Journal of Advanced Research in Scientific Computing

Vol. 3, Issue. 4, 2012, pp. 57-97
Online ISSN: 1943-2364

A hybrid preconditioner to solve nonsymmetric linear system of equations

Amin Rafiei*

Department of Applied Mathematics, Sabzevar Tarbiat Moallem University, Sabzevar, Iran.

Abstract. In this paper, modified left and right-looking versions of SAINV-Ns (Stabilized Approximate Inverse for Nonsymmetric matrices) preconditioner have been presented. Structural modification of prototype versions has been done in such a way that it is possible to make both versions of this preconditioner more robust by using inverse-based dropping technique. Only the information of matrix A is used to generate this preconditioner. To decrease preconditioning time of the right-looking version, matrix A is stored in Compressed Sparse Column format. To construct this version, row traversal of matrix A is also needed. To avoid storing matrix A in Compressed Sparse Row format, the linked list trick has been used. By using some nonsymmetric matrices, modified left and right-looking versions of this preconditioner have been compared. The multilevel nested dissection reordering has also been used as the preprocessing.

Keywords: A-biconjugation process; Left-looking SAINV-Ns; Right-looking SAINV-Ns; Inverse-based dropping technique.

Mathematics Subject Classification 2010: 65F10, 65F08.

1 Introduction

An explicit preconditioner M for the linear system of equations

$$Ax = b, \tag{1.1}$$

is an approximation of A^{-1}. This preconditioner will change the linear system (1.1) to the right or left preconditioned systems

$$MAx = Mb, \qquad AMu = b, \; Mu = x. \tag{1.2}$$

*Correspondence to: Amin Rafiei, Department of Applied Mathematics, Sabzevar Tarbiat Moallem University, Sabzevar, Iran. Email: `rafiei.am@gmail.com, rafiei@sttu.ac.ir`
†Received: 27 July 2011, accepted: 31 October 2011.

To solve the preconditioned systems (1.2) with Krylov subspace methods, matrix-vector multiplication will be needed during each inner step.

An implicit preconditioner M^{-1} for the linear system (1.1) is an approximation of A. This preconditioner will change the original system (1.1) to the right or left preconditioned systems

$$M^{-1}Ax = M^{-1}b, \qquad AM^{-1}u = b,\ M^{-1}u = x. \tag{1.3}$$

To solve the preconditioned systems (1.3) with Krylov subspace methods, two different systems should be solved with forward and backward substitution methods during each inner step.

One can consider a hybrid preconditioner as an approximation of A and A^{-1}. Suppose this preconditioner converts the original system (1.1) to the preconditioned system. To solve the preconditioned system with Krylov subspace methods, we need matrix-vector multiplication and solving one backward (forward) system during each inner step.

In [10], we presented a hybrid preconditioner for nonsymmetric matrices which we termed it SAINV-Ns and is breakdown-free for positive definite matrices. This preconditioner comes out of the A-biconjugation process. Thus, it has left and right-looking versions. The SAINV-Ns preconditioner has two unit triangular and one diagonal factors. The computations of the triangular factors are interlaced together. This preconditioner has also the ability to be an explicit preconditioner. In [10], we only proposed the prototype version of the right-looking version of this preconditioner and we implemented it in a naive way. In this paper, we propose the prototype form of the left-looking version of this preconditioner. We also present modified forms of the left and right-looking versions which are structurally different from prototype versions. We also implement both modified left and right-looking versions efficiently. In this implementation, we have used the inverse-based dropping technique and the linked lists trick.

In this paper, we use $X_{i,:}$ and $X_{:,i}$ as the notations for the i-th row and the i-th column of an arbitrary matrix X, respectively. We also use $X_{i_1:i_2,j_1:j_2}$ as the notation of a submatrix of X where row indices are between i_1 and i_2 and column indices are between j_1 and j_2.

In Section 2, at first the mathematical basis of the prototype form of the right-looking version of SAINV-Ns preconditioner will be considered. Then, we will explain the main reason of why we need to modify this form and the modified form of the right-looking version will be presented. All our considerations in Section 2, will be generalized for the left-looking version of SAINV-Ns preconditioner in Section 3. In Section 4, we have exploited Krylov subspace methods and some nonsymmetric test matrices to compare the modified form of the left and right-looking versions of SAINV-Ns preconditioner. In Section 5, conclusions will be presented.

2 Right-looking version of SAINV-Ns preconditioner

Suppose that matrix A has the inverse factorization

$$A^{-1} = ZD^{-1}W^T, \tag{2.1}$$

in which Z and W are unit upper triangular matrices and D is a diagonal one. If in Algorithm 1, no dropping will be applied, then the factorization (2.1) will be computed. Otherwise, the approximate inverse factorization

$$A^{-1} \approx ZD^{-1}W^T, \tag{2.2}$$

will be computed. Structure of this algorithm is as the following [1,3]:

Algorithm 1 (right-looking version of the A-biconjugation algorithm)

For $i = 1, 2, ..., n$ Do:
  $w_i^{(0)} = e_i, \quad z_i^{(0)} = e_i.$
End Do
For $i = 1, 2, ..., n$ Do:
  $v_i = Ae_i, \quad u_i = A^T e_i$ {not positive definite}
  $v_i = Aw_i^{(i-1)}, \quad u_i = A^T z_i^{(i-1)}$ {positive definite}
  $p_i^{(i-1)} = (w_i^{(i-1)})^T v_i, \quad q_i^{(i-1)} = (z_i^{(i-1)})^T u_i$
  For $j = i+1, ..., n$ Do:
    $q_j^{(i-1)} = (w_j^{(i-1)})^T v_i, \quad p_j^{(i-1)} = (z_j^{(i-1)})^T u_i$
    apply a dropping rule to $q_j^{(i-1)}$ *and to* $p_j^{(i-1)}$
    $w_j^{(i)} = w_j^{(i-1)} - (\frac{q_j^{(i-1)}}{q_i^{(i-1)}})w_i^{(i-1)}, \quad z_j^{(i)} = z_j^{(i-1)} - (\frac{p_j^{(i-1)}}{p_i^{(i-1)}})z_i^{(i-1)}$
    for all $l \leq i$ *apply a dropping rule to* $w_{lj}^{(i)}$ *and to* $z_{lj}^{(i)}$
  End Do
End Do
Let $w_i = w_i^{(i-1)}$, $z_i = z_i^{(i-1)}$ and $d_{ii} = p_i^{(i-1)}$, for $1 \leq i \leq n$.
Return $W = [w_1, \ldots, w_n]$, $D = (d_{ii})$ and $Z = [z_1, \ldots, z_n]$.

The next proposition will allow us to interlace the computation of the parameters $p_j^{(i-1)}$ and $q_j^{(i-1)}$ in this algorithm.

Proposition 2.1. Parameters $p_j^{(i-1)}$ and $q_j^{(i-1)}$ obtained from Algorithm 1, satisfy the following relations for $1 \leq i \leq j \leq n$:

$$p_j^{(i-1)} = a_{ij} - \sum_{k=1}^{i-1} \frac{p_j^{(k-1)}}{p_k^{(k-1)}} q_i^{(k-1)}, \qquad q_j^{(i-1)} = a_{ji} - \sum_{k=1}^{i-1} \frac{q_j^{(k-1)}}{q_k^{(k-1)}} p_i^{(k-1)}. \tag{2.3}$$

Proof. See [10]. □

Besides having the factorization (2.1), suppose that matrix A has the factorization

$$A = LDU, \tag{2.4}$$

in which L^T and U are unit upper triangular matrices and D is a diagonal one. Since this factorization is unique, from relations (2.1) and (2.4) one can conclude that

$$Z = U^{-1}, \qquad W^T = L^{-1}. \tag{2.5}$$

Therefore, entries of matrix U can be computed as [2, 10]:

$$U_{ij} = (D^{-1}W^T A)_{ij} = \frac{p_j^{(i-1)}}{p_i^{(i-1)}}. \tag{2.6}$$

Relation (2.6) converts relation (2.3) to the following form:

$$p_j^{(i-1)} = a_{ij} - \sum_{k=1}^{i-1} U_{kj} q_i^{(k-1)}. \tag{2.7}$$

The main motivation of the prototype form of the right-looking version of SAINV-Ns preconditioner was on removing the effect of matrix A^T and not computing the Z factor in Algorithm 1. In [10], we presented an algorithm that computes the factorization

$$A^{-1} = U^{-1}D^{-1}W^T, \tag{2.8}$$

and exploits relation (2.7) to compute $p_j^{(i-1)}$ parameters. The algorithm is in the form:

Algorithm 2 (Prototype version of the right-looking SAINV-Ns)

1. $w_i^{(0)} = e_i, \quad 1 \le i \le n.$
2. For $i = 1, 2, ..., n$ Do:
3. $\quad v_i = Ae_i$ {not positive definite}
4. $\quad v_i = Aw_i^{(i-1)}$ {positive definite}
5. $\quad p_i^{(i-1)} = q_i^{(i-1)} = (w_i^{(i-1)})^T v_i$
6. $\quad$ For $j = i+1, ..., n$ Do:
7. $\quad\quad q_j^{(i-1)} = (w_j^{(i-1)})^T v_i$
8. $\quad\quad$ *apply a dropping rule to* $q_j^{(i-1)}$
9. $\quad\quad p_j^{(i-1)} = a_{ij} - \sum_{k=1}^{i-1} U_{kj} q_i^{(k-1)}$
10. $\quad\quad w_j^{(i)} = w_j^{(i-1)} - (\frac{q_j^{(i-1)}}{q_i^{(i-1)}}) w_i^{(i-1)}$
11. $\quad\quad$ *for all* $l \le i$ *apply a dropping rule to* $w_{lj}^{(i)}$
12. $\quad\quad U_{ij} = \frac{p_j^{(i-1)}}{p_i^{(i-1)}}$
13. $\quad\quad$ *apply a dropping rule to* U_{ij}

14. End Do
15. End Do
16. Let $w_i = w_i^{(i-1)}$ and $d_{ii} = p_i^{(i-1)}$, for $1 \leq i \leq n$.
17. Return $W = [w_1, \ldots, w_n]$, $D = (d_{ii})$ and $U = (U_{ij})$.

If in Algorithm 2 dropping will be applied, then the approximate factorization

$$A^{-1} \approx U^{-1} D^{-1} W^T, \tag{2.9}$$

will be computed.

At step i of Algorithm 2, the $U_{i,:}$ is computed. Thus, one can use Compressed Sparse Row format of storage [12] for this matrix. But at this step, the submatrix $U_{1:i-1,i+1:n}$ should be traversed column-wise. Suppose we define entries of the lower triangular matrix Q as:

$$Q_{ji} = q_j^{(i-1)}, \qquad j \geq i. \tag{2.10}$$

At step i of Algorithm 2, the $Q_{:,i}$ is computed. Thus, one can use Compressed Sparse Column format of storage [12] for matrix Q. But at this step, to compute parameters $p_j^{(i-1)}$, row traversal of $Q_{i,:}$ is needed.

To compute lines 3 and 4 of Algorithm 2 in sparse-sparse mode, the matrix A should be stored in Compressed Sparse Column format. But we need to access $A_{i,i+1:n}$ in each step i of this algorithm.

Therefore in Algorithm 2, matrices A and Q are stored in Compressed Sparse Column format, but these two matrices are traversed row-wise. Furthermore, matrix U is stored in Compressed Sparse Row format and is accessed column-wise.

Suppose we swap k and j loops in Algorithm 2. The new following algorithm will be obtained:

Algorithm 3 (**Modified version of the right-looking SAINV-Ns**)
1. $w_i^{(0)} = e_i, \quad 1 \leq i \leq n.$
2. For $i = 1, 2, ..., n$ Do:
3. $v_i = Ae_i$ {not positive definite}
4. $v_i = Aw_i^{(i-1)}$ {positive definite}
5. $p_i^{(i-1)} = q_i^{(i-1)} = (w_i^{(i-1)})^T v_i$
6. $d_{ii} = q_i^{(i-1)} \;\; or \;\; p_i^{(i-1)}$
7. For $j = i+1, ..., n$ Do:
8. $q_j^{(i-1)} = (w_j^{(i-1)})^T v_i$
9. *apply a dropping rule to* $q_j^{(i-1)}$
10. $w_j^{(i)} = w_j^{(i-1)} - (\frac{q_j^{(i-1)}}{q_i^{(i-1)}}) w_i^{(i-1)}$
11. *for all* $l \leq i$ *apply a dropping rule to* $w_{lj}^{(i)}$
12. End Do

13. $p_j^{(i-1)} = a_{ij}, \quad i+1 \leq j \leq n.$
14. For $k = 1, ...i-1$ Do:
15. For $j = i+1, ...n$ Do:
16. $p_j^{(i-1)} = p_j^{(i-1)} - U_{kj} q_i^{(k-1)}$
17. End Do:
18. End Do:
19. $U_{ij} = \frac{p_j^{(i-1)}}{p_i^{(i-1)}}, \quad i+1 \leq j \leq n.$
20. *apply a dropping rule to* U_{ij}
21. End Do
22. Let $w_i = w_i^{(i-1)}$ and $d_{ii} = p_i^{(i-1)}$, for $1 \leq i \leq n$.
23. Return $W = [w_1, \ldots, w_n]$, $D = (d_{ii})$ and $U = (U_{ij})$.

In Algorithm 3, matrices A and Q are stored in Compressed Sparse Column format and we use the linked lists trick to access them row-wise [9]. In this algorithm, matrix U is stored and accessed row-wise. This will give the ability to use the inverse-based dropping technique to drop entries of this matrix. Suppose that ε_U is the drop tolerance parameter for matrix U. We drop entry U_{ij} of this matrix when the dropping criterion

$$|U_{ij}| \|U^{-1} e_i\|_\infty \leq \varepsilon_U, \tag{2.11}$$

is satisfied [4,11]. This is the safest way of dropping. Since matrix U is computed row-wise, it is possible to compute an approximation of $\|e_i^T U^{-T}\|_\infty$ which can be viewed as an approximation of $\|U^{-1} e_i\|_\infty$ by the following formula:

$$\|U^{-1} e_i\|_\infty \leq \|U^{-1} e_i\|_1 = \|e_i^T U^{-T}\|_\infty. \tag{2.12}$$

The approximation of $\|e_i^T U^{-T}\|_\infty$ is computed by the adaptations of the condition estimator algorithm [5,7].

We select $\varepsilon_{Q,W}$ as the drop tolerance parameter for entries of both Q and W matrices in Algorithm 3. Entry $w_{lj}^{(i)}$ is dropped when the criterion

$$|w_{lj}^{(i)}| \leq \varepsilon_{Q,W}, \tag{2.13}$$

is satisfied. With respect to what has been surveyed in [4,11], we use the criterion

$$|(\frac{q_j^{(i-1)}}{q_i^{(i-1)}})| \|W_{:,i}\|_\infty \leq \varepsilon_{Q,W}, \tag{2.14}$$

to drop entry $q_j^{(i-1)}$. This type of dropping is also an inverse-based dropping technique.

3 Left-looking version of SAINV-Ns preconditioner

There is also the left-looking version of the A-biconjugation algorithm [1,3]. This algorithm is as the following:

Algorithm 4 (left-looking version of the A-biconjugation algorithm)

1. For $i = 1, ..., n$ Do:
2. $w_i^{(0)} = e_i, \quad z_i^{(0)} = e_i.$
3. For $j = 1, ..., i-1$ Do:
4. $q_i^{(j-1)} = (w_i^{(j-1)})^T A e_j, \quad p_i^{(j-1)} = (z_i^{(j-1)})^T A^T e_j$
5. *apply a dropping rule to* $q_i^{(j-1)}$ *and to* $p_i^{(j-1)}$
6. $w_i^{(j)} = w_i^{(j-1)} - (\frac{q_i^{(j-1)}}{q_j^{(j-1)}}) w_j^{(j-1)}, \quad z_i^{(j)} = z_i^{(j-1)} - (\frac{p_i^{(j-1)}}{p_j^{(j-1)}}) z_j^{(j-1)}$
7. *for all* $l \le j$ *apply a dropping rule to* $w_{li}^{(j)}$ *and to* $z_{li}^{(j)}$
8. End Do
9. $q_i^{(i-1)} = (w_i^{(i-1)})^T A e_i, \quad p_i^{(i-1)} = (z_i^{(i-1)})^T A^T e_i$ {not positive definite}
10. $q_i^{(i-1)} = (w_i^{(i-1)})^T A w_i^{(i-1)}, \quad p_i^{(i-1)} = (z_i^{(i-1)})^T A^T z_i^{(i-1)}$ {positive definite}
11. End Do
12. Let $w_i = w_i^{(i-1)}$, $z_i = z_i^{(i-1)}$ and $d_{ii} = p_i^{(i-1)}$, for $1 \le i \le n$.
13. Return $W = [w_1, \ldots, w_n]$, $D = (d_{ii})$ and $Z = [z_1, \ldots, z_n]$.

When no dropping is applied, Algorithms 1 and 4 compute the same Z, D and W matrices. But the computations are done in different ways. Proposition 2.1 prepares the opportunity to compute parameter $p_i^{(j-1)}$ of Algorithm 4 in the form:

$$p_i^{(j-1)} = a_{ji} - \sum_{k=1}^{j-1} (\frac{p_i^{(k-1)}}{p_k^{(k-1)}}) q_j^{(k-1)} = a_{ji} - \sum_{k=1}^{j-1} U_{ki} q_j^{(k-1)}. \tag{3.1}$$

By using parameters $p_i^{(j-1)}$ and $p_j^{(j-1)}$ in Algorithm 4, one can compute entry U_{ji} of matrix U by the relation

$$U_{ji} = \frac{p_i^{(j-1)}}{p_j^{(j-1)}}. \tag{3.2}$$

Suppose we remove effect of matrix A^T in Algorithm 4. This means that the matrix Z is not computed any more and all the related computations to this matrix will be removed. If in this case, we compute parameter $p_i^{(j-1)}$ and entry U_{ji} by relations (3.1) and (3.2), respectively, then the new following algorithm will be presented:

Algorithm 5 (Prototype version of the left-looking SAINV-Ns)

1. For $i = 1, ..., n$ Do:
2. $w_i^{(0)} = e_i$
3. For $j = 1, ..., i-1$ Do:
4. $q_i^{(j-1)} = (w_i^{(j-1)})^T A e_j$
5. *apply a dropping rule to* $q_i^{(j-1)}$

6. $w_i^{(j)} = w_i^{(j-1)} - (\frac{q_i^{(j-1)}}{q_j^{(j-1)}})w_j^{(j-1)}$
7. *for all* $l \leq j$ *apply a dropping rule to* $w_{li}^{(j)}$
8. $p_i^{(j-1)} = a_{ji} - \sum_{k=1}^{j-1} U_{ki} q_j^{(k-1)}$
9. $U_{ji} = \frac{p_i^{(j-1)}}{p_j^{(j-1)}}$
10. *apply a dropping rule to* U_{ji}
11. End Do
12. $p_i^{(i-1)} = q_i^{(i-1)} = (w_i^{(i-1)})^T A e_i$ {not positive definite}
13. $p_i^{(i-1)} = q_i^{(i-1)} = (w_i^{(i-1)})^T A w_i^{(i-1)}$ {positive definite}
14. End Do
15. Let $w_i = w_i^{(i-1)}$ and $d_{ii} = p_i^{(i-1)}$, for $1 \leq i \leq n$.
16. Return $W = [w_1, \ldots, w_n]$, $D = (d_{ii})$ and $U = (U_{ij})$.

When no dropping is applied, Algorithms 3 and 5 compute the same U, D and W matrices. Matrix U is computed column-wise in Algorithm 5. We replace the computation format of matrix U in this algorithm by the computation format of this matrix from Algorithm 3. A new algorithm will be presented which is in the form:

Algorithm 6 (Modified version of the left-looking SAINV-Ns)

1. For $i = 1, ..., n$ Do:
2. $w_i^{(0)} = e_i$
3. For $j = 1, ..., i-1$ Do:
4. $q_i^{(j-1)} = (w_i^{(j-1)})^T A e_j$
5. *apply a dropping rule to* $q_i^{(j-1)}$
6. $w_i^{(j)} = w_i^{(j-1)} - (\frac{q_i^{(j-1)}}{q_j^{(j-1)}})w_j^{(j-1)}$
7. *for all* $l \leq j$ *apply a dropping rule to* $w_{li}^{(j)}$
8. End Do
9. $p_i^{(i-1)} = q_i^{(i-1)} = (w_i^{(i-1)})^T A e_i$ {not positive definite}
10. $p_i^{(i-1)} = q_i^{(i-1)} = (w_i^{(i-1)})^T A w_i^{(i-1)}$ {positive definite}
11. $p_j^{(i-1)} = a_{ij}, \quad i+1 \leq j \leq n.$
12. For $k = 1, ..., i-1$ Do:
13. For $j = i+1, ..., n$ Do:
14. $p_j^{(i-1)} = p_j^{(i-1)} - U_{kj} q_i^{(k-1)}$
15. End Do
16. End Do
17. $U_{ij} = \frac{p_j^{(i-1)}}{p_i^{(i-1)}}, \quad i+1 \leq j \leq n.$
18. *apply a dropping rule to* U_{ij}
19. End Do
20. Let $w_i = w_i^{(i-1)}$ and $d_{ii} = p_i^{(i-1)}$, for $1 \leq i \leq n$.

21. Return $W = [w_1, \ldots, w_n]$, $D = (d_{ii})$ and $U = (U_{ij})$.

In Algorithm 6, matrix U is computed row-wise. This will give the opportunity to use the criterion

$$|U_{ji}| \|U^{-1} e_j\|_\infty \leq \varepsilon_U, \tag{3.3}$$

to drop entry U_{ji} [11]. Since the whole matrix U is not available, then an approximation of $\|U^{-1}e_j\|_\infty$ is being used. We notice again that an approximation of $\|e_j^T U^{-T}\|_\infty$ is used as an upper bound for $\|U^{-1}e_j\|_\infty$.

Suppose that $\varepsilon_{Q,W}$ is the same drop tolerance parameter for both Q and W matrices. We use criterion

$$|w_{li}^{(j)}| \leq \varepsilon_{Q,W}, \tag{3.4}$$

to drop entry $w_{li}^{(j)}$ and the criterion

$$|(\frac{q_i^{(j-1)}}{q_j^{(j-1)}})| \|W_{:,j}\|_\infty \leq \varepsilon_{Q,W}, \tag{3.5}$$

to drop entry $q_i^{(j-1)}$ [11].

4 Numerical experiments

In this section we have used modified left and right-looking versions of SAINV-Ns as the right preconditioner to solve the linear system of equations with Bicgstab, GMRES(30) and TFQMR [12] methods. The codes were written in Fortran and were compiled with *ifort* Intel compiler. All the experiments were done on a machine with one quad Intel(R) CPU and 8 GB of RAM memory. We have used 106 nonsymmetric matrices of the collection [6] as the test matrices. For all the experiments, the initial and the right hand side vectors of the $Ax = b$ system, are the zero vector and $b = Ae$ where $e = [1, \cdots, 1]^T$. For all the experiments, the stopping criterion is:

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq 10^{-10},$$

where r_k and r_0 are the k-th and the initial residuals of the system.

Table 1: nonsymmetric test matrices

Group/Matrix	n	nnz	sym
$Schenk\_IBMSDS/3D\_51448\_3D$	51448	537038	99%
$Bai/af23560$	23560	460598	99%
$Engwirda/airfoil\_2d$	14214	259688	100%
$HB/arc130$	130	1037	98%
$Bourchtein/atmosmodd$	1270432	8814880	76%
$Bourchtein/atmosmodj$	1270432	8814880	100%
$Hamm/bcircuit$	68902	375558	100%
$VanHeukelum/cage9$	3534	41594	100%
$VanHeukelum/cage10$	11397	150645	100%
$VanHeukelum/cage11$	39082	559722	100%
$VanHeukelum/cage12$	130228	2032536	100%
$DRIVCAV/cavity05$	1182	32632	90%
$DRIVCAV/cavity10$	2597	76171	94%
$DRIVCAV/cavity11$	2597	71601	94%
$DRIVCAV/cavity12$	2597	76258	94%
$DRIVCAV/cavity13$	2597	71601	94%
$DRIVCAV/cavity16$	4562	137887	95%
$DRIVCAV/cavity17$	4562	131735	95%
$DRIVCAV/cavity18$	4562	138040	95%
$DRIVCAV/cavity19$	4562	131735	95%
$DRIVCAV/cavity20$	4562	138040	95%
hline $Lucifora/cell1$	7055	30082	100%
$Lucifora/cell2$	7055	30082	100%
$Muite/Chebyshev3$	4101	36879	50%
$Watson/chem\_master1$	40401	201201	100%
$Oberwolfach/chipcool0$	20082	281150	100%
$Oberwolfach/chipcool1$	20082	281150	100%
$Bomhof/Circuit\_1$	2624	35823	100%
$Bomhof/Circuit\_2$	4510	21199	81%
$Bomhof/Circuit\_3$	12127	48137	77%
$Bai/ck104$	104	992	83%
$Langemyr/comsol$	1500	97645	100%
$IBM\_Austin/coupled$	11341	97193	100%
$QLi/crashbasis$	160000	1750416	55%
$Bai/cryg10000$	10000	49699	100%
$IBM\_EDA/dc1$	116835	766396	85%
$IBM\_EDA/dc2$	116835	766396	85%
$IBM\_EDA/dc3$	116835	766396	85%
$Sanghavi/ecl32$	51993	380415	92%
$Averous/epb1$	14734	95053	73%

Table 2: nonsymmetric test matrices

Group/Matrix	n	nnz	sym
$Averous/epb2$	25228	175028	67%
$Averous/epb3$	84617	463625	67%
$FIDAP/ex24$	2283	47901	100%
$FIDAP/ex29$	2870	23754	100%
$FIDAP/ex31$	3909	91223	100%
$FIDAP/ex36$	3079	53099	100%
$FIDAP/ex37$	3565	67591	100%
$FIDAP/ex40$	7740	456188	100%
$Oberwolfach/flowmeter5$	9669	67391	100%
HB/fs_183_1	183	998	42%
HB/fs_183_6	183	1000	42%
$Garon/garon1$	3175	84723	100%
$HVDC/hvdc2$	189860	1339638	99%
$Schenk\_IBMSDS/ibm\_matrix\_2$	51448	537038	99%
$Hollinger/jan99jac040sc$	13694	72734	0%
$Hollinger/jan99jac100sc$	34454	190224	0%
$Hollinger/jan99jac120sc$	41374	229385	0%
$Mathworks/Kaufhold$	8765	42471	100%
$Norris/lung2$	109460	492564	0%
$QLi/majorbasis$	160000	1750416	55%
$Schenk\_IBMSDS/matrix-9$	103430	1250518	100%
$Schenk\_IBMSDS/matrix-new\_3$	125329	893984	99%
$Hamm/memplus$	17758	99147	100%
$Sandia/mult\_dcop\_01$	25187	193276	61%
$Sandia/mult\_dcop\_02$	25187	193276	61%
$Sandia/mult\_dcop\_03$	25187	193216	61%
$FEMLAB/ns3Da$	20414	1679599	100%
$Bai/olm5000$	5000	19996	67%
$FEMLAB/poisson3Db$	13514	352762	100%
$Grund/poli\_large$	15575	33033	0%
$LiuWenzhuo/powersim$	15838	64424	59%
$HB/psmigr\_3$	3140	543160	48%
$Rajat/Raj1$	263743	1300261	100%
$Rajat/rajat03$	7602	32653	100%
$Rajat/rajat15$	37261	443573	100%
$Rajat/rajat16$	94294	476766	99%
$Rajat/rajat18$	94294	479151	63%
$Rajat/rajat20$	86916	604299	99%
$Rajat/rajat22$	39899	195429	98%
$Rajat/rajat25$	87190	606489	99%

Table 3: nonsymmetric test matrices.

Group/Matrix	n	nnz	sym
$Rajat/rajat27$	20640	97353	96%
$Rajat/rajat28$	87190	606489	99%
$Rajat/rajat31$	4690002	20316253	100%
$Bova/rma10$	46835	2329092	100%
$Hamm/scircuit$	170998	958936	100%
$HB/sherman3$	5005	20033	100%
$Wang/swang1$	3169	20841	100%
$Wang/swang2$	3169	20841	100%
$Brunetiere/thermal$	3456	66528	100%
$CEMW/tmt\_unsym$	917825	4584801	100%
$MathWorks/tomography$	500	28726	100%
$Norris/torso2$	115967	1033473	99%
$Norris/torso3$	259156	4429042	85%
$IBM\_EDA/trans4$	116835	749800	85%
$IBM\_EDA/trans5$	116835	749800	85%
$TSOPF/TSOPF\_RS\_b39\_c7$	14098	252446	6%
$TSOPF/TSOPF\_RS\_b39\_c19$	38098	684206	6%
$TSOPF/TSOPF\_RS\_b39\_c30$	60098	1079986	6%
$TOKAMAK/utm5940$	5940	83842	53%
$Simon/venkat01$	62424	1717792	100%
$Simon/venkat25$	62624	1717763	100%
$Simon/venkat50$	62424	1717777	100%
$Quaglino/viscoplastic2$	32769	381326	57%
$Wang/wang3$	26064	177168	100%
$Wang/wang4$	26068	177196	100%
$Zhao/Zhao1$	33861	166453	92%

In Tables 1-3, column $Group/Matrix$ contains the name of the group in which the matrix belongs to and also the name of the matrix. In these tables, n and nnz are the dimension and the number of nonzero entries of the matrix, respectively. Column sym is the percentage of the symmetric nonzero pattern of the matrix. In column sym, 100% means that the matrix has symmetric nonzero pattern. Otherwise the nonzero pattern of the matrix is not symmetric. If the matrix has symmetric nonzero pattern, then we have reordered the matrix with MLND reordering [8]. Otherwise we have used this reordering on $A + A^T$. With respect to the information provided by [6], all the test matrices are only nonsymmetric and not positive definite.

In Tables 4-6, the notation + means that the stopping criterion has not been satisfied after 5000 number of iterations. In these tables, the columns It and Sec_{ite} are the number of iterations of the Krylov subspace methods and the iteration time, respectively.

Table 4: results of iterative methods without preconditioning

Method	Bicgstab		GMRES(30)		TFQMR	
	It	Sec_{ite}	It	Sec_{ite}	It	Sec_{ite}
$3D\_51448\_3D$	658	3.37	+	+	+	+
$af23560$	+	+	+	+	4823	9.24
$airfoil\_2d$	+	+	+	+	+	+
$arc130$	23	10^{-10}	11	10^{-10}	21	10^{-10}
$atmosmodd$	625	45.59	1974	306.5	1001	81.0
$atmosmodj$	629	45.82	3901	606.5	833	67.64
$bcircuit$	+	+	+	+	+	+
$cage9$	39	10^{-10}	32	10^{-10}	39	0.02
$cage10$	29	0.02	28	0.02	29	0.01
$cage11$	39	0.11	30	0.119	31	0.09
$cage12$	35	0.389	29	0.409	29	0.339
$cavity05$	1369	0.109	+	+	+	+
$cavity10$	2343	0.43	+	+	+	+
$cavity11$	+	+	+	+	+	+
$cavity12$	+	+	+	+	+	+
$cavity13$	+	+	+	+	+	+
$cavity16$	2665	0.85	+	+	+	+
$cavity17$	+	+	+	+	+	+
$cavity18$	+	+	+	+	+	+
$cavity19$	+	+	+	+	+	+
$cavity20$	+	+	+	+	+	+
$cell1$	1387	0.189	+	+	+	+
$cell2$	+	+	+	+	+	+
$Chebyshev3$	+	+	+	+	+	+
$chem\_master1$	1033	0.819	+	+	+	+
$chipcool0$	+	+	+	+	+	+
$chipcool1$	+	+	+	+	+	+
$Circuit\_1$	1463	0.129	661	0.09	327	0.039
$Circuit\_2$	973	0.089	3706	0.579	+	+
$Circuit\_3$	+	+	+	+	+	+
$ck104$	215	10^{-10}	270	10^{-10}	133	10^{-10}
$comsol$	+	+	+	+	+	+
$coupled$	4081	1.86	+	+	2781	1.25
$crashbasis$	501	5.32	641	8.82	593	6.86
$cryg10000$	+	+	+	+	+	+
$dc1$	+	+	+	+	+	+
$dc2$	+	+	+	+	+	+
$dc3$	+	+	+	+	+	+
$ecl32$	+	+	+	+	+	+
$epb1$	1033	0.51	2643	1.82	1559	0.809

Table 5: results of iterative methods without preconditioning

Method	Bicgstab		GMRES(30)		TFQMR	
	It	Sec_{ite}	It	Sec_{ite}	It	Sec_{ite}
$epb2$	847	0.68	1549	1.98	681	0.58
$epb3$	+	+	+	+	+	+
$ex24$	+	+	+	+	+	+
$ex29$	141	0.01	151	0.02	145	0.01
$ex31$	+	+	+	+	+	+
$ex36$	+	+	+	+	+	+
$ex37$	123	0.029	165	0.04	113	0.02
$ex40$	+	+	+	+	+	+
$flowmeter5$	+	+	+	+	+	+
fs_183_1	+	+	+	+	575	0.01
fs_183_6	1289	0.01	+	+	853	0.01
$garon1$	+	+	+	+	+	+
$hvdc2$	+	+	+	+	+	+
ibm_matrix_2	700	3.57	+	+	+	+
$jan99jac040sc$	+	+	+	+	+	+
$jan99jac100sc$	+	+	+	+	+	+
$jan99jac120sc$	+	+	+	+	+	+
$Kaufhold$	+	+	+	+	+	+
$lung2$	+	+	+	+	+	+
$majorbasis$	+	+	+	+	+	+
$matrix-9$	3341	34.52	+	+	2747	29.5
$matrix-new\_3$	+	+	+	+	+	+
$memplus$	2899	1.51	+	+	1933	1.02
$mult\_dcop\_01$	+	+	+	+	+	+
$mult\_dcop\_02$	+	+	+	+	+	+
$mult\_dcop\_03$	+	+	+	+	+	+
$ns3Da$	+	+	2936	25.53	1573	13.07
$olm5000$	+	+	+	+	+	+
$poisson3Db$	513	7.53	1019	16.57	733	10.76
$poli\_large$	45	0.01	61	0.04	51	0.2
$powersim$	+	+	+	+	+	+
$psmigr\_3$	21	0.029	14	0.029	23	0.04
$Raj1$	+	+	+	+	+	+
$rajat03$	2457	0.319	+	+	+	
$rajat15$	+	+	+	+	+	+
$rajat16$	+	+	+	+	+	+
$rajat18$	+	+	+	+	+	+
$rajat20$	+	+	+	+	+	+
$rajat22$	+	+	+	+	+	+
$rajat25$	+	+	+	+	+	+

Table 6: results of iterative methods without preconditioning

Method	Bicgstab		GMRES(30)		TFQMR	
	It	Sec_{ite}	It	Sec_{ite}	It	Sec_{ite}
$rajat27$	+	+	+	+	+	+
$rajat28$	+	+	+	+	+	+
$rajat31$	+	+	+	+	+	+
$rma10$	+	+	+	+	+	+
$scircuit$	+	+	+	+	+	+
$sherman3$	+	+	+	+	+	+
$swang1$	37	10^{-10}	34	10^{-10}	37	0.01
$swang2$	65	10^{-10}	71	0.01	77	0.01
$thermal$	41	0.01	37	0.02	55	0.03
tmt_unsym	+	+	+	+	+	+
$tomography$	+	+	1321	0.1	299	0.02
$torso2$	79	0.529	63	0.629	89	0.639
$torso3$	409	9.81	340	10.31	329	8.53
$trans4$	+	+	+	+	+	+
$trans5$	+	+	+	+	+	+
$TSOPF\_RS\_b39\_c7$	+	+	+	+	+	+
$TSOPF\_RS\_b39\_c19$	+	+	+	+	+	+
$TSOPF\_RS\_b39\_c30$	+	+	+	+	+	+
$utm5940$	+	+	+	+	+	+
$venkat01$	+	+	+	+	+	+
$venkat25$	+	+	+	+	+	+
$venkat50$	+	+	+	+	+	+
$viscoplastic2$	+	+	+	+	+	+
$wang3$	+	+	+	+	+	+
$wang4$	+	+	+	+	+	+
$Zhao1$	97	0.09	49	0.069	51	0.04

In Tables 7-9 the density of both left and right-looking versions are defined by:

$$density = \frac{nnz(U) + nnz(W) + n}{nnz(A)},$$

in which $nnz(U)$ and $nnz(W)$ are the number of nonzero entries of the U and W matrices. In these tables, Sec_p is the preconditioning time. $\varepsilon_{Q,W}$ is the drop tolerance parameter for both Q and W matrices and ε_U is the drop tolerance parameter for matrix U.

Table 7: Properties of the preconditioners.

Method	left-looking SAINV-Ns				right-looking SAINV-Ns			
	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$
$3D\_51448\_3D$	0.01	0.01	0.659	1.02	0.01	0.01	285.9	13.77
$af23560$	0.01	0.01	3.01	71	0.01	0.01	375.01	53.9
$airfoil\_2d$	0.01	0.01	0.17	95	0.01	0.01	31.52	25.33
$arc130$	0.01	0.01	10^{-15}	0.4	0.01	0.01	0.28	0.8
$atmosmodd$	0.01	0.01	14.97	4.63	0.01	0.01	51.08	5.29
$atmosmodj$	0.01	0.01	15.22	4.63	0.01	0.01	51.29	5.29
$bcircuit$	0.01	0.01	0.129	1.69	0.01	0.01	0.6	2.01
$cage9$	0.01	0.01	0.019	1.24	0.01	0.01	0.31	1.24
$cage10$	0.01	0.01	0.09	1.14	0.01	0.01	0.49	1.139
$cage11$	0.01	0.01	0.36	1.10	0.01	0.01	1.34	1.1
$cage12$	0.01	0.01	1.44	1.07	0.01	0.01	5.48	1.07
$cavity05$	0.01	0.01	0.02	1.54	0.01	0.01	0.399	4.23
$cavity10$	0.01	0.01	0.05	1.34	0.01	0.01	0.929	5.77
$cavity11$	0.01	0.01	0.08	1.43	0.01	0.01	0.86	5.31
$cavity12$	0.01	0.01	0.09	1.83	0.01	0.01	0.94	5.78
$cavity13$	0.01	0.01	0.099	1.97	0.01	0.01	0.84	5.32
$cavity16$	0.01	0.01	0.09	1.38	0.01	0.01	2.82	7.86
$cavity17$	0.01	0.01	0.09	1.36	0.01	0.01	2.6	7.31
$cavity18$	0.01	0.01	0.129	1.47	0.01	0.01	2.74	7.92
$cavity19$	0.01	0.01	0.119	1.48	0.01	0.01	2.49	7.3
$cavity20$	0.01	0.01	0.15	1.65	0.01	0.01	2.75	7.9
$cell1$	0.01	0.01	0.04	5.33	0.01	0.01	0.38	7.4
$cell2$	0.01	0.01	0.04	5.33	0.01	0.01	0.47	7.49
$Chebyshev3$	0.01	0.01	10^{-15}	1.85	0.01	0.01	0.299	1.87
$chem\_master1$	0.01	0.01	0.25	5.73	0.01	0.01	1.12	7.14
$chipcool0$	0.01	0.01	0.35	2.87	0.01	0.01	1.09	3.12
$chipcool1$	0.01	0.01	0.409	2.86	0.01	0.01	1.08	3.12
$Circuit\_1$	0.01	0.01	0.059	0.85	0.01	0.01	0.299	0.858
$Circuit\_2$	0.01	0.01	0.02	1.35	0.01	0.01	0.25	1.38
$Circuit\_3$	0.01	0.01	0.15	1.74	0.01	0.01	0.48	3.76
$ck104$	0.01	0.01	10^{-15}	1.03	0.01	0.01	0.219	1.39
$comsol$	0.01	0.01	0.25	2.33	0.01	0.01	0.429	1.68
$coupled$	0.01	0.01	0.45	1.76	0.01	0.01	0.77	1.9
$crashbasis$	0.01	0.01	0.89	2.1	0.01	0.01	2.52	2.18
$cryg10000$	0.01	0.01	0.05	4.63	0.01	0.01	3.25	29.65
$dc1$	0.01	0.01	64.24	0.978	0.01	0.01	76.03	1.05
$dc2$	0.01	0.01	64.97	0.965	0.01	0.01	76.92	1.04
$dc3$	0.01	0.01	65.04	0.965	0.01	0.01	76.7	1.04
$ecl32$	0.01	0.01	5.86	9.69	0.01	0.01	986.1	127.39
$epb1$	0.01	0.01	0.119	5.03	0.01	0.01	0.589	6.41

Table 8: Properties of the preconditioners.

Method	left-looking SAINV-Ns				right-looking SAINV-Ns			
	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$
$epb2$	0.01	0.01	0.229	4.62	0.01	0.01	28.11	32.56
$epb3$	0.01	0.01	0.709	5.44	0.01	0.01	107.79	45.54
$ex24$	0.01	0.01	0.22	5.29	0.01	0.01	0.529	5.61
$ex29$	0.01	0.01	0.01	1.75	0.01	0.01	0.24	1.75
$ex31$	0.01	0.01	0.489	3.29	0.01	0.01	0.959	3.87
$ex36$	0.01	0.01	0.09	2.81	0.01	0.01	0.58	5.5
$ex37$	0.01	0.01	0.019	0.699	0.01	0.01	0.26	0.705
$ex40$	0.01	0.01	0.699	1.14	0.01	0.01	31.89	10.45
$flowmeter5$	0.01	0.01	0.08	4.65	0.01	0.01	0.46	5.68
fs_183_1	0.01	0.01	10^{-15}	1.22	0.01	0.01	0.21	1.18
fs_183_6	0.01	0.01	10^{-15}	1.15	0.01	0.01	0.219	1.16
$garon1$	0.01	0.01	0.09	1.62	0.01	0.01	2.36	9.72
$hvdc2$	0.01	0.01	1.26	3.48	0.01	0.01	9.17	6.39
ibm_matrix_2	0.01	0.01	0.679	1.03	0.01	0.01	152.9	8.6
$jan99jac040sc$	0.01	0.01	1.6	12.62	0.01	0.01	42.31	75.31
$jan99jac100sc$	0.01	0.01	5.93	15.51	0.01	0.01	233.02	110.95
$jan99jac120sc$	0.01	0.01	7.89	16.41	0.01	0.01	263.18	133.79
$Kaufhold$	0.01	0.01	10^{-15}	0.472	0.01	0.01	0.24	0.47
$lung2$	0.01	0.01	0.109	1.63	0.01	0.01	1.07	3.95
$majorbasis$	0.01	0.01	0.57	1.38	0.01	0.01	635.27	26.57
$matrix-9$	0.01	0.01	1.57	1.66	0.01	0.01	10465.8	88.77
$matrix-new\_3$	0.01	0.01	0.9	0.573	0.01	0.01	1095.15	16.86
$memplus$	0.01	0.01	0.039	0.668	0.01	0.01	0.32	0.781
$mult\_dcop\_01$	0.01	0.01	3.93	7.72	0.01	0.01	10.37	15.13
$mult\_dcop\_02$	0.01	0.01	1.95	1.28	0.01	0.01	5.71	9.51
$mult\_dcop\_03$	0.01	0.01	2.22	0.785	0.01	0.01	6.23	11.17
$ns3Da$	0.01	0.01	15.38	2.81	0.01	0.01	874.11	17.38
$olm5000$	0.01	0.01	0.01	7.08	0.01	0.01	0.3	7.25
$poisson3Db$	0.01	0.01	2.93	1.78	0.01	0.01	6.85	1.9
$poli\_large$	0.01	0.01	0.02	2.8	0.01	0.01	0.28	3.45
$powersim$	0.01	0.01	0.02	2.79	0.01	0.01	0.32	3.3
$psmigr\_3$	0.01	0.01	0.129	0.97	0.01	0.01	14.01	2.51
$Raj1$	0.01	0.01	13.51	2.73	0.01	0.01	57.35	4.32
$rajat03$	0.01	0.01	0.029	2.82	0.01	0.01	0.29	2.41
$rajat15$	0.01	0.01	0.589	1.40	0.01	0.01	1.97	2.31
$rajat16$	0.01	0.01	6.82	1.35	0.01	0.01	7.25	1.00
$rajat18$	0.01	0.01	6.79	1.54	0.01	0.01	7.3	1.32
$rajat20$	0.01	0.01	7.55	1.61	0.01	0.01	8.12	1.47
$rajat22$	0.01	0.01	0.259	1.2	0.01	0.01	0.799	2.71
$rajat25$	0.01	0.01	7.32	1.51	0.01	0.01	7.97	1.45

Table 9: Properties of the preconditioners.

Method	left-looking SAINV-Ns				right-looking SAINV-Ns			
	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$
$rajat27$	0.01	0.01	0.899	1.99	0.01	0.01	0.389	2.35
$rajat28$	0.01	0.01	7.69	1.49	0.01	0.01	8.46	1.49
$rajat31$	0.01	0.01	15.63	1.81	0.01	0.01	187.92	1.95
$rma10$	0.01	0.01	9.57	2.94	0.01	0.01	277.22	15.24
$scircuit$	0.01	0.01	0.35	1.83	0.01	0.01	1.48	2.44
$sherman3$	0.01	0.01	0.02	4.25	0.01	0.01	0.34	5.03
$swang1$	0.01	0.01	10^{-15}	1.39	0.01	0.01	0.239	1.39
$swang2$	0.01	0.01	0.009	1.43	0.01	0.01	0.229	1.43
$thermal$	0.01	0.01	0.02	0.625	0.01	0.01	0.239	0.626
tmt_unsym	0.01	0.01	5.93	5.938	0.01	0.01	21.02	7.04
$tomography$	0.01	0.01	0.039	1.19	0.01	0.01	0.25	1.19
$torso2$	0.01	0.01	0.369	1.38	0.01	0.01	0.95	1.38
$torso3$	0.01	0.01	0.849	0.478	0.01	0.01	2.16	0.482
$trans4$	0.01	0.01	68.16	1.04	0.01	0.01	80.07	1.06
$trans5$	0.01	0.01	70.05	1.056	0.01	0.01	81.92	1.07
$TSOPF\_RS\_b39\_c7$	0.01	0.01	0.159	1.29	0.01	0.01	0.77	3.02
$TSOPF\_RS\_b39\_c19$	0.01	0.01	0.37	1.3	0.01	0.01	1.96	3.39
$TSOPF\_RS\_b39\_c30$	0.01	0.01	0.58	1.22	0.01	0.01	2.43	3.02
$utm5940$	0.01	0.01	0.66	8.03	0.01	0.01	8.3	25.44
$venkat01$	0.01	0.01	3.6	3.11	0.01	0.01	18.12	4.83
$venkat25$	0.01	0.01	8.71	5.1	0.01	0.01	693.17	37.79
$venkat50$	0.01	0.01	9.12	5.22	0.01	0.01	684.78	37.83
$viscoplastic2$	0.01	0.01	34.94	26.6	0.01	0.01	91.9	41.78
$wang3$	0.01	0.01	0.349	5.31	0.01	0.01	5.32	13.49
$wang4$	0.01	0.01	0.27	4.48	0.01	0.01	503.38	153.92
$Zhao1$	0.01	0.01	0.27	4.57	0.01	0.01	365.25	195.66

In Tables 10-18, results of Krylov subspace methods has been reported. In these tables, It is the number of iterations and Sec_{tot} is:

$$Sec_{tot} = Sec_p + Sec_{ite},$$

where Sec_{ite} is the iteration time of the method. For both left and right-looking versions of SAINV-Ns preconditioner, when the pivot entry is less than the machine precision, then we have replaced it by the square root of the machine precision.

Table 10: Bicgstab method with right preconditioner

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$3D\_51448\_3D$	127	2.15	+	+
$af23560$	71	4.28	+	+
$airfoil\_2d$	95	0.6	+	+
$arc130$	3	10^{-15}	+	+
$atmosmodd$	115	54.52	97	82.77
$atmosmodj$	133	60.84	111	87.75
$bcircuit$	599	4.1	213	2.03
$cage9$	9	1.99	9	0.33
$cage10$	9	0.109	11	0.51
$cage11$	11	0.459	11	5.76
$cage12$	11	1.77	11	0.28
$cavity05$	84	4.0	+	+
$cavity10$	149	0.1	+	+
$cavity11$	449	0.31	+	+
$cavity12$	1289	0.63	+	+
$cavity13$	2473	1.16	+	+
$cavity16$	207	0.269	+	+
$cavity17$	421	0.419	+	+
$cavity18$	773	0.819	+	+
$cavity19$	1327	1.25	+	+
$cavity20$	1661	1.68	+	+
$cell1$	43	7.00	33	0.4
$cell2$	59	6.99	23	0.48
$Chebyshev3$	+	+	+	+
$chem\_master1$	73	0.84	53	1.55
$chipcool0$	63	0.78	55	1.43
$chipcool1$	53	0.769	45	1.36
$Circuit\_1$	9	0.0599	+	+
$Circuit\_2$	11	0.02	157	0.27
$Circuit\_3$	875	0.599	+	+
$ck104$	11	10^{-15}	23	0.219
$comsol$	465	0.519	295	0.609
$coupled$	19	0.47	33	0.809
$crashbasis$	11	1.27	41	3.75
$cryg10000$	1955	1.55	+	+
$dc1$	23	64.48	19	76.22
$dc2$	21	65.19	25	77.17
$dc3$	31	65.36	59	77.29
$ecl32$	121	8.31	+	+
$epb1$	83	0.399	57	0.77

Table 11: Bicgstab method with right preconditioner

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$epb2$	33	0.439	+	+
$epb3$	109	2.65	+	+
$ex24$	2287	1.449	+	+
$ex29$	7	0.01	7	0.24
$ex31$	325	1.05	+	+
$ex36$	221	0.18	+	+
$ex37$	11	0.0199	11	0.26
$ex40$	2325	12.39	+	+
$flowmeter5$	55	0.17	35	0.52
fs_183_1	7	10^{-15}	61	0.21
fs_183_6	5	10^{-15}	45	0.219
$garon1$	1607	0.879	+	+
$hvdc2$	+	+	+	+
ibm_matrix_2	131	2.25	+	+
$jan99jac040sc$	23	1.72	+	+
$jan99jac100sc$	41	6.56	+	+
$jan99jac120sc$	37	8.6	+	+
$Kaufhold$	3	10^{-15}	3	0.24
$lung2$	7	0.18	+	+
$majorbasis$	11	0.84	+	+
$matrix-9$	55	3.3	+	+
$matrix-new\_3$	71	2.52	+	+
$memplus$	49	0.069	17	0.34
$mult\_dcop\_01$	37	4.2	17	0.339
$mult\_dcop\_02$	9	1.96	+	+
$mult\_dcop\_03$	11	2.25	+	+
$ns3Da$	55	16.97	+	+
$olm5000$	49	0.02	+	+
$poisson3Db$	83	6.32	73	9.61
$poli\_large$	7	0.0299	11	0.289
$powersim$	39	0.07	29	0.34
$psmigr\_3$	13	0.159	+	+
$Raj1$	1455	62.03	+	+
$rajat03$	19	0.029	+	+
$rajat15$	273	3.23	243	3.62
$rajat16$	475	11.21	+	+
$rajat18$	515	11.08	+	+
$rajat20$	1283	20.56	+	+
$rajat22$	495	1.57	+	+
$rajat25$	579	12.96	+	+

Table 12: Bicgstab method with right preconditioner

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$rajat27$	161	0.3	+	+
$rajat28$	441	11.95	+	+
$rajat31$	17	23.74	+	+
$rma10$	+	+	+	+
$scircuit$	861	16.19	+	+
$sherman3$	75	0.05	43	0.36
$swang1$	9	10^{-15}	9	0.239
$swang2$	9	9.994	9	0.229
$thermal$	9	0.02	7	0.239
tmt_unsym	1113	236.43	727	166.02
$tomography$	11	0.0399	11	0.26
$torso2$	9	0.52	9	1.09
$torso3$	9	1.23	19	2.89
$trans4$	7	68.24	9	80.17
$trans5$	11	70.19	13	82.07
$TSOPF\_RS\_b39\_c7$	141	0.539	+	+
$TSOPF\_RS\_b39\_c19$	103	1.21	+	+
$TSOPF\_RS\_b39\_c30$	125	2.13	+	+
$utm5940$	539	2.58	+	+
$venkat01$	17	4.26	37	19.83
$venkat25$	101	13.96	+	+
$venkat50$	149	16.98	+	+
$viscoplastic2$	15	35.52	+	+
$wang3$	57	0.759	+	+
$wang4$	47	0.56	+	+
$Zhao1$	9	0.33	+	+

Table 13: GMRES(30) method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$3D\_51448\_3D$	89	1.9	+	+
$af23560$	62	4.17	+	+
$airfoil\_2d$	248	1.48	+	+
$arc130$	3	10^{-15}	+	+
$atmosmodd$	136	82.61	108	102.91
$atmosmodj$	136	82.94	110	103.96
$bcircuit$	853	8.48	175	2.27
$cage9$	9	0.029	9	0.31
$cage10$	9	0.119	9	0.5
$cage11$	10	0.449	10	1.42
$cage12$	10	1.78	10	5.76
$cavity05$	83	0.04	+	+
$cavity10$	625	0.34	+	+
$cavity11$	+	+	+	+
$cavity12$	+	+	+	+
$cavity13$	+	+	+	+
$cavity16$	+	+	+	+
$cavity17$	+	+	+	+
$cavity18$	+	+	+	+
$cavity19$	+	+	+	+
$cavity20$	+	+	+	+
$cell1$	26	0.06	26	0.4
$cell2$	18	0.049	16	0.48
$Chebyshev3$	88	0.04	+	+
$chem\_master1$	75	1.00	48	1.56
$chipcool0$	60	0.81	53	1.47
$chipcool1$	54	0.819	40	1.379
$Circuit\_1$	7	0.059	205	0.349
$Circuit\_2$	9	0.02	51	0.27
$Circuit\_3$	1853	1.79	+	+
$ck104$	10	10^{-15}	13	0.219
$comsol$	+	+	907	0.92
$coupled$	15	0.48	21	0.799
$crashbasis$	10	1.26	29	3.57
$cryg10000$	+	+	+	+
$dc1$	13	64.42	14	76.22
$dc2$	16	65.21	20	77.2
$dc3$	21	65.37	26	77.09
$ecl32$	233	11.16	+	+
$epb1$	91	0.449	58	0.819

Table 14: GMRES(30) method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$epb2$	27	0.419	+	+
$epb3$	105	2.97	+	+
$ex24$	+	+	+	+
$ex29$	7	0.01	7	0.25
$ex31$	1300	3.09	+	+
$ex36$	1329	0.78	+	+
$ex37$	10	0.0199	10	0.26
$ex40$	+	+	+	+
$flowmeter5$	61	0.22	25	0.51
fs_183_1	6	10^{-15}	15	0.21
fs_183_6	4	10^{-15}	4	0.22
$garon1$	2014	1.319	+	+
$hvdc2$	1828	78.48	+	+
ibm_matrix_2	91	1.98	+	+
$jan99jac040sc$	19	1.72	+	+
$jan99jac100sc$	29	6.43	+	+
$jan99jac120sc$	29	8.5	+	+
$Kaufhold$	2	10^{-15}	2	0.25
$lung2$	6	0.2	+	+
$majorbasis$	10	0.89	+	+
$matrix-9$	49	3.319	+	+
$matrix-new\_3$	60	2.65	+	+
$memplus$	38	0.089	14	0.34
$mult\_dcop\_01$	21	4.11	+	+
$mult\_dcop\_02$	8	1.97	+	+
$mult\_dcop\_03$	8	2.25	+	+
$ns3Da$	51	16.88	+	+
$olm5000$	26	0.02	+	+
$poisson3Db$	71	6.14	68	9.63
$poli\_large$	6	0.029	8	0.289
$powersim$	6	0.029	23	0.369
$psmigr\_3$	11	0.149	+	+
$Raj1$	+	+	+	+
$rajat03$	12	0.039	49	0.329
$rajat15$	176	1.91	486	6.12
$rajat16$	77	7.88	+	+
$rajat18$	1702	28.91	+	+
$rajat20$	1718	31.83	+	+
$rajat22$	2469	11.9	+	+
$rajat25$	+	+	+	+

Table 15: GMRES(30) method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$rajat27$	478	1.24	+	+
$rajat28$	241	11.05	+	+
$rajat31$	14	26.9	+	+
$rma10$	329	24.69	+	+
$scircuit$	2199	58.07	+	+
$sherman3$	100	0.06	29	0.369
$swang1$	7	10^{-15}	7	0.239
$swang2$	8	0.00999	8	0.229
$thermal$	7	0.029	7	0.2399
tmt_unsym	+	+	+	+
$tomography$	10	0.049	10	0.25
$torso2$	8	0.529	8	11
$torso3$	8	1.259	14	2.8
$trans4$	6	68.26	7	80.16
$trans5$	10	70.19	9	82.05
$TSOPF\_RS\_b39\_c7$	54	0.34	+	+
$TSOPF\_RS\_b39\_c19$	55	0.91	+	+
$TSOPF\_RS\_b39\_c30$	55	1.41	+	+
$utm5940$	+	+	+	+
$venkat01$	15	4.22	34	19.81
$venkat25$	102	14.3	+	+
$venkat50$	155	17.73	+	+
$viscoplastic2$	14	35.51	+	+
$wang3$	55	0.789	+	+
$wang4$	40	0.56	+	+
$Zhao1$	9	0.34	+	+

Table 16: TFQMR method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$3D\_51448\_3D$	139	2.33	+	+
$af23560$	69	4.23	+	+
$airfoil\_2d$	101	0.63	+	+
$arc130$	3	10^{-15}	+	+
$atmosmodd$	117	55.93	97	83.54
$atmosmodj$	119	56.89	101	84.95
$bcircuit$	+	+	257	2.4
$cage9$	11	0.019	11	0.32
$cage10$	11	0.109	11	0.51
$cage11$	13	0.47	13	1.43
$cage12$	13	1.86	11	5.77
$cavity05$	93	0.03	+	+
$cavity10$	143	0.1	+	+
$cavity11$	533	0.29	+	+
$cavity12$	1201	0.58	+	+
$cavity13$	1609	0.799	+	+
$cavity16$	197	0.259	+	+
$cavity17$	403	0.409	+	+
$cavity18$	779	0.819	+	+
$cavity19$	1409	1.3	+	+
$cavity20$	1665	1.68	+	+
$cell1$	43	0.06	41	0.41
$cell2$	+	+	19	0.5
$Chebyshev3$	+	+	+	+
$chem\_master1$	79	0.899	61	1.6
$chipcool0$	69	0.84	61	1.48
$chipcool1$	61	0.829	51	1.41
$Circuit\_1$	9	0.0599	915	0.459
$Circuit\_2$	11	0.02	407	0.319
$Circuit\_3$	787	0.549	+	+
$ck104$	11	10^{-15}	23	0.219
$comsol$	759	0.709	161	0.549
$coupled$	19	0.47	35	0.799
$crashbasis$	11	1.25	43	3.81
$cryg10000$	+	+	+	+
$dc1$	17	64.42	17	76.21
$dc2$	21	65.2	35	77.28
$dc3$	31	65.36	71	77.41
$ecl32$	109	8.09	+	+
$epb1$	91	0.419	65	0.8

Table 17: TFQMR method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$epb2$	33	0.409	+	+
$epb3$	119	2.86	+	+
$ex24$	+	+	+	+
$ex29$	7	0.02	7	0.25
$ex31$	479	1.34	+	+
$ex36$	233	0.19	+	+
$ex37$	11	0.0199	11	0.27
$ex40$	1731	9.44	+	+
$flowmeter5$	65	0.18	39	0.52
fs_183_1	7	10^{-15}	35	0.21
fs_183_6	5	10^{-15}	45	0.2199
$garon1$	865	1.62	+	+
$hvdc2$	1497	53.26	+	+
ibm_matrix_2	127	2.29	+	+
$jan99jac040sc$	23	1.72	+	+
$jan99jac100sc$	39	6.53	+	+
$jan99jac120sc$	37	8.63	+	+
$Kaufhold$	3	10^{-15}	3	0.24
$lung2$	7	0.179	+	+
$majorbasis$	11	0.86	+	+
$matrix-9$	71	3.87	+	+
$matrix-new\_3$	91	3.04	+	+
$memplus$	45	0.079	17	0.34
$mult\_dcop\_01$	39	4.21	+	+
$mult\_dcop\_02$	9	1.96	+	+
$mult\_dcop\_03$	11	2.25	+	+
$ns3Da$	55	16.96	+	+
$olm5000$	47	0.03	+	+
$poisson3Db$	75	6.00	77	9.72
$poli\_large$	7	0.020004	9	0.289
$powersim$	41	0.07	29	0.36
$psmigr\_3$	13	0.149	+	+
$Raj1$	+	+	+	+
$rajat03$	19	0.039	397	0.42
$rajat15$	179	1.68	461	5.15
$rajat16$	397	10.67	+	+
$rajat18$	603	12.14	+	+
$rajat20$	639	14.27	+	+
$rajat22$	711	2.31	+	+
$rajat25$	487	2.21	+	+

Table 18: TFQMR method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$rajat27$	171	0.309	+	+
$rajat28$	741	15.32	+	+
$rajat31$	19	25.08	+	+
$rma10$	+	+	+	+
$scircuit$	1307	25.39	+	+
$sherman3$	81	0.04	43	0.369
$swang1$	9	0.01	9	0.2399
$swang2$	9	0.0099	9	0.239
$thermal$	9	0.02	7	0.25
tmt_unsym	1595	346.29	1247	273.109
$tomography$	13	0.0399	13	0.26
$torso2$	9	0.529	9	1.11
$torso3$	9	1.229	19	2.94
$trans4$	7	68.25	9	80.15
$trans5$	13	70.21	13	82.06
$TSOPF\_RS\_b39\_c7$	147	0.569	+	+
$TSOPF\_RS\_b39\_c19$	+	+	+	+
$TSOPF\_RS\_b39\_c30$	+	+	+	+
$utm5940$	547	2.58	+	+
$venkat01$	17	4.28	39	19.95
$venkat25$	111	14.52	+	+
$venkat50$	157	17.47	+	+
$viscoplastic2$	15	35.54	+	+
$wang3$	69	0.849	+	+
$wang4$	45	0.54	+	+
$Zhao1$	9	0.33	+	+

Tables 7-9 show that for almost all matrices, the density of the modified right-looking version is bigger than the density of the modified left-looking version. But as it is clear from Tables 10-18, for almost all matrices, the modified left-looking version makes three krylov subspaces methods convergent in less number of iterations than the modified right-looking version. There are some examples in which the density of the modified right-looking version is much bigger than the density of the modified left-looking version, but the Krylov subspace methods are divergent with modified right-looking version and are convergent with modified left-looking version.

In Tables 7-9, the Sec_p of the modified left-looking version is less than the Sec_p of the modified right-looking version except for matrix $rajat27$.

After analyzing the densities of both modified left and right-looking versions, one can conclude that for almost all matrices, the modified right-looking version tends to be denser than the modified left-looking version. With respect to the results of Krylov subspace methods, it can also be concluded that for almost all matrices, the modified left-looking version is more robust than the modified right-looking version to reduce the number of iterations of the Krylov subspace methods.

All the experiments have also been done with $\varepsilon_U = \varepsilon_{Q,W} = 0.1$ for both modified left and right-looking versions. The results have been reported in Tables 19-30. In these tables, ε_U, $\varepsilon_{Q,W}$, *density*, *It*, Sec_p and Sec_{tot} have the same meanings as before. The results of these tables also confirm the previous conclusions.

Table 19: Properties of the preconditioners.

Method	left-looking SAINV-Ns				right-looking SAINV-Ns			
	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$
$3D\_51448\_3D$	0.1	0.1	0.18	0.349	0.1	0.1	89.54	5.62
$af23560$	0.1	0.1	0.279	1.23	0.1	0.1	156.23	26.65
$airfoil\_2d$	0.1	0.1	0.05	0.599	0.1	0.1	13.73	14.03
$arc130$	0.1	0.1	10^{-15}	0.331	0.1	0.1	0.22	0.868
$atmosmodd$	0.1	0.1	2.81	1.257	0.1	0.1	7.24	1.26
$atmosmodj$	0.1	0.1	2.819	1.257	0.1	0.1	7.47	1.26
$bcircuit$	0.1	0.1	0.0899	1.08	0.1	0.1	0.47	1.37
$cage9$	0.1	0.1	10^{-15}	0.262	0.1	0.1	0.22	0.263
$cage10$	0.1	0.1	0.01	0.198	0.1	0.1	0.25	0.198
$cage11$	0.1	0.1	0.04	0.164	0.1	0.1	0.319	0.165
$cage12$	0.1	0.1	0.119	0.138	0.1	0.1	0.59	0.138
$cavity05$	0.1	0.1	0.01	0.482	0.1	0.1	0.42	4.13
$cavity10$	0.1	0.1	0.009	0.384	0.1	0.1	0.919	5.71
$cavity11$	0.1	0.1	0.01	0.367	0.1	0.1	0.889	5.22
$cavity12$	0.1	0.1	0.01	0.447	0.1	0.1	0.909	5.78
$cavity13$	0.1	0.1	0.02	0.478	0.1	0.1	0.839	5.25
$cavity16$	0.1	0.1	0.02	0.302	0.1	0.1	2.83	7.76
$cavity17$	0.1	0.1	0.020004	0.282	0.1	0.1	2.4	7.17
$cavity18$	0.1	0.1	0.20004	0.314	0.1	0.1	2.76	7.78
$cavity19$	0.1	0.1	0.30014	0.301	0.1	0.1	2.38	7.18
$cavity20$	0.1	0.1	0.20004	0.341	0.1	0.1	2.92	7.89
$cell1$	0.1	0.1	10^{-15}	1.77	0.1	0.1	0.25	1.95
$cell2$	0.1	0.1	0.0199	1.78	0.1	0.1	0.239	1.95
$Chebyshev3$	0.1	0.1	10^{-15}	1.345	0.1	0.1	0.309	1.3
$chem\_master1$	0.1	0.1	0.0599	1.75	0.1	0.1	0.369	1.86
$chipcool0$	0.1	0.1	0.0599	0.641	0.1	0.1	0.42	0.649
$chipcool1$	0.1	0.1	0.05	0.639	0.1	0.1	0.379	0.647
$Circuit\_1$	0.1	0.1	0.0599	0.496	0.1	0.1	0.32	0.5
$Circuit\_2$	0.1	0.1	10^{-15}	0.965	0.1	0.1	0.27	0.934
$Circuit\_3$	0.1	0.1	0.0699	1.25	0.1	0.1	0.369	1.91
$ck104$	0.1	0.1	10^{-15}	0.803	0.1	0.1	0.26	1.12
$comsol$	0.1	0.1	0.02	0.381	0.1	0.1	0.25	0.315
$coupled$	0.1	0.1	0.119	0.774	0.1	0.1	0.38	0.819
$crashbasis$	0.1	0.1	0.309	1.01	0.1	0.1	1.05	1.11
$cryg10000$	0.1	0.1	0.0199	1.66	0.1	0.1	0.45	4.64
$dc1$	0.1	0.1	52.67	0.79	0.1	0.1	71.13	0.84
$dc2$	0.1	0.1	53.75	0.775	0.1	0.1	71.41	0.824
$dc3$	0.1	0.1	55.27	0.782	0.1	0.1	71.84	0.831
$ecl32$	0.1	0.1	0.669	2.53	0.1	0.1	862.21	108.07
$epb1$	0.1	0.1	0.0299	1.35	0.1	0.1	0.27	1.51

Table 20: Properties of the preconditioners.

Method	left-looking SAINV-Ns				right-looking SAINV-Ns			
	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$
$epb2$	0.1	0.1	0.05	1.13	0.1	0.1	0.339	1.32
$epb3$	0.1	0.1	0.15	1.42	0.1	0.1	0.589	1.59
$ex24$	0.1	0.1	0.22	4.82	0.1	0.1	0.54	5.71
$ex29$	0.1	0.1	10^{-15}	0.72	0.1	0.1	0.22	0.697
$ex31$	0.1	0.1	0.36	2.187	0.1	0.1	0.63	2.31
$ex36$	0.1	0.1	0.0099	0.632	0.1	0.1	0.55	4.71
$ex37$	0.1	0.1	10^{-15}	0.197	0.1	0.1	0.239	0.196
$ex40$	0.1	0.1	0.0599	0.184	0.1	0.1	40.46	12.05
$flowmeter5$	0.1	0.1	0.02	1.35	0.1	0.1	0.269	1.41
fs_183_1	0.1	0.1	10^{-15}	1.02	0.1	0.1	0.21	1.01
fs_183_6	0.1	0.1	10^{-15}	0.925	0.1	0.1	0.21	1.13
$garon1$	0.1	0.1	9.99	0.395	0.1	0.1	2.05	8.24
$hvdc2$	0.1	0.1	0.369	1.35	0.1	0.1	1.68	2.04
ibm_matrix_2	0.1	0.1	0.179	0.351	0.1	0.1	47.52	3.86
$jan99jac040sc$	0.1	0.1	0.199	3.76	0.1	0.1	40.08	71.45
$jan99jac100sc$	0.1	0.1	0.65	4.49	0.1	0.1	235.08	109.65
$jan99jac120sc$	0.1	0.1	0.88	4.79	0.1	0.1	349.64	125.56
$Kaufhold$	0.1	0.1	0.01	0.454	0.1	0.1	0.23	0.454
$lung2$	0.1	0.1	0.08	1.28	0.1	0.1	0.71	2.61
$majorbasis$	0.1	0.1	0.24	0.778	0.1	0.1	690.09	28.54
$matrix-9$	0.1	0.1	0.349	0.496	0.1	0.1	6308.2	53.17
$matrix-new\_3$	0.1	0.1	0.229	0.225	0.1	0.1	1037.95	15.77
$memplus$	0.1	0.1	0.01	0.49	0.1	0.1	0.26	0.502
$mult\_dcop\_01$	0.1	0.1	2.619	1.67	0.1	0.1	8.59	12.27
$mult\_dcop\_02$	0.1	0.1	1.98	0.65	0.1	0.1	4.73	7.46
$mult\_dcop\_03$	0.1	0.1	2.07	0.733	0.1	0.1	3.2	2.41
$ns3Da$	0.1	0.1	1.69	1.36	0.1	0.1	1000.21	18.26
$olm5000$	0.1	0.1	0.0199	6.007	0.1	0.1	0.26	4.74
$poisson3Db$	0.1	0.1	0.33	0.369	0.1	0.1	88.6	2.71
$poli\_large$	0.1	0.1	0.02	2.04	0.1	0.1	0.27	3.27
$powersim$	0.1	0.1	0.099	1.4	0.1	0.1	0.27	1.93
$psmigr\_3$	0.1	0.1	0.02	0.0147	0.1	0.1	17.03	2.619
$Raj1$	0.1	0.1	4.22	1.245	0.1	0.1	15.01	1.53
$rajat03$	0.1	0.1	10^{-15}	1.27	0.1	0.1	0.25	1.37
$rajat15$	0.1	0.1	0.139	0.567	0.1	0.1	0.54	0.645
$rajat16$	0.1	0.1	5.77	0.831	0.1	0.1	6.67	0.819
$rajat18$	0.1	0.1	6.05	1.02	0.1	0.1	6.69	1.06
$rajat20$	0.1	0.1	5.05	0.922	0.1	0.1	5.73	0.912
$rajat22$	0.1	0.1	0.15	0.91	0.1	0.1	0.569	1.12
$rajat25$	0.1	0.1	4.88	0.851	0.1	0.1	5.72	0.825

Table 21: Properties of the preconditioners.

Method	left-looking SAINV-Ns				right-looking SAINV-Ns			
	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$	ε_U	$\varepsilon_{Q,W}$	Sec_p	$density$
$rajat27$	0.1	0.1	0.0799	1.3	0.1	0.1	0.38	2.19
$rajat28$	0.1	0.1	5.55	0.909	0.1	0.1	6.4	0.916
$rajat31$	0.1	0.1	5.59	1.33	0.1	0.1	46.84	1.58
$rma10$	0.1	0.1	5.43	1.5	0.1	0.1	293.07	15.02
$scircuit$	0.1	0.1	0.21	1.12	0.1	0.1	0.829	1.4
$sherman3$	0.1	0.1	0.01	1.29	0.1	0.1	0.239	1.36
$swang1$	0.1	0.1	10^{-15}	0.536	0.1	0.1	0.229	0.537
$swang2$	0.1	0.1	10^{-15}	0.565	0.1	0.1	0.23	0.567
$thermal$	0.1	0.1	0.01	0.287	0.1	0.1	0.26	0.262
tmt_unsym	0.1	0.1	1.58	1.98	0.1	0.1	3.91	1.98
$tomography$	0.1	0.1	0.0199	0.722	0.1	0.1	0.239	0.722
$torso2$	0.1	0.1	0.14	0.607	0.1	0.1	0.579	0.608
$torso3$	0.1	0.1	0.27	0.164	0.1	0.1	1.35	0.205
$trans4$	0.1	0.1	66.19	0.914	0.1	0.1	68.86	0.959
$trans5$	0.1	0.1	52.00	0.855	0.1	0.1	66.93	0.924
$TSOPF\_RS\_b39\_c7$	0.1	0.1	0.08	0.764	0.1	0.1	0.72	2.611
$TSOPF\_RS\_b39\_c19$	0.1	0.1	0.21	0.76	0.1	0.1	1.61	2.75
$TSOPF\_RS\_b39\_c30$	0.1	0.1	0.299	0.708	0.1	0.1	2.26	2.64
$utm5940$	0.1	0.1	0.099	1.993	0.1	0.1	8.17	24.001
$venkat01$	0.1	0.1	0.51	0.745	0.1	0.1	1.26	3.12
$venkat25$	0.1	0.1	0.989	1.29	0.1	0.1	728.52	35.81
$venkat50$	0.1	0.1	1.02	1.32	0.1	0.1	718.58	35.92
$viscoplastic2$	0.1	0.1	20.35	18.21	0.1	0.1	93.45	40.24
$wang3$	0.1	0.1	0.08	1.614	0.1	0.1	0.37	1.83
$wang4$	0.1	0.1	0.0399	0.869	0.1	0.1	118.64	71.12
$Zhao1$	0.1	0.1	0.079	1.74	0.1	0.1	287.15	164.16

Table 22: Bicgstab method with right preconditioner

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$3D\_51448\_3D$	493	4.04	+	+
$af23560$	297	2.2	+	+
$airfoil\_2d$	231	0.449	+	+
$arc130$	7	10^{-15}	+	+
$atmosmodd$	253	43.97	283	522.28
$atmosmodj$	277	48.14	253	47.86
$bcircuit$	+	+	+	+
$cage9$	19	0.009	17	0.24
$cage10$	17	0.029	17	0.27
$cage11$	19	0.12	19	0.389
$cage12$	19	0.43	19	0.88
$cavity05$	195	0.04	+	+
$cavity10$	329	0.099	+	+
$cavity11$	853	0.24	+	+
$cavity12$	2471	0.689	+	+
$cavity13$	+	+	+	+
$cavity16$	519	0.26	+	+
$cavity17$	709	0.31	+	+
$cavity18$	1257	0.57	+	+
$cavity19$	2407	1.06	+	+
$cavity20$	+	+	+	+
$cell1$	215	0.0899	133	0.289
$cell2$	1163	0.409	159	0.299
$Chebyshev3$	+	+	+	+
$chem\_master1$	237	0.949	211	1.05
$chipcool0$	151	0.46	151	0.73
$chipcool1$	137	0.43	135	0.659
$Circuit\_1$	15	0.05999	+	+
$Circuit\_2$	15	0.01	161	0.299
$Circuit\_3$	+	+	+	+
$ck104$	19	10^{-15}	45	0.26
$comsol$	2037	0.56	1779	0.779
$coupled$	53	0.159	63	0.43
$crashbasis$	35	1.129	+	+
$cryg10000$	+	+	+	+
$dc1$	27	52.94	51	71.65
$dc2$	29	54.02	35	71.74
$dc3$	45	55.76	139	73.13
$ecl32$	425	4.16	+	+
$epb1$	219	0.22	205	0.459

Table 23: Bicgstab method with right preconditioner

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$epb2$	103	0.259	115	0.559
$epb3$	365	3.13	685	5.86
$ex24$	+	+	+	+
$ex29$	15	10^{-15}	15	0.22
$ex31$	+	+	+	+
$ex36$	1191	0.309	+	+
$ex37$	21	10^{-15}	37	0.249
$ex40$	+	+	+	+
$flowmeter5$	215	0.15	197	0.379
fs_183_1	7	10^{-15}	185	0.210006
fs_183_6	7	10^{-15}	535	0.22
$garon1$	+	+	+	+
$hvdc2$	+	+	+	+
ibm_matrix_2	871	7.4	+	+
$jan99jac040sc$	79	0.35	+	+
$jan99jac100sc$	193	1.98	+	+
$jan99jac120sc$	535	5.37	+	+
$Kaufhold$	3	0.02	3	0.23001
$lung2$	13	0.21	+	+
$majorbasis$	23	0.72	+	+
$matrix-9$	137	2.96	+	+
$matrix-new\_3$	149	2.98	+	+
$memplus$	45	0.05	21	0.28
$mult\_dcop\_01$	47	2.73	+	+
$mult\_dcop\_02$	51	2.05	+	+
$mult\_dcop\_03$	13	2.09	+	+
$ns3Da$	+	+	+	+
$olm5000$	1799	0.579	+	+
$poisson3Db$	215	4.04	+	+
$poli\_large$	11	0.0200004	53	0.299
$powersim$	159	0.139	337	0.539
$psmigr\_3$	19	0.05	+	+
$Raj1$	+	+	+	+
$rajat03$	41	0.01	+	+
$rajat15$	623	2.69	955	4.43
$rajat16$	+	+	+	+
$rajat18$	+	+	+	+
$rajat20$	+	+	+	+
$rajat22$	+	+	+	+
$rajat25$	+	+	+	+

Table 24: Bicgstab method with right preconditioner

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$rajat27$	+	+	+	+
$rajat28$	1311	16.09	+	+
$rajat31$	91	47.27	+	+
$rma10$	+	+	+	+
$scircuit$	+	+	+	+
$sherman3$	27	0.0499	173	0.279
$swang1$	+	+	17	0.2299
$swang2$	+	+	17	0.23
$thermal$	17	0.020004	15	0.26
tmt_unsym	+	+	+	+
$tomography$	15	0.0299	15	0.25
$torso2$	17	0.36	17	0.77
$torso3$	23	1.05	+	+
$trans4$	15	66.38	19	69.07
$trans5$	33	52.35	37	68.28
$TSOPF\_RS\_b39\_c7$	+	+	+	+
$TSOPF\_RS\_b39\_c19$	+	+	+	+
$TSOPF\_RS\_b39\_c30$	+	+	+	+
$utm5940$	1111	0.939	+	+
$venkat01$	45	1.379	87	2.76
$venkat25$	777	19.97	+	+
$venkat50$	+	+	+	+
$viscoplastic2$	139	24.35	+	+
$wang3$	131	0.46	+	+
$wang4$	153	0.33	+	+
$Zhao1$	37	0.209	+	+

Table 25: GMRES(30) method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$3D\_51448\_3D$	364	3.88	+	+
$af23560$	375	3.2	+	+
$airfoil\_2d$	+	+	+	+
$arc130$	5	10^{-15}	+	+
$atmosmodd$	276	94.27	273	97.32
$atmosmodj$	374	126.76	370	132.54
$bcircuit$	+	+	+	+
$cage9$	16	10^{-15}	16	0.229
$cage10$	16	0.04	16	0.27
$cage11$	17	0.13	17	0.419
$cage12$	17	0.489	17	0.909
$cavity05$	297	0.05	+	+
$cavity10$	+	+	+	+
$cavity11$	+	+	+	+
$cavity12$	+	+	+	+
$cavity13$	+	+	+	+
$cavity16$	+	+	+	+
$cavity17$	+	+	+	+
$cavity18$	+	+	+	+
$cavity19$	+	+	+	+
$cavity20$	+	+	+	+
$cell1$	+	+	+	+
$cell2$	88	0.069	61	0.279
$Chebyshev3$	+	+	+	+
$chem\_master1$	756	4.33	649	3.74
$chipcool0$	201	0.77	202	1.13
$chipcool1$	209	0.78	254	1.19
$Circuit\_1$	10	0.059	93	0.349
$Circuit\_2$	12	10^{-15}	54	0.28
$Circuit\_3$	+	+	+	+
$ck104$	14	10^{-15}	27	0.26
$comsol$	+	+	+	+
$coupled$	38	0.159	51	0.449
$crashbasis$	30	1.299	+	+
$cryg10000$	+	+	+	+
$dc1$	19	52.94	27	71.53
$dc2$	20	54.05	28	71.84
$dc3$	28	55.75	102	73.32
$ecl32$	+	+	+	+
$epb1$	300	0.579	272	0.69

Table 26: GMRES(30) method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$epb2$	92	0.369	109	0.709
$epb3$	447	5.63	674	8.61
$ex24$	+	+	+	+
$ex29$	14	10^{-15}	14	0.22
$ex31$	+	+	+	+
$ex36$	+	+	+	+
$ex37$	19	10^{-15}	19	0.239
$ex40$	+	+	+	+
$flowmeter5$	653	0.66	651	0.839
fs_183_1	6	10^{-15}	12	0.21
fs_183_6	6	10^{-15}	15	0.21
$garon1$	+	+	+	+
$hvdc2$	+	+	+	+
ibm_matrix_2	353	4.069	+	+
$jan99jac040sc$	61	0.359	+	+
$jan99jac100sc$	146	1.85	+	+
$jan99jac120sc$	180	2.72	+	+
$Kaufhold$	2	0.0100002	2	0.23001
$lung2$	11	0.2200012	+	+
$majorbasis$	20	0.79	+	+
$matrix-9$	168	4.709	+	+
$matrix-new\_3$	268	7.38	+	+
$memplus$	40	0.09	18	0.28
$mult\_dcop\_01$	26	2.719	+	+
$mult\_dcop\_02$	19	2.05	+	+
$mult\_dcop\_03$	10	2.09	+	+
$ns3Da$	+	+	+	+
$olm5000$	1483	0.69	+	+
$poisson3Db$	256	5.75	+	+
$poli\_large$	10	0.04	25	0.309
$powersim$	138	0.229	336	0.769
$psmigr\_3$	17	0.05	+	+
$Raj1$	+	+	+	+
$rajat03$	28	0.02	57	0.289
$rajat15$	1090	6.72	+	+
$rajat16$	+	+	+	+
$rajat18$	+	+	+	+
$rajat20$	+	+	+	+
$rajat22$	+	+	+	+
$rajat25$	+	+	+	+

Table 27: GMRES(30) method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$rajat27$	+	+	+	+
$rajat28$	+	+	+	+
$rajat31$	90	105.04	+	+
$rma10$	+	+	+	+
$scircuit$	+	+	+	+
$sherman3$	713	0.279	526	0.429
$swang1$	15	10^{-15}	15	0.239
$swang2$	16	10^{-15}	15	0.230001
$thermal$	14	0.02	14	0.26
tmt_unsym	+	+	+	+
$tomography$	13	0.0199	13	0.2399
$torso2$	15	0.4	15	0.789
$torso3$	19	1.2	+	+
$trans4$	12	66.39	14	69.09
$trans5$	24	52.39	24	67.3
$TSOPF\_RS\_b39\_c7$	+	+	+	+
$TSOPF\_RS\_b39\_c19$	+	+	+	+
$TSOPF\_RS\_b39\_c30$	+	+	+	+
$utm5940$	+	+	+	+
$venkat01$	42	1.419	78	2.82
$venkat25$	856	24.54	+	+
$venkat50$	1586	44.94	+	+
$viscoplastic2$	105	23.6	+	+
$wang3$	155	0.79	+	+
$wang4$	181	0.679	+	+
$Zhao1$	37	0.279	+	+

Table 28: TFQMR method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$3D\_51448\_3D$	347	2.91	+	+
$af23560$	317	2.4	+	+
$airfoil\_2d$	235	0.529	+	+
$arc130$	7	10^{-15}	+	+
$atmosmodd$	293	52.7	263	50.59
$atmosmodj$	289	52.3	257	49.83
$bcircuit$	+	+	+	+
$cage9$	19	0.00999	17	0.2200012
$cage10$	19	0.0100097	19	0.259
$cage11$	19	0.11	19	0.399
$cage12$	19	0.449	19	0.869
$cavity05$	193	0.03	+	+
$cavity10$	357	0.0099	+	+
$cavity11$	1179	0.32	+	+
$cavity12$	+	+	+	+
$cavity13$	+	+	+	+
$cavity16$	527	0.25	+	+
$cavity17$	817	0.37	+	+
$cavity18$	1915	0.86	+	+
$cavity19$	+	+	+	+
$cavity20$	+	+	+	+
$cell1$	+	+	167	0.309
$cell2$	+	+	221	0.319
$Chebyshev3$	+	+	+	+
$chem\_master1$	333	1.349	249	1.22
$chipcool0$	175	0.499	183	0.83
$chipcool1$	151	0.43	155	0.749
$Circuit\_1$	13	0.059	845	0.46
$Circuit\_2$	17	0.01	+	+
$Circuit\_3$	+	+	+	+
$ck104$	19	10^{-15}	43	0.26
$comsol$	725	0.22	477	0.4
$coupled$	47	0.149	65	0.419
$crashbasis$	35	1.169	+	+
$cryg10000$	+	+	+	+
$dc1$	23	52.91	45	71.58
$dc2$	27	54.04	39	71.79
$dc3$	43	55.72	95	72.77
$ecl32$	457	4.58	+	+
$epb1$	265	0.279	241	0.5

Table 29: TFQMR method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$epb2$	95	0.259	113	0.589
$epb3$	381	3.46	789	6.96
$ex24$	+	+	+	+
$ex29$	17	10^{-15}	17	0.229
$ex31$	+	+	+	+
$ex36$	1193	0.319	+	+
$ex37$	21	0.00999	21	0.25
$ex40$	+	+	+	+
$flowmeter5$	297	0.21	217	0.339
fs_183_1	7	10^{-15}	27	0.2200012
fs_183_6	7	10^{-15}	71	0.21000067
$garon1$	1589	0.54	+	+
$hvdc2$	+	+	+	+
ibm_matrix_2	369	3.4	+	+
$jan99jac040sc$	67	0.329	+	+
$jan99jac100sc$	125	1.5	+	+
$jan99jac120sc$	165	2.28	+	+
$Kaufhold$	3	0.0100002	3	0.23001
$lung2$	13	0.2	+	+
$majorbasis$	23	0.74	+	+
$matrix-9$	163	3.51	+	+
$matrix-new\_3$	205	4.24	+	+
$memplus$	45	0.0399	21	0.28
$mult\_dcop\_01$	43	2.739	+	+
$mult\_dcop\_02$	47	2.05	+	+
$mult\_dcop\_03$	13	2.109	+	+
$ns3Da$	+	+	+	+
$olm5000$	235	0.09	+	+
$poisson3Db$	249	4.76	+	+
$poli\_large$	11	0.03	59	0.309
$powersim$	139	0.119	315	0.519
$psmigr\_3$	19	0.05	+	+
$Raj1$	+	+	+	+
$rajat03$	41	0.01	441	0.369
$rajat15$	601	2.76	+	+
$rajat16$	+	+	+	+
$rajat18$	+	+	+	+
$rajat20$	+	+	+	+
$rajat22$	+	+	+	+
$rajat25$	2415	25.22	+	+

Table 30: TFQMR method with right preconditioner.

Method	left-looking SAINV-Ns		right-looking SAINV-Ns	
	It	Sec_{tot}	It	Sec_{tot}
$rajat27$	715	0.97	+	+
$rajat28$	+	+	+	+
$rajat31$	123	65.5	+	+
$rma10$	+	+	+	+
$scircuit$	+	+	+	+
$sherman3$	217	0.059	211	0.279
$swang1$	17	10^{-15}	17	0.2299
$swang2$	19	0.00999	17	0.2300014
$thermal$	19	0.0100021	17	0.269
tmt_unsym	+	+	+	+
$tomography$	17	0.01999	17	0.2399
$torso2$	19	0.38	21	0.819
$torso3$	23	1.11	+	+
$trans4$	17	66.4	19	69.06
$trans5$	43	52.46	49	67.44
$TSOPF\_RS\_b39\_c7$	+	+	+	+
$TSOPF\_RS\_b39\_c19$	+	+	+	+
$TSOPF\_RS\_b39\_c30$	+	+	+	+
$utm5940$	967	0.84	+	+
$venkat01$	47	1.4	109	3.12
$venkat25$	821	21.45	+	+
$venkat50$	979	25.61	+	+
$viscoplastic2$	105	23.42	+	+
$wang3$	135	0.51	+	+
$wang4$	155	0.359	+	+
$Zhao1$	39	0.22	+	+

5 Conclusion

In this paper, modified left and right-looking versions of the SAINV-Ns preconditioner were presented. The inverse-based dropping technique was applied to construct both modified left and right-looking versions. Numerical experiments on 106 nonsymmetric matrices indicate that the preconditioning time of the modified left-looking version is less than the preconditioning time of the modified right-looking version. The modified right-looking version tends to be denser than the modified left-looking version. But the modified left-looking version is more effective to be used as the right preconditioner to reduce the number of iterations of the Krylov subspace methods.

Acknowledgments

I would like to thank B.T. Haghighi.

References

[1] M. Benzi, M. Tůma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. SIAM J. Sci. Comput., 1998, 19(3): 968 - 994.

[2] M. Benzi, M. Tůma. A robust incomplete factorization preconditioner for positive definite matrices. Numer. Linear Algebra Appl., 2003, 10: 385 - 400, doi: 10.1002/nla.320.

[3] M. Benzi, J.K. Cullum, M. Tůma. Robust approximate inverse preconditioning for the conjugate gradient method. SIAM J. Sci. Comput., 2000, 22: 1318 - 1332.

[4] M. Bollhöfer, Y. Saad. On the relations between ILUs and factored approximate inverses. SIAM J. Matrix Anal. Appl., 2002, 24(1): 219 - 237.

[5] M. Bollhöfer. A robust and efficient ILU that incorporates the growth of the inverse triangular factors. SIAM J. Sci. Comput., 2003, 25(1): 86 - 103.

[6] T. Davis. University of Florida Sparse Matrix Collection. 2009. http://www.cise.ufl.edu/research/sparse/matrices/. Accessed 17 December 2009.

[7] G.H. Golub, C.F. Van Loan. Matrix Computation. 3rd Edition. The John Hopkins University Press, Baltimore, MD, 1996.

[8] G. Karypis, V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput., 1998, 20(1): 359 - 392.

[9] N. Li, Y. Saad, E. Chow. Crout version of ILU for general sparse matrices. SIAM J. Sci. Comput., 2003, 25(2): 716 - 728.

[10] A. Rafiei, F. Toutounian. New breakdown-free variant of AINV method for nonsymmetric positive definite matrices. J. Comput. Appl. Math., 2008, 219(1): 72 - 80, doi: 10.1016/j.cam.2007.07.003.

[11] A. Rafiei, M. Bollhöfer. Extension of inverse-based dropping techniques for ILU preconditioners. Submitted 2010.

[12] Y. Saad. Iterative Methods for Sparse Linear Systems. PWS publishing, Boston, 1996.