CONVERGENCE ESTIMATES FOR MULTIGRID ALGORITHMS
WITH KACZMARZ SMOOTHING

KAB SEOK KANG AND DO YOUNG KWAK*

Abstract. We prove some estimates for the convergence of multigrid algorithms with Kaczmarz smoothing applied to symmetric positive definite problems. To estimate the convergence of multigrid algorithms, we assume some properties of smoothing procedure which are satisfied by Kaczmarz, Gauss-Seidel, and Jacobi smoothing and are weaker than earlier assumptions. We use product formula for multigrid algorithm to show the V-cycle convergence factor is \( \delta = 1 - \frac{1}{2G-1} \). The theory is presented in an abstract setting which can be applied to finite element multigrid. Also, numerical example is provided.

1. Introduction. Many authors presented various convergence analyses of multigrid methods which are often based on certain assumptions concerning the smoothing process. These assumptions are sometimes verified for specific examples in [2, 5, 6]. In this paper, we provide a weaker assumption under which multigrid algorithms are shown to converge.

Assumptions concerning the smoothing process which given in [6] are satisfied by point, line and block Jacobi and point, line and block Gauss-Seidel smoothing but not by Kaczmarz smoothing, for which the constant \( C_R \) the assumption (C.1) in [6] grows with \( 1/h^2 \). Instead, the latter satisfies a weaker condition introduced in this paper.

The outline of the remainder of this paper is as follows. Section 2 describes the basic multigrid algorithm in an abstract setting and gives some of the conditions on the smoothers. In §3, we prove multigrid convergence under the weaker assumption. Kaczmarz smoothing procedures are described and analyzed in §4. Some numerical experiments showing the Kaczmarz smoother satisfies our weak assumption are reported there. Finally, in §5, we discuss the finite element multigrid applications.

2. The Multigrid Algorithm. We assume that there is a sequence of nested finite-dimensional inner product spaces \( \mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots \subset \mathcal{M}_j \) with \( (\cdot, \cdot)_k \). In addition, we assume that there are symmetric positive definite operators \( A_k: \mathcal{M}_k \rightarrow \mathcal{M}_k \) for \( k = 1, \ldots, j \). We denote \( A(\cdot, \cdot) = (A_j, \cdot) \). The multigrid algorithm is an iterative procedures for the solution of the problem on \( \mathcal{M}_j \), i.e., given \( f \in \mathcal{M}_j \) find \( u \in \mathcal{M}_j \) satisfying

\[
A_j u = f.
\]

We define the projectors \( P_{k-1}^k: \mathcal{M}_k \rightarrow \mathcal{M}_{k-1} \) and \( P_{k-1}: \mathcal{M}_k \rightarrow \mathcal{M}_{k-1} \) by, for all \( \phi \in \mathcal{M}_{k-1}, (P_{k-1}^k v, \phi)_{k-1} = (v, \phi)_k \) and \( (P_{k-1} v, \phi) = A(v, \phi) \), respectively. "T" and "*" will denote adjoint with respect to \( (\cdot, \cdot)_k \) and \( (\cdot, \cdot) \), respectively.

Also, we require a sequence of linear smoothing operators \( R_k: \mathcal{M}_k \rightarrow \mathcal{M}_k \) for \( k = 2, \ldots, j \). We shall always take \( R_1 = A_1^{-1} \).

We set

\[
R_k^{(l)} = \begin{cases} 
R_k & \text{if } l \text{ is odd}, \\
R_k^T & \text{if } l \text{ is even}
\end{cases}
\]

and set \( K_k = I - R_k A_k \).

* Department of Mathematics, Korea Advanced Institute of Science and Technology, Taejon, Korea 305-701, email: dykwak@math.kaist.ac.kr, ksk002@math.kaist.ac.kr.

We next define a multigrid process for iteratively computing the solution of (1).

**Algorithm** Set $B^*_k = A^{-1}_i$. Assume that $B^*_k - 1$ has been defined and define $B^*_k g$ for $g \in \mathcal{M}_k$ as follows:
1. Set $v^0 = 0$ and $q^0 = 0$.
2. Define $v^i$ for $i = 1, 2, \ldots, m(k)$ by $v^i = v^{i-1} + R_k^{(i+m(k))}(g - A_k v^{i-1})$.
3. Define $w^{m(k)} = v^{m(k)} + q^p$, where $q^i$, for $i = 1, \ldots, p$, is defined by
   $$ q^i = q^{i-1} + B^*_{k-1}[P^0_{k-1}(g - A_k v^{m(k)}) - A_{k-1} q^{i-1}] $$
4. Define $w^i$ for $i = m(k) + 1, \ldots, 2m(k)$ by $w^i = w^{i-1} + R_k^{(i+m(k))}(g - A_k w^{i-1})$.
5. Set $B^*_k g = w^{m(k)}$.

In the above algorithm, by defining $B^*_k g = w^{m(k)}$, we get nonsymmetric multigrid algorithm $B^*_k$. From the above algorithm, fundamental recurrence relations for the nonsymmetric and the symmetric multigrid algorithm are

$$
I - B^*_k A_k = [(I - P_{k-1}) + (I - B^*_k - 1 A_{k-1})^p P_{k-1}] R^{(m(k))}_k,
$$

$$
I - B^*_k A_k = (R^{(m(k))}_k)^* [(I - P_{k-1}) + (I - B^*_k - 1 A_{k-1})^p P_{k-1}] R^{(m(k))}_k
$$

on $\mathcal{M}_k$ where

$$
R^{(m(k))}_k = \begin{cases} 
\frac{(K^*_k K_k)^{m(k)/2}}{K_k (K^*_k K_k)^{(m(k)-1)/2}} & \text{if } m(k) \text{ is even}, \\
K_k (K^*_k K_k)^{(m(k)-1)/2} & \text{if } m(k) \text{ is odd}.
\end{cases}
$$

To estimate the convergence of multigrid algorithm, we need some conditions concerning the smoothing operators. The conditions which were often assumed by many authors ([1–7]) are

**(C.1)** There is a constant $C_R$ which does not depend on $k$ such that the smoothing procedure satisfies

$$
\lambda_k^{-1} \|u\|_k^2 \leq C_R(\tilde{R}_k u, u)_k \quad \text{for all } u \in \mathcal{M}_k.
$$

Here, $\tilde{R}_k$ is either $(I - K^*_k K_k) A_k^{-1}$ or $(I - K_k K^*_k) A_k^{-1}$. $\lambda_k$ is the largest eigenvalue of $A_k$.

**(C.2)** Let $T_k = R_k A_k$. There is a constant $\theta < 2$ not depending on $k$ satisfying

$$
A(T_k u, T_k v) \leq \theta A(T_k v, v).
$$

But (C.1) is not satisfied by some smoothers, say Kaczmarz smoothing. Thus we modify (C.1) as follows

**(SM.1)** There is a constant $C_R$ which does not depend on $k$ such that the smoothing procedure satisfies

$$
\lambda_k^{-2} (A_k u, u)_k \leq C_R(\tilde{R}_k u, u)_k \quad \text{for all } u \in \mathcal{M}_k.
$$

3. **Convergence Estimates for Multigrid Algorithms.** To estimate the convergence of multigrid algorithm, we need some properties concerning the operator $A_k$ and the subspaces. These are as follows:

**(O.1)** There exists a sequence of linear operators $Q_k : \mathcal{M}_j \rightarrow \mathcal{M}_k$ for $k = 1, \ldots, j$, with $Q_k = I$ satisfying the following properties. There are constants $C_3$ and $C_4$ not depending on $k$ for which
\begin{align*}
(2) \quad & A^{-1}_k(Q_k - Q_{k-1})u, (Q_k - Q_{k-1})u_k, \leq C_1 \lambda_k^{-2} (A_k u, u)_k \quad \text{for } k = 2, \ldots, j, \\
& A(Q_k u, Q_k u) \leq C_2 A(u, u) \quad \text{for } k = 1, \ldots, j-1.
\end{align*}

**Theorem 1.** Assume that (O.1) hold. Let \( R_k \) hold (SM.1) and (C.2). Let \( B_j \) be defined by symmetric multigrid algorithm with \( p = 1 \). Then

\begin{align*}
(3) \quad A((I - B_j^p) v, v) \leq \delta_j A(v, v) \quad \text{for all } v \in \mathcal{M}_j
\end{align*}

hold with \( \delta_j = 1 - \frac{1}{C(j-1)^2} \) where \( C = [(1 + C_2^{1/2})(2\theta/(2 - \theta))^{1/2} + (C_v C_1)^{1/2}]^2 \).

**Proof.** We observe that

\[ I - B_j^p A_j = (I - B_j^p A_j)^* (I - B_j^p A_j) \]

and for \( T_k = (I - (E_k^{(m(k))})^*) P_k \), that

\[ (I - B_j^p A_j)^* = (I - T_j)(I - T_{j-1}) \cdots (I - T_1). \]

To use a product analysis, we set \( E_0 = I \) and \( E_k = (I - T_k) E_{k-1} \). We get

\[ A(u, u) - A(E_j u, E_j u) = \sum_{k=1}^j [A(E_{k-1} u, E_{k-1} u) - A(E_k u, E_k u)] \]

\[ = \sum_{k=1}^j A((2I - T_k) E_{k-1} u, T_k E_{k-1} u). \]

Note that \( I - B_j^p A_j = E_j^p \) and hence the inequality (3) will follow if we show that

\begin{align*}
(4) \quad & A(u, u) \leq C(j-1) \sum_{k=1}^j A((2I - T_k) E_{k-1} u, T_k E_{k-1} u).
\end{align*}

From the fact that \( Q_j = I \), we get

\begin{align*}
A(u, u) & = \sum_{k=2}^j A(E_{k-1} u, (Q_k - Q_{k-1})u) + A(u, Q_1 u) \\
& \quad + \sum_{k=2}^j A((I - E_{k-1}) u, (Q_k - Q_{k-1})u).
\end{align*}

For the first sum on the right-hand side (5), from (O.1) and (SM.1), we see that

\begin{align*}
& \sum_{k=2}^j A(E_{k-1} u, (Q_k - Q_{k-1})u) = \sum_{k=2}^j (A_k^{-1} A_k^2 P_k E_{k-1} u, (Q_k - Q_{k-1})u)_k \\
& \leq \sum_{k=2}^j A(A_k P_k E_{k-1} u, A_k P_k E_{k-1} u)^{1/2} \cdot (A_k^{-1} (Q_k - Q_{k-1})u, (Q_k - Q_{k-1})u)_k^{1/2} \\
& \leq (C_v C_1)^{1/2} A^{1/2}(u, u) \sum_{k=2}^j A^{1/2}((I - K_k^* K_k) P_k E_{k-1} u, P_k E_{k-1} u) \\
& \leq (C_v C_1(j-1))^{1/2} A^{1/2}(u, u) \left( \sum_{k=2}^j A((I - K_k^* K_k) P_k E_{k-1} u, P_k E_{k-1} u) \right)^{1/2}.
\end{align*}

The remainder of proof is the same with the proof of Theorem 4.3 in [6]. \( \Box \)
4. Smoothing Procedures in Multigrid Algorithms. We define the multiplicative smoother (Gauss-Seidel smoother) by the following algorithm.

**Algorithm 4.1.** Let \( f \in \mathcal{M}_k \), we define \( R_k f \in \mathcal{M}_k \) as follows:
1. Set \( v_0 = 0 \).
2. Define \( v_i \) for \( i = 1, \ldots, l \) by \( v_i = v_{i-1} + A^{-1}_{k,i} Q_k^i (f - A_k v_{i-1}) \) where \( A_{k,i} \) is an \( i \)-th diagonal element of \( A_k \) and \( Q_k^i \) is a projection onto \( \text{span}\{e_i\} \) with respect to \((\cdot, \cdot)_k\).
3. Set \( R_k f = v_l \).

Let \( P_k^i \) is the projection onto \( \text{span}\{e_i\} \) with respect to \((A_k, \cdot)_k\). It immediately follows from the identity \( A_{k,i} P_k^i = Q_k^i A_k \) that \( K_k = (I - P_k^1) \cdots (I - P_k^l) \). It was shown that Gauss-Seidel smoother satisfies (C.1)[(6)].

**Theorem 2** Let \( R_k \) be a smoother which satisfy (C.1). Then \( R_k \) satisfy (SM.1).

**Proof.** Let \( u = A_k w \), then (C.1) becomes
\[
\lambda_k^{-1}(A_k w, A_k w)_k \leq C_R[(A_k w, w)_k - (A_k K_k w, K_k w)_k].
\]
From the fact, for all \( w \in \mathcal{M}_k \), \((A_k w, w)_k \leq \lambda_k(w, w)_k\), (SM.1) is satisfied. \( \square \)

Kaczmaz smoother is defined by the following algorithm.

**Algorithm 4.2.** Let \( f \in \mathcal{M}_k \). We define \( \tilde{R}_k f \in \mathcal{M}_k \) as follows:
1. Set \( v_0 = 0 \).
2. Define \( v_i \) for \( i = 1, \ldots, l \) by \( v_i = v_{i-1} - \frac{a_i}{a_i^T a_i} (a_i^T v_{i-1} - f_i) \) where \( a_i^T = i \)-th row of \( A_k \).
3. Set \( R_k f = v_l \).

From the above algorithm, we know that
\[
K_k = I - R_k A_k = (I - B_k^1) \cdots (I - B_k^l) \quad \text{where} \quad B_k^i = \frac{a_i a_i^T}{a_i^T a_i} = \frac{A_k^T Q_k^i A_k}{(A_k Q_k^i A_k)_i}.
\]

Let \( S_k^i \) be the projection with respect to \((A_k A_k^T, \cdot)_k\). Then we get \((A_k A_k^T)_{ii} S_k^i = Q_k^i (A_k A_k^T)_{ii}\). From \( B_k^i = A_k^T S_k^i A_k^T \), (6) becomes \( K_k = A_k^T (I - S_k^1) \cdots (I - S_k^l) A_k^T \).

Above presentation reflects that Kaczmaz iteration can be regarded as a Gauss-Seidel iteration applied to \( A_k A_k^T u = f \). In fact, from Gauss-Seidel iteration, we know that
\[
(I - S_k^1) \cdots (I - S_k^l) = I - (D + L)^{-1}(A_k A_k^T) \quad \text{where} \quad A_k A_k^T = D + L + L^T.
\]

Therefore
\[
K_k = I - A_k^T (D + L)^{-1} A_k.
\]

**Numerical verification of (SM.1) and (C.2).** We consider an elliptic partial differential equation of the form
\[
\begin{align*}
- \sum_{i,j=1}^{2} \frac{\partial}{\partial x_i} \left( a_{ij}(x) \frac{\partial v}{\partial x_j}(x) \right) &= f(x) \quad \text{in} \ \Omega, \\
v(x) &= 0 \quad \text{on} \ \partial \Omega
\end{align*}
\]
where \( \Omega \) is a unit square and \((a_{ij})_{i,j=1}^{2} \) is a symmetric positive matrix.

To obtain \( A_k \), we discretize \( \Omega \) by quasi-uniform triangular element with \( h_k = 2^{-k} \) and define \( \mathcal{M}_k \) to be the set of piecewise linear functions which vanish on \( \partial \Omega \).

First, for \((a_{ij}) = I\), we calculate \( C_R \)’s in (C.1) and (SM.1) of damped Jacobi with \( \omega = 0.8 \), Gauss-Seidel, and Kaczmarz smoothing.
Table I

Table I show that Kaczmarz smoother does not satisfy (C.1).
Next, we calculate $C_R$ in (SM.1) and $\theta$ in (C.2) of Kaczmarz smoother for other $(a_{ij})^*$s.

<table>
<thead>
<tr>
<th>$h$</th>
<th>Jacobi (C.1)</th>
<th>Gauss-Seidel (C.1)</th>
<th>Kaczmarz (C.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>1.409707</td>
<td>1.118052</td>
<td>2.281606</td>
</tr>
<tr>
<td>1/16</td>
<td>1.512971</td>
<td>1.123504</td>
<td>8.349859</td>
</tr>
<tr>
<td>1/32</td>
<td>1.551332</td>
<td>1.124665</td>
<td>32.667680</td>
</tr>
<tr>
<td>1/64</td>
<td>1.559570</td>
<td>1.124900</td>
<td>129.937300</td>
</tr>
</tbody>
</table>

Table II

Here $p(x) = e^{0.2x + 0.7y}$ and $q(x) = \text{diag}(e^{0.2x + 0.3y}, x + 0.5)$.

5. Finite Elements Applications. We shall consider the problem of approximating the solution $v$ of (7). The form $A$ corresponding to the above operator is given by

$$A(v, w) = \sum_{i,j=1}^{n} \int_{\Omega} a_{ij} \frac{\partial v}{\partial x_i} \frac{\partial w}{\partial x_j} \, dx$$

for all $v, w \in M_k$.

Clearly, $U \in H^1_0(\Omega)$ is the solution of $A(U, \theta) = (F, \theta)$ for all $\theta \in H^1_0(\Omega)$.

By positive definiteness of $\{a_{ij}\}$, $\| \cdot \|_A = A^{1/2} (\cdot, \cdot)$ is a norm on $H^1_0(\Omega)$ and this norm is equivalence to $\| \cdot \|_1$ which denote $H^1(\Omega)$-norm.

We let $t_k^{(j)}$ be a sequence of quasi-uniform triangulations of size $h_k$ for $k = 1, \ldots, j$. We define $M_k$ as in the previous section. Since $\Omega$ is polygonal, the subspaces are nested.

Let $\{y_k\}$ be the collection of nodes corresponding to the triangulation for $M_k$. Let $(u, v)_k = h_k^2 \sum_i u(y_k^i)v(y_k^i)$. Note that the quasi-uniformity of the triangulations implies that the norm $\| \cdot \|_k$ is equivalent to the $L^2$ norm on the subspace $M_k$. The operator $A_k, k = 1, \ldots, j$, are then defined by, for all $\phi \in M_k, (A_k v, \phi)_k = A(v, \phi)$.

Let $Q_k$ denote the $L^2(\Omega)$ projection onto $M_k$. We know that, since the triangulations are quasi-uniform and inverse property, for all $v \in H^1_0(\Omega)$,

$$\| (I - Q_k) v \| \leq ch_k \| v \|_1 \quad \text{and} \quad \| Q_k v \|_1 \leq C \| v \|_1.$$

From (8) and definition, we get

$$\begin{align*}
(A_k^{-1}(Q_k - Q_{k-1})v, (Q_k - Q_{k-1})v)_k &= \sup_{u \in M_k} \frac{(Q_k - Q_{k-1})v, (Q_k - Q_{k-1})v)_k}{(A_k u, u)_k} = \frac{((Q_k - Q_{k-1})v, (Q_k - Q_{k-1})v)_k}{((A_k u, u)_k)_{k}} \\
&\leq \frac{\| Q_k - Q_{k-1} \|_k \| v \|_1}{ch_k^2 \| v \|_1} \leq C h_k^2 \| k \|_k \| v \|_1 \\
&\leq C h_k^2 \| (A_k u, v) \|_k \leq C \chi_k^{-2} (A_k u, v)_k
\end{align*}$$

(9)
since \( \| (Q_k - Q_{k-1})v \| \leq \| (I - Q_k)v \| + \| (I - Q_{k-1})v \| \) and \( \lambda_k = O(h_k^{-2}) \). Combining (8) and (9) shows that (0.1).

REFERENCES


