Pivoting strategy for an ILU preconditioner

A. Rafiei*
Hakim Sabzevari University

Mahdi Mohseni and Fatemeh Rezaei Fazel†
Hakim Sabzevari University

Abstract

In this paper, a complete pivoting strategy for the right-looking version of $RIF-NS$ preconditioner is presented.

Keywords: preconditioning, pivoting, right-looking version of $RIF-NS$ preconditioner
Mathematics Subject Classification [2010]: 65F10, 65F50, 65F08.

1 Introduction

Consider the linear system of equations of the form $Ax = b$, where the coefficient matrix $A \in \mathbb{R}^{n\times n}$ is nonsingular, large, sparse and nonsymmetric and also $x, b \in \mathbb{R}^n$. An ILU preconditioner M of this system is in the form of $M = LDU \approx A$. This preconditioner will change the original system to the left preconditioned system $M^{-1}Ax = M^{-1}b$. For a proper preconditioner, instead of solving the original system, it is better to solve the left preconditioned system by the Krylov subspace methods [3]. In [1], we have proposed an ILU preconditioner for system $Ax = b$. This preconditioner is termed the $RIF-NS$ and has two left- and right-looking versions.

2 Pivoting strategy for the right-looking $RIF-NS$ preconditioner

Algorithm 1, uses the complete pivoting strategy to compute the right-looking version of $RIF-NS$ preconditioner. Here we explain the step i of this algorithm. At the beginning of this step, $\Pi = \Pi_{i-1}...\Pi_1$ and $\Sigma = \Sigma_1...\Sigma_{i-1}$ are the row and the column permutation matrices, respectively. For $k < i$, the matrices Π_k and Σ_k are the row and the column permutation matrices associated to step k of this algorithm. At the beginning of this step, the parameters m_i, n_i, $iter$, $satisfied\_p$ and $satisfied\_q$ are initialized in line 3. At the end of this step, m_i and n_i will be the total number of row and column pivoting associated to step i. The parameter $iter$ is used to compute the pivot entry in this step. $satisfied\_p$ ($satisfied\_q$) shows whether or not we need to the row (column) pivoting strategy. In line 7 of the algorithm, the vector $(q_i^{(i-1)}, \cdots, q_n^{(i-1)})$ is computed. Suppose that $|q_k^{(i-1)}| = max_{m\geq i+1}|q_m^{(i-1)}|$. If the criterion $|q_i^{(i-1)}| < \alpha\,|q_k^{(i-1)}|$ is satisfied for

*rafiei.am@gmail.com, a.rafiei@hsu.ac.ir.
†Speaker, mmohsenidehsorkh@yahoo.com, rezaeefazel@gmail.com

$\alpha \in (0,1]$, then the row pivoting strategy is applied in lines 8-11 of the algorithm. Suppose that $|p_l^{(i-1)}| = max_{m \geq i+1} |p_m^{(i-1)}|$. If the criterion $|p_i^{(i-1)}| < \alpha\ |p_l^{(i-1)}|$ is satisfied for an $\alpha \in (0,1]$, then the column pivoting strategy is applied in lines 17-20 of the algorithm. After the column pivoting, $satisfied\_p$ is set to $true$ in line 22 and the algorithm will alternate between the row and the column pivoting. After the internal $while$ loop, the pivot entry d_{ii} is set equal to $q_i^{(i-1)}$. In lines 25-28 of the algorithm, the i-th column of matrices W and L, and the i-th row of matrix U are computed.

3 Numerical results

In this section, we have formed 6 artificial linear systems where the coefficient matrices are downloaded from [2] and the exact solution of these systems is the vector $[1, \cdots, 1]^T$. We have used two parameters 0.1 and 1.0 as α to compute the right-looking version of $RIF-NS$ preconditioner with complete pivoting strategy. We have used The command *bicgstab* in $Matlab$ software to solve the original and the left preconditioned systems by the $BICGSTAB$ method. The stopping criterion for all linear systems is satisfied when the relative residual is less than 10^{-6}. We have considered the zero vector as the initial solution for all systems. The density of the preconditioners is defined as :

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

where $nnz(L)$, $nnz(U)$ and $nnz(A)$ are the number of nonzero entries of matrices L, U and A. Table 1, shows the matrix properties and the information of $BICGSTAB$ method to solve the original linear systems. In this table, n and nnz are the dimension and the number of nonzero entries of the matrix. In Tables 1 and 2, the parameters it and $flag$ indicate the number of iterations and the status of the convergence. The parameter $iter$ can be an integer+0.5 indicating convergence halfway through an iteration. When $flag$ is equal to 0, it means that the method has been converged to the desired tolerance within the 2500 iterations. $flag = 2$ shows that the preconditioner is ill-conditioned and $flag = 4$ indicates that one of the scalar quantities calculating during the method became too small or too large to continue computing.

46th Annual Iranian Mathematics Conference
25-28 August 2015
Yazd University

Algorithm 1 Right-looking version of $RIF-NS$ with complete pivoting

Input: $A \in \mathbb{R}^{n\times n}$ and $\tau_w, \tau_l, \tau_u \in (0,1)$ be drop tolerances. **Output:** $\Pi A \Sigma \approx M = LDU$

```
w_i^(0) = e_i, 1 ≤ i ≤ n
for i = 1 to n do
   m_i = n_i = 0, iter = 0, satisfied_p = satisfied_q = false
   while not satisfied_q do
      iter = iter + 1
      If iter = 1, then set q_i^(i-1) = (w_i^(i-1))^T (ΠAΣ)e_i. Otherwise set q_i^(i-1) = p_i^(i-1).
      q_j^(i-1) = (w_j^(i-1))^T (ΠAΣ)e_i, i+1 ≤ j ≤ n
      if |q_i^(i-1)| < α max_{m≥i+1} |q_m^(i-1)| then
         m_i = m_i + 1,   π_{m_i}^(i-1) = I_n and satisfied_p = false
         Choose k such that |q_k^(i-1)| = max_{m≥i+1} |q_m^(i-1)|. Then, interchange columns i and k of W − I and rows i and k of
         π_{m_i}^(i-1) and L − I. Also interchange elements q_i^(i-1) and q_k^(i-1) and do the update Π = π_{m_i}^(i-1) Π
      end if
      satisfied_q = true
      if not satisfied_p then
         p_i^(i-1) = q_i^(i-1)
         p_j^(i-1) = (ΠAΣ)_{ij},   i+1 ≤ j ≤ n.
         p_j^(i-1) = p_j^(i-1) − L_ik d_kk U_kj for k = 1 to i−1 and j = i+1 to n
         if |p_i^(i-1)| < α max_{m≥i+1} |p_m^(i-1)| then
            n_i = n_i + 1,   σ_{n_i}^(i-1) = I_n and satisfied_q = false
            Choose l such that |p_l^(i-1)| = max_{m≥i+1} |p_m^(i-1)|. Then, interchange columns i and l of σ_{n_i}^(i-1) and U − I. Also,
            interchange elements p_i^(i-1) and p_l^(i-1) and do the update Σ = Σσ_{n_i}^(i-1)
         end if
      end if
      satisfied_p = true
   end while
   d_ii = q_i^(i-1)
   for j = i+1 to n do
      w_j^(i) = w_j^(i-1) − (q_j^(i-1)/d_ii) w_i^(i-1) and for all l ≤ i, if |w_lj^(i)| < τ_w, then set w_lj^(i) = 0
      L_ji = q_j^(i-1)/d_ii,   U_ij = p_j^(i-1)/d_ii. If |L_ji| < τ_l, then set L_ji = 0. If |U_ij| < τ_u, then set U_ij = 0.
   end for
end for
Return L = (L_ij)_{1≤i,j≤n}, U = (U_ij)_{1≤i,j≤n}, D = diag(d_ii)_{1≤i≤n}, Π and Σ.
```

Table 1

Matrix	n	nnz	without preconditioner	
			it	$flag$
$bwm200$	200	796	109.5	0
str_400	363	3157	0	4
$tols90$	90	1746	28	4
str_0	363	2454	0	4
$tub100$	100	396	106.5	0
$08blocks$	300	592	1	4

In Table 2, the notation $RLRIF-NSP(\alpha)$ refers to the right-looking version of $RIF-NS$ preconditioner with complete pivoting strategy which is computed by using the parameter α.

Table 2

Method	RLRIF-NSP(0.1)					RLRIF-NSP(1.0)					RLRIF-NS		
Matrix	$density$	$Rpiv$	$Cpiv$	$iter$	$flag$	$density$	$Rpiv$	$Cpiv$	$iter$	$flag$	$density$	$iter$	$flag$
$bwm200$	1.0012	0	0	23.5	0	1.2073	84	81	19.5	0	1.0012	23.5	0
str_400	0.5854	357	5	11	0	0.6097	383	57	0	2	5.5958	0	2
$tols90$	0.1523	18	0	2.5	0	0.4370	20	3	2.5	0	0.3070	12	0
str_0	0.5028	358	0	2	0	0.5676	362	29	2	0	3.9238	0	2
$tub100$	1.0050	0	0	8	0	1.1591	62	59	7.5	0	1.0050	8	0
$08blocks$	1.4797	292	0	1.5	0	118.3513	32	5	0	2	2	0.5	4

The $RLRIF - NS$ is a notation for the right-looking version of $RIF - NS$ preconditioner. The columns *Rpiv* and *Cpiv* show the total number of row and column pivoting. In this table, the information in the columns $flag$ and *iter* associated to the three preconditioners indicate that for all of the matrices, one of the preconditioners $RLRIF - NSP(1.0)$ or $RLRIF - NSP(0.1)$ gives better results of the $BICGSTAB$ method than the $RLRIF - NS$ preconditioner. This means that the complete pivoting strategy with one of the values $\alpha = 1.0$ or $\alpha = 0.1$ has a good effect on the quality of the right-looking version of $RIF - NS$ preconditioner.

References

[1] A. Rafiei, M. Bollhöfer, *Robust Incomplete Factorization for Nonsymmetric matrices*, Numerische Mathematik, 118(2), 247-269 (2011).

[2] T. Davis, *University of Florida Sparse Matrix Collection.*, *http://www.cise.ufl.edu/research/sparse/matrices.*, Accessed 2015.

[3] Y. Saad, *Iterative Methods for Sparse Linear Systems.* PWS publishing, New York., (1996).