2(\Omega) , \end{aligned}$$

therefore the corresponding norm is

$$\|\mathbf{v}\|_\mathbf{V} = \|\mathbf{v}\|_{(H^1(\Omega))^2\times H^2(\Omega)} = \{\sum_{\alpha=1}^{2}\|v_\alpha\|^2_{(H^1(\Omega))^2} + \|v_3\|^2_{H^2(\Omega)}\}^{\frac{1}{2}} . \tag{3.10}$$

The existence and uniqueness of a solution for the problem (3.8) is proved in e.g. [3]:

Theorem 3.1 *Assume that* $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$, *meas*$(\Gamma_0) > 0$ *and* $\mathbf{p} \in (L^2(\Omega))^3$. *Then the bilinear form (3.6) is continuous and* $\mathbf{V}_\Gamma$*-elliptic, the linear form (3.7) is continuous and hence the problem (3.8) has a unique solution.*

4 Finite element discretization

In this section we introduce one conforming and one nonconforming discretization for the shell equations (3.8), which are based on rectangular finite elements. We now suppose that $\Omega \subset \mathbf{R}^2$ is a bounded domain equipped with an appropriate partition $\mathcal{T}_h$ into a finite number of rectangles. We denote the corresponding set of nodes with $\mathcal{N}_h$.

4.1 Discretization by conforming Bogner-Fox-Schmidt elements

A conforming discretization requires the use of finite elements of the class C^0 for the approximation of the tangential displacement components u_1, u_2 and finite elements of the class C^1 for the approximation of the normal displacement u_3. In order to avoid the effect of *membran locking* we use the bicubic Bogner-Fox-Schmidt (BFS) C^1 element (cf. Fig. 2a) for all three displacement components. Such a procedure was proposed in [10]. Then we get the finite element space

$$\begin{aligned} \mathbf{V}_h &= V_h \times V_h \times V_h \ , \\ V_h &= \left\{ v_h \in C^1(\bar{\Omega}) \ : \ v_h|_K \in P_3(K), \ \forall K \in \mathcal{T}_h \right\} , \\ P_3(K) &= \left\{ p(x) = \sum_{k_1,k_2 \le 3} \gamma_{k_1 k_2} x_1^{k_1} x_2^{k_2} \right\} . \end{aligned} \tag{4.1}$$

Let τ denote the set $\tau = \{(0,0),(1,0),(0,1),(1,1)\}$ where the index $\alpha = (\alpha_1, \alpha_2) \in \tau$ corresponds for $\alpha = \mathbf{0} = (0,0)$ with the value of the function, for $\alpha = \mathbf{e}_1 = (1,0)$ and $\alpha = \mathbf{e}_2 = (0,1)$ with the values of the first derivatives, and for $\alpha = \mathbf{1} = (1,1)$ with the value of the mixed derivative in the node. Using the finite element nodal basis

$$\begin{aligned} \Phi_h &= \left(\varphi_{(i,\alpha,P)}\right)_{(i,\alpha,P)\in\{1,2,3\}\times\tau\times\mathcal{N}_h} , \\ \partial_1^{\alpha_1}\partial_2^{\alpha_2}\varphi_{(j,\beta,Q)}\big|_{i,P} &= \delta_{ij}\delta_{\alpha\beta}\delta_{PQ} \ , \ \forall (i,\alpha,P), \ (j,\beta,Q) \in \{1,2,3\} \times \tau \times \mathcal{N}_h \end{aligned}$$

every function $\mathbf{v}_h \in \mathbf{V}_h$ is uniquely defined from the vector of the nodal values $\underline{\mathbf{v}}_h \in \mathbf{R}^{3\cdot 4\cdot|\mathcal{N}_h|}$:

$$\mathbf{V}_h \ni \mathbf{v}_h = \Phi_h \underline{\mathbf{v}}_h \longleftrightarrow \underline{\mathbf{v}}_h \in \mathbf{R}^{3\cdot 4\cdot|\mathcal{N}_h|} \ . \tag{4.2}$$

Now we replace the variational problem (3.8) by the discrete variational problem

$$\textit{Find}\ \mathbf{u}_h \in \mathbf{V}_h \ \textit{such that}\ a(\mathbf{u}_h, \mathbf{v}_h) = f(\mathbf{v}_h) \ , \forall \mathbf{v}_h \in \mathbf{V}_h \ , \tag{4.3}$$

where

$$\mathbf{V}_h = \mathrm{span}(\Phi_h) \subset \mathbf{V}_\Gamma \ . \tag{4.4}$$

The solution of the variational problem (4.3) in $\mathbf{V}_h$ leads to a finite element equation system with a symmetric, positive definite stiffness matrix $\mathbf{K}_h$ for the calculation of the vector of the nodal values of the approximate solution $\mathbf{u}_h$:

$$\mathbf{K}_h \underline{\mathbf{u}}_h = \underline{\mathbf{f}}_h \ . \tag{4.5}$$

Due to [5] the discretization error can be estimated by the following theorem:

Theorem 4.1 *Let* $\mathbf{u}$ *be the solution of of the variational problem (3.8) and* $\mathbf{u}_h$ *be the solution of the discrete variational problem (4.3).*

Then for $\mathbf{u} \in H^2 \times H^2 \times H^3$ *the discretization error* $\|\mathbf{u} - \mathbf{u}_h\|_{\mathbf{V}}$ *is of the order* $\mathcal{O}(h^1)$. *In addition, we assume that the geometry of the shell can be exactly approximated in the finite element space. Then for* $\mathbf{u} \in H^3 \times H^3 \times H^4$ *the discretization error* $\|\mathbf{u} - \mathbf{u}_h\|_{\mathbf{V}}$ *is of the order* $\mathcal{O}(h^2)$.

4.2 Discretization by nonconforming Adini elements

We now define the nonconforming Adini element as a counterpart to the conforming BFS element by decreasing the polynomial degree in such a way that every node contains only three degrees of freedom. Since the Adini element (cf. Fig. 2b) is of the class C^0 the resulting discretization is conforming according to the tangential displacement components u_1, u_2 and nonconforming according to the normal displacement component u_3. The resulting finite element space is

$$\begin{aligned} \tilde{\mathbf{V}}_h &= \tilde{V}_h \times \tilde{V}_h \times \tilde{V}_h \ , \\ \tilde{V}_h &= \left\{ \tilde{v}_h \in C^0(\bar{\Omega}) \ : \ \tilde{v}_h|_K \in \tilde{P}_3(K), \ \forall K \in \mathcal{T}_h \right\} , \\ \tilde{P}_3(K) &= \left\{ p(x) = \sum_{k_1,k_2 \le 3, \ \min(k_1,k_2)\le 1} \gamma_{k_1 k_2} x_1^{k_1} x_2^{k_2} \right\} , \\ \tilde{\tau} &= \{(0,0),(1,0),(0,1),\} = \{\mathbf{0}, \ \mathbf{e}_1, \ \mathbf{e}_2\} \ . \end{aligned} \tag{4.6}$$

We introduce the finite element nodal basis

$$\tilde{\Phi}_h = \left(\tilde{\varphi}_{(i,\alpha,P)} \right)_{(i,\alpha,P)\in\{1,2,3\}\times\tilde{\tau}\times\mathcal{N}_h} ,$$

which leads to the isomorphism

$$\tilde{\mathbf{V}}_h \ni \tilde{\mathbf{v}}_h = \tilde{\Phi}_h \underline{\tilde{\mathbf{v}}}_h \longleftrightarrow \underline{\tilde{\mathbf{v}}}_h \in \mathbf{R}^{3\cdot 3\cdot|\mathcal{N}_h|} \ .$$

We replace the variational problem (3.8) by the discrete variational problem

$$\mathit{Find}\ \tilde{\mathbf{u}}_h \in \tilde{\mathbf{V}}_h\ \mathit{such\,that}\ \tilde{a}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = f(\tilde{\mathbf{v}}_h) \ , \forall \tilde{\mathbf{v}}_h \in \tilde{\mathbf{V}}_h \ , \tag{4.7}$$

where the space $\tilde{\mathbf{V}}_h$ is modified according to the Dirichlet boundary conditions:

$$\tilde{\mathbf{V}}_h = \{ \tilde{\mathbf{v}}_h \in \tilde{\mathbf{V}}_h \ : \ \tilde{\mathbf{v}}_h(P) = \mathbf{0}, \ \frac{\partial \tilde{v}_{h,3}}{\partial t}(P) = \frac{\partial \tilde{v}_{h,3}}{\partial \eta}(P) = 0, \forall P \in \mathcal{N}_h \cap \Gamma_0 \} \ . \tag{4.8}$$

Since the space $\tilde{V}_h$ is not a subspace of the Sobolev space $H^2(\Omega)$ we introduce the following seminorms:

$$|\tilde{v}_{h,i}|_{\beta,\mathcal{T}_h} = \left\{ \sum_{K\in\mathcal{T}_h} |\tilde{v}_{h,i}|^2_{H^\beta(K)} \right\}^{\frac{1}{2}}, \tag{4.9}$$

$$|\tilde{\mathbf{v}}_h|_{\mathcal{T}_h} = \left\{ \sum_{\alpha=1}^{2} |v_{h,\alpha}|^2_{1,\mathcal{T}_h} + |v_{h,3}|^2_{2,\mathcal{T}_h} \right\}^{\frac{1}{2}} . \tag{4.10}$$

In the formulation (4.7) the bilinear form (3.6) is replaced by

$$\tilde{a}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = \sum_{K\in\mathcal{T}_h} \int_K \varepsilon E^{\alpha\beta\lambda\mu} \{ \gamma_{\alpha\beta}(\tilde{\mathbf{u}}_h)\gamma_{\lambda\mu}(\tilde{\mathbf{v}}_h) + \frac{\varepsilon^2}{12} \bar{\rho}_{\alpha\beta}(\tilde{\mathbf{u}}_h)\bar{\rho}_{\lambda\mu}(\tilde{\mathbf{v}}_h) \} \sqrt{a} dx^1 dx^2 \ . \tag{4.11}$$

Theorem 4.2 *Assume that $\phi \in \left(\mathcal{C}^3(\bar{\Omega})\right)^3$. Then the bilinear form (4.11) is continuous and elliptic in the space $\tilde{\mathbf{V}}_h$ with respect to the seminorm (4.10):*

$$\tilde{a}(\tilde{\mathbf{v}}_h, \tilde{\mathbf{v}}_h) \ge \tilde{\mu}_E |\tilde{\mathbf{v}}_h|^2_{\mathcal{T}_h}, \tag{4.12}$$

$$|\tilde{a}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h)| \le \tilde{\mu}_C |\tilde{\mathbf{u}}_h|_{\mathcal{T}_h} |\tilde{\mathbf{v}}_h|_{\mathcal{T}_h} \ . \tag{4.13}$$

In order to prove the theorem, we now use three lemmas which were shown in [3] to prove Theorem 4.1.

Lemma 4.1 *Suppose $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$. Then there exists a constant $c_1 > 0$ such that*

$$a(\mathbf{v},\mathbf{v}) \geq c_1 \Big\{ \sum_{\alpha,\beta=1}^{2} \|\gamma_{\alpha\beta}(\mathbf{v})\|^2_{L^2(\Omega)} + \sum_{\alpha,\beta=1}^{2} \|\bar{\rho}_{\alpha\beta}(\mathbf{v})\|^2_{L^2(\Omega)} \Big\} \tag{4.14}$$

for all $\mathbf{v} \in \mathbf{V}$.

Lemma 4.2 *Let $\mathbf{E}(\Omega)$ be the space defined by the relation*

$$\begin{cases} \mathbf{E}(\Omega) = \{\mathbf{v} = (v_\alpha, v_3) \in (L^2(\Omega))^2 \times H^1(\Omega) \ : \\ \qquad \gamma_{\alpha\beta}(\mathbf{v}) \in L^2(\Omega),\ \bar{\rho}_{\alpha\beta}(\mathbf{v}) \in L^2(\Omega)\}. \end{cases} \tag{4.15}$$

We assume $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$. Then $\mathbf{E}(\Omega) = (H^1(\Omega))^2 \times H^2(\Omega) = \mathbf{V}$.

Lemma 4.3 *Assume that $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$. Then there exist two strictly positive constants c_2 and c_3 such that*

$$c_2 \|\mathbf{v}\|_{\mathbf{V}} \leq \|\mathbf{v}\|_{\mathbf{E}(\Omega)} \leq c_3 \|\mathbf{v}\|_{\mathbf{V}},\ \ \forall \mathbf{v} \in \mathbf{V}, \tag{4.16}$$

where

$$\|\mathbf{v}\|_{\mathbf{E}(\Omega)} = \Big\{ \sum_{\alpha=1}^{2} \|v_\alpha\|^2_{L^2(\Omega)} + \|v_3\|^2_{H^1(\Omega)} + \sum_{\alpha,\beta=1}^{2} \|\gamma_{\alpha\beta}(\mathbf{v})\|^2_{L^2(\Omega)} + \sum_{\alpha,\beta=1}^{2} \|\bar{\rho}_{\alpha\beta}(\mathbf{v})\|^2_{L^2(\Omega)} \Big\}^{\frac{1}{2}}. \tag{4.17}$$

In other words, $\|\cdot\|_{\mathbf{E}(\Omega)}$ and $\|\cdot\|_{\mathbf{V}}$ are equivalent norms on the space $\mathbf{V}$.

Now we define the following two seminorms in the spaces $\mathbf{V}$ rsp. $\mathbf{E}(\Omega)$:

$$|\mathbf{v}|_{\mathbf{V}} = |\mathbf{v}|_{(H^1(\Omega))^2 \times H^2(\Omega)} = \{ \sum_{\alpha=1}^{2} |v_\alpha|^2_{(H^1(\Omega))^2} + |v_3|^2_{H^2(\Omega)} \}^{\frac{1}{2}}, \tag{4.18}$$

$$|\mathbf{v}|_{\mathbf{E}(\Omega)} = \Big\{ \sum_{\alpha,\beta=1}^{2} \|\gamma_{\alpha\beta}(\mathbf{v})\|^2_{L^2(\Omega)} + \sum_{\alpha,\beta=1}^{2} \|\bar{\rho}_{\alpha\beta}(\mathbf{v})\|^2_{L^2(\Omega)} \Big\}^{\frac{1}{2}}. \tag{4.19}$$

Then we obtain from Lemma 4.3

Lemma 4.4 *Assume that $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$. Then the following two-sided inequality is guilty:*

$$c_2 |\mathbf{v}|_{\mathbf{V}} \leq |\mathbf{v}|_{\mathbf{E}(\Omega)} \leq c_3 |\mathbf{v}|_{\mathbf{V}},\ \ \forall \mathbf{v} \in \mathbf{V}, \tag{4.20}$$

where the constants c_2 and c_3 are taken from Lemma 4.3.

Finally, using the seminorm (4.19) the following lemma can be shown.

Lemma 4.5 *Assume that $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$. Then the following inequality holds:*

$$|a(\mathbf{u},\mathbf{v})| \leq c_4 |\mathbf{u}|_{\mathbf{E}(\Omega)} |\mathbf{v}|_{\mathbf{E}(\Omega)},\ \ \forall \mathbf{u},\ \mathbf{v} \in \mathbf{V}\ , \tag{4.21}$$

where c_4 denotes a strictly positive constant.

Putting (4.20) into (4.14) and (4.21) we get the following lemma about the ellipticity and continuitity of the bilinear form (3.6) in the space $\mathbf{V}$:

Lemma 4.6 *Assume that $\boldsymbol{\phi} \in (\mathcal{C}^3(\bar{\Omega}))^3$. Then the bilinear form (3.6) is continuous and elliptic in the space* $\mathbf{V}$ *with respect to the seminorm (4.18):*

$$
\begin{aligned}
a(\mathbf{v}, \mathbf{v}) &\geq \mu_E |\mathbf{v}|^2_{\mathbf{V}}, && (4.22)\\
|a(\mathbf{u}, \mathbf{v})| &\leq \mu_C |\mathbf{u}|_{\mathbf{V}} |\mathbf{v}|_{\mathbf{V}} \ . && (4.23)
\end{aligned}
$$

In order to prove Theorem 4.2 we have only to apply Lemma 4.6 to all rectangles $K \in \mathcal{T}_h$, i.e. $\Omega = K$, and to take the sum over all of these rectangles.

Remark. We did not use the full norm in Theorem 4.2 because there are no Dirichlet boundary conditions in the interior rectangles K of the triangulation $\mathcal{T}_h$ and therefore Theorem 3.1 is not applicable to these interior rectangles. So we had to show Lemma 4.6 which can be applied to all rectangles $K \in \mathcal{T}_h$.

The solution of the variational problem (4.7) in the space $\tilde{\mathbf{V}}_h$ leads to a finite element equation system with the s.p.d. stiffness matrix $\tilde{\mathbf{K}}_h$ for the calculation of the vector of the nodal values of the approximate solution $\tilde{\mathbf{u}}_h$:

$$\tilde{\mathbf{K}}_h \underline{\tilde{\mathbf{u}}}_h = \underline{\mathbf{f}}_h \ . \qquad (4.24)$$

Since we could not find a general estimation of the discretization error of the Adini elements for Koiters model in the literature (for the plate such estimation is given in [5]) we leave this as an open problem. In practice (see Subsection 7.2) the discrete solution $\tilde{\mathbf{u}}_h$ of (4.7) seams to converge to the solution $\mathbf{u}$ of the initial problem (3.8).

4.3 Relations between the spaces of the BFS and Adini elements

Let us define two mappings between the finite element spaces of the BFS elements $\mathbf{V}_h$ and the Adini elements $\tilde{\mathbf{V}}_h$ in the following natural way.

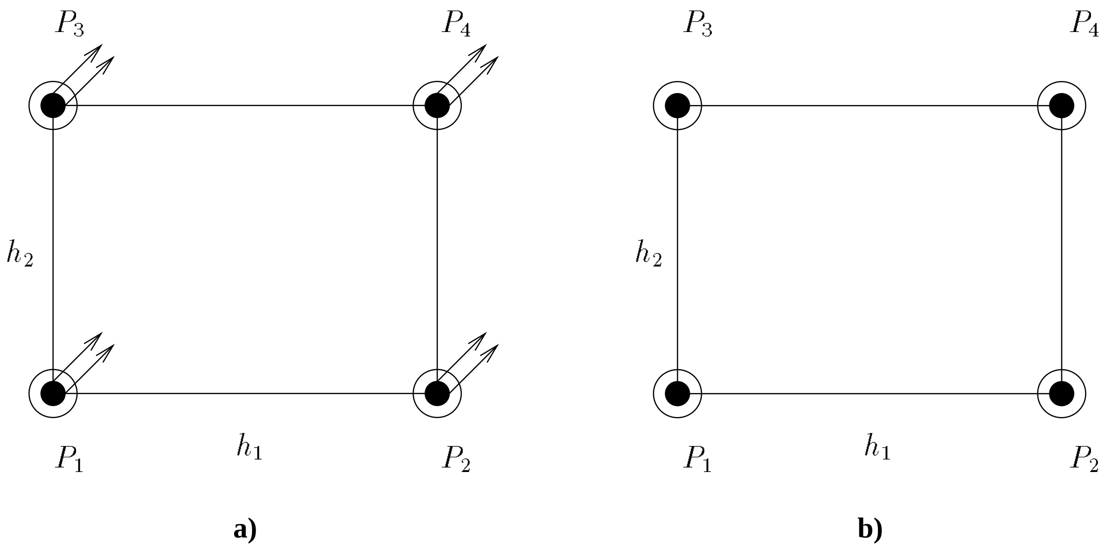


Figure 2: Bogner-Fox-Schmidt element and Adini element.

First we define the mapping $\boldsymbol{\mathcal{E}}_h : \ \tilde{\mathbf{V}}_h \rightarrow \mathbf{V}_h$ as

$$\boldsymbol{\mathcal{E}}_h \tilde{\mathbf{v}}_h = \{\mathcal{E}_h \tilde{v}_{h,i}\}_{i \in \{1,2,3\}} \ , \qquad (4.25)$$

where the mapping $\mathcal{E}_h : \tilde{V}_h \to V_h$ is defined by

$$\begin{aligned} \mathcal{E}_h \tilde{v}_h|_P &= \tilde{v}_h|_P \ , \\ \nabla \mathcal{E}_h \tilde{v}_h|_P &= \nabla \tilde{v}_h|_P \ , \\ \partial_{12} \mathcal{E}_h \tilde{v}_h|_P &= 0 \ , \ \forall P \in \mathcal{N}_h. \end{aligned} \tag{4.26}$$

Conversely, the mapping $\boldsymbol{\mathcal{F}}_h : \mathbf{V}_h \to \tilde{\mathbf{V}}_h$ is given by

$$\boldsymbol{\mathcal{F}}_h \mathbf{v}_h = \{\mathcal{F}_h v_{h,i}\}_{i \in \{1,2,3\}} \ , \tag{4.27}$$

where the mapping $\mathcal{F}_h : V_h \to \tilde{V}_h$ is the natural restriction

$$\begin{aligned} \mathcal{F}_h v_h|_P &= v_h|_P \ , \\ \nabla \mathcal{F}_h v_h|_P &= \nabla v_h|_P \ , \ \forall P \in \mathcal{N}_h. \end{aligned} \tag{4.28}$$

Obviously the operator $\boldsymbol{\mathcal{E}}_h$ is right-inverse to the operator $\boldsymbol{\mathcal{F}}_h$:

$$\boldsymbol{\mathcal{F}}_h \boldsymbol{\mathcal{E}}_h \tilde{\mathbf{v}}_h = \tilde{\mathbf{v}}_h. \tag{4.29}$$

The following lemma was proved in [10].

Lemma 4.7 *There exist such mesh size independent constants $\beta_{\mathcal{E}}$ and $\beta_{\mathcal{F}}$ that for all $\tilde{v}_h \in \tilde{V}_h$ and $v_h \in V_h$ we have*

$$|\mathcal{E}_h \tilde{v}_h|^2_{H^2(\Omega)} \leq \beta_{\mathcal{E}} |\tilde{v}_h|^2_{2,\mathcal{T}_h} \ , \tag{4.30}$$

$$|\mathcal{F}_h v_h|^2_{2,\mathcal{T}_h} \leq \beta_{\mathcal{F}} |v_h|^2_{H^2(\Omega)} \ . \tag{4.31}$$

Using this lemma we can prove Lemma 4.8 which we will need in Subsection 5.2 for the application of the fictitious space lemma to the construction of multilevel preconditioners for the finite element space of the Adini elements.

Lemma 4.8 *There exist such mesh size independent constants $\beta_{\boldsymbol{\mathcal{E}}}$ and $\beta_{\boldsymbol{\mathcal{F}}}$ that for all $\tilde{\mathbf{v}}_h \in \tilde{\mathbf{V}}_h$ and $\mathbf{v}_h \in \mathbf{V}_h$ the following estimations hold:*

$$|\boldsymbol{\mathcal{E}}_h \tilde{\mathbf{v}}_h|^2_{\mathbf{V}} \leq \beta_{\boldsymbol{\mathcal{E}}} |\tilde{\mathbf{v}}_h|^2_{\mathcal{T}_h} \ , \tag{4.32}$$

$$|\boldsymbol{\mathcal{F}}_h \mathbf{v}_h|^2_{\mathcal{T}_h} \leq \beta_{\boldsymbol{\mathcal{F}}} |\mathbf{v}_h|^2_{\mathbf{V}} \ . \tag{4.33}$$

Proof: By the use of Lemma 4.7 together with the fact that the functions of $\tilde{V}_h$ belong to $H^1(\Omega)$ we find:

$$\begin{aligned} |\boldsymbol{\mathcal{E}}_h \tilde{\mathbf{v}}_h|^2_{\mathbf{V}} &= \sum_{\alpha=1}^{2} |\mathcal{E}_h \tilde{v}_{h,\alpha}|^2_{H^1(\Omega)} + |\mathcal{E}_h \tilde{v}_{h,3}|^2_{H^2(\Omega)} \\ &= \sum_{\alpha=1}^{2} |\tilde{v}_{h,\alpha}|^2_{1,\mathcal{T}_h} + |\mathcal{E}_h \tilde{v}_{h,3}|^2_{H^2(\Omega)} \\ &\leq \sum_{\alpha=1}^{2} |\tilde{v}_{h,\alpha}|^2_{1,\mathcal{T}_h} + \beta_{\mathcal{E}} |\tilde{v}_{h,3}|^2_{2,\mathcal{T}_h} \\ &\leq \beta_{\boldsymbol{\mathcal{E}}} \underbrace{\Big(\sum_{\alpha=1}^{2} |\tilde{v}_{h,\alpha}|^2_{1,\mathcal{T}_h} + |\tilde{v}_{h,3}|^2_{2,\mathcal{T}_h} \Big)}_{|\tilde{\mathbf{v}}|^2_{\mathcal{T}_h}} \ . \end{aligned}$$

Inequality (4.32) is established. Similary we can prove the inequatity (4.33). ∎

5 Multilevel preconditioning

Suppose that Ω is equipped with an initial partition $\mathcal{T}_0$ into a finite number of rectangles, and generate $\mathcal{T}_1, \dots, \mathcal{T}_J = \mathcal{T}_h$ by dyadically refining the initial partition. Thus, $\mathcal{T}_l$ consists of rectangles similar to those contained in $\mathcal{T}_0$ but scaled by a factor 2^{-l}. The corrsponding sets of nodal points are $\mathcal{N}_1, \dots, \mathcal{N}_J = \mathcal{N}_h$.

5.1 Multilevel preconditioning for BFS elements

Using the BFS elements we obtain a sequence of finite element spaces $\mathbf{V}_l = \mathrm{span}\Phi_l$:

$$\mathbf{V}_0 \subset \mathbf{V}_1 \subset \dots \subset \mathbf{V}_J = \mathbf{V}_h. \tag{5.1}$$

Now we choose the following additive splitting of the space $\mathbf{V}_h$

$$\mathbf{V}_h = \sum_{l=0}^{J} \sum_{i=1}^{3} \sum_{\alpha \in \tau} \sum_{P \in \mathcal{N}_l} \mathbf{V}^l_{(i,\alpha,P)} \tag{5.2}$$

into onedimensional subspaces $\mathbf{V}^l_{(i,\alpha,P)} = \mathrm{span}(\varphi^l_{(i,\alpha,P)})$ equipped with the scalar product

$$(u,v)_{\mathbf{V}^l_{(i,\alpha,P)}} = \begin{cases} 2^{2l}(u,v)_{L_2}, & i = 1,2 \\ 2^{4l}(u,v)_{L_2}, & i = 3 \end{cases} \quad , \; u,v \in \mathbf{V}^l_{(i,\alpha,P)}$$

On the subspaces we choose auxilary bilinear forms. For the auxilary bilinear forms different choices are possible:

$$\begin{aligned} b^l_{(i,\alpha,P)}(u,v) &= (u,v)_{\mathbf{V}^l_{(i,\alpha,P)}} \; , \; u,v \in \mathbf{V}^l_{(i,\alpha,P)}, & (5.3) \\ b^l_{(i,\alpha,P)}(u,v) &= a(u,v) \; , \quad u,v \in \mathbf{V}^l_{(i,\alpha,P)} \; . & (5.4) \end{aligned}$$

The choice (5.3) leads to the BPX scheme [4] while (5.4) is called Multilevel Diagonal Scaling (MDS-BPX) scheme [16]. In order to show that our subspace splitting is *stable*, we will rewrite the subspace splitting in the following, more explicit way:

$$\{\mathbf{V}_h; a\} = \sum_{l=0}^{J} \sum_{i=1}^{3} \sum_{\alpha \in \tau} \sum_{P \in \mathcal{N}_l} \{\mathbf{V}^l_{(i,\alpha,P)}; b^l_{(i,\alpha,P)}\} \; . \tag{5.5}$$

The following Lemma shows that our subspace splitting is stable:

Lemma 5.1 *The splitting (5.5) is stable, i.e. the following inequality holds:*

$$a(\mathbf{u}_h, \mathbf{u}_h) \sim |||\mathbf{u}_h|||^2 \equiv \inf \sum_{l=0}^{J} \sum_{i=1}^{3} \sum_{\alpha \in \tau} \sum_{P \in \mathcal{N}_l} b^l_{(i,\alpha,P)}(u^l_{(i,\alpha,P)}, u^l_{(i,\alpha,P)}), \;\; \forall \mathbf{u}_h \in \mathbf{V}_h \tag{5.6}$$

where the infimum is taken over all splittings

$$\mathbf{u}_h = \sum_{l=0}^{J} \sum_{i=1}^{3} \sum_{\alpha \in \tau} \sum_{P \in \mathcal{N}_l} u^l_{(i,\alpha,P)} \; , \; u^l_{(i,\alpha,P)} \in \mathbf{V}^l_{(i,\alpha,P)} \tag{5.7}$$

and where the following estimation for the spectrum of the splitting (5.5) by positive constants $\underline{c}$ and $\overline{c}$, which are independent of the mesh size, holds:

$$\underline{c} \le \lambda_{min} = \min_{\mathbf{u}_h \in \mathbf{V}_h, \mathbf{u}_h \neq 0} \frac{a(\mathbf{u}_h, \mathbf{u}_h)}{|||\mathbf{u}_h|||^2} \le \lambda_{max} = \max_{\mathbf{u}_h \in \mathbf{V}_h, \mathbf{u}_h \neq 0} \frac{a(\mathbf{u}_h, \mathbf{u}_h)}{|||\mathbf{u}_h|||^2} \le \overline{c} \; . \tag{5.8}$$

Proof: The proof is similar to this of [10],[14]. It is based on the use of the decomposition theorem (Theorem 15 in [13]). Since the bilinear form (3.6) is $\mathbf{V}_\Gamma$-elliptic and continuous (Theorem 3.1) and $\mathbf{V}_h$ is a subspace of $\mathbf{V}_\Gamma$, the bilinear form can be replaced through the norm in $\mathbf{V}$:

$$c_1\|\mathbf{u}_h\|_{\mathbf{V}}^2 \le a(\mathbf{u}_h,\mathbf{u}_h) \le c_2\|\mathbf{u}_h\|_{\mathbf{V}}^2 \ , \forall \mathbf{u}_h \in \mathbf{V}_h \ . \tag{5.9}$$

For the auxilary bilinear forms the following estimation

$$c_3\|u^l_{(i,\alpha,P)}\|^2_{\mathbf{V}^l_{(i,\alpha,P)}} \le b^l_{(i,\alpha,P)}(u^l_{(i,\alpha,P)},u^l_{(i,\alpha,P)}) \le c_4\|u^l_{(i,\alpha,P)}\|^2_{\mathbf{V}^l_{(i,\alpha,P)}} \ , \tag{5.10}$$

with $c_3 = c_4 = 1$ for the bilinear form (5.3) and with geometric dependent constants for the bilinear form (5.4), holds true, and hence the bilinear forms can be replaced by the corresponding norms in $\mathbf{V}^l_{(i,\alpha,P)}$. Therefore, it remains to show the norm equivalence

$$c_5\|\mathbf{u}_h\|_{\mathbf{V}}^2 \le \inf \sum_{l=0}^{J}\sum_{i=1}^{3}\sum_{\alpha\in\tau}\sum_{P\in\mathcal{N}_l} \|u^l_{(i,\alpha,P)}\ \|^2_{\mathbf{V}^l_{(i,\alpha,P)}} \le c_6\|\mathbf{u}_h\|_{\mathbf{V}}^2 \ , \tag{5.11}$$

where the infimum is taken over all splittings (5.7). Obviously, then the constants are $\underline{c} = c_1\cdot c_3\cdot c_5$ and $\overline{c} = c_2\cdot c_4\cdot c_6$.

Denoting

$$\mathbf{u}_l = \sum_{i=1}^{3}\sum_{\alpha\in\tau}\sum_{P\in\mathcal{N}_l} u^l_{(i,\alpha,P)}$$

and using the basis property of Φ_l we have for each splitting (5.7) a corresponding unique splitting with

$$\mathbf{u}_h = \sum_{l=0}^{J}\mathbf{u}_l \ , \ \mathbf{u}_l \in \mathbf{V}_l$$

and vice versa. But from the L_2 stability of the basis Φ_l, we have

$$\|\mathbf{u}_l\|^2_{L_2} \sim \sum_{i=1}^{3}\sum_{\alpha\in\tau}\sum_{P\in\mathcal{N}_l} \|u^l_{(i,\alpha,P)}\|^2_{L_2} \ , \ \forall \mathbf{u}_l \in \mathbf{V}_l \ , \tag{5.12}$$

with constants in the two-sided inequality that depend only on the initial partition $\mathcal{T}_0$. Now we use the decomposition theorem with $s = 1,2$; for the finite element spaces $\{V_l\}_{l=1}^J$ spanned by BFS elements and $0 < s < \frac{5}{2}$ it reads as follows:

$$\|u_J\|^2_{H^s(\Omega)} \sim \inf_{u_J=\sum_{l=0}^J u_l, u_l\in V_l} \sum_{l=0}^{J} 2^{2sl}\|u_l\|^2_{L_2} \ , \ \forall u_J \in V_J \ . \tag{5.13}$$

Therefore, the constants c_5 and c_6 can be calculated from the L_2-stability constants of the basis and the constants of the decomposition theorem. ∎

Now we introduce the *additive Schwarz operator* $\mathcal{P}$ associated with the splitting (5.5) and consider the following operator equation:

$$\mathcal{P}\mathbf{u}_h = \mathbf{g}_h \ , \tag{5.14}$$

$$\mathcal{P}\mathbf{u}_h = \sum_{l=0}^{J}\sum_{i=1}^{3}\sum_{\alpha\in\tau}\sum_{P\in\mathcal{N}_l} \frac{a(\mathbf{u}_h,\varphi^l_{(i,\alpha,P)})}{b^l_{(i,\alpha,P)}(\varphi^l_{(i,\alpha,P)},\varphi^l_{(i,\alpha,P)})}\cdot\varphi^l_{(i,\alpha,P)} \ , \tag{5.15}$$

$$\mathbf{g}_h = \sum_{l=0}^{J}\sum_{i=1}^{3}\sum_{\alpha\in\tau}\sum_{P\in\mathcal{N}_l} \frac{f(\varphi^l_{(i,\alpha,P)})}{b^l_{(i,\alpha,P)}(\varphi^l_{(i,\alpha,P)},\varphi^l_{(i,\alpha,P)})}\cdot\varphi^l_{(i,\alpha,P)} \ .$$

The theorem of the additive Schwarz preconditioner [13] together with Lemma 5.1 immediatly leads to

Theorem 5.1 *The solution of the discrete variational problem (4.3) coincides with that of the operator equation (5.14). The Schwarz operator* $\mathcal{P} : \mathbf{V}_h \to \mathbf{V}_h$ *is symmetric and positive definite. Moreover,*

$$\underline{c} \leq \lambda_{min} = \lambda_{min}(\mathcal{P}) \leq \lambda_{max}(\mathcal{P}) = \lambda_{max} \leq \overline{c} \ . \tag{5.16}$$

Especially, the spectral condition number $\kappa(\mathcal{P})$ *is bounded by the mesh size independent constant* $\overline{c}/\underline{c}$.

Remarks. Note that

(i) the question of the robustness of the preconditioner according to the geometric parameters remains to be open, because in the proof of the Lemma 5.1 the constants c_1, c_2 and in the case of multilevel diagonal scaling the constants c_3, c_4 depend on the geometry of the shell,

(ii) due to the clustering trick (cf. [13]) we can collect all subspaces with the level number $\leq j_0$ into one subspace $\mathbf{V}_{j_0}$ $(0 \leq j_0 < J)$ and consider the modified splitting

$$\mathbf{V}_h = \mathbf{V}_{j_0} + \sum_{l=j_0}^{J} \sum_{i=1}^{3} \sum_{\alpha \in \tau} \sum_{P \in \mathcal{N}_l} \mathbf{V}^l_{(i,\alpha,P)} \ . \tag{5.17}$$

We equip the space $\mathbf{V}_{j_0}$ with the scalar product (3.10) and the auxilary bilinear form $b^{j_0} = a(.,.)$. Then the assertions of Theorem 5.1 remain true. In practice, the corresponding preconditioners are often advantegous since they include the exact solution of a coarse grid problem corresponding to the level j_0.

How the expression $\mathcal{P}\mathbf{u}_h$ in (5.15) can be interpreted? We can represent $\mathbf{u}_h = \mathbf{u}_J \in \mathbf{V}_h$ by its nodal basis coefficient vector $\underline{\mathbf{u}}_h$, see (4.2). The first step, the calculation of the values $a(\mathbf{u}_h, \varphi^J_{(i,\alpha,P)})$, is equivalent to the the multiplication $\mathbf{K}_h\underline{\mathbf{u}}_h$ by the stiffness matrix $\mathbf{K}_h$. In the second step, for $l = J-1, \ldots, 0$, we determine the remaining $a(\mathbf{u}_h, \varphi^l_{(i,\alpha,P)})$ recursivly by using the linear expressions of the basis functions from the next finer level (cf. Theorem (5.2)). In the third step we carry out the scaling in dependence on the choice of the auxilary bilinear forms (5.3) or (5.4). The fourth step consists of evaluating the nodal basis vector $\mathcal{P}\mathbf{u}_h \in \mathbf{V}_h$ by taking the sum in (5.15). Obviously, this last operation is adjoint to that in step 2. Steps 2-4 can be interpreted as a simple V-cycle; they represent the symmetric preconditioner $\mathbf{C}_h^{-1}$ (5.18). Therefore, the operator equation (5.14) is equivalent to the preconditioned equation system

$$\mathbf{C}_h^{-1}\mathbf{K}_h\underline{\mathbf{u}}_h = \mathbf{C}_h^{-1}\underline{\mathbf{f}}_h$$

and

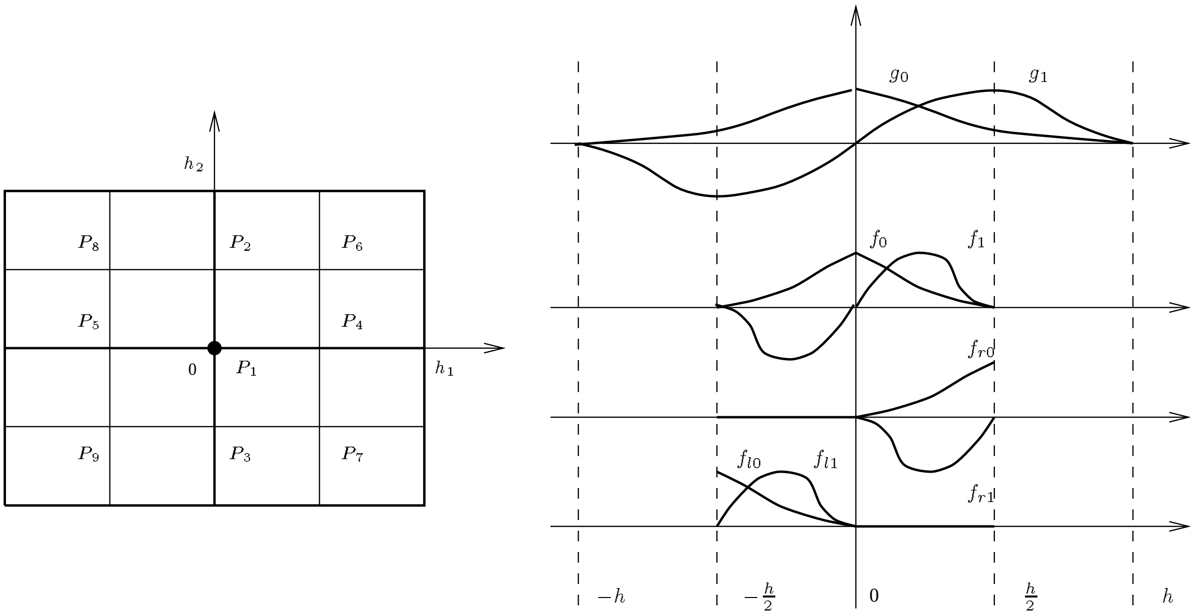
$$\mathbf{C}_h^{-1} = \mathbf{S}\mathbf{D}^{-1}\mathbf{S}^T \ . \tag{5.18}$$

Here $\mathbf{S}$ is the transformation matrix between the finite element basis Φ_J and the generating system $\Psi = (\Phi_0, \Phi_1, \ldots, \Phi_J)$ and $\mathbf{D}$ is the main diagonal of the scaled mass matrix rsp. the stiffness matrix of the generating system.

The matrix $\mathbf{S}$ can be written in the following form:

$$\begin{aligned} \mathbf{S} &= [\mathcal{I}_0^J|\mathcal{I}_1^J|\cdots|\mathcal{I}_{J-1}^J|\mathcal{I}_J] \ , \\ \mathcal{I}_l^J &= \mathcal{I}_{J-1}^J\mathcal{I}_{J-2}^{J-1}\cdot\ldots\cdot\mathcal{I}_l^{l+1} \ , \end{aligned}$$

where the interpolation matrix $\mathcal{I}_l^{l+1}$ is the transpose of the restriction matrix $\mathcal{I}_{l+1}^l$. In order to explain the restriction matrix we consider one node $P_1 \in \mathcal{N}_l$ of the discretization level l with the discretization parameters h_1 in x-direction and h_2 in y-direction together with its neighbouring nodes $P_1, \ldots, P_9 \in \mathcal{N}_{l+1}$ of the discretization level $l+1$ (cf. Fig. 3a).



a) **b)**

Figure 3: Node P_1 with neighbouring nodes and the one dimensional basis functions of two consecutive levels.

Theorem 5.2 *The restriction* $\underline{w}^l = \mathcal{I}^l_{l+1}\underline{w}^{l+1}$ *for all nodes of the level* $l+1$ *is given by:*

$$\underline{w}^l_{i,P_1} = B^T(\underline{w}^{l+1}_{i,P_j})_{j=1,\ldots,9} \ \forall i \in \{1,2,3\}\ , \tag{5.19}$$

with the basis transformation matrix

$$B = \begin{pmatrix} I & \otimes & I \\ A_1^+ & \otimes & I \\ A_1^- & \otimes & I \\ I & \otimes & A_2^+ \\ I & \otimes & A_2^- \\ A_1^+ & \otimes & A_2^+ \\ A_1^- & \otimes & A_2^+ \\ A_2^+ & \otimes & A_2^- \\ A_2^- & \otimes & A_2^- \end{pmatrix}\ , \tag{5.20}$$

where $\otimes$ *denotes the Kronecker product and the transformation matrices* $A^\pm_\alpha$ *are:*

$$A^+_\alpha = \begin{pmatrix} \frac{1}{2} & \frac{h_\alpha}{8} \\ -\frac{3}{2h_\alpha} & -\frac{1}{4} \end{pmatrix}\ , \quad A^-_\alpha = \begin{pmatrix} \frac{1}{2} & -\frac{h_\alpha}{8} \\ \frac{3}{2h_\alpha} & -\frac{1}{4} \end{pmatrix}\ . \tag{5.21}$$

Proof: According to an idea of A. Meyer we first find the basis transformation for the corresponding one-dimensional basis functions and then - using the Kronecker product - obtain the result of the theorem.

We define the two functions

$$p_0(x) \ = \ \frac{(x-h)^2(h+2x)}{h^3}\ , \tag{5.22}$$

$$p_1(x) = \frac{(x-h)^2 x}{h^2} \; . \tag{5.23}$$

Now we give the one-dimensional basis functions of the point 0 for the two elements $[0,h]$ and $[-h,0]$, see Fig. 3b:

$$g_0(x) = \begin{cases} p_0(x) & \text{if } x \in [0,h] \\ p_0(-x) & \text{if } x \in [-h,0) \; , \end{cases}$$

$$g_1(x) = \begin{cases} p_1(x) & \text{if } x \in [0,h] \\ -p_1(-x) & \text{if } x \in [-h,0) \; ; \end{cases}$$

we intoduce the basis $\Psi(x) = (g_0(x)|g_1(x))$. The corresponding basis functions of the next level in the elements $[-\frac{h}{2},0]$ and $[0,\frac{h}{2}]$, see also Fig. 3b, are

$$f_0(x) = \begin{cases} p_0(2x) & \text{if } x \in [0,\frac{h}{2}] \\ p_0(-2x) & \text{if } x \in [-\frac{h}{2},0) \; , \end{cases}$$

$$f_1(x) = \begin{cases} \frac{1}{2}p_1(2x) & \text{if } x \in [0,\frac{h}{2}] \\ -\frac{1}{2}p_1(-2x) & \text{if } x \in [-\frac{h}{2},0) \; ; \end{cases}$$

the basis is $F(x) = (f_0(x)|f_1(x))$. In the element $[0,\frac{h}{2}]$ the basis functions according to the point $\frac{h}{2}$ are

$$f_{r0}(x) = \begin{cases} f_0(\frac{h}{2}-x) & \text{if } x \in [0,\frac{h}{2}] \\ 0 & \text{if } x \in [-\frac{h}{2},0) \; , \end{cases}$$

$$f_{r1}(x) = \begin{cases} -f_1(\frac{h}{2}-x) & \text{if } x \in [0,\frac{h}{2}] \\ 0 & \text{if } x \in [-\frac{h}{2},0) \; ; \end{cases}$$

the basis is $R(x) = (f_{r0}(x)|f_{r1}(x))$. In the element $[-\frac{h}{2},0]$ the basis functions according to the point $-\frac{h}{2}$ are

$$f_{l0}(x) = \begin{cases} 0 & \text{if } x \in [0,\frac{h}{2}] \\ f_0(\frac{h}{2}+x) & \text{if } x \in [-\frac{h}{2},0) \; , \end{cases}$$

$$f_{l1}(x) = \begin{cases} 0 & \text{if } x \in [0,\frac{h}{2}] \\ f_1(\frac{h}{2}+x) & \text{if } x \in [-\frac{h}{2},0) \; ; \end{cases}$$

the basis is $L(x) = (f_{l0}(x)|f_{l1}(x))$. Computations show, that in the elements $[0,\frac{h}{2}]$ and $[-\frac{h}{2},0]$ the basis functions of Ψ can expressed in the following way by the basis functions of the next finer level:

$$\Psi(x) = (L(x)|F(x)|R(x)) \begin{pmatrix} A^- \\ I \\ A^+ \end{pmatrix} \; , \tag{5.24}$$

where the matrices $A^{\pm}$ are defined like in (5.21) with the mesh size h.

Using the Kronecker product, we can give the two-dimensional basis $\Phi_1(x,y)$ according to the point $P_1 = (0,0)$. In what follows, the upper index 1 (rsp. index 2) denotes functions and matrices according to the x-direction (rsp. y-direction). Therefore, the basis is:

$$\Phi_1(x,y) = \Psi^1(x) \otimes \Psi^2(y) = [g_0^1(x)g_0^2(y) \; g_0^1(x)g_1^2(y) g_1^1(x)g_0^2(y) \; g_1^1(x)g_1^2(y)] \; .$$

Note, that due to the definition of the Kronecker product the second and third basis functions are transposed in comparision to the usual BFS basis. We take this into consideration when we enumerate the functions. Since the relations (5.24) are guilty for $\Psi^1(x)$ and $\Psi^2(y)$ and using the properties of the Kronecker product

$$\begin{aligned} A\otimes(B+C) &= A\otimes B + A\otimes C\ , \\ (A\otimes B)(C\otimes D) &= AC\otimes BD \end{aligned}$$

we can express the basis functions of $\Phi_1(x,y)=\Psi^1(x)\otimes\Psi^2(y)$ from the two-dimensional basis functions of the next finer level:

$$\begin{aligned} \Phi_1 &= \Psi^1\otimes\Psi^2 \\ &= [L^1A_1^- + F^1I + R^1A_1^+]\otimes[L^2A_2^- + F^2I + R^2A_2^+] \\ &= \underbrace{(L^1\otimes L^2)}_{\phi_9}(A_1^-\otimes A_2^-) + \underbrace{(L^1\otimes F^2)}_{\phi_3}(A_1^-\otimes I) + \underbrace{(L^1\otimes R^2)}_{\phi_7}(A_1^-\otimes A_2^+) + \\ &\quad \underbrace{(F^1\otimes L^2)}_{\phi_5}(I\otimes A_2^-) + \underbrace{(F^1\otimes F^2)}_{\phi_1}(I\otimes I) + \underbrace{(F^1\otimes R^2)}_{\phi_4}(I\otimes A_2^+) + \\ &\quad \underbrace{(R^1\otimes L^2)}_{\phi_8}(A_1^+\otimes A_2^-) + \underbrace{(R^1\otimes F^2)}_{\phi_2}(A_1^+\otimes I) + \underbrace{(R^1\otimes R^2)}_{\phi_6}(A_1^+\otimes A_2^+) + \\ \Phi_1 &= [\phi_1\ \phi_2\ \phi_3\ \phi_4\ \phi_5\ \phi_6\ \phi_7\ \phi_8\ \phi_9\ ]\underbrace{\begin{pmatrix} I & \otimes & I \\ A_1^+ & \otimes & I \\ A_1^- & \otimes & I \\ I & \otimes & A_2^+ \\ I & \otimes & A_2^- \\ A_1^+ & \otimes & A_2^+ \\ A_1^- & \otimes & A_2^+ \\ A_2^+ & \otimes & A_2^- \\ A_2^- & \otimes & A_2^- \end{pmatrix}}_{B}\ , \end{aligned}$$

where ϕ_i are the two-dimensional bases of the next finer level for the point P_i (cf. Fig. 3a). ■

Remark. Note, that the matrices $A_\alpha^\pm$ depend on the real mesh sizes h_1 and h_2, what has to be taken into account during the hierarchical preconditioning.

5.2 Multilevel preconditioning for Adini elements

Since for the sequence of finite element spaces $\tilde{\mathbf{V}}_l = \mathrm{span}\tilde{\Phi}_l$ of the Adini elements the monotonicity condition (5.1) is violeted, we cannot use the theory of the previous subsection (the decomposition theorem in the proof of Lemma 5.1 requires the monotonicity condition).

On the other hand, in the previous subsection we have constructed an optimal preconditioner for the space of the BFS elements and in Subsection 4.3 we have defined mappings between the spaces of the BFS and Adini elements and proved some properties of these mappings. Now we will use the fictitious space lemma from Nepomnyaschikh (see [11]) to transfer the preconditioner from the space of the BFS elements into the space of the Adini elements. The idea was proposed in [13] and used in [10] for plate problems. We write the fictitious space lemma in the following form:

Lemma 5.2 *Let H and H_0 be Hilbert spaces with the scalar products $(\cdot,\cdot)$ and $[\cdot,\cdot]$, respectivly. Let $\mathcal{A} : H \to H$ and $\mathcal{A}_0 : H_0 \to H_0$ be linear, s.p.d., and continous operators in the spaces H and H_0. Further we assume, that there is a preconditioner $\mathcal{B}_0 : H_0 \to H_0$ for the operator $\mathcal{A}_0$ which satisfies*

$$\lambda_0[w,w]_{\mathcal{A}_0} \leq [\mathcal{B}_0^{-1}\mathcal{A}_0 w, w] \leq \lambda_1[w,w]_{\mathcal{A}_0}, \ \forall w \in H_0. \tag{5.25}$$

Suppose that $\mathcal{R}$ is a linear operator such that $\mathcal{R} : H_0 \to H$ and

$$\|\mathcal{R}w\|_{\mathcal{A}}^2 \leq c_R \|w\|_{\mathcal{A}_0}^2 \tag{5.26}$$

is fulfilled for all $w \in H_0$. Moreover, there exists a linear operator $\mathcal{Q}$ such that $\mathcal{Q} : H \to H_0$ for which the conditions $\mathcal{R}\mathcal{Q}v = v$ and

$$\|\mathcal{Q}v\|_{\mathcal{A}_0}^2 \leq c_Q^{-1}\|v\|_{\mathcal{A}}^2 \tag{5.27}$$

are valid for all $v \in H$. Then for the preconditioner $\mathcal{B} : H \to H$ defined by $\mathcal{B}^{-1} = \mathcal{R}\mathcal{B}_0^{-1}\mathcal{R}^*$ the following estimations of the eigenvalues hold:

$$\begin{aligned} \lambda_{\max}(\mathcal{B}^{-1}\mathcal{A}) &\leq c_R\lambda_1 \ , \\ \lambda_{\min}(\mathcal{B}^{-1}\mathcal{A}) &\geq c_Q\lambda_0 \ . \end{aligned} \tag{5.28}$$

The operator $\mathcal{R}^*$ is adjoint to $\mathcal{R}$ w.r.t. the scalar products $(\cdot,\cdot)$ and $[\cdot,\cdot]$, i.e., we have $\mathcal{R}^* : H \to H_0$ and $[\mathcal{R}^*v, w] = (v, \mathcal{R}w), \ \forall v \in H, \ \forall w \in H_0$.

Using this lemma we can prove the following theorem.

Theorem 5.3 *Let $\mathbf{F}_h$ be the matrix representation of the mapping $\boldsymbol{\mathcal{F}}_h : \mathbf{V}_h \to \tilde{\mathbf{V}}_h$ defined in (4.27). Then for*

$$\tilde{\mathbf{C}}_h^{-1} = \mathbf{F}_h \mathbf{C}_h^{-1} \mathbf{F}_h^T \tag{5.29}$$

with $\mathbf{C}_h$ from (5.18) there exist mesh size independent constants $\underline{c}$ and $\overline{c}$ which satisfy the following spectral inequality:

$$\underline{c}\tilde{\mathbf{C}}_h^{-1} \leq \tilde{\mathbf{K}}_h \leq \overline{c}\tilde{\mathbf{C}}_h^{-1},$$

where $\tilde{\mathbf{K}}_h$ is the stiffness matrix according to the variational problem (4.7).

Proof: The proof is similar to this in [10]. We use the fictitious space lemma with the mappings $\boldsymbol{\mathcal{F}}_h$ and $\boldsymbol{\mathcal{E}}_h$ between the spaces of the BFS and Adini elements. Together with Lemma 4.6, Theorem 4.2, and Lemma 4.8 we obtain:

$$\begin{aligned} \tilde{a}(\boldsymbol{\mathcal{F}}_h\mathbf{v}_h, \boldsymbol{\mathcal{F}}_h\mathbf{v}_h) &\leq \tilde{\mu}_C|\boldsymbol{\mathcal{F}}_h\mathbf{v}_h|_{\mathcal{T}_h}^2 \\ &\leq \tilde{\mu}_C\beta_{\boldsymbol{\mathcal{F}}}|\mathbf{v}_h|_{\mathbf{V}}^2 \\ &\leq \tilde{\mu}_C\beta_{\boldsymbol{\mathcal{F}}}\mu_E a(\mathbf{v}_h,\mathbf{v}_h), \ \forall \mathbf{v}_h \in \mathbf{V}_h, \\ a(\boldsymbol{\mathcal{E}}_h\tilde{\mathbf{v}}_h, \boldsymbol{\mathcal{E}}_h\tilde{\mathbf{v}}_h) &\leq \mu_C\beta_{\boldsymbol{\mathcal{E}}}\tilde{\mu}_E\tilde{a}(\tilde{\mathbf{v}}_h,\tilde{\mathbf{v}}_h), \ \forall \tilde{\mathbf{v}}_h \in \tilde{\mathbf{V}}_h. \end{aligned}$$

Using (5.28) we get

$$\begin{aligned} \underline{c} &= \mu_C\beta_{\boldsymbol{\mathcal{E}}}\tilde{\mu}_E\lambda_{\min}(\mathbf{C}_h^{-1}\mathbf{K}_h), \\ \overline{c} &= \tilde{\mu}_C\beta_{\boldsymbol{\mathcal{F}}}\mu_E\lambda_{\max}(\mathbf{C}_h^{-1}\mathbf{K}_h), \end{aligned}$$

where $\lambda_{\min}(\mathbf{C}_h^{-1}\mathbf{K}_h)$ and $\lambda_{\max}(\mathbf{C}_h^{-1}\mathbf{K}_h)$ are mesh size independent due to Theorem 5.1. ■

6 Parallelization and implementation

6.1 The parallelization of the data structure

The parallelization concept of the program is based on the non-overlapping domain decomposition data structure (cf. [7]) such that our implementations are well-suited for MIMD parallel machines under message-passing. We use two types of vectors which we call additive (type I) and overlapping (type II):

- type I : the solution vector $\underline{\mathbf{u}}$ is represented locally on each processor P_i by the vector $\underline{\mathbf{u}}_i = A_i \underline{\mathbf{u}}$,
- type II : the load vector $\underline{\mathbf{f}}$ is represented locally on each processor P_i by $\underline{\mathbf{f}}_i$ with $\underline{\mathbf{f}} = \sum\limits_{i=1}^{p} A_i^T \underline{\mathbf{f}}_i$, where p is the number of processors.

A_i is the super element connectivity matrix of the subdomain $\overline{\Omega}_i$ (located on the processor P_i) with the dimension $N_i \times N$ (N: number of unknowns of the global problem, N_i: number of unknowns on the subdomain $\overline{\Omega}_i$) which maps a global vector $\underline{\mathbf{g}} \in \mathbf{R}^N$ on a local vector $\underline{\mathbf{g}}_i \in \mathbf{R}^{N_i}$. Using the Boolean matrix A_i, we can write the stiffness matrix $\mathbf{K}$ in the form

$$\mathbf{K} = \sum_{i=1}^{p} A_i^T \mathbf{K_i} A_i \ , \tag{6.1}$$

where $\mathbf{K_i}$ is the super element stiffness matrix belonging to $\overline{\Omega}_i$.

6.2 Assembly of the stiffness matrix and the load vector

Since we use the data structure of the previos subsection, the assembly of the super element stiffness matrix $\mathbf{K_i}$ and the super element load vector $\mathbf{f_i}$ can be carried out on every processor without communication. This is an important advantage, because due to the complicate bilinear form (3.6) with (3.3), (3.4) the assembly of the stiffness matrix is very expansive.

For the numerical integration we use the Gaussian quadrature formula. With the help of a subroutine the mapping function $\boldsymbol{\phi}$ and its partial derivatives up to the second order are included into the program. Then in every integration point the local basis vectors (2.1), (2.2), and (2.3), the components (2.4), (2.6) of the first fundamental form, (2.9), (2.10) of the second fundamental form, (2.11) of the third fundamental form, (2.12) of the Christoffel symbol and (3.5) of the elasticity coefficients are calculated. The partial derivatives of the mixed components of the second fundamental form (2.10) which are used in the bilinear form (cf. (3.4)) are computed via numerical differentiation. By using all these values and putting the BFS ansatz functions (16 per element) into the bilinear form (3.6) and carrying out the weighted summation over all intergration points, the element integration subroutine calculatates six element stiffness submatrices E_{xx}, E_{xy}, E_{xz}, E_{yy}, E_{yz}, E_{zz} of dimension 16×16. Finally, the assembly subroutine assembles these submatrices for all elements in such a way, that every node contains $3 \times 4 = 12$ nodal values for the 3 directions (x,y,z) and 4 derivatives (value, first partial derivatives, mixed derivative) (see Section 4). For the Adini elements the assembly is realized in a similar way.

6.3 The parallel preconditioned CG-method

The PPCG methods which are based on the data structure of Subsection 6.1 require communication between the processors for computing of scalar products, within the preconditioners, and for solving the systems of equations on the coarse mesh. A detailed description of the used parallelization of the PCG method is given in [7].

For the BFS elements we now consider the realization of the preconditioning step $\underline{\mathbf{w}}_h = \mathbf{C}_h^{-1}\underline{\mathbf{d}}_h$ with the matrix $\mathbf{C}_h^{-1}$ from (5.18). The residuum vector $\underline{\mathbf{d}}_h$ is of the additive type. The transformation action $\underline{\mathbf{v}} = \mathbf{S}^T\underline{\mathbf{d}}_h$ does not require any communication. Since the resulting vector $\underline{\mathbf{v}}$ is of the additive type, it now has to be transformed into the overlapping type. This requires a communication over the *coupling nodes* of all subdomains for all levels of the generating system.

Now we solve the coarse-grid system using a direct solver on the basis of the Cholesky factorization of the matrix $\mathbf{K}_{j_0}$. The matrix is stored in the processor P_0, and before starting the PPCG algorithm step, all processors send their part of right-hand side vector to the processor P_0. After the forward and backward substitution steps for solving the coarse grid system on processor P_0, this processor sends the solution to all processors.

The scaling and multiplication of the resulting vector with $\mathbf{S}$ does not require any communication again. Details of the parallel implementation of the classic BPX and MDS-BPX preconditioners can be find in [1].

In the case of Adini elements due to (5.29) we first have to transform the residuum vector $\tilde{\underline{\mathbf{d}}}_h$ of the Adini elements space into the residuum vector $\underline{\mathbf{d}}_h = \mathbf{F}_h^T\tilde{\underline{\mathbf{d}}}_h$ of the BFS element space. This simple operation does not require any communication. Then we carry out the preconditioning step $\underline{\mathbf{w}}_h = \mathbf{C}_h^{-1}\underline{\mathbf{d}}_h$ as described above. At the end, we transform the correction $\underline{\mathbf{w}}_h$ back into the Adini elements space by calculating $\tilde{\underline{\mathbf{w}}}_h = \mathbf{F}_h\underline{\mathbf{w}}_h$ what also does not need any communication.

7 Numerical results

In this section we consider three kinds of shells: plates, cylinders, and hyperboloids. For every shell we choose a parametrization and state the most important tensor components. For all numerical examples we compare the iteration numbers for different numbers of processors. We use only the MDS-BPX preconditioner, because computations with the classic BPX preconditioner which are not documented here lead to much higher iteration numbers. All algorithms, described in the Section 6, are implemented in the program SPC-EL 2,5D.

In correspondence to Section 6, for parallelization we decompose the reference domain Ω in p subdomains where p is the number of processors. Therefore, in this section every subdomain corresponds to one processor. The computations are carried out on a Parsytec GC/PP-128 machine. This machine is provided with 128 processors of the type PowerPC-601 installed at 64 nodes in a 2D-grid topology. Each processor has a memory of 16 MByte.

7.1 The MDS-BPX preconditioner for BFS elements

7.1.1 The plate

We use the parametrization

$$\phi^1(x^1,x^2) = x^1\ ,\ \phi^2(x^1,x^2) = x^2\ ,\ \phi^3(x^1,x^2) = 0\ ,$$

which leads to the following local bases:

$$\mathbf{a}_1 = \mathbf{a}^1 = \mathbf{e}_1 \ ; \mathbf{a}_2 = \mathbf{a}^2 = \mathbf{e}_2 \ ; \mathbf{a}_3 = \mathbf{a}^3 = \mathbf{e}_3$$

and the tensor components

$$\begin{cases} a_{11} = a^{11} = a_{22} = a^{22} = a = 1 \ ; \ a_{12} = a^{12} = 0 \ ; \\ b_{\alpha\beta} = b_\alpha^\beta = b^{\alpha\beta} = 0 \ ; \\ c_{\alpha\beta} = 0 \ ; \\ \Gamma^\varrho_{\alpha\beta} = 0 \ . \end{cases}$$

In order to compare our results with these of the MDS-BPX preconditioner in [14] we consider three kinds of domains Ω: the unit square (Fig. 4a, see also Fig. 5), an L–shaped domain (Fig. 4b) and a slit domain (Fig. 4c). We use the Poisson coefficient $\nu = 0.3$ (steel). The numbers of unknowns and the corresponding iteration numbers of the MDS

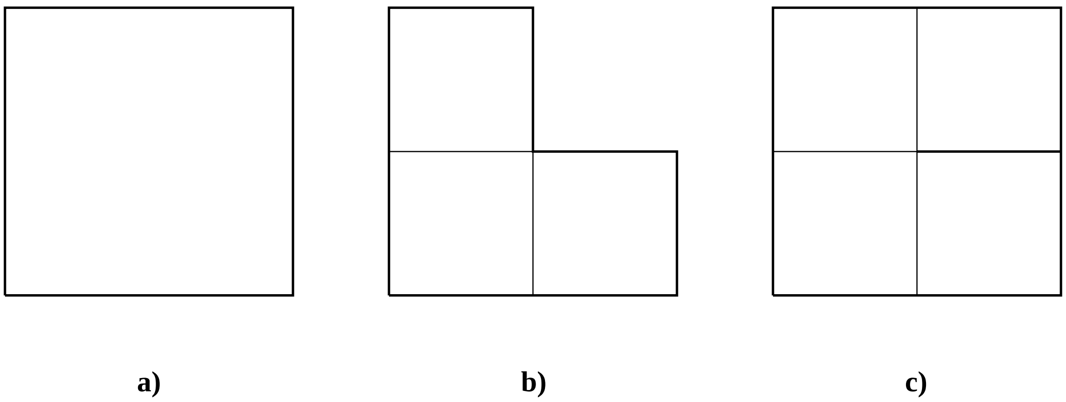


Figure 4: Domains Ω and initial partitions $\mathcal{T}_0$.

preconditioned CG method for all three plates are shown in the Table 1. The calculations were carried out on one processor. As the stopping criteria of the iteration, we use the relative accuracy 10^{-6} in the the near-energy norm $\mathbf{K}_h\mathbf{C}^{-1}\mathbf{K}_h$. For the unit square the

J	Unit square		L–shaped domain		Slit domain	
2	36	7	132	15	180	15
3	196	13	644	19	868	19
4	900	15	2820	21	3780	21
5	3844	17	11780	–	15740	–

Table 1: Plates, multilevel preconditioning: 1 subdomain.

iteration numbers of the MDS-BPX preconditioner are nearly the same as in [14] while the iteration numbers of the L–shaped domain and the slit domain are notably lower than these in [14].

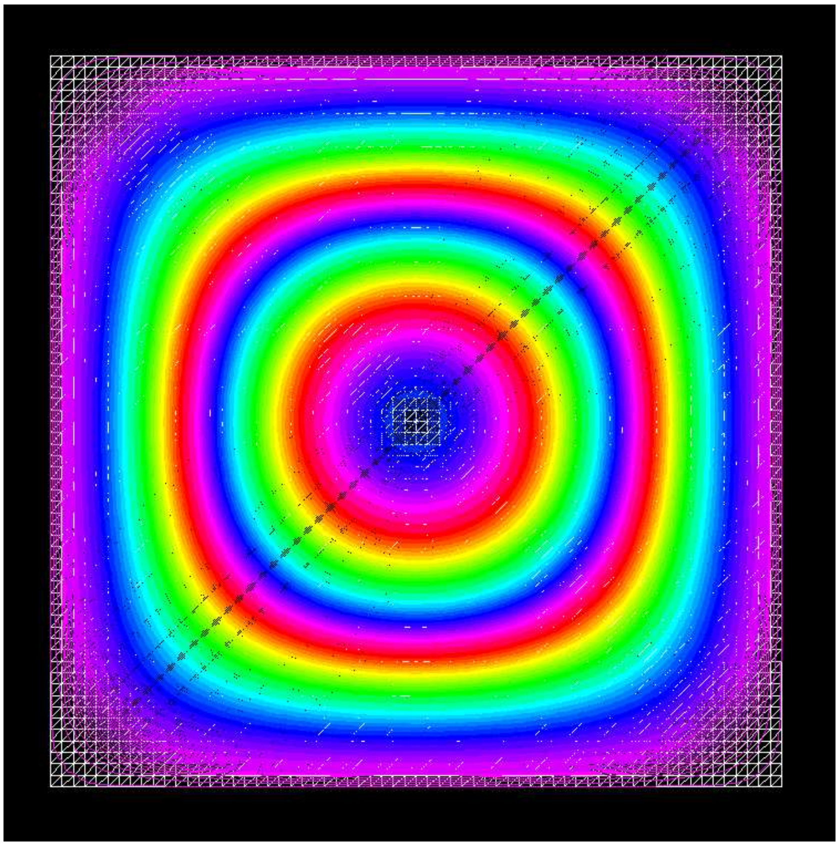


Figure 5: Plate under constant vertical load.

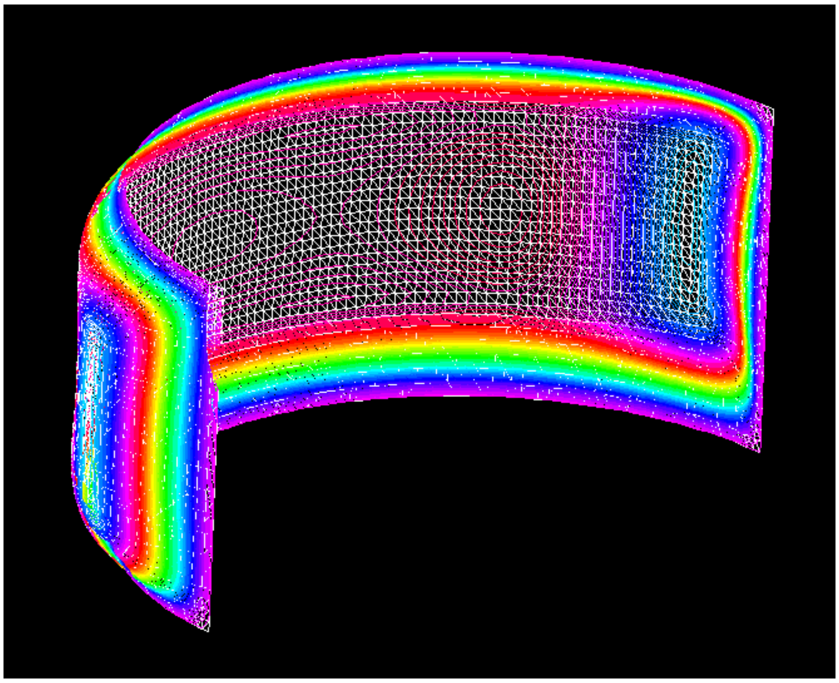


Figure 6: Arch under constant radial load.

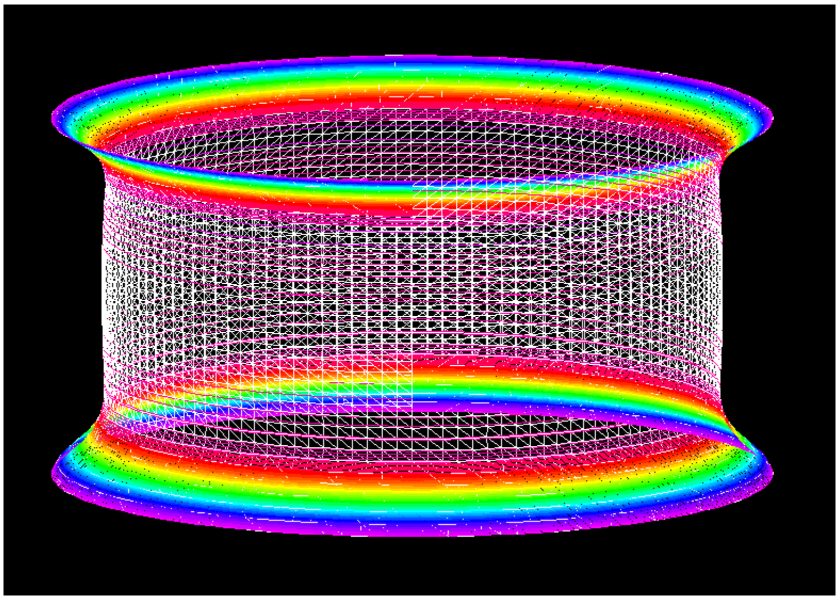


Figure 7: Cylinder under constant radial load.

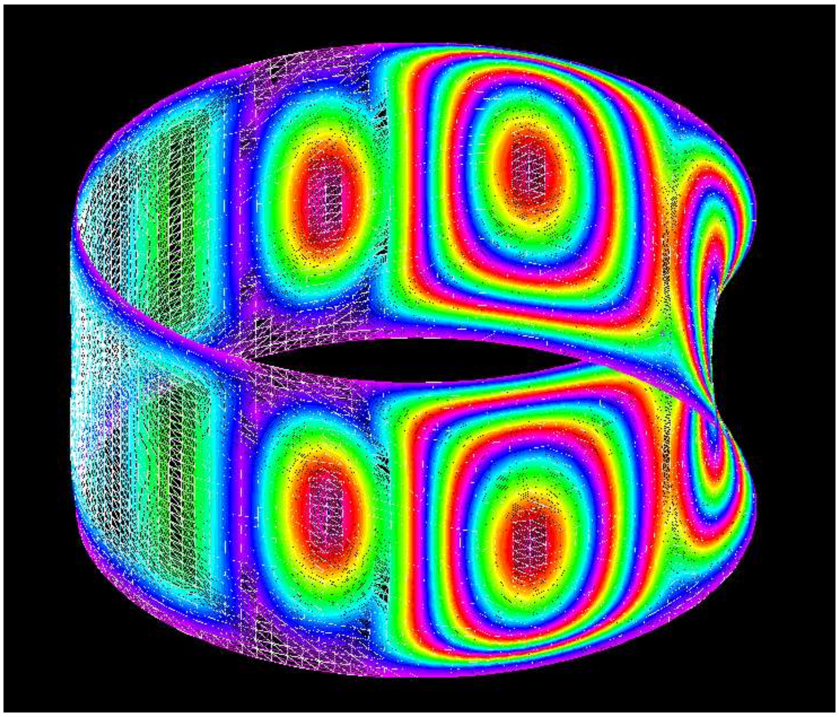


Figure 8: Cylinder under wind load.

7.1.2 Arch and full cylinder

We use the parametrization

$$\phi^1(x^1,x^2) = R\cos\frac{x^1}{R}\ ,\ \phi^2(x^1,x^2) = R\sin\frac{x^1}{R}\ ,\ \phi^3(x^1,x^2) = x^2\ ,$$

which enables identical orthonormal covariant and contravariant local bases to be obtained:

$$\mathbf{a}_1 = \mathbf{a}^1 = -\sin\frac{x^1}{R}\mathbf{e}_1 + \cos\frac{x^1}{R}\mathbf{e}_2\ ;\ \mathbf{a}_2 = \mathbf{a}^2 = \mathbf{e}_3\ ;\ \mathbf{a}_3 = \mathbf{a}^3 = \cos\frac{x^1}{R}\mathbf{e}_1 + \sin\frac{x^1}{R}\mathbf{e}_2\ .$$

The tensor components are

$$\begin{cases} a_{11} = a^{11} = a_{22} = a^{22} = a = 1\ ;\ a_{12} = a^{12} = 0\ ; \\ b_{11} = b_1^1 = b^{11} = -\frac{1}{R}\ ;\ b_{\alpha 2} = b_\alpha^2 = b^{\alpha 2} = 0\ ; \\ c_{11} = \frac{1}{R^2}\ ;\ c_{12} = c_{22} = 0\ ; \\ \Gamma_{\alpha\beta}^{\varrho} = 0\ . \end{cases}$$

We consider two kinds of these shells: an arch and a full cylinder.

The radius of the arch (cf. Fig. 6) is $R = 1$ and its lenght is $L = 1$; therefore we have the following reference domain:

$$\Omega = \{(x^1,x^2)|0 \le x^1 \le \pi\ , 0 \le x^2 \le 1\}\ .$$

We further assume, that the arch is clamped on its complete lateral boundary, i.e.

$$\Gamma_0 = \partial\Omega\ .$$

The thickness of the shell is $\varepsilon = 0.01$. We use the Poisson coefficient $\nu = 0.3$. Finally, we assume, that the arch is loaded by constant radial forces; therefore we use $p^1 = p^2 = 0$, $p^3 = const. > 0$. The iteration numbers and times (and pure arithmetic times) in seconds of MDS preconditioned CG methods including coarse grid solution on the levels j_0 (cf. 5.17) for the arch are given in the Tables 2 and 3. As the stopping criteria of the iteration, we use the relative accuracy 10^{-5} in the near energy norm $\mathbf{K}_h\mathbf{C}^{-1}\mathbf{K}_h$. As we

J	N	$j_0 = 0$ (4×1)		$j_0 = 1$ (8×2)		$j_0 = 2$ (16×4)	
2	540	32	1.22 (0.60)		–		–
3	2604	32	2.05 (1.19)	21	3.48 (1.76)		–
4	11340	33	4.50 (3.36)	22	4.72 (2.28)	19	10.94 (3.73)
5	47244	33	13.42 (11.88)	22	6.93 (3.87)	20	14.37 (4.29)
6	192780		–	22	13.69 (9.79)	21	19.35 (6.08)
7	778764		–		–	21	27.98 (11.88)

Table 2: Arch, multilevel preconditioning: 4, 16, 64 subdomains.

can see, the iteration numbers became nearly constant for increasing level number J. We also obtain that in the case of a large number of subdomains, the iteration numbers became low, because the geometry of the shell is well described by the coarse grid system (which is solved directly). On the other hand the enlargement of the number of subdomains leads

J	N	$j_0 = 0$ (4×2)		$j_0 = 1$ (8×4)		$j_0 = 2$ (16×8)	
2	1260	34	1.82 (0.79)		–		–
3	5580	34	2.67 (1.34)	21	3.89 (1.23)		–
4	23436	35	5.59 (3.82)	22	5.44 (1.73)	20	18.31 (7.42)
5	96012	35	15.51 (13.34)	23	8.32 (3.46)	21	23.49 (8.27)
6	388620		–	24	16.36 (10.19)	22	30.23 (10.26)
7	1563660		–		–	23	42.47 (17.11)

Table 3: Arch, multilevel preconditioning: 8, 32, 128 subdomains.

to a decreasing efficency of the paralleliztion. The iteration numbers are approximately like that for the same problem in [10] despite the fact that in [10] the shell model of Novozhilov [12] is used. However, for cylindrical shells both models only differ in one component of the middle surface change of curvature tensor $\bar{\rho}_{\alpha\beta}$ (see (3.4)).

We now want to investigate the influence of the geometry parameters R and ε on the iteration numbers of the preconditioner. In order to do this, we use the arch with the domain that is splitted into 4 subdomains (see Table 2, $j_0 = 0$). First we vary the radius R. The results are shown in the Table 4. Then we vary the thickness ε. The results are given in Table 5. The tables show that the iteration numbers of the MDS-BPX

J	N	R				
		0.5	1.0	2.0	4.0	8.0
2	540	41	32	25	22	22
3	2604	42	32	26	23	22
4	11340	42	33	26	23	22
5	47244	42	33	27	24	23

Table 4: Arch, multilevel preconditioning for different radii: 4 subdomains.

J	N	ε				
		0.0001	0.001	0.01	0.1	1.0
2	540	89	72	32	22	23
3	2604	191	78	32	22	24
4	11340	180	76	33	23	25
5	47244	176	76	33	23	26

Table 5: Arch, multilevel preconditioning for different thicknesses: 4 subdomains.

preconditioner strongly depend on both geometry parameters R and ε. On the other hand, for every value of both geometry parameters the iteration numbers became quickly (nearly) constant for an increasing level number J.

Now we consider a full cylinder (cf. Fig. 7). We use the same parameters like in the case of the arch with two exceptions. Since the shell is a cylinder, we naturally must use the following reference domain:

$$\Omega = \{(x^1, x^2) | 0 \le x^1 \le 2\pi \ , 0 \le x^2 \le 1\} \ .$$

The second difference is the change of the boundary conditions: the plate is clamped at the boundary at $x^2 = 0$, single supported at the boundary at $x^2 = 1$ ($\mathbf{u}|_{\Gamma_1} = 0$):

$$\begin{aligned} \Gamma_0 &= \{(x^1, x^2) \in \partial\Omega \ : \ x^2 = 0\} \ , \\ \Gamma_1 &= \{(x^1, x^2) \in \partial\Omega \ : \ x^2 = 1\} \ , \end{aligned}$$

and there are periodic boundary conditions defined by

$$\mathbf{u}(0, x^2) = \mathbf{u}(2\pi, x^2) \ , \ \forall x^2 \in [0, 1] \ .$$

The iteration numbers and times of the calculation for this cylinder problem are shown in Tables 6, 7. As we can see, the iteration numbers are very small in contradiction

J	N	$j_0 = 0$ (4×1)	$j_0 = 1$ (8×2)	$j_0 = 2$ (16×4)
2	704	13 0.98 (0.83)	–	–
3	2944	12 1.22 (1.03)	11 0.91 (0.43)	–
4	12032	12 2.00 (1.75)	11 1.43 (0.64)	10 4.14 (2.68)
5	48640	13 5.64 (5.29)	11 2.98 (1.92)	11 5.11 (2.98)
6	195584	–	12 9.31 (7.62)	11 6.49 (3.78)
7	784384	–	–	12 10.87 (7.27)

Table 6: Cylinder, multilevel preconditioning: 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)	$j_0 = 1$ (8×4)	$j_0 = 2$ (16×8)
2	1472	11 0.85 (0.73)	–	–
3	6016	11 1.15 (0.91)	10 1.21 (0.66)	–
4	24320	11 2.05 (1.73)	10 1.58 (0.85)	9 7.09 (5.17)
5	97792	11 5.20 (4.76)	11 2.69 (1.69)	11 8.76 (6.66)
6	392192	–	12 6.47 (5.11)	11 10.37 (6.49)
7	1570816	–	–	12 15.18 (10.05)

Table 7: Cylinder, multilevel preconditioning: 8, 32, 128 subdomains.

to the results of [10], where much higher iteration numbers were obtained for nearly the same problem (other single supported boundary conditions at $x^2 = 1$ were used). First investigations expect to find the reason in different communication techniques.

Since in the last problem constant radial forces lead to a radial symmetric solution, we now consider the same problem for the cylindric shell, but with unsymmetric load (cf. Fig. 8). For this we use the wind load from [2], which is shown in Fig. 9.

The distribution function c is analytically given as

$$c(\vartheta) \quad = \quad -0.2273 + 0.3762\cos(\vartheta) + 0.5148\cos(2\vartheta) +$$

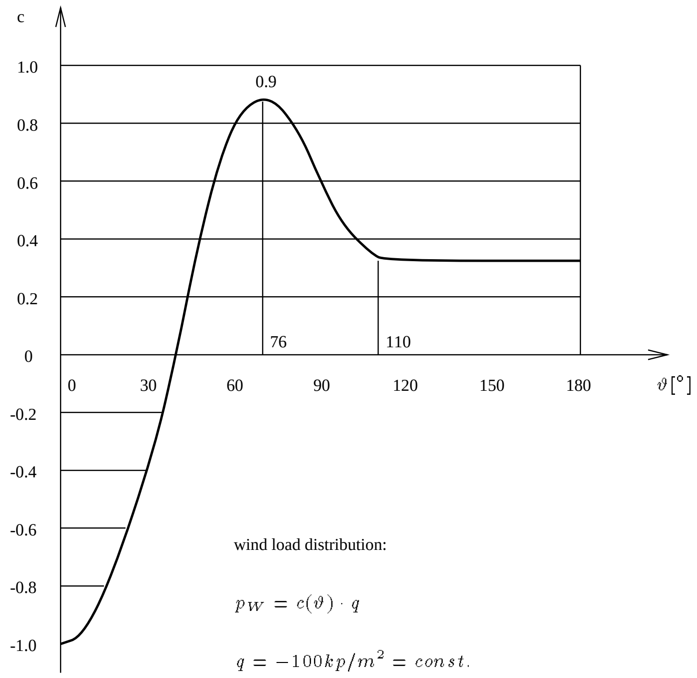


Figure 9: Distribution of the wind load.

$$
\begin{aligned}
&+0.3509\cos(3\vartheta) + 0.0452\cos(4\vartheta) - \\
&-0.0719\cos(5\vartheta) - 0.0077\cos(6\vartheta) + \\
&+0.0287\cos(7\vartheta) - 0.0024\cos(8\vartheta) + \\
&-0.0129\cos(9\vartheta) + 0.0044\cos(10\vartheta)\ .
\end{aligned}
$$

So we have the following components of the vector field $p6i\mathbf{a}_i$: $p^1 = p^2 = 0,\ p^3 = p_W(x^1)$. The iteration numbers and times of the calculation for this cylinder problem are shown in Tables 8, 9. As we can see, the iteration numbers are much higher than that for

J	N	$j_0 = 0\ (4 \times 1)$	$j_0 = 1\ (8 \times 2)$	$j_0 = 2\ (16 \times 4)$
2	704	50 3.94 (3.39)	–	–
3	2944	46 4.90 (4.27)	39 4.37 (3.01)	–
4	12032	45 7.21 (6.37)	40 5.51 (3.62)	19 5.97 (3.29)
5	48640	46 19.50 (18.25)	40 8.98 (6.53)	20 7.47 (3.75)
6	195584	–	41 20.85 (17.52)	20 9.91 (5.19)
7	784384	–	–	20 16.83 (10.65)

Table 8: Cylinder under wind load, multilevel preconditioning: 4, 16, 64 subdomains.

the cylinder under constant load. These iteration numbers better correspond to the eigenvalues of the MDS preconditioner, see Subsection A.1.

J	N	$j_0 = 0$ (4×2)		$j_0 = 1$ (8×4)		$j_0 = 2$ (16×8)	
2	1472	59	4.68 (3.83)		–		–
3	6016	60	5.95 (4.75)	53	4.73 (1.99)		–
4	24320	58	10.45 (8.94)	55	6.81 (3.05)	41	17.10 (8.89)
5	97792	59	26.86 (24.70)	58	12.16 (7.23)	45	22.36 (10.36)
6	392192		–	60	29.88 (23.45)	46	29.35 (13.90)
7	1570816		–		–	51	49.88 (28.71)

Table 9: Cylinder under wind load, multilevel preconditioning: 8, 32, 128 subdomains.

7.1.3 Cooling tower

Most of these towers have a middle surface of the *hyperbolic* type. One example of such a tower is given in Fig. 10 together with a parametrization ($\mathbf{\Phi}$-1) taken from [2]. (In [2] the parametrization is only used to describe the tower; in the calculations the geometry of the tower is approximated by flat facets.) Now we adapt the parametrization of the

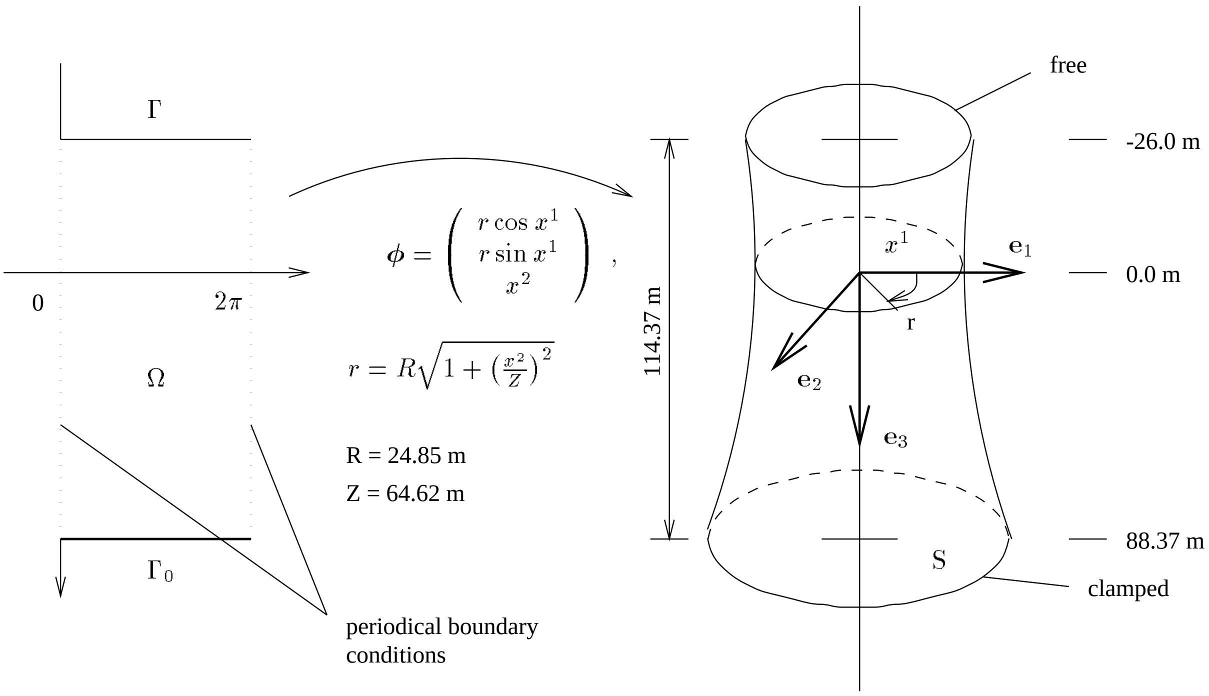


Figure 10: Cooling tower with parametrization ($\mathbf{\Phi}$-1) according to Argyris.

cooling tower used in [3] to our cooling tower (the advantage of this parametrization will be seen later). Therefore, the surface of the tower is the image of the rectangle Ω of the plane

$$\Omega = \{(x^1, x^2) | 0 \leq x^1 \leq 2\pi \ , \ \underline{x}^2 \leq x^2 \leq \overline{x}^2\} \ , \ \underline{x}^2 = -0.93941, \ \overline{x}^2 = 0.3829$$

via the mapping

$$\phi^1(x^1, x^2) = R\frac{\cos x^1}{\cos x^2} \ , \ \phi^2(x^1, x^2) = R\frac{\sin x^1}{\cos x^2} \ , \ \phi^3(x^1, x^2) = Z\tan x^2 + Z_0 \ ,$$

where $R = 24.85m$, $Z = 64.60m$, $Z_0 = 88.35m$. We denote this parametrization by (Φ-2). The covariant local basis is given by the tangential basis vectors

$$\mathbf{a}_1 = \frac{R}{\cos x^2} \begin{pmatrix} -\sin x^1 \\ \cos x^1 \\ 0 \end{pmatrix} ; \ \mathbf{a}_2 = \frac{1}{(\cos x^2)^2} \begin{pmatrix} R \cos x^1 \sin x^2 \\ R \sin x^1 \sin x^2 \\ Z \end{pmatrix} ;$$

and the normal basis vector

$$\mathbf{a}_3 = \frac{1}{\sqrt{R^2(\sin x^2)^2 + Z^2}} \begin{pmatrix} Z \cos x^1 \\ Z \sin x^1 \\ -R \sin x^2 \end{pmatrix} .$$

Now we give the most important tensor components.
First fundamental form:

$$a_{11} = \frac{R^2}{(\cos x^2)^2} \ , \ a_{12} = 0 \ , \ a_{22} = \frac{R^2(\sin x^2)^2 + Z^2}{(\cos x^2)^4} \ , \ a = \frac{R^2[R^2(\sin x^2)^2 + Z^2]}{(\cos x^2)^6} \ .$$

Second fundamental form:

$$b_{11} = -b_{22} = -\frac{RZ}{\cos x^2 \sqrt{R^2(\sin x^2)^2 + Z^2}} \ , \ b_{12} = b_{21} = 0 \ .$$

Christoffel symbols:

$$\left\{ \begin{array}{l} \Gamma^1_{11} = \Gamma^1_{12} = \Gamma^1_{11} = 0 \ ; \ \Gamma^2_{12} = \tan x^2 \ ; \\ \Gamma^2_{11} = -\frac{R^2 \sin x^2 \cos x^2}{R^2(\sin x^2)^2 + Z^2} ; \ \Gamma^2_{22} = \frac{R^2[1+(\sin x^2)^2]+2Z^2}{R^2(\sin x^2)^2+Z^2} \tan x^2 . \end{array} \right.$$

With this parametrization for a cooling tower with the thickness $\varepsilon = 0.14m$, Young's modulus $E = 3 \cdot 10^9 kp/m^2$, and the Poisson coefficient $\nu = 0.2$ under a wind pressure (as described in the previous subsection) with the boundary that is clamped at $\underline{x}^2 = -0.93941$ and free at $\overline{x}^2 = 0.3829$, our program finds nearly the same solution of the displacement field $\mathbf{u}$ as the best of the methods (SHEBA) given in [2] (see Fig. 11, see also Fig. 12 for the complete result of the deformation calculated by the use of the parametrization ($\boldsymbol{\Phi}$-2)). But using the parametrization ($\boldsymbol{\Phi}$-1) for the geometric description of the cooling tower, which is more complicated than the parametrization ($\boldsymbol{\Phi}$-2), our program finds a much worse solution (Fig. 11)

The iteration numbers and times of the calculation for the cooling tower problem using parametrization ($\boldsymbol{\Phi}$-2) are shown in the Tables 10, 11. The stopping criterion of the iteration is the same as in the cylinder case. The tables show high, but nearly constant iteration numbers. We also see that the enlargement of the number of subdomains reduces the iteration number substancially.

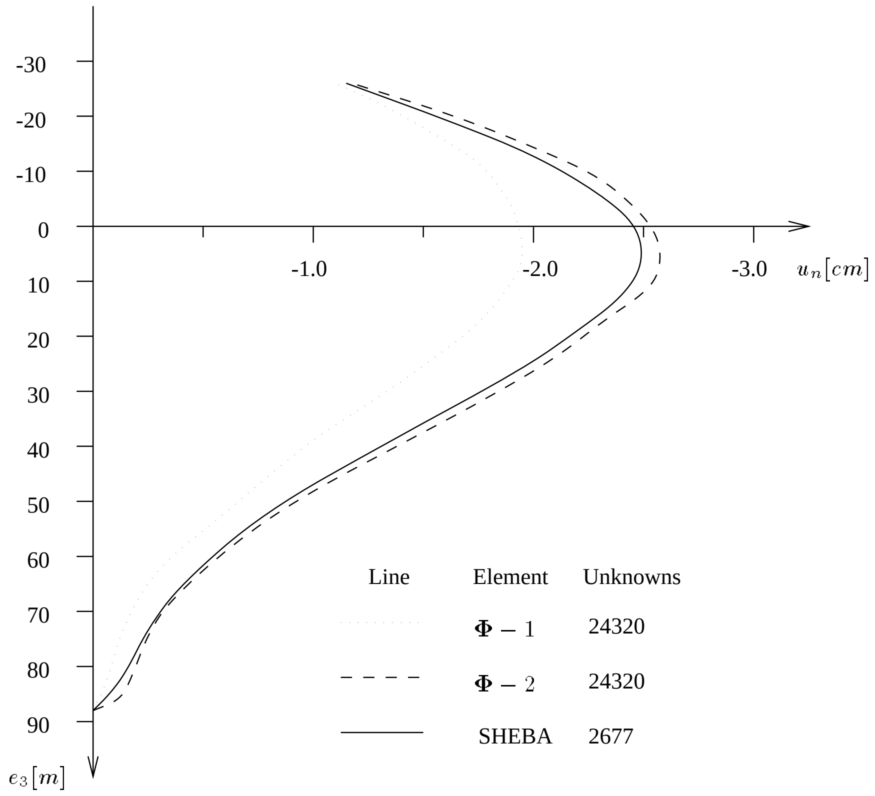


Figure 11: Normal displacements u_n for the BFS elements with different parametrizations and for the SHEBA element.

J	N	$j_0 = 0$ (4×1)		$j_0 = 1$ (8×2)		$j_0 = 2$ (16×4)	
2	704	208	5.75 (3.52)		–		–
3	2944	206	9.92 (6.90)	120	7.29 (3.02)		–
4	12032	200	23.64 (19.66)	118	10.75 (5.11)	63	14.73 (6.14)
5	48640	203	77.83 (71.97)	117	20.45 (13.31)	66	19.76 (7.72)
6	195584		–	123	56.73 (46.82)	89	36.26 (15.99)
7	784384		–		–	113	82.98 (50.08)

Table 10: Cooling tower under wind load, multilevel prec.: 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)		$j_0 = 1$ (8×4)		$j_0 = 2$ (16×8)	
2	1472	200	7.64 (4.61)		–		–
3	6016	191	19.22 (15.31)	102	8.75 (3.52)		–
4	24320	192	32.79 (27.65)	99	12.08 (5.35)	34	15.07 (8.05)
5	97792	194	87.84 (80.30)	100	20.68 (12.15)	34	18.62 (8.82)
6	392192		–	100	49.75 (38.86)	36	26.59 (11.76)
7	1570816		–		–	39	50.09 (25.67)

Table 11: Cooling tower under wind load, multilevel prec.: 8, 32, 128 subdomains.

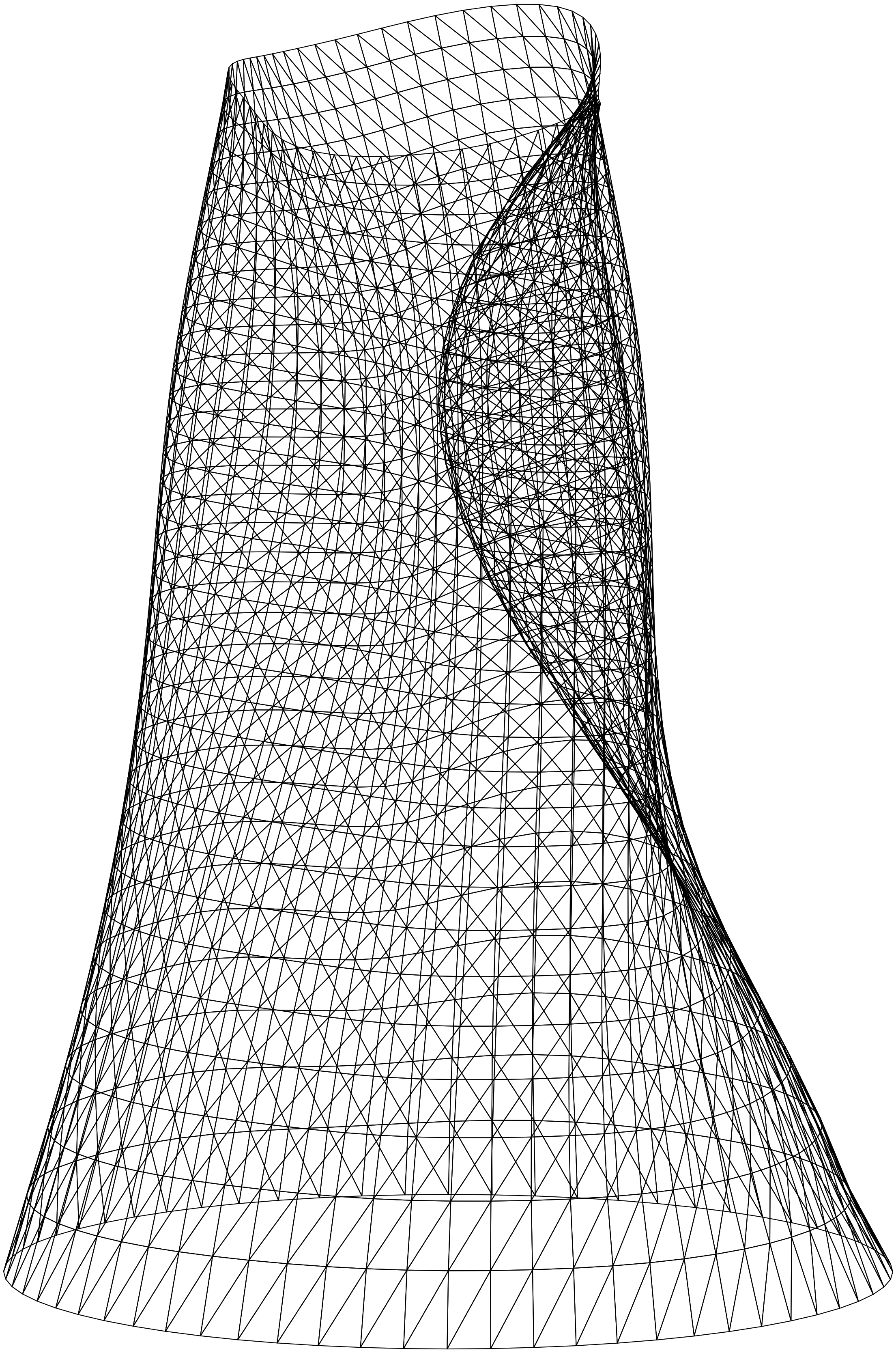


Figure 12: Cooling tower under wind load (deformations are drawn strongly amplified).

7.2 The MDS-BPX preconditioner for Adini elements

In this subsection we present the results of the MDS-BPX preconditioner for the same examples (plate, cylinder, hyperboloid) like in Subsection 7.1. All assumptions and conditions of the calculations are also the same. We start with the results of the deformation of the plate (see Subsection 7.1.1) which are shown in Table 12. As we can see, the iteration

J	Unit square		L–shaped domain		Slit domain	
2	27	6	99	14	135	15
3	147	12	483	18	651	18
4	675	14	2119	20	2835	21
5	2883	17	8835	–	11805	–

Table 12: Plates, multilevel preconditioning: 1 subdomain.

numbers are nearly the same like in the case of BFS elements. For low level numbers they are somewhat lower while for increasing levels they became as high as that of the BFS elements.

Now we show the iteration numbers for the arch (Table 13) and for the cylinder under wind load (Table 14).

J	N	$j_0 = 0$	$j_0 = 1$	$j_0 = 2$	N	$j_0 = 0$	$j_0 = 1$	$j_0 = 2$
2	405	31	–	–	945	33	–	–
3	1953	32	21	–	4185	34	21	–
4	8505	33	21	17	17577	35	22	20
5	35433	33	22	19	72009	35	23	21
6	144585	–	22	21	291465	–	23	22
7	584073	–	–	21	1172745	–	–	22

Table 13: Arch, multilevel preconditioning: 4, 16, 64, 8, 32, 128 subdomains.

J	N	$j_0 = 0$	$j_0 = 1$	$j_0 = 2$	N	$j_0 = 0$	$j_0 = 1$	$j_0 = 2$
2	528	47	–	–	1104	55	–	–
3	2208	45	37	–	4512	59	50	–
4	9024	45	39	18	18240	57	54	40
5	36480	46	40	20	73344	58	57	45
6	146688	–	40	20	294144	–	60	45
7	588288	–	–	20	1178112	–	–	48

Table 14: Cylinder under wind load, multilevel preconditioning: 4, 16, 64, 8, 32, 128 subdomains.

Finally, the iteration numbers for the cooling tower are given in Table 15. The iteration

J	N	$j_0 = 0$	$j_0 = 1$	$j_0 = 2$	N	$j_0 = 0$	$j_0 = 1$	$j_0 = 2$
2	528	146	–	–	1104	164	–	–
3	2208	192	105	–	4512	188	95	–
4	9024	199	114	61	18240	191	97	33
5	36480	205	116	65	73344	193	99	34
6	146688	–	120	88	294144	–	100	36
7	588288	–	–	113	1178112	–	–	39

Table 15: Cooling tower, multilevel preconditioning: 4, 16, 64, 8, 32, 128 subdomains.

numbers of the arch, the full cylinder, and the cooling tower lead to the same conclusion like in the case of the plate. It is worth mentioning that in the case of the cooling tower the graph of the normal displacement u_n for the Adini elements (we use again the parametrization ($\mathbf{\Phi}$-2)) is nearly the same like that in Fig. 11 (note that we have used the same parametrization ($\mathbf{\Phi}$-2)).

8 Conclusions

For all classes of shells which were considered in Section 7, the iteration numbers of the MDS preconditioner seam to be independent of the mesh size. The iteration numbers of the Adini elements are a little bit lower than these of the BFS elements. In the case of a large number of subdomains, the iteration numbers are very low since the geometry of the shell is well described by the coarse grid system. However, the direct coarse grid solver allows only an efficient parallelization of the preconditioner for a small number of subdomains. Therefore, it may be advantageous to use other methods for coarse grid solving which are more suitable for massive parallelization. All in all, the presented preconditioner is robust and fast. Since the condition number of the subspace splitting of Section 5 is mesh size independent, it is also possible to construct corresponding multiplicative methods like multigrid. In general, they lead to a faster convergence of the PPCG method than the additive preconditioners wheras the parallelization is less effective.

A Eigenvalues of the MDS-BPX preconditioner for some examples

Here we give estimations of the eigenvalues and condition numbers of the MDS preconditioner for the examples from section 7. These estimations were calculated by the use of the gradient method described in [15]. While the estimations of the maximum eigenvalues are very close to reality, the estimations of the minimum eigenvalues are only a rough approximation to the existing ones. However, these estimations show the order of the condition numbers of the preconditioners.

A.1 Eigenvalues of the MDS-BPX preconditioner for the BFS elements

J	Unit square				L–shaped domain				Slit domain			
	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	36	0.67	1.84	2.74	132	0.36	2.28	6.30	180	0.35	2.29	6.54
3	196	0.62	2.63	4.25	644	0.32	2.98	9.37	868	0.30	2.98	9.83
4	900	0.60	3.27	5.44	2820	0.28	3.55	12.59	3780	0.27	3.56	13.37
5	3844	0.59	3.79	6.40	11780	–	–	–	15740	–	–	–

Table 16: Plates, one-dimensional calculation: eigenvalues and condition numbers of multilevel preconditioning for 1 subdomain.

J	Unit square				L–shaped domain				Slit domain			
	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	108	0.20	4.52	22.06	396	0.21	4.66	22.07	540	0.20	4.67	23.30
3	588	0.20	5.00	24.90	1932	0.20	5.13	25.62	2604	0.20	5.13	25.51
4	2700	0.20	5.16	25.64	8460	0.20	5.28	26.24	11340	0.21	5.29	25.46
5	11532	0.20	5.22	25.73	35340	–	–	–	47220	–	–	–

Table 17: Plates: eigenvalues and condition numbers of multilevel preconditioning for 1 subdomain.

J	N	$j_0 = 0\ (4 \times 1)$			$j_0 = 1\ (8 \times 2)$			$j_0 = 2\ (16 \times 4)$		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	540	0.12	6.06	51.18	–		–	–		–
3	2604	0.13	6.64	49.81	0.21	5.37	26.08	–		–
4	11340	0.14	6.76	47.23	0.21	5.84	28.44	0.21	5.00	23.82
5	47244	0.15	6.79	44.83	0.21	5.98	28.23	0.21	5.48	25.68
6	192780	–		–	0.27	5.92	22.32	0.27	5.63	20.94
7	778764	–		–	–		–	0.33	5.65	16.92

Table 18: Arch, multilevel preconditioning: eigenvalues and condition numbers for 4, 16, 64 subdomains.

J	N	$j_0 = 0\ (4 \times 2)$			$j_0 = 1\ (8 \times 4)$			$j_0 = 2\ (16 \times 8)$		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	1260	0.099	5.63	56.86	–		–	–		–
3	5580	0.11	6.00	55.92	0.18	4.97	27.93	–		–
4	23436	0.11	6.09	53.98	0.19	5.44	28.98	0.22	4.93	22.02
5	96012	0.14	6.11	44.06	0.20	5.58	27.31	0.22	5.40	24.90
6	388620	–		–	0.21	5.62	26.60	0.21	5.54	25.86
7	1563660	–		–	–		–	0.21	5.59	26.19

Table 19: Arch, multilevel preconditioning: eigenvalues and condition numbers for 8, 32, 128 subdomains.

J	N	$j_0 = 0\ (4 \times 1)$			$j_0 = 1\ (8 \times 2)$			$j_0 = 2\ (16 \times 4)$		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	704	0.057	7.19	125.99	–		–	–		–
3	2944	0.089	8.08	90.85	0.091	5.76	63.67	–		–
4	12032	0.096	8.23	85.72	0.099	6.14	62.22	0.21	4.99	23.32
5	48640	0.099	8.25	82.85	0.10	6.24	62.44	0.30	5.47	18.48
6	195584	–		–	0.11	6.27	57.80	0.30	5.61	18.94
7	784384	–		–	–		–	0.32	5.65	17.73

Table 20: Cylinder, multilevel preconditioning: eigenvalues and condition numbers for 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)			$j_0 = 1$ (8×4)			$j_0 = 2$ (16×8)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	1472	0.033	5.42	163.58	–		–	–		–
3	6016	0.027	5.82	212.19	0.026	4.81	182.48	–		–
4	24320	0.027	5.93	216.96	0.026	5.25	199.03	0.026	4.76	180.86
5	97792	0.027	5.97	218.32	0.027	5.40	201.67	0.026	5.19	198.68
6	392192	–		–	0.027	5.84	215.81	0.027	5.43	205.31
7	1570816	–		–	–		–	0.027	5.93	220.17

Table 21: Cylinder, multilevel preconditioning: eigenvalues and condition numbers for 8, 32, 128 subdomains.

J	N	$j_0 = 0$ (4×1)			$j_0 = 1$ (8×2)			$j_0 = 2$ (16×4)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	704	0.0050	13.05	2604.87	–		–	–		–
3	2944	0.0080	15.49	1928.56	0.014	12.43	865.98	–		–
4	12032	0.010	16.52	1541.66	0.019	13.78	739.06	0.060	9.65	160.08
5	48640	0.015	16.66	1136.82	0.019	11.92	613.22	0.048	10.16	213.06
6	195584	–		–	0.023	14.02	600.84	0.023	10.27	439.97
7	784384	–		–	–		–	0.012	10.29	844.18

Table 22: Cooling tower, multilevel preconditioning: eigenvalues and condition numbers for 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)			$j_0 = 1$ (8×4)			$j_0 = 2$ (16×8)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	1472	0.0068	11.71	1722.84	–		–	–		–
3	6016	0.0096	13.90	1445.00	0.016	10.79	680.90	–		–
4	24320	0.011	14.63	1310.80	0.019	11.78	615.73	0.11	7.82	73.02
5	97792	0.013	14.72	1138.06	0.021	11.94	569.41	0.12	8.27	71.90
6	392192	–		–	0.025	11.97	481.90	0.15	8.39	57.56
7	1570816	–		–	–		–	0.13	8.43	67.36

Table 23: Cooling tower, multilevel preconditioning: eigenvalues and condition numbers for 8, 32, 128 subdomains.

A.2 Eigenvalues of the MDS-BPX preconditioner for the Adini elements

J	Unit square				L–shaped domain				Slit domain			
	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	27	0.67	1.78	2.65	99	0.38	2.25	5.96	135	0.36	2.26	6.20
3	147	0.63	2.56	4.11	483	0.33	2.97	9.03	651	0.31	2.98	9.50
4	675	0.61	3.26	5.39	2119	0.29	3.56	12.21	2835	0.27	3.56	12.98
5	2883	0.59	3.79	6.38	8835	–	–	–	11805	–	–	–

Table 24: Plates, one-dimensional calculation: eigenvalues and condition numbers of multilevel preconditioning for 1 subdomain.

J	Unit square				L–shaped domain				Slit domain			
	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ	N	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	81	0.63	4.25	6.79	297	0.41	4.42	10.84	405	0.45	4.44	9.80
3	441	0.60	4.96	8.22	1449	0.34	5.10	15.19	1953	0.40	5.12	12.71
4	2025	0.61	5.16	8.41	6357	0.29	5.27	18.00	8505	0.35	5.29	15.11
5	8649	0.60	5.22	8.67	26505	–	–	–	35415	–	–	–

Table 25: Plates: eigenvalues and condition numbers of multilevel preconditioning for 1 subdomain.

J	N	$j_0 = 0$ (4×1)			$j_0 = 1$ (8×2)			$j_0 = 2$ (16×4)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	405	0.12	5.96	48.33	–		–	–		–
3	1953	0.14	6.64	48.89	0.31	5.19	16.59	–		–
4	8505	0.14	6.76	47.85	0.33	5.83	17.64	0.45	4.76	10.58
5	35433	0.14	6.79	47.12	0.37	5.98	16.14	0.39	5.45	14.13
6	144585	–		–	0.37	5.87	15.96	0.37	5.61	15.19
7	584073	–		–	–		–	0.36	5.64	15.54

Table 26: Arch, multilevel preconditioning: eigenvalues and condition numbers for 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)			$j_0 = 1$ (8×4)			$j_0 = 2$ (16×8)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	945	0.10	5.52	53.49	–		–	–		–
3	4185	0.11	6.00	54.73	0.18	4.76	26.43	–		–
4	17577	0.12	6.10	53.01	0.19	5.43	28.62	0.23	4.71	20.90
5	72009	0.14	6.13	43.24	0.21	5.58	27.19	0.22	5.38	24.77
6	291465	–		–	0.21	5.62	26.58	0.21	5.54	25.84
7	1172745	–		–	–		–	0.21	5.59	26.18

Table 27: Arch, multilevel preconditioning: eigenvalues and condition numbers for 8, 32, 128 subdomains.

J	N	$j_0 = 0$ (4×1)			$j_0 = 1$ (8×2)			$j_0 = 2$ (16×4)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	528	0.062	6.98	112.09	–		–	–		–
3	2208	0.093	8.07	87.25	0.094	5.68	60.21	–		–
4	9024	0.098	8.23	84.02	0.10	6.15	60.93	0.22	4.79	21.66
5	36480	0.10	8.25	81.53	0.11	6.26	58.98	0.32	5.45	16.80
6	146688	–		–	0.11	6.29	56.91	0.30	5.61	18.44
7	588288	–		–	–		–	0.32	5.65	17.63

Table 28: Cylinder, multilevel preconditioning: eigenvalues and condition numbers for 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)			$j_0 = 1$ (8×4)			$j_0 = 2$ (16×8)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	1104	0.033	5.33	162.14	–		–	–		–
3	4512	0.027	5.84	216.17	0.026	4.74	180.52	–		–
4	18240	0.027	5.95	217.36	0.026	5.27	199.95	0.026	4.66	177.08
5	73344	0.027	5.98	218.79	0.027	5.42	202.43	0.026	5.19	198.85
6	294144	–		–	0.027	5.80	214.39	0.027	5.43	205.31
7	1178112	–		–	–		–	0.027	5.93	220.18

Table 29: Cylinder, multilevel preconditioning: eigenvalues and condition numbers for 8, 32, 128 subdomains.

J	N	$j_0 = 0$ (4×1)			$j_0 = 1$ (8×2)			$j_0 = 2$ (16×4)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	528	0.0099	12.31	1234.37	–		–	–		–
3	2208	0.0094	15.38	1641.44	0.019	11.84	641.17	–		–
4	9024	0.011	16.52	1497.38	0.020	13.73	680.80	0.067	9.37	139.79
5	36480	0.015	16.66	1127.67	0.021	13.99	653.46	0.049	10.17	206.52
6	146688	–		–	0.023	14.03	600.95	0.023	10.28	440.60
7	588288	–		–	–		–	0.011	10.30	845.07

Table 30: Cooling tower, multilevel preconditioning: eigenvalues and condition numbers for 4, 16, 64 subdomains.

J	N	$j_0 = 0$ (4×2)			$j_0 = 1$ (8×4)			$j_0 = 2$ (16×8)		
		$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ	$\underline{\lambda}$	$\overline{\lambda}$	κ
2	1104	0.0093	11.01	1189.42	–		–	–		–
3	4512	0.010	13.80	1381.48	0.018	10.30	584.41	–		–
4	18240	0.011	14.62	1297.65	0.019	11.74	596.67	0.11	7.57	69.11
5	73344	0.013	14.72	1133.14	0.021	11.94	561.34	0.12	8.25	69.03
6	294144	–		–	0.025	11.98	478.64	0.15	8.40	57.37
7	1178112	–		–	–		–	0.13	8.44	67.41

Table 31: Cooling tower, multilevel preconditioning: eigenvalues and condition numbers for 8, 32, 128 subdomains.

References

[1] T. Apel, F. Milde, M. Thess. SPC-PM Po 3D - Programmer's Manual. Preprint SPC 95_34, Fakultät Mathematik und Naturwissenschaften, TU Chemnitz–Zwickau, 1996.

[2] F. Argyris, H.-P. Mlejnek *Die Methode der Finite Elemente I.* Friedr. Vieweg & Sohn, Braunschweig, 1986.

[3] M. Bernadou. *Finite Element Methods for Thin Shell Problems.* JohnWiley & Sons, Chichester, 1996.

[4] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1-22, 1990.

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

[6] P.G. Ciarlet and B. Miara. On the ellipticity of linear shell models. *Z. Angew. Math. Phys.*, 43(2):243-253, 1992.

[7] G. Haase, U.Langer, A.Meyer. Parallelisierung und Vorkonditionierung des CG-Verfahrens durch Gebietszerlegung. In *Teubner Scripten zur Numerik*, III, Stuttgart, 1992.

[8] W.T. Koiter. On the foundations of the linear theory of thin elastic shells. *Proc. Kon. Ned. Akad. Wetensch*, B 73:169-195, 1970.

[9] L.D. Landau and E.M. Lifschitz. *Lehrbuch der theoretischen Physik. Elastizitätstheorie.* Akademie-Verlag, Berlin, 1965.

[10] H. Matthes. *Die nichtüberlappende Gebietszerlegungsmethode zur Parallelisierung und Vorkonditionierung iterativer Verfahren zur Lösung von Platten- und Schalenproblemen.* PhD thesis, Fakultät Mathematik und Naturwissenschaften, TU Chemnitz–Zwickau, 1996.

[11] S.V. Nepomnyaschikh. Mesh theorems of traces, normalization of function traces and their inversion. *Sov. J. Numer. Anal. Math. Model.*, 6(3):223-242, 1991.

[12] V.V. Novozhilov. *Thin shell theory.* Wolters-Noordhoff Publ., Groningen, 1970.

[13] P. Oswald. *Multilevel Finite Element Approximation: Theory and Applications.* Teubner Scripten zur Numerik. B.G. Teubner, Stuttgart, 1994.

[14] P. Oswald. Multilevel preconditioners for discretizations of the biharmonic equation by rectangular finite elements. *Numerical Linear Algebra with applications*, 2(6):487-505, 1995.

[15] A. A. Samarskij and E. S. Nikolajev. *Numerical Methods for Grid Equations. Vol. II: Iterative Methods.* Birkhäuser, Basel Boston Berlin, 1989.

[16] X. Zhang. Multilevel Schwarz methods. *Numer. Math.*, 63:521–539, 1992.