Bull. Korean Math. Soc. **34** (1997), No. 4, pp. 667–678

PERFORMANCE ANALYSIS OF A FLEXIBLE RESTARTED FOM(k) ALGORITHM

JAE HEON YUN

ABSTRACT. This paper contains a convergence analysis of a flexible restarted FOM(k)(FFOM(k)), and its performance is compared with FGMRES(k). Performances of these two algorithms with variable preconditioners are also compared with those of preconditioned FOM(k) and GMRES(k). Numerical experiments show that FFOM(k) performs as well as, or better than for some problems, FGMRES(k).

1. Introduction

Many iterative methods based on the Krylov subspace techniques for solving large sparse nonsymmetric linear systems have been proposed in the last decades. Although iterative methods lack the robustness of direct methods, they are effective for the large class of problems arising from the elliptic partial differential equations. For the robustness and acceleration of convergence of iterative methods, preconditioning techniques such as incomplete factorization preconditioners have been presented in the mid-seventies [1, 3].

In order to be able to enhance robustness of iterative methods, we should be able to change preconditioner if a given preconditioner is not suitable for the problem at hand. To this end, Saad [5] proposed the *flexible GMRES*(FGMRES) which allows changes in the preconditioning at every step. An important property of FGMRES is that it satisfies the residual norm minimization property over the preconditioned Krylov subspace just as in the standard GMRES algorithm [6].

Received May 27, 1997.
1991 Mathematics Subject Classification: 65F10.
Key words and phrases: FOM algorithm, Krylov subspace, breakdown, stagnation, preconditioner, FGMRES algorithm.

Saad [4] also proposes the Full Orthogonalization Method(FOM) which uses the Galerkin property rather than the residual norm minimization property of GMRES. In this paper, we introduce a *flexible restarted FOM(k)*(FFOM(k)) which can be derived from the right-preconditioned FOM(k) in the similar way as was done in the restarted FGMRES(k) [5]. It is well-known that GMRES(k) can not break down unless it has already converged, while FGMRES(k) and FOM(k) may break down without convergence. To this end, we analyze convergence and breakdown properties of FFOM(k), and then numerical results of FFOM(k) with a certain criterion are compared with those of FGMRES(k).

Throughout the paper, we consider a linear system $Ax = b$, where $A \in R^{n\times n}$ is a large sparse nonsymmetric nonsingular matrix, $x \in R^n$, and $b \in R^n$. Given a set of vectors $\{p_0, p_1, \ldots, p_k\}$, let $\langle p_0, p_1, \ldots, p_k\rangle$ denote the subspace spanned by $\{p_0, p_1, \ldots, p_k\}$. For a given vector c_0, let the m-th *Krylov subspace* $K_m(A, c_0)$ denote the subspace $\langle c_0, Ac_0, \ldots, A^{m-1}c_0\rangle$. $(\cdot,\cdot)$ denotes the Euclidean inner product on $R^n \times R^n$, and $\|\cdot\|$ denotes the Euclidean vector norm on R^n as well as matrix norm associated with the Euclidean vector norm.

2. A flexible restarted FOM(k) algorithm(FFOM(k))

The FOM algorithm introduced in [4] for solving a linear system $Ax = b$ uses the Galerkin condition on the Krylov subspace to generate a sequence of approximate solutions. The right preconditioning technique is to apply a Krylov subspace method to a modified linear system $AM^{-1}(Mx) = b$, where M is a right preconditioner that approximates A and can be easily inverted. The restarted FOM(k) algorithm with a fixed right preconditioner M is obtained by applying FOM(k) to $AM^{-1}y = b$, where $y = Mx$. As was done in FGMRES(k) [5], a *flexible FOM(k)*(FFOM(k)) algorithm which uses a variable preconditioner M_j at the j-th step can be easily derived from FOM(k) algorithm with a fixed right preconditioner M:

ALGORITHM 2.1 : FFOM(k) algorithm

1. Choose x_0 and compute $r_0 = b - Ax_0$
   Compute $v_1 = r_0 / \| r_0 \|$ and set $\beta = \| r_0 \|$
2. **for** $j = 1, 2, \ldots, k$

Compute $z_j = M_j^{-1} v_j$
Compute $\hat{v}_{j+1} = A z_j$
for $i = 1, 2, \ldots, j$
$h_{ij} = (\hat{v}_{j+1}, v_i)$
$\hat{v}_{j+1} = \hat{v}_{j+1} - h_{ij} v_i$
Compute $h_{j+1,j} = \| \hat{v}_{j+1} \|$ and $v_{j+1} = \hat{v}_{j+1} / h_{j+1,j}$

3. Form the approximate solution:
Compute $x_k = x_0 + Z_k y_k$, where $y_k = H_k^{-1} \beta e_1$, $Z_k = [z_1, z_2, \ldots, z_k]$ is the $n \times k$ matrix, H_k is the upper $k \times k$ Hessenberg matrix whose entries are the scalars h_{ij}, and $e_1 = [1, 0, \ldots, 0]^T \in R^k$
4. Restart:
Compute $r_k = b - A x_k$. **If** $\| r_k \| / \| r_0 \| <$ (tolerance), stop. Otherwise, set $x_0 = x_k$ and $v_1 = r_k / \| r_k \|$, and then go to 2

If $M_j = M$ for all j, then Algorithm 2.1 is equivalent to the FOM(k) with a fixed right preconditioner M. It is easy to see that FFOM(k) satisfies

$$AZ_l = V_{l+1} \widehat{H}_l = V_l H_l + \hat{v}_{l+1} e_l^T \quad \text{for} \quad 1 \le l \le k, \tag{1}$$

where $V_{l+1} = [v_1, \ldots, v_{l+1}]$ is the $n \times (l+1)$ matrix with orthonormal columns and $\widehat{H}_l$ is the upper $(l+1) \times l$ Hessenberg matrix whose only nonzero entries are the scalars h_{ij} generated by the FFOM(k) algorithm. From equation (1), one can obtain

$$V_l^T A Z_l = H_l. \tag{2}$$

Equation (2) shows that x_l in the FFOM(k) is chosen so that $r_l = r_0 - AZ_l y_l$ is orthogonal to $\langle v_1, v_2, \ldots, v_l \rangle$. Notice that x_l in FGMRES(k) is chosen so that r_l is orthogonal to $\langle Az_1, Az_2, \ldots, Az_l \rangle$. Instead of forming a preconditioner M_j at the j-th step explicitly and then computing $z_j = M_j^{-1} v_j$, z_j is chosen so that z_j is an approximate solution to $Ax = v_j$. An approximate solution z_j to $Ax = v_j$ can be obtained using any iterative methods available, e.g., SOR, GMRES, CGNR, BiCGSTAB, etc.

If H_l is nonsingular, FFOM(k) does not break down, i.e., $x_l = x_0 + Z_l y_l$ exists. Moreover, if H_l is nonsingular and $h_{l+1,l} = 0$, then it can be easily shown that x_l in the FFOM(k) will be the exact solution to $Ax = b$. The solution y_l to $H_l y = \beta e_1$ can be solved efficiently using QR factorization which is carried out by Givens rotations. That is, H_l is factorized into $H_l = Q_l^T R_l$, where Q_l is an orthogonal matrix of order l and R_l is an upper triangular matrix of order l. For $l = 1$, $Q_1 = [1]$ and $R_1 = [h_{11}]$. The matrix Q_l is formed by the product $F_{l-1} \cdots F_2 F_1$, where the matrix $F_m (1 \leq m \leq l-1)$ represents rotation matrix of order l:

$$F_m = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & 0 & \\ & & 1 & & & & & \\ & & & c_m & -s_m & & & \\ & & & s_m & c_m & & & \\ & & & & & 1 & & \\ & 0 & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \leftarrow \text{row } m \\ \leftarrow \text{row } m+1 \\ \\ \\ \\ \end{matrix} .$$

Suppose that $\hat{h}_{ll}$ denotes the l-th diagonal element of the upper triangular matrix R_l. Then, c_l and s_l should be chosen as follows:

$$c_l = \frac{\hat{h}_{ll}}{\sqrt{\hat{h}_{ll}^2 + h_{l+1,l}^2}} \text{ and } s_l = -\frac{h_{l+1,l}}{\sqrt{\hat{h}_{ll}^2 + h_{l+1,l}^2}}. \tag{3}$$

Brown[2] showed that FOM algorithm satisfies

$$\| r_l \| = |s_1 s_2 \cdots s_l| \, \beta \, / \, c_l. \tag{4}$$

In the similar way as was done in the FOM, it can be shown that (4) still holds for the FFOM(k) algorithm. From (4), we can obtain

$$\begin{aligned} \| r_l \| &= \left| c_{l-1} \frac{s_l}{c_l} \right| \, \| r_{l-1} \| \\ &\leq \left| \frac{s_l}{c_l} \right| \, \| r_{l-1} \| \\ &= \left| \frac{h_{l+1,l}}{\hat{h}_{ll}} \right| \, \| r_{l-1} \| \end{aligned} \tag{5}$$

(5) implies that to accelerate the convergence rate of the residual norm in the FFOM(k) we should try to make $\left| \frac{h_{l+1,l}}{\hat{h}_{ll}} \right|$ as small as possible. To this end, we consider in the next section how to choose a variable

preconditioner M_j at the j-th step so that FFOM(k) converges fast to the exact solution without breakdown and stagnation.

3. Convergence analysis of FFOM(k)

Notice that in the FFOM(k) a variable preconditioner M_j is not formed explicitly, but $z_j(= M_j^{-1}v_j)$ is directly chosen as an approximate solution to $Ax = v_j$.

LEMMA 3.1. *Suppose that* $\| Az_j - v_j \| = \| A(M_j^{-1}v_j) - v_j \| \le \varepsilon$ *for* $1 \le j \le l$. *Then, for each* $1 \le j \le l$

$$(6) \qquad \begin{aligned} 1 - \varepsilon \le h_{jj} \le 1 + \varepsilon \\ |h_{ij}| \le \varepsilon \ (i \ne j,\ 1 \le i \le j - 1) \end{aligned}$$

Moreover, if $\varepsilon < \frac{1}{e+1}$ *for a positive number* e, *then* $0 \le \frac{h_{j+1,j}}{h_{jj}} < \frac{1}{e}$ *for* $1 \le j \le l$.

Proof. For each $1 \le j \le l$, $Az_j = \sum_{i=1}^{j} h_{ij}v_i + h_{j+1,j}v_{j+1}$. Since the vectors $v_1, v_2, \ldots, v_{j+1}$ are orthonormal,

$$\| Az_j - v_j \|^2 = \sum_{i=1}^{j-1} h_{ij}^2 + (h_{jj} - 1)^2 + h_{j+1,j}^2.$$

Since $\| Az_j - v_j \| \le \varepsilon$, $|h_{ij}| \le \varepsilon$ for $i \ne j$ and $1 \le i \le j + 1$, and $|h_{jj} - 1| \le \varepsilon$. Thus, (6) is proved. if $\varepsilon < \frac{1}{e+1}$ for a positive number e, from (6) $h_{jj} > \frac{e}{e+1} > e\varepsilon$. Since $0 \le h_{j+1,j} \le \varepsilon$, $0 \le \frac{h_{j+1,j}}{h_{jj}} < \frac{1}{e}$ is obtained. □

For simplicity of exposition, we define new scalars $\hat{h}_{ij}$ which are generated from the following algorithm:

for $j = 1, 2, \ldots, k$
  $\hat{h}_{1j} = h_{1j}$
for $i = 2, \ldots, k$
  for $j = i, \ldots, k$
    $\hat{h}_{ij} = s_{i-1}\hat{h}_{i-1,j} + c_{i-1}h_{ij}$,

where h_{ij}'s are scalars generated from FFOM(k) algorithm, and c_i and s_i are scalars defined by equation (3). Then, it is easy to show that for each $1 \le i \le k$, $\hat{h}_{ii}$ generated from the above algorithm is the i-th diagonal element of the upper triangular matrix R_i such that $H_i = Q_i^T R_i$.

LEMMA 3.2. *Let α and e be positive numbers. Suppose that $s^2 + c^2 = 1$, where $0 \le s \le \frac{1}{\sqrt{1+e^2}}$ and $c \ge 0$. Then, $s + \alpha c$ has a maximum value*

$$\begin{cases} \sqrt{1+\alpha^2} & \text{if } \alpha \ge e \\ \frac{1+\alpha e}{\sqrt{1+e^2}} & \text{if } \alpha \le e \end{cases}$$

Proof. Let $f(s) = s + \alpha c = s + \alpha\sqrt{1 - s^2}$, where $0 \le s \le \frac{1}{\sqrt{1+e^2}}$. By simple calculus, the lemma can be easily shown. □

THEOREM 3.3. *Let e be a given number such that $1 \le e \le 2$, $d = 2e^3 + e + 1$, $\beta = \frac{1}{\sqrt{1+e^2}}$, and $f(e) = \frac{2e^3}{\sqrt{5e^2+d+(2e^5+d)\beta}}$. If $\| Az_j - v_j \| \le \varepsilon$ for $1 \le j \le l$ and $\varepsilon < f(e)$, then $0 \le \frac{h_{j+1,j}}{\hat{h}_{jj}} < \frac{1}{e}$ for $1 \le j \le l$, i.e., H_j's are nonsingular for $1 \le j \le l$.*

Proof. For $l = 1$, $\hat{h}_{11} = h_{11}$. Since $\varepsilon < f(e) < \frac{1}{1+e}$, by Lemma 3.1 $0 \le \frac{h_{21}}{\hat{h}_{11}} < \frac{1}{e}$. Suppose that the theorem holds for $l = m$. Then, we must show that the theorem holds for $l = m+1$. By induction hypothesis, one obtains

$$|s_j| = \frac{h_{j+1,j}}{\sqrt{\hat{h}_{jj}^2 + h_{j+1,j}^2}} < \beta \text{ and } e\beta < |c_j| \le 1 \text{ for } 1 \le j \le m.$$

Applying Givens rotations to the upper Hessenberg matrix H_{m+1} of order $m+1$,

$$\begin{aligned} \hat{h}_{m+1,m+1} &= s_m \hat{h}_{m,m+1} + c_m h_{m+1,m+1} \\ &= s_m(s_{m-1}\hat{h}_{m-1,m+1} + c_{m-1}h_{m,m+1}) + c_m h_{m+1,m+1} \\ &= s_m s_{m-1}\hat{h}_{m-1,m+1} + s_m c_{m-1} h_{m,m+1} + c_m h_{m+1,m+1} \\ &\ \ \vdots \\ &= s_m \cdots s_2 s_1 h_{1,m+1} + s_m \cdots s_2 c_1 h_{2,m+1} + s_m \cdots s_3 c_2 h_{3,m+1} + \\ &\quad \cdots + s_m s_{m-1} c_{m-2} h_{m-1,m+1} + s_m c_{m-1} h_{m,m+1} + c_m h_{m+1,m+1}. \end{aligned}$$

Since $|h_{i,m+1}| \leq \varepsilon$ for $i \leq m$,

$$\begin{aligned}
&|\hat{h}_{m+1,m+1} - c_m h_{m+1,m+1}| \\
&\quad \leq \varepsilon\, |s_m|\, (\, |s_{m-1} \cdots s_2 s_1| + |s_{m-1} \cdots s_2 c_1| + \ldots + |s_{m-1} c_{m-2}| + |c_{m-1}|\,) \\
&\quad = \frac{\varepsilon}{2}\, |s_m|\, (\, |s_{m-1} \cdots s_1| + |s_{m-1} \cdots s_2|\, (\, |s_1| + |c_1|\,) + |s_{m-1} \cdots s_3|\, (\, |c_1| \cdot \\
&\qquad |s_2| + |c_2|\,) + \ldots + |s_{m-1}|\, (\, |c_{m-3}||s_{m-2}| + |c_{m-2}|\,) + (\, |c_{m-2}||s_{m-1}| \\
&\qquad + 2|c_{m-1}|)\,) \\
&\quad \leq \frac{\varepsilon}{2}\, |s_m|\, (\, |s_{m-1} \cdots s_1| + |s_{m-1} \cdots s_2|\, (\, |s_1| + |c_1|\,) + |s_{m-1} \cdots s_3|\, (\, |s_2| \\
&\qquad + |c_2|\,) + \ldots + |s_{m-1}|\, (\, |s_{m-2}| + |c_{m-2}|\,) + (\, |s_{m-1}| + 2|c_{m-1}|\,)\,).
\end{aligned}$$

Using Lemma 3.2 and $|s_j| < \beta$ for $1 \leq j \leq m$,

$$\begin{aligned}
|\hat{h}_{m+1,m+1} - c_m h_{m+1,m+1}| &\leq \frac{\varepsilon}{2}\, \beta\, (\beta^{m-1} + \beta^{m-1}(1+e) + \ldots \\
&\qquad + \beta^2\, (1+e) + \sqrt{5}\,) \\
&< \frac{\varepsilon}{2}\, \beta \left(\frac{\beta^2}{1-\beta}(1+e) + \sqrt{5} \right).
\end{aligned} \tag{7}$$

Using (7) and Lemma 3.1,

$$\begin{aligned}
\hat{h}_{m+1,m+1} &\geq c_m h_{m+1,m+1} - \frac{\varepsilon}{2}\, \beta \left(\frac{\beta^2}{1-\beta}(1+e) + \sqrt{5} \right) \\
&> \beta\, e\, (1-\varepsilon) - \frac{\varepsilon}{2}\, \beta \left(\frac{\beta^2}{1-\beta}(1+e) + \sqrt{5} \right) \\
&= \beta \left(e - \varepsilon \left(\frac{\beta^2}{1-\beta} \frac{1+e}{2} + e + \frac{\sqrt{5}}{2} \right) \right).
\end{aligned}$$

If $\varepsilon < f(e)$, then $\beta \left(e - \varepsilon \left(\frac{\beta^2}{1-\beta} \frac{1+e}{2} + e + \frac{\sqrt{5}}{2} \right) \right) > e\varepsilon$. Since $0 \leq h_{m+2,m+1} < \varepsilon$, $\hat{h}_{m+1,m+1} > e h_{m+2,m+1}$ and so $0 \leq \frac{h_{m+2,m+1}}{\hat{h}_{m+1,m+1}} < \frac{1}{e}$. Therefore, the proof is complete. □

THEOREM 3.4. *Let e be a given number such that $e \geq 2$, $\hat{d} = 2e^5 + 3e^3 + e^2 + e + 1$, $\beta = \sqrt{1+e^2}$, and $f(e) = \frac{2e^3\sqrt{1+e^2}}{e^3 + (1+\beta)\hat{d}}$. If $\| Az_j - v_j \| \leq \varepsilon$ for $1 \leq j \leq l$ and $\varepsilon < f(e)$, then $0 \leq \frac{h_{j+1,j}}{\hat{h}_{jj}} < \frac{1}{e}$ for $1 \leq j \leq l$.*

Proof. For $l = 1$, $\hat{h}_{11} = h_{11}$. Since $\varepsilon < f(e) < \frac{1}{1+e}$, by Lemma 3.1 $0 \le \frac{h_{21}}{\hat{h}_{11}} < \frac{1}{e}$. Suppose that the theorem holds for $l = m$. Then, we must show that the theorem holds for $l = m + 1$. In the similar way as was done in Theorem 3.3, one obtains the following inequality

$$
\begin{aligned}
|\hat{h}_{m+1,m+1} - c_m h_{m+1,m+1}| &\le \frac{\varepsilon}{2}\beta\,(\beta^{m-1} + \beta^{m-1}(1+e) + \ldots \\
&\quad + \beta^2\,(1+e) + \beta(1+2e)\,) \\
&< \frac{\varepsilon}{2}\beta^2\left(\frac{1+e}{1-\beta} + e\right).
\end{aligned} \tag{8}
$$

Using (8) and Lemma 3.1,

$$
\begin{aligned}
\hat{h}_{m+1,m+1} &> \beta\, e(1-\varepsilon) - \frac{\varepsilon}{2}\beta^2\left(\frac{1+e}{1-\beta} + e\right) \\
&= \beta\left(e - \varepsilon\left(\frac{\beta}{1-\beta}\frac{1+e}{2} + \frac{\beta}{2}e + e\right)\right)
\end{aligned}
$$

If $\varepsilon < f(e)$, then $\beta\left(e - \varepsilon\left(\frac{\beta}{1-\beta}\frac{1+e}{2} + \frac{\beta}{2}e + e\right)\right) > e\,\varepsilon$. Since $0 \le h_{m+2,m+1} < \varepsilon$, the proof is complete. □

By a routine calculus, it can be shown that $f(e)$ is a decreasing function for $e \ge 2$. Table 1 shows the values of $f(e)$ for $e \ge 1$ which is defined in Theorems 3.3 and 3.4. The larger e is, the faster FFOM(k) converges to the exact solution. So, the estimation $f(e)$ for ε should be larger for smaller number of e. From this point of view, it can be said that $f(e)$ is not a good estimation for ε when $e < 1.8$.

e	$f(e)$	e	$f(e)$
1.0	0.1909	2.0	0.2459
1.5	0.2425	3.0	0.2160
1.7	0.2473	4.0	0.1837
1.75	0.2476	5.0	0.1576
1.8	0.2477	6.0	0.1374
1.85	0.2476	7.0	0.1214
1.9	0.2472	8.0	0.1086
2.0	0.2459	9.0	0.0982

TABLE 1: Values of $f(e)$ for $e \ge 1$

From Table 1, it can be seen that the maximum of $f(e)$ for $e \geq 1$ is about 0.2477 when e is equal to about 1.8. Hence, the following corollary can be obtained from (5).

COROLLARY 3.5. *Suppose that* $\| Az_j - v_j \| \leq \varepsilon$ *for* $1 \leq j \leq k$. *If* $\varepsilon < f(1.8) \approx 0.2477$, *then FFOM(k) converges to the exact solution without breakdown and stagnation, and* $\frac{\|r_j\|}{\|r_{j-1}\|} < \frac{1}{1.8}$ *for* $1 \leq j \leq k$.

4. Numerical results

We report some numerical experiments comparing performances of FFOM(20) with those of FGMRES(20). These algorithms using variable preconditioners are also compared with standard ILU(0) right-preconditioned FOM(20) and GMRES(20) which are called PFOM(20) and PGMRES(20), respectively. The tests were performed using 64-bit arithmetics. In all cases, the iteration was started with $x_0 = 0$ and the iterations were terminated when $\frac{\|r_n\|}{\|r_0\|} < 10^{-8}$.

The preconditioned vectors $z_j = M_j^{-1} v_j$ were computed using m iteration steps of the right-preconditioned BiCGSTAB[7] with minimal residual smoothing technique which is called PBSTABMR. PBSTABMR was also started with zero initial vector. Minimal residual smoothing technique is used to avoid an irregular convergence behavior of BiCGSTAB. An advantage of using BiCGSTAB to compute preconditioned vectors z_j is that it uses less storages than other GMRES-type methods. It was shown in Corollary 3.5 that ε set to $f(1.8) \approx 0.2477$ guarantees the convergence of FFOM(k) to the exact solution. For most problems PBSTABMR yields z_j satisfying $\| v_j - Az_j \| \leq \varepsilon$ within a few iteration steps, but for some problems it converges too slow, so that its execution is limited to m iteration steps for efficient performance, where m is a fixed number. Performance of FFOM(k) varies depending upon m, and an optimal number of m depends upon problems to be considered. Numerical experiments show that the optimal number of m ranges from 2 to 5. *Matrix-vector products*(SPMV) for computing Ay and *preconditioner solves*(SPSV) for computing $M^{-1}y$ are counted for performance evaluation of each algorithm. Vector updates and inner products are neglected since their execution time is relatively small compared with SPMV and SPSV.

EXAMPLE 4.1. In the first example, we consider a block tridiagonal matrix A of the form

$$A = \begin{pmatrix} D & B & & & \\ C & D & B & & \\ & \ddots & \ddots & \ddots & \\ & & C & D & B \\ & & & C & D \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} 4 & \alpha & & & \\ \beta & 4 & \alpha & & \\ & \ddots & \ddots & \ddots & \\ & & \beta & 4 & \alpha \\ & & & \beta & 4 \end{pmatrix},$$

where $\alpha = -1 + \delta$, $\beta = -1 - \delta$, $B = (-1 + \delta)I$, $C = (-1 - \delta)I$. We consider two cases of $\delta = 0.2$ and $\delta = 0.5$. The right-hand side vector b is chosen so that $b = A[1, 1, \dots, 1]^T$. $m = 5$ is chosen for computation of preconditioned vectors z_j.

EXAMPLE 4.2. As the second example, we consider the problems arising from the centered difference discretization of PDE problems of the form

$$-\Delta u + \gamma(xu_x + yu_y) + \beta u = f$$

on square regions $[0, 1] \times [0, 1]$ with zero Dirichlet boundary conditions. We consider two different cases of the parameters β and γ - $\beta = -100$ and $\gamma = 10$, and $\beta = 10$ and $\gamma = 1000$. The grids we have used consist of 32 and 48 internal mesh points in each direction leading to matrices of size $n = 1024$ and $n = 2304$, respectively. Once the matrix A is constructed, the right-hand side vector b is chosen so that $b = A[1, 1, \dots, 1]^T$. $m = 2$ is chosen for computation of preconditioned vectors z_j.

Tables 2 and 3 contain numerical results for Examples 4.1 and 4.2 respectively, and they contain data of the form $a + b = c$, where a represents the number of SPMV operation, b represents the number of SPSV operation, and c represents the sum of a and b. NC in Table 3 means that algorithm does not satisfy the termination criterion within 600 iteration steps. FFOM(20) performs as well as FGMRES(20) for Example 4.1, and FFOM(20) performs better than FGMRES(20) for Example 4.2. FFOM(20) and FGMRES(20) outperform PFOM(20) and PGMRES(20) for the problems for which PFOM(20) and PGMRES(20) perform poorly.

n	δ	PFOM(20)	FFOM(20)	PGMRES(20)	FGMRES(20)
2500	0.2	$57 + 57 = 114$	$62 + 46 = 108$	$58 + 58 = 116$	$62 + 46 = 108$
	0.5	$25 + 25 = 50$	$39 + 28 = 67$	$25 + 25 = 50$	$39 + 28 = 67$
4900	0.2	$101 + 101 = 202$	$81 + 66 = 147$	$101 + 101 = 202$	$81 + 66 = 147$
	0.5	$39 + 39 = 78$	$47 + 36 = 83$	$39 + 39 = 78$	$47 + 36 = 83$

TABLE 2: Numerical results for Example 4.1

n	β	γ	PFOM(20)	FFOM(20)	PGMRES(20)	FGMRES(20)
1024	-100	10	NC	$123 + 92 = 215$	NC	$131 + 98 = 229$
	10	1000	$209 + 209 = 418$	$595 + 470 = 1065$	$224 + 224 = 448$	$686 + 542 = 1228$
2304	-100	10	NC	$157 + 118 = 275$	NC	$159 + 120 = 279$
	10	1000	$167 + 167 = 334$	$592 + 468 = 1060$	$158 + 158 = 316$	$688 + 544 = 1232$

TABLE 3: Numerical results for Example 4.2

5. Concluding Remarks

Most of the existing preconditioned iterative methods use a fixed preconditioner which can be usually found using various incomplete factorization techniques, see [1, 3] for details. For indefinite and/or highly nonsymmetric matrices, the performance of an iterative method with a fixed preconditioner can be unpredictable, see the performances of PFOM(20) and PGMRES(20) in Table 3. It was shown that FFOM(k) yields a sequence of approximate solutions which converges to the exact solution by choosing a suitable preconditioner every iteration. Hence, a big advantage of FFOM(k) with a variable preconditioner is its robustness. However, PFOM(k) can fail to yield a sequence of approximate solutions which converges to the exact solution.

If PFOM(k) with a fixed preconditioner does not perform well, then we need to use another preconditioner for which it performs well. Finding a good preconditioner for PFOM(k) requires complicated computational steps, while a good variable preconditioner for FFOM(k) can be easily obtained by using any iterative methods available. Numerical experiments show that FFOM(k) performs as well as, or better than for some problems, FGMRES(k). Hence, it may be concluded that FFOM(k) can be used as a good substitute for FGMRES(k).

References

[1] R. Barrett et al, *Templates for the solution of linear systems: Building blocks for iterative methods*, SIAM, Philadelphia, PA, 1993.

[2] P. N. Brown, *A theoretical comparison of the Arnoldi and GMRES algorithms,* SIAM J. Sci. Statist. Comput. **12** (1991), 58-78.

[3] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix,* Math. Comput. **31** (1977), 148-162.

[4] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems,* Math. Comput. **37** (1981), 105-126.

[5] ——, *A flexible inner-outer preconditioned GMRES algorithm,* SIAM J. Sci. Statist. Comput. **14** (1993), 461-469.

[6] Y. Saad and M. H. Schultz, *GMRES : a generalized minimum residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Statist. Comput. **7** (1986), 856-869.

[7] H. A. van der Vorst, *BiCGSTAB: A fast and smoothly converging variant of BiCG for the solution of non-symmetric linear systems,* SIAM J. Sci. Statist. Comput. **12** (1992), 631-644.

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 361-763, Korea
E-mail: gmjae@cbucc.chungbuk.ac.kr