MULTILEVEL ITERATIVE METHODS AND DEFLATION

Radim Blaheta
Institute of Geonics, Academy of Sciences of the Czech Republic
Studentská 1768, 708 00 Ostrava - Poruba, the Czech Republic
e-mail: blaheta@ugn.cas.cz

Key words: multilevel preconditioners, deflation, nonsymmetric multiplicative preconditioner, generalized CG method, coarse grid, aggregation

Abstract. In this paper, we show that the deflation method can be viewed as a possible implementation of the CG method with multilevel preconditioner. Further, we demonstrate efficiency and robustness of different implementations of multilevel preconditioners with different "coarse grid" spaces by solving a simple model problem.

1 INTRODUCTION

This paper concerns the solution of linear systems

\[ Au = b, \quad u, b \in \mathbb{R}^n, \quad A \in \mathbb{R}^{n \times n}, \]  

which arise e.g. from the finite element analysis of stationary potential flow, porous media flow or elasticity problems in a finite element space \( V_h \) isomorphic with \( \mathbb{R}^n \). In this paper, we shall assume that \( A \) is a symmetric positive definite (SPD) matrix. The system (1) is typically large and ill-conditioned, so that an iterative solution method with suitable preconditioner will be a proper choice of solver.

The construction of efficient preconditioners frequently requires separate handling of local and global phenomena, like smoothing and coarse grid correction in the multigrid methods and two-level Schwarz preconditioners.

There are different possible combinations of a local preconditioner \( M \) and a global preconditioner \( B \) involving additive, multiplicative and symmetric multiplicative combination techniques. In the context of Schwarz preconditioners, we speak also about hybrid preconditioners when a local additive one-level Schwarz preconditioner \( M \) is combined multiplicatively with the coarse grid correction.

The described preconditioners are introduced in Sections 2 and 3. We shall show that all these preconditioners including their inexact variants arising from an inexact solution of the subproblems can be implemented in either preconditioned conjugate gradient (CG) or generalized preconditioned conjugate gradient (GPCG) methods. More detail can be found in [4, 6].
A special implementation of the symmetric multiplicative preconditioner with exact solution of the coarse grid subproblem and projection of the solution to the coarse grid space as an initial guess was introduced in [10], see also [15]. This implementation, described in Section 4, allows to save substantially the computational work and provide us a bridge to the idea of deflation.

The deflation idea, cf. [13, 8, 2, 9] and others, is described in Section 5, where we also show equivalence of the algorithm from the previous section with an implementation of the deflation method.

In the last section, we present numerical results concerning application of the described preconditioners and different choices of the coarse grid space. We compare both efficiency of the described methods and their robustness with respect to the inexact solution of the coarse grid problem. This comparison complements the results from recent papers [11, 12]. More details will be provided in a future paper.

The notation $\langle u, v \rangle = u^T v$ for $u, v, \in \mathbb{R}^n$ is used throughout the paper.

2 MULTILEVEL PRECONDITIONERS

Frequently, we are interested in a combination of preconditioners with different features, e.g. a local preconditioner $M$ and a global preconditioner $B$ with properties specified in Lemma 1. This combination can be done additively or multiplicatively, see the following definitions:

**additive preconditioner** $G_A : r \mapsto g$

(i) $g = Br + Mr$,

**multiplicative preconditioner** $G_{M1} : r \mapsto g$

(i) $g_1 = Br$,

(ii) $g = g_1 + M(r - Ag_1)$,

**symmetric multiplicative preconditioner** $G_{SM} : r \mapsto g$

(i) $g_1 = Br$,

(ii) $g_2 = g_1 + M(r - Ag_1)$,

(iii) $g = g_2 + B(r - Ag_2)$.

These preconditioners can be written in a matrix form as:

$$G_A = B + M,$$

$$G_{M1} = B + M(I - AB),$$

$$G_{SM} = (I - BA)M(I - AB) + 2B - BAB.$$  

It is easy to show the following properties of the introduced preconditioners.
Lemma 1 If $M$ is symmetric positive definite and $B$ is symmetric positive semidefinite, then $G_A$ and $G_{SM}$ are symmetric positive definite. On the other hand, $G_{M1}$ is nonsymmetric.

Proof It is obvious that $G_A$ is SPD, $G_{M1}$ is nonsymmetric and $G_{SM}$ is symmetric positive semidefinite. If $\langle G_{SM}v, v \rangle = 0$ then simultaneously
\begin{align*}
\langle M(I - AB)v, (I - AB)v \rangle &= 0 \quad (5) \\
\langle (2B - BAB)v, v \rangle &= 0. \quad (6)
\end{align*}
The first identity implies $(I - AB)v = 0$ and consequently
\begin{align*}
(B - BAB)v = 0, \quad \langle (I - AB)v, A^{-1}v \rangle &= 0, \quad \text{i.e.} \quad \langle Bv, v \rangle = \langle A^{-1}v, v \rangle.
\end{align*}
Thus, if (5) and (6) hold, then
\begin{align*}
\langle (2B - BAB)v, v \rangle &= \langle Bv, v \rangle = \langle A^{-1}v, v \rangle
\end{align*}
which implies $v = 0$. \qed

Note that the symmetry and positive definiteness are arguments for using $G_A$ and $G_{SM}$ preconditioners. On the other hand, there is different computational work involved in the preconditioners, for example zero, one and two matrix-vector multiplications with matrix $A$ are involved in $G_A$, $G_{M1}$ and $G_{SM}$, respectively. From the point of view of a compromise between the amount of labour and efficiency, preconditioner with one matrix-vector multiplication with $A$ could be most favourable.

Frequently, the global preconditioner is constructed with the aid of a coarse grid finite element space $V_H \subset V_h$ and prolongation restriction operators. This construction can be also generalized and rewritten into an algebraic form as follows. Let
\begin{align*}
V_0 &= \text{span} \{z_1, \ldots, z_r\}, \quad z_i \in \mathbb{R}^n \\
i.e.
V_0 &= \{v = Zw, w \in \mathbb{R}^r\}, \quad (7)
\end{align*}
where $Z \in \mathbb{R}^{n \times r}$ is a full rank matrix with columns $z_1, \ldots, z_r, r \ll n$. Then
\begin{align*}
B &= ZA_0^{-1}Z^T, \quad A_0 = Z^TAZ. \quad (8)
\end{align*}
Note that in this case,
\begin{align*}
BAB &= ZA_0^{-1}Z^TA_0^{-1}Z^T = ZA_0^{-1}Z^T = B, \quad (9)
\end{align*}
therefore
\begin{align*}
G_{SM} = (I - P_0)M(I - P_0^T) + B \quad (10)
\end{align*}
where $P_0 = BA$ is an $A-$orthogonal projection operator $V \rightarrow V_0$.

In the sequel, we will assume that $B$ possess the properties (8)–(10).
3 IMPLEMENTATION OF THE PRECONDITIONERS

Symmetric positive definite preconditioners $G_{SM}$ and $G_A$ can be implemented into the standard PCG algorithm cf. [1].

Further, we also consider the case of inaccurate implementation of $B = Z A_0^{-1} Z^T$, which means that we use $\tilde{B}$ instead of $B$, where

$$\tilde{B} w = v$$

is computed in the following steps

1. $w_1 = Z^T w$,
2. $w_2$ is obtained by solving $A_0 w_2 = w_1$ by an inner CG method with the accuracy $\|A_0 w_2 - w_1\| \leq \varepsilon_0 \|w_1\|$,
3. $v = Z w_2$.

given $u^0$

$$r^0 = b - A u^0$$
$$g^0 = G r^0$$
$$v^0 = g^0$$

for $i = 0, 1, \ldots$ until $\|r^i\| \leq \varepsilon \|b\|$

$$u^{i+1} = u^i + \alpha_i v^i$$
$$r^{i+1} = r^i - \alpha_i A v^i$$
$$g^{i+1} = G r^{i+1}$$
$$v^{i+1} = g^{i+1} + \sum_{k=1}^{\min(i,k)} \beta^{(k)}_{i+1} v^{i+1-k}$$

end

Figure 1: GPCG$[m]$ algorithm with a preconditioner $G$.

In this case, it is possible to apply the CG method without a guarantee of convergence or a generalized CG methods [1], [14], [4], which guarantees the convergence. In this paper, we shall use the GPCG$[m]$ algorithm, see Fig. 1, where $\alpha_i$ and $\beta_i^{(k)}$ are defined as follows

$$\alpha_i = \langle r^i, g^i \rangle / \langle Av^i, v^i \rangle$$
$$\beta_i^{(k)} = (\langle g^{i+1}, r^{i+2-k} \rangle - \langle g^{i+1}, r^{i+1-k} \rangle) / (\langle g^{i+1-k}, r^{i+1-k} \rangle).$$

The GPCG$[m]$ method is described and investigated e.g. in [4]. In this paper, we shall use the simplest case $m=1$. For an SPD preconditioner, GPCG$[m]$ is equivalent to the CG method.
When using the GPCG method, the local preconditioner $M$ can also involve inexactly solved subproblems, but this issue will not be discussed here.

On the other hand, we shall utilize the fact that the GPCG method allows to apply also nonsymmetric preconditioner $G_{M1}$ and its inexact version. It can be useful to mention the spectral relation between the spectrum $\sigma(G_{SM}A)$ and $\sigma(G_{M1}A)$.

**Lemma 2** $\sigma(G_{SM}A) = \sigma(G_{M1}A)$

**Proof**

\[
\begin{align*}
I - G_{M1}A &= I - P_0 - MA(I - P_0) = (I - MA)(I - P_0) \\
I - G_{SM}A &= I - P_0 - (I - P_0)MA(I - P_0) \\
&= (I - P_0)(I - MA)(I - P_0)
\end{align*}
\]

Thus (see Lemma 3), $\sigma(I - G_{SM}A) = \sigma(I - G_{M1}A)$ and $\sigma(G_{SM}A) = \sigma(G_{M1}A)$.

**Lemma 3** For any two $n \times n$ matrices $X, Y$, it holds $\sigma(XY) = \sigma(YX)$.

**Proof** Let $\lambda \in \sigma(XY)$, then there is $u \neq 0$, $XYu = \lambda u$. Therefore, $YXYu = \lambda Yu$. Now, if $Yu \neq 0$ then $Yu$ is an eigenvector of $YX$ and $\lambda \in \sigma(YX)$. If $Yu = 0$ then $0 = XYu = \lambda u$, which implies $\lambda = 0$. But then also $0 \in \sigma(YX)$ with the corresponding eigenvector $v = X^{-1}u$ for $X$ regular or $v \neq 0, Xv = 0$ for $X$ singular. In the same way, we can prove the opposite inclusion $\sigma(YX) \subset \sigma(XY)$.

**4 A NONSYMMETRIC IMPLEMENTATION OF $G_{SM}$**

In the case of exact computation of subproblems, the computational work of $G_{SM}$ can be substantially reduced if we start from a special initial guess

\[ u^0 = Bb + v, \quad \text{where } v \in \text{range}(I - P_0). \]

The idea of reduction, introduced in [10], uses the following properties.

**Lemma 4** Consider the PCG method with the preconditioner $G_{SM}$. If $u^0 = Bb + (I - P_0)w$, where $w \in \mathbb{R}^n$, then for $i = 0, 1, \ldots$ we get

\[ Bru^i = 0, \quad P_0g^i = 0, \quad P_0v^i = 0. \]

**Proof** Let $i = 0$, then

\[
\begin{align*}
Br^0 &= B(b - ABB - A(I - P_0)w) = 0, \\
g^0 &= G_{SM}r^0 = (I - BA)M(I - AB)r^0, \quad \text{i.e. } P_0g^0 = 0, \\
v^0 &= g^0 \Rightarrow P_0v^0 = 0.
\end{align*}
\]
\[ u^0 = Bb \]
\[ r^0 = b - Au^0 \]
\[ g^0 = G_M r^0 \]
\[ v^0 = g^0 \]

for \( i = 0, 1, \ldots \) until \( \| r^i \| \leq \varepsilon \| b \| \)
\[ u^{i+1} = u^i + \alpha_i v^i \]
\[ r^{i+1} = r^i - \alpha_i A v^i \]
\[ g^{i+1} = G_M r^{i+1} \]
\[ v^{i+1} = g^{i+1} + \beta_i v^i \]

end

Figure 2: Special implementation of PCG with \( G_{SM} \equiv G_M \).

Now we can continue by induction.

\[ B r^{i+1} = B (r^i - \alpha A v^i) = B r^i - \alpha P_0 v^i = 0, \]
\[ g^{i+1} = G_M r^{i+1} = (I - BA) M (I - AB) r^{i+1}, \text{ i.e. } P_0 g^{i+1} = 0, \]
\[ v^{i+1} = g^{i+1} + \beta_i v^i \Rightarrow P_0 v^{i+1} = 0. \]

\[ \square \]

If \( Br = 0 \) then application of the preconditioner \( G_{SM} \) is realized essentially in two steps like a multiplicative preconditioner \( G_M \)

**multiplicative preconditioner** \( G_M : r \mapsto g \)

(i) \( g_2 = Mr \)

(ii) \( g = (I - BA) g_2 \)

i.e.

\[ G_M = (I - BA) M = (I - P_0) M. \] (13)

The whole implementation gives the algorithm shown in Fig. 2, where \( \alpha_i = \langle r^i, g^i \rangle / \langle Av^i, v^i \rangle \) and \( \beta_i = \langle g^{i+1}, r^{i+1} \rangle / \langle g^i, r^i \rangle \).

5 **DEFLATION METHOD**

We shall show that the algorithm presented in the previous section represents an implementation of the deflation method, which can be described as follows.

Let us split the exact solution \( u^* = A^{-1} b \) of (1) into two components

\[ u^* = u_0 + \bar{u}, \ u_0 = P_0 u^*, \ \bar{u} = (I - P_0) u^* \] (14)

Then the first component is directly computable because

\[ u_0 = P_0 A^{-1} b = B AA^{-1} b = Bb. \] (15)
For the second component, we have
\[ A\vec{u} = (I - P_0)A^{-1}b = (I - P_0^T)b = b - Au_0 = b_D. \]  
(16)

As \( AP_0 = ABA = P_0^TA \) and \( P_D = I - P_0^T \) is again a projection, a multiplication of (16) by \( P_D^2 \) and equality \( P_D\tilde{A} = A(I - P_0) \) shows that \( \vec{u} \) is a solution of the following transformed system
\[ \tilde{A}\vec{u} = b_D, \quad \tilde{A} = (I - P_0)^TA(I - P_0). \]  
(17)

The matrix \( \tilde{A} \) is symmetric and positive semidefinite with a "large" nullspace \( \text{null}(\tilde{A}) = \text{range}(P_0) \), the dimension of \( \text{null}(\tilde{A}) \) is equal to \( r \). The system (17) is consistent and the effective condition number \( \text{effcond}(\tilde{A}) \), which is defined as a ratio of largest and smallest positive eigenvalue, is expected to be less than \( \text{cond}(A) \). Therefore, the system (17) can be solved again by the CG method and the convergence of CG is expected to be faster than for the original system. Owing these properties, it may be convenient to solve (1) by the following three step deflation procedure:

(i) compute \( u_0 = Bb \),

(ii) solve \( \tilde{A}\vec{u} = b - Au_0 \),

(iii) get the solution of (1) in the form \( u = u_0 + (I - P_0)\vec{u} \).

If \( M = EE^T \) is a SPD preconditioner than a transformed solution \( \hat{u} = E^{-1}\vec{u} \) can found by application of the standard CG procedure to the transformed system
\[ \hat{A}\hat{u} = \hat{b}_D, \]  
(18)

\[ \hat{A} = E^T(I - P_0)^TA(I - P_0), \quad \hat{b}_D = E^Tb_D = E^T(b - Au_0), \quad \hat{u} = (I - P_0)E\hat{u}. \]

It leads to an implementation of the deflation method, which can be seen in Fig. 3 (left). Note that accuracy of solving (18) can be controlled by the following overall condition.
\[ \|b - Au\| = \|b - Au_0 - A(I - P_0)E\hat{u}\| \leq \varepsilon \|b\|. \]  
(19)

A transformation of variables
\[ u = (I - P_0)E\hat{u}, \quad v = (I - P_0)E\hat{v}, \quad \hat{r} = E^Tr, \quad \hat{w} = E^Tw \]

with the following relations then makes it possible to transform the implementation of the deflation method to an algorithm, which can be seen in Fig. 3 (right).
solve (17) by CG with accuracy condition (19)

\[ \hat{u} = 0 \]
\[ \hat{r} = b_D = E^T(b - Au_0) \]
\[ \hat{v} = \hat{r} \]

\[ \rho = \langle \hat{r}, \hat{r} \rangle \]
for \( i = 0, 1, \ldots \)

\[ \hat{\omega} = \hat{A}\hat{v} \]
\[ \alpha = \rho/\langle \hat{\omega}, \hat{v} \rangle \]
\[ \hat{u} \leftarrow \hat{u} + \alpha\hat{\omega} \]
\[ \hat{r} \leftarrow \hat{r} - \alpha\hat{A}\hat{v} \]
\[ \beta = 1/\rho \]
\[ \rho = \langle \hat{r}, \hat{r} \rangle \]
\[ \beta = \rho\beta \]
\[ \hat{v} \leftarrow \hat{r} + \beta\hat{v} \]

end

\[ u = u_0 + (I - P_0)\hat{u} \]

Figure 3: Deflation methods: basic (left) and transformed (right).

1. By induction, we can show that the identity \( BE^{-T}\hat{r} = 0 \), and consequently \( (I - P_0^T)E^{-T}\hat{r} = E^{-T}\hat{r} \), holds for the residuals \( \hat{r} \) of the algorithm in Fig. 3 - left.

\[ i = 0 \Rightarrow BE^{-T}\hat{r}_0 = B(b - Ab) = 0 \]
\[ i \geq 0 \Rightarrow BE^{-T}\hat{r}_{i+1} = BE^{-T}\hat{r}_i - \alpha BE^{-T}\hat{A}\hat{v} = -\alpha P_0(I - P_0)E\hat{v} = 0. \]

2. \( \rho = \langle \hat{r}, \hat{r} \rangle = \langle E^T r, E^T r \rangle = \langle r, EETr \rangle = \langle r, Mr \rangle = \langle (I - P_0^T)E^{-T}\hat{r}, Mr \rangle = \langle r, (I - P_0)Mr \rangle = \langle r, g \rangle \)

3. \( w = E^{-T}\hat{\omega} = E^{-T}E^T A(I - P_0)^2E\hat{v} = Av \)

4. \( \langle \hat{\omega}, \hat{v} \rangle = \langle E^T(I - P_0)^T A(I - P_0)E\hat{v}, \hat{v} \rangle = \langle Av, (I - P_0)E\hat{v} \rangle = \langle w, v \rangle \)

5. \( (I - P_0)E\hat{v}_{i+1} = (I - P_0)E\hat{r}_i + \beta(I - P_0)E\hat{v}_i \)
\[ \Rightarrow v_{i+1} = (I - P_0)EETr_i + \beta v_i, \text{ i.e. } v \leftarrow g + \beta v \]
The accuracy of in the transformed algorithm is controlled by the following overall condition, which is equivalent to (19),

\[ \| b - A(u_0 + u) \| \leq \varepsilon \| b \|. \]  

(20)

Note that in the transformed algorithm, we add \( u_0 \) to \( u \) in the end, but we can do this addition also in the beginning. Then our algorithm will be exactly the same as the algorithm from Section 4 (Fig. 2), which is a special implementation of \( G_{SM} \).

Note that another transformation

\[ r = E^T(I - P_0)^T \hat{r} \]

\[ w = E^T(I - P_0)^T \hat{w} \]

\[ \hat{u} = Eu \]

\[ \hat{v} = Ev \]

leads to another transformed algorithm with the preconditioner \( G_D \), which is defined as follows (cf. [9], [11] and [12]):

**deflation preconditioner** \( G_D : r \mapsto g \)

(i) \( w = (I - AB)r^i \),

(ii) \( g = Mw \),

i.e.

\[ G_D = M(I - AB) = G_{M2}^T. \]  

(21)

The global and local preconditioners are in \( G_D \) and \( G_{M2} \) applied in a reverse order. It seems that the arrangement from \( G_{M2} \) has some benefits, at least it does not require additional projections.

For the relations among spectral properties of the introduced preconditioners, we refer to the papers [9], [11] and [12].

The main result from [12] states that

if \( \sigma(G_D A) = \{0, \ldots, 0, \mu_{r+1}, \ldots, \mu_n\} \) then \( \sigma(G_{SM} A) = \{1, \ldots, 1, \mu_{r+1}, \ldots, \mu_n\} \).

### 6 NUMERICAL EXPERIMENTS

The efficiency of various preconditioners described in this paper can be compared by solving a simple model problem

\[ \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = f \quad \text{in} \quad \Omega = (0, 2) \times (0, 3), \]

\[ u = 0 \quad \text{on} \quad \partial \Omega, \]
where \( f = 7.5 + 2.5x_1 + 1.1x_2 \).

The problem is discretized by the linear triangular finite elements on a uniform grid with the mesh size \( h = 1/30 \).

The preconditioner \( M \) is defined as one-level additive Schwarz preconditioner defined by decomposition of \( \Omega \) into eight subdomains \( \Omega_k = (0, 2) \times (x_k, x_{k+1}) \) of the same size \( x_{k+1} - x_k = 13h \) with the overlap \( \delta = 2h \). The subproblems corresponding to the subdomains are solved exactly.

The preconditioner \( B \) is defined by the subspace \( V_0 \subset V_h \), which is defined either by discretization of the boundary value problem on coarser grid with mesh size \( H = 6h \) or by regular aggregation of clusters of \( 6 \times 6 \) nodes. The corresponding subproblem is either solved by a direct method (MATLAB backslash procedure) or by inner CG with relative accuracy \( \varepsilon_0 \) as described in Section 3. Note that \( \varepsilon_0 = 0 \) indicates exact solution.

The required numbers of iterations for the accuracy \( \varepsilon = 10^{-4} \) and various preconditioners can be seen in Tables 1 and 2. The preconditioners are used within the GPCG[1] method.

<table>
<thead>
<tr>
<th>( \varepsilon )</th>
<th>( u_0 )</th>
<th>( G_A )</th>
<th>( G_{SM} )</th>
<th>( G_{M1} )</th>
<th>( G_{M2} )</th>
<th>( G_D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>11</td>
<td>8</td>
<td>8</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>0</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>( 10^{-1} )</td>
<td>0</td>
<td>11</td>
<td>8</td>
<td>8</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>( 10^{-1} )</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>( 10^{-1} )</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>( \vdots )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>( 10^{-5} )</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>( 10^{-5} )</td>
<td>( Bb )</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Numbers of iterations for GPCG[1] and various preconditioners. The preconditioner \( B \) is constructed via coarse grid \( H = 3h \). div > 199.

7 CONCLUDING REMARKS

The aim of this paper is to clear up relations among various multilevel and deflation preconditioners and show their efficiency. The latter issue still needs more work to be done. The present conclusions from the numerical experiments are the following.

- The deflation method can be viewed as an implementation of the CG with \( G_{SM} \) preconditioning. This implementation requires a special initial guess and provides a decrease of the computational work.
Another cheaper modification of $G_{SM}$ is the $G_{M1}$ preconditioner. According to the presented numerical experiments, $G_{M1}$ seems to be much more robust with respect to the inexact solution of the coarse grid problem. On the other hand, it is known that efficiency of $G_{M1}$ decreases in the case of less regular problems, as e.g. problems with discontinuous coefficients, see [5]. A comparison of preconditioners in such cases has to be done.

The second coarse grid space, which was considered in the numerical experiments, use the aggregation technique. It was first introduced in the multigrid context, see e.g. [3] and [7], but it was also intensively studied in the context of deflation, see [13] and [9].

Acknowledgement. Many thanks are due to P. Byczanski for his effort in numerical experiments. The work was supported by the grant 1ET400300415 of the Academy of Sciences of the Czech Republic (Programme Information Society).

REFERENCES


<table>
<thead>
<tr>
<th>$\varepsilon_0$</th>
<th>$u_0$</th>
<th>$G_A$</th>
<th>$G_{SM}$</th>
<th>$G_{M1}$</th>
<th>$G_{M2}$</th>
<th>$G_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>23</td>
<td>16</td>
<td>16</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>0</td>
<td>$Bb$</td>
<td>22</td>
<td>16</td>
<td>17</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>0</td>
<td>24</td>
<td>16</td>
<td>17</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>$Bb$</td>
<td>24</td>
<td>16</td>
<td>17</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>$Bb$</td>
<td>25</td>
<td>16</td>
<td>17</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$Bb$</td>
<td>22</td>
<td>16</td>
<td>17</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$Bb$</td>
<td>22</td>
<td>16</td>
<td>17</td>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>$Bb$</td>
<td>22</td>
<td>16</td>
<td>17</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>$Bb$</td>
<td>22</td>
<td>16</td>
<td>17</td>
<td>16</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 2: Numbers of iterations for GPCG[1] and various preconditioners.

The preconditioner $B$ is constructed via aggregation $3 \times 3$. div $> 199$. 

...


