Technische Universität Chemnitz

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Arnd Meyer

Hierarchical Preconditioners for Higher Order Elements and Applications in Computational Mechanics

Preprint SFB393/99-02

Abstract: From the very efficient use of hierarchical techniques for the quick solution of finite element equations in case of linear elements, we discuss the generalization of these preconditioners to higher order elements. Here, especially elements based on cubic polynomials require more complicate tasks such as the definition of ficticious spaces and the Ficticious Space Lemma. A numerical example demonstrates that iteration numbers similar to the linear case are obtained.

Preprint-Reihe des Chemnitzer SFB 393

SFB393/99-02 **Januar 1999**

Contents

Author's addresses:

Arnd Meyer
TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

http://www.tu-chemnitz.de/sfb393/

1. Solving Finite Element Equations by Preconditioned Conjugate Gradients

We consider the usual weak formulation of a second order partial differential equation:

$$\text{find } u \in H^1(\Omega) \text{ (fulfilling Dirichlet–type boundary conditiones on } \Gamma_D \subset \partial\Omega)$$

$$\text{with } a(u,v) = \langle f, v\rangle \quad \forall v \in H_0^1(\Omega) = \{v \in H^1(\Omega) : v = 0|_{\Gamma_D}\}. \tag{1.1}$$

For simplicity let Ω be a polygonal domain in $\mathbb{R}^d$ ($d = 2$ or $d = 3$), so using a fine triangulation

$\mathcal{T}_L = \{T \subset \Omega\}$, T are triangles (quadrilaterals) if $d = 2$
T are tetrahedrons (pentahedrons/hexahedrons) if d=3

with the nodes a_i (represented by their numbers i), we define a finite dimensional subspace $\mathbb{V} \subset H^1(\Omega)$ and define the finite element solution u_h from

$$a(u_h, v) = \langle f, v\rangle \quad \forall v \in \mathbb{V} \cap H_0^1(\Omega). \tag{1.2}$$

(The usual generalizations of approximating $a(\cdot,\cdot)$ or $\langle f, \cdot\rangle$ or the domain Ω from $\cup T$ are straight forward, but not considered here).
Let $\Phi = (\varphi_1(x), \ldots, \varphi_N(x))$ be the row vector of the f.e. basis functions defined in $\mathbb{V}$, then we use the mapping $\mathbb{V} \ni u \longleftrightarrow \underline{u} \in \mathbb{R}^N$ by

$$u = \Phi\underline{u}. \tag{1.3}$$

With (1.3) the equation (1.2) is transformed into the vector space, equivalently to

$$K\underline{u}^{ex} = \underline{b}, \tag{1.4}$$

when

$$K = (a(\varphi_j, \varphi_i))_{i,j=1}^N, \quad \underline{b} = (\langle f, \varphi_i\rangle)_{i=1}^N \quad \text{and} \quad u_h = \Phi\underline{u}^{ex}.$$

So we have to solve the linear system (1.4), which is large but sparse. For its dimension N approximately 10^3 or more the problems in using Gaussian elimination diverge, so we consider efficient iterative solvers, such as the conjugate gradient method with modern preconditioners. The preconditioner in the vector space is a symmetric positiv definit matrix C^{-1} (constant over the iteration process) for which 3 properties should be fulfilled as best as possible:

(P1): The action $\underline{w} := C^{-1}\underline{r}$ should be cheap ($\mathcal{O}(N)$ arithmetical operations). Here $\underline{r} = K\underline{u} - \underline{b}$ is the residual vector of an approximate solution $\underline{u} \approx \underline{u}^{ex}$ and $\underline{w}$, the preconditioned residual, has to be an appproximation to the error $\underline{u} - \underline{u}^{ex}$.

(P2): The condition number of $C^{-1}K$, i.e.

$$\kappa(C^{-1}K) = \lambda_{\max}(C^{-1}K)/\lambda_{\min}(C^{-1}K)$$

should be small, this results in a small number of $\sim \kappa(C^{-1}K)^{1/2}$ iterations for reducing a norm of $\underline{r}$ under a given tolerance ϵ.

(P3): The action $\underline{w} = C^{-1}\underline{r}$ should work in parallel according to the data distribution of all large data (all vectors/matrices with $\mathcal{O}(N)$ storage) over the processors of a parallel computer.

In the past, preconditioning was a matrix–technique (compare: incomplete factorizations), nowadays the construction of efficient preconditioners uses the analytical knowledge of the f.e. spaces. So, we transform the equation $\underline{w} = C^{-1}\underline{r}$ into the f.e. space $\mathbb{V}$ for further investigation of more complicate higher order finite elements:

Lemma 1: The preconditioning operation $\underline{w} = C^{-1}\underline{r}$ on $\underline{r} = K\underline{u} - \underline{b}$ in $\mathbb{R}^N$ is equivalent to the definition of a 'preconditioned function' $w \in \mathbb{V}$ with

$$w = \sum_{i=1}^{N} \psi_i \langle \mathfrak{r}, \psi_i \rangle$$

for a special basis Ψ in $\mathbb{V}$.

Proof: With $\underline{w} = C^{-1}\underline{r}$ we define $w = \Phi\underline{w}$. For a given $\underline{u} \in \mathbb{R}^N$, we have $u = \Phi\underline{u} \in \mathbb{V}$ and define a 'residual functional' $\mathfrak{r} \in H^{-1}(\Omega)$ with

$$a(u,v) - \langle f, v\rangle = \langle \mathfrak{r}, v\rangle \quad \forall v \in H_0^1(\Omega).$$

In the CG algorithm, we have the values $\langle \mathfrak{r}, \varphi_i\rangle$ within our residual vector $\underline{r}$:

$$\begin{aligned} \underline{r} = K\underline{u} - \underline{b} &= \Big(\sum a(\varphi_j, \varphi_i) u_j - \langle f, \varphi_i \rangle \Big)_{i=1}^{N} \\ &= \big(a(u,\varphi_i) - \langle f, \varphi_i\rangle\big)_{i=1}^{N} = \big(\langle \mathfrak{r}, \varphi_i\rangle\big)_{i=1}^{N} \end{aligned}$$

So, $w = \Phi\underline{w} = \Phi C^{-1}\underline{r}$ can be written with any factorization $C^{-1} = FF^T$ (square root of C^{-1} or Cholesky decompotion etc.) as

$$w = \Phi F F^T \underline{r} = \sum_{i=1}^{N} \psi_i \langle \mathfrak{r}, \psi_i\rangle,$$

whenever $\Psi = (\psi_1 \ldots \psi_N) = \Phi F$ is another basis in $\mathbb{V}$, transformed with the regular $(N \times N)$–matrix F.

Remark 1: If no preconditioning is used, we have $C^{-1} = F = I$, the definition of w is

$$w = \sum_{i=1}^{N} \varphi_i \langle \mathfrak{r}, \varphi_i \rangle$$

with our nodal basis Φ.

Remark 2: The well–known hierarchical preconditioner due to YSERENTANT (see next Chapter) constructs the matrix F directly from the basis transformation of nodal basis functions Φ into hierarchical base functions Ψ and the action $\underline{w} := C^{-1}\underline{r}$ is indeed

$$\underline{w} := FF^T\underline{r}.$$

2. Basic Facts on Hierarchical and BPX Preconditioners for Linear Elements

Let the triangulation $\mathcal{T}_L$ be the result of L refinement steps starting on a given coarse mesh $\mathcal{T}_0$. For simplicity let each triangle T' in $\mathcal{T}_{l-1}$ be subdivided into 4 equal subtriangles of $\mathcal{T}_l$. Then the mesh history is stored within a list of nodal numbers, where each new node $a_i = \frac{1}{2}(a_j + a_k)$ from subdividing the edge (a_j, a_k) in $\mathcal{T}_{l-1}$ is stored together with his 'father' nodes:

$$(i, j, k) = (\text{Son, Father1, Father2}).$$

This list is ordered from coarse to fine due to the history. Note that in the quadrilateral case this list contains (Son, Father1, Father2) if an edge is subdivided, but additionally

$$(\text{Son, Father1, Father2, Father3, Father4})$$

with the 'Son' as interior node and all four vertices of a quadrilateral subdivided into 4 parts. With this definition we have the finite element spaces $\mathbb{V}_l$ belonging to the triangulation $\mathcal{T}_l$ equipped with the usual nodal basis Φ_l

$$(\mathbb{V}_l = span\Phi_l, l = 0, \ldots, L).$$

All functions in this basis are piecewise (bi–)linear with respect to $\mathcal{T}_l$. From

$$\mathbb{V}_0 \subset \mathbb{V}_1 \subset \cdots \subset \mathbb{V}_L, \tag{2.1}$$

we can define a hierarchical basis Ψ_L in $\mathbb{V}_L$ recursively:
Let $\Psi_0 = \Phi_0$, from Ψ_{l-1} the hierarchical basis in $\mathbb{V}_{l-1} = span\Psi_{l-1} = span\Phi_{l-1}$ we define

$$\Psi_l = (\Psi_{l-1} \vdots \Phi_l^{new}) \tag{2.2}$$

where Φ_l^{new} contains all 'new' nodal basis functions of $\mathbb{V}_l$ belonging to nodes a_i that are new ('Sons') in $\mathcal{T}_l$ (not defined in $\mathcal{T}_{l-1}$).
For $l = L$, we have $\mathbb{V}_L = span\Psi_L = span\Phi_L$, so another basis additionally to Φ_L is defined and there exists a regular $(N \times N)$–Matrix F with $\Psi_L = \Phi_L F$ which is used in our preconditioning procedure as considered in Chapter 1.
From [11] the following facts are derived:

(P2) is fulfilled with $\kappa(C^{-1}K) = \mathcal{O}(L^2) = \mathcal{O}(|\log h|^2)$ from the good condition of the 'hierarchical stiffness matrix' $K_H = F^T K F \quad (\kappa(K_H) = \kappa(C^{-1}K))$, which is a consequence of 'good' angles between the subspaces $\mathbb{V}_{l-1}$ and $(\mathbb{V}_l - \mathbb{V}_{l-1})$.

(P1) is fulfilled from the recursiv refinement formula: Consider the spaces $\mathbb{V}_{l-1}$ and $\mathbb{V}_l$ with the bases Φ_{l-1} and Φ_l ($\dim \mathbb{V}_l = N_l$). Then from $\mathbb{V}_{l-1} \subset \mathbb{V}_l$ there is an $(N_l \times N_{l-1})$–Matrix $\tilde{P}_{l-1}$ with

$$\Phi_{l-1} = \Phi_l \tilde{P}_{l-1}. \tag{2.3}$$

Here $\tilde{P}_{l-1}$ is explicitly known

$$\tilde{P}_{l-1} = \begin{pmatrix} I \\ \cdots \\ P_{l-1} \end{pmatrix} \tag{2.4}$$

from

$$\varphi_i^{(l-1)} = \varphi_i^{(l)} + \frac{1}{2} \sum_{j \in \mathcal{N}(i)} \varphi_j^{(l)}. \tag{2.5}$$

(The sum runs over all 'new' nodes j that are neighbours of i, the set $\mathcal{N}(i)$). That is, P_{l-1} hat values $\frac{1}{2}$ at position (j, i) iff j is 'Son' of i (an edge (i, i') from $\mathcal{T}_{l-1}$ was subdivided into (i, j) and (i', j) in $\mathcal{T}_l$), so the value $\frac{1}{2}$ occurs exactly twice in each row of P_{l-1}.

From this definition for all $l = 1, \ldots, L$ follows that F is a product of transformations from level to level, from which the matrix vector multiply $\underline{w} := FF^T\underline{r}$ becomes very cheap according to the following two basic algorithms:

(A1): $y := F^T \underline{r}$ is done by:

1. $y := \underline{r}$
2. for all entries within the list (backwards) do:

$$\begin{cases} y(\text{Father1}) := & y(\text{Father1}) + \frac{1}{2} y(\text{Son}) \\ y(\text{Father2}) := & y(\text{Father2}) + \frac{1}{2} y(\text{Son}) \end{cases}$$

(A2): $\underline{w} := Fy$ is done by:

1. $\underline{w} := y$
2. for all entries within the list do:

$$w(\text{Son}) := w(\text{Son}) + \frac{1}{2}(w(\text{Father1}) + w(\text{Father2}))$$

Note: In the quadrilateral case sometimes 4 fathers exist then $\frac{1}{2}$ is to be replaced by $\frac{1}{4}$ due to another refinement equation.

Remark 1: This preconditioner works perfectly in a couple of applications in $2D$. Basically it has been successfully used for simple potential problems, but a generalization to linear elasticity problems (plane stress or plane strain 2D) is simple. Here, we use this technique for each single component of the vector function $\vec{u} \in (H^1(\Omega))^2$. The condition number $\kappa(C^{-1}K)$ is enlarged by the constant from Korn's inequality.

Remark 2: For 3D problems, a growing condition number $\kappa(C^{-1}K) = \mathcal{O}(h^{-1})$ would appear. To overcome this difficulty, the BPX preconditioner has to be used [2, 8]. According to (2.3) we have

$$\begin{array}{l} \Phi_l = \Phi_L Q_l \ \forall l = 0, \dots, L \\ (Q_l = \tilde{P}_L \cdot \ldots \cdot \tilde{P}_l \ \text{ is } \ (N \times N_l)). \end{array} \tag{2.6}$$

Then the BPX preconditioner is defined as

$$w = \mathcal{C}^{-1}\mathfrak{r} = \sum_{l=0}^{L} \sum_{i=1}^{N_l} \varphi_i^{(l)} \langle \mathfrak{r}, \varphi_i^{(l)} \rangle \cdot d_i^l \tag{2.7}$$

which is (from Chapter 1) equivalent to:

$$\underline{w} = \sum_{l=0}^{L} Q_l D_l Q_l^T \underline{r}. \tag{2.8}$$

Here, $D_l = diag(d_1^l, \dots, d_{N_l}^l)$ are scale factors, which can be chosen as $2^{(d-2)l}$ or as inverse main diagonal entries of the stiffness matrices K_l belonging to $\Phi_l : K_l = Q_l^T K Q_l$.
For this preconditioner the fact $\kappa(C^{-1}K) \leq$ const (independent of h) can be proven and the algorithm for (2.8) is similar to (A1) and (A2).

3. Generalizing Hierarchical Techniques to Higher Order Elements

From the famous properties of the preconditioning technique in Chapter 2, we should wish to construct similar preconditioners for higher order finite elements and especially for shell and plate elements.

We propose the same nested triangulation as in Chapter 2: $\mathcal{T}_0, \ldots, \mathcal{T}_L$. Let n_l be the total number of nodes in $\mathcal{T}_l$, then in Chapter 2 we had $N_l = n_l$ (it was one degree of freedom per node). Now this is different, usually $N_l > n_l$ (at least for $l = L$, where the finite element space $\mathbb{V}_L = \mathbb{V}$ for approximating our bilinear form is defined).

For using hierarchical–like techniques we have 3 possibilities:

Technique 1: For some kinds of higher order elements, we define the f.e. spaces $\mathbb{V}_l$ on each level and obtain nested spaces

$$\mathbb{V}_0 \subset \mathbb{V}_1 \subset \ldots \subset \mathbb{V}_L.$$

In this case the same procedure as in Algorithms A1/A2 can be used, but due to a more complicate refinement formula (instead of (2.5)), the algorithms are to be adapted.

Example 1: Bogner–Fox–Schmidt–elements on quadrilaterals (with bicubic functions and 4 d.o.f. per node), see [9, 10].

Technique 2: Usually we cannot guarantee that the spaces $\mathbb{V}_l$ are nested (i.e. $\mathbb{V}_{l-1} \not\subset \mathbb{V}_l$), but our finest space $\mathbb{V} = \mathbb{V}_L$ belonging to the triangulation $\mathcal{T}_L$ contains all piecewise linear functions on $\mathcal{T}_L$. So we have the nested spaces $\mathbb{V}_0^{(1)} \subset \ldots \subset \mathbb{V}_L^{(1)} \subset \mathbb{V}_L$, when $\mathbb{V}_l^{(1)}$ are the piecewise linear functions on $\mathcal{T}_l$ (as in Chapter 2). Then we have to represent $\mathbb{V}_L = \mathbb{V}_L^{(1)} \dot{+} \mathbb{W}_L$ (direct sum) and prove that $\gamma = \cos\angle(\mathbb{V}_L^{(1)}, \mathbb{W}_L) < 1$. This angle is defined from the $a(\cdot,\cdot)$ energy–inner product

$$\gamma^2 = \sup \frac{a^2(u,v)}{a(u,u)a(v,v)}, \quad \text{where} \tag{3.1}$$

the supremum is taken over all $u \in \mathbb{V}_L^{(1)}$ and $v \in \mathbb{W}_L$. If $\gamma < 1$ (independent of h), the preconditioner works as in Chapter 2 for $l = 0, \ldots, L$ (and linear elements: A1/A2) and additionally there is one transformation from the nodal basis Φ_L of $\mathbb{V}_L$ into the hierarchical basis $(\Phi_L^{(1)} \vdots \Phi_L^{new})$ of $(\mathbb{V}_L^{(1)} \dot{+} \mathbb{W}_L)$ and back. Again we have to calculate a special refinement formula for this last step:

$$\Phi_L^{(1)} = \Phi_L \tilde{P}_L. \tag{3.2}$$

Here, $\tilde{P}_L$ has another structure as $\tilde{P}_l(l < L)$ from Chapter 2. The entries of P_L are defined from

$$\varphi_i^{(1)} = \varphi_i^{higher} + \sum_{j \in N(i)} \alpha_{ij} \varphi_j^{higher}, \tag{3.3}$$

where $\Phi_L^{(1)} = (\varphi_1^{(1)}, \ldots, \varphi_{n_L}^{(1)})$ are the piecewise linear basis functions and φ_j^{higher} the f.e. basis functions that span $\mathbb{V}_L$ (for example piecewise polynomials of higher order).

Example 2: $\mathbb{V}_L = \mathbb{V}_L^{(2)}$ (piecewise quadratic polynomials on 6–node triangles). Here, $\alpha_{ij} = \frac{1}{2}$ iff i is vertex node of $\mathcal{T}_L$ and j the node on the midpoint of an edge $(a_i, a_{i'})$. From $\alpha_{ij} = \frac{1}{2}$ follows that Algorithm A1/A2 can be used without change (one level more, all edge nodes are 'sons' of the vertex nodes of this edges).

Example 3: $\mathbb{V}_L = \mathbb{V}_L^{(2)}$ (piecewise biquadratic polynomials on 9–node quadrilaterals).
Here,

$$\alpha_{ij} = \begin{cases} \frac{1}{2} & j \text{ midpoint of an edge } (i, i') \\ \frac{1}{4} & j \text{ interior node} \end{cases}$$

Again Algorithm A1/A2 works without change, one level more.

Example 4: $\mathbb{V}_L = \mathbb{V}_L^{(2,red)}$ (reduced biquadratic polynomials on 8–node quadrilaterals).
Here, $\alpha_{ij} = \frac{1}{2}$, again use Algorithm A1/A2.

Example 5: $\mathbb{V}_L = \mathbb{V}_L^{(3,red)}$ (piecewise reduced cubic polynomials on triangles, the cubic bubble is removed such that all quadratic polynomials are included). This example has to be considered, because this space occurs as ficticious space in Technique 3. Here, we define the following functions:
$u(x) \in \mathbb{V}_L$ is a reduced cubic polynomial on each $T \in \mathcal{T}_L$ (defined by 9 values on the vertices of T).
We choose $u_i = u(a_i)$ and $u_{i,j} = \frac{\partial u}{\partial s_{ij}}|_{a_i}$, the tangential derivatives along the edges of T: $a_{ij} = a_j - a_i, s_{ij} = a_{ij}/|a_{ij}|$.
Globally, we have $|\mathcal{N}(i)| + 1$ degrees of freedom at each node a_i: $u_i = u(a_i)$ and $u_{i,j} = \frac{\partial u}{\partial s_{ij}}|a_i \, \forall j \in \mathcal{N}(i)$. From this definition, we easily find

$$\varphi_i^{(1)}(x) = \varphi_i^{(3)}(x) + \sum_{j \in \mathcal{N}(i)} \frac{1}{|a_{ij}|}(\varphi_{j,i}(x) - \varphi_{i,j}(x)). \tag{3.4}$$

Here, $\varphi^{(3)}(x)$ is the f.e. basis function with

$$\varphi_i^{(3)}(a_j) = \delta_{ij} \text{ and } \nabla\varphi_i^{(3)}(a_j) = (0,0)^T \, \forall i, j \tag{3.5}$$

(with support of all T around a_i) and $\varphi_{i,j}(x)$ fulfils

$$\varphi_{i,j}(a_k) = 0\,\forall i,j,k, \quad \frac{\partial}{\partial s_{ij}}\varphi_{i,j}(a_i) = 1 \tag{3.6}$$

$$\frac{\partial}{\partial s_{ik}}\varphi_{i,j}(x) = 0$$

(along all edges $(i,k) \neq (i,j)$, so the support are two triangles that share the edge (i,j)).
In the splitting $\mathbb{V}_L^{(3,red)} = \mathbb{V}_L^{(1)} + \mathbb{W}_L$, $\mathbb{W}_L$ is spanned by all functions $\varphi_{i,j}(x)$.

Technique 3: There are examples, where neither the spaces $\mathbb{V}_l$ are nested, nor piecewise linear functions $\mathbb{V}_L^{(1)}$ are contained within the finest space $\mathbb{V}_L$. Here, we can use the Ficticious Space Lemma (see [7, 8]), constructing a preconditioner which is written as

$$\mathcal{C}^{-1} = \mathcal{R}\tilde{\mathcal{C}}^{-1}\mathcal{R}^*, \tag{3.7}$$

if a ficticious space $\tilde{\mathbb{V}}$ exists (in general of higher dimension than $\mathbb{V}_L$) having a good preconditioner $\tilde{\mathcal{C}}^{-1}$. The key is the definition of a restriction operator

$$\mathcal{R} : \tilde{\mathbb{V}} \to \mathbb{V}_L$$

with small energy norm.

Example 6: For more complicate plate analysis, the space $\mathbb{V}^{HC}$ of Hermite–triangles is used (DKT–elements, HCT–elements). Here $u \in \mathbb{V}^{HC}$ is again a reduced cubic polynomial on each $T \subset \mathcal{T}_L$, but we use globally 3 degrees of freedom per node: $u_i = u(a_i)$ and $\vec{\Theta}_i = \nabla u(a_i)$.
Here, ∇u is discontinuous on the edges, but continuous on each node a_i. From this property the spaces cannot be nested and the functions in $\mathbb{V}_L^{(1)}$ have discontinuous gradients on all vertices a_i, so $\mathbb{V}_L^{(1)} \not\subset \mathbb{V}^{HC}$. From $\mathbb{V}^{HC} \subset \mathbb{V}^{(3,red)}$, we define a restriction operator $\mathcal{R} : \mathbb{V}_L^{(3,red)} \to \mathbb{V}_L = \mathbb{V}^{HC}$ and use the preconditioner for $\mathbb{V}_L^{(3,red)}$ in Example 5 for defining $\tilde{\mathcal{C}}^{-1}$, which leads to a good preconditioner for this space $\mathbb{V}^{HC}$. The definition of $\mathcal{R}$ is not unique, we use an easy choice, some kind of averaging of $\frac{\partial u}{\partial s_{ij}}|_{a_i}$ to $\nabla u(a_i)$:

Let $\tilde{u} \in \mathbb{V}_L^{(3,red)}$ be represented by

$$u_i = \tilde{u}(a_i) \text{ and } u_{i,j} = \frac{\partial \tilde{u}}{\partial s_{ij}}|_{a_i} \quad (\forall i, \forall j \in \mathcal{N}(i))$$

then we define $u = \mathcal{R}\tilde{u} \in \mathbb{V}^{HC}$ with $u_i = u(a_i)$ and $\vec{\Theta}_i = \nabla u(a_i)$ from

$$\vec{\Theta}_i = \frac{1}{m_i} S_i\, \underline{u}_i \quad (\underline{u}_i = (u_{i,j})\forall j).$$

The matrix S_i contains all normalized vectors s_{ij} for all edges meeting a_i. So,

$$\vec{\Theta}_i = \frac{1}{m_i} \sum_{j \in \mathcal{N}(i)} s_{ij} \cdot \frac{\partial u}{\partial s_{ij}}|_{a_i} = \frac{1}{m_i} \sum u_{i,j} s_{ij} \tag{3.8}$$

4. Numerical Examples

Let us demonstrate the preconditioners proposed in Chapter 3 at one example that allows some of the finite elements discussed in Example 1 to Example 6.
So, we choose Ω as a rectangle with prescribed Dirichlet type boundary conditions and solve a simple Laplace equation

$$\left.\begin{array}{rcl} -\triangle u &=& 0|_{\Omega} \\ u &=& g|_{\partial\Omega}, \end{array}\right\}$$

so, $\quad a(u,v) = \int\limits_{\Omega} (\nabla u) \cdot (\nabla v) d\Omega \quad$ and we use the following discretization:

a) piecewise linear functions ($\mathbb{V}_L = \mathbb{V}_L^{(1)}$)
on a triangular mesh (3–node–triangles)

(b) piecewise bilinear functions ($\mathbb{V}_L = \mathbb{V}_L^{(1)}$)
on a quadrilateral mesh (4–node–quadrilaterals)

(c) piecewise quadratic functions ($\mathbb{V}_L = \mathbb{V}_L^{(2)}$)
on a triangular mesh (6–node–triangles)

(d) piecewise biquadratic functions ($\mathbb{V}_L = \mathbb{V}_L^{(2)}$)
on quadrilaterals (9–node–quadrilaterals)

(e) piecewise reduced quadratic functions ($\mathbb{V}_L = \mathbb{V}_L^{(2,red)}$)
on quadrilaterals (8–node–quadrilaterals)

(f) piecewise reduced cubic functions ($\mathbb{V}_L = \mathbb{V}^{HC}$)
on a triangular mesh (Hermite cubic triangles)

The preconditioners for cases (a) to (e) were simply described in Examples 1 to 4 in Chapter 3. We will give the matrix representation for the preconditioner of case (f) from combining Examples 5 and 6:
For $\mathbb{V}_L = \mathbb{V}^{HC}$, we have to solve

$$a(u,v) = \langle f, v \rangle \ \forall v \in \mathbb{V}_L \cap H_0^1(\Omega)$$

in solving

$$K\underline{u}^{ex} = \underline{b}.$$

Here $u = \Phi\underline{u}$ is represented by $u_i = u(a_i)$ and $\vec{\Theta}_i = \nabla u(a_i)$, so $\underline{u} \in \mathbb{R}^{3n}$. The preconditioner $\mathcal{C}^{-1}$ from Chapter 1 is an operator which maps the residual functional $\mathfrak{r}(u)$ to the preconditioned function w. From the ficticious space lemma, we set

$$\mathcal{C}^{-1} = \mathcal{R}\tilde{\mathcal{C}}^{-1}\mathcal{R}^*$$

with $\mathcal{R} : \tilde{\mathbb{V}} = \mathbb{V}_L^{(3,red)} \to \mathbb{V}_L = \mathbb{V}^{HC}$ as in Example 6. In the ficticious space $\tilde{\mathbb{V}}$, we use the preconditioner of Example 5. From Example 5 the matrix representation of this preconditioner is

$$\tilde{C}^{-1} = Q_L \begin{pmatrix} FF^T & \mathbb{O} \\ \mathbb{O} & I \end{pmatrix} Q_L^T \tag{4.1}$$

with F from Chapter 1. Here, Q_L transforms the basis $(\Phi^{(3)} \vdots \Phi^{(edge)})$ of the cubic functions in $\mathbb{V}^{(3,red)}$ into the hierarchical basis $(\Phi^{(1)} \vdots \Phi^{(edge)})$.

$\Phi^{(1)}$	$=$	$(\varphi_1^{(1)}, \ldots, \varphi_n^{(1)})$	piecewise linear functions
$\Phi^{(3)}$	$=$	$(\varphi_1^{(3)}, \ldots, \varphi_n^{(3)})$	piecewise reduced cubic with property (3.5)
$\Phi^{(edge)}$	$=$	$(\varphi_{i,j}) \forall$ edges (i,j) at node i	piecewise reduced cubic with property (3.6)

So,

$$Q_L = \begin{pmatrix} I & \vdots & \mathbb{O} \\ P_L & \vdots & I \end{pmatrix} \tag{4.2}$$

and entries of P_L are found in (3.4). Combining this with Example 6, we have

$$C^{-1} = RQ_L \begin{pmatrix} FF^T & \mathbb{O} \\ \mathbb{O} & I \end{pmatrix} Q_L^T R^T, \tag{4.3}$$

when R is the matrix representation of $\mathcal{R}$. The implementation of C^{-1} is the following algorithm (note that RQ_L is done at once for saving storage, this is a $(3n \times 3n)$–matrix).
The preconditioner C^{-1} acts on a residual vector $\underline{r} \in \mathbb{R}^{3n}$ $(n = n_L)$ with the entries

$$r_i = \langle \mathfrak{r}, \varphi_{i,0} \rangle \quad \text{and} \quad \tilde{\Theta}_i = \begin{pmatrix} \langle \mathfrak{r}, \varphi_{i,1} \rangle \\ \langle \mathfrak{r}, \varphi_{i,2} \rangle \end{pmatrix}$$

defined with the basis functions $(\varphi_{i,\alpha})$ in $\mathbb{V}^{HC}$:

$$\begin{array}{ll} \varphi_{i,0}(a_j) = \delta_{ij} & \varphi_{i,1}(a_j) = \varphi_{i,2}(a_j) = 0 \\ \nabla\varphi_{i,0}(a_j) = (0,0)^T & \nabla\varphi_{i,1}(a_j) = \delta_{ij}(1,0)^T \\ & \nabla\varphi_{i,2}(a_j) = \delta_{ij}(0,1)^T. \end{array}$$

Analogously, the result $\underline{w} = C^{-1}\underline{r}$ contains entries w_i and $\vec{\Theta}_i$ (for $w(a_i)$ and $\nabla w(a_i)$). From the definition

$$C^{-1} = R\tilde{C}^{-1}R^T = RQ_L \begin{pmatrix} FF^T & \mathbb{O} \\ \mathbb{O} & I \end{pmatrix} Q_L^T R^T$$

we have to implement the matrix-vector-multiply

$$(RQ_L)^T \underline{r}$$

and $\underline{w} = RQ_L y$, which is done at once for saving storage, i.e. there is no need to store the (approximately 7n) values $u_{i,j} = s_{ij}^T \vec{\Theta}_i$. This is contained in Algorithms B1 (for $y := (RQ_L)^T \underline{r}$) and B2 (for $\underline{w} := RQ_L \underline{y}$).

Algorithm B1: for each edge (i, j) do

$$\left| \begin{array}{l} y_i := r_i - a_{ij}^T(\tilde{\Theta}_i + \tilde{\Theta}_j)/|a_{ij}|^2 \\ y_j := r_j + a_{ij}^T(\tilde{\Theta}_i + \tilde{\Theta}_j)/|a_{ij}|^2 \end{array} \right.$$

Algorithm B2: for each edge (i, j) do

$$\left| \begin{array}{l} \vec{\Theta}_i := \vec{\Theta}_i + a_{ij}(a_{ij}^T\tilde{\Theta}_i + w_j - w_i)/|a_{ij}|^2 \\ \vec{\Theta}_j := \vec{\Theta}_j + a_{ij}(a_{ij}^T\tilde{\Theta}_j + w_j - w_i)/|a_{ij}|^2 \end{array} \right.$$

$\vec{\Theta}_i$ are evaluated from a Jacobi–preconditioning on the input $\tilde{\Theta}_i$, w_i are the results of Algorithms A1/A2 on the n–vector y.
The hierarchical–like preconditioners for the elements (a) to (f) require a very small number of PCG–iterations as presented in the following table.

L	n^*	(a)	(b)	(c)	(d)	(e)	(f)*
1	289	17	13	19	14	13	19
2	1,089	20	15	22	17	15	21
3	4,225	23	18	24	19	17	24
4	16,641	25	20	26	20	19	26
5	66,049	26	21	27	23	21	27
6	283,169	27	23	28	24	22	28
7	1,050,625	28	25	29	–	23	–

* note that n is the number of nodes in the finest mesh $\mathcal{T}_L$ which is equal to dimension N of the linear system in cases (a) to (d), in (e) $N \approx \frac{3}{4}n$ but in (f) N is $3n$.

References

[1] J. H. Bramble, J. E. Pasciak, A. H. Schatz (1986-89): The Construction of Preconditioners for Elliptic Problems by Substructuring I – IV,
Mathematics of Computation,
47,(1986) 103–134,
49,(1987) 1–16,
51,(1988) 415–430,
53,(1989) 1–24.

[2] I. H. Bramble, J. E. Pasciak, J. Xu (1990): Parallel Multilevel Preconditioners,
Math. Comp. **55** 191, 1-22.

[3] G. Haase, U. Langer, A. Meyer (1991): The Approximate Dirichlet Domain Decomposition Method,
Part I: An Algebraic Approach,
Part II: Application to 2nd-order Elliptic B.V.P.s,
Computing **47** 137-151/153-167.

[4] G. Haase, U. Langer, A. Meyer (1992): Domain Decomposition Preconditioners with Inexact Subdomain Solvers,
J. Num. Lin. Alg. with Appl. **1** 27-42.

[5] G. Haase, U. Langer, A. Meyer, S.V.Nepommnyaschikh (1994): Hierarchical Extension Operators and Local Multigrid Methods in Domain Decomposition Preconditioners,
East-West J. Numer. Math. **2** 173-193.

[6] A. Meyer (1990): A Parallel Preconditioned Conjugate Gradient Method Using Domain Decomposition and Inexact Solvers on Each Subdomain,
Computing **45** 217-234.

[7] S.V.Nepommnyaschikh (1991): Ficticious components and subdomain alternating methods,
Sov.J.Numer.Anal.Math.Model. **5** 53-68.

[8] P.Oswald (1994): Multilevel Finite Element Approximation: Theory and Applications,
Teubner Skripten zur Numerik, B.G.Teubner Stuttgart 1994.

[9] M.Thess (1998): Parallel Multilevel Preconditioners for Thin Smooth Shell Finite Element Analysis,
Num.Lin.Alg.with Appl., to appear

[10] M.Thess (1998): Parallel Multilevel Preconditioners for Thin Smooth Shell Problems,
Diss.A, TU Chemnitz.

[11] H. Yserentant (1990): Two Preconditioners Based on the Multilevel Splitting of Finite Element Spaces,
Numer. Math. **58** 163-184.

A The calculation of γ in Example 5

According to (3.1), we have to estimate γ^2 with $(a(u,w))^2 \le \gamma^2 a(u,u)a(w,w)$ for all $u \in \mathbb{V}_L^{(1)}$ (piecewise linear functions) and all $w \in \mathbb{W}_L = span(\varphi_{i,j})$ from (3.6) It is well–known that γ can be estimated from the element level, so we define $a_T(u,v) = \int\limits_T (\nabla u)\cdot(\nabla v)d\Omega$ for each $T \in \mathcal{T}_L$, yielding

$$a(u,v) = \sum_T a_T(u,v)$$

and estimate γ_T with $(a_T(u,w))^2 \le \gamma_T^2 a_T(u,u)a_T(w,w)$ for $u \in \mathbb{V}_L^{(1)}, w \in \mathbb{W}_L$. The required inequqality (3.1) follows from Cauchy–Schwarz in the sum over T

$$\begin{aligned}
(a(u,w))^2 &= \left(\sum_T a_T(u,w)\right)^2 \le \left(\sum_T \gamma_T a_T(u,u)^{1/2} a_T(w,w)^{1/2}\right)^2 \\
&\le \max_T \gamma_T^2 \cdot \sum_T a_T(u,u)\cdot \sum_T a_T(w,w) \\
\text{with } \gamma &\le \max_{T\in\mathcal{T}_L} \gamma_T.
\end{aligned}$$

Hence, we consider u as linear on T and w is a reduced cubic polynomial on T with zero values on the vertices defined by 6 values of the tangential derivatives at the edges. Let T have the vertices a_1, a_2 and a_3 (say), hence $A = \left(a_2 - a_1 \,\vdots\, a_3 - a_1\right)$ transforms the element T onto the master element $\hat{T}$ (with $\hat{a}_1 = (0,0)^T, \hat{a}_2 = (1,0), \hat{a}_3 = (0,1)^T$):

$$x = A\hat{x} + a_1 \longleftrightarrow \hat{x} = A^{-1}(x - a_1).$$

The linear function u is given by its three values $u_i = u(a_i)$, so we have

$$\nabla u = A^{-T}\begin{pmatrix} u_2 - u_1 \\ u_3 - u_1 \end{pmatrix} \quad \text{on } T.$$

Let $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_2 - u_1 \\ u_3 - u_1 \end{pmatrix}$ for abbreviating this, then

$$a_T(u,u) = v^T(A^TA)^{-1}v \cdot \frac{1}{2}\det A.$$

For considering w on T, we have to introduce shape functions $\hat{\varphi}_{i,j}$, on $\hat{T}$, i.e. functions with values $\hat{\varphi}_{i,j}(\hat{a}_k) = 0$ and derivatives $\frac{\partial}{\partial \hat{s}_{ij}}\hat{\varphi}_{i,k}(\hat{a}_l) = \delta_{ik}\cdot\delta_{il}$ (all indicees between 1 and 3, $i \ne j$ and $\hat{s}_{ij} = (\hat{a}_j - \hat{a}_i)/|\hat{a}_j - \hat{a}_i|$).

For fulfilling (3.6) we have to define $\varphi_{i,j}(x) = \hat{\varphi}_{i,j}(\hat{x} = A^{-1}(x - a_1)) \cdot d_{i,j}$. The scale factors $d_{i,j} (j \neq i)$ are inserted into the (6×6)–diagonal matrix

$$\begin{aligned} D &= \text{diag}\,(d_{1,2}, d_{1,3}, d_{2,3}, d_{2,1}, d_{3,1}, d_{3,2}) \\ &= \text{diag}\,\left(|a_{12}|, |a_{13}|, |a_{23}|/\sqrt{2}, |a_{12}|, |a_{13}|, |a_{23}|/\sqrt{2}\right) \end{aligned}$$

and will not influence γ_T.
Hence, a function $w \in \mathbb{W}_L$ on T is represented by the 6–vector $\underline{w} = (w_{1,2}, w_{1,3}, w_{2,3}, w_{2,1}, w_{3,1}, w_{3,2})^T$ as $w = \sum \varphi_{i,j} \cdot w_{i,j} = \sum \hat{\varphi}_{i,j} d_{i,j} w_{i,j}$, so

$$a_T(w, w) = \int_T (\nabla w) \cdot \nabla w d\Omega = \underline{w}^T\, D\, B\, D\, \underline{w}$$

with the (6×6)–element matrix

$$B = \left(\int_T (A^{-T} \hat{\nabla} \hat{\varphi}_{i,j}) \cdot (A^{-T} \hat{\nabla} \hat{\varphi}_{k,l}) d\Omega \right)_{(i,j)(k,l)} .$$

Hence,

$$a_T(w, w) \geq \frac{\det A}{\rho(A^T A)} (D\underline{w})^T \hat{B} (D\underline{w})$$

with the element matrix of the master element

$$\hat{B} = \tfrac{1}{360} \hat{D} \begin{pmatrix} 17 & -3 & 10 & -8 & 8 & 4 \\ -3 & 17 & 0 & -4 & 6 & 8 \\ 10 & 0 & 23 & -13 & 7 & 5 \\ -8 & -4 & -13 & 19 & -17 & -11 \\ 8 & 6 & 7 & -17 & 47 & 25 \\ 4 & 8 & 5 & -11 & 25 & 19 \end{pmatrix} \hat{D}$$
$$\hat{D} = \text{diag}\,(1, 1, 1, \sqrt{2}, 1, \sqrt{2}).$$

In the same way $a_T(u, w) = \det A \cdot v^T A^{-1} A^{-T} \hat{C} D \underline{w}$ with the (2×6)–matrix

$$\hat{C} = \left(\int_{\hat{T}} \hat{\nabla} \hat{\varphi}_{i,j} d\Omega \right)_{(i,j)} = \frac{1}{12} \begin{pmatrix} 0 & -1 & 0 & 1 & -1 & -1 \\ -1 & 0 & -1 & 1 & -2 & -1 \end{pmatrix} \hat{D}$$

so,

$$\begin{aligned} \gamma_T^2 &\leq \max \frac{[v^T (A^T A)^{-1} \hat{C} (D\underline{w}]^2}{(D\underline{w})^T \hat{B} (D\underline{w}) \cdot v^T (A^T A)^{-1} v} \cdot 2\rho(A^T A) \\ &= 2\rho(A^T A) \cdot \rho\left((A^T A)(A^T A)^{-1} \hat{C} \hat{B}^{-1} \hat{C}^T (A^T A)^{-T} \right) \\ &\leq \frac{\rho(A^T A)}{\lambda_{\min}(A^T A)} 2\rho(\hat{C} \hat{B}^{-1} \hat{C}^T). \end{aligned}$$

A direct calculation yields $\hat{C}\hat{B}^{-1}\hat{C}^T = \begin{pmatrix} 40 & 7 \\ 7 & 40 \end{pmatrix} \frac{360}{330 \cdot 144}$ so

$$\gamma^2 \leq 0,72\ (\kappa_2(A))^2$$

with the spectral condition number of A, which is small for good meshes. So, γ is a constant smaller than 1.

B The Use of the Ficticious Space Lemma

From [8] we recall the Ficticious Space Lemma in a simplified manner as required here:

Ficticious Space Lemma: Let $\mathbb{V} \subset \tilde{\mathbb{V}} \subset \mathbb{H}$ Banach spaces with $\mathcal{A} : \mathbb{H} \to \mathbb{H}^*$ an operator equivalent to $\langle \mathcal{A}u, v \rangle = a(u,v)\ \forall u, v \in \mathbb{H}$, where $\langle \cdot, \cdot \rangle$ denotes the dual pairing $\mathbb{H}^* \times \mathbb{H} \to \mathbb{R}$

1) Let $\mathcal{R} : \tilde{\mathbb{V}} \to \mathbb{V}$ with $a(\mathcal{R}\tilde{u}, \mathcal{R}\tilde{u}) \leq c_R\, a(\tilde{u}, \tilde{u})\ \forall \tilde{u} \in \tilde{\mathbb{V}}$

2) Let $\mathcal{Q} : \mathbb{V} \to \tilde{\mathbb{V}}$ with $\mathcal{R}\mathcal{Q}u = u\ \forall u \in \mathbb{V}$
and $a(\mathcal{Q}u, \mathcal{Q}u) \leq c_Q^{-1} a(u,u)\ \forall u \in \mathbb{V}$.

3) Let $\tilde{\mathcal{C}}^{-1} : \mathbb{H}^* \to \tilde{\mathbb{V}}$ be a good preconditioner for $\mathcal{A}$ in $\tilde{\mathbb{V}}$:

$$\underline{\alpha}\ a(\tilde{u}, \tilde{u}) \leq a(\tilde{\mathcal{C}}^{-1}\mathcal{A}\tilde{u}, \tilde{u}) \leq \bar{\alpha}\ a(\tilde{u}, \tilde{u})\ \forall \tilde{u} \in \tilde{\mathbb{V}}$$

Then $\mathcal{C}^{-1} = \mathcal{R}\tilde{\mathcal{C}}^{-1}\mathcal{R}^*$ is a good preconditioner for $\mathcal{A}$ in $\mathbb{V}$ i.e.

$$\underline{\gamma} a(u,u) \leq a(\mathcal{C}^{-1}\mathcal{A}u, u) \leq \bar{\gamma} a(u,u)\ \forall u \in \mathbb{V}$$

with $\bar{\gamma}/\underline{\gamma} \leq (\bar{\alpha}/\underline{\alpha}) \cdot (c_R\ c_Q)$.

Remark: If $\mathbb{V}, \tilde{\mathbb{V}}$ are finite dimensional subspaces with a mesh parameter h, then $c_R, c_Q, \underline{\alpha}, \bar{\alpha}$ do not depend on h.
So, we have to define and investigate

$$\begin{aligned} & \mathcal{R} : \tilde{\mathbb{V}} = \mathbb{V}^{(3,red)} \to \mathbb{V} = \mathbb{V}^{HC} \\ \text{and} \quad & \mathcal{Q} : \mathbb{V} \to \tilde{\mathbb{V}}. \end{aligned}$$

From $\mathbb{V} = \mathbb{V}^{HC} \subset \tilde{\mathbb{V}} = \mathbb{V}^{(3,red)}$ we can expand all functions with resprect to the basis $\tilde{\Phi}$ in $\tilde{\mathbb{V}} = \mathbb{V}^{(3,red)}$.

Hence, $\tilde{u} \in \tilde{\mathbb{V}}$ is $\tilde{u} = \tilde{\Phi}\underline{\tilde{u}}$ and $a(\tilde{u}, \tilde{u}) = \underline{\tilde{u}}^T \tilde{K} \underline{\tilde{u}}$. Again $u \in \mathbb{V}$ is represented as $u = \tilde{\Phi}\underline{u}$, $a(u, u) = \underline{u}^T \tilde{K} \underline{u}$. Let $u = \mathcal{R}\tilde{u}$ then we have to calculate

$$\begin{aligned} c_R &= \max_{\tilde{u} \in \tilde{\mathbb{V}}} a(\mathcal{R}\tilde{u}, \mathcal{R}\tilde{u}) / a(\tilde{u}, \tilde{u}) \\ &= \max_{\underline{\tilde{u}}} (\underline{u}^T \tilde{K} \underline{u}) / (\underline{\tilde{u}}^T \tilde{K} \underline{\tilde{u}}) \end{aligned}$$

where $\underline{u}$ is to be defined as coefficient vector of $u = \mathcal{R}\tilde{u}$ w.r.t. $\tilde{\Phi}$. On each node a_i, $\tilde{u} \in \tilde{\mathbb{V}}$ is respresented by $u_i = \tilde{u}(a_i)$ and by a small vector $\underline{\tilde{u}}_i = (\tilde{u}_{i,j}) = (\frac{\partial}{\partial s_{ij}} \tilde{u}|_{a_i})$ for each edge meeting a_i. For defining $u = \mathcal{R}\tilde{u} \in \mathbb{V}^{HC}$ we let $u_i = \tilde{u}(a_i) = u(a_i)$ and set $\Theta_i = \nabla u(a_i)$ as $\Theta_i = \frac{1}{m_i} S_i \underline{\tilde{u}}_i$ with a $(2 \times m_i)$–matrix S_i containg $s_{ij} \forall_j$ as its columns. Obvionsly $m_i > 2$, hence it is a projection from $\underline{\tilde{u}}_i \in \mathbb{R}^{m_i}$ to $\Theta_i \in \mathbb{R}^2$. From the fact that $\tilde{u}_{i,j}$ is the value $\frac{\partial u}{\partial s_{ij}}|_{a_i}$ for each function $u \in \tilde{\mathbb{V}}$, we can represent $u \in \mathbb{V}$ from the values $u_i = u(a_i) = \tilde{u}(a_i)$ and $u_{i,j} = s_{ij}^T \Theta_i$ so $\underline{u}_i = \frac{1}{m_i} S_i^T S_i \underline{\tilde{u}}_i$ defines the coefficients of u w.r.t. $\tilde{\Phi}$. Hence

$$c_R = \max \frac{\underline{\tilde{u}}^T D^T \tilde{K} D \underline{\tilde{u}}}{\underline{\tilde{u}}^T \tilde{K} \underline{\tilde{u}}}$$

with the block diagonal matrix

$$D = \operatorname{diag}\left(1, \frac{1}{m_1} S_1^T S_1, 1, \frac{1}{m_2} S_2^T S_2, \ldots, 1, \frac{1}{m_n} S_n^T S_n\right).$$

The entries of $S_i^T S_i$ are inner products of the normalized edge vectors: $s_{ij}^T s_{ik}$, so $\|\frac{1}{m_i} S_i^T S_i\|_2 < \|\frac{1}{m_i} S_i^T S_i\|_\infty < 1$.
From a splitting of D and of the identity matrix into $D = I_1 + D_2, I = I_1 + I_2$ with

$$\begin{aligned} I_1 &= \operatorname{diag} \;(1, \; \mathbb{O} \;, 1, \; \mathbb{O}, \; \cdots, \; 1, \; \mathbb{O}\;) \\ I_2 &= \operatorname{diag} \;(0, \; I \;, 0, \; I, \; \cdots, \; 0, \; I\;) \\ D_2 &= \operatorname{diag} \;(0, \; \tfrac{1}{m_1} S_1^T S_1 \;, 0, \; \cdots, \; 0, \; \tfrac{1}{m_n} S_n^T S_n) \end{aligned}$$

and $\underline{u}_1 = I_1 \underline{\tilde{u}}, \underline{u}_2 = I_2 \underline{\tilde{u}}$ we obtain

$$c_R = \max \frac{\underline{u}_1^T K_{11} \underline{u}_1 + 2 \underline{u}_1^T K_{12} D_2 \underline{u}_2 + \underline{u}_2^T D_2 K_{22} D_2 \underline{u}_2}{\underline{u}_1^T K_{11} \underline{u}_1 + 2 \underline{u}_1^T K_{12} \underline{u}_2 + \underline{u}_2^T K_{22} \underline{u}_2}$$

with the matrix blocks $K_{ij} = I_i \tilde{K} I_j$ $(i, j = 1, 2)$.
The matrix K_{22} is well–conditioned due to the fact that B from Appendix 1 is the element contribution to K_{22} from an element T. Hence, from $\|D_2\|_2 < 1$ we have

$$\underline{u}_2^T D_2 K_{22} D_2 \underline{u}_2 \leq \kappa_0 \; \underline{u}_2^T K_{22} \underline{u}_2$$

with a constant $\kappa_0 \leq \kappa(K_{22})$.

Again the complete estimation af c_R requires a 'good' angle between the subspace spanned by $\varphi_{i,0}(x)$ (defining K_{11}) and the subspace spanned by $\varphi_{i,j}(x)$ (defining K_{22}). Analogously to Appendix 1 we have on the master element level

$$\gamma^2_{master} = \max_{x \in \mathbb{R}^3} \frac{(\hat{B}_{12}\hat{B}_{22}^{-1}\hat{B}_{12}^T x, x)}{(\hat{B}_{11}x, x)} = \frac{163}{220} \,.$$

Here, $\hat{B}_{22}$ is the matrix $\hat{B}$ in Appendix 1 and

$$\hat{B}_{12}\hat{B}_{22}^{-1}\hat{B}_{12}^T = \frac{1}{360} \cdot \frac{1}{11} \begin{pmatrix} 3260 & -1630 & -1630 \\ -1630 & 1508 & 122 \\ -1630 & 122 & 1508 \end{pmatrix}$$

$$\hat{B}_{11} = \frac{1}{360} \begin{pmatrix} 400 & -200 & -200 \\ -200 & 208 & -8 \\ -200 & -8 & 208 \end{pmatrix} .$$

Hence, there is a constant γ (slightly larger than γ_{master}, but smaller than 1) with

$$(\underline{u}_1^T K_{12} \underline{u}_2)^2 \le \gamma^2 \cdot \underline{u}_1^T K_{11} \underline{u}_1 \cdot \underline{u}_2^T K_{22} \underline{u}_2, \forall \underline{u}_1, \underline{u}_2,$$

and the constant c_R follows from the inequalities

$$\begin{aligned} &\underline{u}_1^T K_{11}\underline{u}_1 + 2\underline{u}_1^T K_{12}(D_2\underline{u}_2) + \underline{u}_2^T D_2 K_{22} D_2 \underline{u}_2 \\ &\le (1+\gamma)(\underline{u}_1^T K_{11}\underline{u}_1 + \underline{u}_2^T D_2 K_{22} D_2 \underline{u}_2) \\ &\le \kappa_0 (1+\gamma)(\underline{u}_1^T K_{11}\underline{u}_1 + \underline{u}_2^T K_{22}\underline{u}_2), \end{aligned}$$

$$\begin{aligned} &\underline{u}_1^T K_{11}\underline{u}_1 + 2\underline{u}_1^T K_{12}\underline{u}_2 + +\underline{u}_2^T K_{22}\underline{u}_2 \\ &\ge (1-\gamma)(\underline{u}_1^T K_{11}\underline{u}_1 + \underline{u}_2^T K_{22}\underline{u}_2) \end{aligned}$$

with $c_R \le \kappa_0(1+\gamma)/(1-\gamma)$.

In the same way, the operator $\mathcal{Q}$ is analyzed. For defining $\mathcal{Q} : \mathbb{V} \to \tilde{\mathbb{V}}$ we have to guarantee $\mathcal{R}\mathcal{Q}u = u$. Hence, if $\mathcal{R}$ was represented by $\underline{\tilde{u}}_i \to \Theta_i = \frac{1}{m_i} S_i \underline{\tilde{u}}_i$, we define $\tilde{u} = \mathcal{Q}u \Longleftrightarrow \underline{\tilde{u}}_i = m_i S_i^T (S_i S_i^T)^{-1} \Theta_i$.
The estimation of c_Q is very similar to c_R.