PRECONDITIONED INEXACT NEWTON-LIKE METHOD FOR LARGE NONSYMMETRIC EIGENVALUE PROBLEMS

HONG-YI Miao
Jiangsu Key Laboratory for NSLSCS, School of Mathematical Science
Nanjing Normal University
and
College of Science, Nanjing Forestry University
Nanjing, 210023, China

LI Wang
Jiangsu Key Laboratory for NSLSCS, School of Mathematical Science
Nanjing Normal University, Nanjing, 210023, China

(Communicated by Yongzhong Song)

Abstract. An efficiently preconditioned Newton-like method for the computation of the eigenpairs of large and sparse nonsymmetric matrices is proposed. A sequence of preconditioners based on the Broyden-type rank-one update formula are constructed for the solution of the linearized Newton system. The properties of the preconditioned matrix are investigated. Numerical results are given which reveal that the new proposed algorithms are efficient.

1. Introduction. In many fields of science and engineering technology, we often need to calculate several eigenvalues with largest (smallest) real (imaginary) parts of large and sparse matrices [3]. Let $A$ be a nonsymmetric $n \times n$ matrix, and consider the eigenvalue problem

$$Au = \lambda u, \lambda \in C, u \in C^n. \tag{1}$$

Many iterative methods have been developed, such as the Krylov subspace methods [12], the Davidson method and the Jacobi-Davidson method [13], etc.

Assume that $(u_1, \lambda_1)$ is an algebraically simple eigenpair of a symmetric matrix $A$, the $k$-th step of standard Newton method to solve the nonlinear equations

$$F(u, \lambda) = \begin{pmatrix} Au - \lambda u \\ \frac{1}{2}(u^Tu - 1) \end{pmatrix} = 0 \tag{2}$$

can be described as

$$\begin{pmatrix} A - \lambda_k I & -u_k \\ u_k^T & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta \lambda \end{pmatrix} = -\begin{pmatrix} Au_k - \lambda_k u_k \\ \frac{1}{2}(u_k^Tu_k - 1) \end{pmatrix},$$

2010 Mathematics Subject Classification. Primary: 65F15, 65N25; Secondary: 65F10.

Key words and phrases. eigenvalues, nonsymmetric matrix, Broyden update, preconditioner, inexact Newton method.

The authors are supported by the National Natural Science Foundation of China under grant 41571380 and 10971102, Major project 16KJA110901 of the Natural Science Foundation of the Jiangsu Higher Education Institution.

* Corresponding author: Li Wang.
the next approximation is set as \( u_{k+1} = u_k + \delta u, \lambda_{k+1} = \lambda_k + \delta \lambda \). Newton method is locally quadratic convergent.

In each step of iteration the approximated eigenvector is required to be normalized, and the current Rayleigh quotient is taken as the approximation of the eigenvalue. Then we have

\[
\left( A - \theta_k I \begin{array}{c} \begin{array}{c} u_k^T \\ 0 \end{array} \end{array} \right) \left( \begin{array}{c} \delta u \\ 0 \end{array} \right) = - \left( \begin{array}{c} \delta \theta \\ 0 \end{array} \right) = - \left( \begin{array}{c} A u_k - \theta_k u_k \\ 0 \end{array} \right),
\]

where \( \theta_k = \frac{u_k^T A u_k}{u_k^T u_k} \) and \( u_k^T u_k = 1 \). Then we arrive the Newton method in the unit sphere or Newton-Grassman method with cubic convergent rate [8, 14]. From 3, \( (A - \theta_k I)\delta u = -(Au_k - \theta_k u_k) + \delta \theta u_k = -r_k + \delta \theta u_k \), where \( \delta \theta \) is selected to ensure that the correction \( \delta u \) is orthogonal to \( u_k \). The aim is to construct a sequence of vectors \( u_k \) by solving the linear systems

\[
(I - u_k u_k^T)(A - \theta_k I)(I - u_k u_k^T)s = -r_k,
\]

ensuring that the correction \( s \) is orthogonal to \( u_k \). Then the next approximation is set as \( u_{k+1} = t(\theta_k) = u_k + s \). The same linear system represents the correction equation in the well-known Jacobi-Davidson method [13], which can be viewed as an accelerated Inexact Newton method [7].

When matrix \( A \) is nonsymmetric, the corresponding Jacobi-Davidson correction equation [9] will be

\[
(I - u_k u_k^*)^T(A - \theta_k I)(I - u_k u_k^*)s = -r_k,
\]

and corresponding Newton equation is

\[
\left( \begin{array}{c} A - \theta_k I \\ u_k^T \\ 0 \end{array} \right) \left( \begin{array}{c} \delta u \\ 0 \end{array} \right) = - \left( \begin{array}{c} A u_k - \theta_k u_k \\ 0 \end{array} \right).
\]

There are some difficulties when using standard Newton method or Jacobi-Davidson method. First, the choice of an initial vector is a very challenge problem. Second, the saddle point matrix in the correction equation are usually ill-conditioned, or at least more ill-conditioned than the original matrix \( A \). Restart strategy can in part solve the first problem. To optimize the initial vector choice, Jacobi-Davidson method expands the subspace and restarts to update the initial vector at each iteration step. Newton-like methods such as Newton-Grassman method and so on [8, 14] are simplified Jacobi-Davidson methods with certain simplified subspace expansion. For the second problem, many authors have tried to find a good preconditioner [1, 8, 16] in order to solve the equation 3 efficiently with Krylov subspace methods [12]. This will be our focus in this paper since solving the linear equation efficiently is the key point of these methods.

For saddle point problems, numerous solution techniques have been proposed for solving this type of systems, such as HSS preconditioning, constraint preconditioning, nonlinear Uzawa algorithm, SOR-like method [4, 11]. In [6] a preconditioner based on the BFGS rank-two update was proposed to solve the Newton correction equation 3. When the matrix \( A \) is symmetric positive definite (SPD), BFGS update preserves the properties of positive definite. And limited-memory BFGS two-loop recursion formula can greatly deduce the storage of calculation. But when the matrix \( A \) is not SPD, BFGS update will not be suitable. So in this paper we are mainly concerned with the efficient preconditioning of the linear system when matrix \( A \) is
nonsymmetric. Our approach is in the framework of L. Bergamaschi in [5], where a low-rank update preconditioner is suggested for solving nonlinear equations.

This paper is organized as follows: in Section 2 we introduce the preconditioner and the corresponding new algorithm; Section 3 is devoted to the proof of the main theorem which states the convergence of our new algorithm. Section 4 focuses on implementation details. We end with some numerical experiments and conclusions in Sections 5 and 6.

2. Broyden-type rank-one update preconditioner. Following the idea described in [5], we propose a sequence of preconditioners for the Newton systems using the Broyden rank-one update. The initial Newton system is \( J_0 s_0 = -r_0 \), where

\[
J_0 = (I - u_0^*(A - \theta_0 I)I - u_0^*), \quad r_0 = -(Au_0 - \theta_0 u_0),
\]

\( u_0 \) is the initial unit vector. We chose to use an incomplete LU preconditioner: \( P_0 = (I - u_0^*P_0(I - u_0^*) \) with \( P_0 = (LU)^{-1} \) where L and U are incomplete LU factors of \( A - \theta_0 I \). Then a sequence of projected preconditioners for the subsequent linear systems \( J_s k^{-1} s_k = -r_k \) may be defined by using the Broyden-type formula as:

\[
P_{k+1} = (I - u_{k+1}^* u_{k+1}) \hat{P}_{k+1}(I - u_{k+1}^* u_{k+1}),
\]

where

\[
\hat{P}_{k+1} = \hat{P}_k + \frac{(y_k - \hat{P}_k s_k)s_k^*}{s_k^* s_k} \tag{7}
\]

and \( s_k \) is the solution of the \( k \)-th Newton system whereas \( y_k = r_{k+1} - r_k \).

It is well known that Newton method converges quadratically when the solution is simple which implies that \( ||e_{k+1}|| \ll ||e_k|| \) in a suitable neighborhood of the solution, where \( e_k \) is the error between the approximate eigenvector and the exact one in \( k \)-th step. So the residual norm also satisfies \( ||r_{k+1}|| \ll ||r_k|| \). Then we can approximate \( y_k \) with \(-r_k\) and write the preconditioner as:

\[
\hat{P}_{k+1} = \hat{P}_k + \frac{(-r_k - \hat{P}_k s_k)s_k^*}{s_k^* s_k} \tag{8}
\]

The final preconditioner \( \hat{P}_k^{-1} \) can be obtained by inversion using the Sherman-Morrison formula to yield:

\[
\hat{P}_k^{-1} = \hat{P}_k^{-1} - \frac{(\hat{P}_k^{-1} r_k + s_k)s_k^* \hat{P}_k^{-1}}{s_k^* \hat{P}_k^{-1} r_k}. \tag{9}
\]

When the preconditioner times a vector, which is needed when using a Krylov method, we want to compute \( \hat{P}_k^{-1} z_k \) for a given vector \( z_k \). The price is of the application of \( \hat{P}_k^{-1} \), \( k \) dot products and \( k \) daxpy operations. And the updating of the preconditioner is well suited to parallelization [5].

GMRES method [12] can be used to solve the Newton system 3. The costs of memory for storing \( s \) and \( r \) and the cost of preconditioner application will increase with the iteration index \( k \). To overcome it, a restart strategy with fixed dimension \( k_{\text{max}} \) the maximum number of rank one corrections we allow (e.g. ten or twenty). When the iteration counter \( k \) is larger than \( k_{\text{max}} \), the vectors \( s_i, r_i, i = k - k_{\text{max}} \) are substituted with the last computed \( s_k, r_k \).

The implementation of our preconditioned GMRES-INewton method for computing eigenpairs of nonsymmetric matrices is described in Algorithm 1.
Algorithm 1 (Preconditioned GMRES-INewton Method)

1. Input: matrix $A$, initial unit vector $u_0$, tolerance $\tau$ and maximum number of iterations for the outer iteration $ITMAX$, tolerance $\tau_{GMRES}$ and maximum number of iterations for the inner iteration $ITMAX_{GMRES}$, maximum allowed rank-one updates in the Broyden preconditioner $k_{max}$.
2. Compute an incomplete LU factorization of $A - \theta_0 I$: $\tilde{P}_0$
3. For $k:=0,1,2,\ldots$
   (1) Solve Newton equation by the PGMRES method with preconditioner $P_k$ and tolerance $\tau_{GMRES}$ such that $\|J_k s_k + r_k\| \leq \tau_k \|r_k\|$ where $\tau_k$ is the forcing terms.
   (2) $u_{k+1} := \frac{u_k + s_k}{\|u_k + s_k\|}, \theta_{k+1} := \theta_k + \delta_0$.
   (3) Compute the associated residual vector $r_{k+1} = Au_{k+1} - \theta_{k+1} u_{k+1}$.
   Test for convergence. Stop if satisfied.
   (4) Goto(1).

3. Theoretical analysis of the preconditioner. The idea of the Broyden preconditioner is taken from the analysis in [5] where a sequence of preconditioners is devised to precondition the sequence of Newton systems for a general nonlinear problem. One of the “standard assumptions” made in this paper was the nonsingularity of the Jacobian in the solution of the nonlinear system.

Theorem 3.1. [16] Suppose $(u_\ast, \lambda_\ast)$ is the solution of 2, and $\lambda_\ast$ is a simple eigenvalue, then the Jacobian matrix $J_k = \begin{pmatrix} A - \lambda_k I & -u_k \\ u_k^* & 0 \end{pmatrix}$ is nonsingular at $(u_\ast, \lambda_\ast)$.

Due to the continuity, when $\lambda_k$ is close to a simple eigenvalue, $J_k$ is nonsingular as well.

At every step of our Newton method we approximately solve the correction equation

$$ (I - u_k u_k^*(A - \theta_k I)(I - u_k u_k^*)s = -(Au_k - \theta_k u_k) = -r_k $$

and the solution $s$ is orthogonal to $u_k$.

It is easy to show that this correction equation is equivalent to

$$ (A - \theta_k I)(I - u_k u_k^*)s = -r_k + cu_k, $$

where $c$ is a constant number.

When $A - \theta_k I$ is singular, $\theta_k$ is the eigenvalue of $A$ and the nontrivial solution of equation $(A - \theta_k I)x=0$ is the corresponding eigenvector.

When $A - \theta_k I$ is nonsingular, $(I - u_k u_k^*)s = -u_k + (A - \theta_k I)^{-1}u_k$, if $r_k \neq 0$, then $u_k$ is not the eigenvector of $A$, so $(I - u_k u_k^*)s \neq 0$, which means the correction of $s$ is possible.

Then we set

$$ u_{k+1} = \frac{u_k + s}{\|u_k + s\|} = \frac{u_k + s}{\sqrt{1 + \|s\|^2}} = \frac{u_k + s}{\beta} $$

in view of $\|u_k\| = 1$, where we have defined $\beta = \sqrt{1 + \|s\|^2}$.

It is well known that the above Newton iteration converges quadratically if the correction equation is solved exactly. It is even cubically when the matrix is symmetric. Since this is not the case when it is solved iteratively, we simply assume convergence, namely that for a suitable $\delta > 0$ such that $\|e_0\| < \delta$ there is a constant $l < 1$ such that
Theorem 3.2. Let \( F'(u, \lambda) \) be the Jacobian matrix of 2, for arbitrary \((u_1, \lambda_1)\) and \((u_2, \lambda_2)\), then \( \|F'(u_2, \lambda_2) - F'(u_1, \lambda_1)\| \leq \frac{\sqrt{e}}{2} \|u_2 - (u_1, \lambda_1)\| \).

Proof. This follows from a straight calculation.

\[
\|F'(u_2, \lambda_2) - F'(u_1, \lambda_1)\| = \left\| \begin{pmatrix} A - \lambda_1 I & -u_2 \\ u_2 & 0 \end{pmatrix} - \begin{pmatrix} A - \lambda_1 I & -u_1 \\ u_1 & 0 \end{pmatrix} \right\|.
\]

The matrix \( \begin{pmatrix} (\lambda_2 - \lambda_1)I & u_2 - u_1 \\ -(u_2 - u_1)^* & 0 \end{pmatrix} \) is denoted as \( A \). Let \( \lambda_2 - \lambda_1 = a, u_2 - u_1 = (x_1, x_2, ..., x_n)^T \), then by calculation, we have

\[
\|A\|^2 = \frac{3}{2} (a^2 + 2|x_1|^2 + |x_2|^2 + ... + |x_n|^2) \leq \frac{3}{2} \|u_2, \lambda_2\| - (u_1, \lambda_1)\|^2.
\]

So, \( \|F'(u_2, \lambda_2) - F'(u_1, \lambda_1)\| \leq \frac{\sqrt{e}}{2} \|u_2, \lambda_2\| - (u_1, \lambda_1)\| \). □

From above theorem, it is easy to see that the “standard assumptions” \([5]\) on \( F \) are guaranteed and the preconditioned inexact Newton method is convergent.

4. Implementation. 4.1. Combination with subspace method. Subspace methods are often used in eigenvalue computations, such as the popular Jacobi-Davidson method \([13]\). Since we search for the next approximate eigenvector in the subspace that is orthogonal to the previous approximate eigenvector instead of the whole space, the efficiency is greatly improved. So we combined the preconditioned inexact Newton method with subspace method and listed the algorithm below.

Algorithm 2 (Preconditioned GMRES-INewton Subspace Method)

1. Input: matrix \( A \), initial unit vector \( u_0 \), \( \theta_0 \), \( U_0 = [u_0] \), tolerance \( \tau \) and maximum number of iterations for the outer iteration \( ITMAX \), tolerance \( \tau_{PGMRES} \) and maximum number of iterations for the inner iteration \( ITMAX_{PGMRES} \), maximum allowed rank-one updates in the Broyden preconditioner \( k_{max} \).
2. Compute an incomplete LU factorization of \( A - \theta_0 I \): \( P_0 \).
3. For \( k := 1, 2, \ldots \)
   1. Solve Newton equation by the PGMRES method with preconditioner \( P_k \) and tolerance \( \tau_{PGMRES} \) such that \( \|J_k s_k + r_k\| \leq \tau_k \|r_k\| \) where \( \tau_k \) is the forcing terms.
   2. \( u_{k+1} := \frac{u_{k+1}}{u_{k+1} + s_k} \).
   3. \( U_{k+1} = \text{MGS}(U_k, u_{k+1}) \) (Modified Gram-Schmidt).
   4. Compute the eigenpair \((\theta, s)\) of \( H_{k+1} = U_{k+1}^T A U_{k+1} \).
   5. Compute Ritz vector \( u_{k+1} = U_{k+1} s_k \), \( \hat{u}_{k+1} = \hat{u}_{k+1} + \theta u_{k+1} \) and the associated residual vector \( r_{k+1} = \hat{u}_{k+1} - \theta u_{k+1} \). Test for convergence. Stop if satisfied.
   6. Goto(1).
It is obviously that algorithm 2 is more efficient than algorithm 1 since subspace method is combined with preconditioned Newton method.

4.2. Choosing an initial eigenvector guess. As mentioned in Section 1, choosing an initial eigenvalue guess plays an important role in the efficiency of the Newton approach for eigenvalue computation. With the increasing of the dimension of the subspace, the costs of memory and computation will increase correspondingly. Restart strategy is indispensable when these subspace methods are applied to solve eigenproblems. An obvious method is using the current approximate eigenvector as the initial vector to restart the subspace algorithm. But some useful information may be abandoned and the convergence rate will be decreased in turn. Choosing some approximate eigenvectors that near the current one to restart will be better. When the dimension of the search space reach a given upper bound maxdim, some current approximate eigenvectors with the number mindim will be chosen to form the basis of the new search subspace. As suggested in [10], the choice of the two numbers can be

\[ 5 \leq \text{mindim} \leq 10, \quad 20 \leq \text{maxdim} \leq 30. \]

5. Numerical Experiment. In this section, some numerical experiments are implemented to show effectiveness of our theoretical results and to illustrate the numerical behavior of the proposed algorithms. All the numerical experiments were run by Matlab 7.0, under the Windows XP operating system. To be convenient, the classical inexact Newton method with GMRES to solve the correction equation is denoted as GMRES-INewton method and the preconditioned one is denoted as PGMRES-INewton method. The calculated time listed below does not include the time of updating the preconditioner.

In our numerical experiments, the initial vector is selected to be close to \( x_1 \) which is the normalized eigenvector corresponding to the eigenvalue \( \lambda_1 \) of the matrix \( A \). For example, it can be generated as \( u_0 = x_1 + 0.1 \cdot \text{randn}(n, 1) \), where \( \text{randn}(n, 1) \) is an \( n \)-by-1 pseudo-random vector obeying the standard normal distribution. If the initial vector is far away from \( x_1 \) and close to the eigenvector corresponding to another eigenvalue, then the result will not be the eigenvalue \( \lambda_1 \). In actual implementation we can use other methods such as power method and Arnoldi method to get a good approximate initial vector of \( x_1 \).

As discussed in [14], choosing very small \( \tau_k \) or decreasing their values during the iterations can hardly accelerates the rate of convergence. So to save costs, it is critical to choose a proper value of \( \tau_k \) at the start of the iteration and guarantee the convergence of the process. In order to avoid unnecessary small values of \( \tau_k \) at the start and too large values of \( \tau_k \) at the end of the convergence, we can choose a decreasing sequence of values of \( \tau_k \). A practical strategy for the choice of \( \tau_k \) was suggested in [2] to be \( \tau_k = O(\|r_k\|) \).

**Example 5.1** In this example, we try to show that the PGMRES-INewton method is superior to the GMRES-INewton method. Consider the convection-diffusion differential equation

\[-\Delta u(x, y) + p_1 u_x(x, y) + p_2 u_y(x, y) - p_3 u(x, y) = f(x, y)\]

defined on a square region \( [0, 1] \times [0, 1] \) with the boundary condition \( u(x, y) = 0 \), where \( p_1, p_2 \) and \( p_3 \) are nonnegative constants. Discretize the equation with five point difference on a uniform \( N \times N \) grid and number the grid points arranging in row natural order yielding a block tridiagonal matrix of the form
NEWTON-LIKE METHOD FOR NONSYMMETRIC EIGENVALUE PROBLEMS

\[ A = \begin{pmatrix}
    T & (\beta + 1)I \\
    (-\beta + 1)I & T \\
    \vdots & \vdots \\
    (\beta + 1)I & (-\beta + 1)I \\
\end{pmatrix}. \]

with

\[ T = \begin{pmatrix}
    4 - \sigma & \gamma - 1 \\
    -\gamma - 1 & 4 - \sigma & \gamma - 1 \\
    \vdots & \vdots & \ddots & \gamma - 1 \\
    -\gamma - 1 & 4 - \sigma \\
\end{pmatrix}. \]

where \( \beta = p_1 h/2, \gamma = p_2 h/2, \sigma = p_3 h^2 \) and \( h = 1/(N+1) \). The size of the matrix \( A \) is \( n = N^2 \). In this example, we test the problem with \( p_1 = 1, p_2 = p_3 = 0 \), and \( n = 1024 \).

Table 1. The results for the largest real eigenvalue of \( A \)

| Algorithm      | Eigenvalue | CPU time (s) |
|----------------|------------|--------------|
| GMRES-INewton  | 7.9817     | 9.8430       |
| PGMRES-INewton | 7.9817     | 6.9330       |

The computed eigenvalue together with CPU times by the above two methods are listed in Table 1. The results show that the PGMRES-INewton method can accelerate the convergence speed by preconditioning the correction equation in every iteration step. From Table 2 we can see that our preconditioner can greatly decrease the condition number of the initial coefficient matrix.

Example 5.2 [15]. Dielectric channel waveguide arises in many integrated circuit applications. Consider the governing Helmholtz equation for the metric field \( H \)

\[ \begin{cases}
    \nabla^2 H_x + k^2\varphi^2(x,y)H_x = \beta^2 H_x \\
    \nabla^2 H_y + k^2\varphi^2(x,y)H_y = \beta^2 H_y
\end{cases} \]

We discretize the above equation by finite difference yielding a non-Hermitian eigenvalue problem

\[ \begin{pmatrix}
    c_{11} & c_{12} \\
    c_{21} & c_{22}
\end{pmatrix}
\begin{pmatrix}
    H_x \\
    H_y
\end{pmatrix} = \beta^2
\begin{pmatrix}
    B_{11} & 0 \\
    0 & B_{22}
\end{pmatrix}
\begin{pmatrix}
    H_x \\
    H_y
\end{pmatrix} \]

Table 2. The results of GMRES-INewton and PGMRES-INewton method

|           | GMRES-INewton | PGMRES-INewton |
|-----------|---------------|----------------|
| k         | \( \|r_k\| \) | \( \|r_k\| \) |
| 1         | \( 2.8670e+03 \) | 3.1275e+09     |
| 2         | \( 5.6281e+01 \) | \( 6.1342e+04 \) |
| 3         | \( 1.2415e+00 \) | \( 3.3108e+04 \) |
| 4         | \( 1.3586e-04 \) | \( 3.0213e+03 \) |
where $C_{11}, C_{22}$ are five- or tri-diagonal matrices, $C_{12}, C_{21}$ are tri-diagonal or diagonal matrices, and $B_{11}, B_{22}$ are nonsingular diagonal matrices.

There are eigenvalues with negative real part several orders of magnitude larger than the desired eigenvalues with positive real part, and also the desired eigenvalues are clustered when $n$, the size of the matrix in question is large. Therefore, this problem presents a challenge to all existing numerical methods for large eigenproblems. In this example, we test the problem with $n = 2048$, and the corresponding data file is $Dw2048.mat$ that is available from http://www.cise.ufl.edu/research/sparse/matrices/Bai/index.html.

### Table 3. The results for the largest real eigenvalue of $A$

| Algorithm       | eigenvalue | CPU time (s) |
|-----------------|------------|--------------|
| GMRES-INewton   | 0.4338     | 12.5310      |
| PGMRES-INewton  | 0.4338     | 10.4990      |

### Table 4. The results of GMRES-INewton and PGMRES-INewton method

|                      | GMRES-INewton | PGMRES-INewton |
|----------------------|---------------|----------------|
|                      | $||r_k||$    | $|\lambda_k - \lambda^*|$ | cond$(A)$ | $||r_k||$    | $|\lambda_k - \lambda^*|$ | cond$(P^{-1}A)$ |
| k                    |              |                |          |              |                |                |
| 1                    | $2.6740e+02$ | $5.8630e+01$    | $1.2640e+10$ | $2.6740e+02$ | $5.8630e+01$    | $1.2640e+10$    |
| 2                    | $4.3572e+01$ | $1.0270e+01$    | $6.0390e+07$ | $7.6230e+00$ | $9.6810e-02$    | $5.6210e+05$    |
| 3                    | $3.0450e+00$ | $7.8140e-01$    | $2.4630e+05$ | $3.2540e+00$ | $4.2830e-02$    | $3.2790e+04$    |
| 4                    | $6.3580e+02$ | $4.6720e-02$    | $5.1852e+04$ | $4.9130e+02$ | $1.4270e-04$    | $4.5870e+04$    |
| 5                    | $2.3710e-04$ | $1.0390e-03$    | $1.3089e+05$ |              |                |                |

From Table 3 and Table 4 it is easy to see that the preconditioner used in solving the correction equation is useful.

### Example 5.3

To show that our preconditioned method is superior to the original one, especially for seriously nonsymmetric matrix, a random matrix is solved in this example. Since the matrix is produced randomly, it is nonsymmetric seriously. The numerical result which is listed below shows that the more nonsymmetric the matrix is, the more efficient our preconditioned method is. Table 5 shows that the CPU time spent in solving the seriously nonsymmetric matrix with our preconditioned method is about half of that with original method. Table 6 shows the result of complex eigenvalue when the initial approximate eigenvector is complex.

### Table 5. The results for the largest real eigenvalue of a random matrix

| Algorithm       | eigenvalue | CPU time (s) |
|-----------------|------------|--------------|
| GMRES-INewton   | 298.2441   | 15.7350      |
| PGMRES-INewton  | 298.2441   | 9.1470       |
Table 6. The results for the largest imaginary eigenvalue of a random matrix

| Algorithm         | eigenvalue       | CPU time (s) |
|-------------------|------------------|--------------|
| GMRES-INewton     | -0.0001+0.0054i  | 16.2370      |
| PGMRES-INewton    | -0.0001+0.0054i  | 10.4790      |

6. **Conclusion.** In this paper, a preconditioner based on Broyden-type rank-one update formula is presented. We show that the eigenproblem of large and sparse nonsymmetric matrices can be solved by the preconditioned inexact Newton-like method efficiently. The properties of the preconditioned matrix are investigated. The numerical experiments show that the new proposed algorithms are efficient.

**Acknowledgments.** We are very grateful to the reviewers for their valuable comments on the manuscript that led to many improvements.

**REFERENCES**

[1] Z. Z. Bai and G. H. Golub, *Accelerated Hermitian and skew-Hermitian splitting methods for saddle-point problems*, *SIAM J. Numer. Anal.*, **27** (2007), 1–23.
[2] Z. Z. Bai and C. Q. Miao, *On local quadratic convergence of inexact simplified Jacobi-Davidson method*, *Linear Algebra Appl.*, **520** (2017), 215–241.
[3] K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, 1982.
[4] M. Benzi, G. H. Golub and J. Liesen, *Numerical solution of saddle point problems*, *Acta Numer.*, **14** (2005), 1–137.
[5] L. Bergamaschi, R. Bru, A. Martinez and M. Putti, *Quasi-Newton preconditioners for the inexact Newton method*, *Electronic Transactions on Numerical Analysis*, **23** (2006), 76–87.
[6] L. Bergamaschi and A. Martinez, *Efficiently preconditioned inexact Newton methods for large symmetric eigenvalue problems*, arXiv:1312.1553v1 [math.NA], 5 Dec, 2013.
[7] D. R. Fokkema, G. L. G. Sleijpen and H. A. Van der Vorst, *Accelerated inexact Newton schemes for large systems of nonlinear equations*, *SIAM Journal on Scientific Computing*, **19** (1997), 657–674.
[8] M. A. Freitag and A. Spence, *Rayleigh quotient iteration and simplified Jacobi-Davidson method with preconditioned iterative solvers*, *Linear Algebra Appl.*, **428** (2008), 2049–2060.
[9] G. Hagerty, *Finding a Few Eigenvalues of Large Sparse Nonsymmetric Matrices*, Ph.D Thesis, Washington State University, 2001.
[10] M. E. Hochstenbach and Y. Notay, *The Jacobi-Davidson method*, *GAMM-Mitteilungen*, **29** (2006), 368–382.
[11] J. Pestana and A. J. Wathen, *Natural preconditioning and iterative methods for saddle point problems*, *SIAM Rev.*, **57** (2015), 71–91.
[12] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS publishing, Boston, 1996.
[13] G. L. Sleijpen and H. A. Van Der Vorst, *A Jacobi-Davidson method for linear eigenvalue problems*, *SIAM J. Matrix Anal.*, **17** (1996), 401–425.
[14] G. L. Sleijpen and H. A. Van Der Vorst, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, *BIT*, **42** (2002), 159–182.
[15] G. Wu and L. Zhang, *On expansion of search subspaces for large non-Hermitian eigenproblems*, *J. Linear Algebra and Applications*, **454** (2014), 107–129.
[16] K. Wu, Y. Saad and A. Stathopoulos, *Inexact Newton preconditioning techniques for large symmetric eigenvalue problems*, *J. Electronic Transactions on Numerical Analysis*, **7** (1998), 202–214.

Received January 2021; 1st revision February 2021; Final revision March 2021.

E-mail address: nuahmy@163.com
E-mail address: wangli1@njnu.edu.cn