Technische Universität Chemnitz

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Sven Beuchler

A preconditioner for solving the inner problem of the p-version of the FEM

Preprint SFB393/00-25

Abstract

Finding a fast solver for the inner problem of the p-version of the FEM is a difficult question. We discovered, that the system matrix for the inner problem in $2D$ has a similar structure to matrices resulting from discretizations of $-y^2\frac{\partial^2}{\partial x^2} - x^2\frac{\partial^2}{\partial y2}$ in the unit square using h-version of the FEM or finite differences. Applying multi-grid methods with special smoothers, we have a fast solver for the p-version of the FEM.

Preprint-Reihe des Chemnitzer SFB 393

SFB393/00-25 **May 2000**

Author's addresses:

Sven Beuchler
TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz
Germany

email: sven.beuchler@mathematik.tu-chemnitz.de

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

Contents

1 Introduction

Jensen/Korneev [9] and Ivanov/Korneev [7],[8] developed preconditioners for the $p-$version of the FEM in a two-dimensional domain. They used DD-methods. The unknowns are splitted into 3 groups, the interior, the edge and vertex unknowns. The vertex unknowns can be solved separately, cf. Lemma 2.3 [7]. Computing the other unknowns, we factorize the remaining stiffness matrix as follows.

$$\begin{pmatrix} A_{edg} & A_{edg,int} \\ A_{int,edg} & A_{int} \end{pmatrix} = \begin{pmatrix} I & A_{edg,int}A_{int}^{-1} \\ & I \end{pmatrix} \begin{pmatrix} S & \\ & A_{int} \end{pmatrix} \begin{pmatrix} I & \\ A_{int}^{-1}A_{int,edg} & I \end{pmatrix}$$

with the Schur-komplement

$$S = A_{edg} - A_{edg,int}A_{int}^{-1}A_{int,edg}.$$

Computing the interior unknowns, we solve a Dirichlet problem on each quadrangle. The vertex unknowns are computed via the Schur-komplement S.
We need 3 tools, a preconditioner for the interior problem, a preconditioner for the Schur-komplement and a extension operator from the edges of a quadrangle to the interior. Ivanov/Korneev derived 3 types $C_{i,S}$ of preconditioning the Schur-komplement. The condition number for $C_{i,S}^{-1}S$ is in the worst case $\mathcal{O}(log^2 p)$, where p is the polynomial degree. The solution of $C_{i,s}x = y$ costs $\mathcal{O}(p^2)$ arithmetical operations.
Furthermore, Jensen/Korneev found a spectral equivalent preconditioner for the interior problem, which has $\mathcal{O}(p^2)$ nonzero entries. In the case of parallelogram elements, the element stiffness matrix has $\mathcal{O}(p^2)$ nonzero entries, too. But, the suggested methods compute the solution in $\mathcal{O}(p^3)$ arithmetical operations. Finding a fast solver for the preconditioner was an open question. This paper is concerned to the construction more efficient preconditioner for the interior problem.
We derive a preconditioner for the interior problem, such that the number of iterations of the PCG-method shows an increasing as $\mathcal{O}(\log p)$ or less in numerical experiments and costs of $\mathcal{O}(p^2)$ arithmetical operations. The origin of this preconditioner is the multi-grid method.
The paper is organized as follows. In section 2, we consider the stiffness matrix for the model problem and their most important properties. In section

3, we introduce and modify the preconditioner of Jensen/Korneev.. Section 4 shows that the modified preconditioner can be obtained by discretizing ellpitic problems with variable coefficients using finite differences or the h-version of the finite element method. Finally, some numerical experiments are given in section 5.
Throughout this paper, Ω will denote the unit rectangle $(-1,1)^2$, Ω_1 the rectangle $(0,1)^2$. The integer p is the polynomial degree, $\hat{L}_i$ the $i-$th integrated Legendre polynomial. The real number $\lambda_{max}(A)$ will denote the largest eigenvalue of a matrix A. The parameter c will describe a constant, which is independent of p or h.

2 Origin and properties of the stiffness matrix

2.1 Model problem

We try to find a numerical solution of the model problem

$$\begin{aligned} -\triangle u &= f, & (2.1)\\ u\,|_{\partial\Omega} &= 0 & (2.2) \end{aligned}$$

in the domain $\Omega = (-1,1)^2$. Problem (2.1,2.2) is the typical model problem for solving a linear system with the matrix A_{int}.

2.2 Discretization, shape functions

We solve (2.1,2.2) using the $p-$version of the FEM with only one element Ω. As finite element space, we choose

$$M = \{u \in H_0^1(\Omega), u\,|_\Omega \in P^p\},$$

where P_p is the space of all polynomials of degree $\leq p$ in both variables. The discretized problem is: find $u_p \in M$

$$\int_\Omega \nabla u_p \cdot \nabla v_p \mathrm{d}(x,y) = \int_\Omega f v_p \mathrm{d}(x,y)$$

for all $v_p \in M$. As basis in M, we choose the integrated Legendre polynomials, which we define below.
Let for $i = 0, 1, \ldots$

$$L_i(x) = \frac{1}{2^i i!} \frac{\mathrm{d}^i}{\mathrm{d}x^i} (x^2 - 1)^i$$

the i-th Legendre polynomial,

$$\tilde{L}_i(x) = \int_{-1}^{x} L_{i-1}(s) \, \mathrm{d}s$$

the i-th integrated Legendre polynomial and $\forall i \geq 2$

$$\hat{L}_i(x) = \sqrt{\frac{(2i-3)(2i-1)(2i+1)}{4}} \tilde{L}_i(x) = \gamma_i \tilde{L}_i(x)$$

the i-th integrated Legendre polynomial with scaling. By definition,

$$\begin{aligned} \hat{L}_0(x) &= \frac{1+x}{2}, \\ \hat{L}_1(x) &= \frac{1-x}{2}. \end{aligned}$$

The properties

$$\int_{-1}^{1} L_i(x) L_j(x) \, \mathrm{d}x = \delta_{ij} \frac{2}{2i+1}, \tag{2.3}$$

$$\hat{L}_i(x) = \sqrt{\frac{(2i+1)(2i-3)}{4(2i-1)}} (L_i(x) - L_{i-2}(x)), \tag{2.4}$$

$$\hat{L}_i(1) = 0, \tag{2.5}$$

$$\hat{L}_i(-1) = 0, \tag{2.6}$$

$$(i+1) L_{i+1}(x) + i L_{i-1}(x) = (2i+1) x L_i(x). \tag{2.7}$$

are true for $i \geq 2$, [10].
As basis in M, we choose

$$\hat{L}_{ij}(x, y) = \hat{L}_i(x) \hat{L}_j(y), \tag{2.8}$$

with $p \geq i, j \geq 2$. For satisfying (2.2), the polynomials $\hat{L}_0$ and $\hat{L}_1$ are not used, compare (2.5,2.6). The stiffness matrix K is determined by

$$K = (a_{ij,kl})_{i,j=2;k,l=2}^{p} = \int_\Omega \nabla \hat{L}_{ij}(x,y) \cdot \nabla \hat{L}_{kl}(x,y) \ \mathrm{d}(x,y).$$

With

$$\begin{aligned}\int_\Omega \nabla u(x,y) \cdot \nabla v(x,y) \ \mathrm{d}(x,y) &= \int_{-1}^{1}\int_{-1}^{1} (u_x(x,y)v_x(x,y) \\ &\quad + u_y(x,y)v_y(x,y)) \ \mathrm{d}x\mathrm{d}y\end{aligned}$$

and (2.8) we get

$$\begin{aligned} a_{ij,kl} &= \int_\Omega \Big(\frac{\mathrm{d}}{\mathrm{d}x}\hat{L}_i(x)\frac{\mathrm{d}}{\mathrm{d}x}\hat{L}_k(x)\hat{L}_j(y)\hat{L}_l(y) \\ &\quad + \frac{\mathrm{d}}{\mathrm{d}y}\hat{L}_j(y)\frac{\mathrm{d}}{\mathrm{d}y}\hat{L}_l(y)\hat{L}_i(x)\hat{L}_k(x)\Big) \ \mathrm{d}(x,y) \\ &= \int_{-1}^{1} \frac{\mathrm{d}}{\mathrm{d}x}\hat{L}_i(x)\frac{\mathrm{d}}{\mathrm{d}x}\hat{L}_k(x) \ \mathrm{d}x \int_{-1}^{1} \hat{L}_j(y)\hat{L}_l(y) \ \mathrm{d}y \\ &\quad + \int_{-1}^{1} \hat{L}_i(x)\hat{L}_k(x) \ \mathrm{d}x \int_{-1}^{1} \frac{\mathrm{d}}{\mathrm{d}y}\hat{L}_j(y)\frac{\mathrm{d}}{\mathrm{d}y}\hat{L}_l(y) \ \mathrm{d}y \\ &= d_{ik}f_{jl} + f_{ik}d_{jl}, \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} F = \ & (f_{ij})_{i,j=2}^{p} &= \int_{-1}^{1} \hat{L}_i(x)\hat{L}_j(x) \ \mathrm{d}x, \\ D = \ & (d_{ij})_{i,j=2}^{p} &= \int_{-1}^{1} \frac{\mathrm{d}}{\mathrm{d}x}\hat{L}_i(x)\frac{\mathrm{d}}{\mathrm{d}x}\hat{L}_j(x) \ \mathrm{d}x. \end{aligned}$$

Using (2.3,2.4), we determine the entries of the one-dimensional mass matrix, namely

$$F = \begin{pmatrix} 1 & 0 & -c_2 & 0 & \cdots \\ & 1 & 0 & -c_3 & \ddots \\ & \mathrm{SYM} & \ddots & \ddots & \ddots \\ & & & \vdots & 1 \end{pmatrix}$$

and the one-dimensional stiffness matrix, namely

$$D = \operatorname{diag}(d_i)_{i=2}^{p} = \begin{pmatrix} d_2 & 0 & \cdots \\ 0 & d_3 & \ddots \\ 0 & 0 & \ddots \end{pmatrix}$$

with the coefficients

$$\begin{aligned} c_i &= \sqrt{\frac{(2i-3)(2i+5)}{(2i-1)(2i+3)}}, \\ d_i &= \frac{(2i-3)(2i+1)}{2}, \end{aligned}$$

[9]. The stiffness matrix for the two-dimensional Laplace can be written using the matrices F and D by

$$K = F \otimes D + D \otimes F,$$

compare (2.9). Applying a permutation P of rows and columns, we get

$$PKP^{-1} = \begin{pmatrix} K_1 & & & \\ & K_2 & & \\ & & K_3 & \\ & & & K_4 \end{pmatrix}. \tag{2.10}$$

The first block contains the polynomials $\hat{L}_{2i,2j}$, the second $\hat{L}_{2i+1,2j}$, the third $\hat{L}_{2i,2j+1}$ and the fourth $\hat{L}_{2i+1,2j+1}$. If p is odd, all four blocks have the same size. We wish to find a fast solver for a system of linear equations with the matrix K or equivalently, K_i. This solver should perform the solution in not more than $\mathcal{O}(p^2 \log p)$ arithmetical operations.

3 Deriving a preconditioner for K

3.1 Preconditioner of Jensen/Korneev

Jensen/Korneev [9] defined the following preconditioner for K.

LEMMA 3.1 *Let*

$$D_1 = \mathrm{diag}(i^2)_{i=2}^{p},$$

$$T_1 = D_1^{-1} + \frac{1}{2}\begin{pmatrix} 2 & 0 & -1 & 0 & 0 & \cdots \\ & 2 & 0 & -1 & 0 & \cdots \\ \text{SYM} & & 2 & 0 & -1 & \ddots \\ & & & 2 & \ddots & \ddots \\ & & & \vdots & & \\ & & & -1 & 0 & 2 \end{pmatrix}$$

and

$$C_1 = D_1 \otimes T_1 + T_1 \otimes D_1,$$

then, the statements

$$c_1(D_1 v, v) \le (Dv, v) \le c_2(D_1 v, v), \tag{3.1}$$

$$c_3(T_1 v, v) \le (Fv, v) \le c_4(T_1 v, v), \tag{3.2}$$

$$c_1 c_3(C_1 v, v) \le (Kv, v) \le c_2 c_4(C_1 v, v). \tag{3.3}$$

are valid forall v.

Proof: (3.1) is trivial, (3.2) is proved in [9]. (3.3) follows immediately from (3.1,3.2). □

C_1 is simpler than K, but we still need now a fast solver for C_1.

3.2 Modification of the preconditioner

Now, we modify in several steps the preconditioner (3.1–3.3).

LEMMA 3.2 *Let*

$$D_2 = \mathrm{diag}(4\,[\frac{i}{2}]^2)_{i=2}^{p} = \mathrm{diag}(4, 4, 16, 16, 36, 36, \dots),$$

$$T_2 = T_1 - D_1^{-1} = \frac{1}{2}\begin{pmatrix} 2 & 0 & -1 & 0 & 0 & \cdots \\ & 2 & 0 & -1 & 0 & \cdots \\ \text{SYM} & & 2 & 0 & -1 & \ddots \\ & & & 2 & \ddots & \ddots \\ & & & \vdots & & \\ & & & -1 & 0 & 2 \end{pmatrix},$$

and

$$C_2 = D_2 \otimes T_2 + T_2 \otimes D_2,$$

Then, the inequalities

$$\begin{aligned}
(D_2 v, v) &\leq (D_1 v, v) \leq && \frac{9}{4}(D_2 v, v), && (3.4)\\
(T_2 v, v) &\leq (T_1 v, v) \leq && c_0(1+\log p)(T_2 v, v), && (3.5)\\
(C_2 v, v) &\leq (C_1 v, v) \leq && \frac{9}{4} c_0(1+\log p)(C_2 v, v) && (3.6)
\end{aligned}$$

are true forall v.

Proof: (3.4) and the left inequality of (3.5) are trivial, (3.6) is a corollary of (3.4,3.5). For the right inequality of (3.5), we introduce the matrix $\tilde{T} \in \mathbb{C}^{n,n}$

$$\tilde{T} = \frac{1}{2}\mathrm{tridiag}(-1, 2, -1).$$

The relation between n and p will be defined below. Furthermore, we need

$$D_0 = \mathrm{diag}\left(\frac{1}{i^2}\right)_{i=1}^{n}.$$

We estimate now

$$\lambda_{max}(\tilde{T}^{-1}(D_0 + \tilde{T})) = 1 + \lambda_{max}(D_0^{\frac{1}{2}} \tilde{T}^{-1} D_0^{\frac{1}{2}}).$$

The matrix

$$H = D_0^{\frac{1}{2}} \tilde{T}^{-1} D_0^{\frac{1}{2}}$$

can be written explicitly:

$$H = \frac{2}{n+1}\begin{pmatrix}
n & \frac{n-1}{2} & \frac{n-2}{3} & \frac{n-3}{4} & \cdots & \frac{2}{n-1} & \frac{1}{n}\\
\frac{n-1}{2} & \frac{n-1}{2} & \frac{n-2}{3} & \frac{n-3}{4} & \cdots & \frac{2}{n-1} & \frac{1}{n}\\
\frac{n-2}{3} & \frac{n-2}{3} & \frac{n-2}{3} & \frac{n-3}{4} & \cdots & \frac{2}{n-1} & \frac{1}{n}\\
\vdots & & & & \ddots & & \vdots\\
\frac{2}{n-1} & \frac{2}{n-1} & & \cdots & & \frac{2}{n-1} & \frac{1}{n}\\
\frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \cdots & & \frac{1}{n} & \frac{1}{n}
\end{pmatrix}.$$

Using the Perron–Frobenius theorem, [4], we get

$$\lambda_{max}(H) \leq c(1+\log n). \tag{3.7}$$

Furthermore, a permutation Q leads to

$$\begin{aligned} QTQ^{-1} &= \begin{pmatrix} \tilde{T} & \mathbf{0} \\ \mathbf{0} & \tilde{T} \end{pmatrix}, \\ QD_1^{-1}Q^{-1} &= \begin{pmatrix} \frac{1}{4}D_0 & \mathbf{0} \\ \mathbf{0} & \tilde{D} \end{pmatrix} \end{aligned}$$

with

$$\tilde{D} = \operatorname{diag}\left(\frac{1}{(2i+1)^2}\right)_{i=1}^{[\frac{p-1}{2}]}.$$

Therefore, we get for each block a similar estimate as (3.7). For the first block, we have $n = [\frac{p}{2}]$, for the second $n = [\frac{p-1}{2}]$. Hence it follows (3.5).□

COROLLARY 3.3 *For the matrix*

$$C_5 = D_2 \otimes (D_2^{-1} + T_2) + (T_2 + D_2^{-1}) \otimes D_2, \tag{3.8}$$

the estimate

$$\frac{4}{9}(C_5 v, v) \leq (C_1 v, v) \leq \frac{9}{4}(C_5 v, v)\ \forall v$$

is valid.

Proof: The proof follows from (3.4) and the inequality

$$c_a((A+B)v,v) \leq ((A+\tilde{B})v,v) \leq c_b((A+B)v,v)$$

for symmetric and positive definite matrices A, B and $\tilde{B}$ satisfying

$$c_a(Bv,v) \leq (\tilde{B}v,v) \leq c_b(Bv,v)$$

forall v.□

In the following, we assume p is odd. We introduce $n = [\frac{p-1}{2}]+1$. Applying a basis-transformation using the permutation P, (2.10), C_2 and C_5 are block diagonal matrices of 4 identical blocks C_3 and C_6, where

$$\begin{aligned} C_3 &= D_3 \otimes T_3 + T_3 \otimes D_3, & (3.9) \\ C_6 &= D_3 \otimes (T_3 + D_3^{-1}) + (T_3 + D_3^{-1}) \otimes D_3 & (3.10) \end{aligned}$$

with

$$
\begin{aligned}
D_3 &= \operatorname{diag}(4i^2)_{i=1}^{n-1}, \\
T_3 &= \frac{1}{2}\operatorname{tridiag}(-1, 2, -1).
\end{aligned}
$$

Furthermore, we need the matrices

$$
D_4 = 4 \operatorname{diag}\left(i^2 + \frac{1}{6}\right)_{i=1}^{n-1}
$$

and

$$
C_4 = D_4 \otimes T_3 + T_3 \otimes D_4. \tag{3.11}
$$

Applying Lemmata 3.1,3.2 and Corollary 3.3, we get

THEOREM 3.4 . *Let $K_i, i = 1, \ldots, 4$ are the 4 blocks of K. The following statements are valid $\forall v$ and $i = 1, \ldots, 4$:*

$$
\begin{aligned}
c_7(C_3 v, v) &\le (K_i v, v) \le c_8(1 + \log p)(C_3 v, v), \\
c_{11}(C_6 v, v) &\le (K_i v, v) \le c_{12}(C_6 v, v), \\
c_9(C_4 v, v) &\le (K_i v, v) \le c_{10}(1 + \log p)(C_4 v, v).
\end{aligned}
$$

4 Similar systems of linear equations for other methods of discretization

4.1 Finite differences

The matrix C_3 is the system matrix for a discretization of

$$
\begin{aligned}
-y^2 \frac{\partial^2 u}{\partial x^2} - x^2 \frac{\partial^2 u}{\partial y^2} &= g, \\
u \,|_{\partial\Omega_1} &= 0
\end{aligned} \tag{4.1}
$$

in $\Omega_1 = (0, 1)^2$ using finite differences and the grid of Figure 1.

Indeed, we denote the approximation in $\frac{1}{n}(i, j)$ by $u_{i,j}$. We approximate the second derivatives by the usual second order central difference quotient:

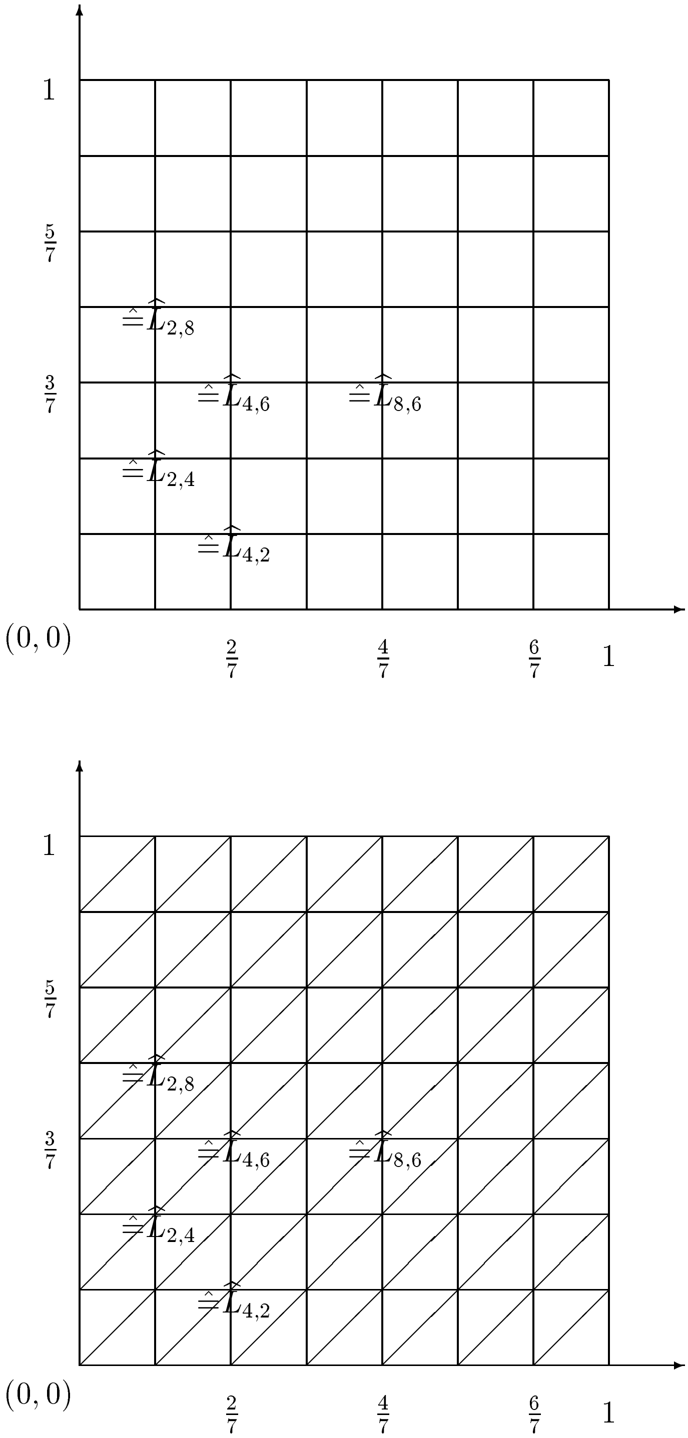

Figure 1: Mesh for h-Version (below), grid (above).

$$
\begin{aligned}
y^2\frac{\partial^2 u}{\partial x^2}\left(\frac{i}{n},\frac{j}{n}\right) &\approx j^2(u_{i+1,j}+u_{i-1,j}-2u_{i,j}),\\
x^2\frac{\partial^2 u}{\partial y^2}\left(\frac{i}{n},\frac{j}{n}\right) &\approx i^2(u_{i,j+1}+u_{i,j-1}-2u_{i,j}).
\end{aligned}
$$

If we insert the boundary condition and sort the unknowns in the order $u_{1,1}, u_{1,2}, \ldots, u_{1,n-1}, u_{2,1}, \ldots, u_{n-1,n-1}$, we get the system matrix $\frac{1}{2}C_3$ (3.9).

REMARK 4.1 *The discretization of*

$$
\begin{aligned}
-y^2\frac{\partial^2 u}{\partial x^2} - x^2\frac{\partial^2 u}{\partial y^2} + \frac{1}{2}\left(\frac{y^2}{x^2}+\frac{x^2}{y^2}\right)u &= g,\\
u\mid_{\partial\Omega_1} &= 0 \qquad (4.2)
\end{aligned}
$$

as above leads to the system matrix $\frac{1}{2}C_6$ *(3.10)*

4.2 h-version of the FEM

We solve in $\Omega_1 = (0,1)^2$

$$
\begin{aligned}
\int_{\Omega_1}\left(y^2\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + x^2\frac{\partial u}{\partial y}\frac{\partial v}{\partial y}\right)\,\mathrm{d}(x,y) &= \int_{\Omega_1} gv\,\mathrm{d}(x,y),\\
u\mid_{\partial\Omega_1} &= 0 \qquad (4.3)
\end{aligned}
$$

using $h-$version of the FEM, namely piecewise linear shape functions and and the triangulation of Figure 1. If we calculate for (4.3) the matrix entry $a_{j,j}^{i,i+1}$ corresponding to the edge with vertices $\frac{1}{n}(i,j)$ and $\frac{1}{n}(i+1,j)$, we have,

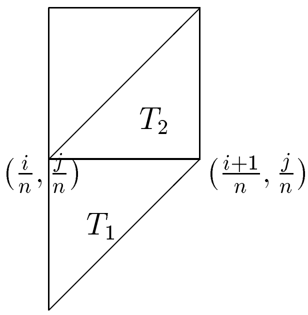

Figure 2: Sketch for calculation the matrix entry between two adjacent nodes.

compare Figure 2,

$$
\begin{aligned}
a_{j,j}^{i,i+1} &= \int_{T_1} \begin{pmatrix} -n \\ n \end{pmatrix} \begin{pmatrix} y^2 & 0 \\ 0 & x^2 \end{pmatrix} \begin{pmatrix} n \\ 0 \end{pmatrix} \,\mathrm{d}(x,y) \\
&\quad + \int_{T_2} \begin{pmatrix} -n \\ 0 \end{pmatrix} \begin{pmatrix} y^2 & 0 \\ 0 & x^2 \end{pmatrix} \begin{pmatrix} n \\ -n \end{pmatrix} \,\mathrm{d}(x,y) \\
&= -n^2 \int_{T_1 \cup T_2} y^2 \,\mathrm{d}(x,y) \\
&= -n^2 \int_{\frac{j-1}{n}}^{\frac{j}{n}} \int_{\frac{i}{n}}^{y+\frac{i-j+1}{n}} y^2 \,\mathrm{d}x\mathrm{d}y - n^2 \int_{\frac{j}{n}}^{\frac{j+1}{n}} \int_{y+\frac{i-j}{n}}^{\frac{i+1}{n}} y^2 \,\mathrm{d}x\mathrm{d}y \\
&= -\frac{1}{n^2}\left(\frac{j^2}{2} - \frac{j}{3} + \frac{1}{12}\right) - \frac{1}{n^2}\left(\frac{j^2}{2} + \frac{j}{3} + \frac{1}{12}\right) \\
&= -\frac{1}{n^2}\left(\frac{1}{6} + j^2\right), \qquad (4.4)
\end{aligned}
$$

where $n > i, j$ and $j > 0$, but $i \geq 0$. By symmetry, we have ($i > 0$, $j \geq 0$)

$$a_{j,j+1}^{i,i} = -\frac{1}{n^2}\left(\frac{1}{6} + i^2\right)$$

and

$$a_{j,j}^{i,i} = -(a_{j,j+1}^{i,i} + a_{j-1,j}^{i,i} + a_{j,j}^{i,i+1} + a_{j,j}^{i,i-1}).$$

All other matrix entries are zero. Inserting the boundary condition, we arrive at a system of linear equations with the system matrix $\frac{1}{2n^2}C_4$ (3.11).

5 Numerical results

5.1 V-cycle of multi-grid

We start with a brief discussion of the V-cycle of the multi-grid algorithm for solving

$$K_l u_l = f_l.$$

For u_l^0 and f_l, the multi-grid algorithm for computing $u_l^1 = M_l(u_l^0, f_l)$ is defined recursively as follows:

- Pre-smoothing:

$$u_l^{0,1} = S_{pre}^{\nu}(u_l^0, f_l). \tag{5.1}$$

- Calculation of the defect:

$$d_l = f_l - K_l u_l^{0,1}. \tag{5.2}$$

- Restriction of the defect:

$$f_{l-1} = I_l^{l-1} d_l. \tag{5.3}$$

- Coarse grid correction: If $l = 1$ solve $K_1 w_1 = f_1$ using a direct method, if $l > 1$

$$w_{l-1} = M_{l-1}(0, f_{l-1}).$$

- Interpolation of the correction:

$$w_l = I_{l-1}^{l} w_{l-1}. \tag{5.4}$$

- Adding the correction:

$$u_l^{0,2} = u_l^{0,1} + w_l.$$

- Post-smoothing:

$$u_l^1 = S_{post}^{\nu}(u_l^{0,2}, f_l). \tag{5.5}$$

5.2 Choice of a good preconditioner

For problems like the Poisson equation, we have good preconditioners: e.g. the BPX-preconditioner [2] or, in 2D, the HB-preconditioner [11]. But, the differential operator in (4.3), or (4.2) is not spectrally equivalent to Laplace. It is an elliptic, but not uniformly elliptic differential operator. Our idea is now taking a preconditioner which is a fast solver for anisotropic problems like

$$-\frac{\partial^2 u}{\partial x^2} - \epsilon \frac{\partial^2 u}{\partial y^2} = f.$$

One solver with rate of convergence independent on the choice of ϵ is the multi-grid algorithm with a line Gauss-Seidel (GS) as smoother, [6] pp.502–533. For problems (4.3) and (4.2), we have anisotropies in both directions. Hence, we take the x-line and the y-line GS as smoother [5]. To obtain a symmetric preconditioner, we do a special smoothing procedure. One pre-smoothing step consist of 1 iteration of the forwards x-line GS and 1 iteration of the forwards y-line GS, one post-smoothing step of 1 iteration of the backwards y- and 1 iteration of the backwards x-line GS.

5.3 Number of iterations of the PCG-method for the a multi-grid preconditioner without mass-matrix

For the following numerical results, we choose a relative accuracy of 10^{-7}. The preconditioner consists of 1 step of the V-cycle of the multi-grid method for C_4 (3.11) using the smoother discussed above for each block. The interpolation and restriction operators are the usual finite element interpolation and restriction operators. All calculations are done on a Pentium-III 500 Mhz. We discuss the 5 cases

1. $f = \delta$,
2. $f = \delta_{(\frac{1}{2},\frac{1}{2})}$,
3. $f = 1$,
4. $f = xy$,
5. $f = 1 + x + y + xy$,

p	$f=\delta$	$f=\delta_{(\frac{1}{2},\frac{1}{2})}$	$f=1+x$ $+y+xy$	$f=xy$	$f=1$
7	9	12	12	6	8
15	12	14	14	7	11
31	15	16	16	8	14
63	16	17	17	9	16
127	17	17	18	10	17
255	18	18	18	10	17
511	18	18	18	11	18
1023	18	18	19	11	18

Table 1: Number of iterations of the PCG-method for K using a multi-grid preconditioner of C_4 for several right hand sides.

where $\delta_{(x,y)}$ is the Delta-Distribution centered in the point (x, y).

REMARK 5.1 1. *Because of the relation*

$$\hat{L}_{2i+1}(0) = 0$$

for all $i \in \mathbb{N}$*, the right hand side of the linear system of the second, third and last block for* $f = \delta$ *is identically zero.*
Using

$$\int_{-1}^{1} \hat{L}_i(t)\ \mathrm{d}t = 0$$

for all $i \geq 3$*, we see that for* $f = 1$ *only the right hand side of the first block has nonzero entries. Hence, only the first block of the stiffness matrix is relevant in both cases.*

2. *From*

$$\int_{-1}^{1} t\hat{L}_i(t)\ \mathrm{d}t = 0$$

for $i = 2$ *and* $i \geq 4$ *follows, that only the last block of the right hand side has nonzero entries for* $f = xy$*.*

p	$f=\delta$	$f=\delta_{(\frac{1}{2},\frac{1}{2})}$	$f=1+x$ $+y+xy$	$f=xy$	$f=1$
7	8	13	12	7	7
15	12	13	12	8	9
31	12	13	13	8	9
63	12	13	13	8	9
127	12	13	13	8	9
255	12	13	13	8	9
511	12	13	13	8	9
1023	12	13	13	8	9

Table 2: Number of iterations of the PCG-method for K using a multi-grid preconditioner of C_6 for several right hand sides.

The results are displayed in Table 1. We see only a slowly increasing of the number of iterations. The speed is lower than $\mathcal{O}(\log p)$. Moreover, we see less iterations if we have only the fourth block, compare case 4.

REMARK 5.2 *If we use as preconditioner 1 multi-grid cycle resulting from* C_3 *(3.9) and bilinear interpolation, we get nearly the same results.*

5.4 Number of iterations for the multi-grid preconditioner with mass-matrix

The discretization of (4.2) leads to a spectral equivalent matrix to K_i. Table 2 displays the number of iterations for the stiffness matrix K using a multi-grid preconditioner for C_6 (3.10). We take as interpolation bilinear interpolation. The choice of the remaining parameter is as before.

For this preconditioner, the number of iterations does not depend on p. As before, the fourth block is better conditioned than the first. The number of iterations is lower for all right hand sides as for the multi-grid solver resulting from C_4.

5.5 Comparison of several smoothers

The method discussed above is assymptotically optimal. But, for which p does the line smoother outperform the usual GS smoother? We compare our

p	Gauss-Seidel		line-GS	
	iterat.	time [sec]	iterat.	time [sec]
7	14	0.00586	15	0.02051
15	16	0.0283	17	0.0864
31	19	0.145	20	0.398
63	24	0.787	21	1.74
127	31	4.28	21	7.40
255	40	23.09	22	31.52
511	52	124.51	23	136.06
1023	65	631.86	23	554.58

Table 3: Comparison between two smoothers.

results for $f = \delta$ with a multi-grid preconditioner using usual GS as smoother. The relative accuracy is 10^{-9}. We take one forwards GS as pre- and one backwards GS as post-smoother, the rest as in chapter 5.3. Especially, the multi-grid preconditioner results from the matrix C_4. Then, we get the results of Table 3. We see that, the method using Gauss-Seidel as smoother is not assymptotically optimal, but faster for $p < 1023$.

5.6 Semi-coarsening

Using uniform refinement, we need two smoothing steps, 1 for applying the x-line smoother and 1 for the y-line smoother. Can we obtain a faster method using only 1 smoothing step? If we apply only one line smoother, we have to change the refinement-strategy. Instead of uniform refinement, we choose a semi-coarsening strategy. We refine only in y-direction, see Figure 3. We only apply the x-line GS as smoother. The unknowns in the circles are put together for the x-line GS.

The coarsest level is level 1 with 1 unknown in the y-direction. The coars-grid system is tridiagonal, and can be solved using Cholesky/Crowd-decompostion in $\mathcal{O}(p)$ arithmetical operations. We choose linear interpolation with respect to the y-direction. The number of iterations of the PCG-method for K using the V-cycle of the multi-grid algorithm discussed above is displayed in Table 4. We take one pre- and one post-smoothing step. The

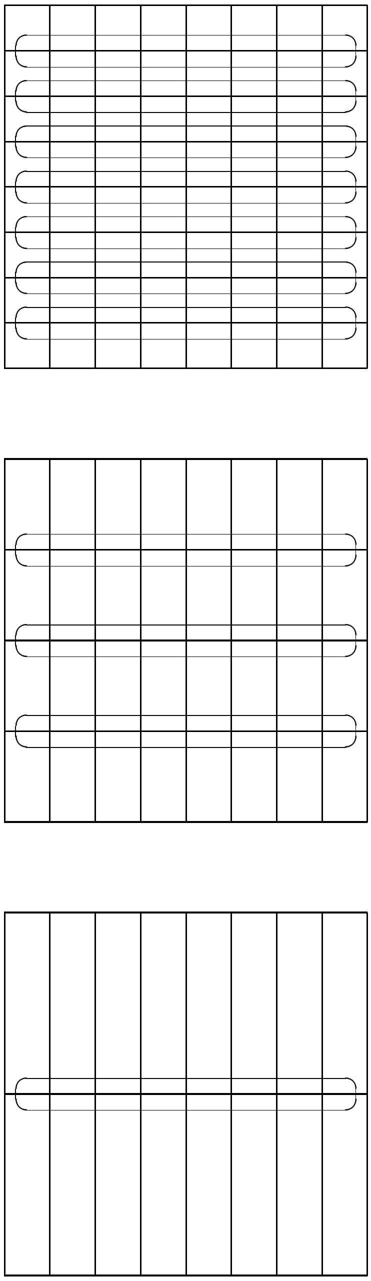

Figure 3: Semi-coarsening and line-smoother in 1 direction.

	$f=\delta_{(\frac{1}{2},\frac{1}{2})}$		$f=\delta_{(\frac{1}{2},-\frac{3}{10})}$	
p	C_3	C_6	C_3	C_6
7	13	13	13	13
15	14	13	14	13
31	16	13	16	13
63	16	13	17	13
127	17	13	17	13
255	17	13	18	13
511	18	13	18	13
1023	18	13	18	13

Table 4: Number of iterations of the PCG-method for K using a multi-grid preconditioner of C_3 and C_6, and semi-coarsening.

relative accuracy is 10^{-7}. We consider two different cases, namely

$$f=\delta_{(\frac{1}{2},\frac{1}{2})}$$

and

$$f=\delta_{(\frac{1}{2},-\frac{3}{10})}.$$

The numbers of iterations are nearly the same as for uniform refinement and do not depend on unsymmetric right hand sides.

5.7 Arithmetical costs

We have seen in the previous chapter that we obtain nearly the same number of iterations if we use uniform refinement or a semi-coarsening strategy. In this chapter, we will estimate the number of arithmetical operations using uniform refinement or semi-coarsening.
For a given $n \times m$ grid, we have

$$W_{int}=W_{rest}=nm \tag{5.6}$$

arithmetical operations for (5.4) and (5.3). A matrix-vector multiplication for a 5-point stencil of a $n \times m$ grid costs

$$W_{mat}=5(m-2)(n-2)+8(m-2)+8(n-2)+12 \tag{5.7}$$

flops. We have n blocks of size m for the line-GS. For one block, we have to solve a tridiagonal linear system of size $m \times m$. Hence, we have

$$W_{right} = m$$

operations for generating the right hand side,

$$W_{chol} = 2m - 1$$

flops for computing the Crowd-decomposition and

$$W_{back} + W_{for} = 3m - 2$$

flops for forwards and backwards elimination. Hence, we need

$$W^{n\times m}_{line-GS} = W_{chol} + W_{back} + W_{for} + W_{right} = n(6m - 3) \tag{5.8}$$

flops. Using (5.6-5.8), we have on level j

$$W_j = 2W^{n\times m}_{line-GS} + W_{int} + W_{rest} + W_{mat} = 19nm - 2m - 8n \tag{5.9}$$

flops if we apply only the x-line GS. For applying x- and y-line GS, we have

$$\begin{aligned} W_j &= 2W^{n\times m}_{line-GS} + 2W^{m\times n}_{line-GS} + W_{int} + W_{rest} + W_{mat} \\ &= 31nm - 8m - 8n \end{aligned} \tag{5.10}$$

flops. With same arguments, we see that we need

$$W_j = 17nm - 6m - 6n \tag{5.11}$$

flops for applying GS. For uniform refinement, we have

$$n(j) = m(j) = 2^j - 1.$$

For semi-coarsening,

$$\begin{aligned} n(j) &= 2^j - 1, \\ m(j) &= 2^l - 1. \end{aligned}$$

Hence, we need

$$W^l = \sum_{j=2}^{l} W_j + W_{coars}$$

flops for 1 iteration of the V-cycle. Thus, we have

$$W_{GS}^{l} = \frac{17}{3}4^{l+1} - 46\ 2^{l+1} + 29l + \frac{193}{3} + W_{coars}$$

arithmetical operations using uniform refinement and GS as smoother. For applying x-line-GS and y-line GS and uniform refinement, we need

$$W_{uni}^{l} \quad = \quad \frac{31}{3}4^{l+1} - 156\ 2^{l} + 47l + \frac{299}{3} + W_{coars}$$

flops. We need

$$W_{semi}^{l} \quad = \quad \frac{19}{2}4^{l+1} - (109 + 21l)\ 2^{l} + 29l + 79 + W_{coars}$$

flops using semi-coarsening and x-line GS.

Hence, for $l \geq 2$

$$W_{uni}^{l} > W_{semi}^{l}$$

and

$$\lim_{l\to\infty} \frac{W_{uni}^{l}}{W_{semi}^{l}} = \frac{62}{57}.$$

This result can be verified in numerical experiments, compare Table 5. We display the time to reduce the error for solving

$$Ku = b$$

up to a factor 10^{-7}. We take the multi-grid preconditioner for C_6. We choose

$$f = \delta_{(\frac{1}{2},\frac{1}{2})}.$$

In each case, we need 13 iterations of the pcg-method to reduce the error up to a factor 10^{-7}, compare Tables 2 and 4. The semi-coarsening strategy is about 10 per cent faster than uniform refinement.

p	Uniform refinement	Semi-coarsening
7	0.0195	0.0078
15	0.0703	0.0469
31	0.2852	0.2302
63	1.1953	1.0430
127	4.9888	4.4960
255	20.641	18.914
511	87.055	78.965
1023	347.789	324.234

Table 5: Time [sec] to reduce the error up to 10^{-7} for several coarsening strategies

6 Additional remarks

For discretizing

$$\begin{aligned} Au := -\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial^2 u}{\partial y^2} &= f, \\ u\,|_{\Omega_1} &= 0 \end{aligned} \tag{6.1}$$

in the unit square Ω_1 using linear or bilinear elements, Bramble/Zhang [3] proved optimal convergence of the multi-grid algorithm. They applied line-GS or line-Jacobi as smoother. Additional assumptions on b are

$$0 < b(x) < c_b,$$

but not

$$b(x) \geq c_a > 0.$$

They proved an approximation property of the type

$$(b(I-P_h)v, (I-P_h)v) \leq C_1 h^2 \parallel (I-P_h)v \parallel_A^2 \tag{6.2}$$

and the smoothing property

$$\frac{1}{2}(Av, v) \leq (J_{x,h}^{-1}v, v) \leq C_2\left((Av, v) + \frac{1}{h^2}(bv, v)\right). \tag{6.3}$$

P_h denotes in (6.2) the Galerkin projection, $J_{x,h}$ in (6.3) the $x-$line smoother. But, we have more difficulties: we have anisotropies in both directions. Hence, we have to apply an alternating line-GS J_h as smoother, where

$$J_h = J_{x,h} + J_{y,h} - J_{x,h} A J_{y,h}.$$

References

[1] S. Beuchler. Lösungsmethoden bei der p-version der fem. Diplomarbeit, TU Chemnitz, December 1999.

[2] J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55(191):1–22, 1991.

[3] J. Bramble and X. Zhang. Uniform convergence of the multigrid v-cycle for an anisotropic problem. *Math. Comp.*, to appear.

[4] F.R. Gantmacher. *Matrizenrechnung II.* Deutscher Verlag der Wissenschaften. Berlin, 1971.

[5] W. Hackbusch. *Multigrid Methods and Applications.* Springer-Verlag. Heidelberg, 1985.

[6] W. Hackbusch and U. Trottenberg. *Multigrid Methods.* Number 960 in Lecture Notes in Mathematics. Springer-Verlag. Berlin-Heidelberg-New York, 1981.

[7] S.A. Ivanov and V.G. Korneev. On the preconditioning in the domain decomposition technique for the p-version finite element method. part i. Technical Report SPC 95-35, Technische Universität Chemnitz-Zwickau, December 1995.

[8] S.A. Ivanov and V.G. Korneev. On the preconditioning in the domain decomposition technique for the p-version finite element method. part ii. Technical Report SPC 95-36, Technische Universität Chemnitz-Zwickau, December 1995.

[9] S. Jensen and V.G. Korneev. On domain decomposition preconditioning in the hierarchical $p-$version of the finite element method. *Comput. Methods. Appl. Mech. Eng.*, 150(1–4):215–238, 1997.

[10] F.G. Tricomi. *Vorlesungen über Orthogonalreihen.* Springer. Berlin-Göttingen-Heidelberg, 1955.

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